

www.vuCTn.com

CS614 MERGED

PPT Slides 1 TO 40

Regards : Zarva Chaudhary

Merged File date 22/12/2022

Admin:

* Zarva Chaudhary *

* Chandhary Moazzam *

* Laiba Maki *

Virtual University of Pakistan

Data Warehousing

Lecture-1

Introduction and Background

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

Introduction and Background

Reference Books

- W. H. Inmon, **Building the Data Warehouse (Second Edition)**, John Wiley & Sons Inc., NY.
- A. Abdullah, “**Data Warehousing for beginners: Concepts & Issues**” (First Edition).
- Paulraj Ponniah, **Data Warehousing Fundamentals**, John Wiley & Sons Inc., NY.

Additional Material

- Research Papers
- Magazine Articles

Summary of course

Topics (Total Lectures = 45)

1. Introduction & Background
2. De-normalization
3. On Line Analytical Processing (OLAP)
4. Dimensional modeling
5. Extract – Transform – Load (ETL)
6. Data Quality Management (DQM)
7. Need for speed (Parallelism, Join and Indexing techniques)
8. Data Mining
9. DWH Implementation steps
10. Complete implementation case study
11. Lab and tool usage
12. Others

Summary of course

Topics

1. Introduction & Background
2. De-normalization
3. On Line Analytical Processing (OLAP)
4. Dimensional modeling

Summary of course

Topics

5. Extract – Transform – Load (ETL)

6. Data Quality Management (DQM)

7. Need for speed (Parallelism, Join and Indexing techniques)

8. Data Mining

9. DWH Implementation steps

Summary of course

Topics

10. Complete implementation case study

11. Lab and tool usage

12. Others

Semester Project

Develop an application for an organization of your choice.

A case study and coding based approach to be followed.

Use 4GL or a high level programming language.

You MUST collect the necessary data and should have a first draft of the project description approved by the instructor **BEFORE** initiating on detailed work.

Semester Project (Cont...)

The project report to include, but is not limited to, the following as documentation:

- Narrative description of business and tables of appropriate data.
- Descriptions of decisions to be supported by information produced by system.
- Summary narrative of results produced.
- Structure charts, dataflow diagrams and/or other diagrams to document the structure of the system.
- Listings of computer models/programs utilized.
- Reports displaying results.
- Recommended decision from results.
- User instructions.

Approach of the course

- Develop an understanding of underlying RDBMS concepts.
- Apply these concepts to VLDB DSS environments and understand where and why they break down?
- Expose the differences between RDBMS and Data Warehouse in the context of VLDB.
- Provide the basics of DSS tools such as OLAP, Data Mining and demonstrate their application.
- Demonstrate the application of DSS concepts and limitations of the OLTP concepts through lab exercises.

Why this course?

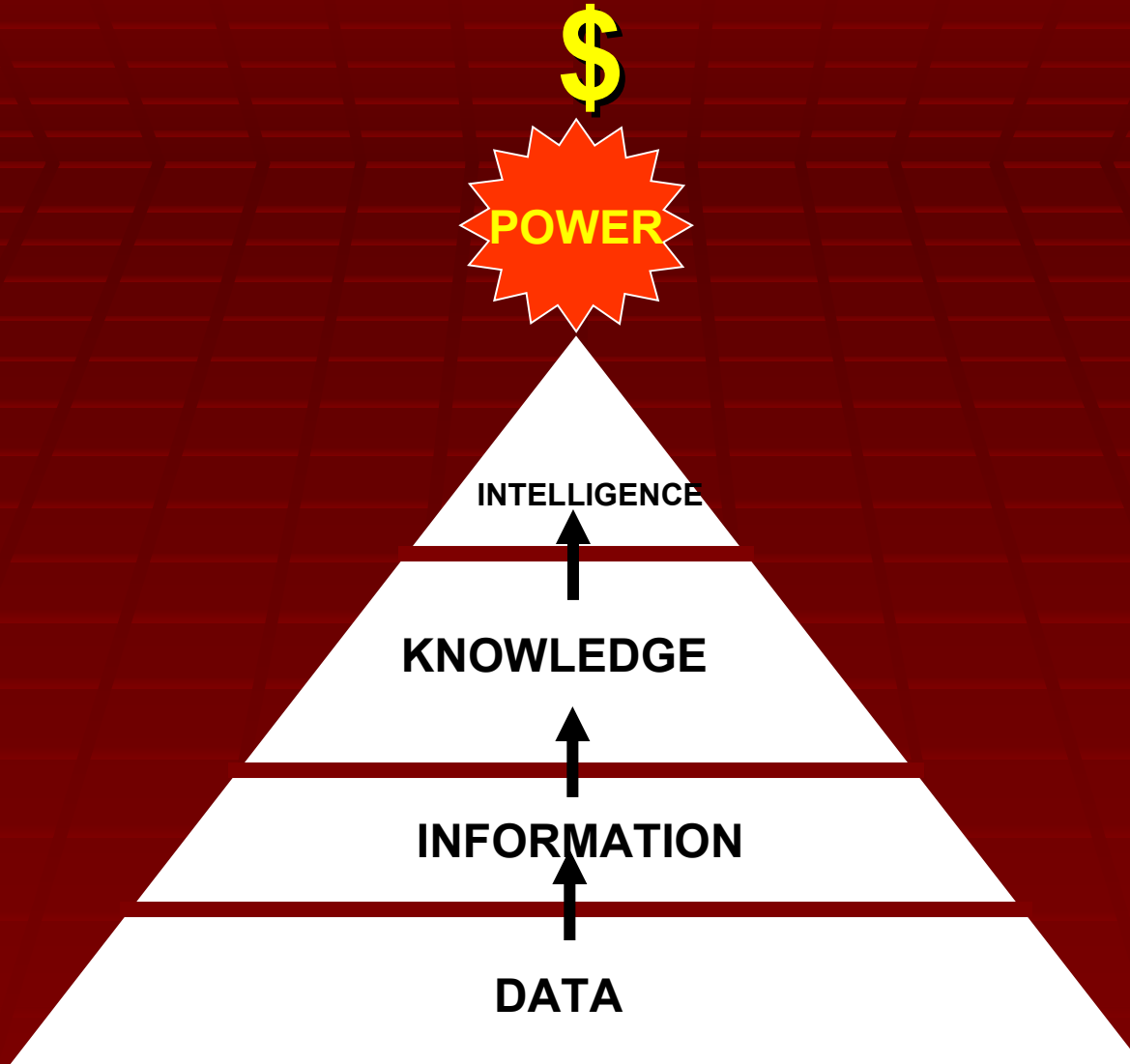
- The world is changing (actually changed), either change or be left behind.
- Missing the opportunities or going in the wrong direction has prevented us from growing.
- **What is the right direction?**
- Harnessing the data, in a knowledge driven economy.

The need

“Drowning in data and starving
for information”

Knowledge is power, Intelligence
is absolute power!

The need



Historical overview

1960

Master Files & Reports



1965

Lots of Master files!



1970

Direct Access Memory & DBMS



1975

Online high performance transaction processing



Historical overview

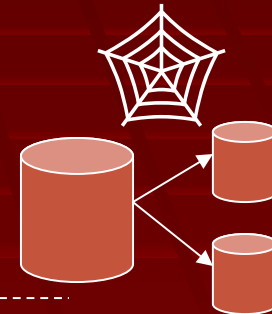
1980

PCs and 4GL Technology (MIS/DSS)



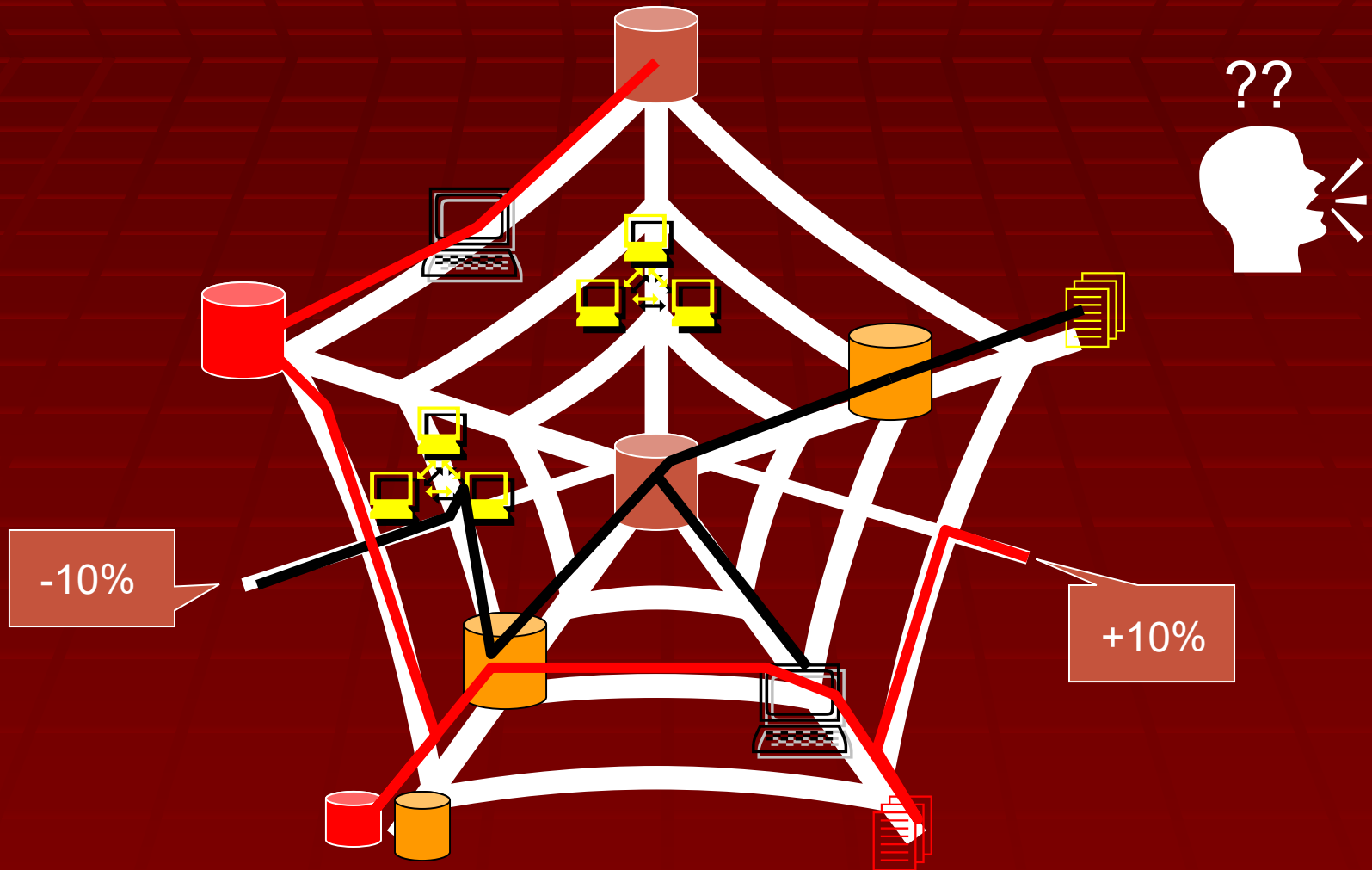
1985 & 1990

Extract programs, extract processing,
The legacy system's web



Historical overview: Crisis of Credibility

What is the financial health of our company?



Virtual University of Pakistan

Data Warehousing

Lecture-2

Introduction and Background

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

Introduction and Background

Why a Data Warehouse (DWH)?

- Data recording and storage is growing.
- History is excellent predictor of the future.
- Gives total view of the organization.
- Intelligent decision-support is required for decision-making.

Reason-1: Why a Data Warehouse?

- Data Sets are growing.

How Much Data is that?

Reason-1: Why a Data Warehouse?

- Size of Data Sets are going up ↑.
- Cost of data storage is coming down ↓.
 - The amount of data average business collects and stores is **doubling every year**
- Total hardware and software cost to store and manage **1 Mbyte** of data
 - 1990: ~ \$15
 - 2002: ~ ¢15 (Down 100 times)
 - By 2007: < ¢1 (Down 150 times)

Reason-1: Why a Data Warehouse?

- A Few Examples
 - WalMart: 24 TB
 - France Telecom: ~ 100 TB
 - CERN: Up to 20 PB by 2006
 - Stanford Linear Accelerator Center (SLAC): 500TB

Caution!

A Warehouse of Data
is NOT a
Data Warehouse

Caution!

Size
is NOT
Everything

Reason-2: Why a Data Warehouse?

- Businesses demand Intelligence (BI).
 - Complex questions from integrated data.
 - “Intelligent Enterprise”

Reason-2: Why a Data Warehouse?

DBMS Approach

List of all items that were sold last month?

List of all items purchased by Tariq Majeed?

The total sales of the last month grouped by branch?

How many sales transactions occurred during the month of January?

Reason-2: Why a Data Warehouse?

Intelligent Enterprise

Which items sell together? Which items to stock?

**Where and how to place the items?
What discounts to offer?**

How best to target customers to increase sales at a branch?

Which customers are most likely to respond to my next promotional campaign, and why?

Reason-3: Why a Data Warehouse?

- Businesses want much more...

- What happened?
- Why it happened?
- What will happen?
- What is happening?
- What do you want to happen?

Stages of
Data
Warehouse



What is a Data Warehouse?

A complete repository of historical corporate data extracted from transaction systems that is available for ad-hoc access by knowledge workers.

What is a Data Warehouse?

Complete repository

History

Transaction System

Ad-Hoc access

Knowledge workers

What is a Data Warehouse?

Transaction System

- Management Information System (MIS)
- Could be typed sheets (NOT transaction system)

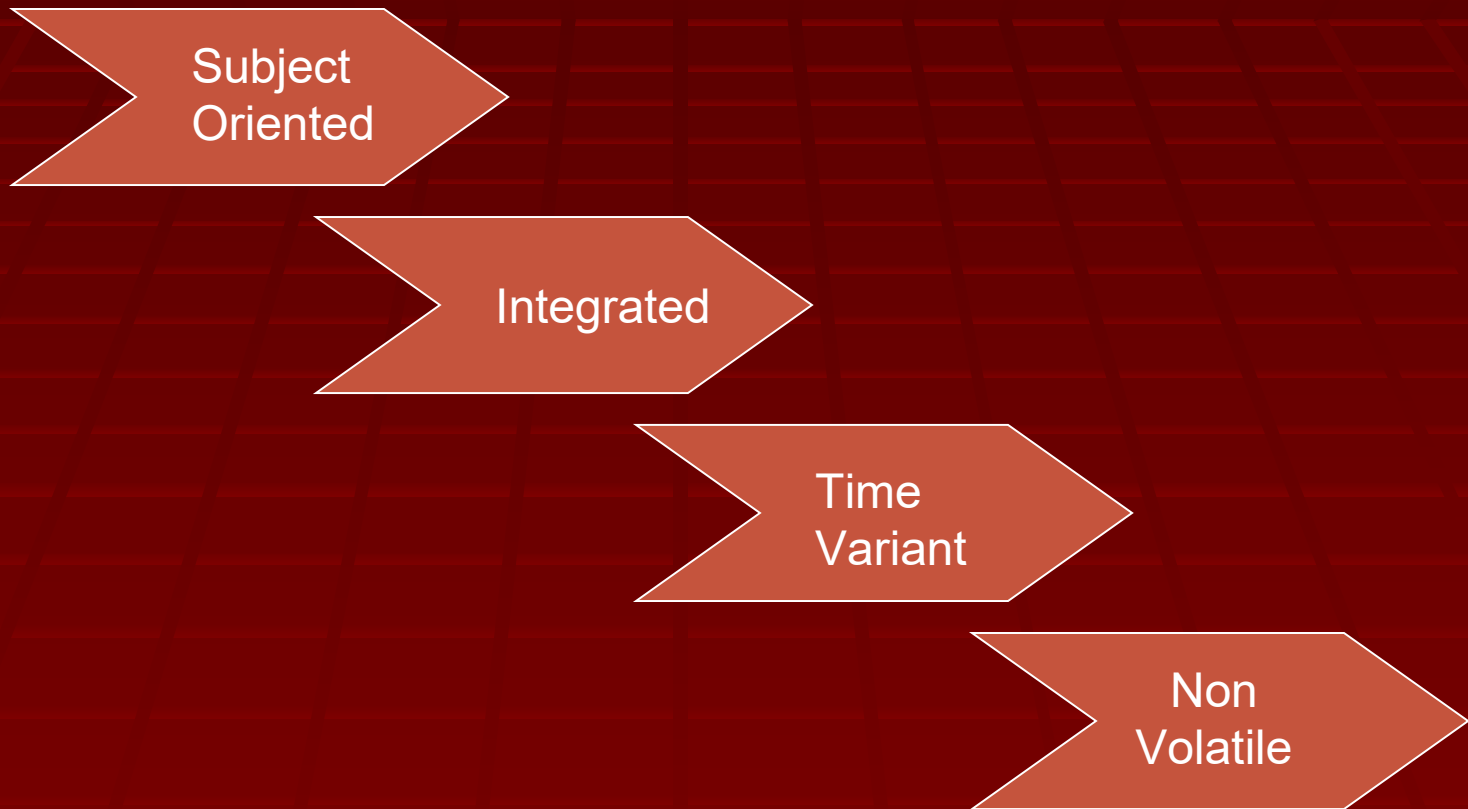
Ad-Hoc access

- Dose not have a certain access pattern.
- Queries not known in advance.
- Difficult to write SQL in advance.

Knowledge workers

- Typically NOT IT literate (Executives, Analysts, Managers).
- NOT clerical workers.
- Decision makers.

Another View of a DWH



Virtual University of Pakistan

Data Warehousing

Lecture-3

Introduction and Background

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

Introduction and Background

What is a Data Warehouse ?

It is a blend of many technologies, the basic concept being:

- Take all data from different operational systems.
- If necessary, add relevant data from industry.
- Transform all data and bring into a uniform format.
- Integrate all data as a single entity.

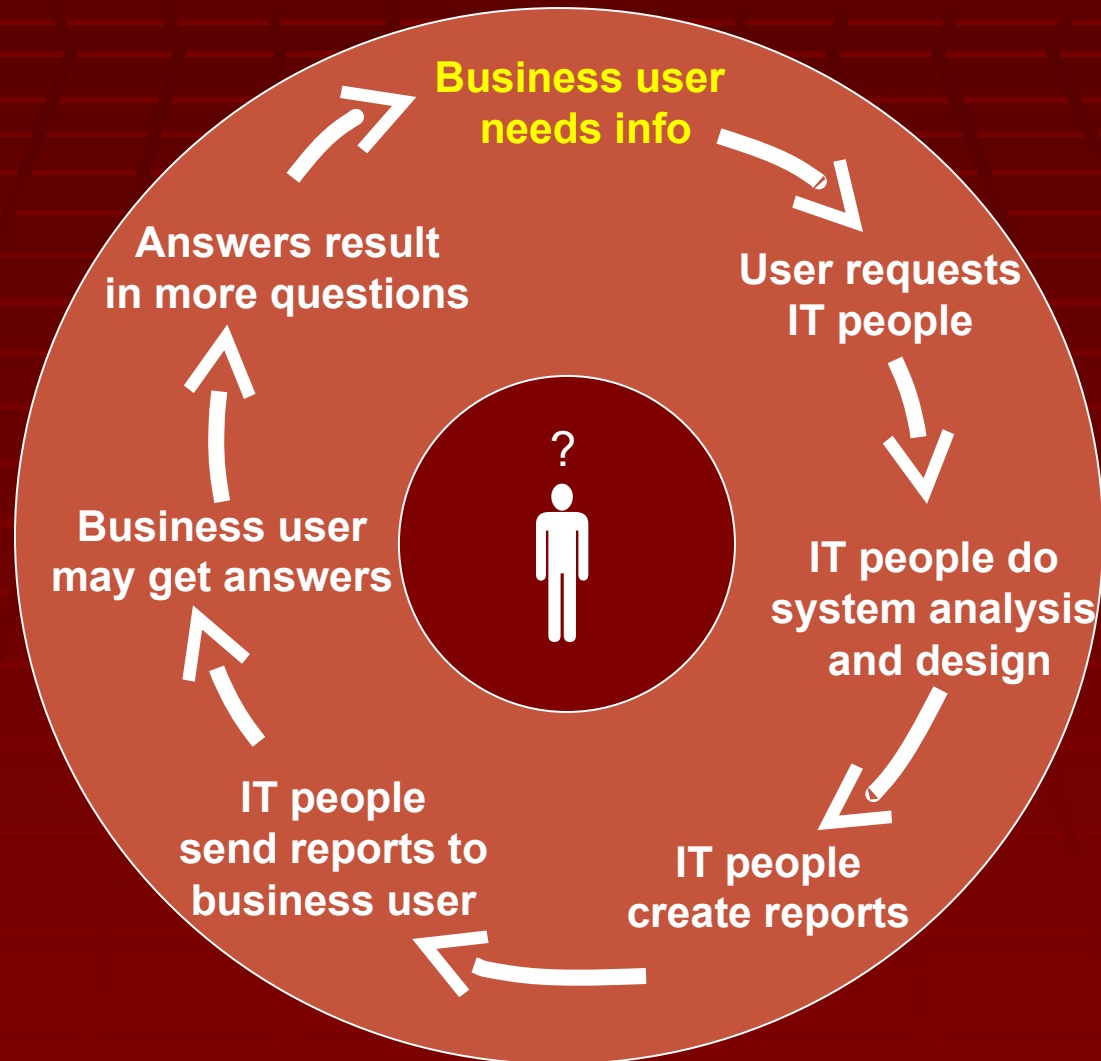
What is a Data Warehouse ? (Cont...)

It is a blend of many technologies, the basic concept being:

- Store data in a format supporting easy access for decision support.
- Create performance enhancing indices.
- Implement performance enhancement joins.
- Run ad-hoc queries with low selectivity.

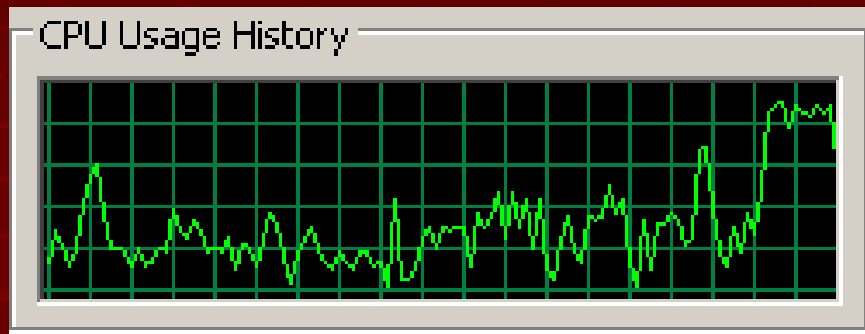
How is it Different?

- Fundamentally different

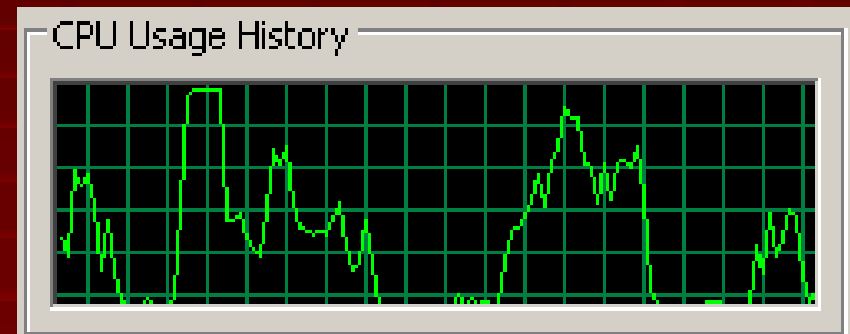


How is it Different?

- Different patterns of hardware utilization



Operational



DWH

Bus Service vs. Train

How is it Different?

- Combines operational and historical data.
 - Don't do data entry into a DWH, OLTP or ERP are the source systems.
 - OLTP systems don't keep history, cant get balance statement more than a year old.
 - DWH keep historical data, even of bygone customers. Why?
 - In the context of bank, want to know why the customer left?
 - What were the events that led to his/her leaving? Why?
 - Customer retention.

How much history?

- Depends on:
 - Industry.
 - Cost of storing historical data.
 - Economic value of historical data.

How much history?

- Industries and history
 - **Telecomm** calls are much much more as compared to bank transactions- **18 months**.
 - **Retailers** interested in analyzing yearly seasonal patterns- **65 weeks**.
 - **Insurance** companies want to do actuary analysis, use the historical data in order to predict risk- **7 years**.

How much history?

Economic value of data

Vs.

Storage cost

**Data Warehouse a
complete repository of data?**

How is it Different?

- Usually (but not always) periodic or batch updates rather than real-time.
- The boundary is blurring for active data warehousing.
- For an ATM, if update not in real-time, then lot of real trouble.
- DWH is for strategic decision making based on historical data. Wont hurt if transactions of last one hour/day are absent.

How is it Different?

- Rate of update depends on:
 - volume of data,
 - nature of business,
 - cost of keeping historical data,
 - benefit of keeping historical data.

Virtual University of Pakistan

Data Warehousing

Lecture-4

Introduction and Background

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

Introduction and Background

How is it Different?

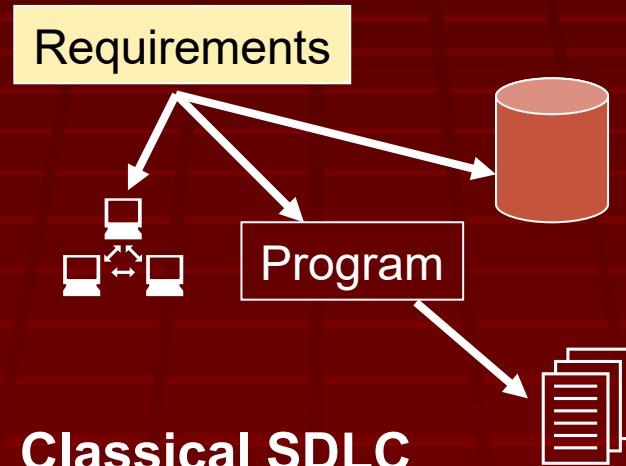
- Starts with a 6x12 availability requirement ... but 7x24 usually becomes the goal.
- Decision makers typically don't work 24 hrs a day and 7 days a week. An ATM system does.
- Once decision makers start using the DWH, and start reaping the benefits, they start liking it...
- Start using the DWH more often, till want it available 100% of the time.

How is it Different?

- Starts with a 6x12 availability requirement ... but 7x24 usually becomes the goal.
 - For business across the globe, 50% of the world may be sleeping at any one time, but the businesses are up 100% of the time.
 - 100% availability not a trivial task, need to take into account loading strategies, refresh rates etc.

How is it Different?

- Does not follow the traditional development model

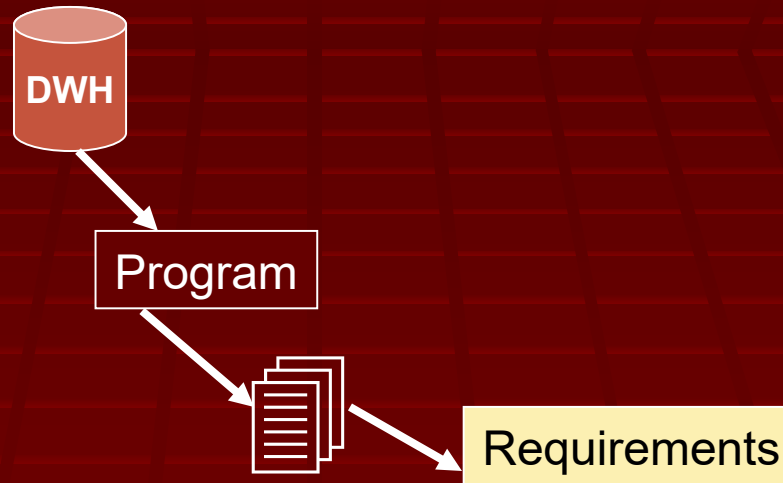


Classical SDLC

- Requirements gathering
- Analysis
- Design
- Programming
- Testing
- Integration
- Implementation

How is it Different?

- Does not follow the traditional development model



DWH SDLC (**CLDS**)

- Implement warehouse
- Integrate data
- Test for biasness
- Program w.r.t data
- Design DSS system
- Analyze results
- Understand requirement

Data Warehouse Vs. OLTP

OLTP (On Line Transaction Processing)

```
Select tx_date, balance from tx_table  
Where account_ID = 23876;
```

Data Warehouse Vs. OLTP

DWH

Select balance, age, sal, gender from
customer_table, tx_table

Where age between (30 and 40) and

Education = 'graduate' and

CustID.customer_table =

Customer_ID.tx_table;

Data Warehouse Vs. OLTP

OLTP	DWH
Primary key used	Primary key NOT used
No concept of Primary Index	Primary index used
Few rows returned	Many rows returned
May use a single table	Uses multiple tables
High selectivity of query	Low selectivity of query
Indexing on primary key (unique)	Indexing on primary index (non-unique)

Data Warehouse Vs. OLTP

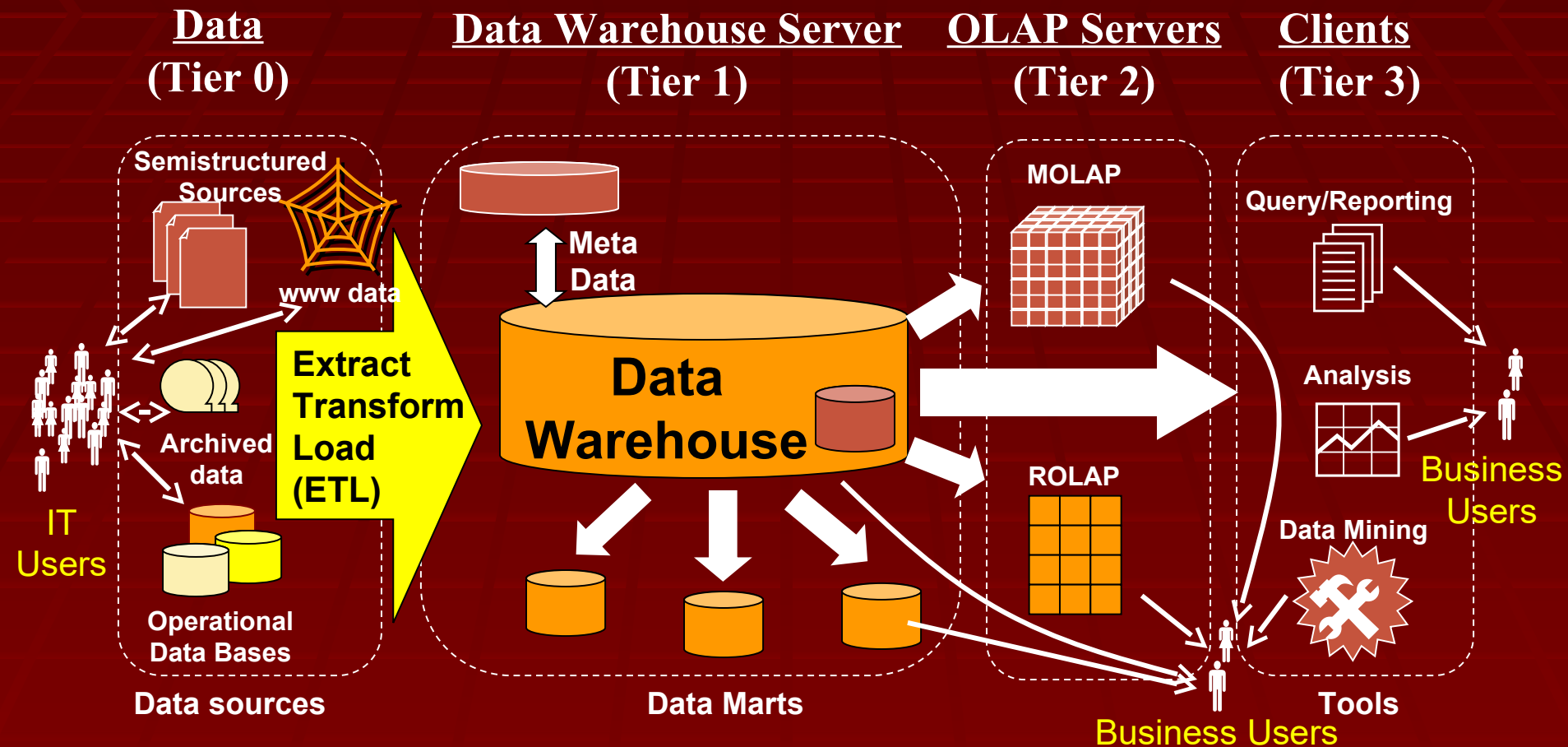
OLTP: OnLine Transaction Processing (MIS or Database System)

	Data Warehouse	OLTP
Scope	* Application –Neutral * Single source of “truth” * Evolves over time * How to improve business	* Application specific * Multiple databases with repetition * Off the shelf application * Runs the business
Data Perspective	* Historical, detailed data * Some summary * Lightly denormalized	* Operational data * No summary * Fully normalized
Queries	* Hardly uses PK * Number of results returned in thousands	* Based on PK * Number of results returned in hundreds
Time factor	* Minutes to hours * Typical availability 6x12	* Sub seconds to seconds * Typical availability 24x7

Comparison of Response Times

- On-line analytical processing (OLAP) queries must be executed in a small number of seconds.
 - Often requires denormalization and/or sampling.
- Complex query scripts and large list selections can generally be executed in a small number of minutes.
- Sophisticated clustering algorithms (e.g., data mining) can generally be executed in a small number of hours (even for hundreds of thousands of customers).

Putting the pieces together



Virtual University of Pakistan

Data Warehousing

Lecture-5

Types & Typical Applications of DWH

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

Types & Typical Applications of DWH

Types of data warehouse

- **Financial**
- **Telecommunication**
- **Insurance**
- Human Resource
- Global
- Exploratory

Types of data warehouse

Financial

- **First data warehouse that an organization builds. This is appealing because:**
 - **Nerve center, easy to get attention.**
 - **In most organizations, smallest data set.**
 - **Touches all aspects of an organization, with a common denomination i.e. money.**
 - **Inherent structure of data directly influenced by the day-to-day activities of financial processing.**

Word of caution, will discuss, if and when time permits.

Types of data warehouse

Telecommunication

Dominated by sheer volume of data.

Many ways to accommodate call level detail:

- Only a few months of call level detail,
- Storing lots of call level detail scattered over different storage media,
- Storing only selective call level detail, etc.
- Unfortunately, for many kinds of processing, working at an aggregate level is simply not possible.

Types of data warehouse

Insurance

Insurance data warehouses are similar to other data warehouses BUT with a few exceptions.

Stored data that is very, very old, used for actuarial processing.

Typical business may change dramatically over last 40-50 years, but not insurance.

In retailing or telecomm there are a few important dates, but in the insurance environment there are many dates of many kinds.

Types of data warehouse

Insurance

Insurance data warehouses are similar to other data warehouses BUT with a few exceptions.

Long operational business cycles, in years.
Processing time in months. Thus the operating speed is different.

Transactions are not gathered and processed, but are in kind of “frozen”.

Thus a very unique approach of design & implementation.

Typical Applications

Impact on organization's core business is to streamline and maximize profitability.

- Fraud detection.
- Profitability analysis.
- Direct mail/database marketing.
- Credit risk prediction.
- Customer retention modeling.
- Yield management.
- Inventory management.

ROI on any one of these applications can justify HW/SW & consultancy costs in most organizations.

Typical Applications

Fraud detection

- By observing data usage patterns.
- People have typical purchase patterns.
- Deviation from patterns.
- Certain cities notorious for fraud.
- Certain items bought by stolen cards.
- Similar behavior for stolen phone cards.

Typical Applications

Profitability Analysis

- Banks know if they are profitable or not.
- Don't know which customers are profitable.
- Typically more than 50% are NOT profitable.
- Don't know which one?
- Balance is not enough, transactional behavior is the key.
- Restructure products and pricing strategies.
- Life-time profitability models (next 3-5 years).

Typical Applications

Direct mail marketing

- Targeted marketing.
- Offering high bandwidth package NOT to all users.
- Know from call detail records of web surfing.
- Saves marketing expense, saving pennies.
- Knowing your customers better.

Typical Applications

Credit risk prediction

- Who should get a loan?
- Customer segregation i.e. stable vs. rolling.
- Qualitative decision making NOT subjective.
- Different interest rates for different customers.
- Do not subsidize bad customer on the basis of good.

Typical Applications

Yield Management

- Works for fixed inventory businesses.
- The price of item suddenly goes to zero.
- Item prices vary for varying customers.
- Example: Air Lines, Hotels etc.
- Price of (say) Air Ticket depends on:
 - How much in advance ticket was bought?
 - How many vacant seats were present?
 - How profitable is the customer?
 - Ticket is one-way or return?

Recent Application

Agriculture Systems

- Agri and related data collected for decades.
- Metrological data consists of 50+ attributes.
- Decision making based on expert judgment.
- Lack of integration results in underutilization.
- What is required, in which amount and when?

Virtual University of Pakistan

Data Warehousing

Lecture-6 Normalization

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

Normalization

Normalization

What is normalization?

What are the goals of normalization?

- Eliminate redundant data.
- Ensure data dependencies make sense.

What is the result of normalization?

What are the levels of normalization?

Always follow purists approach of normalization?

NO

Normalization

Consider a student database system to be developed for a multi-campus university, such that it specializes in one degree program at a campus i.e. BS, MS or PhD.

SID	Degree	Campus	Course	Marks
1	BS	Islamabad	CS-101	30
1	BS	Islamabad	CS-102	20
1	BS	Islamabad	CS-103	40
1	BS	Islamabad	CS-104	20
1	BS	Islamabad	CS-105	10
1	BS	Islamabad	CS-106	10
2	MS	Lahore	CS-101	30
2	MS	Lahore	CS-102	40
3	MS	Lahore	CS-102	20
4	BS	Islamabad	CS-102	20
4	BS	Islamabad	CS-104	30
4	BS	Islamabad	CS-105	40

SID: Student ID

Degree: Registered as BS or MS student

Campus: City where campus is located

Course: Course taken

Marks: Score out of max of 50

Normalization: 1NF

Only contains atomic values, BUT also contains redundant data.

FIRST

SID	Degree	Campus	Course	Marks
1	BS	Islamabad	CS-101	30
1	BS	Islamabad	CS-102	20
1	BS	Islamabad	CS-103	40
1	BS	Islamabad	CS-104	20
1	BS	Islamabad	CS-105	10
1	BS	Islamabad	CS-106	10
2	MS	Lahore	CS-101	30
2	MS	Lahore	CS-102	40
3	MS	Lahore	CS-102	20
4	BS	Islamabad	CS-102	20
4	BS	Islamabad	CS-104	30
4	BS	Islamabad	CS-105	40

Normalization: 1NF

Update anomalies

INSERT. Certain student with SID 5 got admission in a different campus (say) Karachi cannot be added until the student registers for a course.

DELETE. If student graduates and his/her corresponding record is deleted, then all information about that student is lost.

UPDATE. If student migrates from Islamabad campus to Lahore campus (say) SID = 1, then six rows would have to be updated with this new information.

Normalization: 2NF

Every non-key column is fully dependent on the PK

FIRST is in 1NF but not in 2NF because degree and campus are functionally dependent upon only on the column SID of the composite key (SID, course). This can be illustrated by listing the functional dependencies in the table:

SID $\longrightarrow$ campus, degree	}	SID & Campus are NOT unique
campus $\longrightarrow$ degree		
(SID, Course) $\longrightarrow$ Marks		

To transform the table FIRST into 2NF we move the columns SID, Degree and Campus to a new table called REGISTRATION. The column SID becomes the primary key of this new table.

Normalization: 2NF

REGISTRATION

SID	Degree	Campus
1	BS	Islamabad
2	MS	Lahore
3	MS	Lahore
4	BS	Islamabad
5	PhD	Peshawar

SID is now a PK

PERFORMANCE

SID	Course	Marks
1	CS-101	30
1	CS-102	20
1	CS-103	40
1	CS-104	20
1	CS-105	10
1	CS-106	10
2	CS-101	30
2	CS-102	40
3	CS-102	20
4	CS-102	20
4	CS-104	30
4	CS-105	40

PERFORMANCE in 2NF as (SID, Course) uniquely identify Marks

Normalization: 2NF

Presence of modification anomalies for tables in 2NF. For the table REGISTRATION, they are:

- **INSERT:** Until a student gets registered in a degree program, that program cannot be offered!
- **DELETE:** Deleting any row from REGISTRATION destroys all other facts in the table.

Why there are anomalies?

The table is in 2NF but NOT in 3NF

Normalization: 3NF

All columns must be dependent only on the primary key.

Table PERFORMANCE is already in 3NF. The non-key column, marks, is fully dependent upon the primary key (SID, degree).

REGISTRATION is in 2NF but not in 3NF because it contains a transitive dependency.

A transitive dependency occurs when a non-key column that is a determinant of the primary key is the determinate of other columns.

The concept of a transitive dependency can be illustrated by showing the functional dependencies in REGISTRATION:

**REGISTRATION.SID —> REGISTRATION.Degree
REGISTRATION.SID —> REGISTRATION.Campus
REGISTRATION.Campus —> REGISTRATION.Degree**

Note that REGISTRATION.Degree is determined both by the primary key SID and the non-key column campus.

Normalization: 3NF

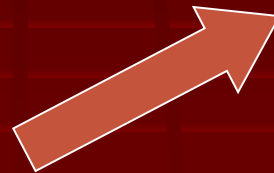
To transform REGISTRATION into 3NF, we create a new table called CAMPUS_DEGREE and move the columns campus and degree into it.

Degree is deleted from the original table, campus is left behind to serve as a foreign key to CAMPUS_DEGREE, and the original table is renamed to STUDENT_CAMPUS to reflect its semantic meaning.

Normalization: 3NF

REGISTRATION

SID	Degree	Campus
1	BS	Islamabad
2	MS	Lahore
3	MS	Lahore
4	BS	Islamabad
5	PhD	Peshawar



STUDENT_CAMPUS

SID	Campus
1	Islamabad
2	Lahore
3	Lahore
4	Islamabad
5	Peshawar



CAMPUS_DEGREE

Campus	Degree
Islamabad	BS
Lahore	MS
Peshawar	PhD

Normalization: 3NF

Removal of anomalies and improvement in queries as follows:

- **INSERT:** Able to first offer a degree program, and then students registering in it.
- **UPDATE:** Migrating students between campuses by changing a single row.
- **DELETE:** Deleting information about a course, without deleting facts about all columns in the record.

Normalization

Conclusions:

- Normalization guidelines are cumulative.
- Generally a good idea to only ensure 2NF.
- 3NF is at the cost of simplicity and performance.
- There is a 4NF with no multi-valued dependencies.
- There is also a 5NF.

Virtual University of Pakistan

Data Warehousing

Lecture-7 De-normalization

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

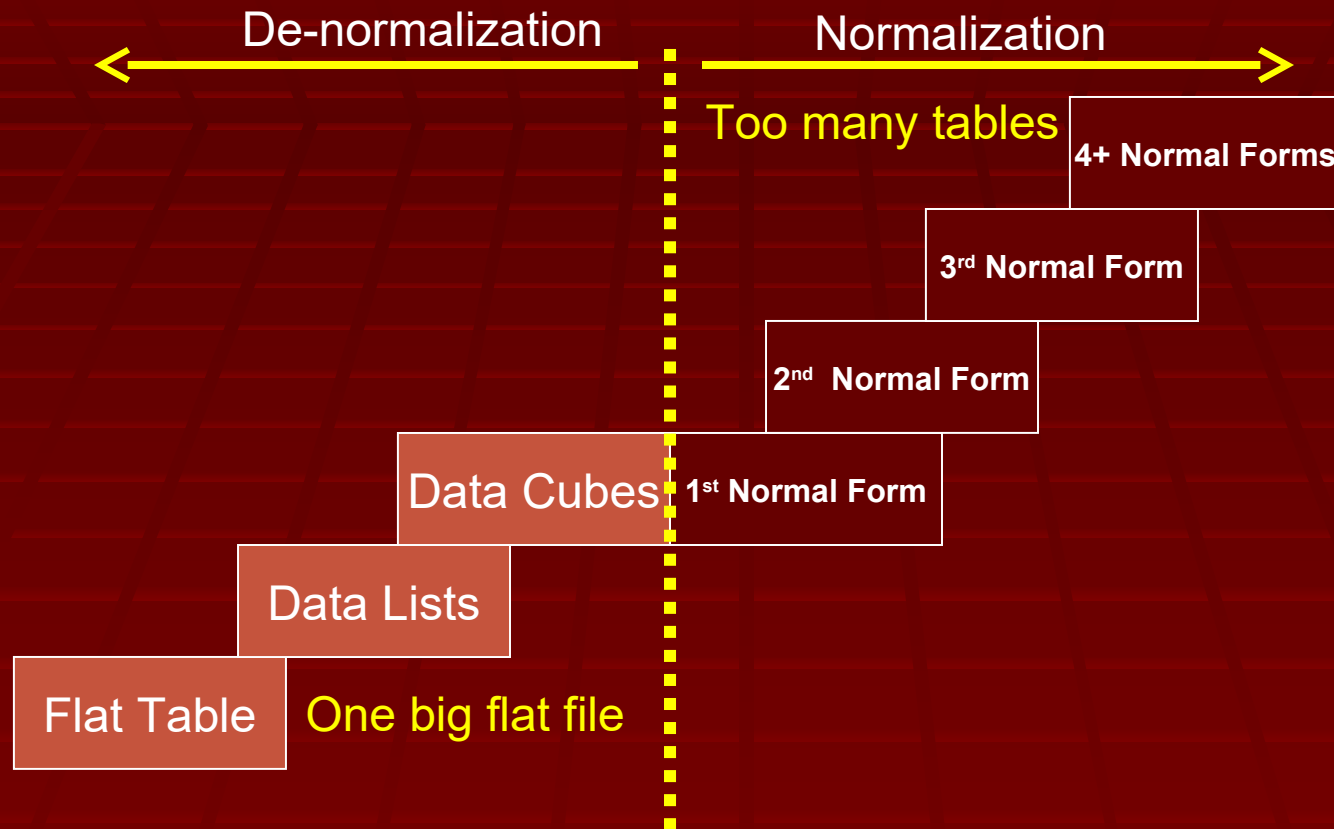
www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

De-normalization

Striking a balance between “good” & “evil”



What is De-normalization?

- It is not chaos, more like a “controlled crash” with the aim of performance enhancement without loss of information.
- Normalization is a rule of thumb in DBMS, but in DSS ease of use is achieved by way of denormalization.
- De-normalization comes in many flavors, such as combining tables, splitting tables, adding data etc., but all done very carefully.

Why De-normalization In DSS?

- Bringing “close” dispersed but related data items.
- Query performance in DSS significantly dependent on physical data model.
- Very early studies showed performance difference in orders of magnitude for different number de-normalized tables and rows per table.
- The level of de-normalization should be carefully considered.

How De-normalization improves performance?

De-normalization specifically improves performance by either:

- Reducing the number of tables and hence the reliance on joins, which consequently speeds up performance.
- Reducing the number of joins required during query execution, or
- Reducing the number of rows to be retrieved from the Primary Data Table.

4 Guidelines for De-normalization

1. Carefully do a cost-benefit analysis (frequency of use, additional storage, join time).
2. Do a data requirement and storage analysis.
3. Weigh against the maintenance issue of the redundant data (triggers used).
4. When in doubt, don't denormalize.

Areas for Applying De-Normalization Techniques

- Dealing with the abundance of star schemas.
- Fast access of time series data for analysis.
 - Fast aggregate (sum, average etc.) results and complicated calculations.
 - Multidimensional analysis (e.g. geography) in a complex hierarchy.
- Dealing with few updates but many join queries.

De-normalization will ultimately affect the database size and query performance.

Five principal De-normalization techniques

1. Collapsing Tables.

- Two entities with a One-to-One relationship.
- Two entities with a Many-to-Many relationship.

2. Splitting Tables (Horizontal/Vertical Splitting).

3. Pre-Joining.

4. Adding Redundant Columns (Reference Data).

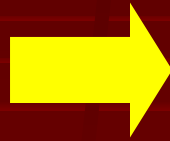
5. Derived Attributes (Summary, Total, Balance etc).

Collapsing Tables

normalized

CoIA	CoIB

CoIA	CoIC



denormalized

CoIA	CoIB	CoIC

- Reduced storage space.
- Reduced update time.
- Does not changes business view.
- Reduced foreign keys.
- Reduced indexing.

Virtual University of Pakistan

Data Warehousing

Lecture-8

De-normalization Techniques

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

De-normalization Techniques

Splitting Tables

Table

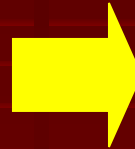
CoIA	CoIB	CoIC

Table_v1

CoIA	CoIB

Table_v2

CoIA	CoIC



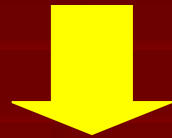
Vertical Split

Table_h1

CoIA	CoIB	CoIC

Table_h2

CoIA	CoIB	CoIC



Horizontal split

Ahsan Abdullah

Splitting Tables: Horizontal splitting...

Breaks a table into multiple tables based upon common column values. Example: Campus specific queries.

GOAL

- **Spreading rows for exploiting parallelism.**
- **Grouping data to avoid unnecessary query load in WHERE clause.**

Splitting Tables: Horizontal splitting

ADVANTAGE

- Enhance security of data.
- Organizing tables differently for different queries.
- Graceful degradation of database in case of table damage.
- Fewer rows result in flatter B-trees and fast data retrieval.

Splitting Tables: Vertical Splitting

- Infrequently accessed columns become extra “baggage” thus degrading performance.
- Very useful for rarely accessed large text columns with large headers.
- Header size is reduced, allowing more rows per block, thus reducing I/O.
- Splitting and distributing into separate files with repeating primary key.
- For an end user, the split appears as a single table through a view.

Pre-joining ...

- Identify frequent joins and append the tables together in the physical data model.
- Generally used for 1:M such as master-detail. RI is assumed to exist.
- Additional space is required as the master information is repeated in the new header table.

Pre-Joining...

Master

Sale_ID	Sale_date	Sale_person

1

M

Tx_ID	Sale_ID	Item_ID	Item_Qty	Sale_Rs

Detail



denormalized

Tx_ID	Sale_ID	Sale_date	Sale_person	Item_ID	Item_Qty	Sale_Rs

Pre-Joining: Typical Scenario

Typical of Market basket query

Join ALWAYS required

Tables could be millions of rows

Squeeze Master into Detail

Repetition of facts. How much?

Detail 3-4 times of master

Adding Redundant Columns...

Table_1

ColA	ColB

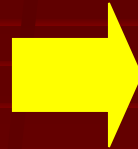


Table_1'

ColA	ColB	ColC

Table_2

ColA	ColC	ColD	...	ColZ

Table_2

ColA	ColC	ColD	...	ColZ

Adding Redundant Columns...

Columns can also be moved, instead of making them redundant. Very similar to pre-joining as discussed earlier.

EXAMPLE

Frequent referencing of code in one table and corresponding description in another table.

- A join is required.
- To eliminate the join, a redundant attribute added in the target entity which is functionally independent of the primary key.

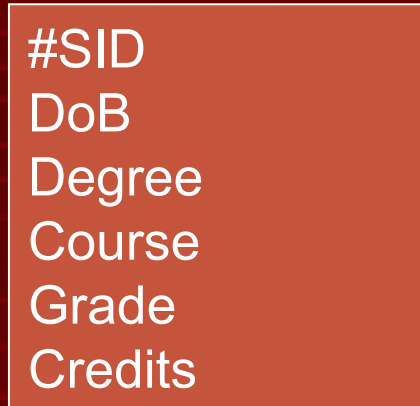
Redundant Columns: Surprise

Note that:

- Actually increases in storage space, and increase in update overhead.
- Keeping the actual table intact and unchanged helps enforce RI constraint.
- Age old debate of RI ON or OFF.

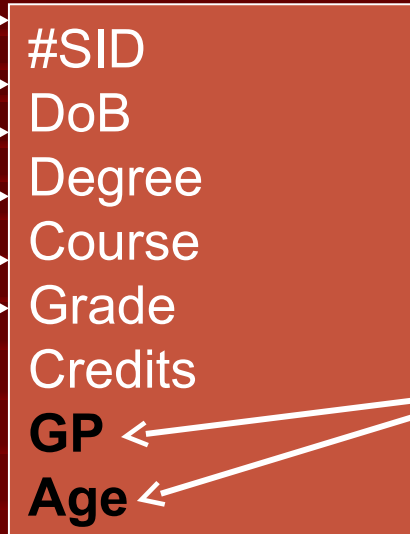
Derived Attributes: Example

Business Data Model



DoB: Date of Birth

DWH Data Model



Derived attributes

- Calculated once
- Used Frequently

Age is also a derived attribute, calculated as *Current_Date* – *DoB* (calculated periodically).

GP (Grade Point) column in the data warehouse data model is included as a derived value. The formula for calculating this field is $\text{Grade} * \text{Credits}$.

Virtual University of Pakistan

Data Warehousing

Lecture-9

Issues of De-normalization

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

Issues of De-normalization

Why Issues?

Issues of Denormalization

- Storage
- Performance
- Ease-of-use
- Maintenance

Industry Characteristics

Master:Detail Ratios

- Health care 1:2 ratio
- Video Rental 1:3 ratio
- Retail 1:30 ratio

Storage Issues: Pre-joining Facts

- Assume 1:2 record count ratio between claim master and detail for health-care application.
- Assume 10 million members (20 million records in claim detail).
- Assume 10 byte member_ID.
- Assume 40 byte header for master and 60 byte header for detail tables.

Storage Issues: Pre-joining (Calculations)

With normalization:

Total space used = $10 \times 40 + 20 \times 60 = 1.6 \text{ GB}$

After denormalization:

Total space used = $(60 + 40 - 10) \times 20 = 1.8 \text{ GB}$

Net result is 12.5% additional space required in raw data table size for the database.

Performance Issues: Pre-joining

Consider the query “How many members were paid claims during last year?”

With normalization:

Simply count the number of records in the master table.

After denormalization:

The member_ID would be repeated, hence need a count distinct. This will cause sorting on a larger table and degraded performance.

Why Performance Issues: Pre-joining

Depending on the query, the performance actually deteriorates with denormalization! This is due to the following three reasons:

- **Forcing a sort due to count distinct.**
- **Using a table with 1.5 times header size.**
- **Using a table which is 2 times larger.**
- **Resulting in 3 times degradation in performance.**

Bottom Line: Other than 0.2 GB additional space, also keep the 0.4 GB master table.

Performance Issues: Adding redundant columns

Continuing with the previous Health-Care example, assuming a 60 byte detail table and 10 byte Sale_Person.

- Copying the Sale_Person to the detail table results in all scans taking 16% longer than previously.
- Justifiable only if significant portion of queries get benefit by accessing the denormalized detail table.
- Need to look at the cost-benefit trade-off for each denormalization decision.

Other Issues: Adding redundant columns

Other issues include, increase in table size, maintenance and loss of information:

- The size of the (largest table i.e.) transaction table increases by the size of the Sale_Person key.
 - For the example being considered, the detail table size increases from 1.2 GB to 1.32 GB.
- If the Sale_Person key changes (e.g. new 12 digit NID), then updates to be reflected all the way to transaction table.
- In the absence of 1:M relationship, column movement will actually result in loss of data.

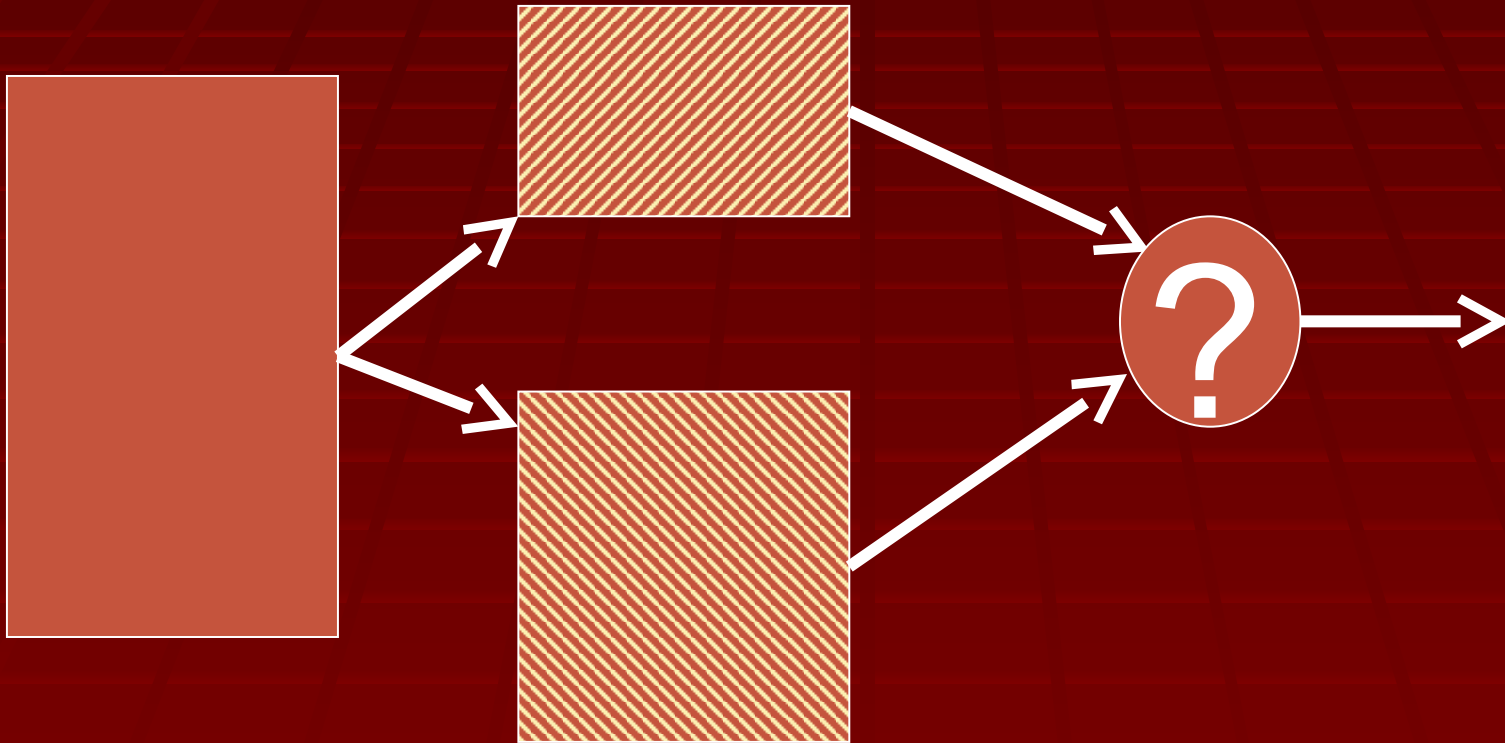
Ease of use Issues: Horizontal Splitting

Horizontal splitting is a Divide&Conquer technique that exploits parallelism. The conquer part of the technique is about combining the results.

Lets see how it works for hash based splitting/partitioning.

- Assuming uniform hashing, hash splitting supports even data distribution across all partitions in a pre-defined manner.
- However, hash based splitting is not easily reversible to eliminate the split.

Ease of use Issues: Horizontal Splitting



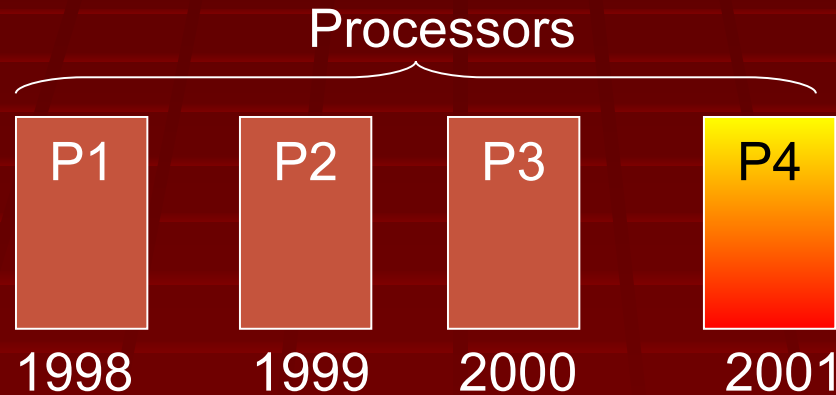
Ease of use Issues: Horizontal Splitting

- Round robin and random splitting:
 - Guarantee good data distribution.
 - Almost impossible to reverse (or undo).
 - Not pre-defined.

Ease of use Issues: Horizontal Splitting

- Range and expression splitting:
 - Can facilitate partition elimination with a smart optimizer.
 - Generally lead to "hot spots" (uneven distribution of data).

Performance Issues: Horizontal Splitting



Dramatic cancellation of airline reservations after 9/11, resulting in "hot spot"

Splitting based on year

Performance issues: Vertical Splitting Facts

Example: Consider a 100 byte header for the member table such that 20 bytes provide complete coverage for 90% of the queries.

Split the member table into two parts as follows:

1. Frequently accessed portion of table (20 bytes), and
2. Infrequently accessed portion of table (80+ bytes). **Why 80+?**

Note that primary key (member_id) must be present in both tables for eliminating the split.

Performance issues: Vertical Splitting Good vs. Bad

Scanning the claim table for most frequently used queries will be 500% faster with vertical splitting

Ironically, for the “infrequently” accessed queries the performance will be inferior as compared to the un-split table because of the join overhead.

Virtual University of Pakistan

Data Warehousing

Lecture-10

Online Analytical Processing (OLAP)

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

DWH & OLAP

- Relationship between DWH & OLAP
- Data Warehouse & OLAP go together.
- Analysis supported by OLAP

Supporting the human thought process

THOUGHT PROCESS



QUERY SEQUENCE



How many such query sequences can be programmed in advance?

Analysis of last example

- Analysis is **Ad-hoc**
 - Analysis is **interactive** (user driven)
 - Analysis is **iterative**
 - Answer to one question leads to a dozen more
 - Analysis is **directional**
 - Drill Down
 - Roll Up
 - Pivot
- More in subsequent slides

Challenges...

- Not feasible to write predefined queries.
 - Fails to remain user_driven (becomes programmer driven).
 - Fails to remain ad_hoc and hence is not interactive.
- Enable ad-hoc query support
 - Business user can not build his/her own queries (does not know SQL, should not know it).
 - On_the_go SQL generation and execution too slow.

Challenges

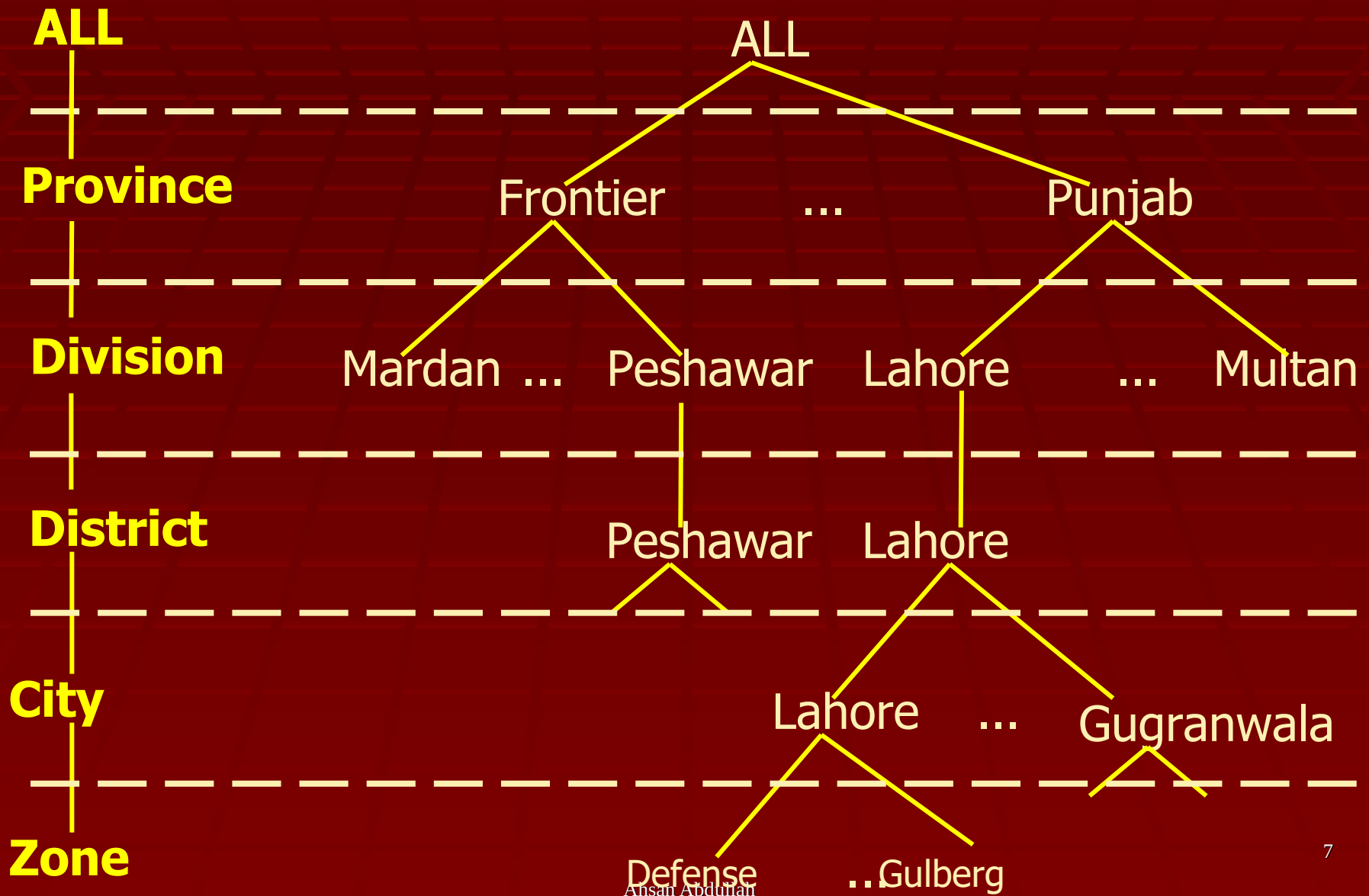
- Contradiction

- Want to compute answers in advance, but don't know the questions

- Solution

- Compute answers to “all” possible “queries”. But how?
 - NOTE: Queries are multidimensional aggregates at some level

“All” possible queries (level aggregates)



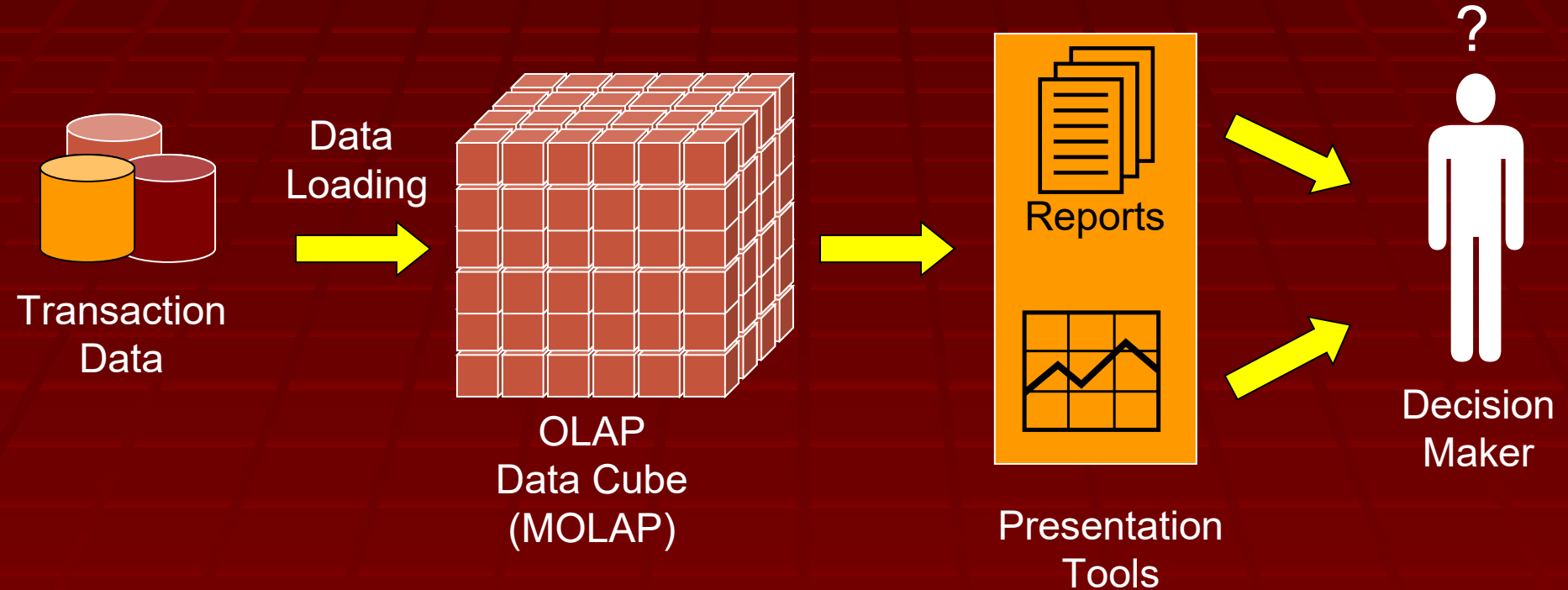
OLAP: Facts & Dimensions

- **FACTS:** Quantitative values (numbers) or “measures.”
 - e.g., units sold, sales \$, C°, Kg etc.
- **DIMENSIONS:** Descriptive categories.
 - e.g., time, geography, product etc.
- DIM often organized in hierarchies representing levels of detail in the data (e.g., week, month, quarter, year, decade etc.).

Where Does OLAP Fit In?

- It is a classification of applications, NOT a database design technique.
- Analytical processing uses multi-level aggregates, instead of record level access.
- Objective is to support very
 - I. fast
 - II. iterative and
 - III. ad-hoc decision-making.

Where does OLAP fit in?



OLTP vs. OLAP

Feature	OLTP	OLAP
Level of data	Detailed	Aggregated
Amount of data per transaction	Small	Large
Views	Pre-defined	User-defined
Typical write operation	Update, insert, delete	Bulk insert
“age” of data	Current (60-90 days)	Historical 5-10 years and also current
Number of users	High	Low-Med
Tables	Flat tables	Multi-Dimensional tables
Database size	Med (10^9 B – 10^{12} B)	High (10^{12} B – 10^{15} B)
Query Optimizing	Requires experience	Already “optimized”
Data availability	High	Low-Med

OLAP FASMI Test

Fast: Delivers information to the user at a fairly constant rate. Most queries answered in under five seconds.

Analysis: Performs basic numerical and statistical analysis of the data, pre-defined by an application developer or defined ad-hocly by the user.

Shared: Implements the security requirements necessary for sharing potentially confidential data across a large user population.

Multi-dimensional: The essential characteristic of OLAP.

Information: Accesses all the data and information necessary and relevant for the application, wherever it may reside and not limited by volume.

...from the *OLAP Report* by Pendse and Creeth.

Virtual University of Pakistan

Data Warehousing

Lecture-11

Multidimensional OLAP (MOLAP)

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

Multidimensional OLAP (MOLAP)

OLAP Implementations

1. **MOLAP**: OLAP implemented with a multi-dimensional data structure.
2. **ROLAP**: OLAP implemented with a relational database.
3. **HOLAP**: OLAP implemented as a hybrid of MOLAP and ROLAP.
4. **DOLAP**: OLAP implemented for desktop decision support environments.

MOLAP Implementations

OLAP has historically been implemented using a multi_dimensional data structure or “cube”.

- Dimensions are key business factors for analysis:
 - **Geographies** (city, district, division, province,...)
 - **Products** (item, product category, product department,...)
 - **Dates** (day, week, month, quarter, year,...)
- Very high performance achieved by $O(1)$ time lookup into “cube” data structure to retrieve pre_aggregated results.

MOLAP Implementations

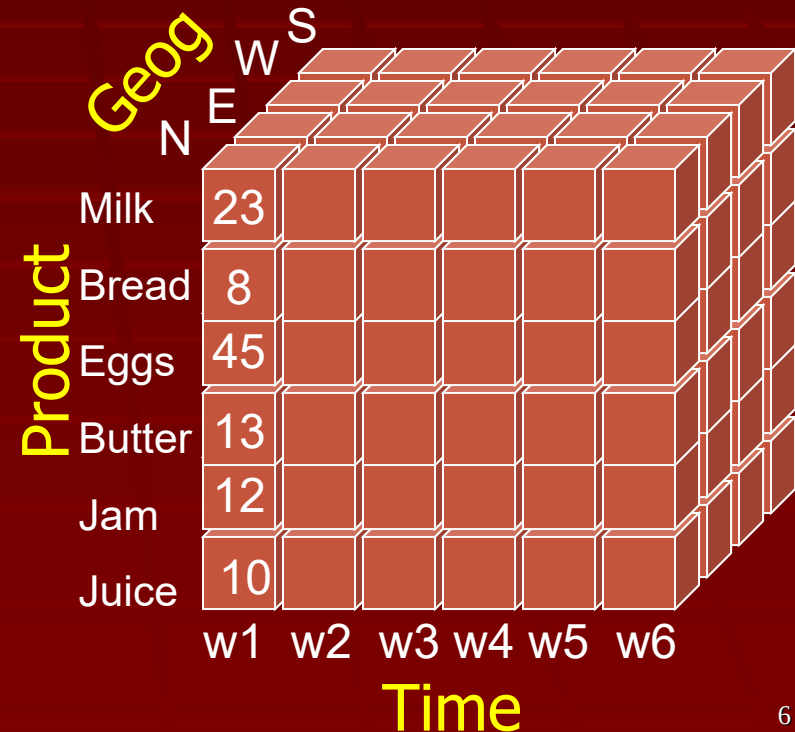
- No standard query language for querying MOLAP
 - *No SQL !*
- Vendors provide proprietary languages allowing business users to create queries that involve pivots, drilling down, or rolling up.
 - E.g. MDX of Microsoft
 - Languages generally involve extensive visual (click and drag) support.
 - Application Programming Interface (API)'s also provided for probing the cubes.

Aggregations in MOLAP

- Sales volume as a function of (i) product, (ii) time, and (iii) geography
- A cube structure created to handle this.

Dimensions: Product, Geography, Time

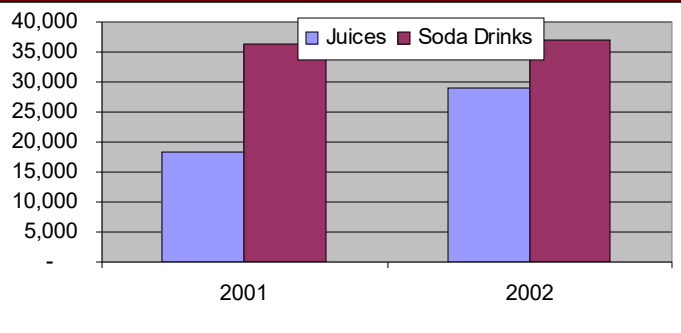
Hierarchical summarization paths



Cube operations

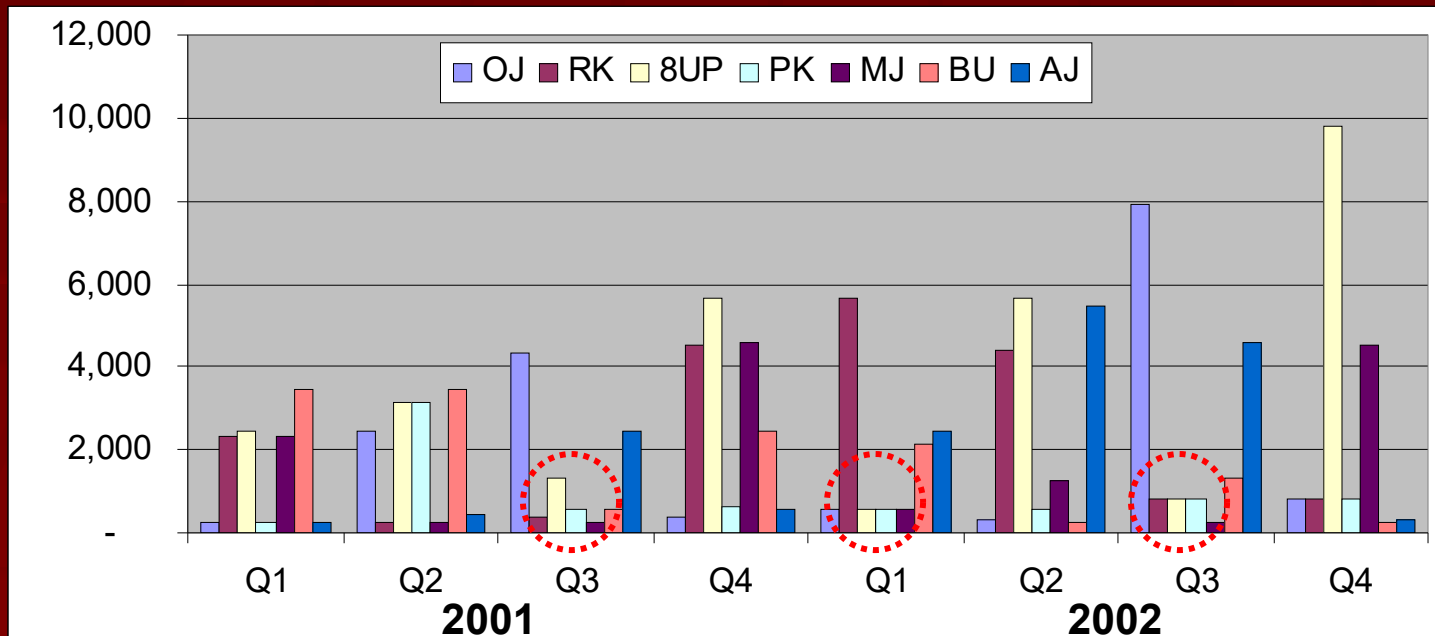
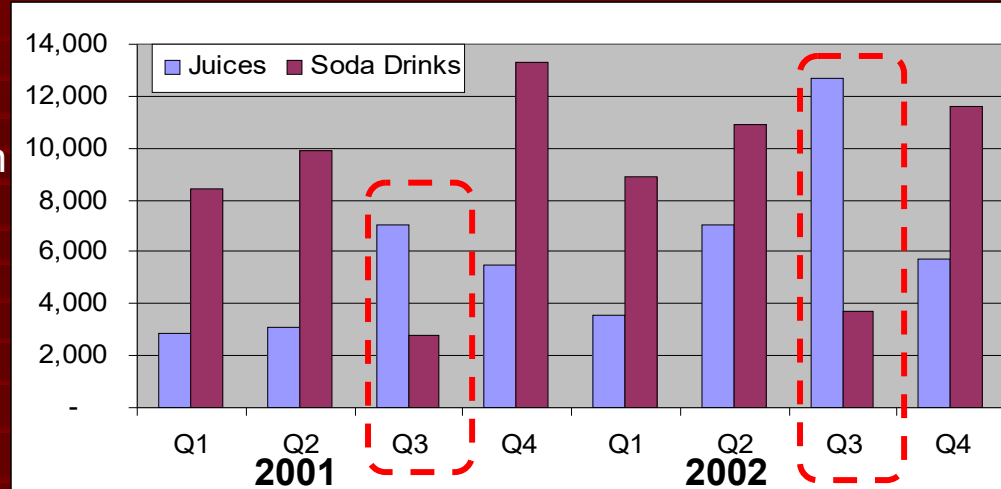
- Drill down: get more details
 - e.g., given summarized sales as above, find breakup of sales by city within each region, or within Sindh
- Rollup: summarize data
 - e.g., given sales data, summarize sales for last year by product category and region
- Slice and dice: select and project
 - e.g.: Sales of soft-drinks in Karachi during last quarter
- Pivot: change the view of data

Querying the cube



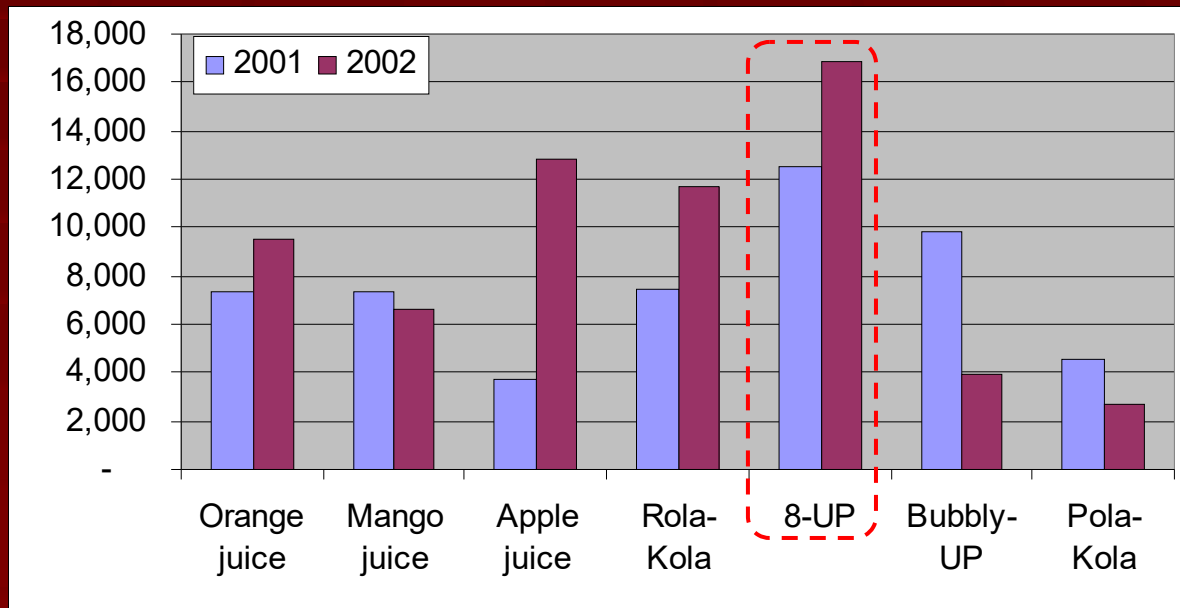
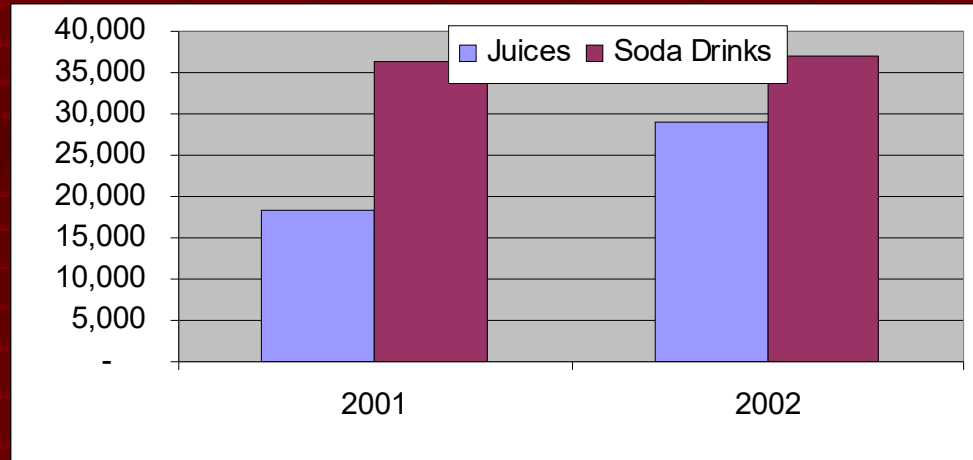
Drill-Down

Roll-Up



Drill-down

Querying the cube: Pivoting



MOLAP evaluation

Advantages of MOLAP:

- Instant response (pre-calculated aggregates).
- Impossible to ask question without an answer.
- Value added functions (ranking, % change).

MOLAP evaluation

Drawbacks of MOLAP:

- Long load time (pre-calculating the cube may take days!).
- Very sparse cube (wastage of space) for high cardinality (sometimes in small hundreds). e.g. number of heaters sold in Jacobabad or Sibi.

MOLAP Implementation issues

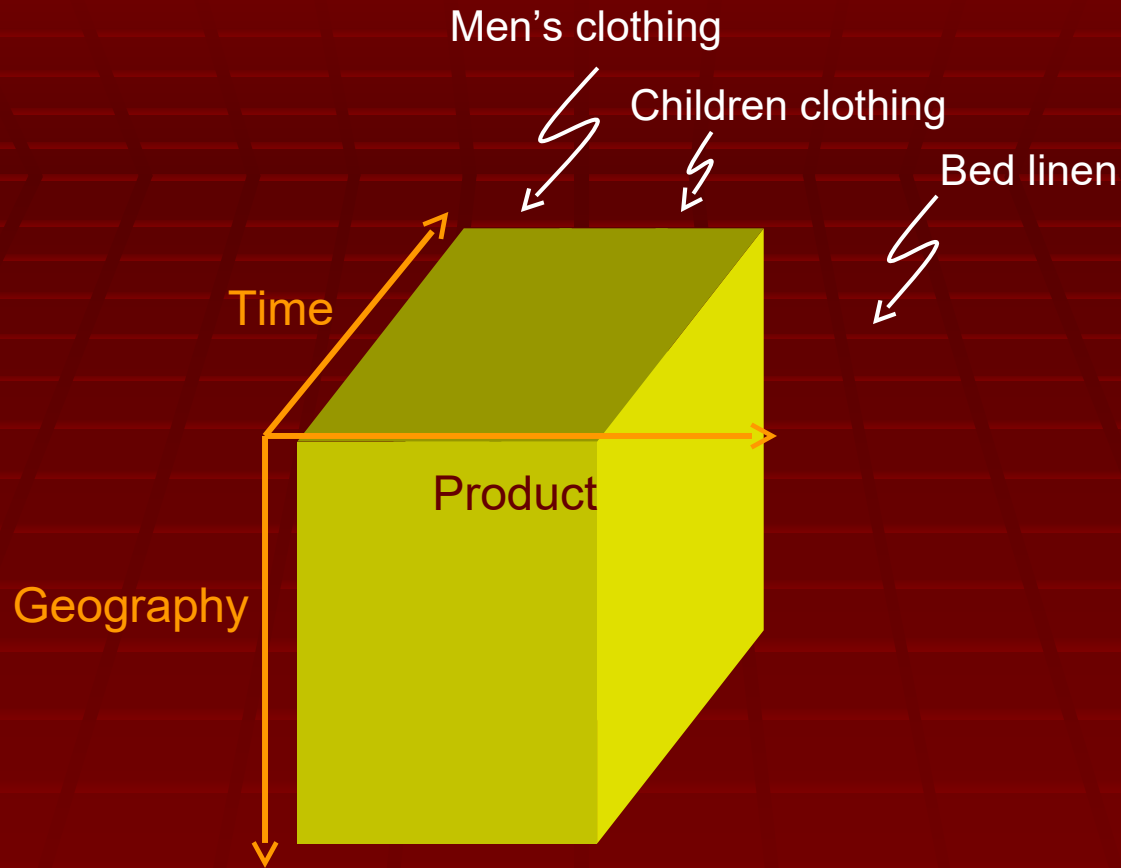
Maintenance issue: Every data item received must be aggregated into every cube (assuming “to-date” summaries are maintained). **Lot of work.**

Storage issue: As dimensions get less detailed (e.g., year vs. day) cubes get much smaller, but storage consequences for building hundreds of cubes can be significant. **Lot of space.**

Partitioned Cubes

- To overcome the space limitation of MOLAP, the cube is partitioned.
- The divide&conquer cube partitioning approach helps alleviate the scalability limitations of MOLAP implementation.
- One logical cube of data can be spread across multiple physical cubes on separate (or same) servers.
- Ideal cube partitioning is completely invisible to end users.
- Performance degradation does occurs in case of a join across partitioned cubes.

Partitioned Cubes: How it looks Like?



Sales data cube partitioned at a major cotton products sale outlet

Virtual Cubes

Used to query two dissimilar cubes by creating a third “virtual” cube by a join between two cubes.

- Logically similar to a relational view i.e. linking two (or more) cubes along common dimension(s).
- Biggest advantage is saving in space by eliminating storage of redundant information.

Example: Joining the store cube and the list price cube along the product dimension, to calculate the sale price without redundant storage of the sale price data.

Summary

Virtual University of Pakistan

Data Warehousing

Lecture-12

Relational OLAP (ROLAP)

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

Relational OLAP (ROLAP)

Why ROLAP?

Issue of scalability i.e. curse of dimensionality for MOLAP

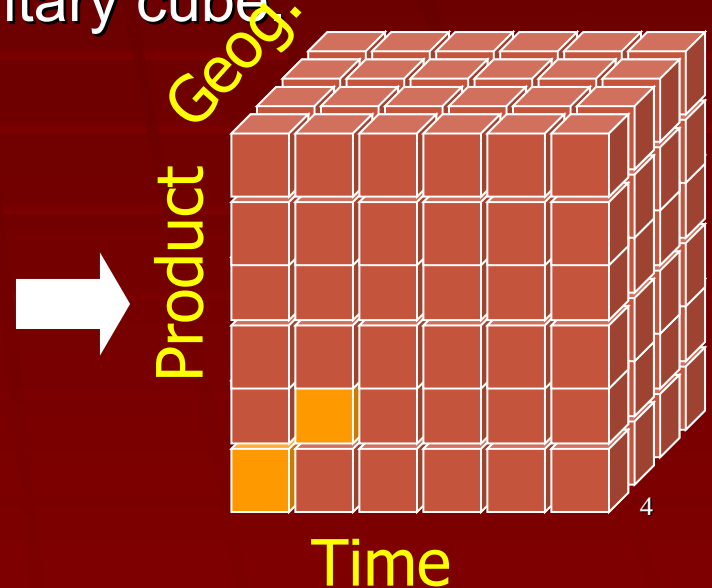
- Deployment of significantly large dimension tables as compared to MOLAP using secondary storage.
- Aggregate awareness allows using pre-built summary tables by some front-end tools.
- Star schema designs usually used to facilitate ROLAP querying (in next lecture).

ROLAP as a “Cube”

- OLAP data is stored in a relational database (e.g. a star schema)
- The fact table is a way of *visualizing* as a “un-rolled” cube.
- So where is the **cube**?
 - It's a matter of perception
 - Visualize the fact table as an elementary cube.

Fact Table

Month	Product	Zone	Sale K Rs.
M1	P1	Z1	250
M2	P2	Z1	500



How to create “Cube” in ROLAP

- Cube is a logical entity containing values of a certain fact at a certain aggregation level at an intersection of a combination of dimensions.
- The following table can be created using 3 queries

Product_ID	Month_ID				
	SUM (Sales_Amt)	M1	M2	M3	ALL
	P1				
	P2				
	P3				
	Total				

How to create “Cube” in ROLAP using SQL

- For the table entries, without the totals

```
SELECT      S.Month_Id, S.Product_Id,  
            SUM(S.Sales_Amt)  
FROM        Sales  
GROUP BY    S.Month_Id, S.Product_Id;
```

- For the row totals

```
SELECT S.Product_Id, SUM (Sales_Amt)  
FROM    Sales  
GROUP BY S.Product_Id;
```

- For the column totals

```
SELECT      S.Month_Id, SUM (Sales)  
FROM        Sales  
GROUP BY    S.Month_Id;
```


Problem With Simple Approach

- Number of required queries increases exponentially with the increase in number of dimensions.
 - Its wasteful to compute all queries.
 - In the example, the first query can do most of the work of the other two queries
 - If we could save that result and aggregate over Month_Id and Product_Id, we could compute the other queries more efficiently

CUBE Clause

- The CUBE clause is part of SQL:1999
 - GROUP BY CUBE (v1, v2, ..., vn)
 - Equivalent to a collection of GROUP BYs, one for each of the subsets of v1, v2, ..., vn

ROLAP & Space Requirement

If one is not careful, with the increase in number of dimensions, the number of summary tables gets very large

Consider the example discussed earlier with the following two dimensions on the fact table...

Time: Day, Week, Month, Quarter, Year, All Days

Product: Item, Sub-Category, Category, All Products

EXAMPLE: ROLAP & Space Requirement

A naïve implementation will require all combinations of summary tables at each and every aggregation level.

②	2001				2002			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Orange juice	232	2,432	4,353	354	535	345	7,897	789
Rola-Kola	2,342	243	353	4,535	5,655	4,424	789	798
8-UP	2,424	3,131	1,313	5,675	567	5,675	789	9,797
Pola-Kola	242	3,112	567	646	567	567	789	798
Mango juice	2,342	243	243	4,564	564	1,232	242	4,553
Bubbly-UP	3,453	3,453	535	2,422	2,131	242	1,321	245
Apple juice	253	456	2,433	567	2,442	5,453	4,566	345

③	2001				2002			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Soda Drinks	8,461	9,939	2,768	13,278	8,920	10,908	3,688	11,638
Juices	2,827	3,131	7,029	5,485	3,541	7,030	12,705	5,687

④	2001	2002
Orange juice	7,371	9,566
Mango juice	7,392	6,591
Apple juice	3,709	12,806
Rola-Kola	7,473	11,666
8-UP	12,543	16,828
Bubbly-UP	9,863	3,939
Pola-Kola	4,567	2,721

⑩	2001	2002
Juices	18,472	28,963
Soda Drinks	36,447	37,156

24 summary tables, add in geography, results in 120 tables

ROLAP Issues

- Maintenance.
- Non standard hierarchy of dimensions.
- Non standard conventions.
- Explosion of storage space requirement.
- Aggregation pit-falls.

ROLAP Issue: Maintenance

Summary tables are mostly a maintenance issue (similar to MOLAP) than a storage issue.

- Notice that summary tables get much smaller as dimensions get less detailed (e.g., year vs. day).
- Should plan for twice the size of the unsummarized data for ROLAP summaries in most environments.
- Assuming "to-date" summaries, every detail record that is received into warehouse must aggregate into EVERY summary table.

ROLAP Issue: Hierarchies

Dimensions are NOT always simple hierarchies

Dimensions can be more than simple hierarchies i.e. item, subcategory, category, etc.

The product dimension might also branch off by trade style that cross simple hierarchy boundaries such as:

Looking at sales of **air conditioners** that cross manufacturer boundaries, such as COY1, COY2, COY3 etc.

Looking at sales of all “**green colored**” items that even cross product categories (washing machine, refrigerator, split-AC, etc.).

Looking at a combination of both.

ROLAP Issue: Convention

Conventions are NOT absolute

Example: What is calendar year? What is a week?

- Calendar:

01 Jan. to 31 Dec or

01 Jul. to 30 Jun. or

01 Sep to 30 Aug.

- Week:

Mon. to Sat. or Thu. to Wed.

ROLAP Issue: Storage space explosion

Summary tables required for non-standard grouping

Summary tables required along different definitions of year, week etc.

Brute force approach would quickly overwhelm the system storage capacity due to a combinatorial explosion.

ROALP Issues: Aggregation pitfalls

- Coarser granularity correspondingly decreases potential cardinality.
- Aggregating whatever that can be aggregated.
- Throwing away the detail data after aggregation.

How to Reduce Summary tables?

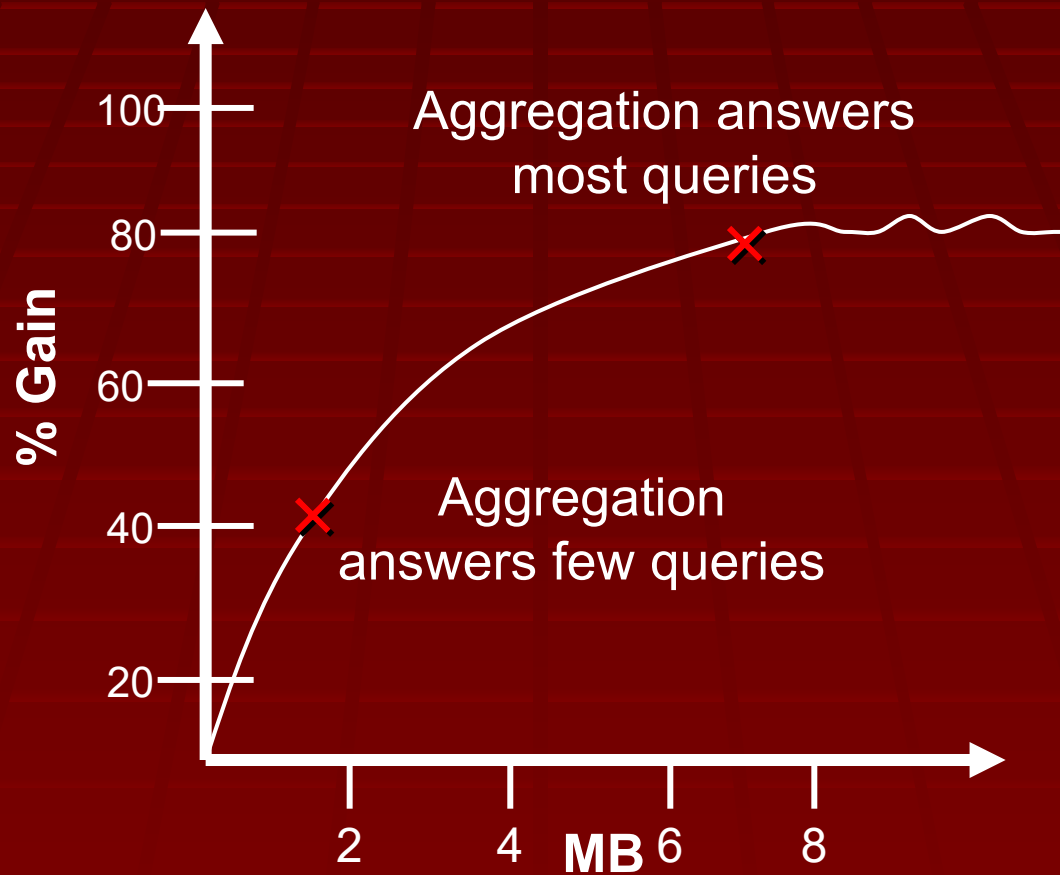
Many ROLAP products have developed means to reduce the number of summary tables by:

- Building summaries on-the-fly as required by end-user applications.
- Enhancing performance on common queries at coarser granularities.
- Providing smart tools to assist DBAs in selecting the "best" aggregations to build i.e. trade-off between speed and space.

Performance vs. Space Trade-Off

- Maximum performance boost implies using lots of disk space for storing every pre-calculation.
- Minimum performance boost implies no disk space with zero pre-calculation.
- Using meta data to determine best level of pre-aggregation from which all other aggregates can be computed.

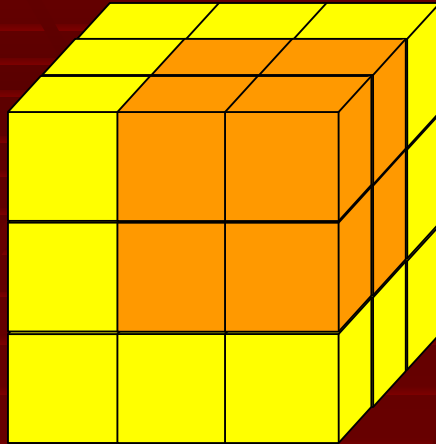
Performance vs. Space Trade-off using Wizard



HOLAP

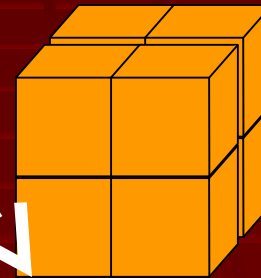
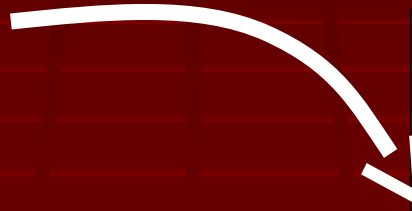
- Target is to get the best of both worlds.
- HOLAP (Hybrid OLAP) allow co-existence of pre-built MOLAP cubes alongside relational OLAP or ROLAP structures.
- How much to pre-build?

DOLAP



Cube on the
remote server

Subset of the cube is
transferred to the local
machine



Local
Machine/Server

End

Virtual University of Pakistan

Data Warehousing

Lecture-13

Dimensional Modeling (DM)

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

Dimensional Modeling (DM)

The need for ER modeling?

- Problems with early COBOLian data processing systems.
- Data redundancies
- From flat file to *Table*, each entity ultimately becomes a *Table* in the physical schema.
- Simple $O(n^2)$ Join to work with *Tables*

Why ER Modeling has been so successful?

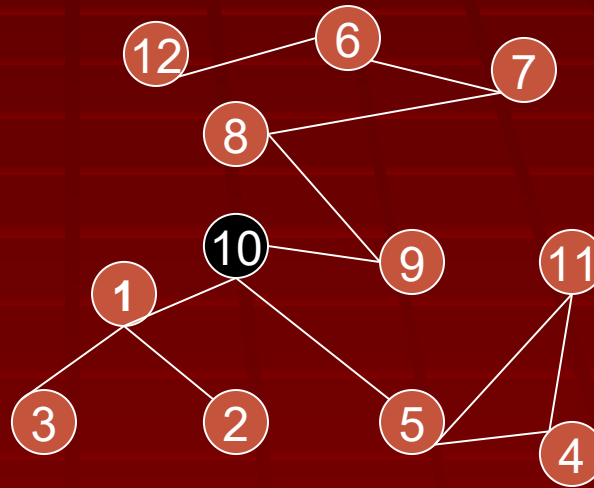
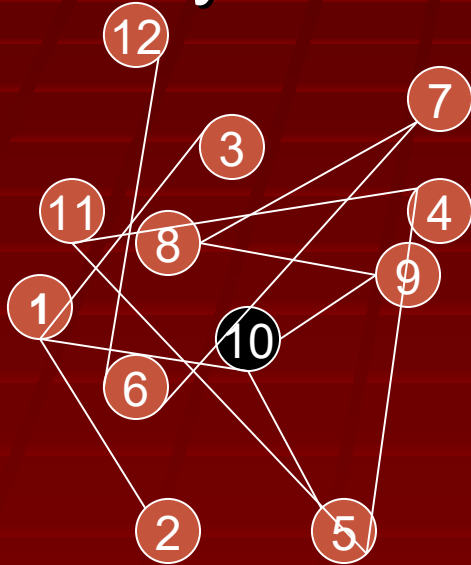
- Coupled with normalization drives out all the redundancy out of the database.
- Change (or add or delete) the data at just one point.
- Can be used with indexing for very fast access.
- Resulted in success of OLTP systems.

Need for DM: Un-answered Qs

- Lets have a look at a typical ER data model first.
- Some Observations:
 - All tables look-alike, as a consequence it is difficult to identify:
 - Which table is more important ?
 - Which is the largest?
 - Which tables contain numerical measurements of the business?
 - Which table contain nearly static descriptive attributes?

Need for DM: Complexity of Representation

- Many topologies for the same ER diagram, all appearing different.
 - Very hard to visualize and remember.



- A large number of possible connections to any two (or more) tables

Need for DM: The Paradox

- **The Paradox:** Trying to make information accessible using tables resulted in an inability to query them!
- ER and Normalization result in large number of tables which are:
 - Hard to understand by the users (DB programmers)
 - Hard to navigate optimally by DBMS software
- Real value of ER is in using tables individually or in pairs
- Too complex for queries that span multiple tables with a large number of records

ER vs. DM

ER	DM
Constituted to optimize OLTP performance.	Constituted to optimize DSS query performance.
Models the <u>micro</u> relationships among data elements.	Models the <u>macro</u> relationships among data elements with an overall <u>deterministic</u> strategy.
A wild variability of the structure of ER models.	All dimensions serve as equal entry points to the fact table.
Very vulnerable to changes in the user's querying habits, because such schemas are asymmetrical.	Changes in users' querying habits can be accommodated by automatic SQL generators.

How to simplify a ER data model?

- Two general methods:
 - De-Normalization
 - Dimensional Modeling (DM)

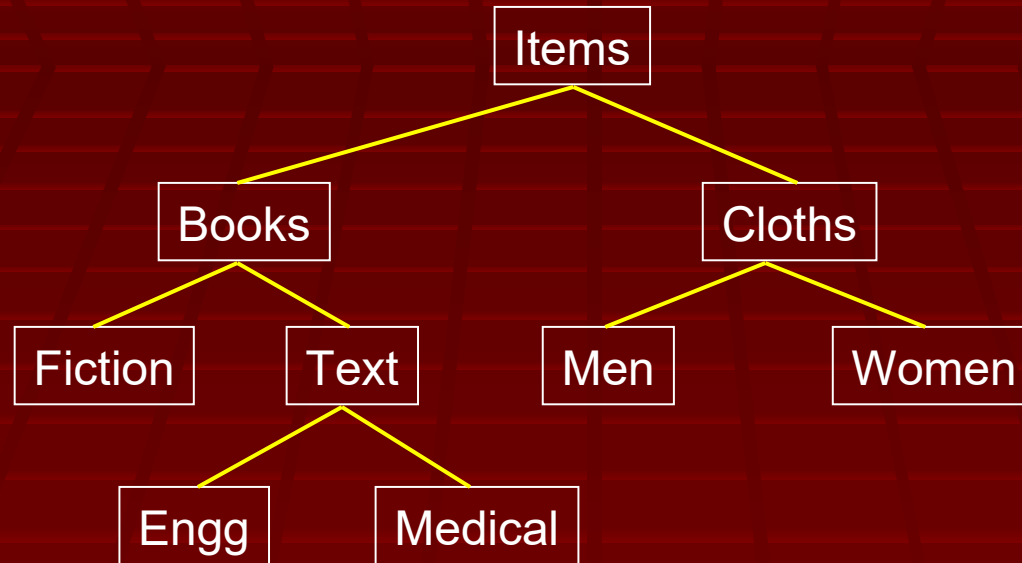
What is DM?...

- A simpler logical model optimized for decision support.
- Inherently dimensional in nature, with a single central fact table and a set of smaller dimensional tables.
- Multi-part key for the fact table
- Dimensional tables with a single-part PK.
- Keys are usually system generated

What is DM?...

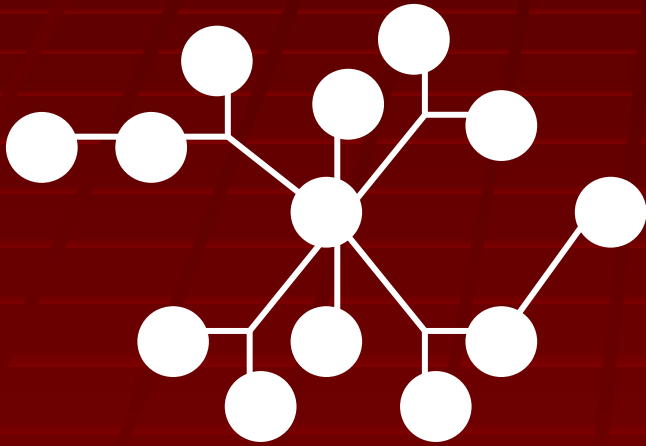
- Results in a star like structure, called star schema or star join.
 - All relationships mandatory M-1.
 - Single path between any two levels.
- Supports ROLAP operations.

Dimensions have Hierarchies

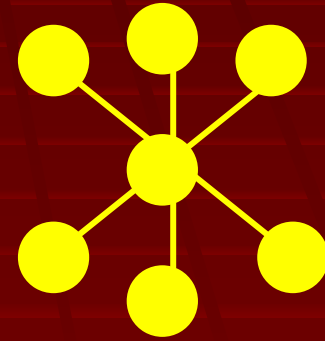


Analysts tend to look at the data through dimension at a particular “level” in the hierarchy

The two Schemas

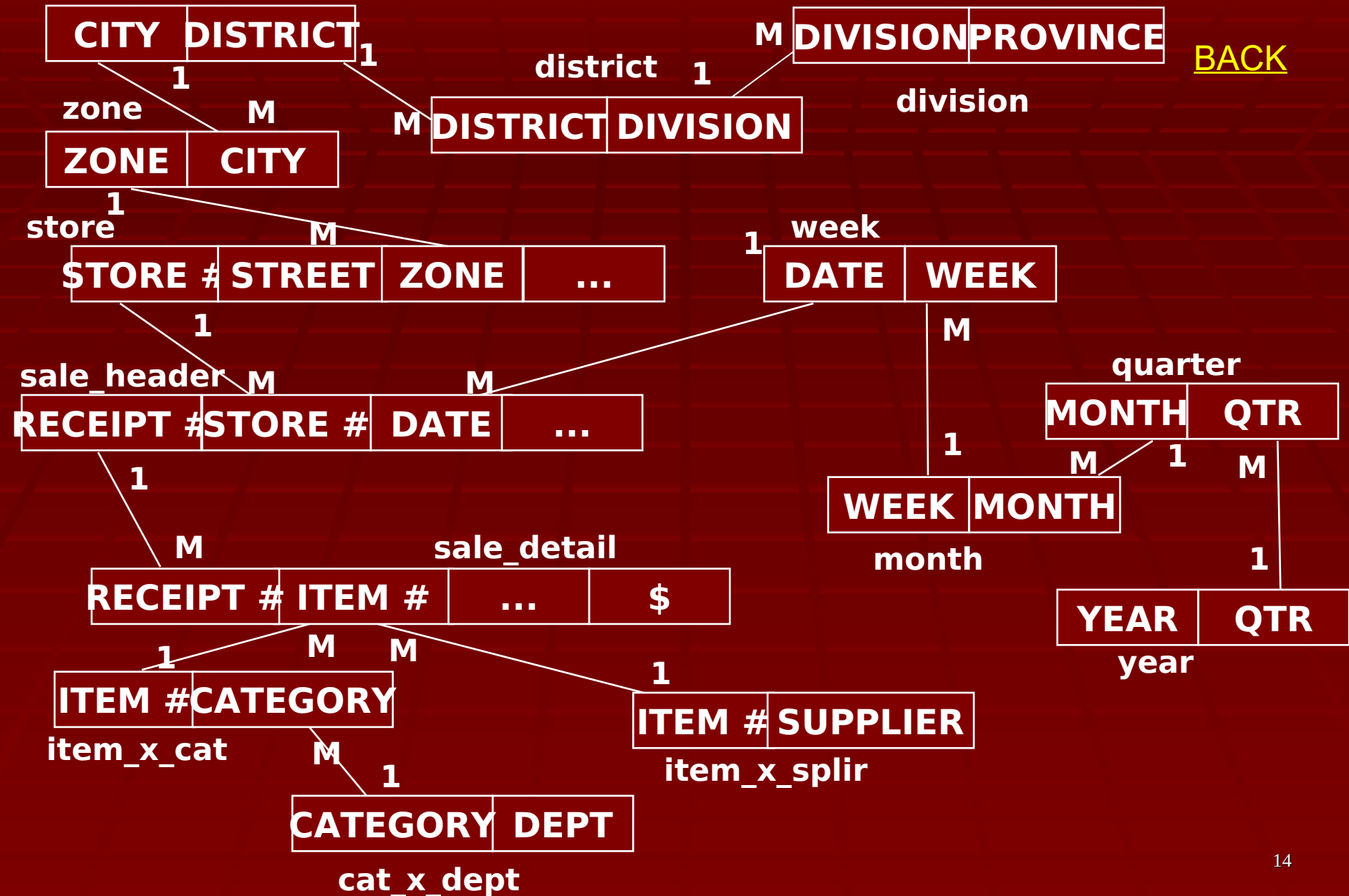


Snow-flake

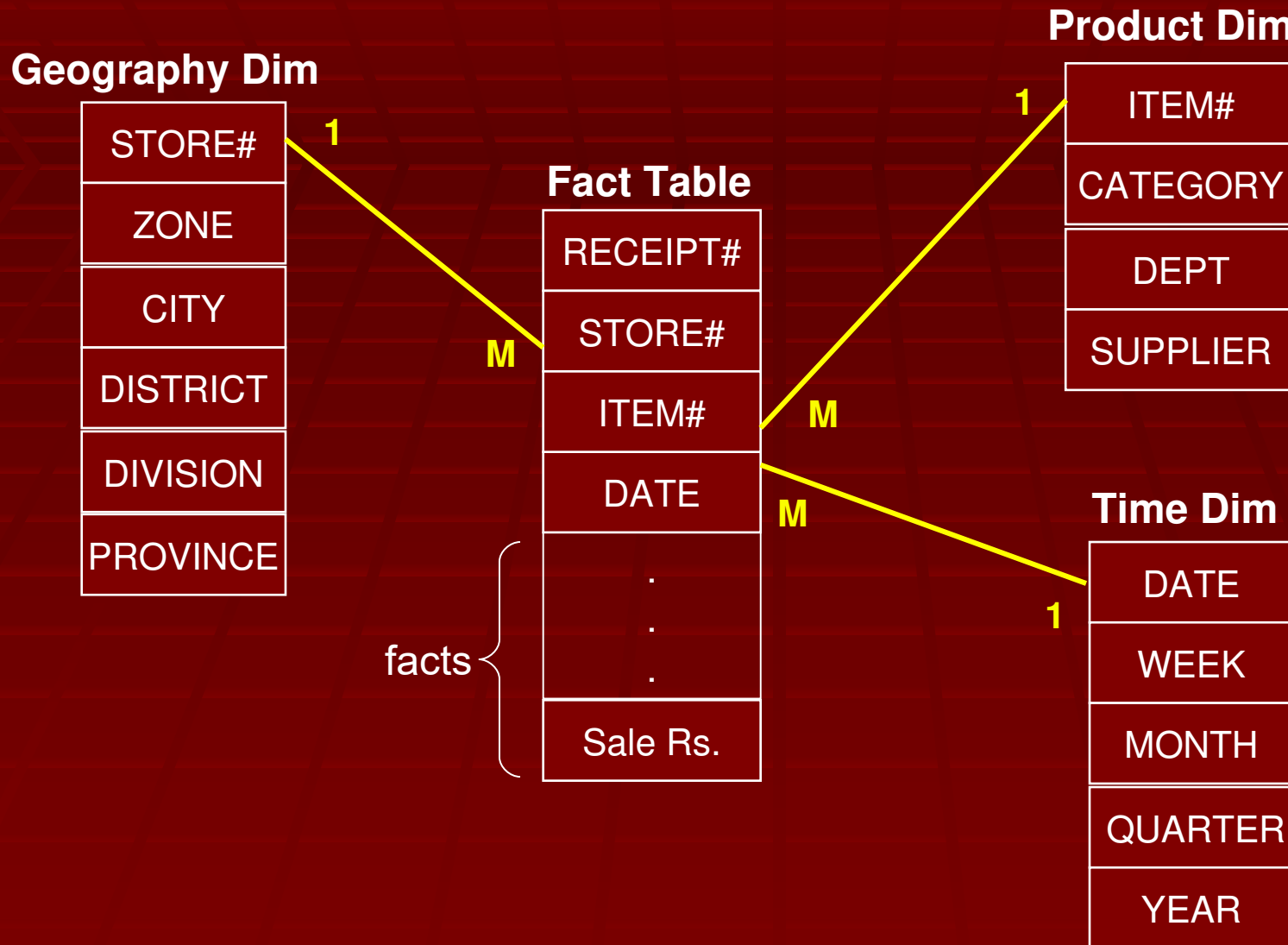


Star

“Simplified” 3NF (Retail)



Vastly Simplified Star Schema



The Benefit of Simplicity

Beauty lies in close correspondence
with the business, evident even to
business users.

Features of Star Schema

Dimensional hierarchies are collapsed into a single table for each dimension. **Loss of Information?**

A single fact table created with a single header from the detail records, resulting in:

- A vastly simplified physical data model!
- Fewer tables (thousands of tables in some ERP systems).
- Fewer joins resulting in high performance.
- Some requirement of additional space.

Quantifying space requirement

Quantifying use of additional space using star schema

There are about 10 million mobile phone users in Pakistan.

Say the top company has half of them = 500,000

Number of days in 1 year = 365

Number of calls recorded each day = 250,000 (assumed)

Maximum number of records in fact table = 91 billion rows

Assuming a relatively small header size = 128 bytes

Fact table storage used = 11 Tera bytes

Average length of city name = 8 characters $\approx$ 8 bytes

Total number of cities with telephone access = 170 (1 byte)

Space used for city name in fact table using Star = $8 \times 0.091 = 0.728$ TB

Space used for city code using snow-flake = $1 \times 0.091 = 0.091$ TB

Additional space used $\approx$ 0.637 Tera byte i.e. about 5.8%

Virtual University of Pakistan

Data Warehousing

Lecture-14

Process of Dimensional Modeling

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

Process of Dimensional Modeling

The Process of Dimensional Modeling

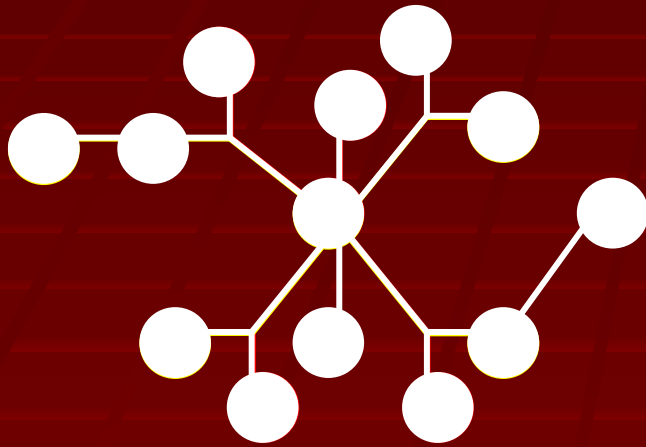
Four Step Method from ER to DM

1. Choose the Business Process
2. Choose the Grain
3. Choose the Facts
4. Choose the Dimensions

Step-1: Choose the Business Process

- A business process is a major operational process in an organization.
- Typically supported by a legacy system (database) or an OLTP.
 - Examples: Orders, Invoices, Inventory etc.
- Business Processes are often termed as Data Marts and that is why many people criticize DM as being data mart oriented.

Step-1: Separating the Process



Snow-flake

Star-1

Star-2

Step-2: Choosing the Grain

- Grain is the fundamental, atomic level of data to be represented.
- Grain is also termed as the unit of analyses.
- Example grain statements
- Typical grains
 - Individual Transactions
 - Daily aggregates (snapshots)
 - Monthly aggregates
- Relationship between grain and expressiveness.
- Grain vs. hardware trade-off.

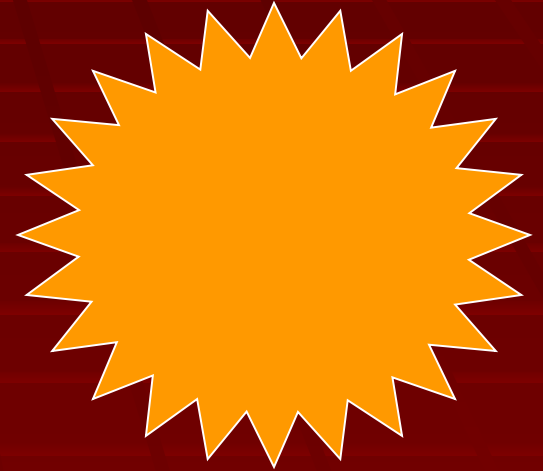
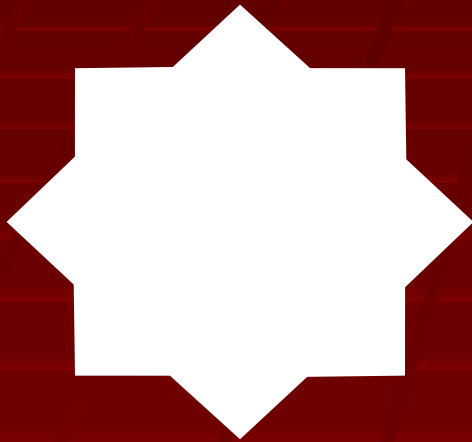
Step-2: Relationship b/w Grain

LOW Granularity

HIGH Granularity



Four aggregates per week
 $4 \times 4 = 16$ values



Two aggregates per week
 $2 \times 4 = 8$ values

Daily aggregates
 $6 \times 4 = 24$ values

The case FOR data aggregation

- Works well for repetitive queries.
- Follows the known thought process.
- Justifiable if used for max number of queries.
- Provides a “big picture” or macroscopic view.
- Application dependent, usually inflexible to business changes (remember lack of absoluteness of conventions).

The case AGAINST data aggregation

- Aggregation is irreversible.
 - Can create monthly sales data from weekly sales data, but the reverse is not possible.
- Aggregation limits the questions that can be answered.
 - What, when, why, where, what-else, what-next

The case AGAINST data aggregation

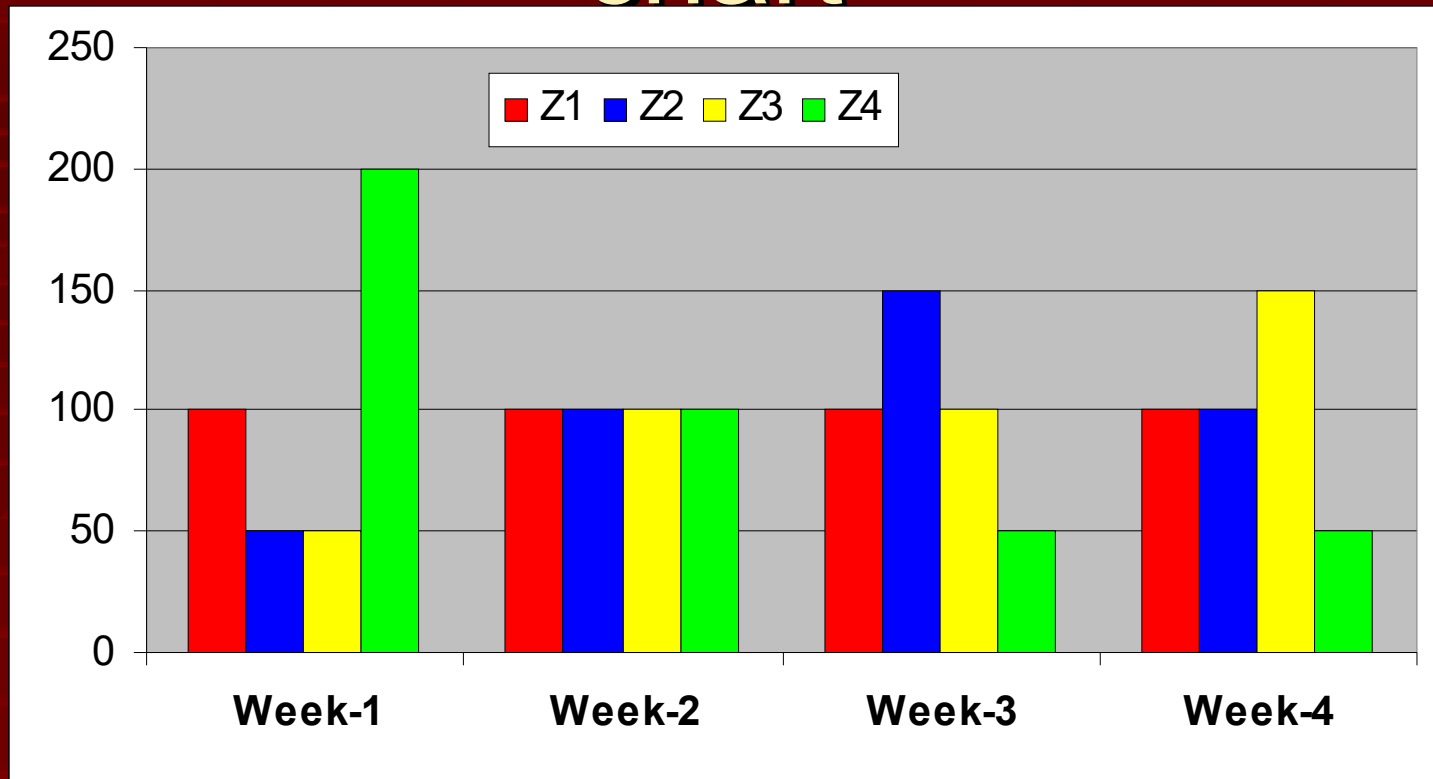
- Aggregation can hide crucial facts.
 - The average of 100 & 100 is same as 150 & 50

Aggregation hides crucial facts

Example

	Week-1	Week-2	Week-3	Week-4	Average
Zone-1	Just looking at the averages i.e. aggregate				100
Zone-2					100
Zone-3					100
Zone-4					100
Average	100	100	100	100	

Aggregation hides crucial facts chart



Z1: Sale is constant (need to work on it)

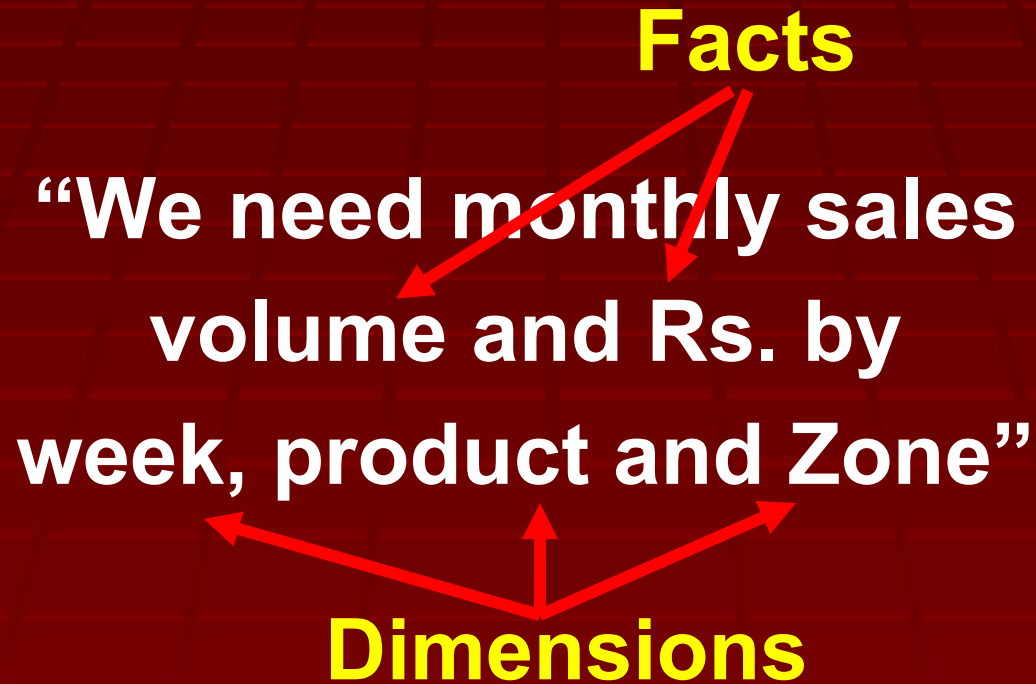
Z2: Sale went up, then fell (need of concern)

Z3: Sale is on the rise, why?

Z4: Sale dropped sharply, need to look deeply.

W2: Static sale

Step 3: Choose Facts statement



Step 3: Choose Facts

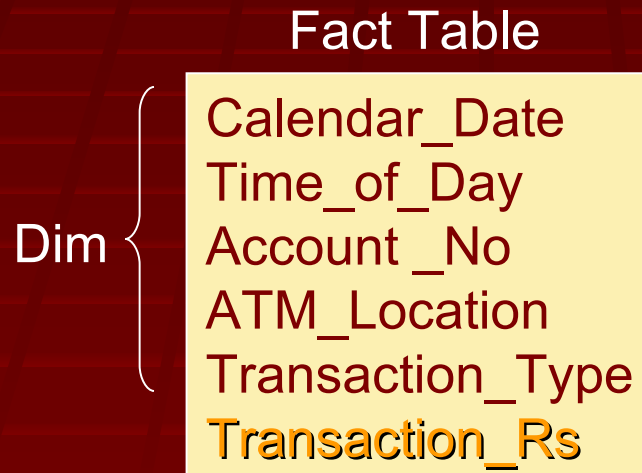
- Choose the facts that will populate each fact table record.
 - Remember that best Facts are Numeric, Continuously Valued and Additive.
 - Example: Quantity Sold, Amount etc.

Step 4: Choose Dimensions

- Choose the dimensions that apply to each fact in the fact table.
 - Typical dimensions: time, product, geography etc.
 - Identify the descriptive attributes that explain each dimension.
 - Determine hierarchies within each dimension.

Step-4: How to Identify a Dimension?

- The single valued attributes during recording of a transaction are dimensions.



Time_of_day: Morning, Mid Morning, Lunch Break etc.

Transaction_Type: Withdrawal, Deposit, Check balance etc.

Step-4: Can Dimensions be Multi-valued?

- Are dimensions ALWAYS single?
 - Not really
 - What are the problems? And how to handle them

- Calendar_Date (of inspection)
- Reg_No
- Technician
- Workshop
- Maintenance_Operation

- How many maintenance operations are possible?
 - Few
 - Maybe more for old cars.

Step-4: Dimensions & Grain

- Several grains are possible as per business requirement.
 - For some aggregations certain descriptions do not remain atomic.
 - Example: Time_of_Day may change several times during daily aggregate, but not during a transaction
- Choose the dimensions that are applicable within the selected grain.

Virtual University of Pakistan

Data Warehousing

Lecture-15

Issues of Dimensional Modeling

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

Issues of Dimensional Modeling

Step 3: Additive vs. Non-Additive facts

- **Additive** facts are easy to work with
 - Summing the fact value gives meaningful results
 - Additive facts:
 - Quantity sold
 - Total Rs. sales
- Non-additive facts:
 - Averages (average sales price, unit price)
 - Percentages (% discount)
 - Ratios (gross margin)
 - Count of distinct products sold

Month	Crates of Bottles Sold
May	14
Jun.	20
Jul.	24
TOTAL	58

Month	% discount
May	10
Jun.	8
Jul.	6
TOTAL	24% ← Incorrect!

Step-3: Classification of Aggregation Functions

- How hard to compute aggregate from sub-aggregates?
 - Three classes of aggregates:
 - Distributive
 - Compute aggregate **directly from sub-aggregates**
 - Examples: MIN, MAX, COUNT, SUM
 - Algebraic
 - Compute aggregate from **constant-sized summary** of subgroup
 - Examples: STDDEV, AVERAGE
 - For AVERAGE, summary data for each group is SUM, COUNT
 - Holistic
 - Require **unbounded amount of information** about each subgroup
 - Examples: MEDIAN, COUNT DISTINCT
 - Usually impractical for a data warehouses!

Step-3: Not recording Facts

- Transactional fact tables don't have records for events that don't occur
 - Example: No records(rows) for products that were not sold.
- This has both advantage and disadvantage.
 - **Advantage:** Benefit of sparsity of data
 - Significantly less data to store for “rare” events
 - **Disadvantage:** Lack of information
 - Example: What products on promotion were not sold?

Step-3: A Fact-less Fact Table

- “Fact-less” fact table
 - A fact table without numeric fact columns
 - Captures relationships between dimensions
 - Use a dummy fact column that always has value 1

Step-3: Example: Fact-less Fact Tables

Examples:

- Department/Student mapping fact table
 - What is the major for each student?
 - Which students did not enroll in ANY course
- Promotion coverage fact table
 - Which products were on promotion in which stores for which days?
 - Kind of like a periodic snapshot fact

Step-4: Handling Multi-valued Dimensions?

- One of the following approaches is adopted:
 - Drop the dimension.
 - Use a primary value as a single value.
 - Add multiple values in the dimension table.
 - Use “Helper” tables.

Step-4: OLTP & Slowly Changing Dimensions

**OLTP systems not good at tracking the past.
History never changes.**

**OLTP systems are not “static” always
evolving, data changing by overwriting.**

**Inability of OLTP systems to track history,
purged after 90 to 180 days.**

**Actually don't want to keep historical data for
OLTP system.**

Step-4: DWH Dilemma: Slowly Changing Dimensions

The responsibility of the DWH to track the changes.

Example: Slight change in description, but the product ID (SKU) is not changed.

Dilemma: Want to track both old and new descriptions, what do they use for the key? And where do they put the two values of the changed ingredient attribute?

Step-4: Explanation of Slowly Changing Dimensions...

- Compared to fact tables, contents of dimension tables are relatively stable.
 - New sales transactions occur constantly.
 - New products are introduced rarely.
 - New stores are opened very rarely.
- The assumption **does not hold** in some cases
 - Certain dimensions **evolve with time**
 - e.g. description and formulation of products change with time
 - Customers get married and divorced, have children, change addresses etc.
 - Land changes ownership etc.
 - Changing names of sales regions.

Step-4: Explanation of Slowly Changing Dimensions...

Although these dimensions change but the **change is not rapid.**

Therefore called “Slowly” Changing Dimensions

Step-4: Handling Slowly Changing Dimensions

- **Option-1: Overwrite History**
 - Example: Code for a city, product entered incorrectly
 - Just overwrite the record changing the values of modified attributes.
 - No keys are affected.
 - No changes needed elsewhere in the DM.
 - **Cannot track history** and hence not a good option in DSS.

Step-4: Handling Slowly Changing Dimensions

■ Option-2: Preserve History

- Example: The packaging of a part change from glued box to stapled box, but the code assigned (SKU) is not changed.
- Create an additional dimension record at the time of change with new attribute values.
- **Segments history** accurately between old and new description
- Requires adding two to three version numbers to the end of key. SKU#+1, SKU#+2 etc.

Step-4: Handling Slowly Changing Dimensions

- **Option-3: Create current valued field**
 - Example: The name and organization of the sales regions change over time, and want to know how sales would have looked with old regions.
 - Add a new field called current_region rename old to previous_region.
 - Sales record keys are not changed.
 - Only **TWO most recent changes** can be tracked.

Step-4: Pros and Cons of Handling

- Option-1: Overwrite existing value
 - + Simple to implement
 - No tracking of history
- Option-2: Add a new dimension row
 - + Accurate historical reporting
 - + Pre-computed aggregates unaffected
 - Dimension table grows over time
- Option-3: Add a new field
 - + Accurate historical reporting to last TWO changes
 - + Record keys are unaffected
 - Dimension table size increases

Virtual University of Pakistan

Data Warehousing

Lecture-16

Extract Transform Load (ETL)

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

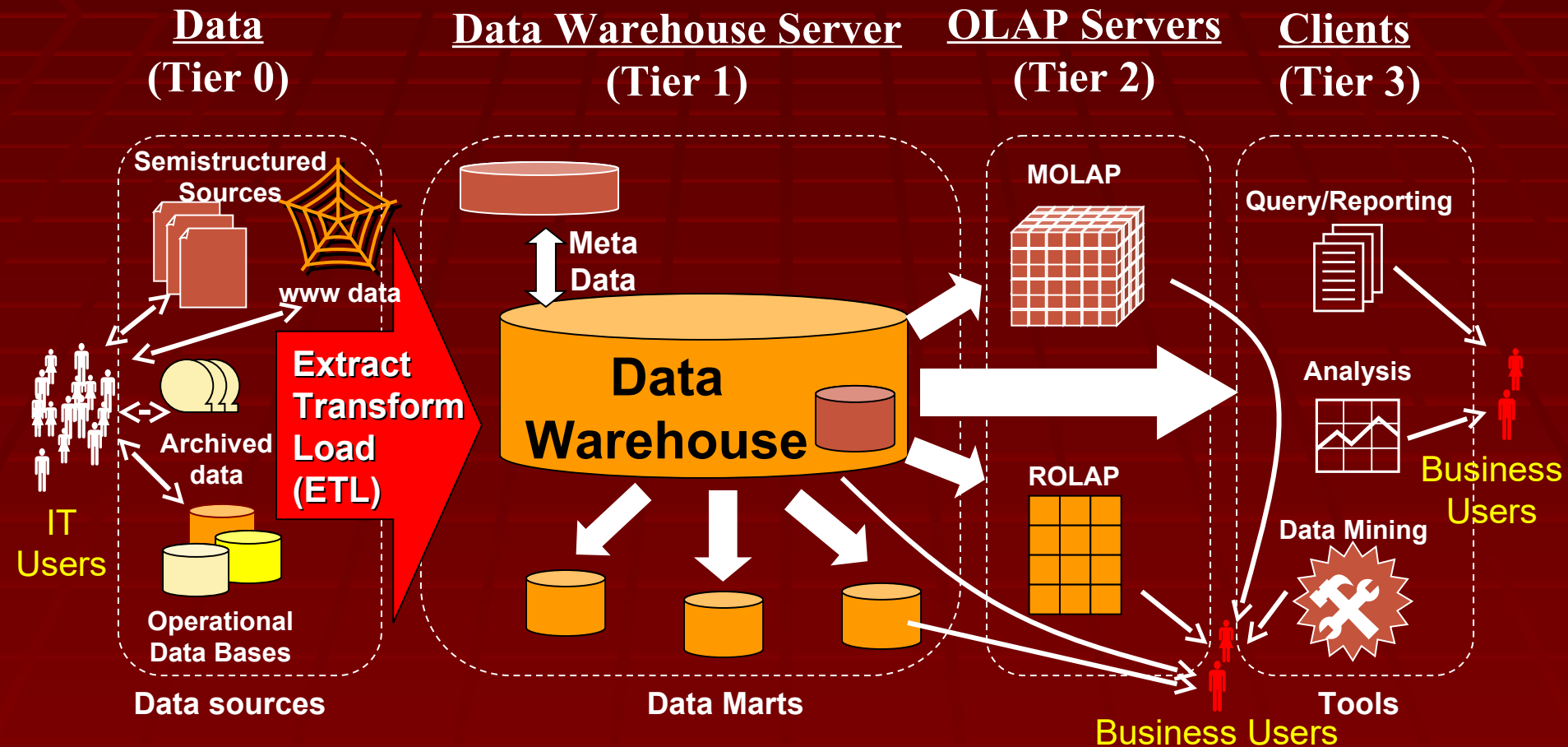
www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan1010@yahoo.com

Extract Transform Load (ETL)

Putting the pieces together



{Comment: All except ETL washed out look}

The ETL Cycle

ETRACT

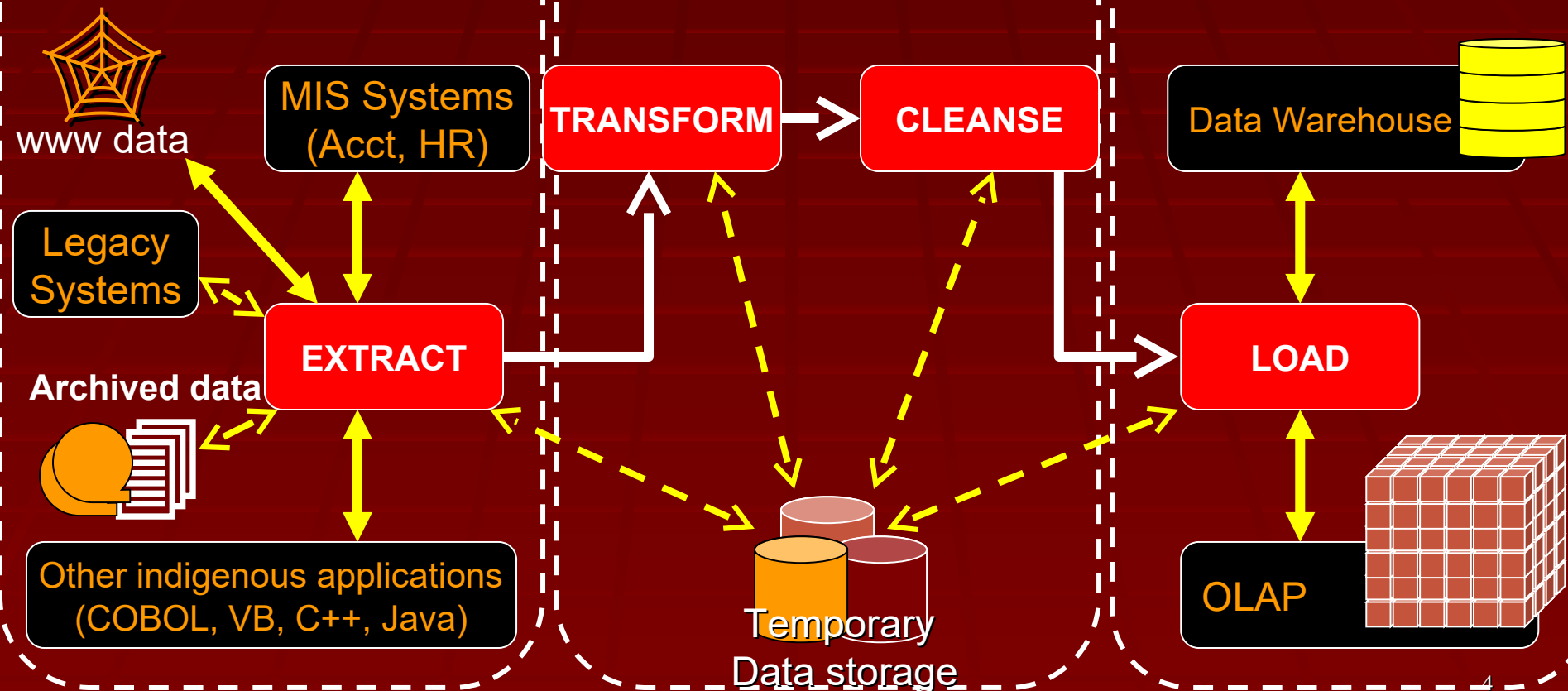
The process of reading data from different sources.

TRANSFORM

The process of transforming the extracted data from its original state into a consistent state so that it can be placed into another database.

LOAD

The process of writing the data into the target source.

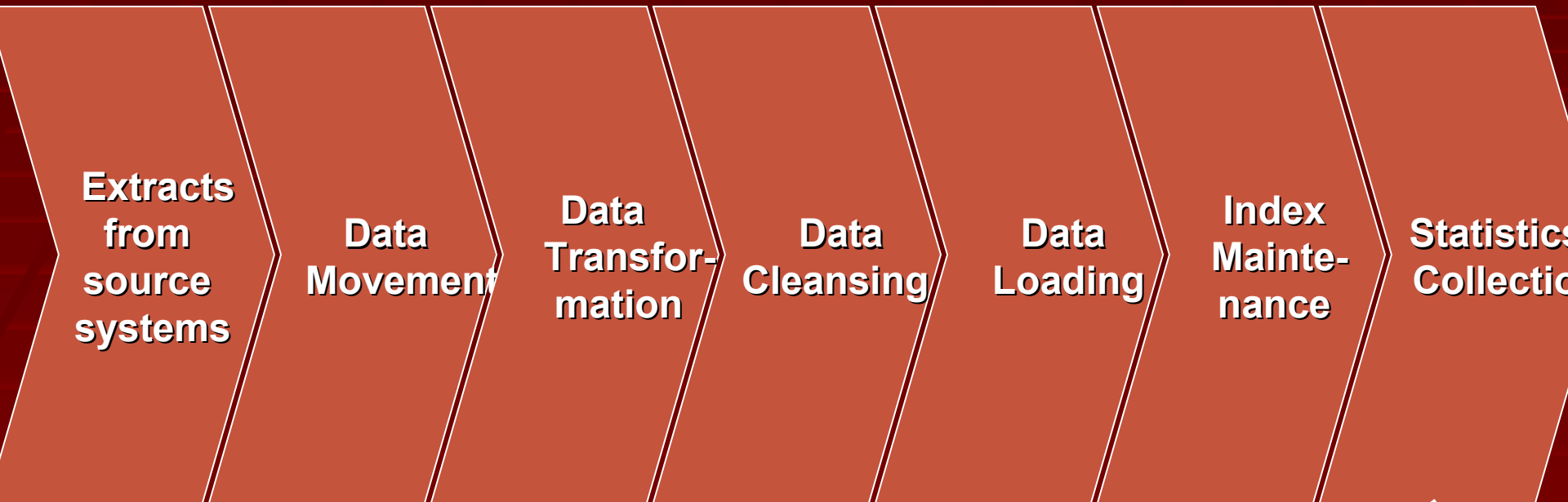


ETL Processing

ETL is independent yet interrelated steps.

It is important to look at the big picture.

Data acquisition time may include...



Note: Backup will come as other elements after “Statistical collection”

Backup



Back-up is a major task, its a DWH not a cube

Overview of Data Extraction

First step of ETL, followed by many.

Source system for extraction are typically OLTP systems.

A very complex task due to number of reasons:

- Very complex and poorly documented source system.
- Data has to be extracted not once, but number of times.
-

The process design is dependent on:

- Which extraction method to choose?
- How to make available extracted data for further processing?

Types of Data Extraction

- **Logical Extraction**
 - Full Extraction
 - Incremental Extraction
- **Physical Extraction**
 - Online Extraction
 - Offline Extraction
 - Legacy vs. OLTP

Logical Data Extraction

- **Full Extraction**

- The data extracted completely from the source system.
- No need to keep track of changes.
- Source data made available as-is with any additional information.

- **Incremental Extraction**

- Data extracted after a well defined point/event in time.
- Mechanism used to reflect/record the temporal changes in data (column or table).
- Sometimes entire tables off-loaded from source system into the DWH.
- Can have significant performance impacts on the data warehouse server.

Physical Data Extraction...

- **Online Extraction**

- Data extracted directly from the source system.
- May access source tables through an intermediate system.
- Intermediate system usually similar to the source system.

- **Offline Extraction**

- Data NOT extracted directly from the source system, instead staged explicitly outside the original source system.
- Data is either already structured or was created by an extraction routine.
- Some of the prevalent structures are:
 - Flat files
 - Dump files
 - Redo and archive logs
 - Transportable table-spaces

Physical Data Extraction

- **Legacy vs. OLTP**
 - **Data moved from the source system**
 - **Copy made of the source system data**
 - **Staging area used for performance reasons**

Data Transformation

- **Basic tasks**
 1. Selection
 2. Splitting/Joining
 3. Conversion
 4. Summarization
 5. Enrichment

Data Transformation Basic Tasks

- Selection

Data Transformation Basic Tasks

- Splitting/joining

Data Transformation Basic Tasks

- Conversion

Data Transformation Basic Tasks: Conversion

Example-1

- Convert common data elements into a consistent form i.e. name and address.

Field format

First-Family-title

Family-title-comma-first

Family-comma-first-title

Field data

Muhammad Ibrahim Contractor

Ibrahim Contractor, Muhammad

Ibrahim, Muhammad Contractor

- Translation of dissimilar codes into a standard code.

Natl. ID ———>NID

National ID ———>NID

F/NO-2

F-2

FL.NO.2

FL.2

FL/NO.2

FL-2

FLAT-2

FLAT#

FLAT,2

FLAT-NO-2

FL-NO.2

————> FLAT No. 2

Data Transformation Basic Tasks: Conversion

Example-2

- Data representation change
 - EBCDIC to ASCII
- Operating System Change
 - Mainframe (MVS) to UNIX
 - UNIX to NT or XP
- Data type change
 - Program (Excel to Access), database format (FoxPro to Access).
 - Character, numeric and date type.
 - Fixed and variable length.

Data Transformation Basic Tasks

- Summarization

Data Transformation Basic Tasks

- Enrichment

Data Transformation Basic Tasks: Enrichment Example

- Data elements are mapped from source tables and files to destination fact and dimension tables.

Input Data

HAJI MUHAMMAD IBRAHIM, GOVT. CONT.
K. S. ABDULLAH & BROTHERS,
MAMOOJI ROAD, ABDULLAH MANZIL
RAWALPINDI, Ph 67855



Parsed Data

First Name:	HAJI MUHAMMAD
Family Name:	IBRAHIM
Title:	GOVT. CONT.
Firm:	K. S. ABDULLAH & BROTHERS
Firm Location:	ABDULLAH MANZIL
Road:	MAMOOJI ROAD
Phone:	051-67855
City:	RAWALPINDI
Code:	46200

- Default values are used in the absence of source data.
- Fields are added for unique keys and time elements.

Aspects of Data Loading Strategies

- Need to look at:
 - Data freshness
 - System performance
 - Data volatility
- Data Freshness
 - Very fresh low update efficiency
 - Historical data, high update efficiency
 - Always trade-offs in the light of goals
- System performance
 - Availability of staging table space
 - Impact on query workload
- Data Volatility
 - Ratio of new to historical data
 - High percentages of data change (batch update)

Three Loading Strategies

- Once we have transformed data, there are three primary loading strategies:
- Full data refresh with BLOCK INSERT or 'block slamming' into empty table.
- Incremental data refresh with BLOCK INSERT or 'block slamming' into existing (populated) tables.
- Trickle/continuous feed with constant data collection and loading using row level insert and update operations.

Virtual University of Pakistan

Data Warehousing

Lecture-17 Issues of ETL

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@cluxing.com

Issues of ETL

Why ETL Issues?

Data from different source systems will be different, poorly documented and dirty. Lot of analysis required.

Easy to collate addresses and names? Not really. No address or name standards.

Use software for standardization. Very expensive, as any “standards” vary from country to country, not large enough market.

Why ETL Issues?

Things would have been simpler in the presence of operational systems, but that is not always the case

Manual data collection and entry. Nothing wrong with that, but potential to introduces lots of problems.

Data is never perfect. The cost of perfection, extremely high vs. its value.

“Some” Issues

- Usually, if not always underestimated
- Diversity in source systems and platforms
- Inconsistent data representations
- Complexity of transformations
- Rigidity and unavailability of legacy systems
- Volume of legacy data
- Web scrapping

Complexity of problem/work underestimated

- Work seems to be deceptively simple.
- People start manually building the DWH.
- Programmers underestimate the task.
- Impressions could be deceiving.
- Traditional DBMS rules and concepts break down for very large heterogeneous historical databases.

Diversity in source systems and platforms

Platform	OS	DBMS	MIS/ERP
Main Frame	VMS	Oracle	SAP
Mini Computer	Unix	Informix	PeopleSoft
Desktop	Win NT	Access	JD Edwards
	DOS	Text file	

Dozens of source systems across organizations

Numerous source systems within an organization

Need specialist for each

Inconsistent data representations

Same data, different representation

Date value representations

Examples:

970314

03/14/1997

March 14 1997

1997-03-14

14-MAR-1997

2450521.5 (Julian date format)

Gender value representations

Examples:

- Male/Female

- 0/1

- M/F

- PM/PF

Multiple sources for same data element

Need to rank source systems on a per data element basis.

Take data element from source system with highest rank where element exists.

“Guessing” gender from name

Something is better than nothing?

Must sometimes establish “group ranking” rules to maintain data integrity.

First, middle and family name from two systems of different rank. People using middle name as first name.

Complexity of required transformations

Simple one-to-one scalar transformations

- 0/1 $\rightarrow$ M/F

One-to-many element transformations

- 4 x 20 address field $\rightarrow$ House/Flat, Road/Street, Area/Sector, City.

Many-to-many element transformations

- House-holding (who live together) and individualization (who are same) and same lands.

Rigidity and unavailability of legacy systems

- Very difficult to add logic to or increase performance of legacy systems.
- Utilization of expensive legacy systems is optimized.
- Therefore, want to off-load transformation cycles to open systems environment.
- This often requires new skill sets.
- Need efficient and easy way to deal with incompatible mainframe data formats.

Volume of legacy data

- Talking about not weekly data, but data spread over years.
- Historical data on tapes that are serial and very slow to mount etc.
- Need lots of processing and I/O to effectively handle large data volumes.
- Need efficient interconnect bandwidth to transfer large amounts of data from legacy sources to DWH.

Web scrapping

- Lot of data in a web page, but is mixed with a lot of “junk”.
- Problems:
 - Limited query interfaces
 - Fill in forms
 - “Free text” fields
 - E.g. addresses
 - Inconsistent output
 - i.e., html tags which mark interesting fields might be different on different pages.
 - Rapid change without notice.

Beware of data quality (or lack of it)

- Data quality is always worse than expected.
- Will have a couple of lectures on data quality and its management.
- It is not a matter of few hundred rows.
- Data recorded for running operations is not usually good enough for decision support.
 - Correct totals don't guarantee data quality.
 - Not knowing gender does not hurt POS.
 - Centurion customers popping up.

ETL vs. ELT

There are two fundamental approaches to data acquisition:

ETL: Extract, Transform, Load in which data transformation takes place on a separate transformation server.

ELT: Extract, Load, Transform in which data transformation takes place on the data warehouse server.

Combination of both is also possible

Virtual University of Pakistan

Data Warehousing

Lecture-18

ETL Detail: Data Extraction & Transformation

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan1010@yahoo.com

ETL Detail: Data Extraction & Transformation

Extracting Changed Data

Incremental data extraction

Incremental data extraction i.e. what has changed, say during last 24 hrs if considering nightly extraction.

Efficient when changes can be identified

This is efficient, when the small changed data can be identified efficiently.

Identification could be costly

Unfortunately, for many source systems, identifying the recently modified data may be difficult or effect operation of the source system.

Very challenging

Change Data Capture is therefore, typically the most challenging technical issue in data extraction.

ONLY yellow part will go to Graphics

Source Systems

Two CDC sources

- Modern systems
- Legacy systems

ONLY yellow part will go to Graphics

CDC in Modern Systems

ONLY yellow part will go to Graphics

- **Time Stamps**
 - Works if timestamp column present
 - If column not present, add column
 - May not be possible to modify table, so add triggers
- **Triggers**
 - Create trigger for each source table
 - Following each DML operation trigger performs updates
 - Record DML operations in a log
- **Partitioning**
 - Table range partitioned, say along date key
 - Easy to identify new data, say last week's data

CDC in Legacy Systems

- **Changes recorded in tapes** Changes occurred in legacy transaction processing are recorded on the log or journal tapes.
- **Changes read and removed from tapes** Log or journal tape are read and the update/transaction changes are stripped off for movement into the data warehouse.

ONLY yellow part will go to Graphics

- **Problems with reading a log/journal tape are many:**
 - Contains lot of extraneous data
 - Format is often arcane
 - Often contains addresses instead of data values and keys
 - Sequencing of data in the log tape often has deep and complex implications
 - Log tape varies widely from one DBMS to another.

CDC Advantages: Modern Systems

Advantages

1. Immediate.
2. No loss of history
3. Flat files NOT required



Modern
Systems

CDC Advantages: Legacy Systems

Advantages

1. No incremental on-line I/O required for log tape
2. The log tape captures all update processing
3. Log tape processing can be taken off-line.
4. No haste to make waste.

Legacy
Systems

Major Transformation Types

- Format revision
- Decoding of fields
- Calculated and derived values
- Splitting of single fields
- Merging of information
- Character set conversion
- Unit of measurement conversion
- Date/Time conversion
- Summarization
- Key restructuring
- Duplication

Major Transformation Types

ONLY yellow part will go to Graphics

- Format revision
- Decoding of fields
- Calculated and derived values
- Splitting of single fields

Covered in De-Norm

Covered in issues

Major Transformation Types

ONLY yellow part will go to Graphics

- **Merging of information**

Not really means combining columns to create one column.
Info for product coming from different sources merging it into single entity.

- **Character set conversion**

For PC architecture converting legacy EBCDIC to ASCII

- **Unit of measurement conversion**

For companies with global branches Km vs. mile or lb vs Kg

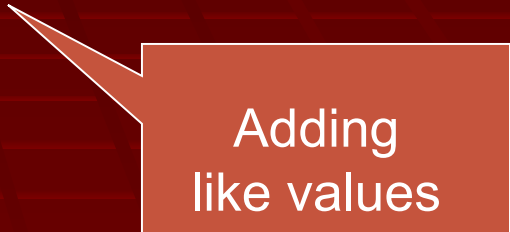
- **Date/Time conversion**

November 14, 2005 as 11/14/2005 in US and 14/11/2005 in the British format.
This date may be standardized to be written as 14 NOV 2005.

Major Transformation Types

ONLY yellow part will go to Graphics

- Aggregation & Summarization
- How they are different?



Adding like values

Summarization with calculation across business dimension is aggregation. Example Monthly compensation = monthly sale + bonus

- Why both are required?
 - Grain mismatch (don't require, don't have space)
 - Data Marts requiring low detail
 - Detail losing its utility

Major Transformation Types

ONLY yellow part will go to Graphics

- **Key restructuring** (inherent meaning at source)

92	42	4979	234
Country_Code	City_Code	Post_Code	Product_Code

- i.e. 92424979234 changed to 12345678

- **Removing duplication**

Incorrect or missing value

Inconsistent naming convention ONE vs 1

Incomplete information

Physically moved, but address not changed

Misspelling or falsification of names

Data content defects

- Domain value redundancy
 - Non-standard data formats
 - Non-atomic data values
 - Multipurpose data fields
 - Embedded meanings
 - Inconsistent data values
 - Data quality contamination

Data content defects Examples

ONLY yellow part will go to Graphics

Domain value redundancy

- Unit of Measure
 - Dozen, Doz., Dz., 12
- Non-standard data formats
 - Phone Numbers
 - 1234567 or 123.456.7
- Non-atomic data fields
 - Name & Addresses
 - Dr. Hameed Khan, PhD

Data content defects Examples

- Embedded Meanings
 - RC, AP, RJ
 - received, approved, rejected

Virtual University of Pakistan

Data Warehousing

Lecture-19

ETL Detail: Data Cleansing

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan@yahoo.com

ETL Detail: Data Cleansing

Background

- **Other names:** Called as data scrubbing or cleaning.
- **More than data arranging:** DWH is NOT just about arranging data, but should be clean for overall health of organization. We drink clean water!
- **Big problem, big effect:** Enormous problem, as most data is dirty. GIGO
ONLY yellow part will go to Graphics
- **Dirty is relative:** Dirty means does not confirm to proper domain definition and vary from domain to domain.
- **Paradox:** Must involve domain expert, as detailed domain knowledge is required, so it becomes semi-automatic, but has to be automatic because of large data sets.
- **Data duplication:** Original problem was removing duplicates in one system, compounded by duplicates from many systems.

Lighter Side of Dirty Data

- Year of birth 1995 current year 2005

{Comment: Show picture of baby}

- Born in 1986 hired in 1985

ONLY yellow part will go to Graphics

- Who would take it seriously? Computers while summarizing, aggregating, populating etc.
- Small discrepancies become irrelevant for large averages, but what about sums, medians, maximum, minimum etc.?

Serious Side of dirty data

- **Decision making at the Government level on investment** based on rate of birth in terms of schools and then teachers. Wrong data resulting in over and under investment.

ONLY yellow part will go to Graphics

- **Direct mail marketing** sending letters to wrong addresses retuned, or multiple letters to same address, loss of money and bad reputation and wrong identification of marketing region.

3 Classes of Anomalies...

- **Syntactically Dirty Data**
 - Lexical Errors
 - Irregularities
- **Semantically Dirty Data**
 - Integrity Constraint Violation
 - Business rule contradiction
 - Duplication
- **Coverage Anomalies**
 - Missing Attributes
 - Missing Records

3 Classes of Anomalies...

■ Syntactically Dirty Data

■ Lexical Errors

- Discrepancies between the structure of the data items and the specified format of stored values

- e.g. number of columns used are unexpected for a tuple (mixed up number of attributes)

■ Irregularities

- Non uniform use of units and values, such as
i.e. in US\$ or PK Rs?

This slide will NOT go to Graphics

■ Semantically Dirty Data

■ Integrity Constraint violation

■ Contradiction

- DoB > Hiring date etc.

■ Duplication

3 Classes of Anomalies...

- **Coverage or lack of it**

This slide will NOT go to Graphics

- **Missing Attribute**

- **Result of omissions while collecting the data.**
 - **A constraint violation if we have null values for attributes where NOT NULL constraint exists.**
 - **Case more complicated where no such constraint exists.**
 - **Have to decide whether the value exists in the real world and has to be deduced here or not.**

Why Coverage Anomalies?

- **Equipment malfunction (bar code reader, keyboard etc.)**
- **Inconsistent with other recorded data and thus deleted.**
- **Data not entered due to misunderstanding/illegibility.**
- **Data not considered important at the time of entry (e.g. Y2K).**

Handling missing data

- Dropping records.
- “Manually” filling missing values.
- Using a global constant as filler.
- Using the attribute mean (or median) as filler.
- Using the most probable value as filler.

Key Based Classification of Problems

- Primary key problems
- Non-Primary key problems

Primary key problems

- Same PK but different data.
- Same entity with different keys.
- PK in one system but not in other.
- Same PK but in different formats.

Non primary key problems...

- Different encoding in different sources.
- Multiple ways to represent the same information.
- Sources might contain invalid data.
- Two fields with different data but same name.

Non primary key problems

- Required fields left blank.
- Data erroneous or incomplete.
- Data contains null values.

Automatic Data Cleansing...

1.Statistical

2.Pattern Based

3.Clustering

4.Association Rules

Automatic Data Cleansing...

1. Statistical Methods

- Identifying outlier fields and records using the values of mean, standard deviation, range, etc., based on Chebyshev's theorem

2. Pattern-based

This slide will NOT go to Graphics

- Identify outlier fields and records that do not conform to existing patterns in the data.
- A pattern is defined by a group of records that have similar characteristics (“behavior”) for $p\%$ of the fields in the data set, where p is a user-defined value (usually above 90).
- Techniques such as partitioning, classification, and clustering can be used to identify patterns that apply to most records.

Automatic Data Cleansing

3. Clustering

- Identify outlier records using clustering based on Euclidian (or other) distance.
- Clustering the entire record space can reveal outliers that are not identified at the field level inspection
- Main drawback of this method is computational time.

This slide will NOT go to Graphics

4. Association rules

- Association rules with high confidence and support define a different kind of pattern.
- Records that do not follow these rules are considered outliers.

Virtual University of Pakistan

Data Warehousing

Lecture-20

Data Duplication Elimination & BSN Method

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan1010@yahoo.com

Why data duplicated?

A data warehouse is created from heterogeneous sources, with heterogeneous databases (different schema/representation) of the same entity.

The data coming from outside the organization owning the DWH, can have even lower quality data i.e. different representation for same entity, transcription or typographical errors.

Problems due to data duplication

Data duplication, can result in costly errors, such as:

- False frequency distributions.
- Incorrect aggregates due to double counting.
- Difficulty with catching fabricated identities by credit card companies.

Data Duplication: Non-Unique PK

- Duplicate Identification Numbers
 - Multiple Customer Numbers

Name	Phone Number	Cust. No.
<i>M. Ismail Siddiqi</i>	<i>021.666.1244</i>	<i>780701</i>
<i>M. Ismail Siddiqi</i>	<i>021.666.1244</i>	<i>780203</i>
<i>M. Ismail Siddiqi</i>	<i>021.666.1244</i>	<i>780009</i>

- Multiple Employee Numbers

Bonus Date	Name	Department	Emp. No.
<i>Jan. 2000</i>	<i>Khan Muhammad</i>	<i>213 (MKT)</i>	<i>5353536</i>
<i>Dec. 2001</i>	<i>Khan Muhammad</i>	<i>567 (SLS)</i>	<i>4577833</i>
<i>Mar. 2002</i>	<i>Khan Muhammad</i>	<i>349 (HR)</i>	<i>3457642</i>

Unable to determine customer relationships (CRM)

Unable to analyze employee benefits trends

Data Duplication: House Holding

- Group together all records that belong to the same household.

Why bother ?

.....	S. Ahad	440, Munir Road, Lahore
.....		
.....	Shiekh Ahad	No. 440, Munir Rd, Lhr
.....		
.....	Shiekh Ahed	House # 440, Munir Road, Lahore

Data Duplication: Individualization

- Identify multiple records in each household which represent the same individual

.....	M. Ahad	440, Munir Road, Lahore
.....		
.....	Maj Ahad	440, Munir Road, Lahore

Address field is standardized.

By coincidence ??_

Formal definition & Nomenclature

- Problem statement:
 - “Given two databases, identify the potentially matched records **Efficiently** and **Effectively**”
- Many names, such as:
 - Record linkage
 - Merge/purge
 - Entity reconciliation
 - List washing and data cleansing.
- Current market and tools heavily centered towards customer lists.

Need & Tool Support

- Logical solution to dirty data is to clean it in some way.
- Doing it manually is very slow and prone to errors.
- Tools are required to do it “cost” effectively to achieve reasonable quality.
- Tools are there, some for specific fields, others for specific cleaning phase.
- Since application specific, so work very well, but need support from other tools for broad spectrum of cleaning problems.

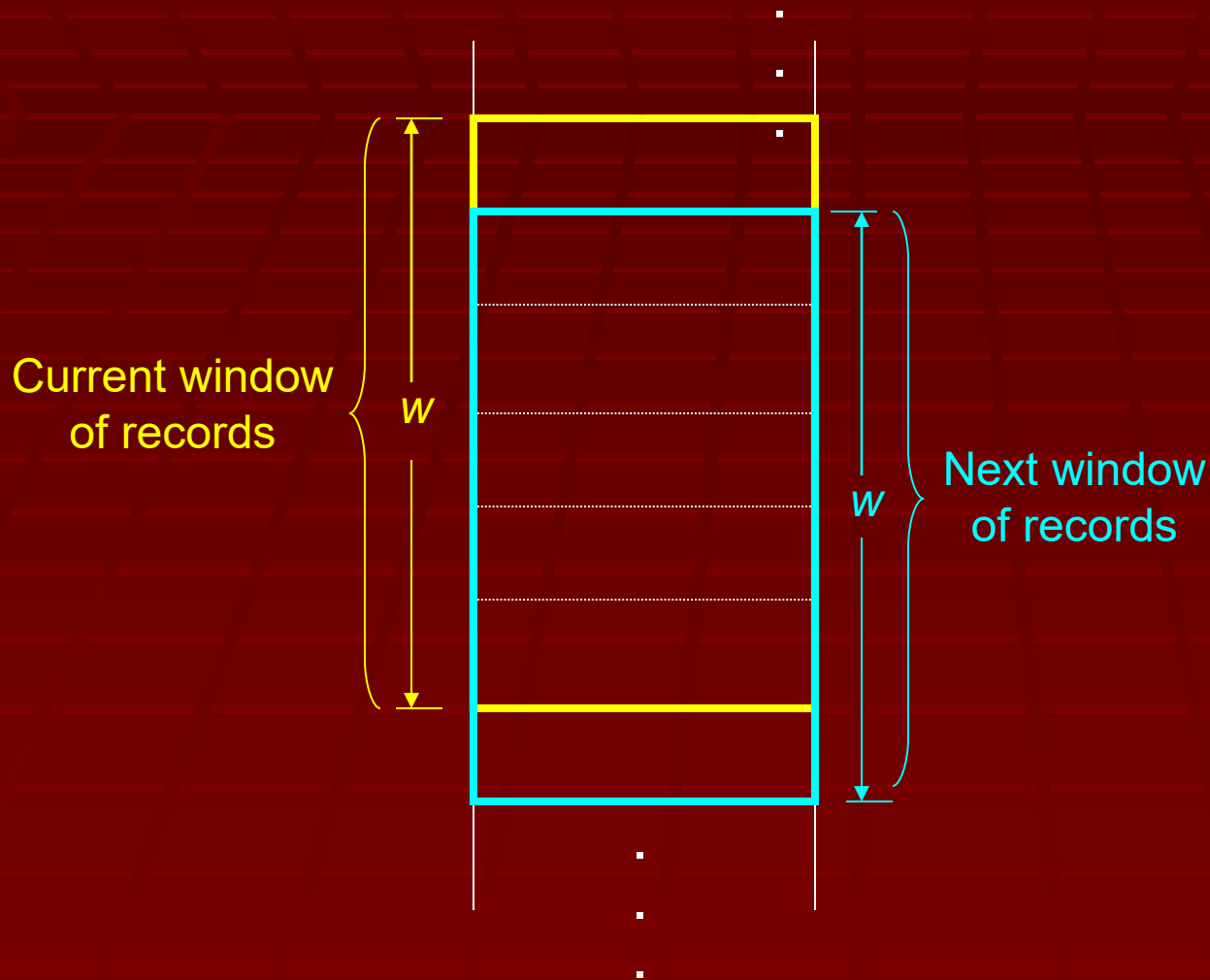
Overview of the Basic Concept

- In its simplest form, there is an identifying attribute (or combination) per record for identification.
- Records can be from single source or multiple sources sharing same PK or other common unique attributes.
- Sorting performed on identifying attributes and neighboring records checked.
- **What if no common attributes or dirty data?**
- The degree of similarity measured numerically, different attributes may contribute differently.

Basic Sorted Neighborhood (BSN) Method

- Concatenate data into one sequential list of N records
- **Steps 1: Create Keys**
 - Compute a key for each record in the list by extracting relevant fields or portions of fields
 - Effectiveness of the this method highly depends on a properly chosen key
- **Step 2: Sort Data**
 - Sort the records in the data list using the key of step 1
- **Step 3: Merge**
 - Move a fixed size window through the sequential list of records limiting the comparisons for matching records to those records in the window
 - If the size of the window is w records then every new record entering the window is compared with the previous $w-1$ records.

BSN Method : Sliding Window



BEFORE	AFTER
0.333333	0.285714
1.142857	0.285714
1.666667	0.333333
0.285714	0.333333
1.333333	0.666667
1	0.666667
0.857143	0.857143
1.333333	0.857143
0.333333	1
0.666667	1
1.333333	1.142857
1.142857	1.142857
0.666667	1.333333
1.333333	1.333333
1	1.333333
1.666667	1.333333
0.857143	1.666667
0.285714	1.666667

BSN Method: Selection of Keys

■ Selection of Keys

- Effectiveness highly dependent on the key selected to sort the records middle name vs. family name,
- A key is a sequence of a subset of attributes or sub-strings within the attributes chosen from the record.
- The keys are used for sorting the entire dataset with the intention that matched candidates will appear close to each other.

First	Middle	Address	NID	Key
Muhammed	Ahmad	440 Munir Road	34535322	AHM440MUN345
Muhammad	Ahmad	440 Munir Road	34535322	AHM440MUN345
Muhammed	Ahmed	440 Munir Road	34535322	AHM440MUN345
Muhammad	Ahmar	440 Munawar Road	34535334	AHM440MUN345

BSN Method: Problem with keys

- Since data is dirty, so keys **WILL** also be dirty, and matching records will not come together.
- Data becomes dirty due to data entry errors or use of abbreviations. Some real examples are as follows:

Technology
Tech.
Techno.
Tchnlgy

- Solution is to use external standard source files to validate the data and resolve any data conflicts.

BSN Method: Problem with keys (e.g.)

If contents of fields are not properly ordered, similar records will NOT fall in the same window.

Example: Records 1 and 2 are similar but will occur far apart.

No	Name	Address	Gender
1	N. Jaffri, Syed	No. 420, Street 15, Chaklala 4, Rawalpindi	M
2	S. Noman	420, Scheme 4, Rwp	M
3	Saiam Noor	Flat 5, Afshan Colony, Saidpur Road, Lahore	F

Solution is to TOKENize the fields i.e. break them further. Use the tokens in different fields for sorting to fix the error.

Example: Either using the name or the address field records 1 and 2 will fall close.

No	Name	Address	Gender
1	Syed N Jaffri	420 15 4 Chaklala No Rawalpindi Street	M
2	Syed Noman	420 4 Rwp Scheme	M
3	Saiam Noor	5 Afshan Colony Flat Lahore Road Saidpur	F

BSN Method: Matching Candidates

Merging of records is a complex inferential process.

Example-1: Two persons with names spelled nearly but not identically, have the exact same address. We infer they are same person i.e. Noma Abdullah and Noman Abdullah.

Example-2: Two persons have same National ID numbers but names and addresses are completely different. We infer same person who changed his name and moved or the records represent different persons and NID is incorrect for one of them.

Use of further information such as age, gender etc. can alter the decision.

Example-3: Noma-F and Noman-M we could perhaps infer that Noma and Noman are siblings i.e. brothers and sisters. Noma-30 and Noman-5 i.e. mother and son.

Complexity Analysis of BSN Method

- Time Complexity: $O(n \log n)$
 - $O(n)$ for Key Creation
 - $O(n \log n)$ for Sorting
 - $O(w n)$ for matching, where $w \leq 2 \leq n$
 - Constants vary a lot
- At least three passes required on the dataset.
- Complexity or rule and window size detrimental.
- For large sets disk I/O is detrimental.

BSN Method: Equational Theory

To specify the inferences we need equational Theory.

- Logic is NOT based on string equivalence.
- Logic based on domain equivalence.
- Requires declarative rule language.

Virtual University of Pakistan

Data Warehousing

Lecture-21

Introduction to Data Quality Management (DQM)

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

Introduction to Data Quality Management (DQM)

What is Quality? Informally

Some things are better than others i.e. they are of higher quality. How much “better” is better?

Is the right item the best item to purchase? How about after the purchase?

What is quality of service? The bank example

What is Quality? Formally

“Quality is conformance to requirements”

P. Crosby, “Quality is Free” 1979

“Degree of excellence”

Webster’s Third New International Dictionary

What is Quality? Examples from Auto Industry

Quality means meeting customer's needs, not necessarily exceeding them.

Quality means improving things customers *care about*, because that makes their lives easier and more comfortable.

Why example from auto-industry?

What is Data Quality?

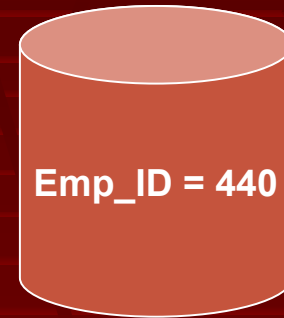
What is Data?

Note Change
the picture



Muhammad Khan

Height = 5'8"
Weight = 160 lbs
Gender = Male
Age = 35 yrs



All data is an abstraction of something real

What is Data Quality?

Intrinsic Data Quality

Electronic reproduction of reality.

Realistic Data Quality

Degree of utility or value of data to business.

Data Quality & Organizations

Intelligent Learning Organization:

High-quality data is an *open, shared* resource with *value-adding* processes.

The dysfunctional learning organization:

Low-quality data is a *proprietary* resource with *cost-adding* processes.

{Comment: Put picture of person in water holding round tube with data written on it}

Orr's Laws of Data Quality

Law #1 - “Data that is not used cannot be correct!”

Law #2 - “Data quality is a function of its use, not its collection!”

Law #3 - “Data will be no better than its most stringent use!”

Law #4 - “Data quality problems increase with the age of the system!”

Law #5 – “The less likely something is to occur, the more traumatic it will be when it happens!”

Total Quality Control (TQM)

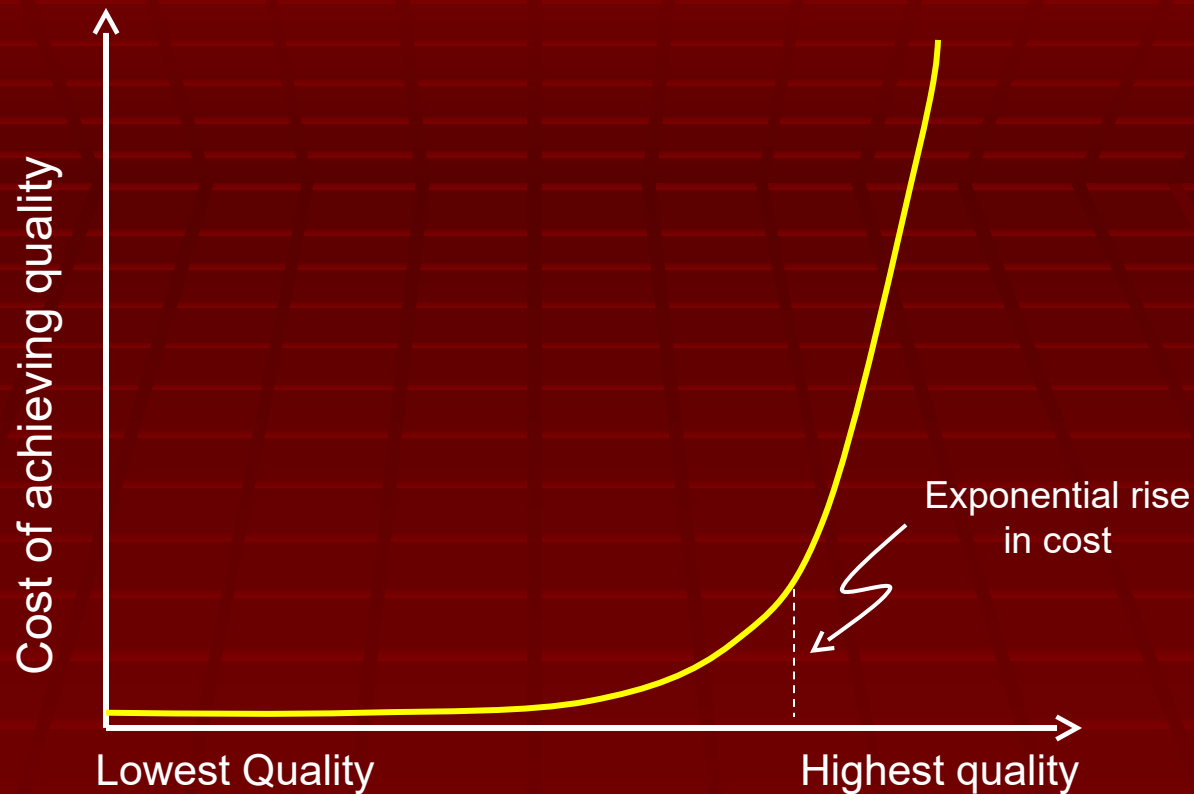
Philosophy of involving all for systematic and continuous improvement.

It is customer oriented. Why?

TQM incorporates the concept of product quality, process control, quality assurance, and quality improvement.

Quality assurance is NOT Quality improvement

Co\$ of fixing data quality



- Defect **minimization** is economical.
- Defect **elimination** is very very expensive.

Co\$t of Data Quality Defects

- **Controllable Costs**
 - Recurring costs for analyzing, correcting, and preventing data errors
- **Resultant Costs**
 - Internal and external failure costs of business opportunities missed.
- **Equipment & Training Costs**

Where data quality is critical?

- **Almost everywhere, some examples:**
- Marketing communications.
- Customer matching.
- Retail house-holding.
- Combining MIS systems after acquisition.

Characteristics or Dimensions of Data Quality

Data Quality Characteristic	Definition
Accuracy	Qualitatively assessing lack of error, high accuracy corresponding to small error.
Completeness	The degree to which values are present in the attributes that require them.

Completeness Vs Accuracy

95% accurate and 100% complete

OR

100% accurate and 95% complete

Which is better?

Depends on data quality (i) tolerances, the (ii) corresponding application and the (iii) cost of achieving that data quality vs. the (iv) business value.

Characteristics or Dimensions of Data Quality

Data Quality Characteristic	Definition
Consistency	A measure of the degree to which a set of data satisfies a set of constraints.
Timeliness	A measure of how current or up to date the data is.
Uniqueness	The state of being only one of its kind or being without an equal or parallel.
Interpretability	The extent to which data is in appropriate languages, symbols, and units, and the definitions are clear.
Accessibility	The extent to which data is available, or easily and quickly retrievable
Objectivity	The extent to which data is unbiased, unprejudiced, and impartial

Virtual University of Pakistan

Data Warehousing

Lecture-22

DQM: Quantifying Data Quality

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

Background

Companies want to measure the quality of their data that requires usable metrics.

Have to deal with both the subjective perceptions and objective measurements.

Text will not go to graphics

Subjective data quality assessments reflect the needs and experiences of stakeholders.

Objective assessments can be task-independent or task-dependent.

Task-independent metrics reflect states of the data without the contextual knowledge of the application.

Task dependent metrics, include organization's business rules, regulations etc.

We will discuss objective assessment and validation techniques (dependent & independent), if time permits will briefly cover subjective assessment too.

More on Characteristics of Data Quality

Data Quality Dim	Definition	Only this column will go to graphics
Believability	The extent to which data is regarded as true and credible.	
Appropriate Amount of Data	The extent to which the volume of data is appropriate for the task at hand.	
Timeliness	A measure of how current or up to date the data is.	
Accessibility	The extent to which data is available, or easily and quickly retrievable	
Objectivity	The extent to which data is unbiased, unprejudiced, and impartial.	
Interpretability	The extent to which data is in appropriate languages, symbols, and units, and the definitions are clear.	
Uniqueness	The state of being only one of its kind or being without an equal or parallel.	

Data Quality Assessment Techniques

- Ratios
- Min-Max

Data Quality Assessment Techniques

- Simple Ratios
 - Free-of-Error
 - Completeness
 - Schema
 - Column
 - Population
 - Consistency

Sub-Sub-bullets will not go to graphics

Ratio of violations to total number of consistency checks.

Data Quality Assessment Techniques

- Min-Max

- Used for multiple values, based on aggregation
- Min is conservative, while max is liberal

Sub-bullets and keys will not go to graphics

- Believability

- Comparison with a standard or experience
- $\text{Min } \{0.8, 0.7, 0.6\} = 0.6$
- Weighted average

- Appropriate Amount of Data

$\text{Min } \{D_p/D_n, D_n/D_p\}$

D_p : Data units provided
 D_n : Data units needed⁶

Data Quality Assessment Techniques

- Min-Max

- Timeliness

- $$\text{Max} \{0, 1 - C/V\} \quad C = A + Dt - It$$

C: Currency

V: Volatility

A: Age

Dt: Delivery time

It: Input time (received in system)

- Accessibility

- $$\text{Max} \{0, 1 - \text{Trd}/\text{Tru}\}$$

Sub-bullets and keys will not go to graphics

Trd: Time between request
by user to delivery

Tru: Request by user to time
data remains useful

Data Quality Validation Techniques

- Referential Integrity (RI).
- Attribute domain.
- Using Data Quality Rules.
- Data Histograming.

Referential Integrity Validation

RI checked every week or month, and no. of orphan records should be going down with time.

Yellow will not go to graphics

RI peculiar to DWH, not for operational systems

Example: How many outstanding payments in the DWH without a corresponding customer_ID in the customer table?

Business Case for RI

Not very interesting to know number of outstanding payments from a business point of view.

Interesting to know the actual amount outstanding, on per year basis, per region basis...

Performance Case for RI

Cost of enforcing RI is very high for large volume DWH implementations, therefore:

- Should RI constraints be turned OFF in a data warehouse? or
- Should those records be “discarded” that violate one or more RI constraints?

3 steps of Attribute Domain Validation

Step-1: Capture and quantify the occurrences of each domain value within each coded attribute of the database.

Yellow will go to graphics

Step-2: Compare actual content of attributes against set of valid values.

Step-3: Investigate exceptions to determine cause and impact of the data quality defects.

Note: Step 3 (above) applies to all defect types.

Attribute Domain Validation: What next?

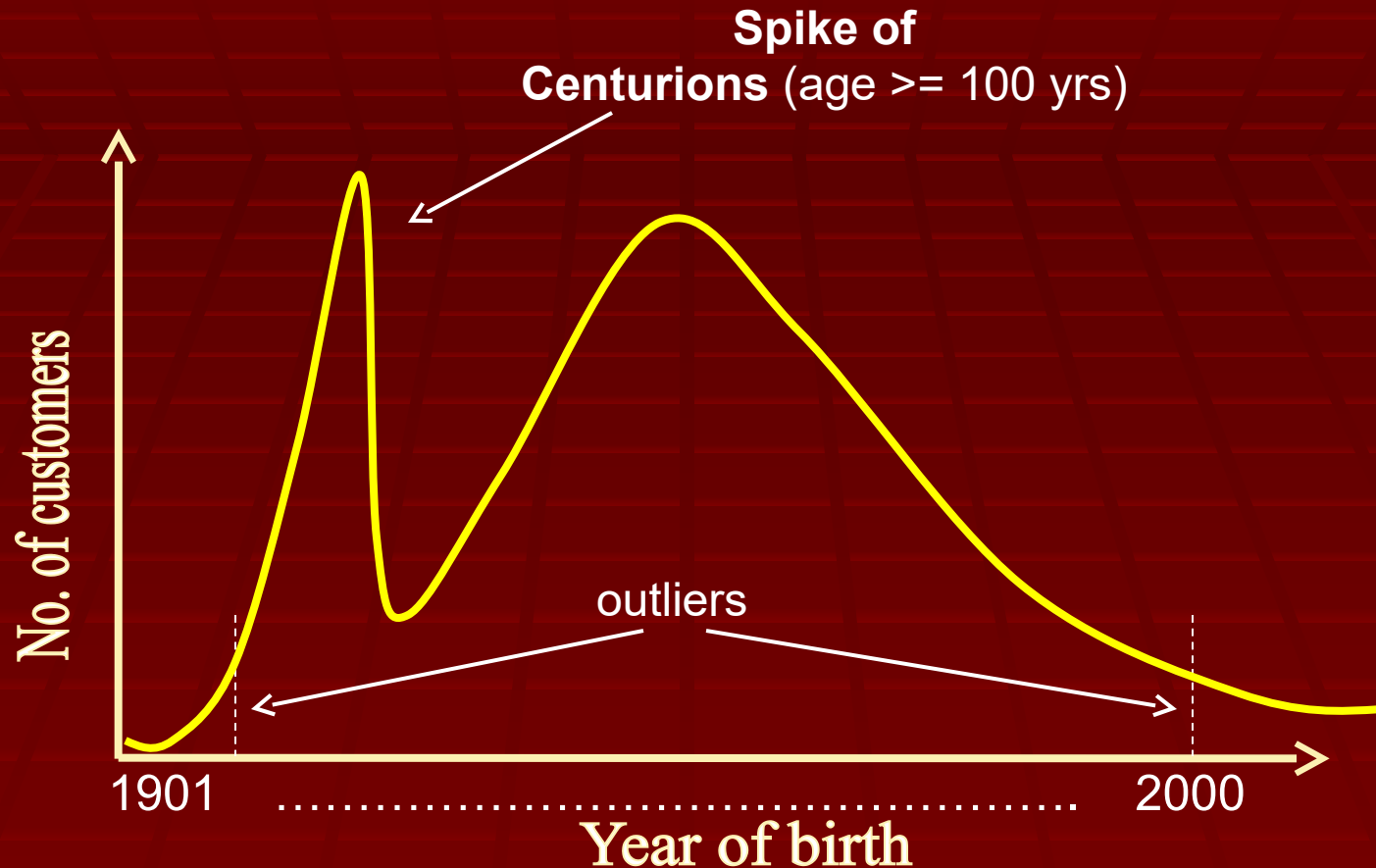
What to do next?

- Trace back to source cause(s).
- Quantify business impact of the defects.
- Assess cost (and time frame) to fix and proceed accordingly.

Data Quality Rules

Historical Data Problem	Rule Type	Generic Rule Set	Specific Rule Set
Equipment identifier fields are often blank.	Null Constraints	If the equipment identifier is blank or null, then, error.	Select equip_id from equip or equip_id = ' ' or equip_id = NULL;
The code for DEBIT/CREDIT is sometimes not 'D' or 'C'.	Domain Validation	If Debit/Credit code is not 'D' or 'C', then, error.	Select debit_code from transaction where debit_code not = 'D' or 'C';
The value of unit price is not greater than zero.	Operational Rule Set	If unit price = \$00.00, then, error.	Select * from equip where unit_price = 00.00;
The total charge for a credit card purchase exceeds \$25,000.	Business Rule Validation	If total_charge is greater than \$25K, then, error.	Select total from charge when total > 25000;

Statistical Validation using Histogram



NOTE: For a certain environment, the above distribution may be perfectly normal.

Virtual University of Pakistan

Data Warehousing

Lecture-23

Total DQM

Ahsan Abdullah

Assoc. Prof. & Head

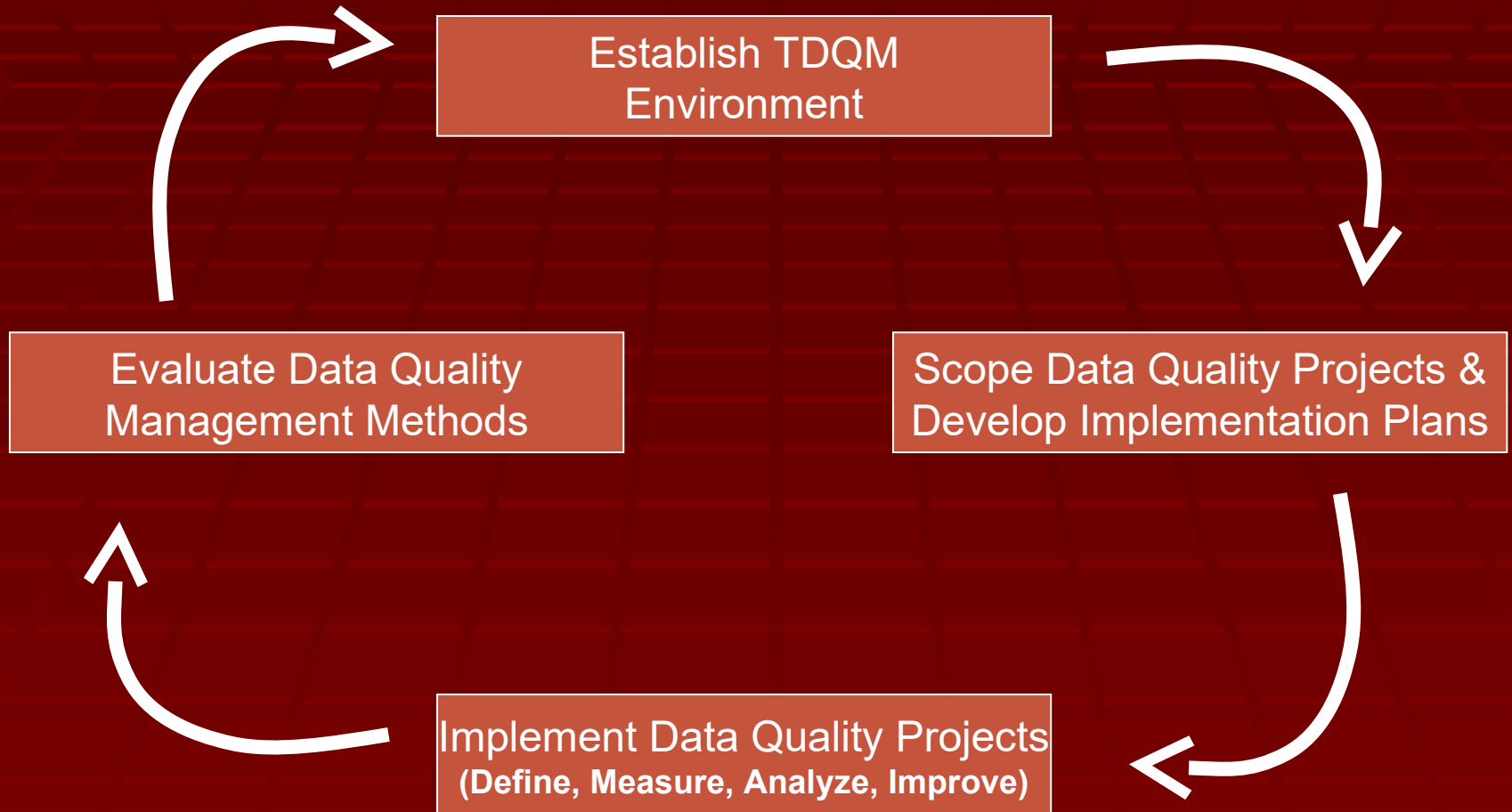
Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

Data Quality Management Process



Data Quality Management Process

1. Establish Data Quality Management Environment

Sub-bullets will not go to graphics

- **IS project managers**
- **Development professionals.**
- **Functional users of legacy information systems with domain knowledge**
- **IS developers know solutions but don't know how and where to modify**

Data Quality Management Process

yellow will go to graphics

2. Scope Data Quality Projects & Develop Implementation Plans

- **Task Summary:** Project goals, scope, and potential benefits
- **Task Description:** Describe data quality analysis tasks
- **Project Approach:** Summarize tasks and tools used to provide a baseline of existing data quality
- **Schedule:** Identify task start, completion dates, and project milestones
- **Resources:** Include costs connected with tools acquisition, labor hours (by labor category), training, travel, and other direct and indirect costs

Data Quality Management Process

3. Implement Data Quality Projects (Define, Measure, Analyze, Improve)

Yellow will go to graphics

- **Define:** Identify functional user DQ requirements and establish DQ metrics
- **Measure:** conformance to current business rules and develop exception reports
- **Analyze:** Verify, validate, and assess poor DQ causes. Define improvement opportunities
- **Improve:** Select/prioritize DQ improvement opportunities i.e. data entry procedures, updating data validation rules, and/or company data standards.

Data Quality Management Process

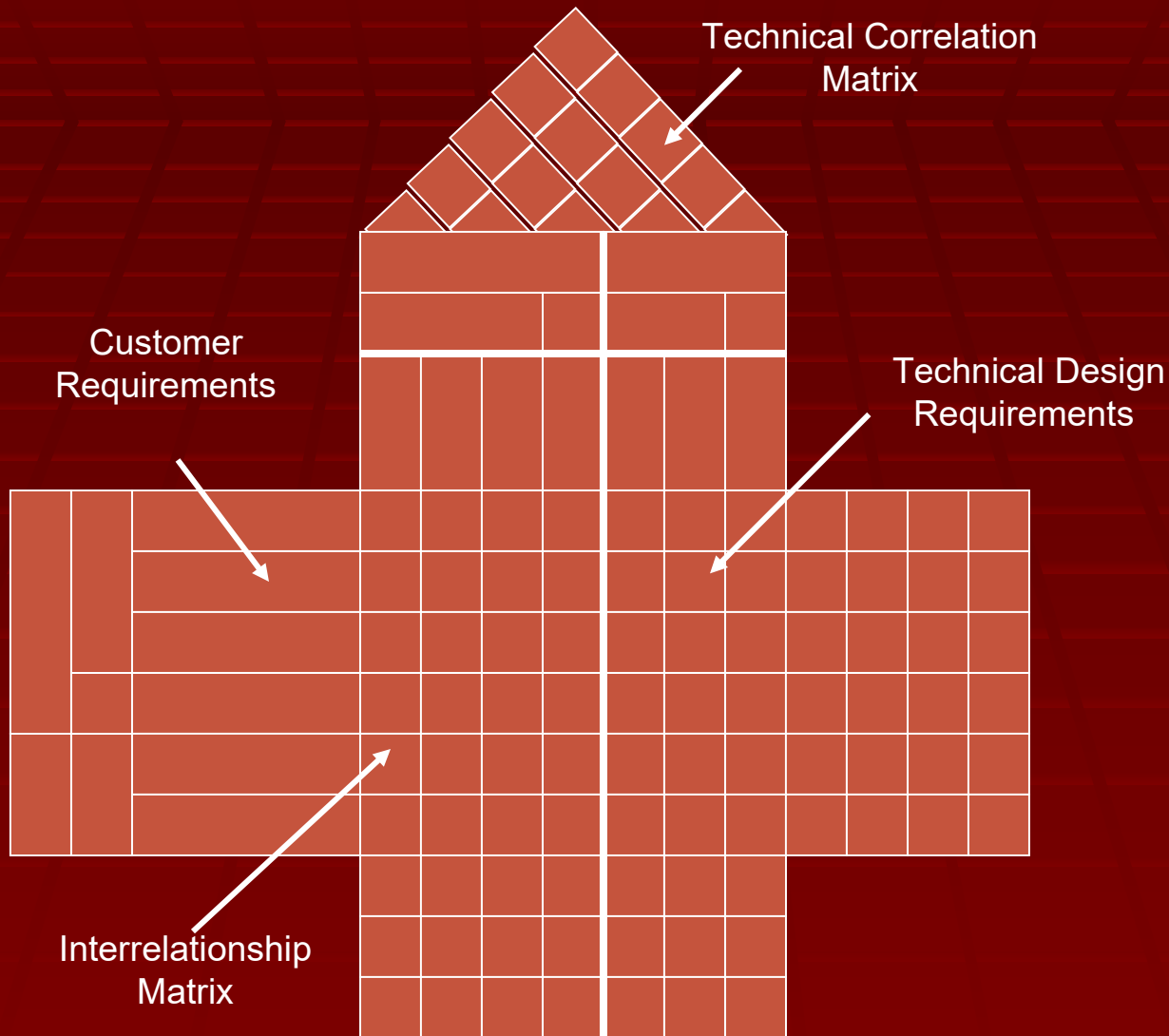
4. Evaluate Data Quality Management Methods

- modifying or rejuvenating existing methods of DQ management
- determining if DQ projects have helped to achieve demonstrable goals and benefits.

Sub-bullets will not go to graphics

Evaluating and assessing DQ work as, it is not a program, but a new way of doing business.

The House of Quality Matrix



How to improve Data Quality?

The four categories of Data Quality Improvement

- Process
- System
- Policy & Procedure
- Data Design

Quality Management Maturity Grid

	Management understanding	Quality organ. status	Problem handling	Cost of quality % of sales	Company attitude
CMM Level-1 Uncertainty					
CMM Level-2 Awakening					
CMM Level-3 Enlightenment					
CMM Level-4 Wisdom					
CMM Level-5 Certainty					

Misconceptions on Data Quality

- **You Can Fix Data**

- Problem NOT in data, but how it was used.
- It is NOT a one time process.
- Buying a cleansing tool is NOT the solution.
- Some live with the problem, cant afford the tool.

- **Data Quality is an IT Problem**

- It is the company problem.
- Define the metrics of quality.
- Business has to strike a balance between quality and ROI.
- Joint business and IT effort.

Sub-bullets will not go to graphics

Misconceptions on Data Quality

- **(All) Problem is in the Data Sources or Data Entry**
 - NOT the only problem.
 - Systems could be responsible, but actually it is the metrics.
 - Two divisions using different codes for same entity.
 - Need to track, trace, check data from creation to usage.
- **The Data Warehouse will provide a single source of truth**
 - In ideal world it is indeed true.
 - In real world maybe multiple data warehouses, data marts, external source i.e. silos of data resulting in multiple sources of “truth”.
 - Even with single source of truth, if transformations and interpretations are different, an issue.

Sub-bullets will not go to graphics



Virtual University of Pakistan

Data Warehousing

Lecture-24

Need for Speed: Parallelism

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

Background

When to parallelize?

Useful for operations that access significant amounts of data.

Useful for operations that can be implemented independent of each other “Divide-&-Conquer”

Parallel execution improves processing for:

- Size** ■ Large table scans and joins
- Size** ■ Creation of large indexes
- D&C** ■ Partitioned index scans
- Size** ■ Bulk inserts, updates, and deletes
- D&C** ■ Aggregations and copying

Are you ready to parallelize?

Parallelism can be exploited, if there is...

- Symmetric multi-processors (SMP), clusters, or Massively Parallel (MPP) systems **AND**
- Sufficient I/O bandwidth **AND**
- Underutilized or intermittently used CPUs (for example, systems where CPU usage is typically less than 30%) **AND**
- Sufficient memory to support additional memory-intensive processes such as sorts, hashing, and I/O buffers

Word of caution

Parallelism can reduce system performance on over-utilized systems or systems with small I/O bandwidth.

Scalability – Size is NOT everything

Index usage

- Hash based
- B-Tree
- Multiple
- Bitmapped

Number of Concurrent Users



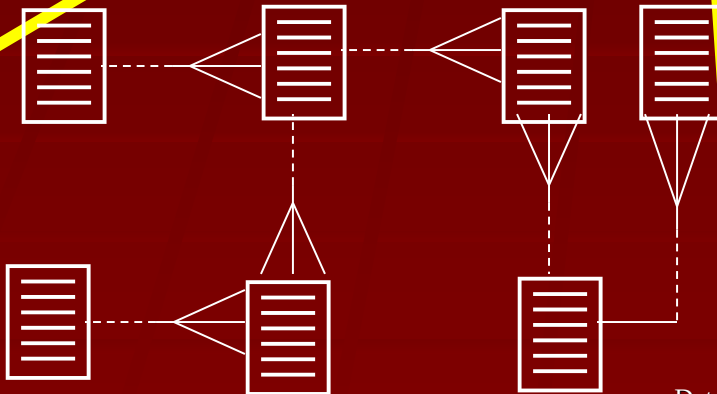
Complexity of Technique

- Simple table retrieval
- Moderate complexity Join
- Propensity analysis
- Clustering

Amount of detailed data



Complexity of Data Model

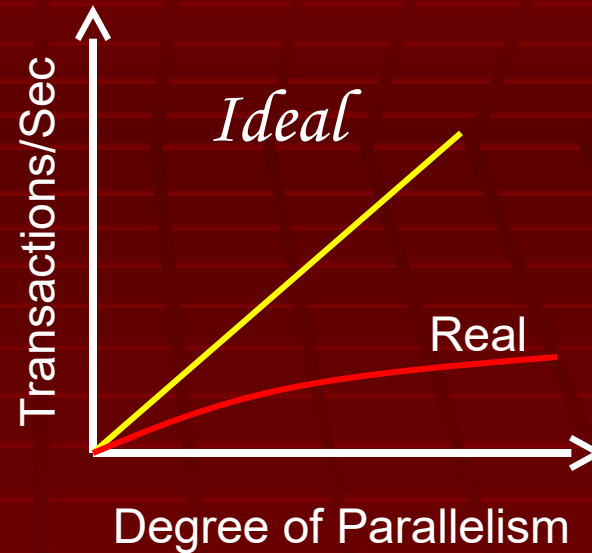


Data Warehousing

Scalability- Speed-Up & Scale-Up

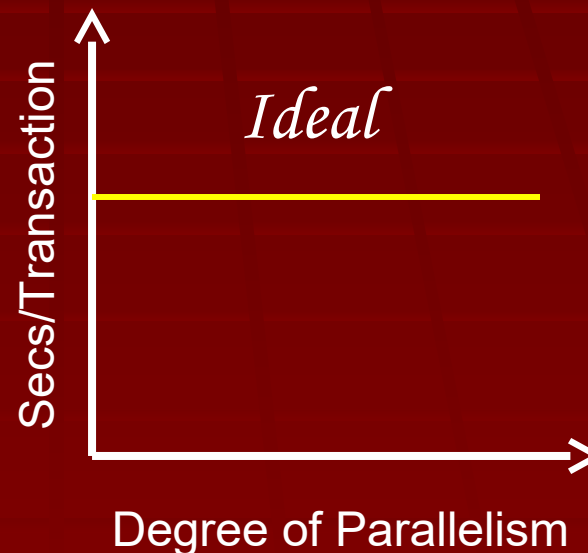
Speed-Up

More resources means proportionally less time for given amount of data.



Scale-Up

If resources increased in proportion to increase in data size, time is constant.



Quantifying Speed-up

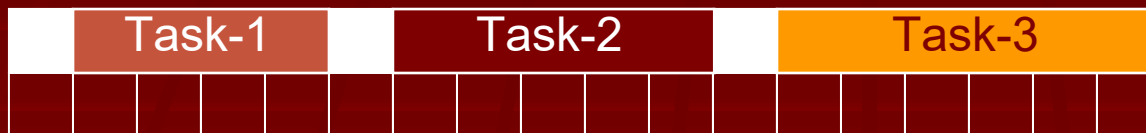
$$\text{Speedup} = \frac{T_s}{T_m}$$

T_s : Time on serial processor

T_m : Time on multiple processors

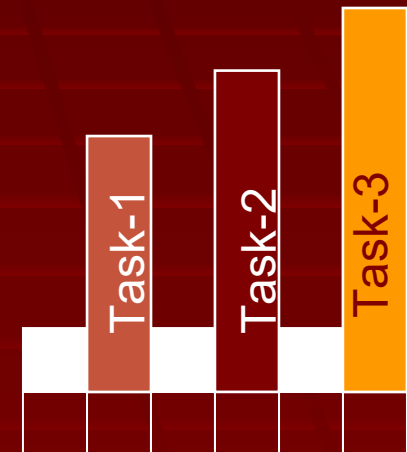
Sequential Execution

■ Control work (“overhead”)



18 time units

Ideal Parallel Execution



6 time units

$$\text{Speedup} = \frac{18}{6} = 300\%$$

Speed-Up & Amdahl's Law

Reveals maximum expected **speedup** from **parallel** algorithms given the **proportion** of task that must be computed **sequentially**. It gives the speedup S as

$$S \leq \frac{1}{f + (1 - f)/N}$$

Only formula and explanation will go to graphics

f is the fraction of the problem that must be computed sequentially
 N is the number of processors

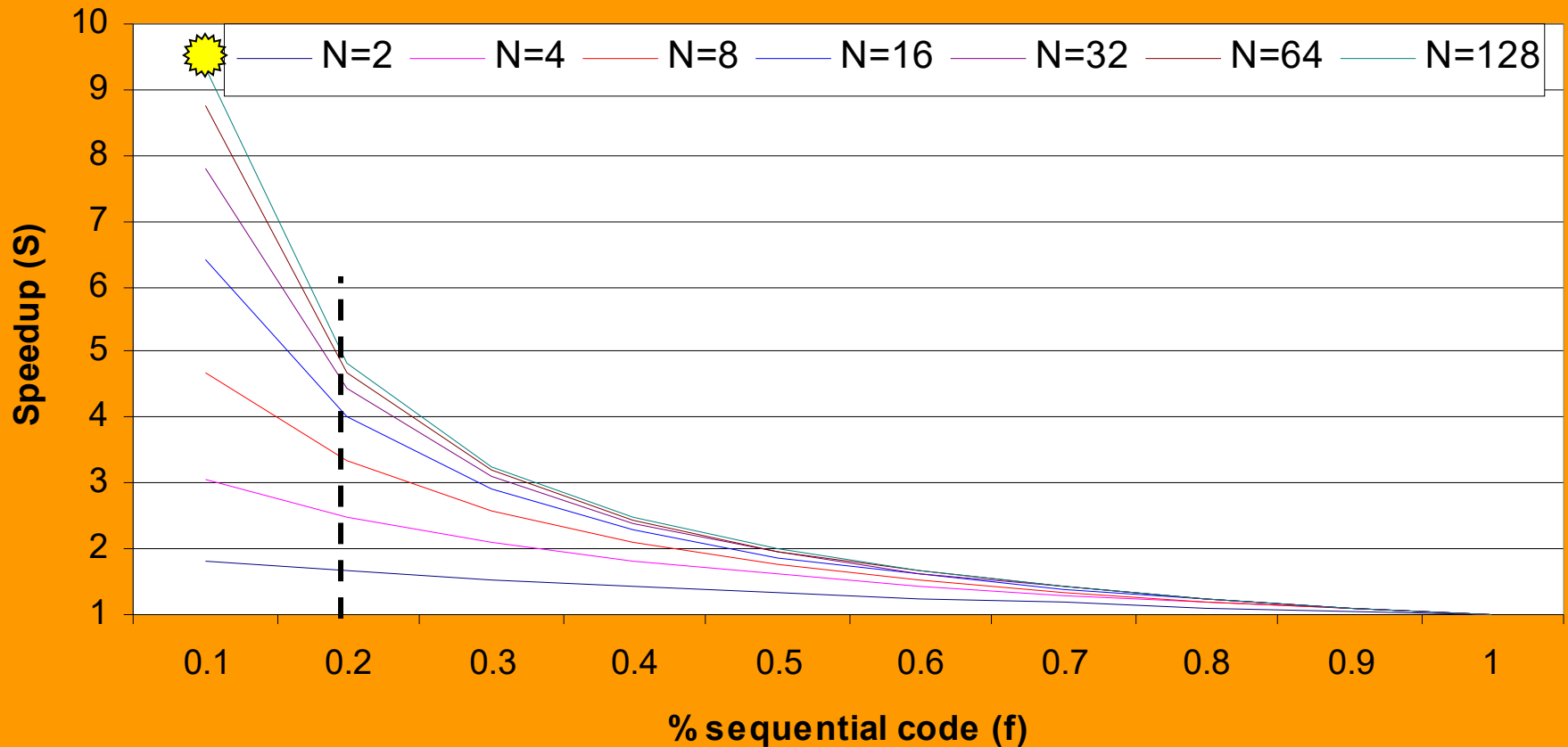
As f approaches 0, S approaches N

Example-1: $f = 5\%$ and $N = 100$ then $S = 16.8$

Example-2: $f = 10\%$ and $N = 200$ then $S = 9.56$

Not
1:1
Ratio
8

Amdahl's Law: Limits of parallelization



For less than 80% parallelism, the speedup drastically drops.

At 90% parallelism, **128** processors give performance of less than **10** processors.

Parallelization OLTP Vs. DSS

There is a big difference.

DSS

Parallelization of a SINGLE query

OLTP

Parallelization of MULTIPLE queries

Or Batch updates in parallel

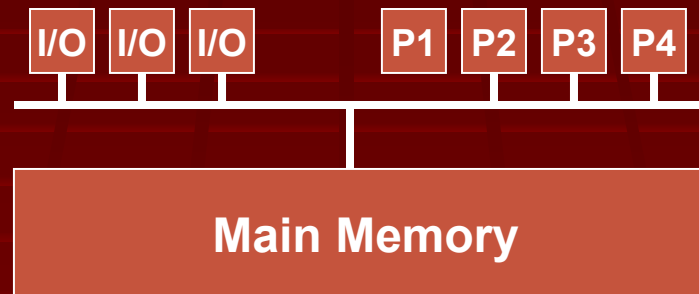
Brief Intro to Parallel Processing

- **Parallel Hardware Architectures**
 - Symmetric Multi Processing (SMP)
 - Distributed Memory or Massively Parallel Processing (MPP)
 - Non-uniform Memory Access (NUMA)
- **Parallel Software Architectures**
 - Shared Memory
 - Shard Disk
 - Shared Nothing

Shared Everything
- **Types of parallelism**
 - Data Parallelism
 - Spatial Parallelism

Symmetrical Multi Processing (SMP)

- A number of independent I/O and number of processors all sharing access to a single large memory space.

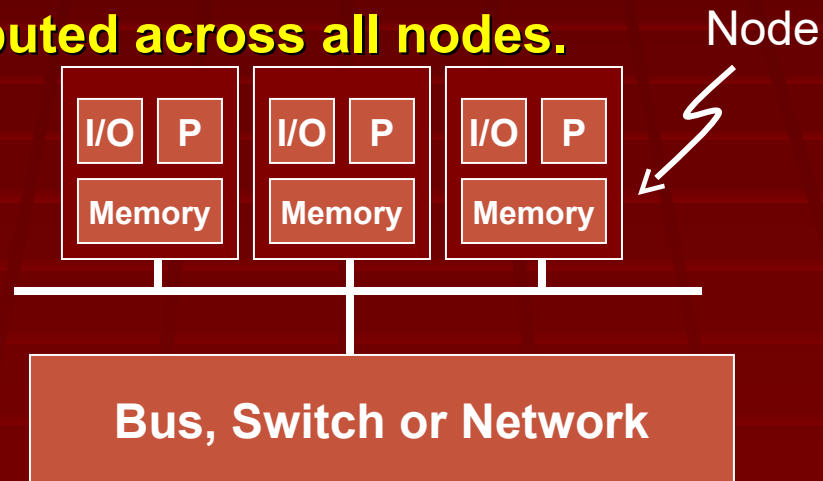


- Typically each CPU executes its job independently.
- Supports both Multi-Tasking and Parallel Processing.
- Have to deal with issues such as Cache Coherence, Processor Affinity and Hot Spots.

Yellow will not go to graphics

Distributed Memory Machines

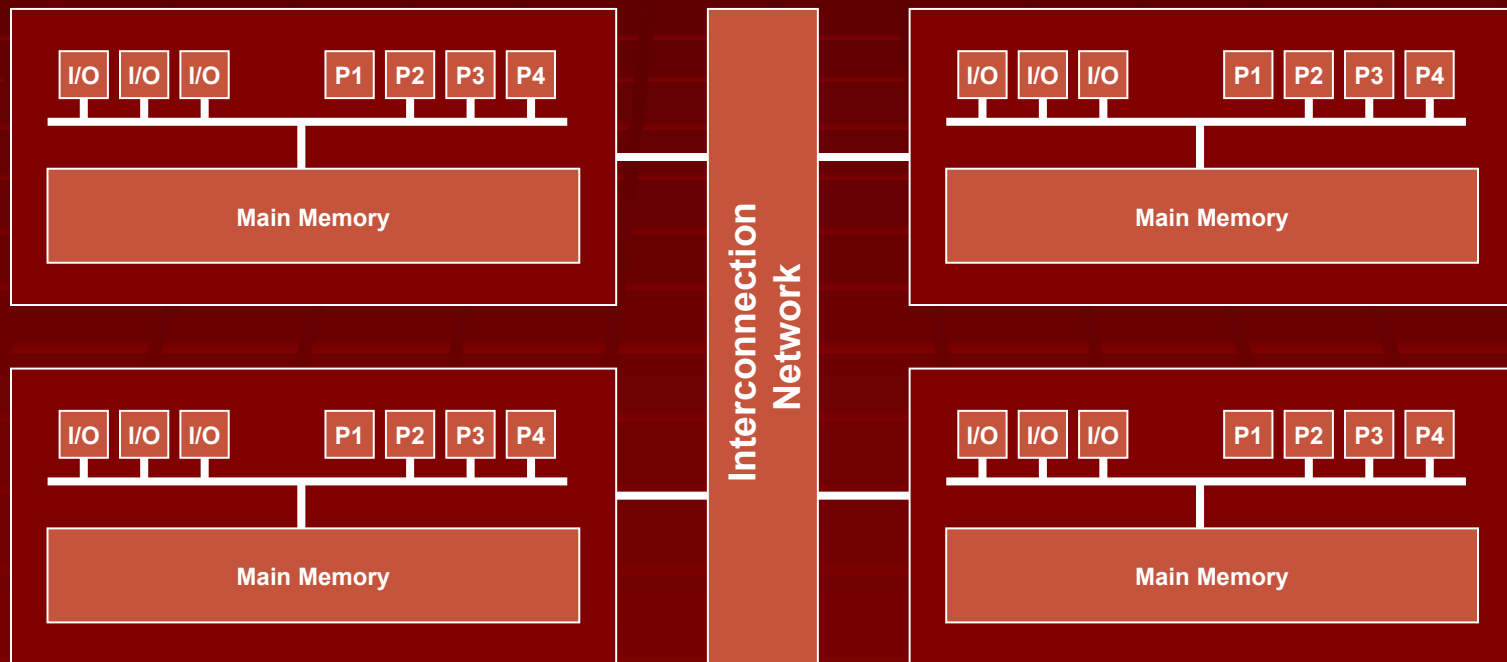
- Composed of a number of self-contained, self-controlled nodes connected through a network interface.
- Each node contains its own CPU, processor, memory and I/O.
- Architecture better known as Massively Parallel Processing (MPP) or cluster computing.
- Memory is distributed across all nodes.



- Network has the tendency to become the bottleneck.
- Issues fundamentally different from those in SMP.

Distributed Shared Memory Machines

A little bit of both worlds !



Shared disk RDBMS Architecture

Adv

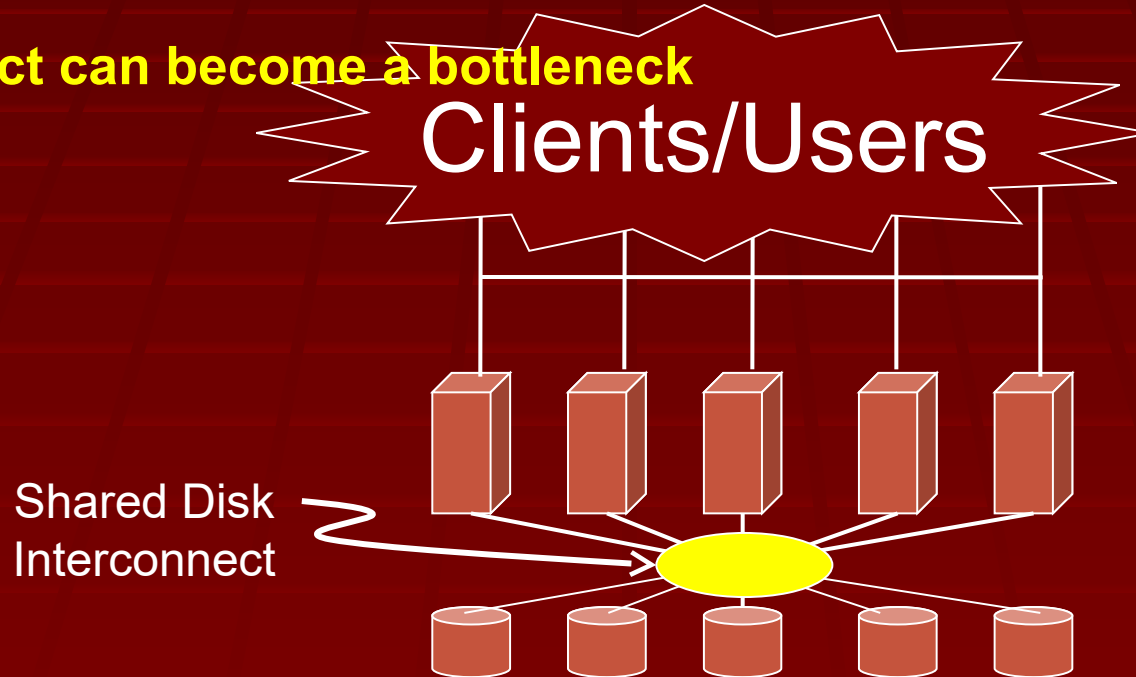
High level of fault tolerance

Dis Adv

Serialization due to locking

Interconnect can become a bottleneck

Yellow ill not go to graphics



Shared Nothing RDBMS Architecture

Adv

Data ownership changes infrequently

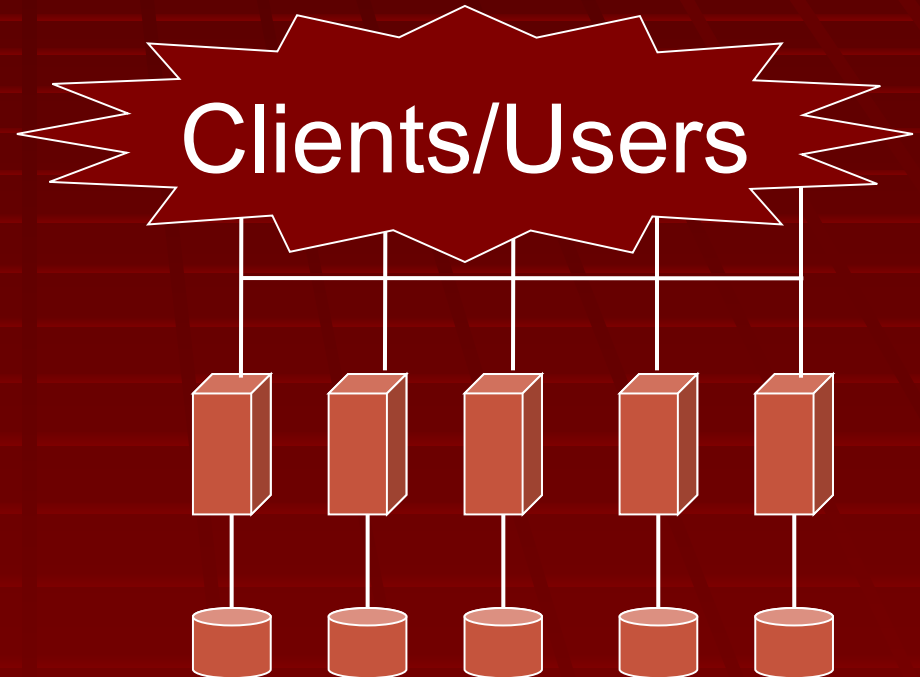
There is no locking

Dis Adv

Data availability low on failure

Very careful with data distribution

Redistribution is expensive



Yellow ill not go to graphics

Shared disk Vs. Shared Nothing RDBMS

- Important note: Do not confuse RDBMS architecture with hardware architecture.
- Shared nothing databases can run on shared everything (SMP or NUMA) hardware.
- Shared disk databases can run on shared nothing (MPP) hardware.

This slide will not go to graphics

Virtual University of Pakistan

Data Warehousing

Lecture-25

Need for Speed: Parallelism Methodologies

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan1010@yahoo.com

Motivation

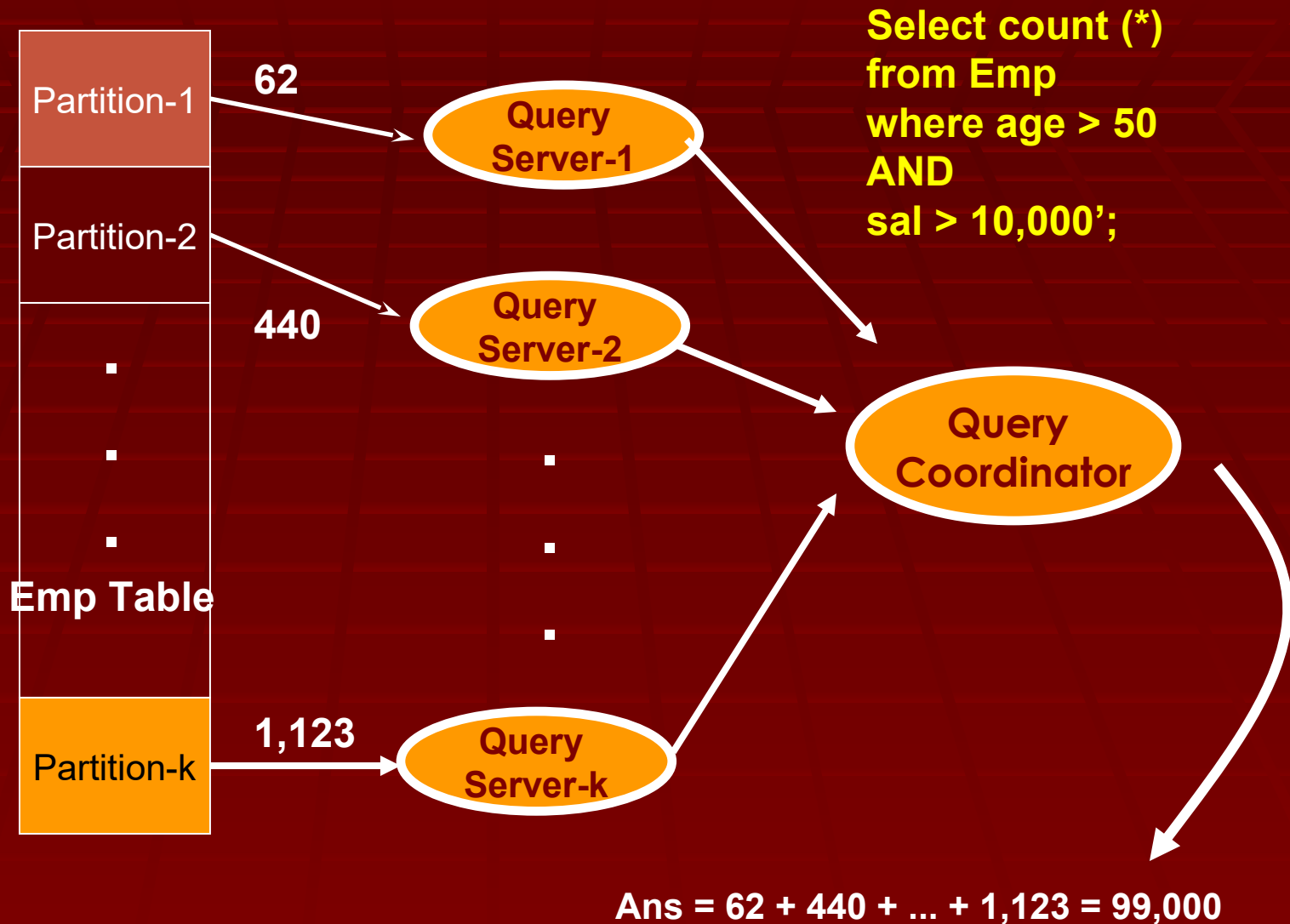
- No need of parallelism if perfect computer
 - with single infinitely fast processor
 - with an infinite memory with infinite bandwidth
 - and its infinitely cheap too (free!)
- Technology is not delivering (going to Moon analogy)
- The Challenge is to build
 - infinitely fast processor out of infinitely many processors of **finite speed**
 - Infinitely large memory with infinite memory bandwidth from infinite many **finite storage units** of **finite speed**

No text goes to graphics

Data Parallelism: Concept

- Parallel execution of a single data manipulation task across multiple partitions of data.
- Partitions static or dynamic No text goes to graphics
- Tasks executed almost-independently across partitions.
- “Query coordinator” must coordinate between the independently executing processes.

Data Parallelism: Example



Data Parallelism: Ensuring Speed-UP

To get a speed-up of N with N partitions, it must be ensured that:

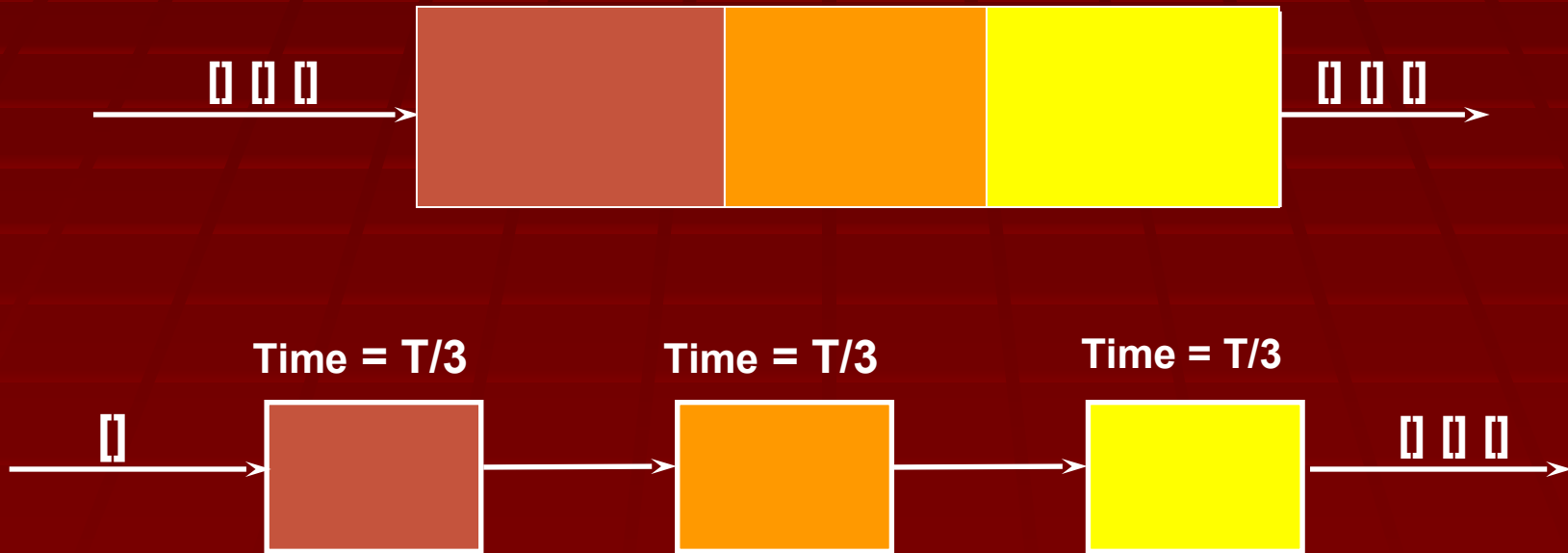
- There are enough computing resources **No text will go to graphics**
- Query-coordinator is very fast as compared to query servers.
- Work done in each partition almost same to avoid performance bottlenecks.
- Same number of records in each partition would not suffice.
- Need to have uniform distribution of records w.r.t filter criterion across partitions.

Temporal Parallelism (pipelining)

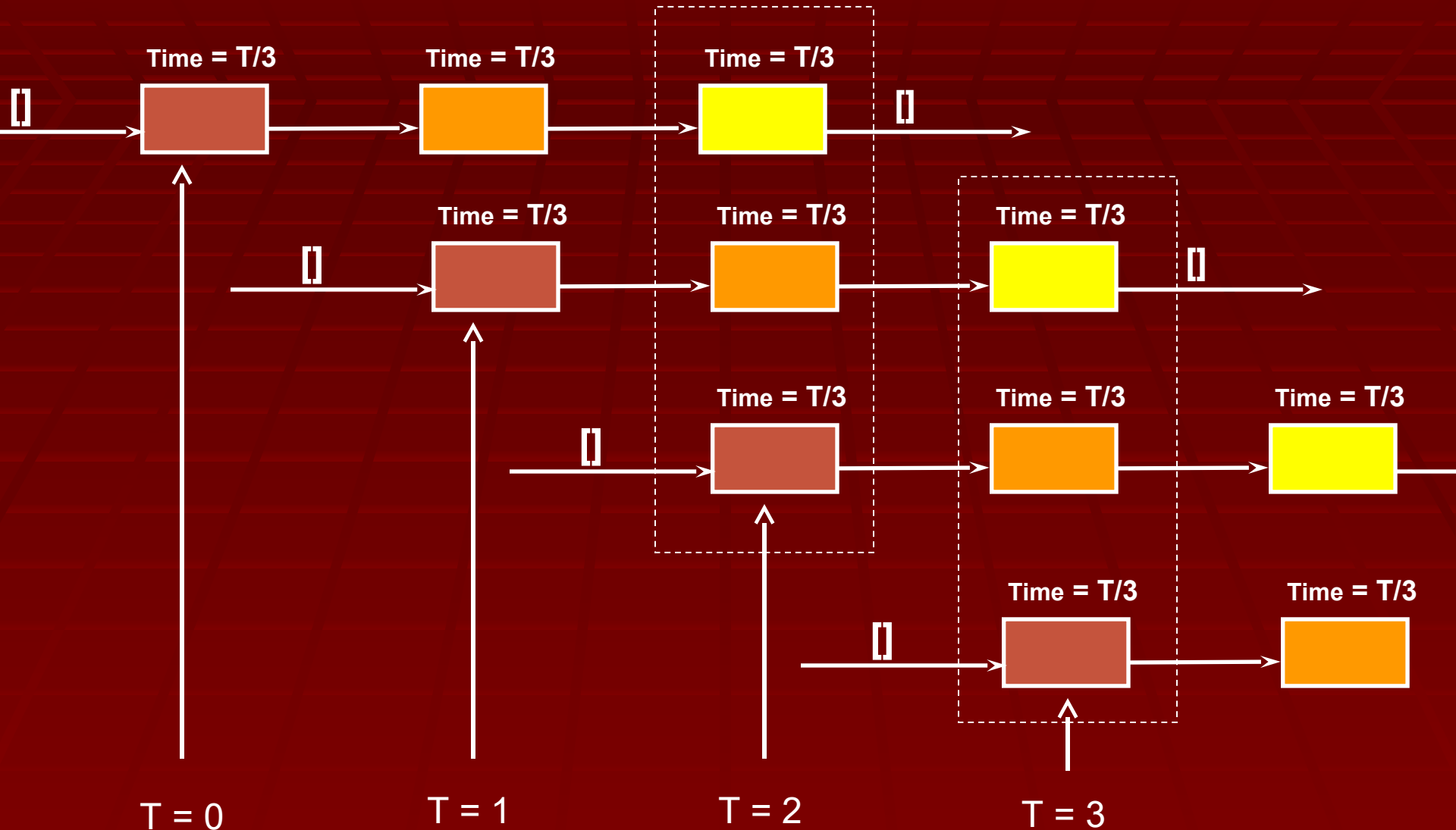
Involves taking a complex task and breaking it down into independent subtasks for parallel execution on a stream of data inputs.

No text goes to graphics

Task Execution Time = T



Pipelining: Time Chart



Pipelining: Speed-Up Calculation

Time for sequential execution of 1 task = T

Time for sequential execution of N tasks = $N * T$

(Ideal) time for pipelined execution of one task using an M stage pipeline = T

(Ideal) time for pipelined execution of N tasks using an M stage pipeline = $T + ((N-1) \times (T/M))$

$$\text{Speed-up (S)} = \frac{NT}{T + (N-1) \times \frac{T}{M}}$$

Pipeline parallelism focuses on increasing throughput of task execution, NOT on decreasing sub-task execution time.

Pipelining: Speed-Up Example

Example: Bottling soft drinks in a factory

10 CRATES LOADS OF BOTTLES

Sequential execution $= 10 \times T$

Fill bottle, Seal bottle, Label Bottle pipeline $= T + T \times (10-1)/3 = 4 \times T$

Speed-up = 2.50

20 CRATES LOADS OF BOTTLES

Sequential execution $= 20 \times T$

Fill bottle, Seal bottle, Label Bottle pipeline $= T + T \times (20-1)/3 = 7.3 \times T$

Speed-up = 2.72

40 CRATES LOADS OF BOTTLES

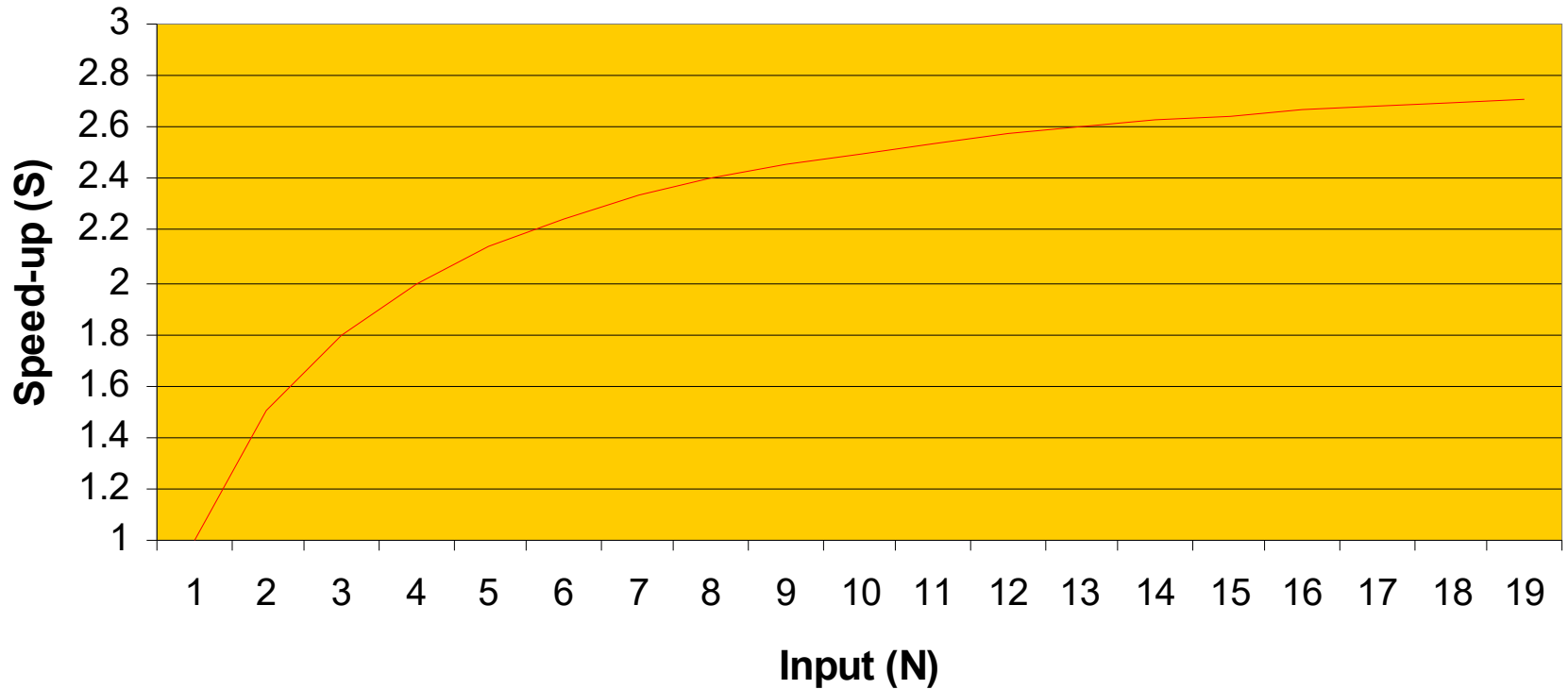
Sequential execution $= 40 \times T$

Fill bottle, Seal bottle, Label Bottle pipeline $= T + T \times (40-1)/3 = 14.0 \times T$

Speed-up = 2.85

Only 1st two examples will go to graphics

Pipelining: Input vs Speed-Up



Asymptotic limit on speed-up for M stage pipeline is M.

The speed-up will NEVER be M, as initially filling the pipeline took T time units.

Pipelining: Limitations

- Relational pipelines are rarely very long
 - Even a chain of length ten is unusual.
- Some relational operators do not produce first output until consumed all their inputs.
 - Aggregate and sort operators have this property. One cannot pipeline these operators.
- Often, execution cost of one operator is much greater than others hence skew.
 - e.g. Sum() or count() vs Group-by() or Join.

Partitioning & Queries

- Let's evaluate how well different partitioning techniques support the following types of data access:
 - **Full Table Scan:** *Scanning the entire relation*
 - **Point Queries:** Locating a tuple, e.g. where $r.A = 313$
 - **Range Queries:** Locating the value of a given attribute lies within a specified range. e.g., where $313 \leq r.A < 786$.

Partitioning & Queries

Round Robin

- Advantages
 - Best suited for sequential scan of entire relation on each query.
 - All disks have almost an equal number of tuples; retrieval work is thus well balanced between disks.
- Range queries are difficult to process
 - No clustering -- tuples are scattered across all disks

yellow goes to graphics

Partitioning & Queries

Hash Partitioning

- Good for sequential access
 - With uniform hashing and using partitioning attributes as a key, tuples will be equally distributed between disks.
- Good for point queries on partitioning attribute
 - Can lookup single disk, leaving others available for answering other queries.
 - Index on partitioning attribute can be local. Updates and inserts are update very efficient even joins. yellow goes to graphics
- Range queries are difficult to process
 - No clustering -- tuples are scattered across all disks

Partitioning & Queries

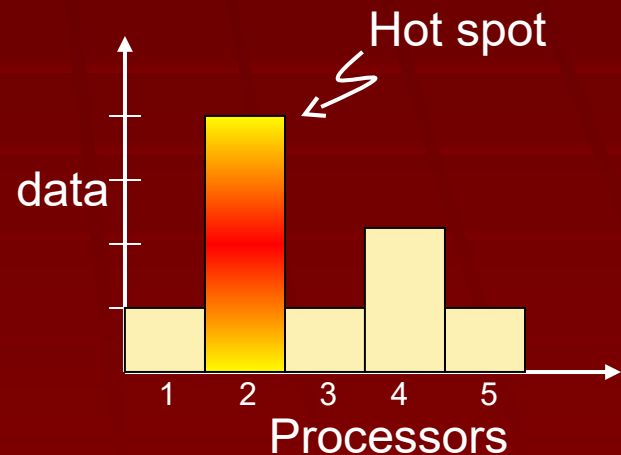
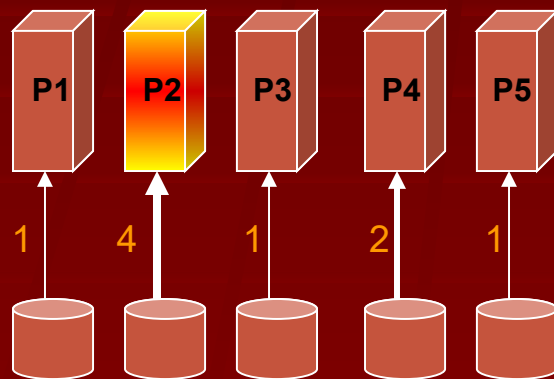
Range Partitioning

- Provides data clustering by partitioning attribute value.
- Good for sequential access
- Good for point queries on partitioning attribute: only one disk needs to be accessed.
- For range queries on partitioning attribute, one or a few disks may need to be accessed
 - Remaining disks are available for other queries.
 - Good if result tuples are from one to a few blocks.
 - If many blocks are to be fetched, they are still fetched from one to a few disks, then potential parallelism in disk access is wasted

yellow goes to graphics

Parallel Sorting

- Scan in parallel, and range partition on the go.
- As partitioned data becomes available, perform “local” sorting.
- Resulting data is sorted and again range partitioned.
- **Problem:** skew or “hot spot”.
- **Solution:** Sample the data at start to determine partition points.



Skew in Partitioning

- The distribution of tuples to disks may be **skewed**
 - i.e. some disks have many tuples, while others may have fewer tuples.
- **Types of skew:**
 - **Attribute-value skew.**
 - Some values appear in the partitioning attributes of many tuples; all the tuples with the same value for the partitioning attribute end up in the same partition.
 - Can occur with range-partitioning and hash-partitioning.
 - **Partition skew.**
 - With range-partitioning, badly chosen partition vector may assign too many tuples to some partitions and too few to others.
 - Less likely with hash-partitioning if a good hash-function is chosen.

yellow goes to graphics

Handling Skew in Range-Partitioning

- To create a balanced partitioning vector
 - **Sort** the relation on the partitioning attribute.
 - **Construct the partition vector** by scanning the relation in sorted order as follows.
 - After every $1/n^{th}$ of the relation has been read, the value of the partitioning attribute of the next tuple is added to the partition vector.
 - n denotes the number of partitions to be constructed.
 - **Duplicate entries or imbalances** can result if duplicates are present in partitioning attributes.

yellow goes to graphics 8

Barriers to Linear Speedup & Scale-up

- **Amdahal' Law**

yellow goes to graphics

- **Startup**

- Time needed to start a large number of processors.
- Increase with increase in number of individual processors.
- May also include time spent in opening files etc.

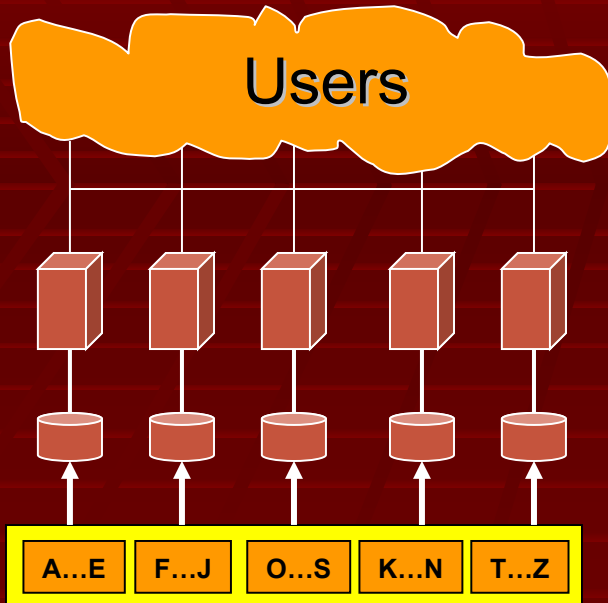
- **Interference**

- Slow down that each processor imposes on all others when sharing a common pool of resources “(e.g. memory).

- **Skew**

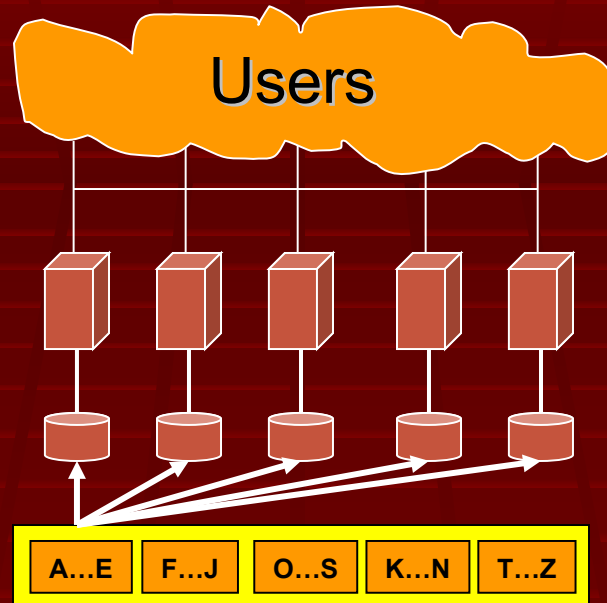
- Variance dominating the mean.
- Service time of the job is service time of its slowest components.

Comparison of Partitioning Techniques



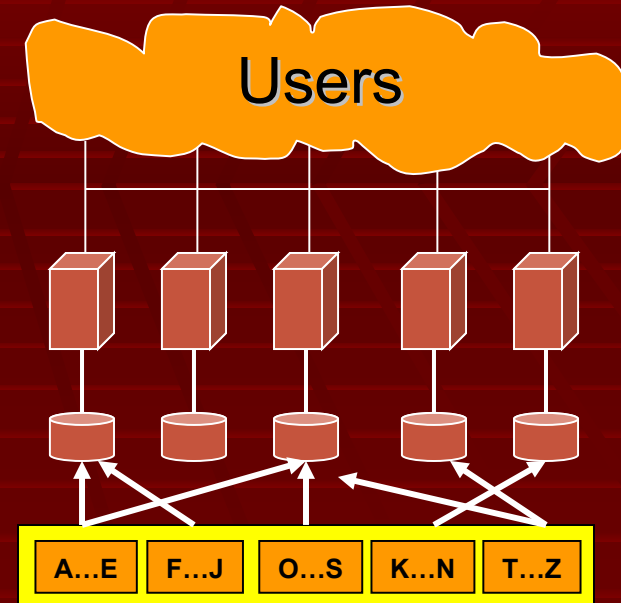
Good for equijoins, range queries, group-by clauses, can result in "hot spots".

Range



Good for load balancing, but impervious to nature of queries.

Round Robin



Good for equijoins, can result in uneven data distribution

Hash

Shared disk/memory less sensitive to partitioning.

Shared nothing can benefit from good partitioning.

Parallel Aggregates

For each aggregate function, need a decomposition:

$$\text{Count}(S) = \sum \text{count}(s_1) + \sum \text{count}(s_2) + \dots$$

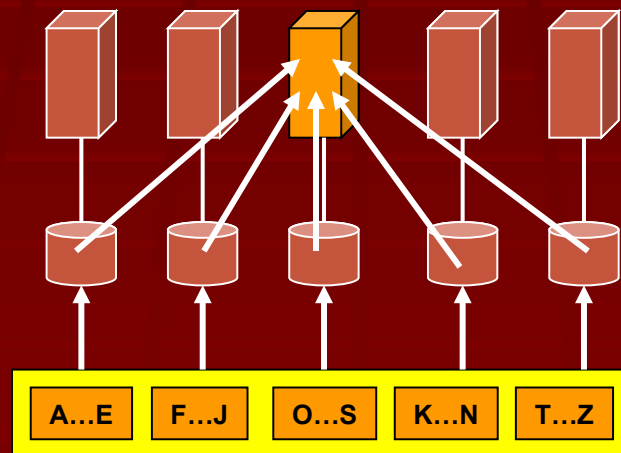
$$\text{Average}(S) = \sum \text{Avg}(s_1) + \sum \text{Avg}(s_2) + \dots$$

For groups:

Distribute data using hashing.

Sub aggregate groups close to the source.

Pass each sub-aggregate to its group's site.

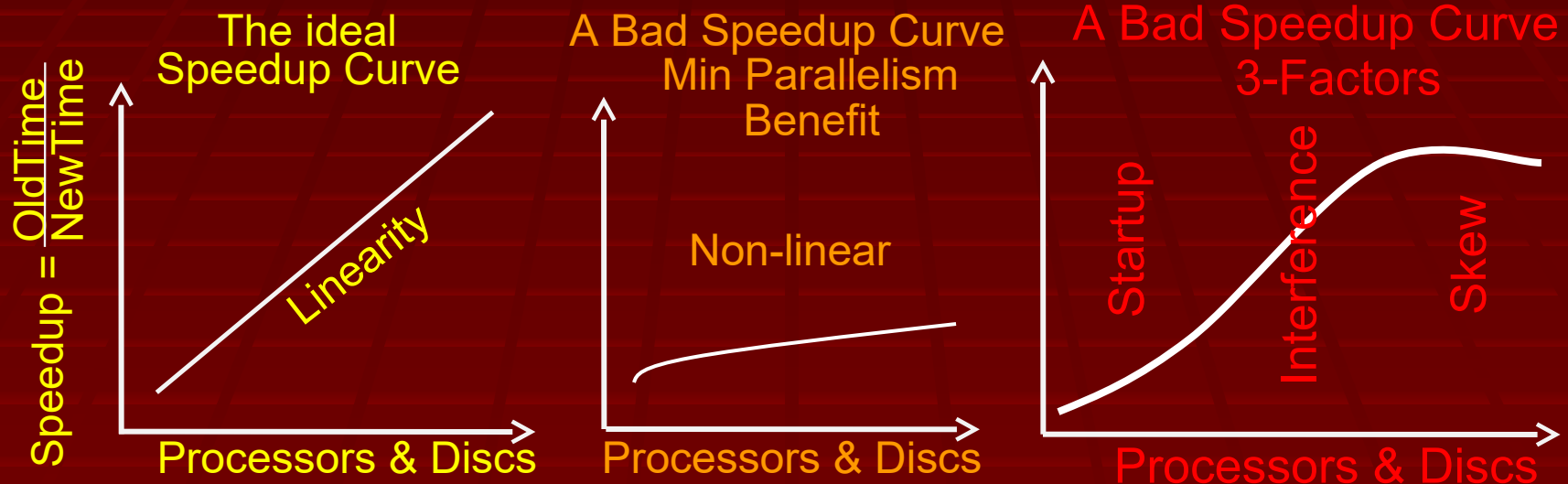


When to use which partitioning Tech?

- When to use Range Partitioning?
- When to Use Hash Partitioning?
- When to Use List Partitioning?
- When to use Round-Robin Partitioning?

Parallelism Goals and Metrics

- Speedup: The **Good**, The **Bad** & The **Ugly**



- Scale-up:

- Transactional Scale-up: Fit for OLTP systems
- Batch Scale-up: Fit for Data Warehouse and OLAP

Virtual University of Pakistan

Data Warehousing

Lecture-26

Need for Speed: Conventional Indexing Techniques

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan1010@yahoo.com

Need For Indexing: Speed

Consider searching your hard disk using the Windows SEARCH command.

- Search goes into directory hierarchies.
- Takes about a minute, and there are only a few thousand files.

Assume a fast processor and (even more importantly) a fast hard disk.

No text goes to graphics

- Assume file size to be 5 KB.
- Assume hard disk scan rate of a million files per second.
- Resulting in scan rate of 5 GB per second.

Largest search engine indexes more than 8 billion pages

- At above scan rate 1,600 seconds required to scan ALL pages.
- This is just for one user!
- No one is going to wait for 26 minutes, not even 26 seconds.

Hence, a sequential scan is simply not feasible.

Need For Indexing: Query Complexity

- How many customers do I have in Karachi?
- How many customers in Karachi made calls during April?
- How many customers in Karachi made calls to Multan during April?
- How many customers in Karachi made calls to Multan during April using a particular calling package?

Need For Indexing: I/O Bottleneck

- Throwing hardware just speeds up the CPU intensive tasks.
- The problem is of I/O, which does not scales up easily.
No text goes to graphics
- Putting the entire table in RAM is very very expensive.
- Therefore, index!

Indexing Concept

- Purely physical concept, nothing to do with logical model.
- Invisible to the end user (programmer), optimizer chooses it, effects only the speed, not the answer.
- With the library analogy, the time complexity to find a book? The average time taken
- Using a card catalog organized in many different ways i.e. author, topic, title etc and is sorted.
- A little bit of extra time to first check the catalog, but it “gives” a pointer to the shelf and the row where book is located.
- The catalog has no data about the book, just an efficient way of searching.

No text goes to graphics

Indexing Goal

Look at as few blocks as possible to find the matching record(s)

Conventional indexing Techniques

- Dense
- Sparse
- Multi-level (or B-Tree)
- Primary Index vs. Secondary Indexes

Dense Index: Concept

Dense Index

Data File

Every key in the data file is represented in the index file



Dense Index: Adv & Dis Adv

- Advantage:

- A dense index, if fits in the memory, is very efficient in locating a record given a key

- Disadvantage:

No text goes to graphics

- A dense index, if too big and doesn't fit into the memory, will be expensive when used to find a record given its key

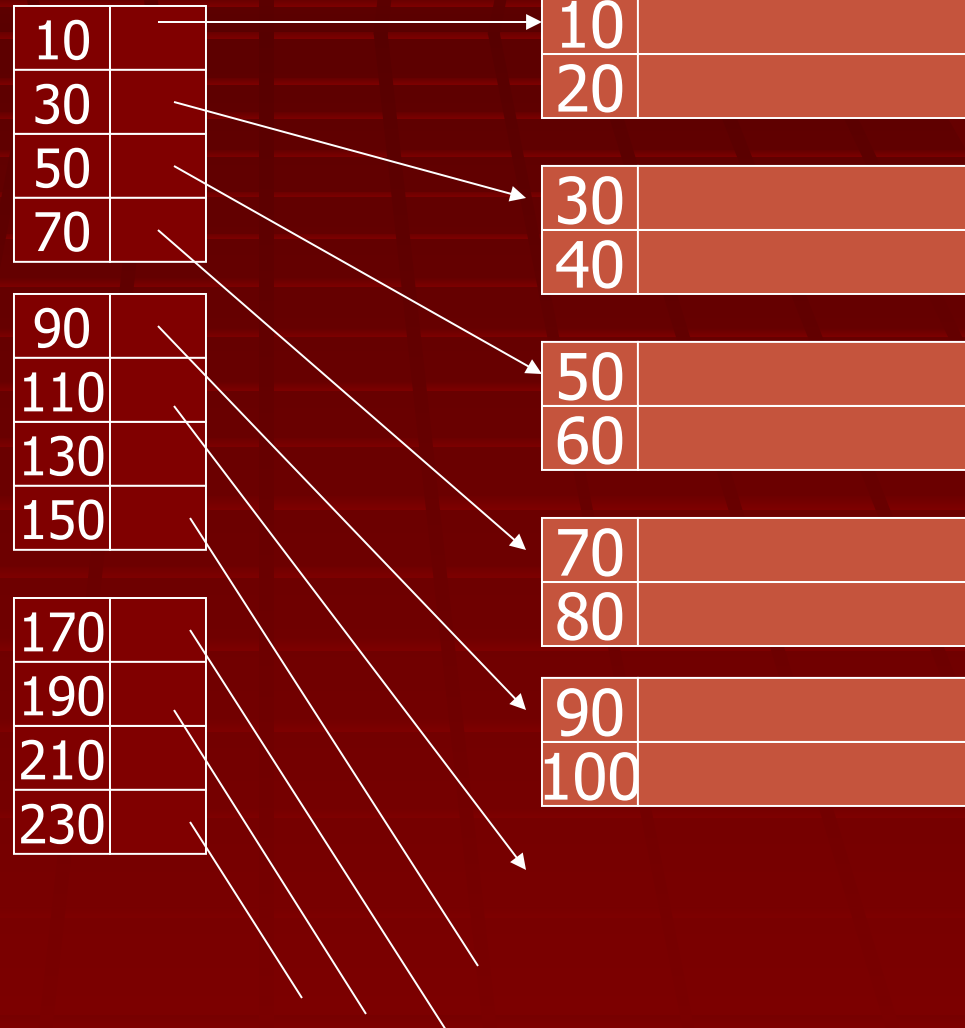
Sparse Index: Concept

Sparse Index

Data File

Normally keeps only one key per data block

Some keys in the data file will not have an entry in the index file



Sparse Index: Adv & Dis Adv

- Advantage:

No text goes to graphics

- A sparse index uses less space at the expense of somewhat more time to find a record given its key
- Support multi-level indexing structure

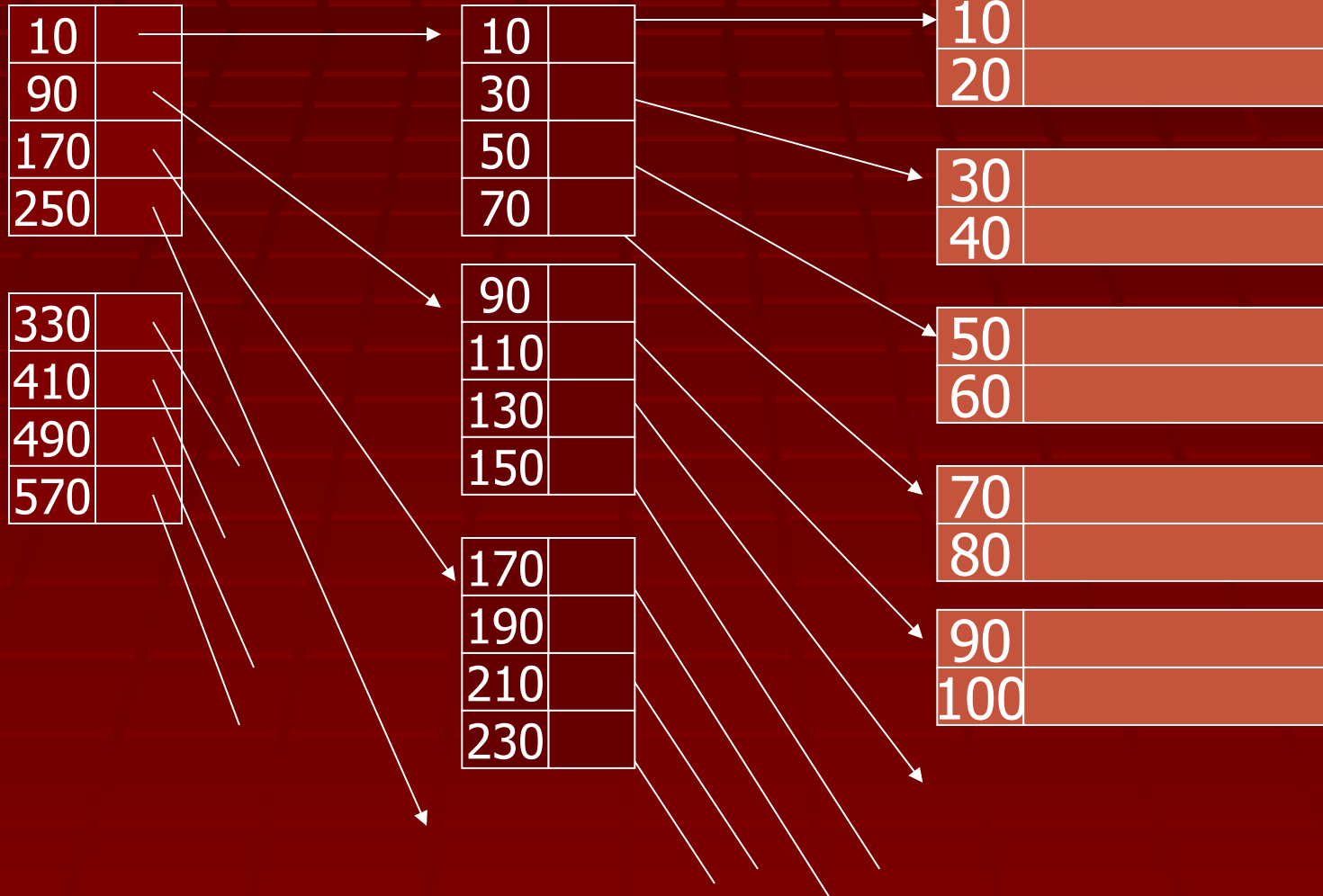
- Disadvantage:

- Locating a record given a key has different performance for different key values

Sparse Index: Multi level

Sparse 2nd level

Data File

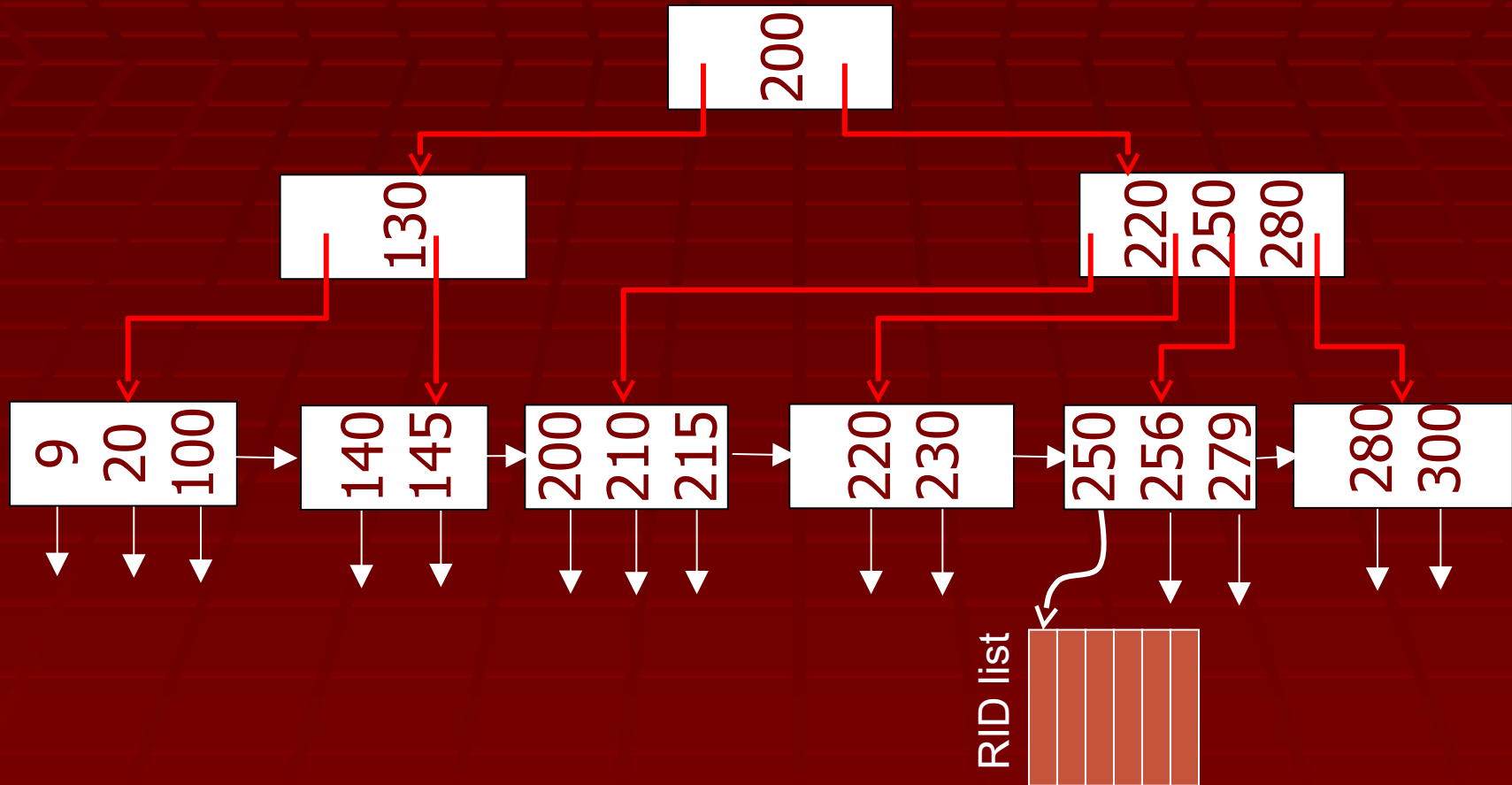


B-tree Indexing: Concept

- Can be seen as a general form of multi-level indexes.
- Generalize usual (binary) search trees (BST).
- Allow efficient and fast exploration at the expense of using slightly more space.
- Popular variant: B⁺-tree
- Support more efficiently queries like:
`SELECT * FROM R WHERE a = 11`
- `SELECT * FROM R WHERE 0 ≤ b and b < 42`

B-tree Indexing: Example

Looking for Empno 250



Each node stored in one disk block

B-tree Indexing: Limitations

- If a table is large and there are fewer unique values.
 - Capitalization is not programmatically enforced (meaning case-sensitivity does matter and “FLASHMAN” is different from “Flashman”).
 - Outcome varies with inter-character spaces.
 - A noun spelled differently will result in different results.
- Nothing will go to graphics
- **Insertion can be very expensive.**

B-tree Indexing: Limitations Example

Given that MOHAMMED is the most common first name in Pakistan, a 5-million row Customers table would produce many screens of matching rows for MOHAMMED AHMAD, yet would skip potential matching values such as the following:

VALUE MISSED	REASON MISSED
Mohammed Ahmad	Case sensitive
MOHAMMED AHMED	AHMED versus AHMAD
MOHAMMED AHMAD	Extra space between names
MOHAMMED AHMAD DR	DR after AHMAD
MOHAMMAD AHMAD	Alternative spelling of MOHAMMAD

Hash Based Indexing

- You may recall that in internal memory, hashing can be used to quickly locate a specific key.
- The same technique can be used on external memory.
- However, advantage over search trees is smaller in external search than internal. **WHY?**
- Because part of search tree can be brought into the main memory.

Hash Based Indexing: Concept

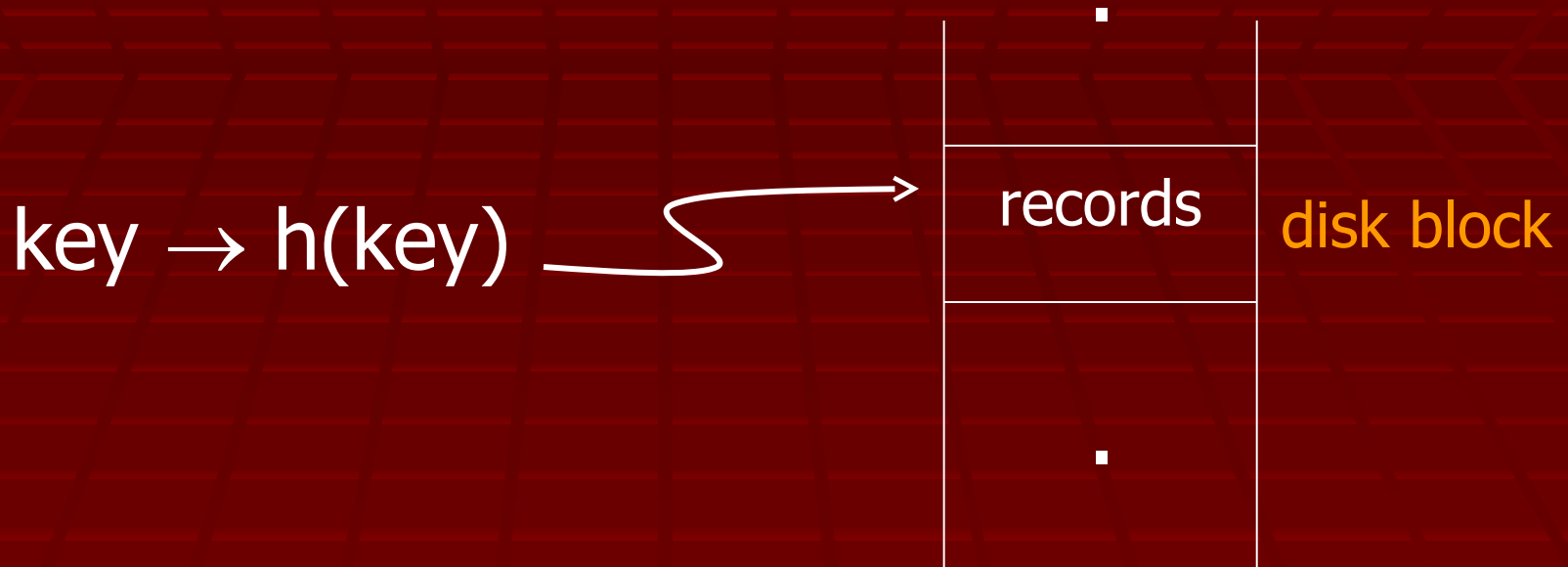
In contrast to B-tree indexing, hash based indexes do not (typically) keep index values in sorted order.

- Index entry is found by hashing on index value requiring exact match.

```
SELECT * FROM Customers WHERE AccttNo= 110240
```

- Index entries kept in hash organized tables rather than B-tree structures.
- Index entry contains ROWID values for each row corresponding to the index value.
- Remember few numbers in real-life to be useful for hashing.

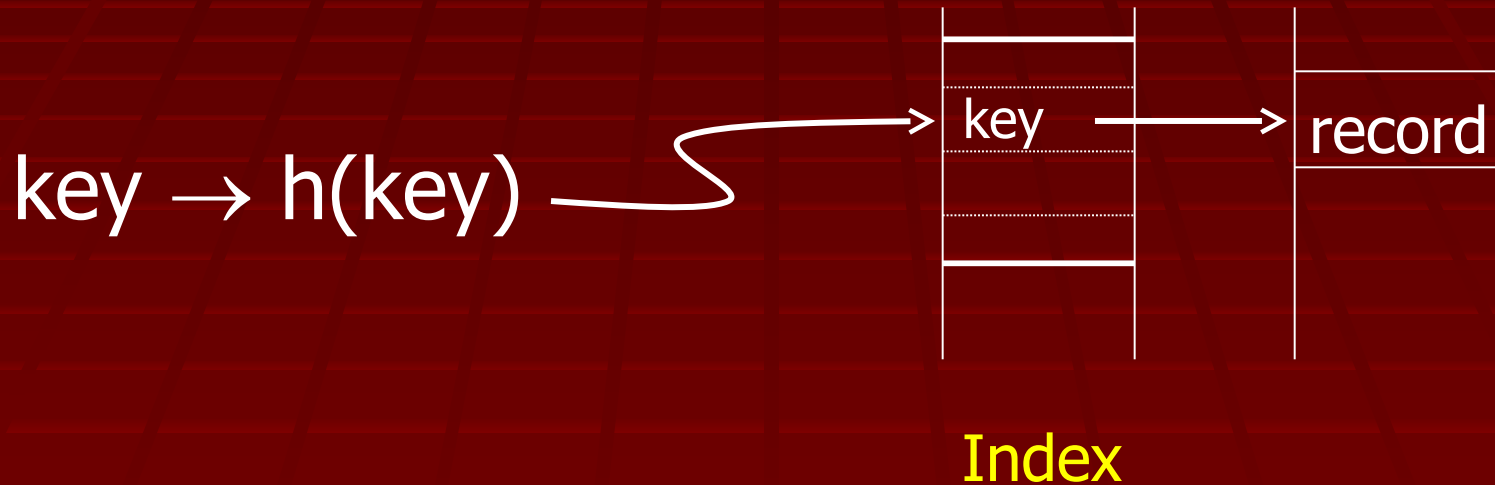
Hashing as Primary Index



Note on terminology:

The word "indexing" is often used synonymously with "B-tree indexing".

Hashing as Secondary Index



Can always be transformed to a secondary index using indirection, as above.

Indexing the Index

B-tree vs. Hash Indexes

- Indexing (using B-trees) good for range searches, e.g.:

SELECT * FROM R WHERE A > 5

- Hashing good for match based searches, e.g.:

SELECT * FROM R WHERE A = 5

Primary Key vs. Primary Index

Relation Students

<u>Name</u>	<u>ID</u>	<u>dept</u>
AHMAD	123	CS
Akram	567	EE
Numan	999	CS

- Primary Key & Primary Index:
 - PK is ALWAYS unique.
 - PI can be unique, but does not have to be.
- In DSS environment, very few queries are PK based.

Primary Indexing: Criterion

- Primary index selection criteria:
 - Common join and retrieval key.
 - Can be unique UPI or non-unique NUPI.
- Limits on NUPI.
 - Only one primary index per table (for hash-based file system).

Primary Indexing Criteria: Example

Call Table

call_id	decimal (15,0) NOT NULL
caller_no	decimal (10,0) NOT NULL
call_duration	decimal (15,2) NOT NULL
call_dt	date NOT NULL
called_no	decimal (15,0) NOT NULL

What should be the primary index of the call table for a large telecom company?

No simple answer!!

Primary Indexing

- Almost all joins and retrievals will occur through the caller_no foreign key.
 - Use caller_no as a NUPI.
- In case of non uniform distribution on caller_no or
- if phone number have very large number of outgoing calls (e.g., an institutional number could easily have several thousand calls).
 - Use call_id as UPI for good data distribution.

Primary Indexing

For a hash-based file system, primary index is free!

- No storage cost.
- No index build required.

OLTP databases use a page-based file system and therefore do not deliver this performance advantage.

Virtual University of Pakistan

Data Warehousing

Lecture-27

Need for Speed: Special Indexing Techniques

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan1010@yahoo.com

Special Index Structures

- Inverted index
- Bit map index
- Cluster index
- Join indexes

Sample table

Student	Name	Age	Campus	Tech
s1	amir	20	Lahore	Elect
s2	javed	20	Islamabad	CS
s3	salim	21	Lahore	CS
s4	imran	20	Peshawar	Elect
s5	majid	20	Karachi	Telecom
s6	taslim	25	Karachi	CS
s7	tahir	21	Peshawar	Telecom
s8	sohaib	26	Peshawar	CS
s9	afridi	19	Lahore	CS

Inverted index: Concept

Inverted Index: Example-1

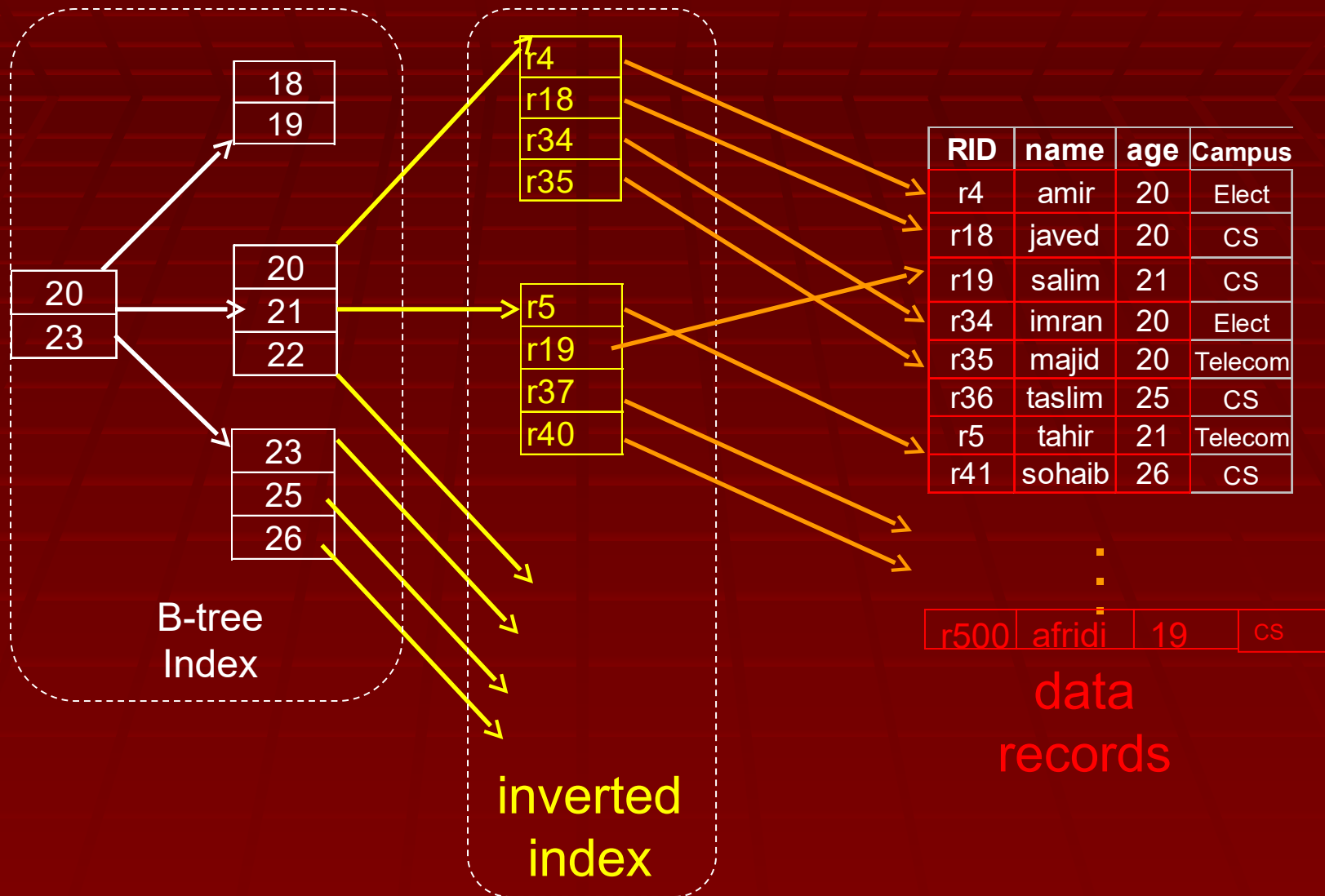
D1: M. Asalm BS Computer Science Lahore Campus

D2: Sana Aslam of Lahore MS Computer Engineering with GPA 3.4 Karachi Campus

Inverted index for the documents D1 and D2 is as follows:

3.4	→ [D2]	Karachi	→ [D2]
Asalm	→ [D1, D2]	Lahore	→ [D1, D2]
BS	→ [D1]	M.	→ [D1]
Campus	→ [D1, D2]	MS	→ [D2]
Computer	→ [D1, D2]	of	→ [D2]
Engineering	→ [D2]	Sana	→ [D2]
GPA	→ [D2]	Science	→ [D1]
		with	→ [D2]

Inverted Index: Example-2



Inverted Index: Query

- Query:
 - Get students with age = 20 and tech = “telecom”
- List for age = 20: r4, r18, r34, r35
- List for tech = “telecom”: r5, r35
- Answer is intersection: r35

Bitmap Indexes: Concept

Bitmap Indexes: Example

- The index consists of bitmaps, with a column for each unique value:

Index on City (larger table):

SID	Islamabad	Lahore	Karachi	Peshawar
1	0	1	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	0	1
5	0	0	1	0
6	0	0	1	0
7	0	0	0	1
8	0	0	0	1
9	0	1	0	0

Index on Tech (smaller table):

SID	CS	Elect	Telecom
1	1	0	0
2	0	1	0
3	0	1	0
4	1	0	0
5	0	0	1
6	0	1	0
7	0	0	1
8	1	0	0
9	1	0	0

Bitmap Index: Query

- Query:
 - Get students with age = 20 and campus = “Lahore”
- List for age = 20: 1101100000
- List for campus = “Lahore”: 1010000001
- Answer is AND : 1000000000
- Good if domain cardinality is small
- Bit vectors can be compressed
 - Run length encoding

Bitmap Index: Compression

Basic Concept

Case-1 1111000011110000001111100000011111 **INPUT**
14#04#14#06#15#06#15 **OUTPUT**

Case-2 1010101010101010101010101010101010101010 **INPUT**
11#01#11#01#11#01#11#01#... **OUTPUT**

Case-3 111111111111111111100000000000000000 **INPUT**
117#017 **OUTPUT**

Bitmap Index: More Queries

- “Which students from Lahore are enrolled in ‘CS’?”
- “How many students are enrolled in ‘CS’?”

Bitmap Index: Adv.

Bitmap Index: Performance Points

Bitmap Index: Dis. Adv.

Cluster Index: Concept

Cluster Index: Example

Student	Name	Age	Campus	Tech
s9	afridi	19	Lahore	CS
s1	amir	20	Lahore	Elect
s2	javed	20	Islamabad	CS
s4	imran	20	Peshawar	Elect
s5	majid	20	Karachi	Telecom
s3	salim	21	Lahore	CS
s7	tahir	21	Peshawar	Telecom
s6	taslim	25	Karachi	CS
s8	sohaib	26	Peshawar	CS

Cluster indexing on AGE

One indexing column at a time

Student	Name	Age	Campus	Tech
s9	afridi	19	Lahore	CS
s2	javed	20	Islamabad	CS
s3	salim	21	Lahore	CS
s6	taslim	25	Karachi	CS
s8	sohaib	26	Peshawar	CS
s1	amir	20	Lahore	Elect
s4	imran	20	Peshawar	Elect
s5	majid	20	Karachi	Telecom
s7	tahir	21	Peshawar	Telecom

Cluster indexing on TECH

Cluster Index: Issues

Join Index: Example

The rows of the table consist entirely of such references, which are the RIDs of the relevant rows.

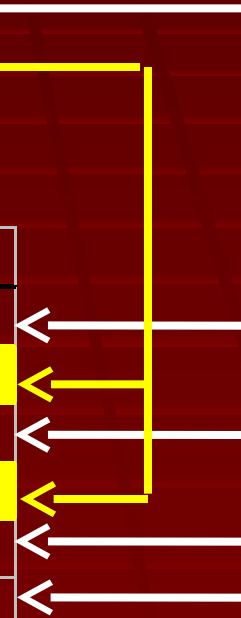
PROGRAM

id	name	NoS	jIndex
p1	BS	10	r1,r3,r5,r6
p2	MS	5	r2,r4

join index

CAMPUS

rld	progid	CID	date	NoS
r1	p1	c1	1	12
r2	p2	c1	1	11
r3	p1	c3	1	50
r4	p2	c2	1	8
r5	p1	c1	2	44
r6	p1	c2	2	4



Materialized views

Virtual University of Pakistan

Data Warehousing

Lecture-28

Need for Speed: Join Techniques

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

Need for Speed: Join Techniques

Background

About Nested-Loop Join

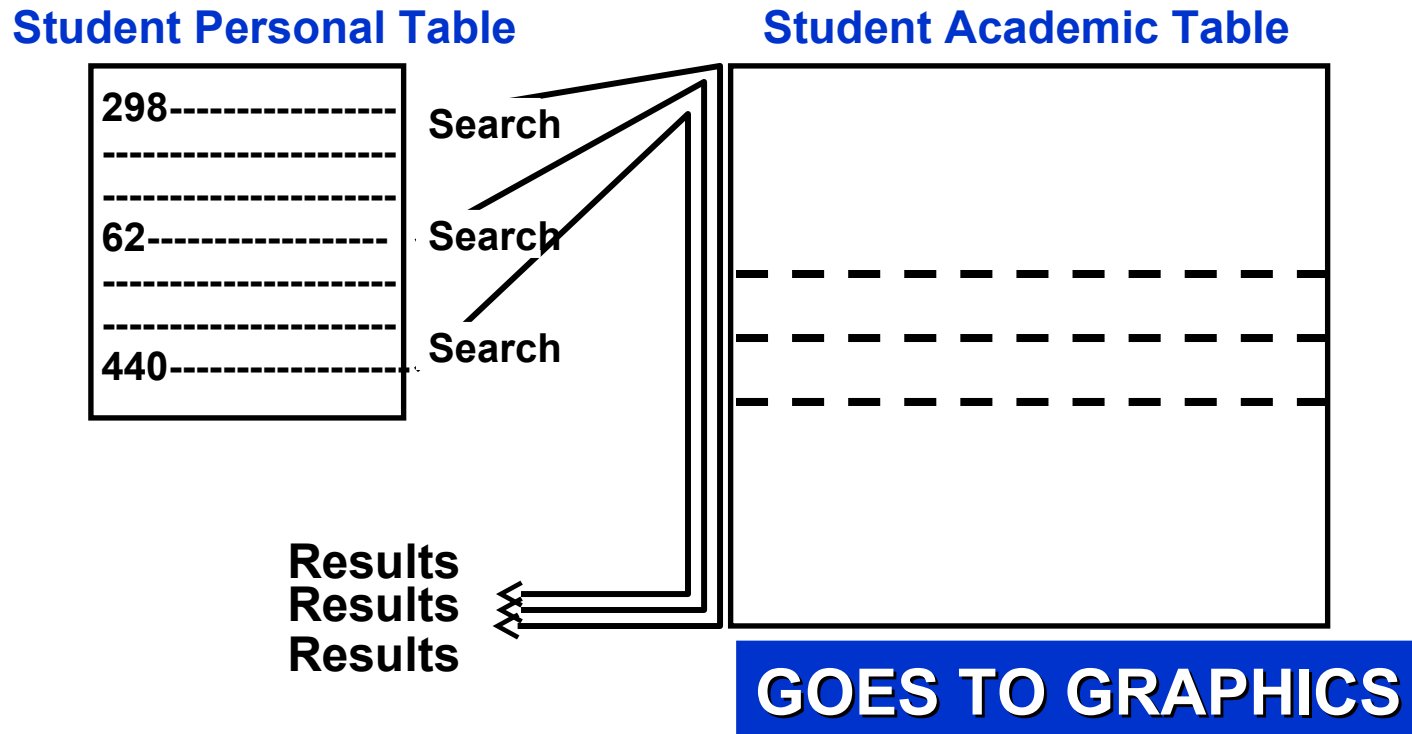
Nested-Loop Join: Code

```
FOR i = 1 to N DO BEGIN    /*      N rows in T1  */
  IF ith row of T1 qualifies THEN BEGIN
    For j = 1 to M DO BEGIN    /*      M rows in T2  */
      IF the ith row of T1 matches to jth row of T2 on join key THEN BEGIN
        IF the jth row of T2 qualifies THEN BEGIN
          produce output row
        END
      END
    END
  END
END
```

GOES TO GRAPHICS

Nested-Loop Join: Working Example

“What is the average GPA of undergraduate male students?”



**For each qualifying row of Personal table,
Academic table is examined for matching rows.**

Nested-Loop Join: Order of Tables

Nested-Loop Join: Cost Formula

Join cost = Cost of accessing *Table_A* +
of qualifying rows in *Table_A* × Blocks of
Table_B to be scanned for each qualifying row

OR

Join cost = Blocks accessed for *Table_A* +
Blocks accessed for *Table_A* × Blocks
accessed for *Table_B*

GOES TO GRAPHICS

Nested-Loop Join: Cost of reorder

Table_A = 500 blocks and
Table_B = 700 blocks.

Qualifying blocks for *Table_A* $QB(A) = 50$

Qualifying blocks for *Table_B* $QB(B) = 100$

Join cost A&B = $500 + 50 \times 700 = 35,500$ I/Os

Join cost B&A = $700 + 100 \times 500 = 50,700$ I/Os

i.e. an increase in I/O of about 43%.

GOES TO GRAPHICS

Nested-Loop Join: Variants

Sort-Merge Join

Sort-Merge Join: Process

Table_A	Table_B
1	1
1	3
2	3
2	4
2	4
4	4
5	5
5	5
5	6
6	6
6	6
6	6
6	7
6	7
7	7
8	7

Table_A	Table_B
1	1
1	3
2	3
2	4
2	4
4	4
5	5
5	5
5	6
6	6
6	6
6	6
6	7
6	7
7	7
8	7

Table_A	Table_B
1	1
1	3
2	3
2	4
2	4
4	4
5	5
5	5
5	6
6	6
6	6
6	6
6	7
6	7
7	7
8	7

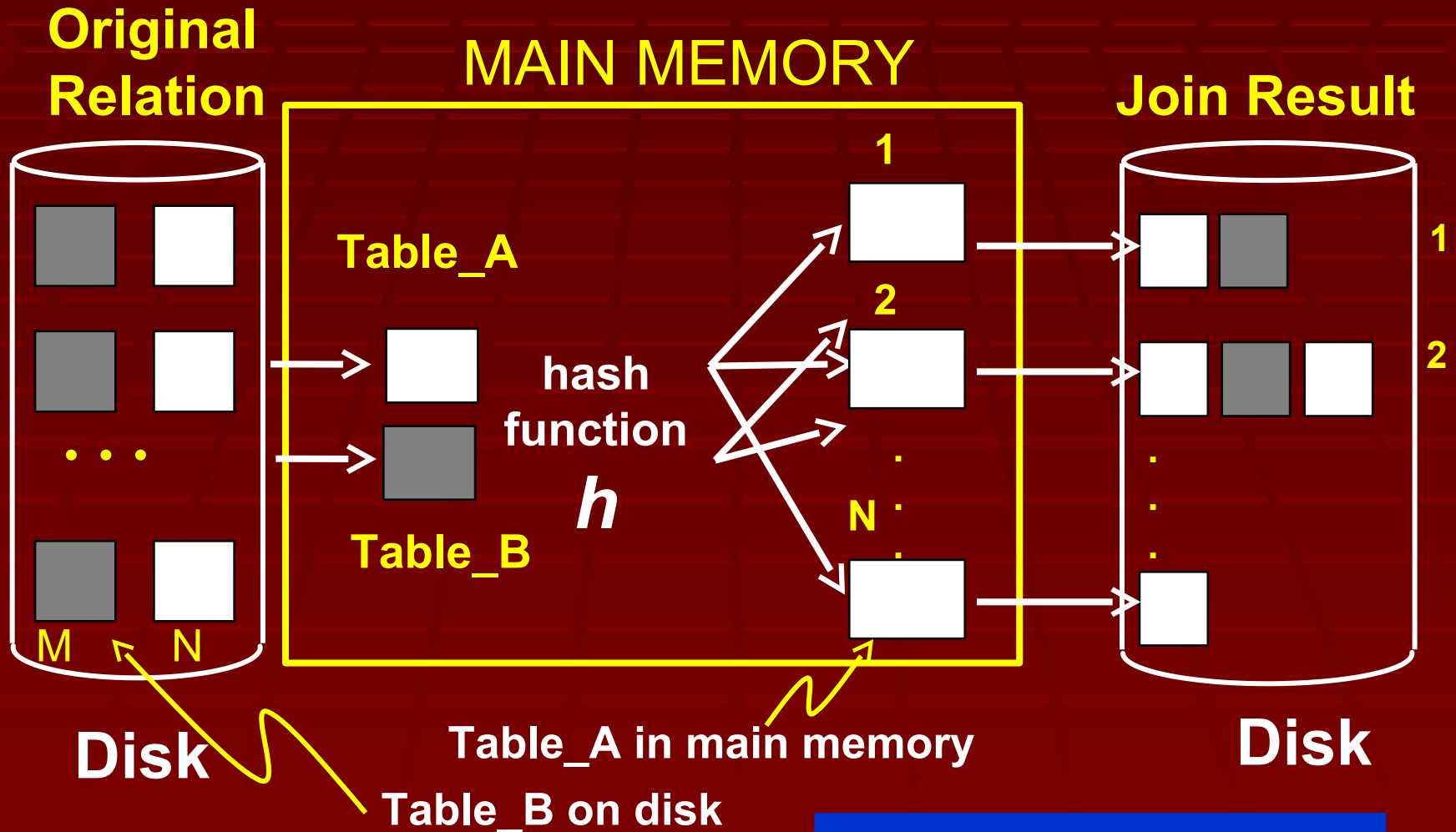
Sort-Merge Join Example

Sort-Merge Join: Note

Hash-Based join

Hash-Based Join: Working

Hash-Based Join: Example



GOES TO GRAPHICS

Hash-Based Join: Large “small” Table

Hash-Based Join: Partition Skew

Hash-Based Join: Intrinsic Skew

Virtual University of Pakistan

Data Warehousing

Lecture-29

Brief Intro. to Data Mining

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

What is Data Mining?: Non technical view

“There are things that we know that we know...

there are things that we know that we don't know...

there are things that we don't know we don't know.”

Donald Rumsfeld

US Secretary of Defence

What is Data Mining?: Slightly formal

What is Data Mining?: Formal view

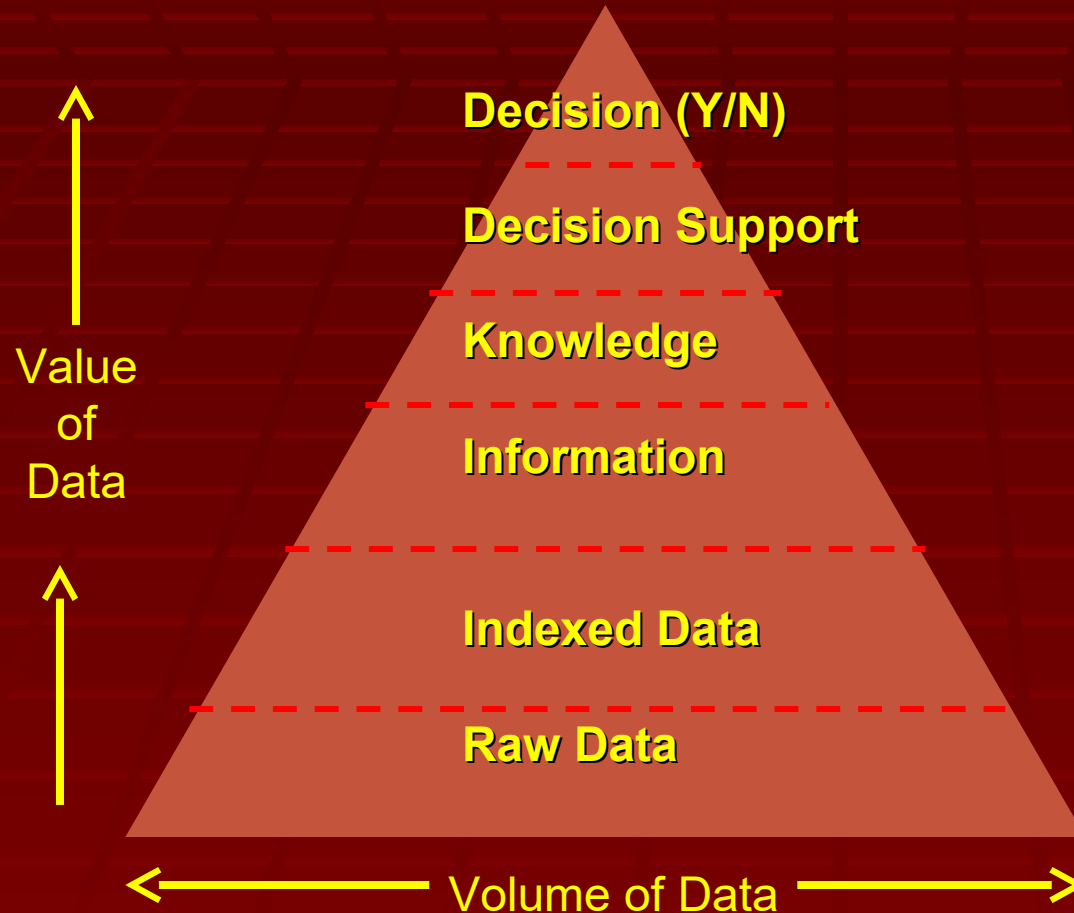
Data mining digs out valuable non-trivial information from large multidimensional apparently unrelated data bases(sets).

Why Data Mining? Huge volume

Claude Shannon's info. theory

**More volume means
less information**

Value vs. Volume



Why Data Mining?: Supply & Demand

Data Mining is **HOT!**

- **10 Hottest Jobs of year 2025**

Time Magazine, 22 May, 2000

- **10 emerging areas of technology**

*MIT's Magazine of Technology Review,
Jan/Feb, 2001*

How Data Mining is different? Traditionally

- **Knowledge Discovery (KDD)**

- **Data Mining (Knowledge-driven exploration)**

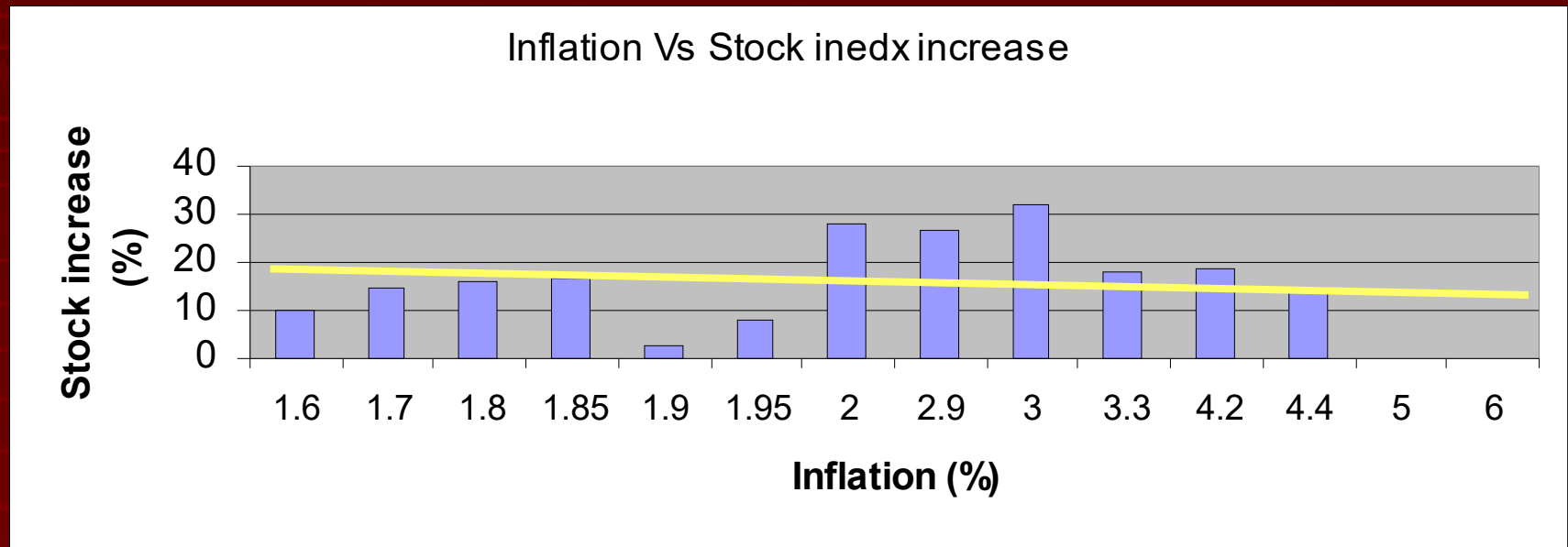
- **Data Warehouses (Data-driven exploration):**

- **Traditional Database (Transactions):**

Data Mining Vs Statistics

Data Mining Vs. Statistics

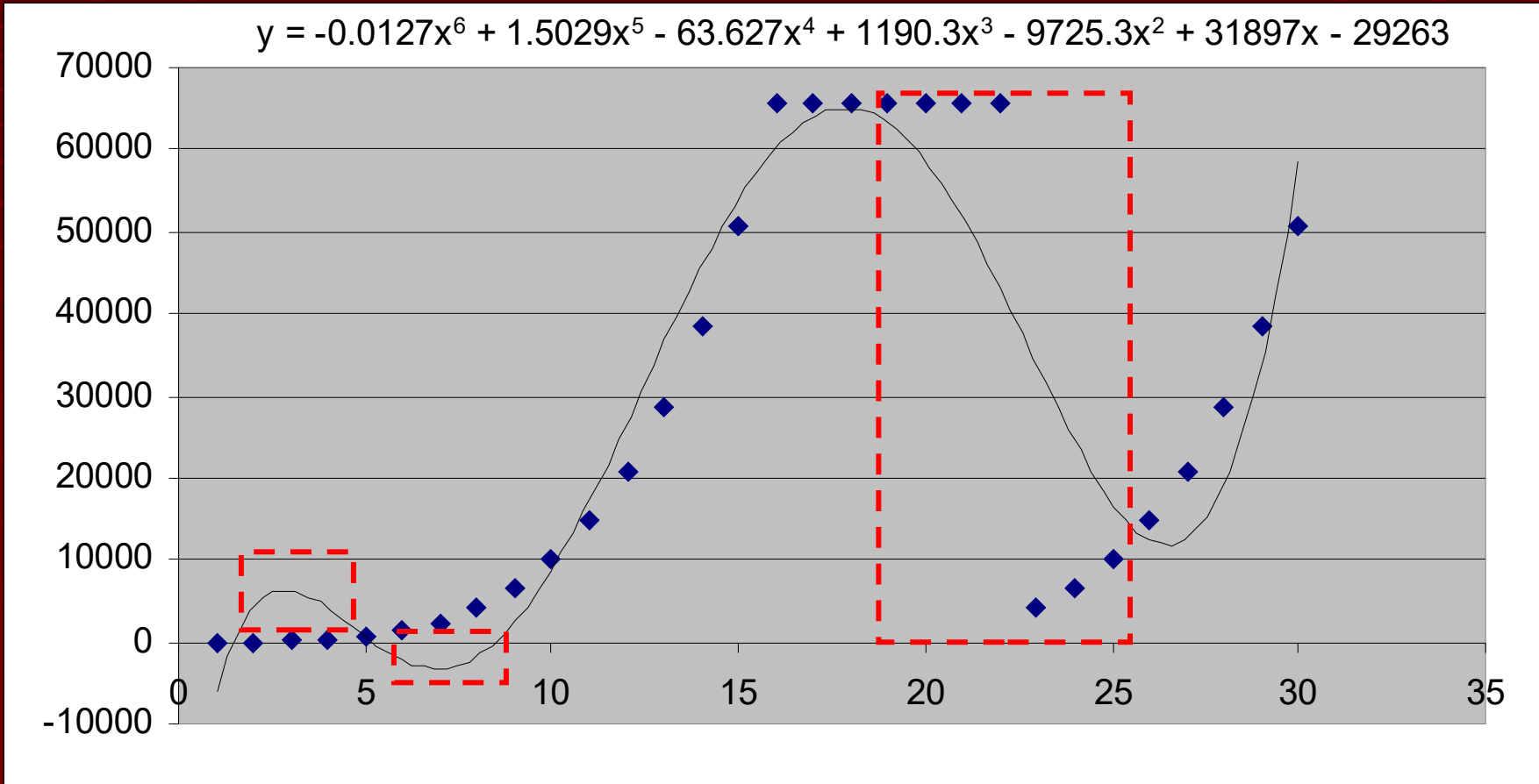
Knowledge extraction using statistics



Q: What will be the stock increase when inflation is 6%?

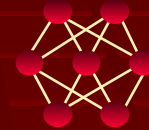
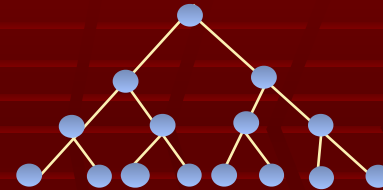
A: Model non-linear relationship using a line $y = mx + c$.
Hence answer is 13%

Failure of regression models

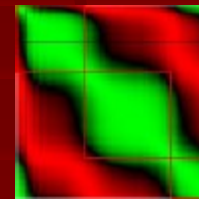


Data Mining is...

- Decision Trees
- Neural Networks
- Rule Induction
- Clustering
- Genetic Algorithms

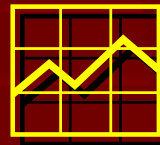


If....
Then...



Data Mining is NOT ...

- Data warehousing
- Ad Hoc Query / Reporting
- Online Analytical Processing (OLAP)
- Data Visualization
- Software Agents



Data Mining: **Business** Perspective

- “knowledge” is worth knowing if it can be used to increase profit by lowering cost or it can be used to increase profit by raising revenue.

Business questions

- **Profiling/Segmentation**
- **Cross-Service**
- **Employee retention:**

Virtual University of Pakistan

Data Warehousing

Lecture-30

What can Data Mining do?

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

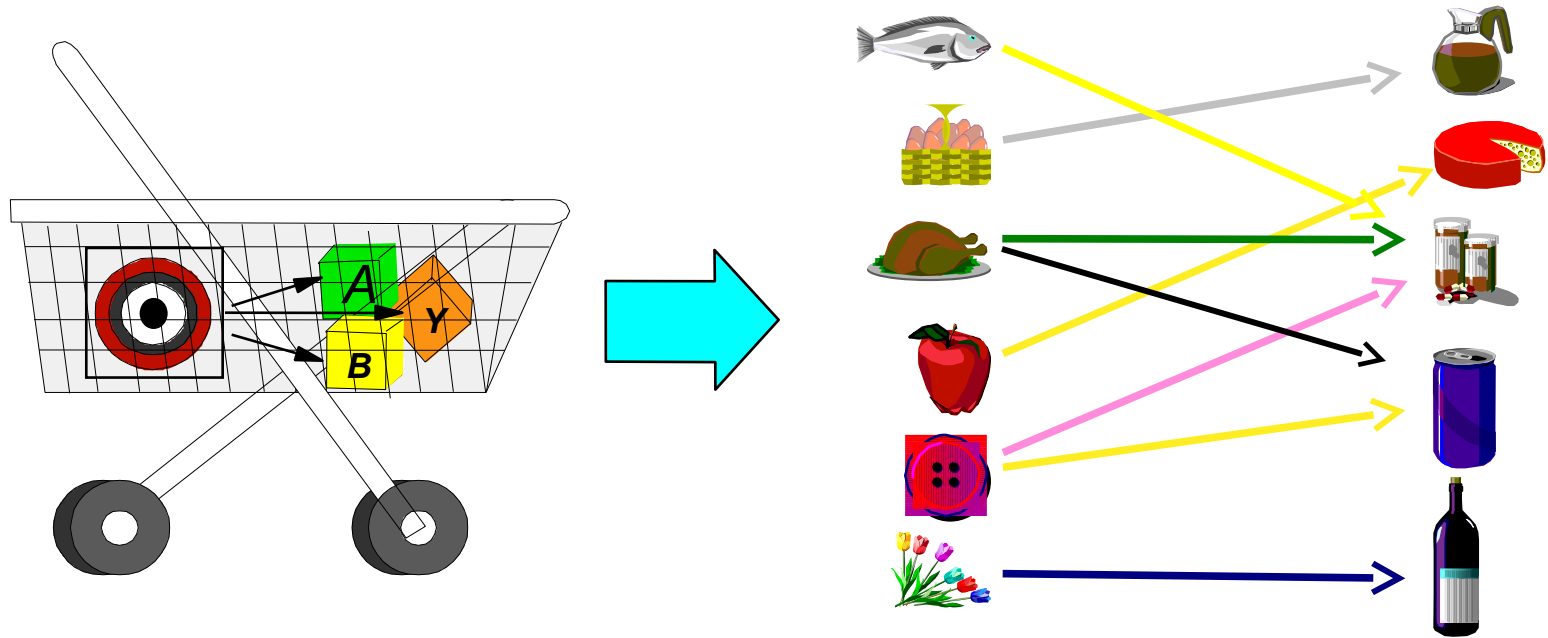
CLASSIFICATION

ESTIMATION

PREDICTION

MARKET BASKET ANALYSIS

MARKET BASKET ANALYSIS



*98% of people who purchased items **A** and **B**
also purchased item **C***

Ask graphics to replace pictures of items with similar pictures

Discovering Association Rules

TID	Items
1	Bread, Cola, Milk
2	Juice, Bread
3	Juice, Cola, Diaper, Milk
4	Juice, Bread, Diaper, Milk
5	Cola, Diaper, Milk

Rules:

$\{\text{Milk}\} \rightarrow \{\text{Cola}\}$

$\{\text{Diaper, Milk}\} \rightarrow \{\text{Juice}\}$

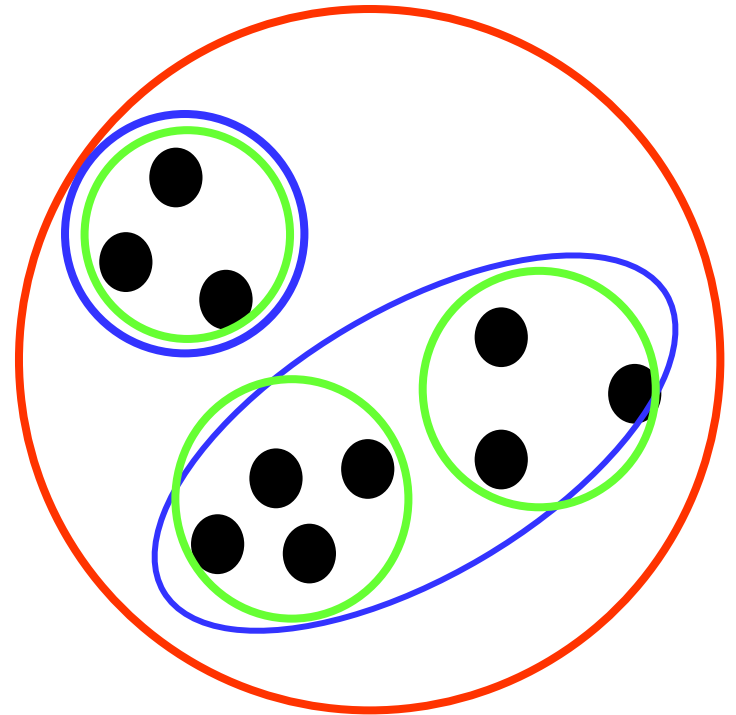
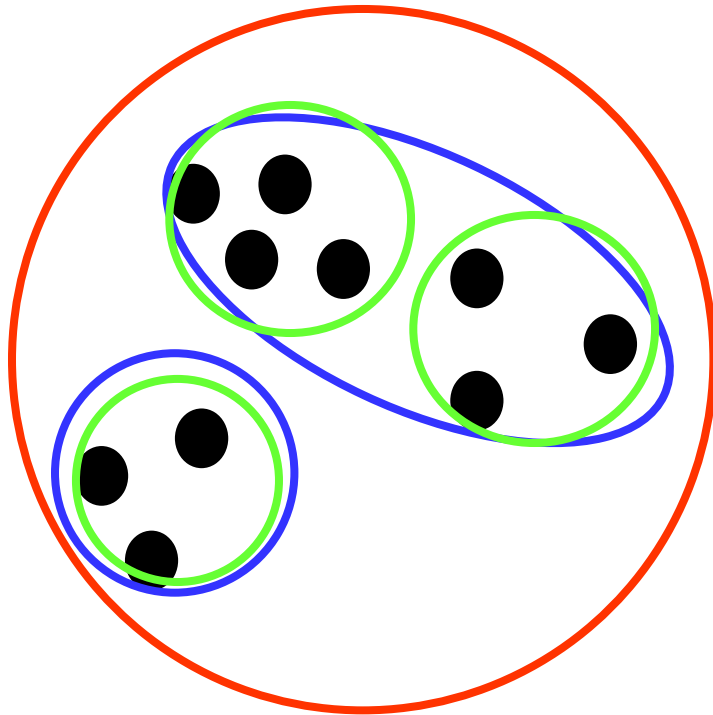
CLUSTERING

Task of segmenting a heterogeneous population into a number of more homogenous sub-groups or clusters.

Examples of Clustering Applications

- Marketing:
- Insurance:
- Land use:
- Seismic studies:

Ambiguity in Clustering



How many clusters?

Two clusters
Four clusters
Six clusters

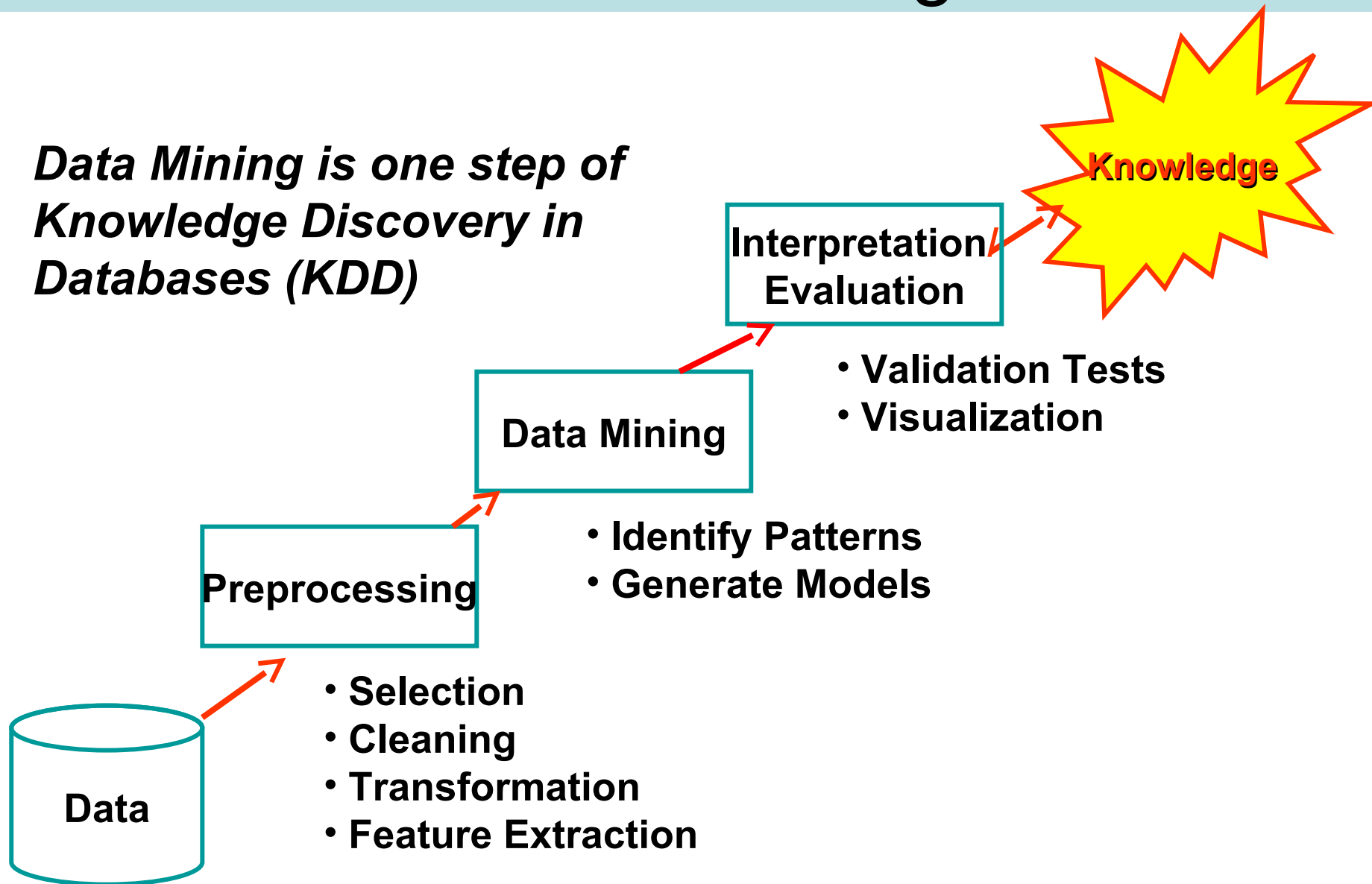
DESCRIPTION

Comparing Methods

- *Accuracy:*
- *Speed:*
- *Robustness:*
- *Scalability:*
- *Interpretability:*
- *Simplicity:*

Where does Data Mining fits in?

Data Mining is one step of Knowledge Discovery in Databases (KDD)



Virtual University of Pakistan

Data Warehousing

Lecture-31

Supervised vs. Unsupervised Learning

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

Data Structures in Data Mining

- Data matrix

- Table or database
- n records and m attributes,
- $n \gg m$

$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	...	$C_{1,m}$
$C_{2,1}$	$C_{2,2}$	$C_{2,3}$		$C_{2,m}$
$C_{3,1}$	$C_{3,2}$	$C_{3,3}$		$C_{3,m}$
...				...
$C_{n,1}$	$C_{n,2}$	$C_{n,3}$	...	$C_{n,m}$

- Similarity matrix

- Symmetric square matrix
- $n \times n$ or $m \times m$

1	$S_{1,2}$	$S_{1,3}$	...	$S_{1,n}$
$S_{2,1}$	1	$S_{2,3}$		$S_{2,n}$
$S_{3,1}$	$S_{3,2}$	1		$S_{3,n}$
...				...
$S_{n,1}$	$S_{n,2}$	$S_{n,3}$	...	1

Main types of DATA MINING

Supervised

- Bayesian Modeling
- Decision Trees
- Neural Networks
- Etc.

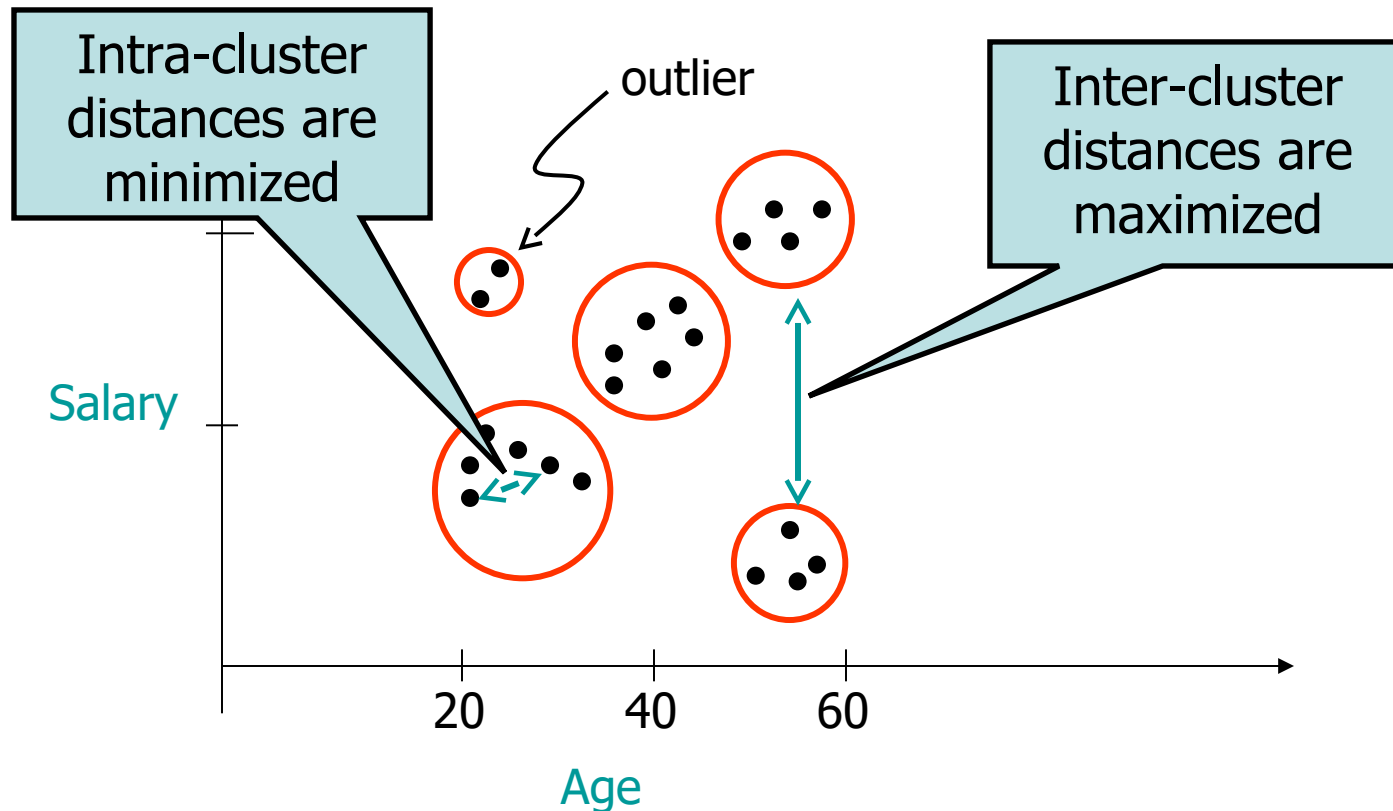
Type and number of classes are known in advance

Unsupervised

- One-way Clustering
- Two-way Clustering

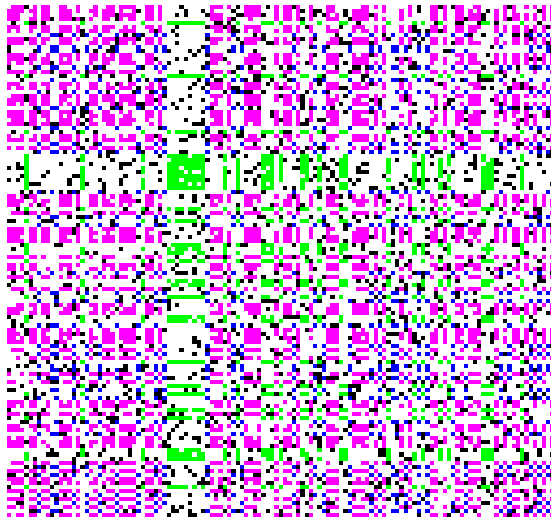
Type and number of classes are **NOT** known in advance

Clustering: Min-Max Distance

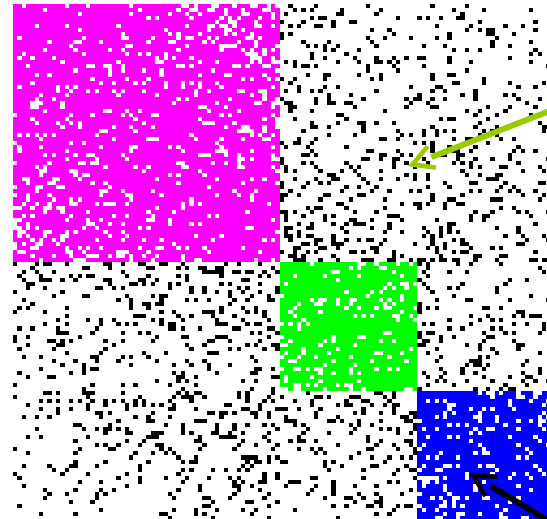


How Clustering works?

One-way clustering example



INPUT

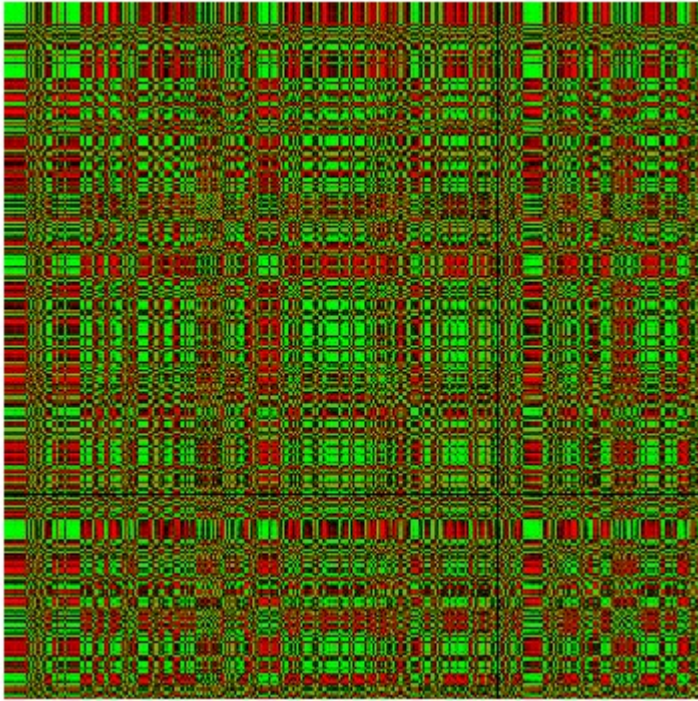


OUTPUT

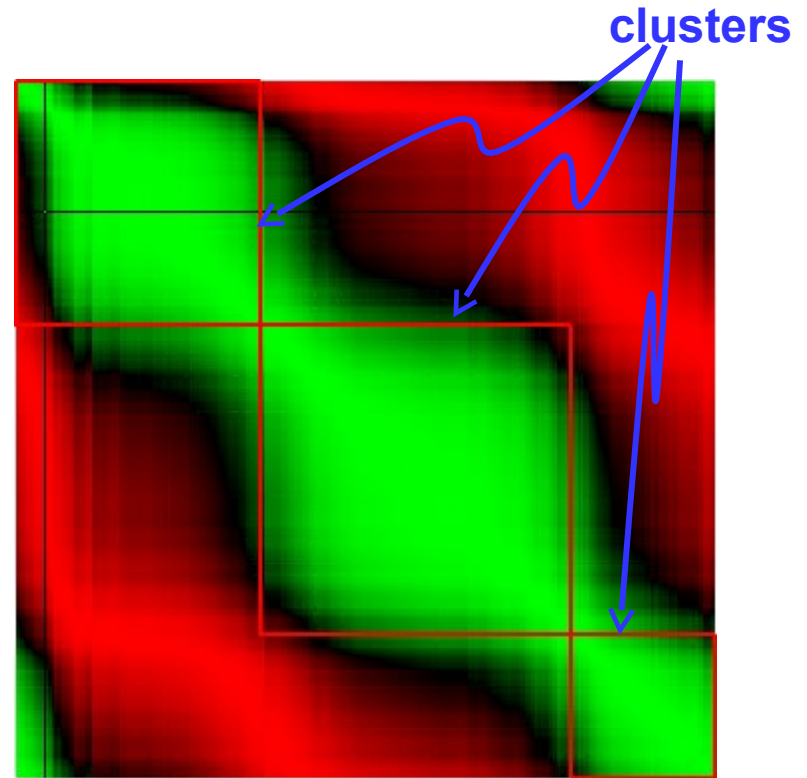
Black spots
are noise

White spots
are missing
data

Data Mining Agriculture data

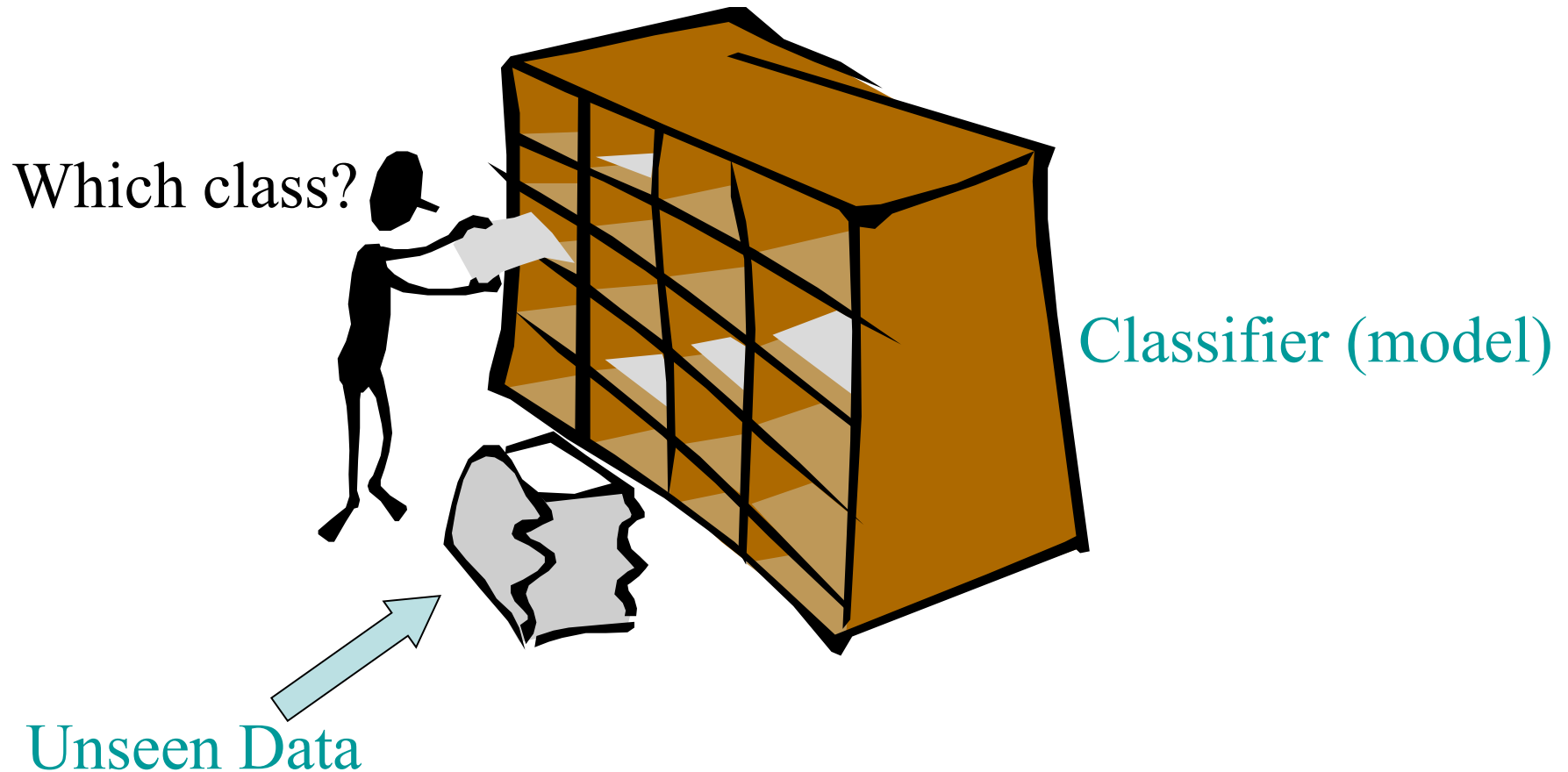


INPUT

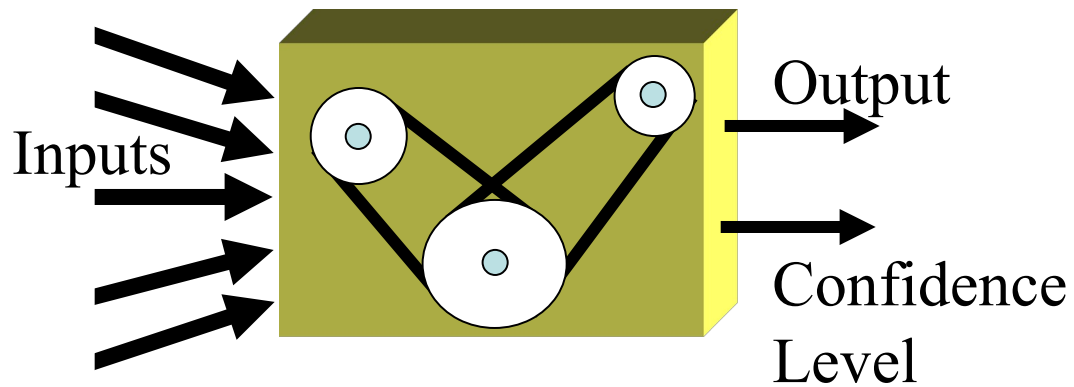


Clustered OUTPUT

Classification

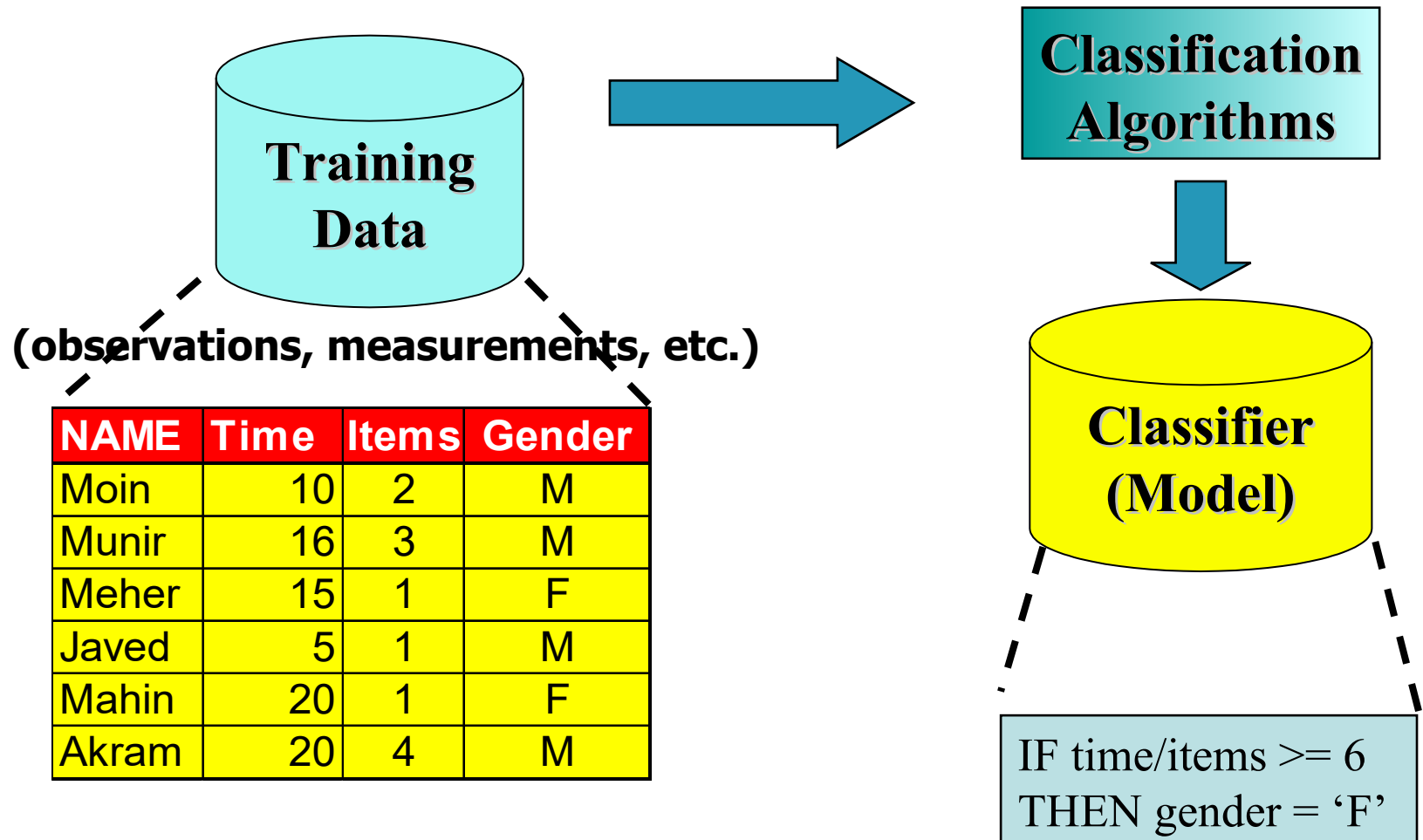


How Classification work?

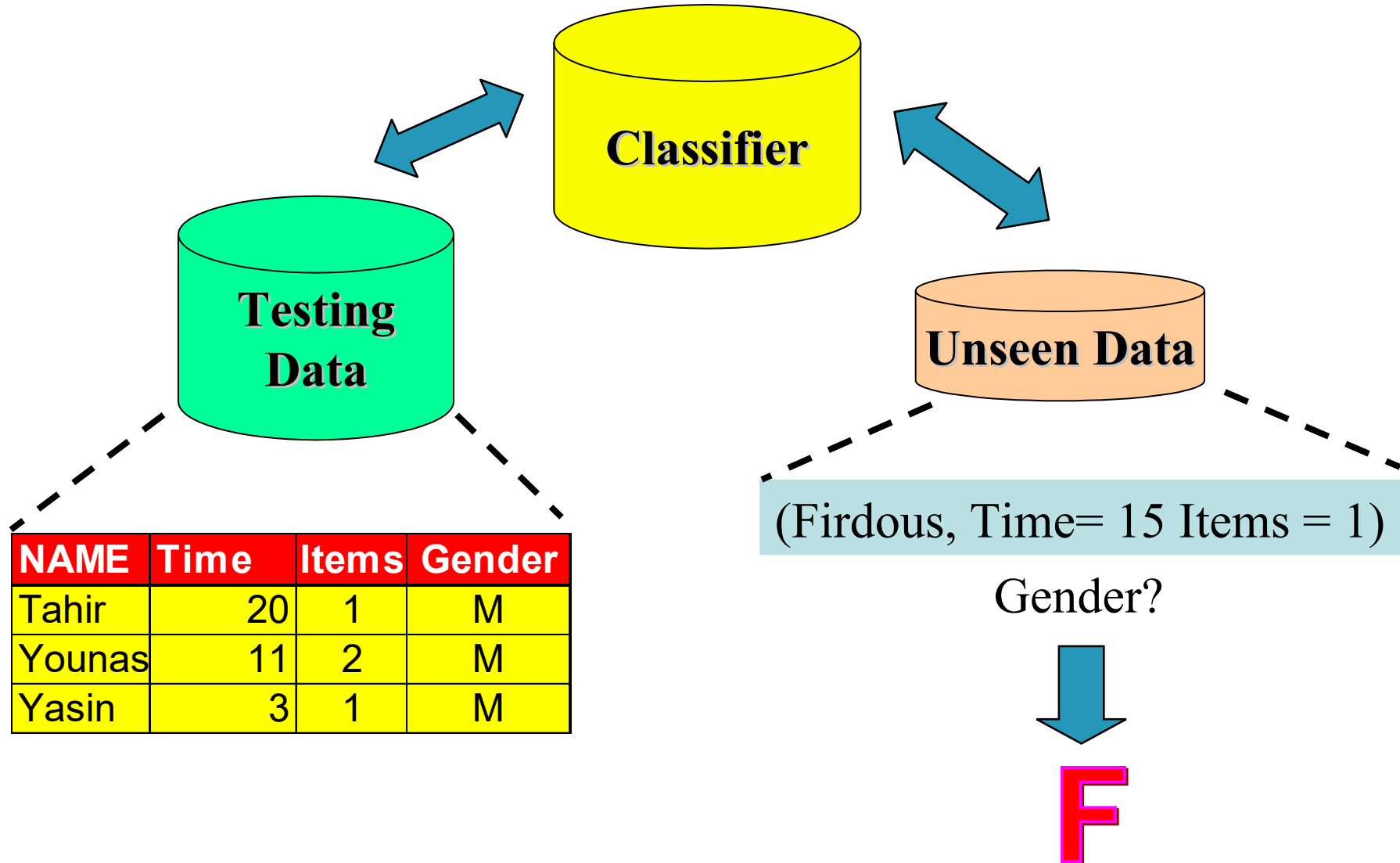


Classification Process (1): Model Construction

Relationship between shopping time and items bought

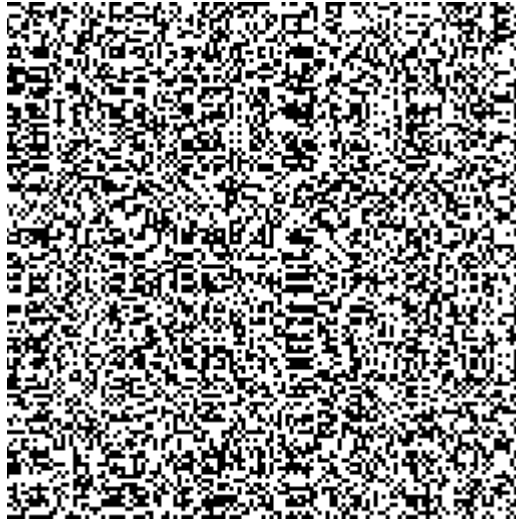


Classification Process (2): Use the Model in Prediction

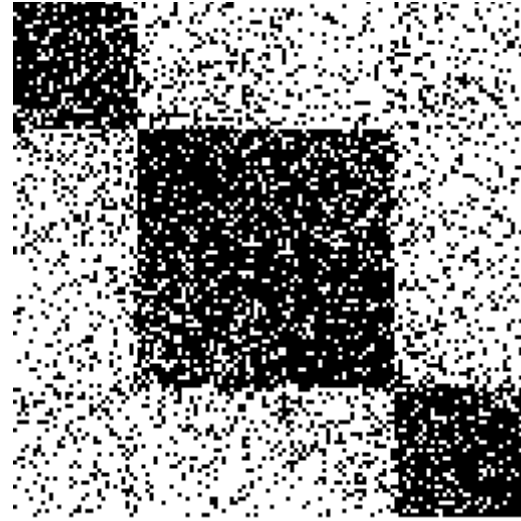


Clustering vs. Cluster Detection

Clustering vs. Cluster Detection Example



A

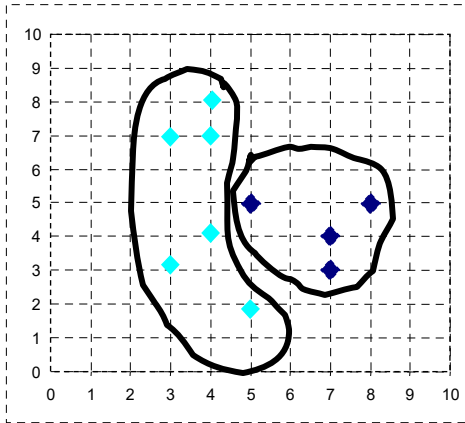


B

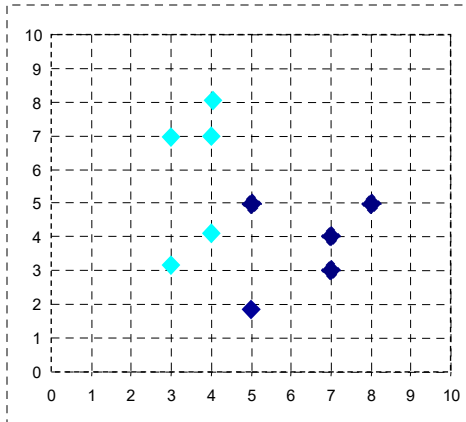
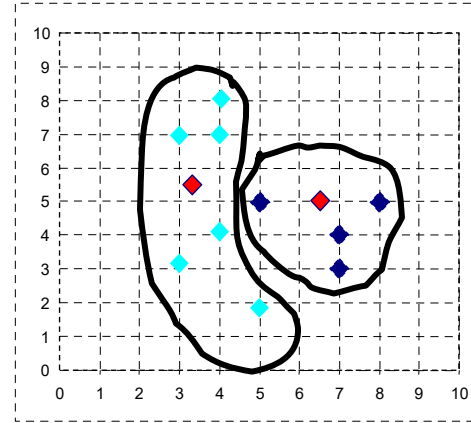
The *K-Means* Clustering

The *K-Means* Clustering: Example

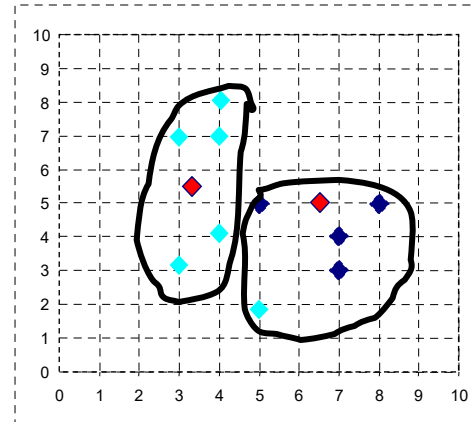
A



B



D



C

The *K-Means* Clustering: Comment

Virtual University of Pakistan

Data Warehousing

Lecture-32

DWH Lifecycle: Methodologies

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan101@yahoo.com

DWH Lifecycle: Methodologies

Data Warehouse vs. Typical IS

DWH & SW Engineering

Implementation strategies

Development methodologies

Waterfall Model

Spiral Model

Rapid Application Development (RAD)

5 Keys to successful RAD

Development Methodologies

Development Methodologies

Development Methodologies

Where do you start from?

What specific problems the DWH will solve?

Where do you start from?

What criteria use to measure success?

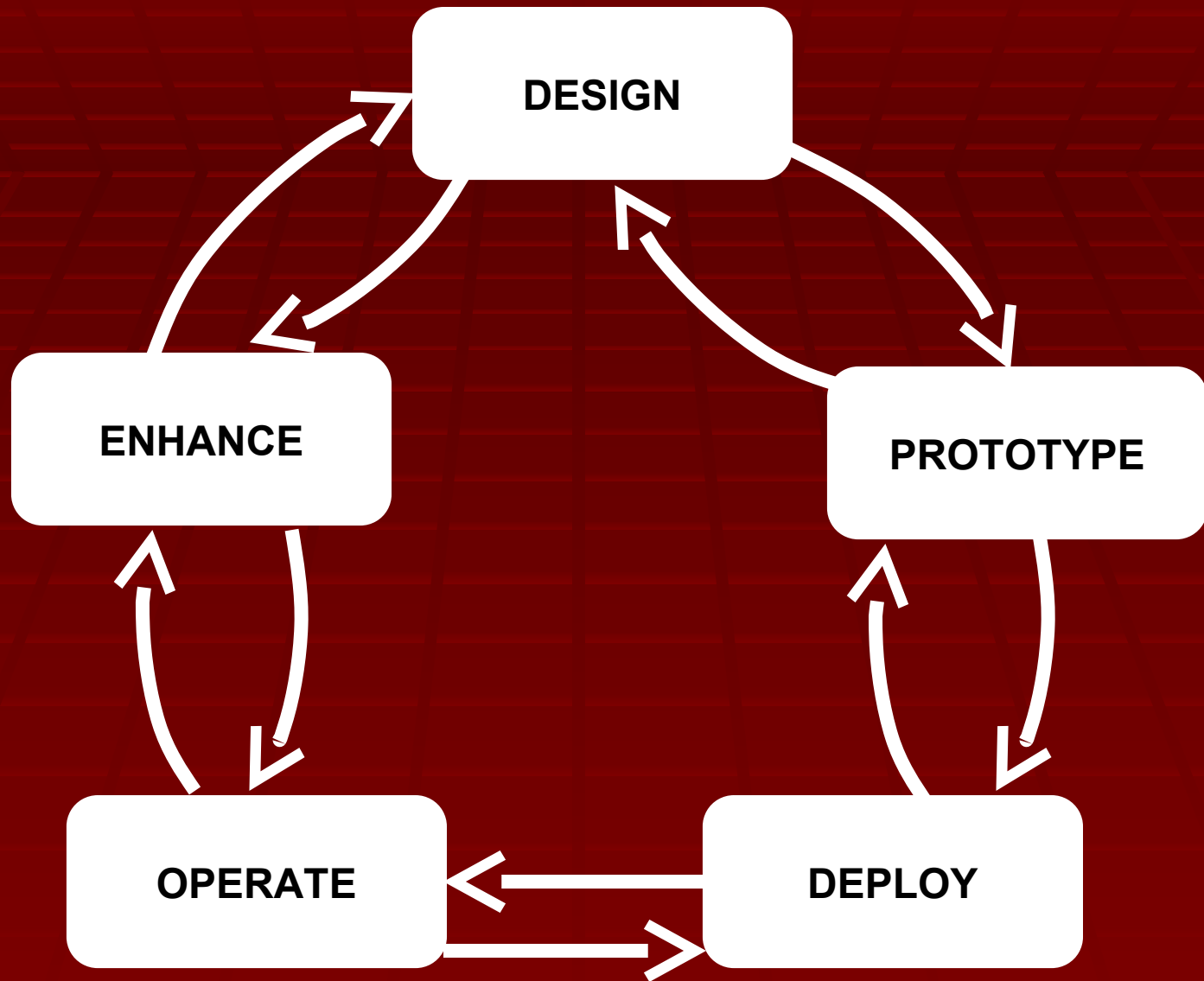
Where do you start from?

How to manage time and money?

Where do you start from?

What skills are required?

DWH Life Cycle Model



Virtual University of Pakistan

Data Warehousing

Lecture-33

DWH Implementation: Goal Driven Approach (1)

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

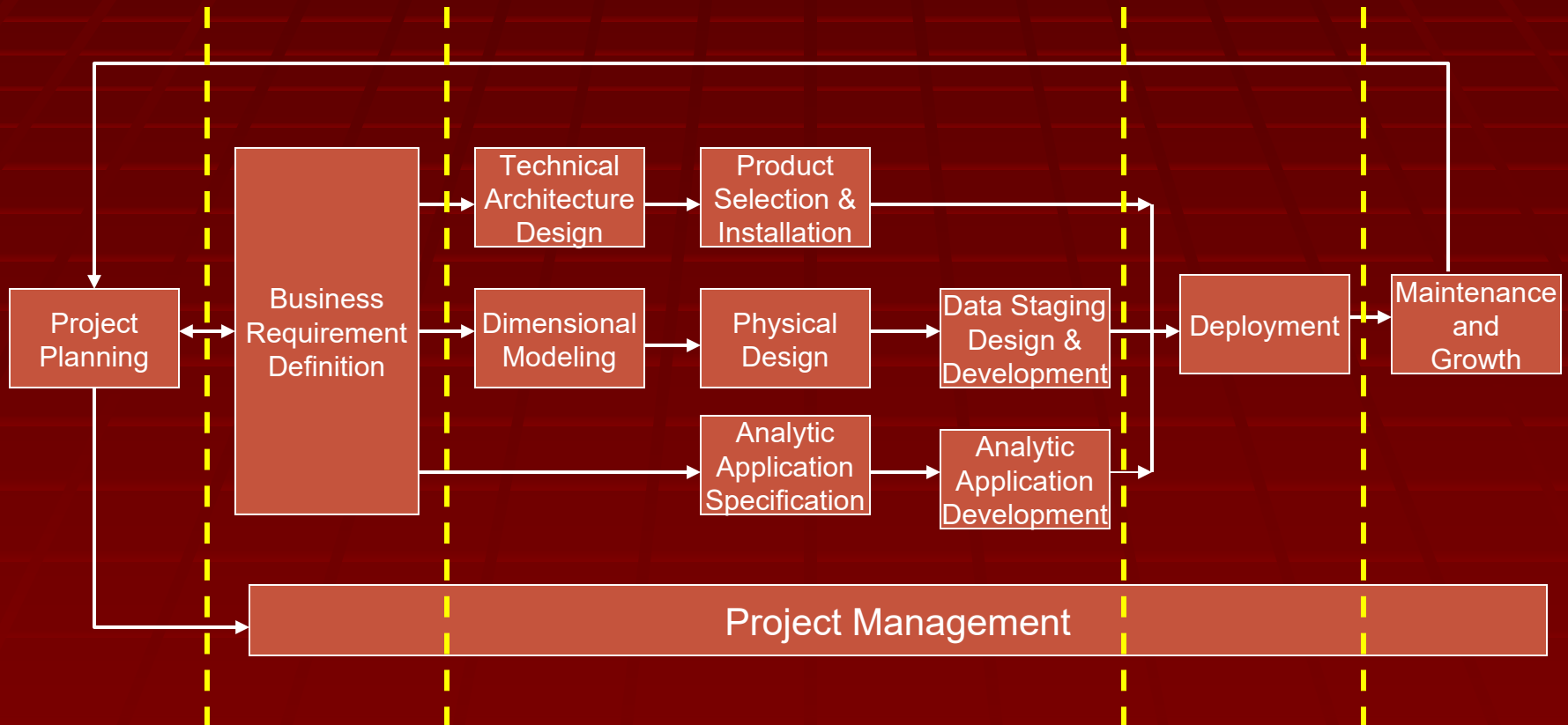
Email: ahsan101@yahoo.com

Data Warehouse (DWH) Lifecycle

Kimball's Approach

Business Dimensional Lifecycle Road map

Graphics



DWH Lifecycle: Key steps

DWH Lifecycle- *Step 1: Project Planning*

DWH Lifecycle- *Step 1: Project Planning*

DWH Lifecycle- *Step 1: Project Planning*

Establishing the core team

DWH Lifecycle- *Step 1: Project Planning*

DWH Lifecycle- *Step : 2 Requirements Definition*

DWH Lifecycle- *Step:2 Requirements Definition*

DWH Lifecycle- *Step:2 Requirements Definition*

DWH Lifecycle- *Step:2 Requirements Definition*

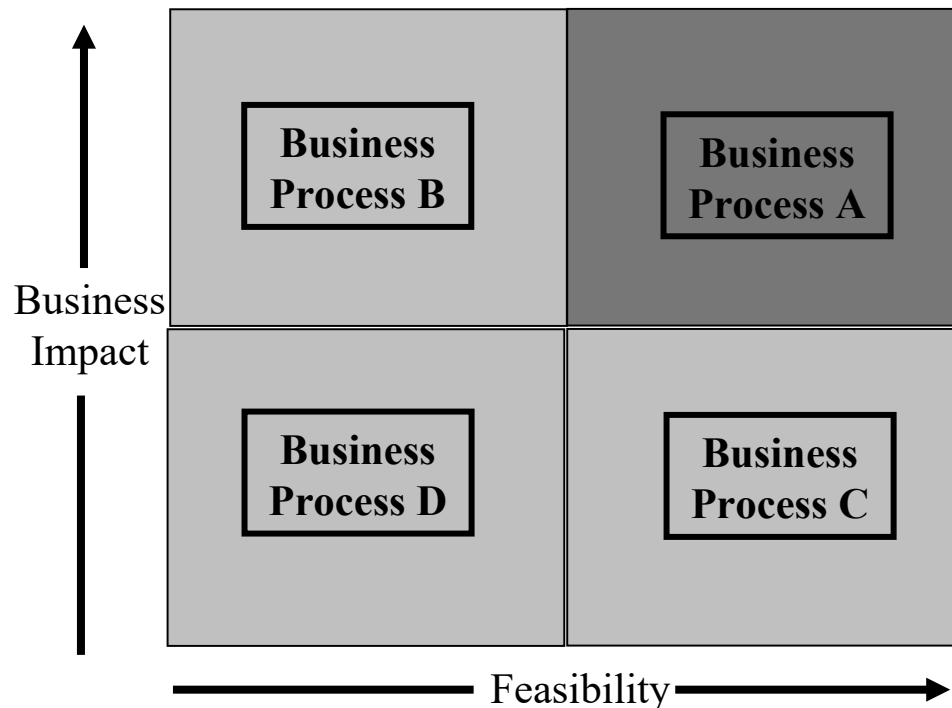
DWH Lifecycle- *Step:2 Requirements Definition*

DWH Lifecycle- *Step:2 Requirements Definition*

■ Post Collection

■ Prioritization

- Review & Prioritize findings
- Quadrant Technique



Graphics

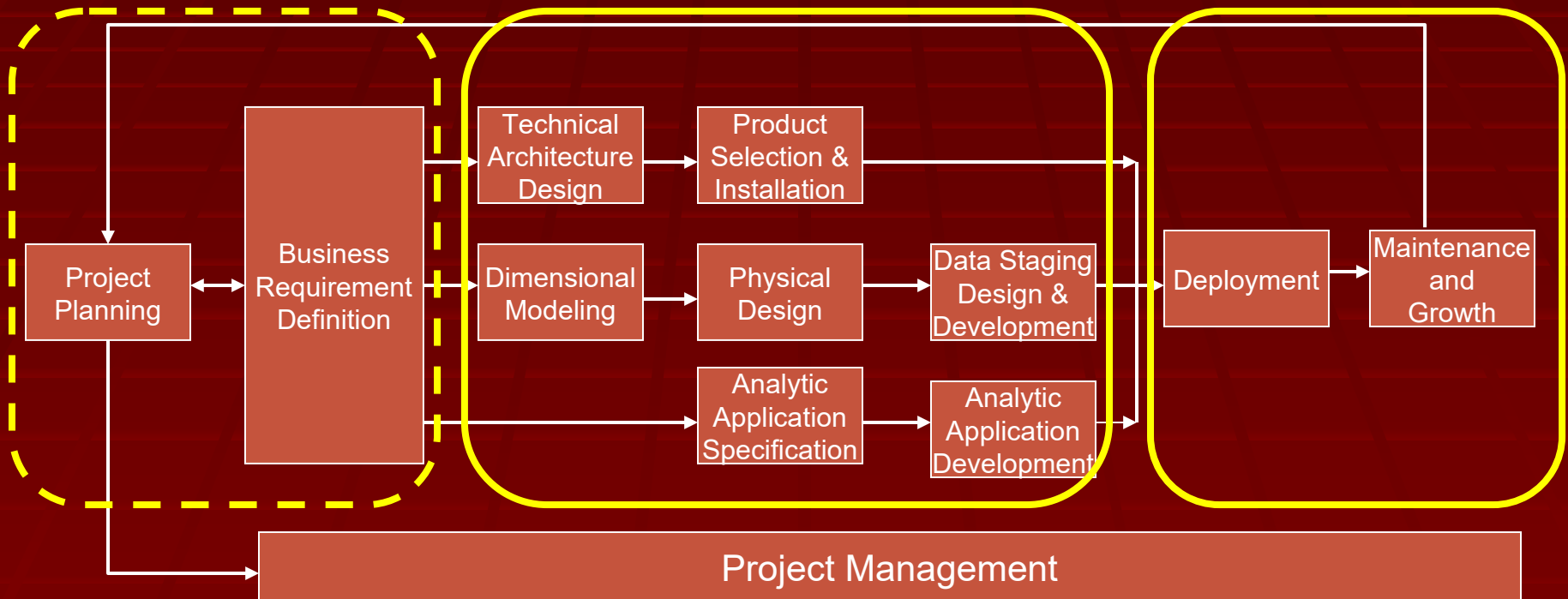
Lecture-34

DWH Implementation: Goal Driven Approach (2)

Data Warehouse (DWH) Lifecycle

Kimball's Approach

Business Dimensional Lifecycle Road map



Technical Architecture Design

Technical Architecture Design: More

DWH Lifecycle- *Step 3.1:Technology Track*

DWH Lifecycle- *Step 3.3: Analytic Applications Track*

DW Lifecycle- Step 4: Deployment

DW Lifecycle- *Step 5: Maintenance and Growth*

Virtual University of Pakistan

Data Warehousing

Lecture-35

DWH Implementation: Pitfalls, Mistakes, Keys

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

National University of Computers & Emerging Sciences, Islamabad

Email: ahsan1010@yahoo.com

5 Signs of trouble

11-Possible Pitfalls...

11-Possible Pitfalls...

11-Possible Pitfalls

Pitfall: Searching for the “Silver Bullet”

Pitfall: Extremes of Tech. Arch. Design

Top 10-Common Mistakes to Avoid...

Top 10-Common Mistakes to Avoid

Top 7-Key Steps for a smooth DWH implementation...

Top 7-Key Steps for a smooth DWH implementation...

Conclusions

Lecture-36
Course Project

Two parts

Part-I

Part-II

Part-II(a): Identify Organization

Large and Typical Early Adopters

Example DWH Target Organizations

Part-II(b): Life-Cycle Study...

Part-II(b): Life-Cycle Study...

What if you are not entertained?

What if you are never entertained?

Contents of Project Reports

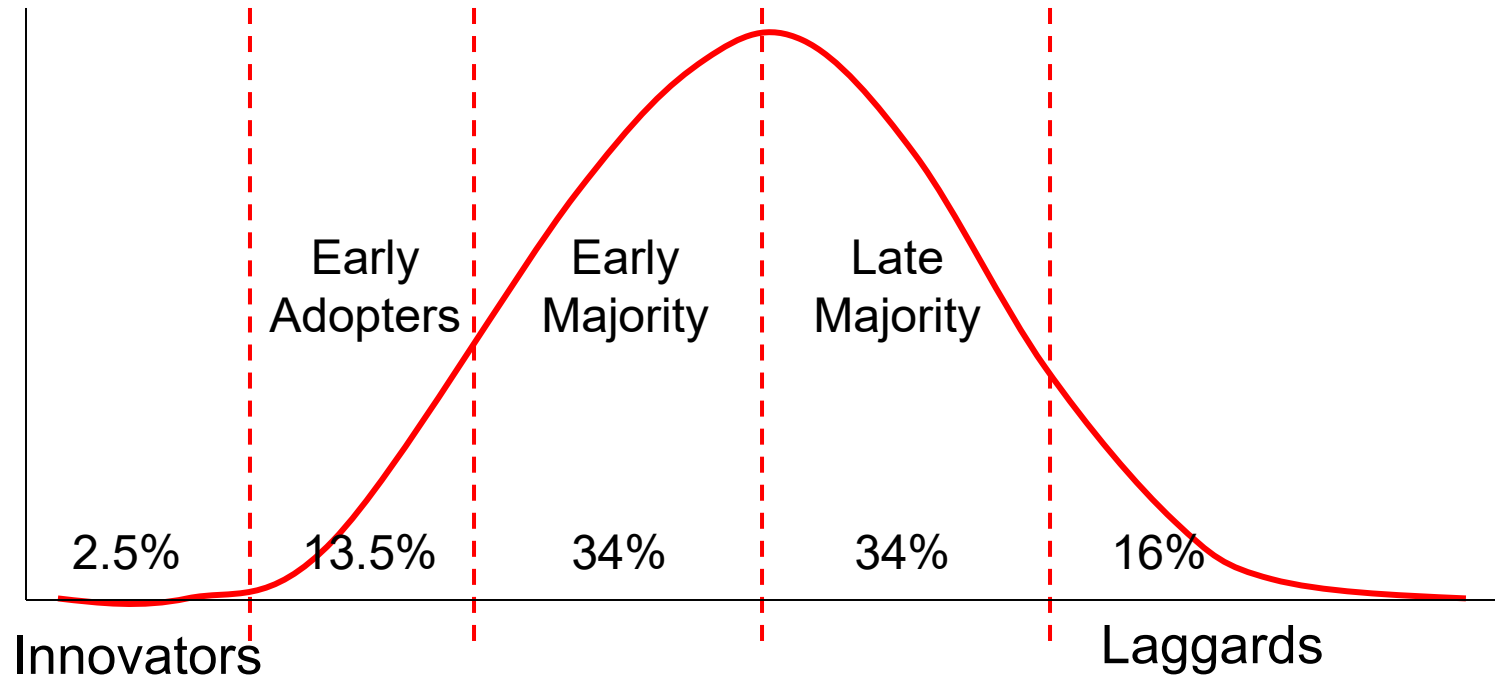
Format of Project Reports: Main

Format of Project Reports: Other

Why would companies entertain you?

Why you may be entertained?

Adoption/Innovation Curve



Categories explained

Lessons

Adoption/Innovation & DWH

Early Adopter

Now have TB data.
Improve performance with large number of users.
24x7 Availability.

Early Majority

Built Data Mart instead of DWH
How to develop a DWH?
How to make Meta Data repository?

Late Majority

Want to buy solutions instead of technology?
The more packaged, the better.
Selecting and retrofitting with business processes.

Virtual University of Pakistan

Data Warehousing

Lecture-37

Case Study: Agri-Data Warehouse

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

Background

Area under study

Area under study: Map



Graphics

Key

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

Markaz

- Bosan
- Qadirpurran
- Multan
- Makhdum Rashid
- Mumtazabad
- Shujabad
- Hafizwala
- Jalapur Pirwala
- Qasba Marral

Major Players

Pests & Predators

Graphics



Pests

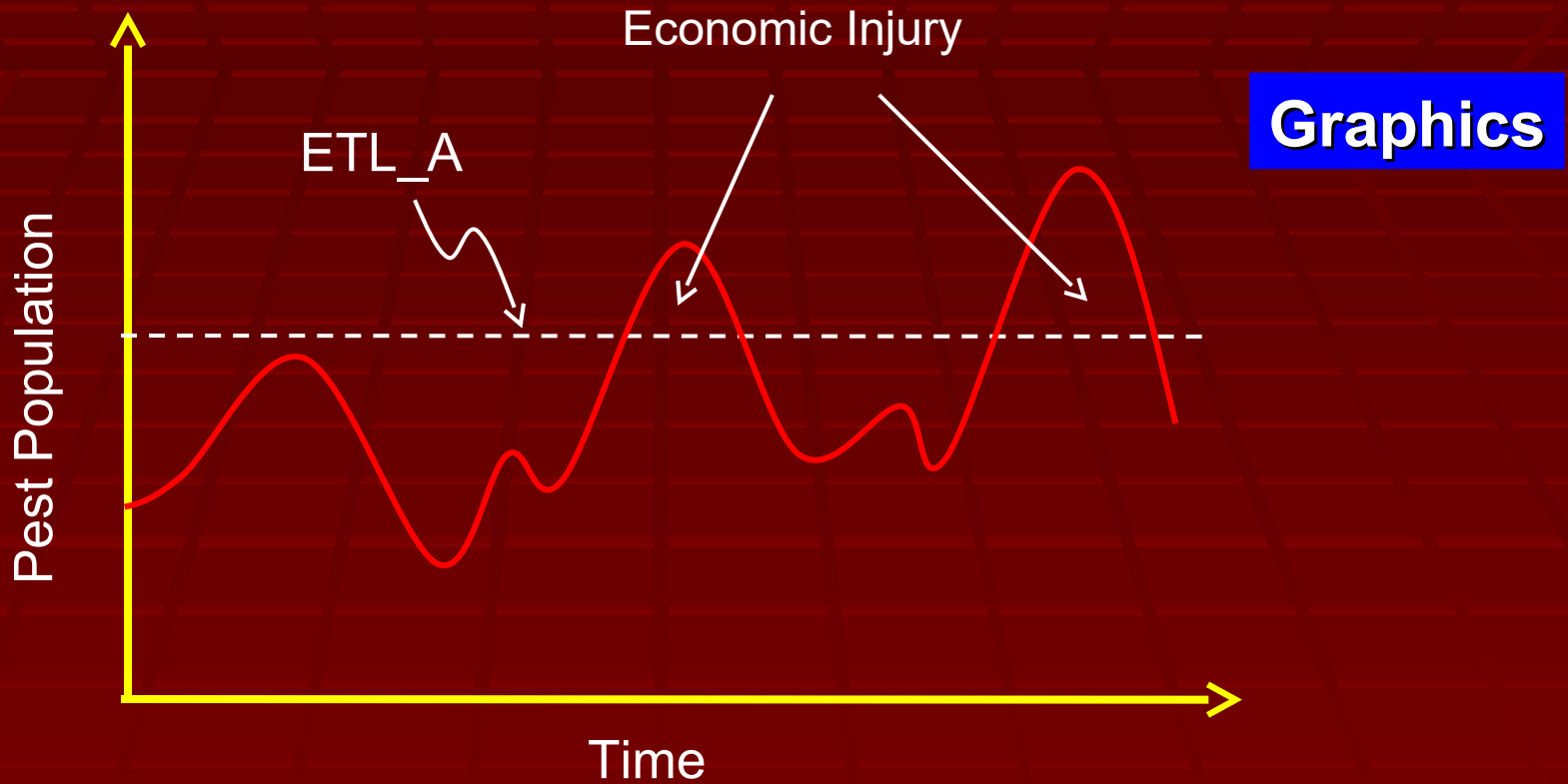


Predators

Source: United States Department of Agriculture (USDA)

Economic Threshold Level (ETL_A)

ETL_A: Graph



The need

The need: IT in Agriculture

Agro-Informatics

- *“I.T. sector is at the heart of the economic revival of Pakistan”* President of Pakistan, Launching of VU, Mar. 23, 2003.
- Agriculture is the backbone of our economy, about 70% of the population is dependent on it. ALL Goes to graphics
- IT is an enabler, and has the potential to benefit everyone when applied in Agriculture.
- IT + Agriculture: A win-win scenario.

Agro-Informatics: Interesting URL

ALL Goes to graphics

To know more about Agro-Informatics, visit

www.nu.edu.pk/cairindex.asp

How to go about?

The 12-step Approach of Shaku Atre

Phase_1: Planning & Design	Phase_2: Building & Testing
1. Determine Users' Needs 2. Determine DBMS Server Platform 3. Determine Hardware Platform 4. Information & Data Modeling 5. Construct Metadata Repository	6. Data Acquisition & Cleansing 7. Data Transform, Transport & Populate 8. Determine Middleware Connectivity 9. Prototyping, Querying & Reporting 10. Data Mining 11. On Line Analytical Processing

Phase_3: Roll-Out & Maintenance
12. Deployment & System Management

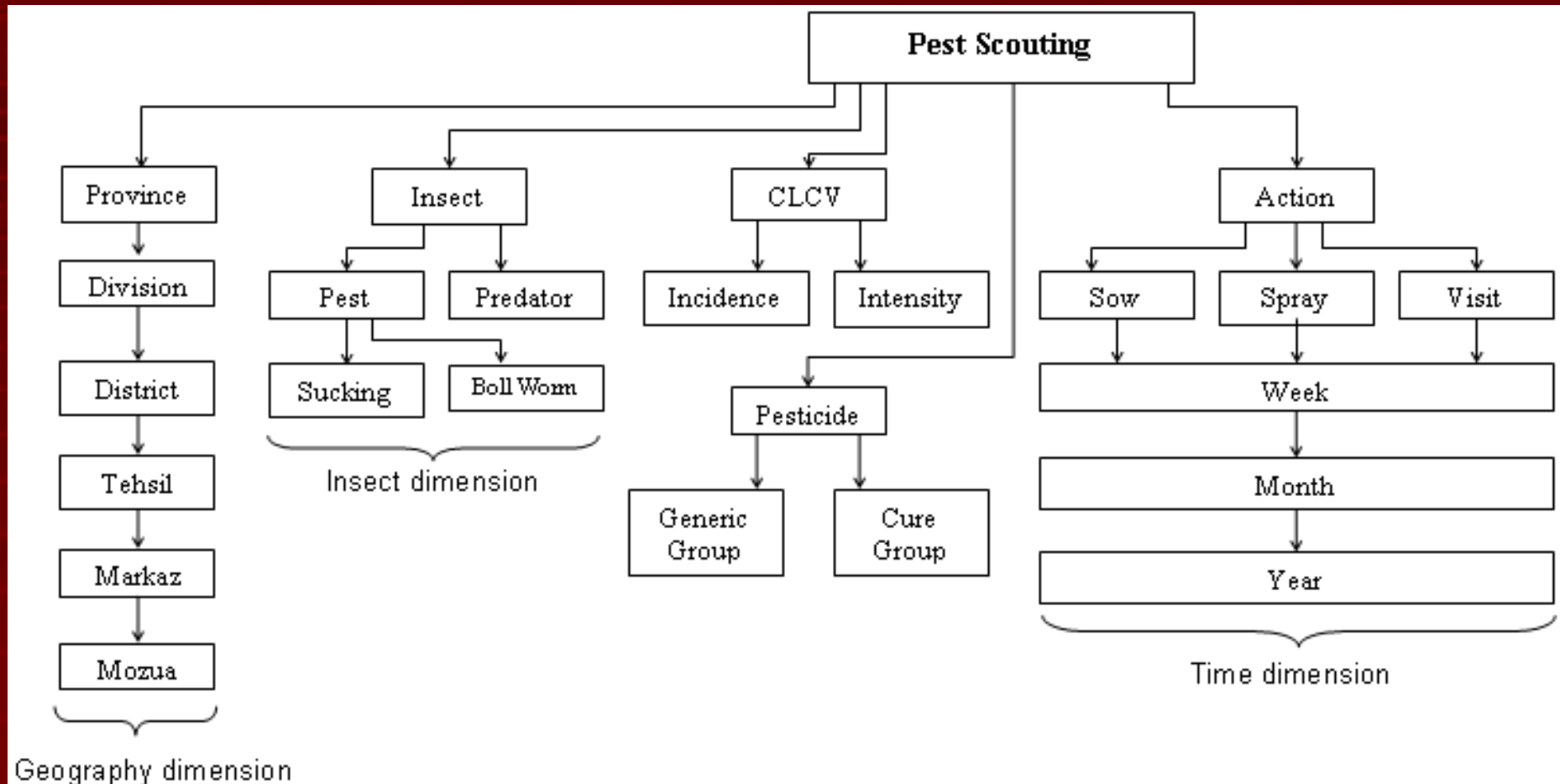
ALL Goes to graphics

Step-1: Determine User's needs

Step-1: Determine Users needs

Steps-2&3: Determine DBMS Sever & Hardware Platform

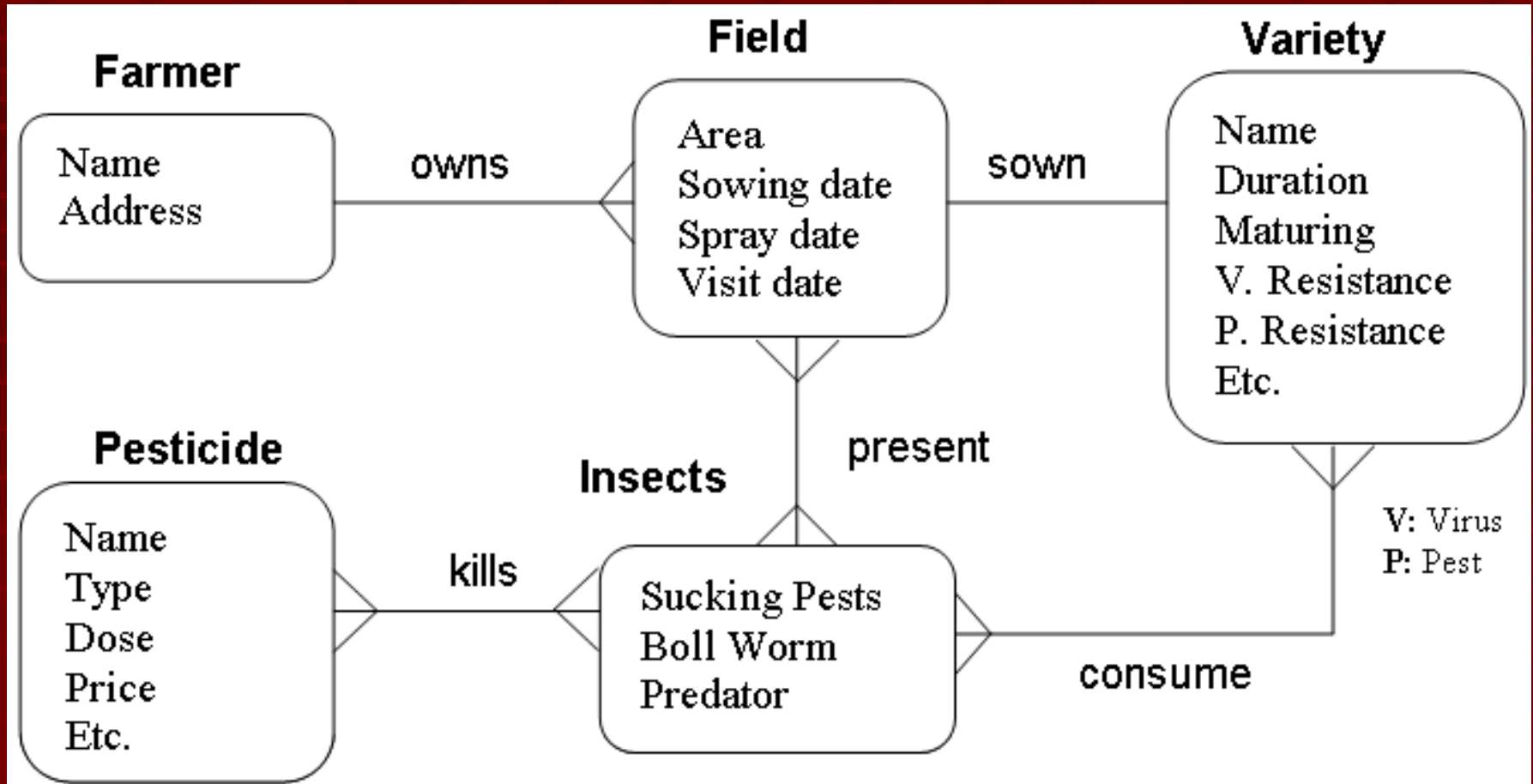
Step-4: Information & Data Modeling



Dimensional Modeling

Graphics

Step-4: Information & Data Modeling

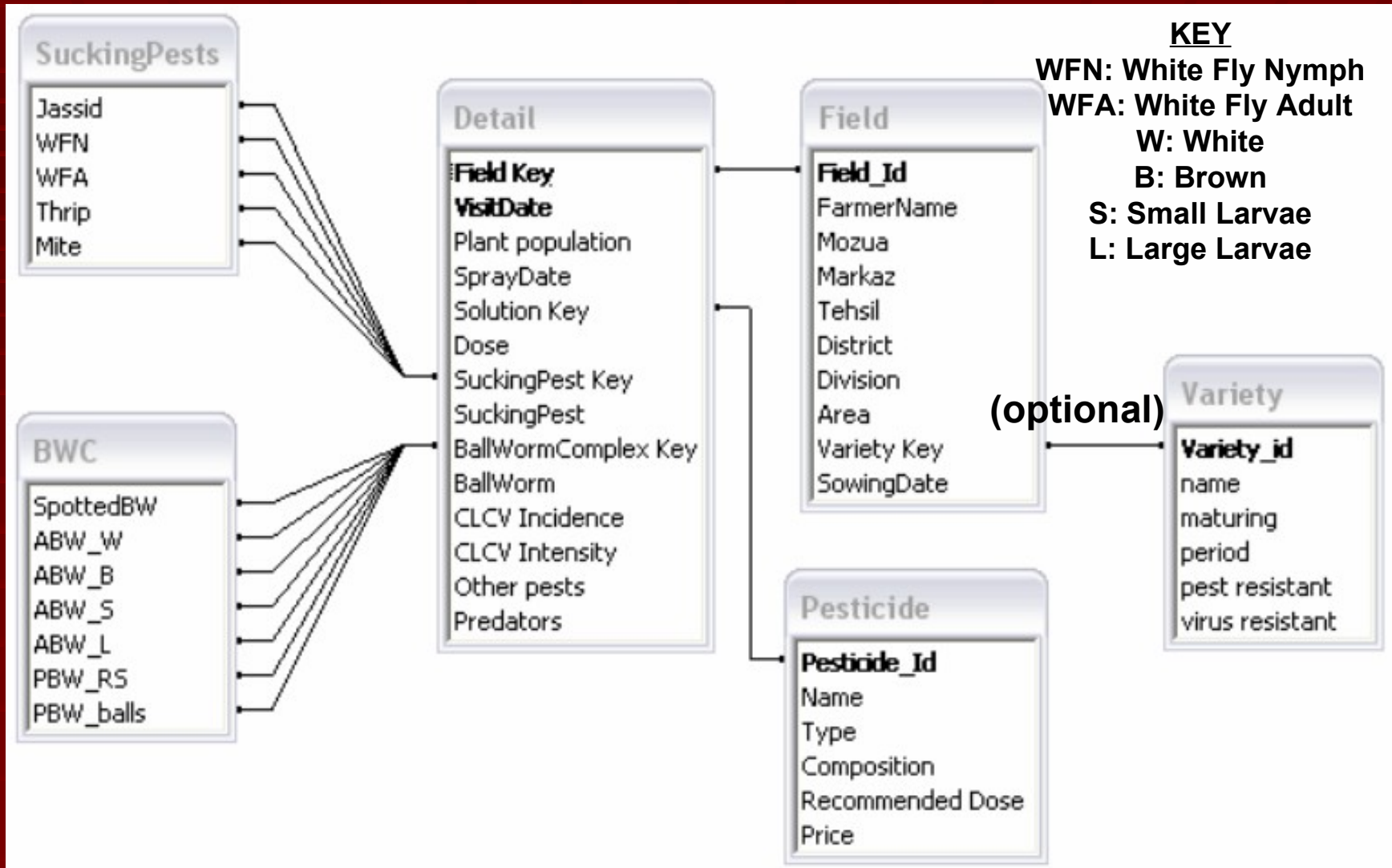


Simplified ERD

Graphics

Other field inputs such as irrigation, fertilizer etc. not included as data not available

Step-4: Information & Data Modeling



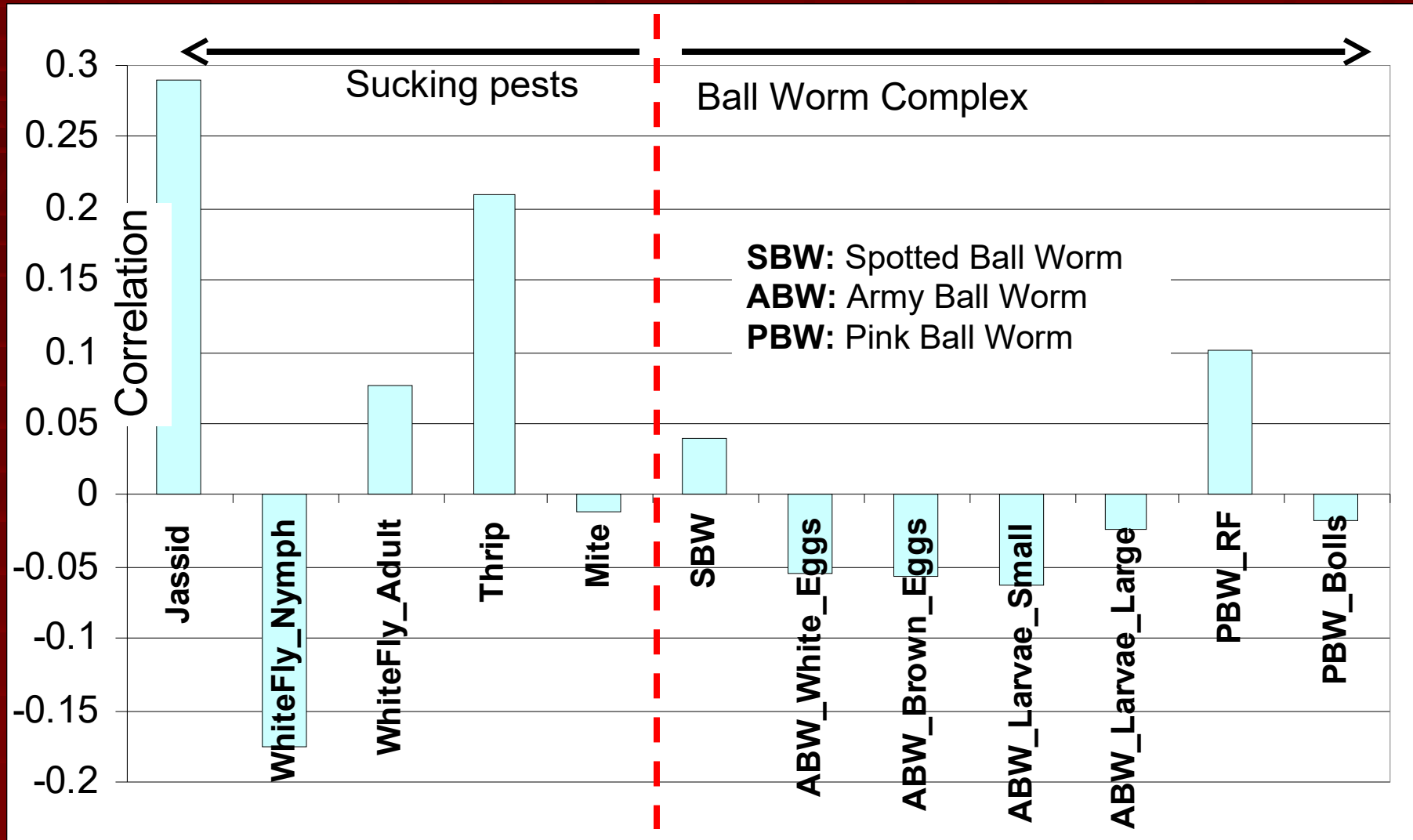
Logical & Physical Design

Graphics

Step-5: Construct Metadata Repository

Step-5: Surprise case

Graphics



If pest population is low, predator population will also be low, because there will be less “food” for predators to live on i.e. pests.

Virtual University of Pakistan

Data Warehousing

Lecture-38

Case Study: Agri-Data Warehouse

Ahsan Abdullah

Assoc. Prof. & Head

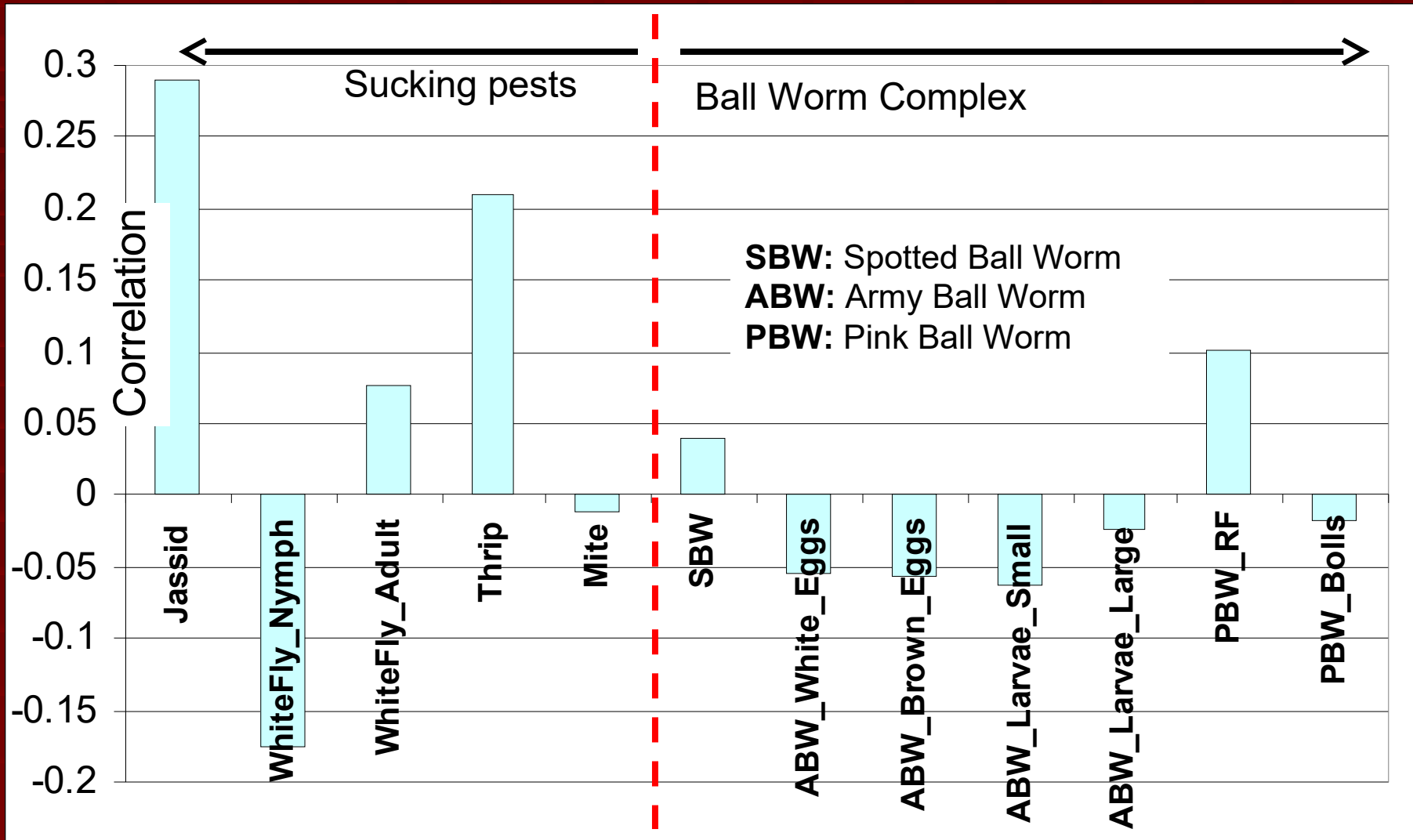
Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

Step-5: Surprise case

Graphics



If pest population is low, predator population will also be low, because there will be less “food” for predators to live on i.e. pests.

Step-6: Data Acquisition & Cleansing

Sr. No.	Name & Address of the Farmer	Date of Visit	Variety	Area in Acres	Date of Sowing	Plant Pop. (000)	SUCKING PESTS					BOLLWORM INFESTATION										CLEV		
							No. Left		Damage Value (%)		Thrip	MITE	SBW Larvae 25 Plants	ABW LARVAE				PBW PUPAE		Other Pests	Pred. (C/W)	% ECDEN CE	INTEN SITY	
							JAS SID	Whitefly	Thrip	MITE				SBW Larvae 25 Plants	No. of Plants				PUPAE					
															W	B	S	L	RF					Bols
	TEH. MULTAN																							
*	Markiz Mianqaz Abad-3		4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
1	M. Yasin m/ Habiba Sial	3/10	CIM 473	2	28/5/19	-	2.1	2.5	6	-	*	3	3	2	3	2	1	-	AW	-	-	-		
2	-do-	"	CIM 499	3	20/5/21	-	2.5	3.4	5	-	2	2	1	2	-	-	-	-	-	-	-	-		
3	Abdullah Hussain m/ Kot Rab Nawaz	"	CIM 473	6	5/6/20	0.1	2.1	3.5	8	-	*	3	3	4	3	2	-	-	-	-	-	-		
4	-do-	"	CIM 446	2	30/5/22	-	2.7	2.1	5	-	2	2	1	1	-	2	-	-	-	-	-	-		
5	M. Jaffar m/-do- Ariani Roda.	"	945	1	28/5/21	-	2.1	3.1	4	-	*	3	2	3	2	2	-	-	AW	-	-	-		
6	M. Fiaz m/-do-	"	CIM 473	5	2/6/20	-	2.2	3.4	6	-	2	2	3	3	2	2	-	-	11	-	-	-		

Graphics

Hand filled pest scouting sheet

Sr. No.	Name & Address of the Farmer	Date of Visit	Variety	Area in Acres	Date of Sowing	Plant Pop (000)	SUCKING PESTS					BOLLWORM INFESTATION										CLEV			
							No. Left		Damage Value (YES/NO)		Thrip	MITE	SBW Larvae 25 Plants	ABW LARVAE				PBW PUPAE		Other Pests	Pred. (C/W)	% ECDEN CE	INTEN SITY	Othr of Sowing	
							JAS SID	Whitefly	Thrip	MITE				SBW Larvae 25 Plants	No. of Plants				PUPAE						
															W	B	S	L	RF						Bols
1	TEHSIL SHUJABAD.																								
2	MARKIZ SHUJABAD.	14/3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
-do-	Ghulam Hussan Bastimithu	16/6	CIM473	4F	6/6	21	0.4	1.5	1.1	2															
-do-	Nazir Ahmad Balalwajhay	"	SH901	2F	"	22	0.3	1.2	1.4	3															
-do-	"	"	"	1F	21/5	19	0.2	-	1.3	2															
-do-	"	"	CIM446	2F	26/5	20	0.4	0.4	2.4	4															
-do-	Muneer Hussain Talkot	"	CIM473	1F	23/5	21	0.1	1.3	2.3	3															
-do-	"	"	FI901	4R	15/5	20	1.6	1.1	2.9	4															
-do-	M. Farooq Rukan Hatti	"	N-78	2R	18/5	19	0.4	2.0	4.2	2															
-do-	Wali Muhammad RS	13/8	FI901	2F	28/5	20	0.1	0.2	1.6	3															
-do-	Niaz Hussain	"	"	"	"	"	"	"	"	"															
-do-	"	"	KAR	2R	2/6	21	0.5																		

Typed pest scouting sheet

Step-6: Issues

Step-6: Why the issues?

Step-7: Transform, Transport & Populate

Motivation For Transformation

Graphics

Static Attributes	
1	Farmer Name
2	Farmer Address
3	Field Acreage
4	Variety(ies) Sown
5	Sowing date
6	Sowing method

Dynamic Attributes	
1	Date of Visit
2	Pest Population
3	CLCV
4	Predator Population
5	Pesticide Spray Dates
6	Pesticide(s) Used

Step-7: Resolving the issue

Graphics

	Before	After
Area (acers): 2001	23,293	14,187
Area (acers): 2002	26,088	13,693
Farmers	2,696	1,567

Step-8: Middleware Connectivity

Step-9-11: Prototyping, Querying & Reporting

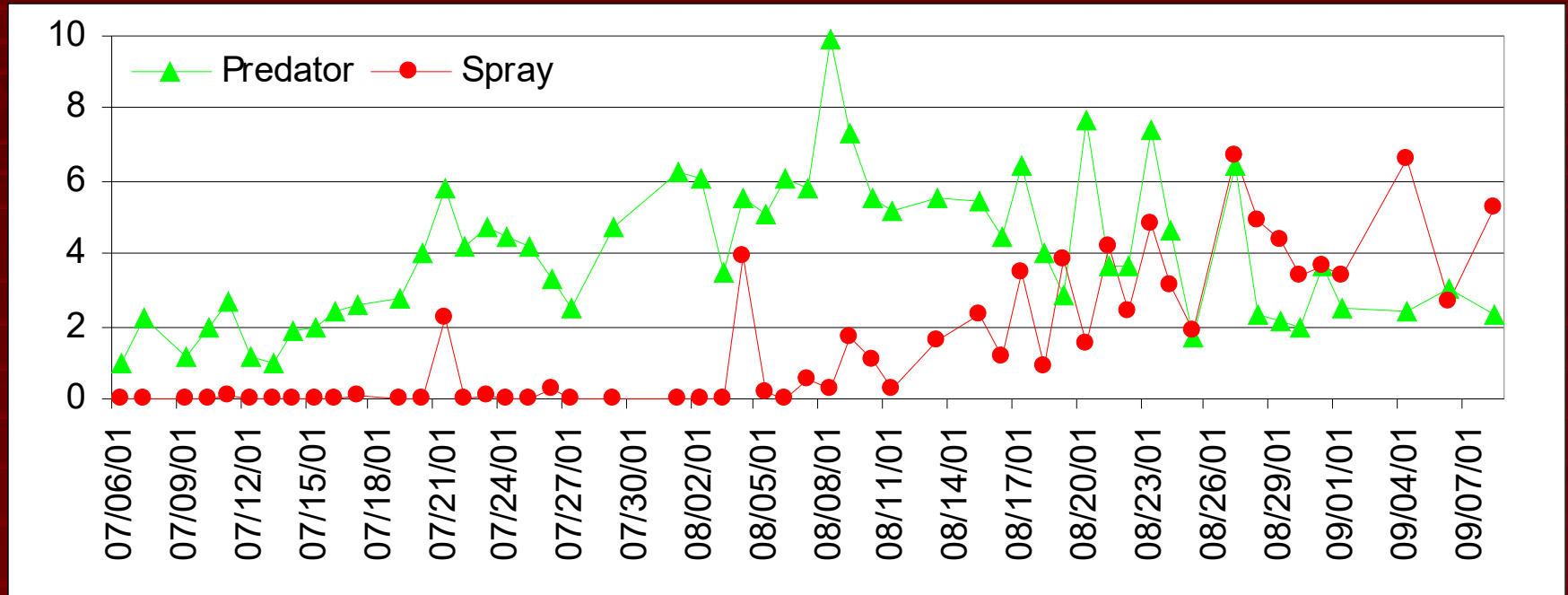
```
SELECT Date_of_Visit, AVG(Predators),  
        AVG(Dose1+Dose2+Dose3+Dose4)  
FROM Scouting_Data  
WHERE Date_of_Visit < #12/31/2001#  
      and predators > 0  
GROUP BY Date_of_Visit;
```

Graphics

Step-12: Deployment & System Management

Agri-DSS usage: Data Validation

Agri-DSS usage: Data Validation Graph

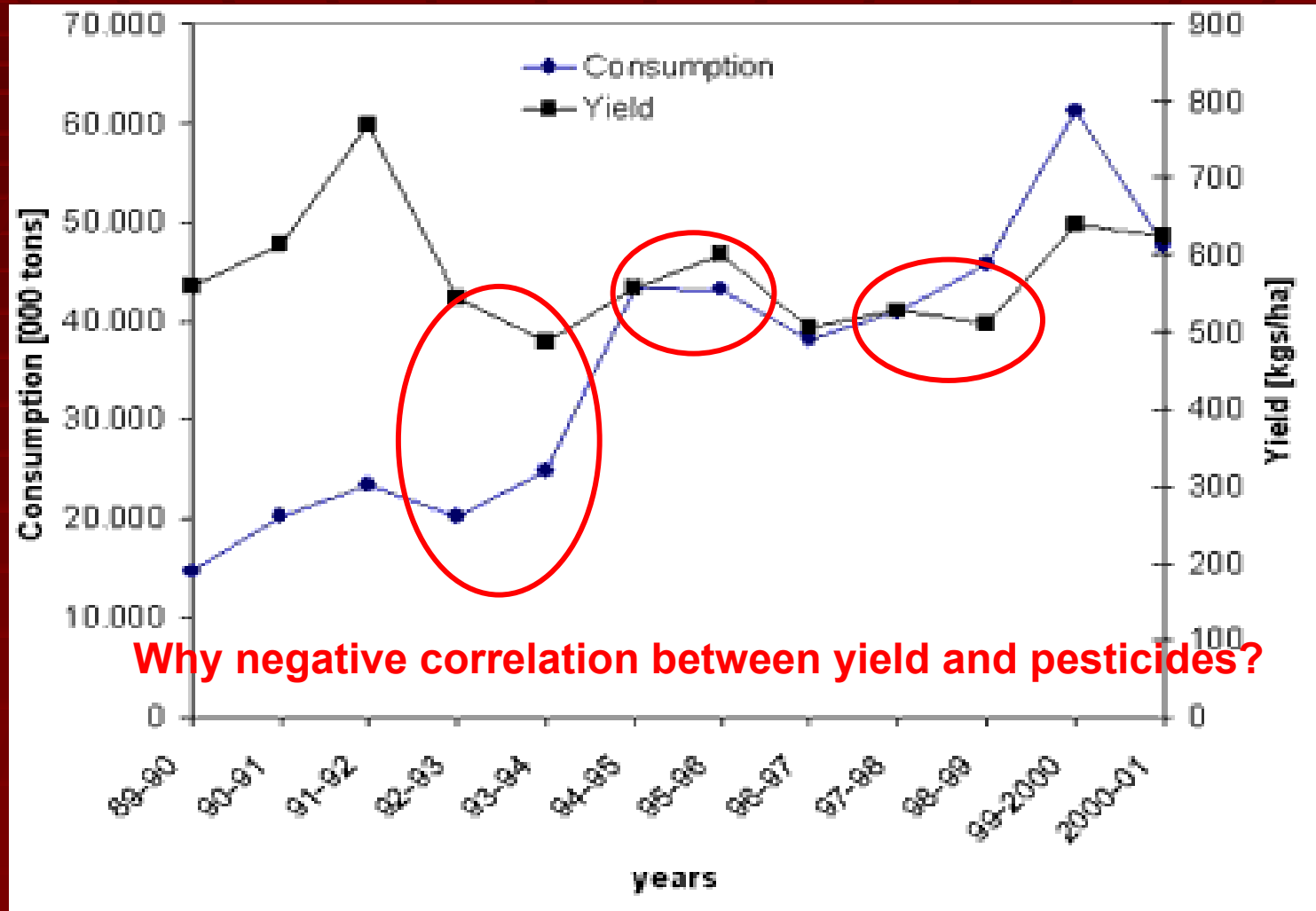


ALL goes to graphics

Agri-DSS usage: FAO report

Graph

Graphics

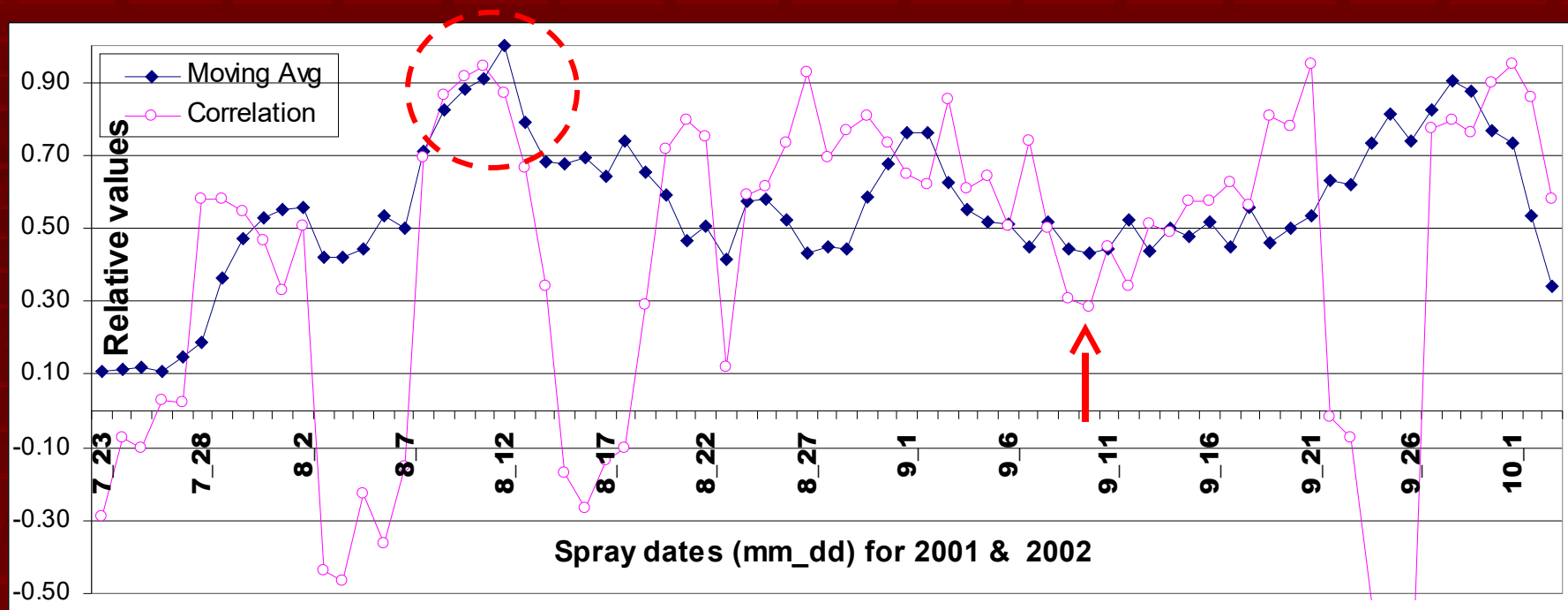


Why negative correlation between yield and pesticides?

Using pesticides to increase yield.

Agri-DSS usage: Spray Dates

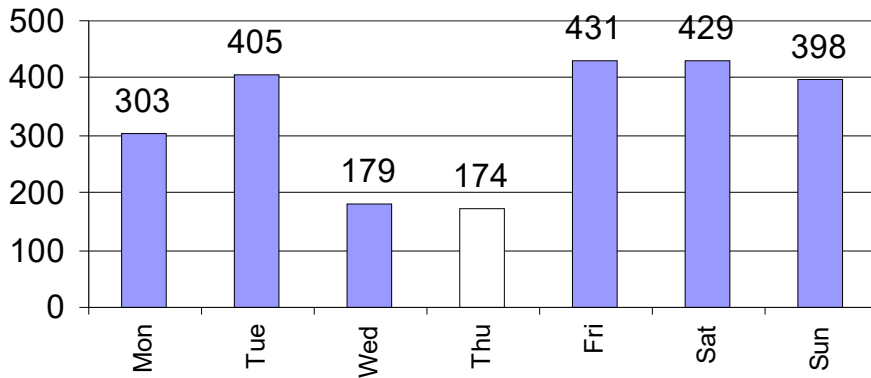
Agri-DSS usage: Spray Dates Graph



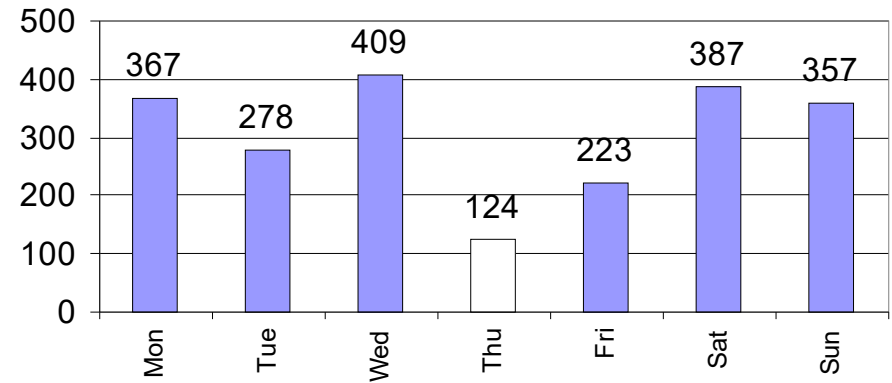
Agri-DSS usage: Explaining Findings

Agri-DSS usage: Sowing Dates

2001: Sowing week_day



2002: Sowing week_day



Graphics

Conclusions & Lessons

- ETL is a big issue.
- Each farmer is repeatedly visited
- There is a skewness in the scouting data.
- Decision-making goes all the way “down” to the extension level.

All goes to graphics

Virtual University of Pakistan

Data Warehousing

Lab Lect-1

DTS: Introduction

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

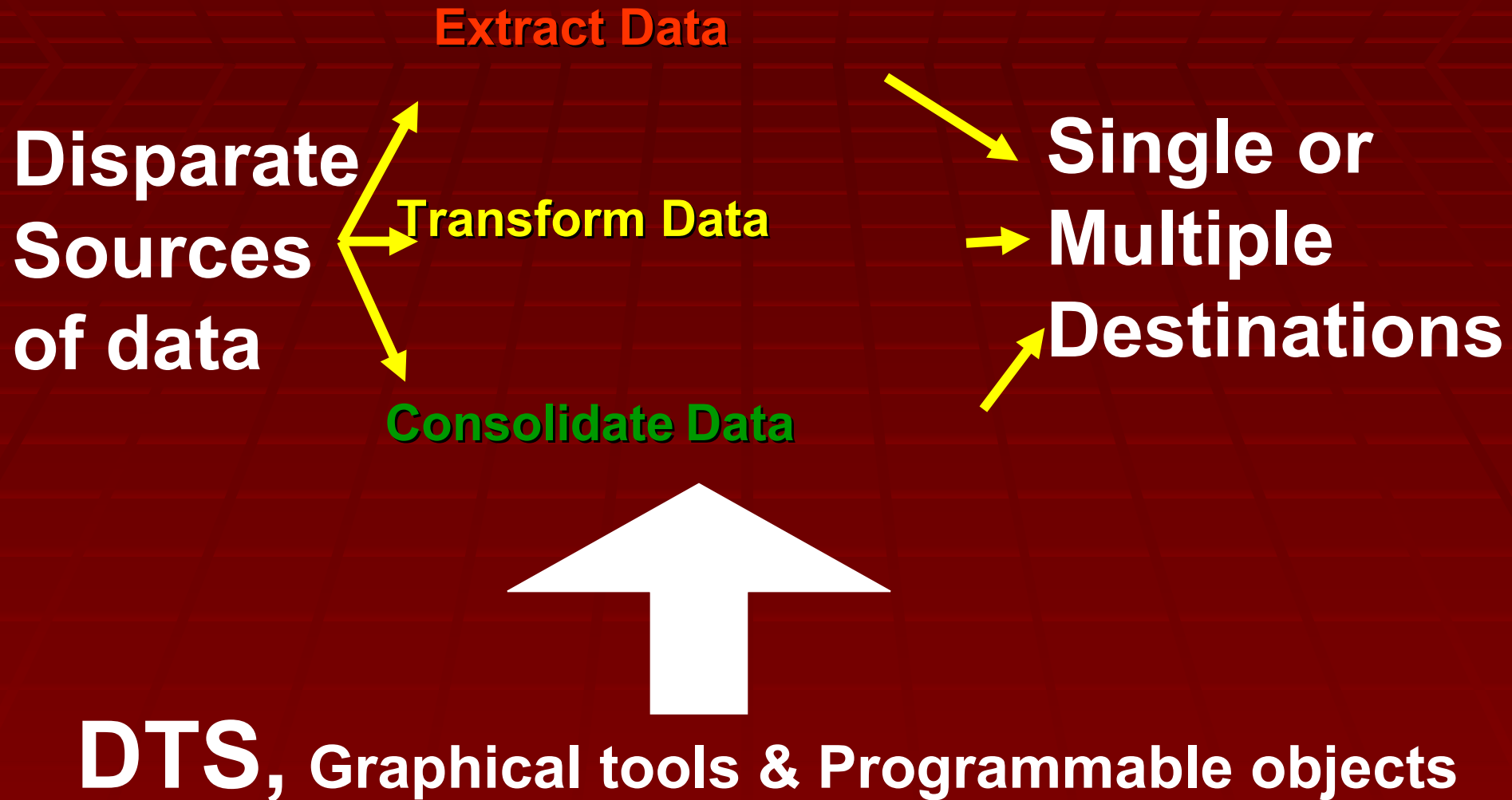
Reference Book

Carl Rabeler

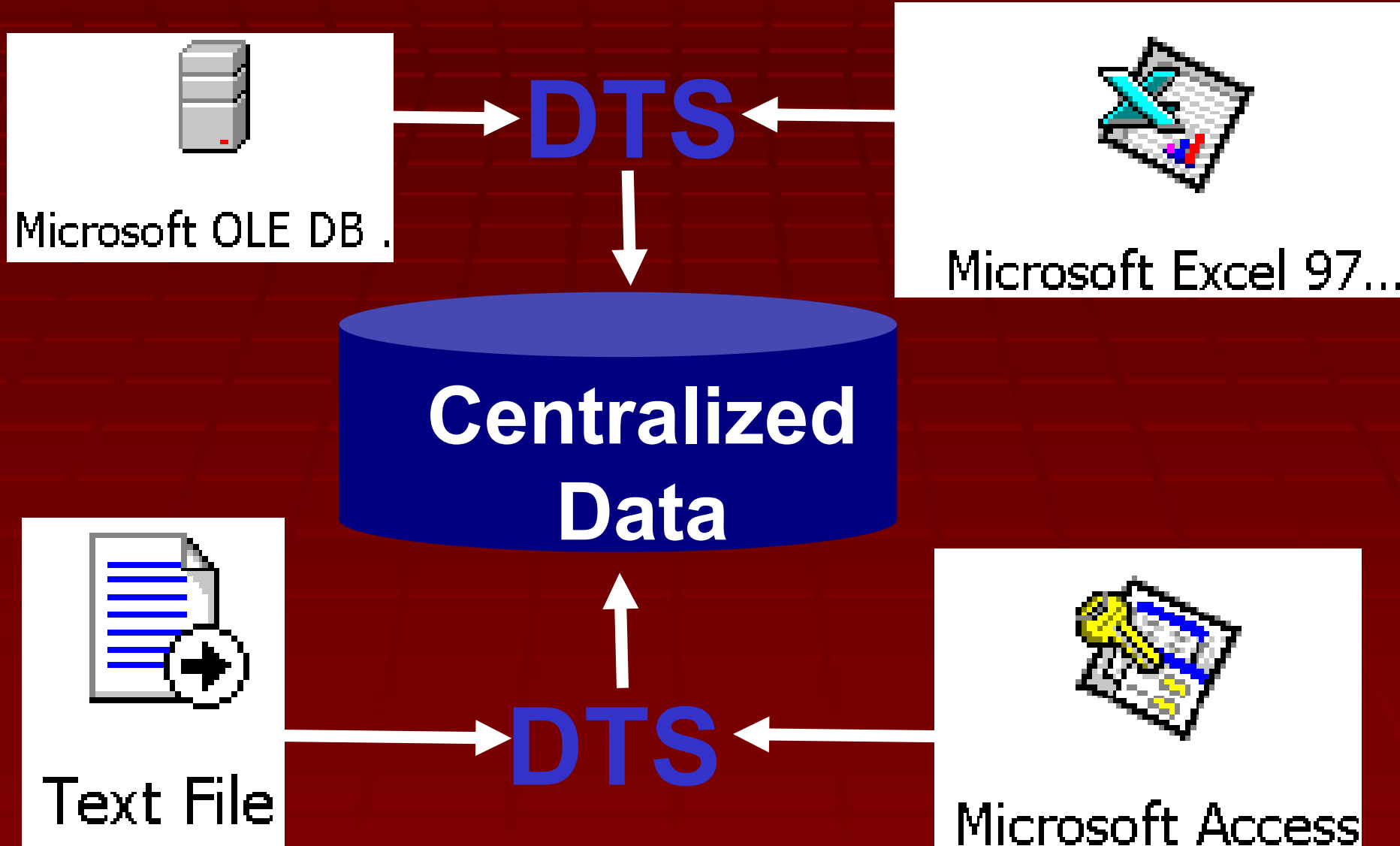
MS SQL Server 2000 DTS Step-By-Step

Data Transformation Services

DTS Overview



DTS Overview: Concept



DTS Overview: Example

MUHAMMED ANWAR	8-JUN-1968
CH MUHMD. ASLAM	23-Nov-1980

DTS

MOHAMMAD ANWAR	08/06/1968
CHOUDHARY MOHAMMAD ASLAM	23/11/1980
AHMAD JAHANZEB	05/08/1980
MOHAMMAD FARRUKH	08/11/1979

DTS

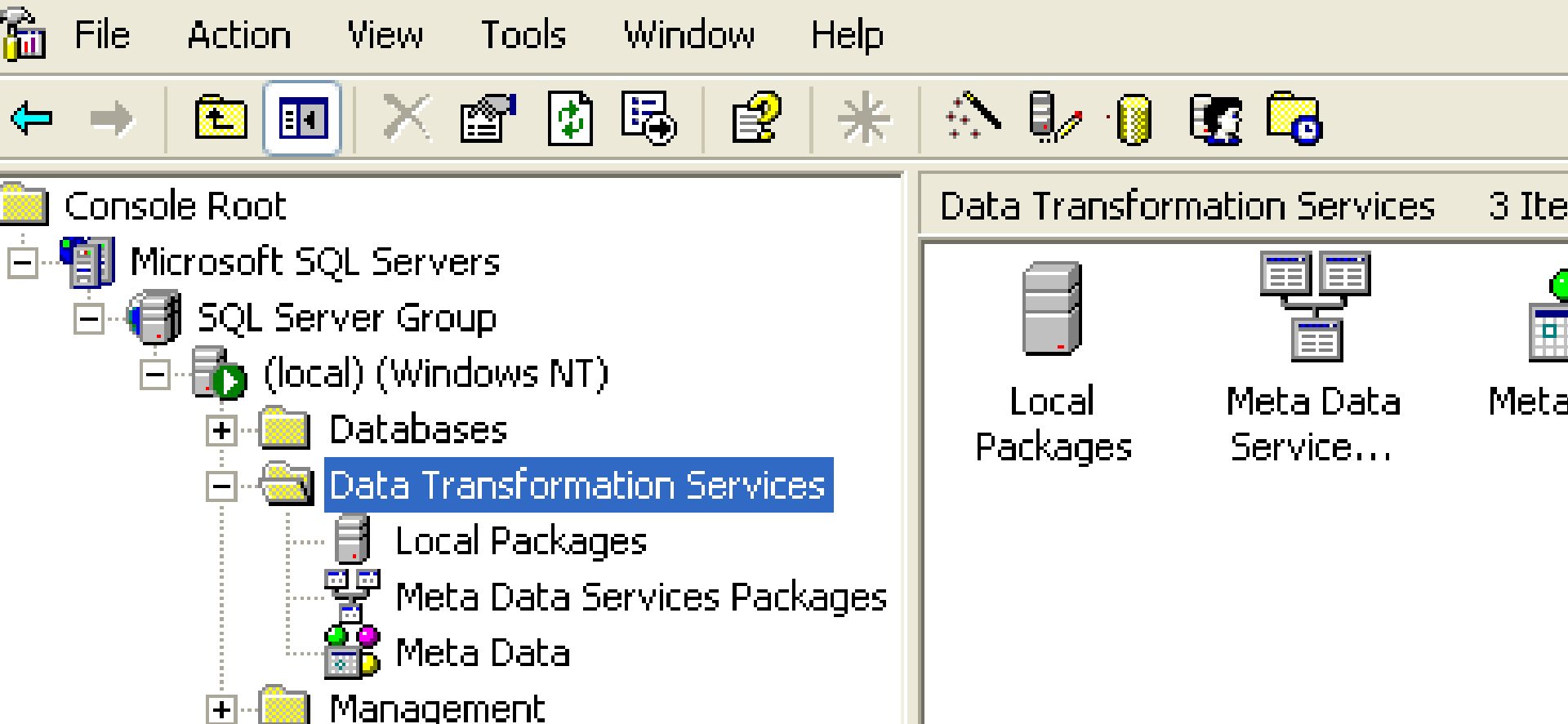
AHMED JAHANZEB	8-5-80
M. FARRUKH	11-8-79

DTS Overview: Operations

DTS Overview: Tools

SQL Server Enterprise Manager

SQL Server Enterprise Manager - [Console Root\Microsoft SQL Servers\



Path: Start >> Programs >> Microsoft SQL Server >> Enterprise Manager

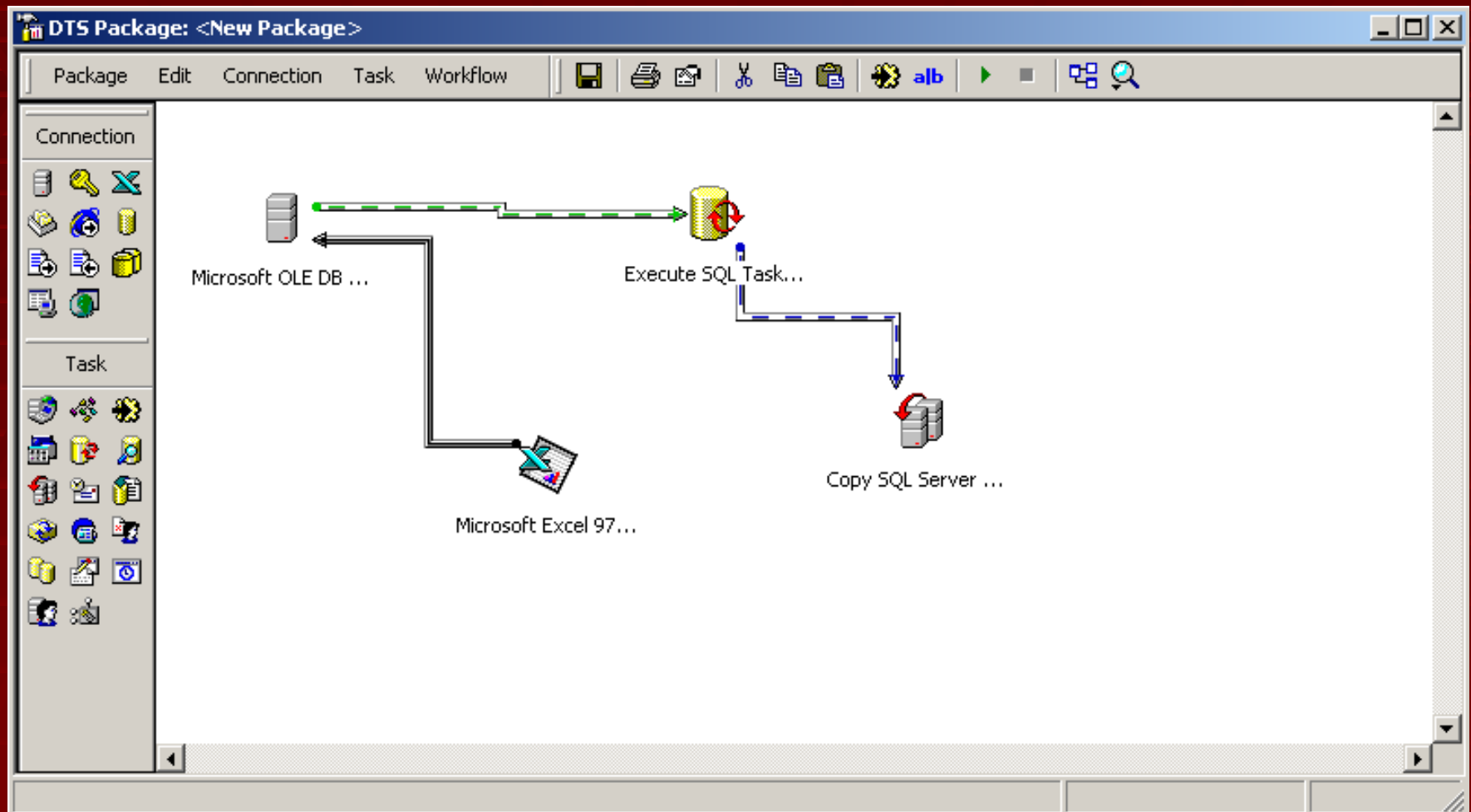
DTS Overview: Market

DTS Basics

1. DTS Package

- **Connections**
- **DTS tasks**
- **DTS transformations**
- **Workflows**

1. DTS Package: Contents



1. DTS Package: Execution

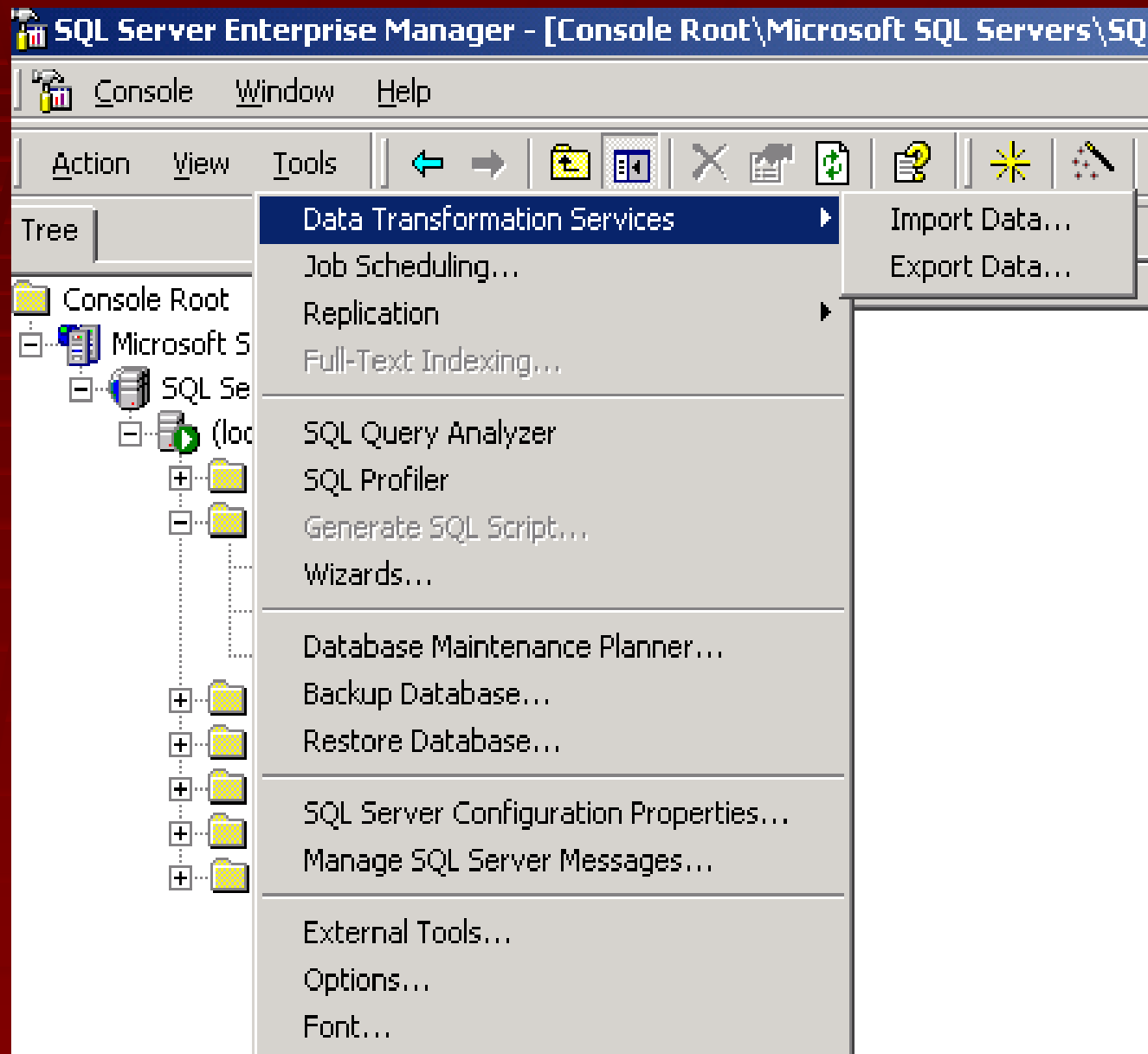
1. DTS Package: Creating

- **Import/Export wizard**
- **DTS Designer**
- **Programming DTS applications**

1. DTS Package: Import/Export Wizard

1. Expand tree node 'Data Transformation Services' and select the option for available location to save package

2. Tool>Data Transfer Service>Import/Export



1. DTS Package: Import/Export Wizard

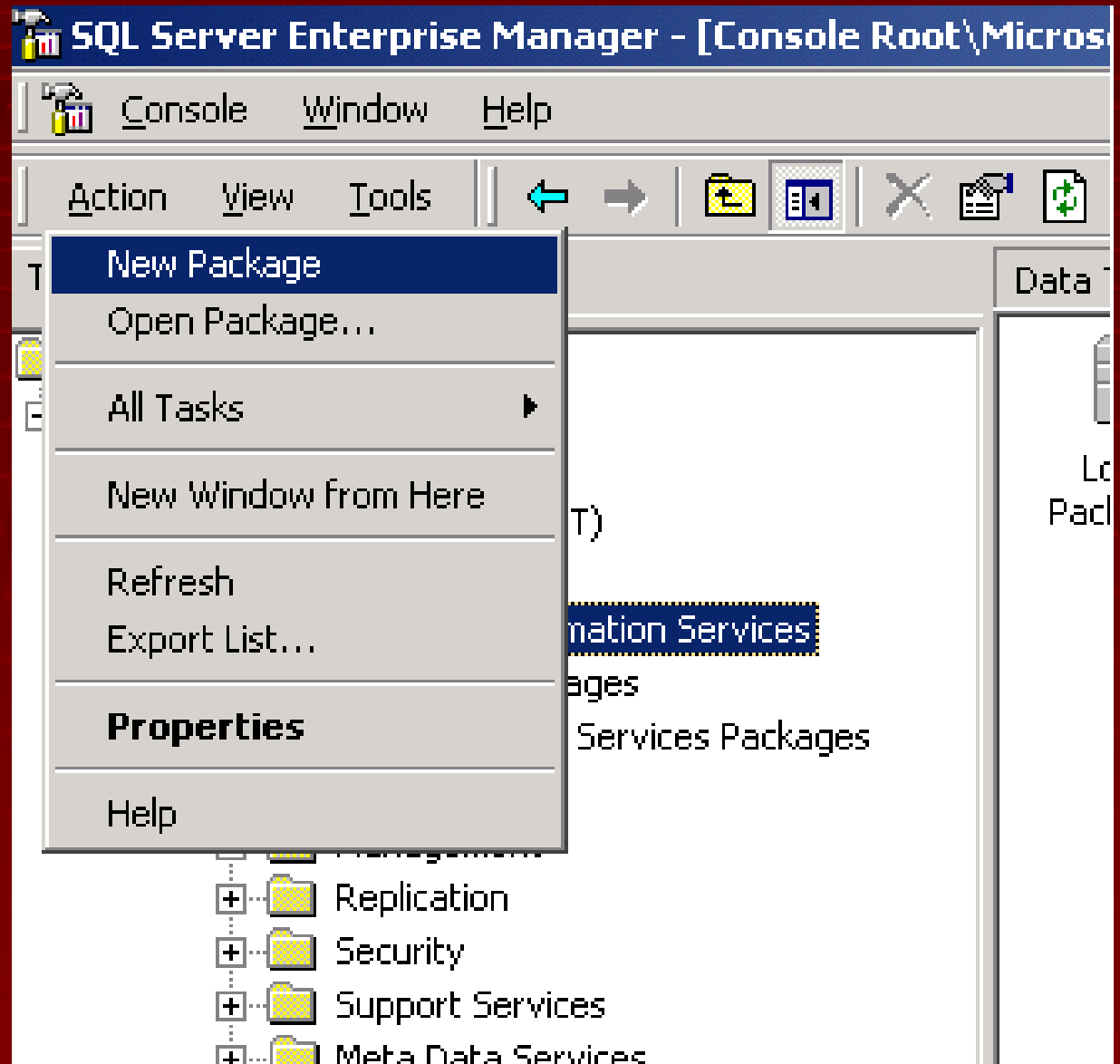


1. DTS Package: Import/Export Wizard

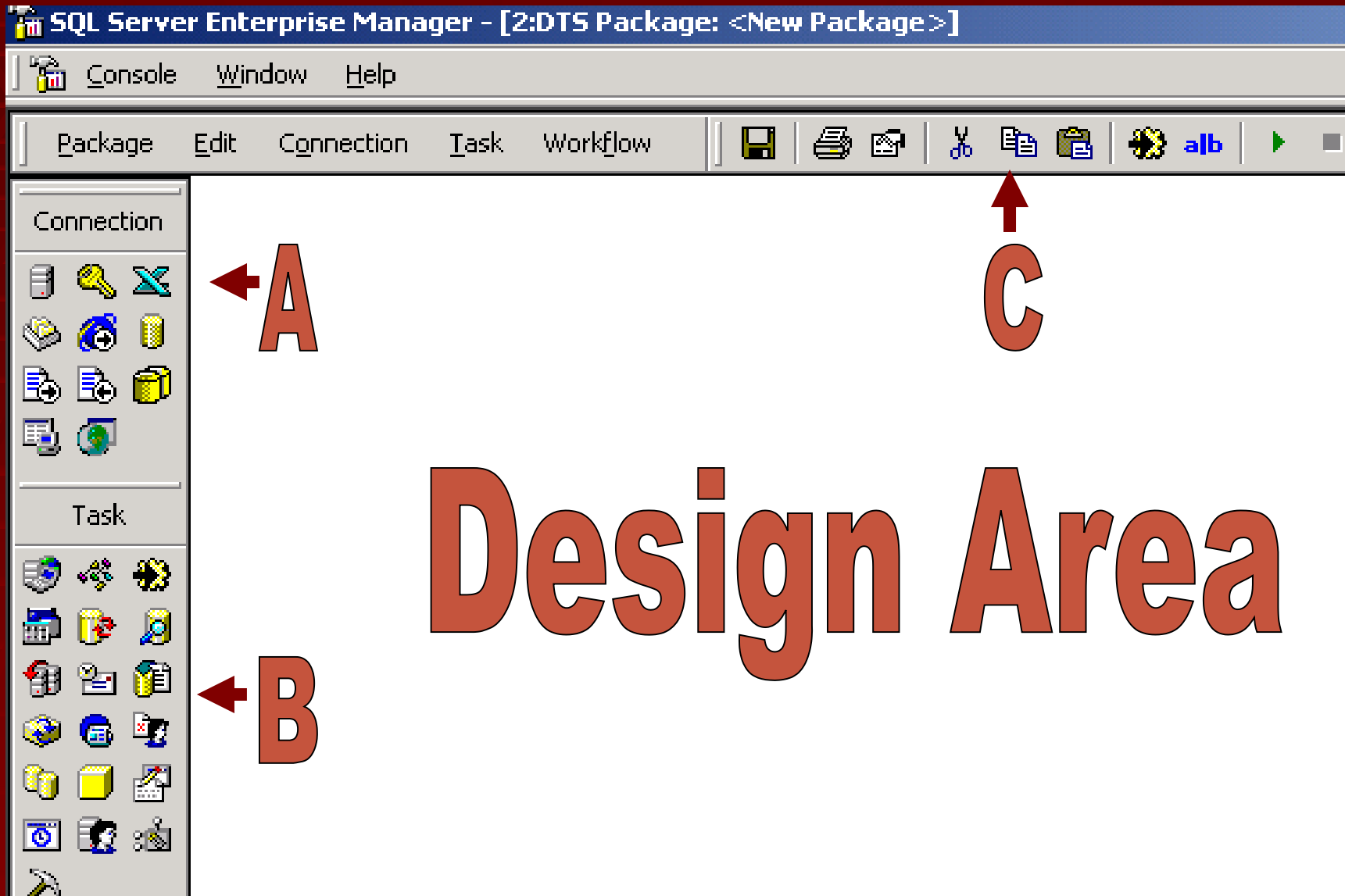
1. DTS Package: Designer

1. Expand tree node 'Data Transformation Services' and select the option for available location to save package

2. Action>New Package



1. DTS Package: Designer

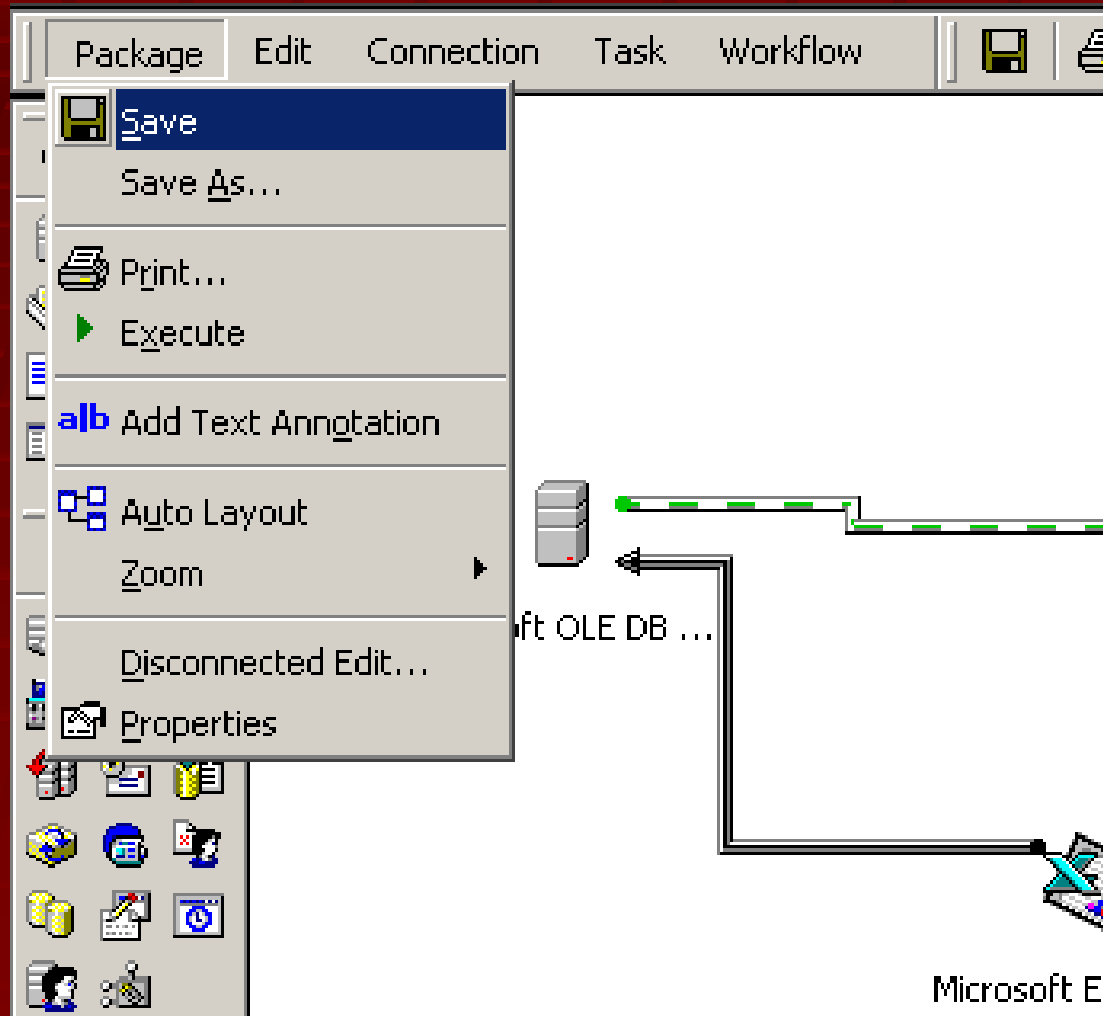


1. DTS Package: Designer

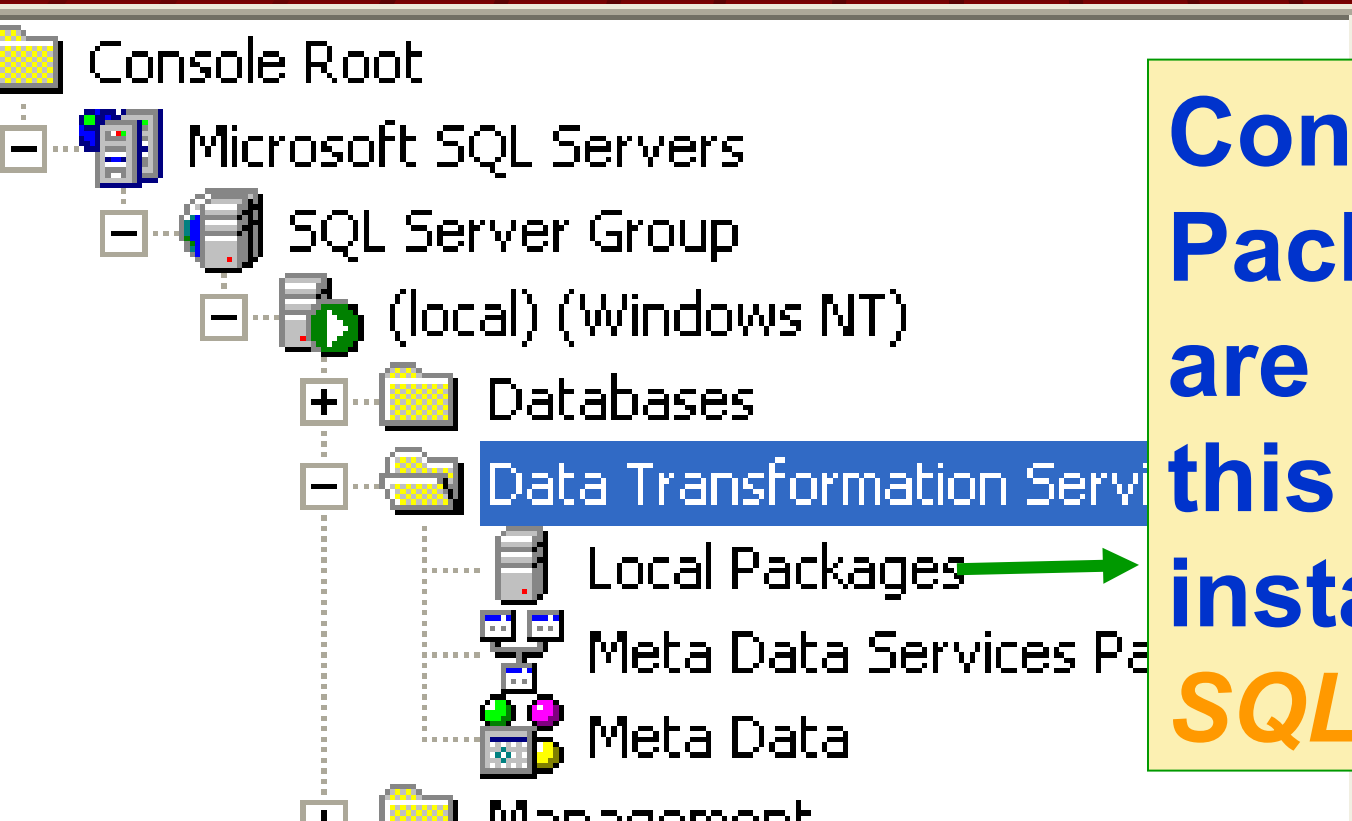
1. DTS Package: Programming

1. DTS Package: Saving

1. DTS Package: Saving

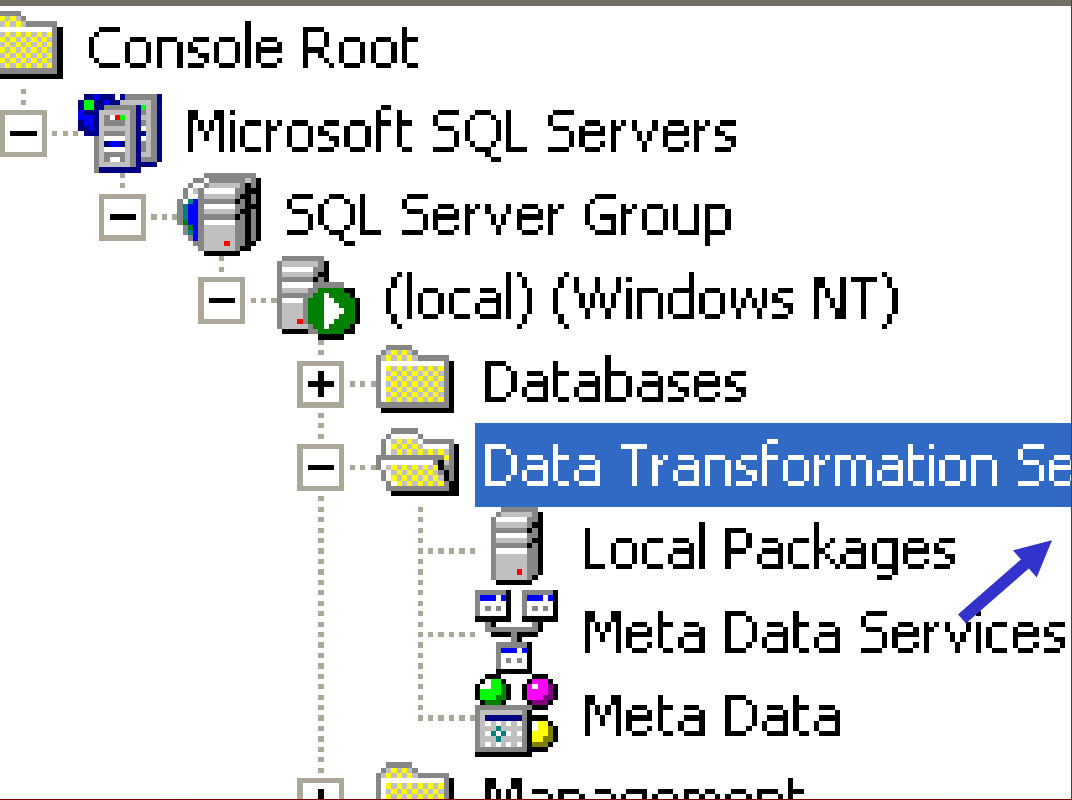


1. DTS Package: Saving



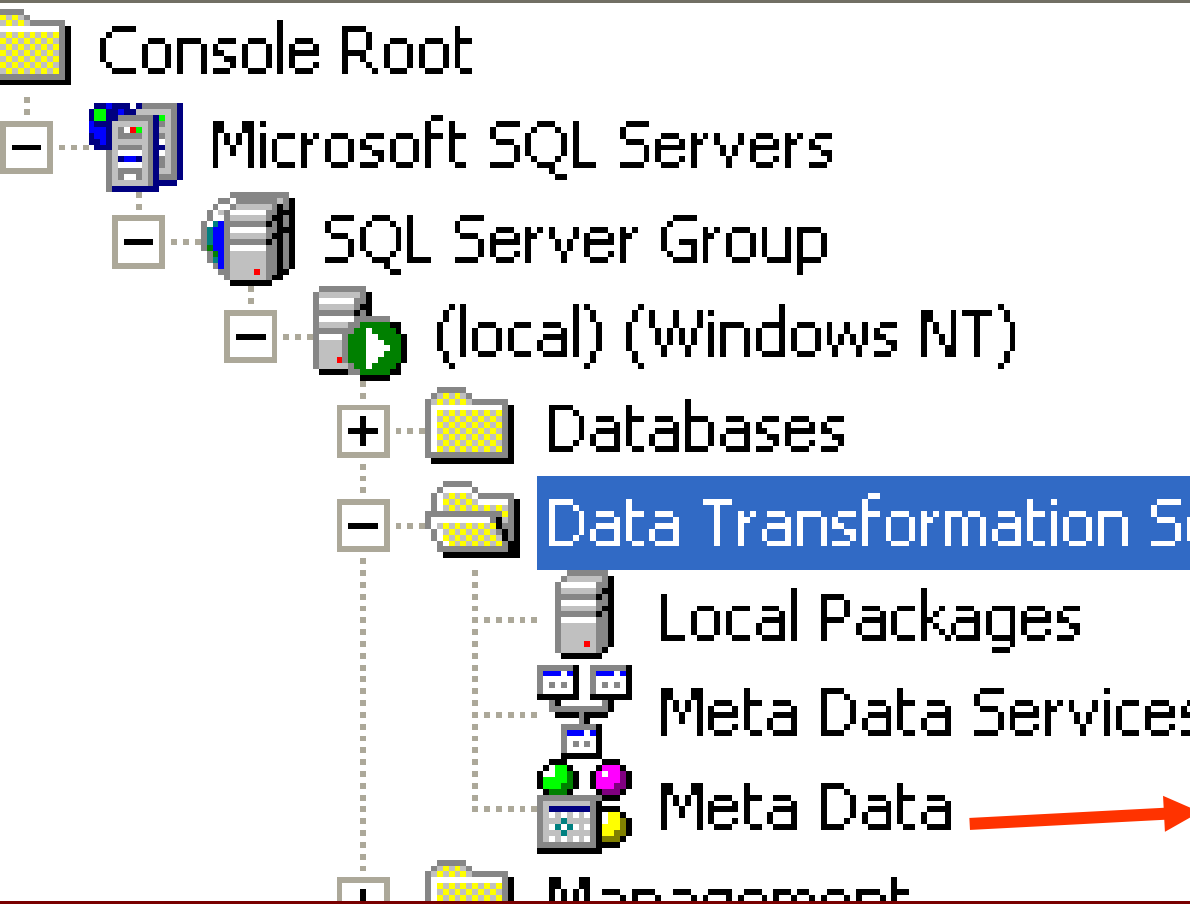
Contains Packages that are saved to this particular instance of SQL Server

1. DTS Package: Saving



Contains Packages that are saved to **Meta Data Services** of this instance of **SQL Server**. It maintains version information of each package saved to it.

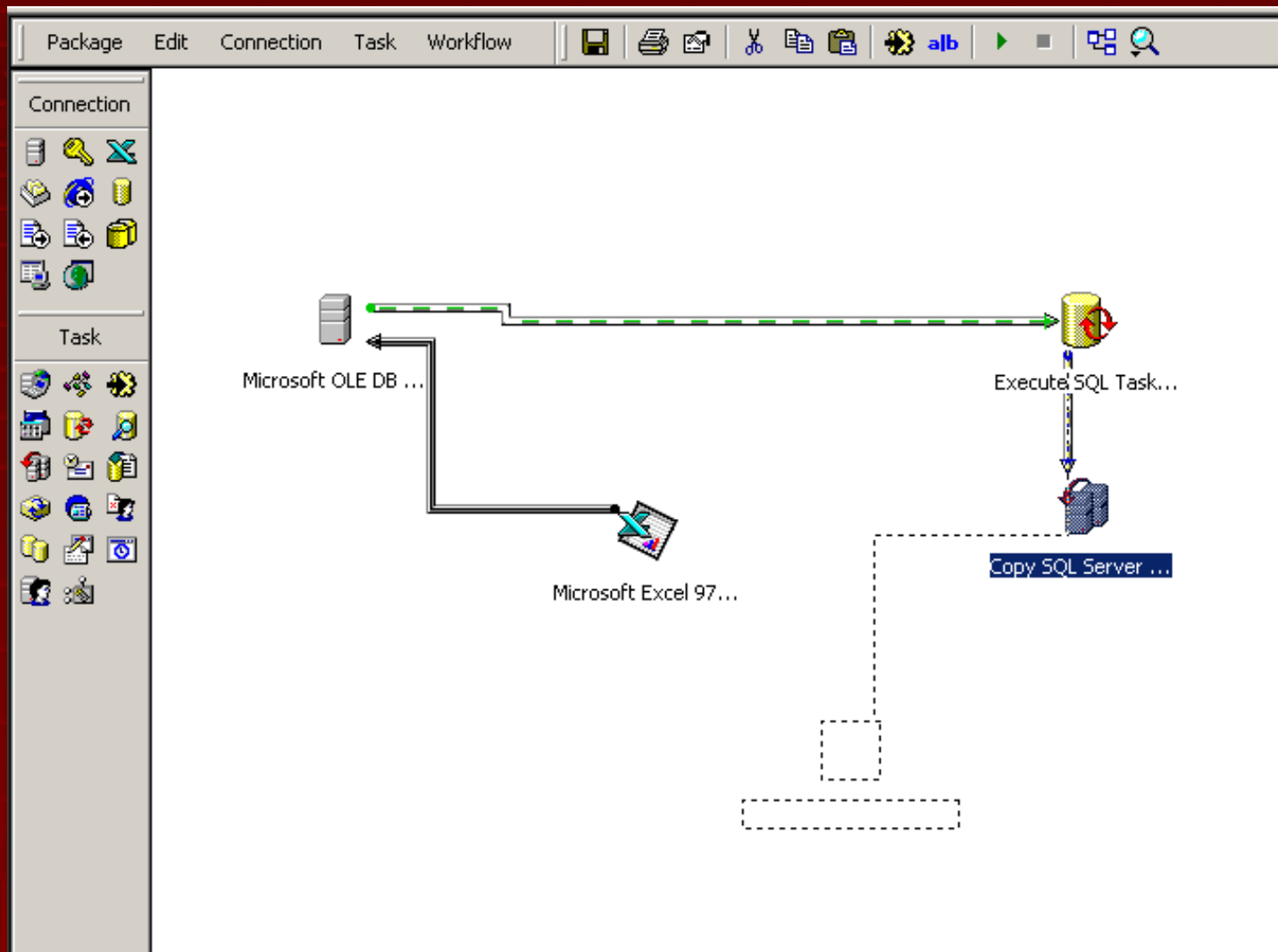
1. DTS Package: Saving



It is a repository of metadata information of databases scanned to **Meta Data Services Packages**. It also provides version tracking facility of Packages.

1. DTS Package: Operations

1. DTS Package: Editing



1. DTS Package: Password protection

- **Save dialog box allows to set passwords**

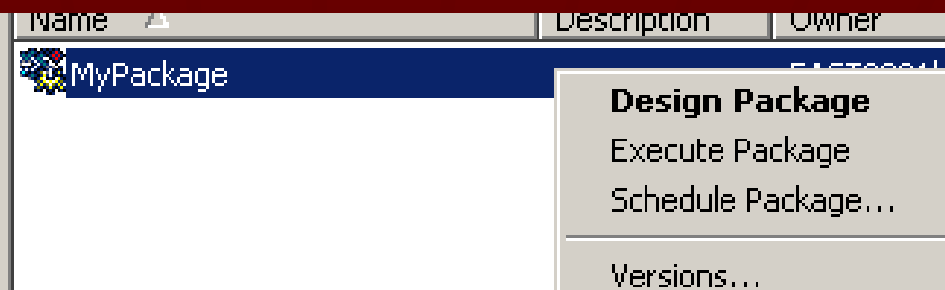
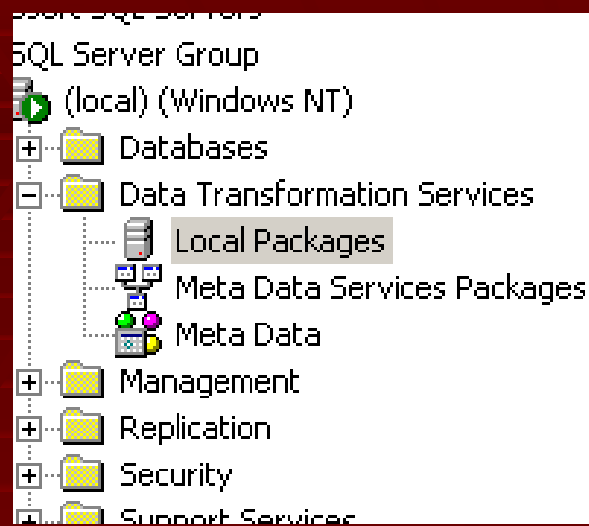


The screenshot shows a 'Save' dialog box for a DTS package. It has a light gray background and a white border. The fields are as follows:

Package name:	MyPackage		
Owner password:		User password:	
Location:	SQL Server 		

- **Owner password puts limits on both editing and execution of the package**

1. DTS Package: Scheduling/Execution



A screenshot of the 'Edit Recurring Job Schedule' dialog box. The 'Job name' is 'MyPackage 2'. The 'Occurs' section has 'Daily' selected. The frequency is 'Every 1 day(s)'. The 'Daily frequency' section has 'Occurs once at' selected with a time of '12:00:00 AM'. The 'Occurs every' section is unselected. The 'Starting at' is '12:00:00 AM' and the 'Ending at' is '11:59:59 PM'. The 'Duration' section has 'Start date' as '08/02/05' and 'End date' as '08/02/05', with 'No end date' selected. The dialog has 'OK', 'Cancel', and 'Help' buttons at the bottom.

Job name: MyPackage 2

Occurs: ☒ Daily ☐ Weekly ☐ Monthly

Every 1 day(s)

Daily frequency: ☒ Occurs once at: 12:00:00 AM ☐ Occurs every: 1 Hour(s) Starting at: 12:00:00 AM Ending at: 11:59:59 PM

Duration: Start date: 08/02/05 End date: 08/02/05 ☒ No end date

OK Cancel Help

1. DTS Package: Versioning

DTS Package Versions [X]

 Select the version of the DTS package that you want to edit/delete.

Package name: MyPackage

Versions:

Create date	Description
2005-07-30 14:33:30.787	
2005-07-30 14:30:33.500	

[Edit] [Delete] [Close] [Help]

2. DTS Tasks

2. DTS Tasks (Cont.)



Set of all possible tasks in designer, drag the required task in design area and set its properties

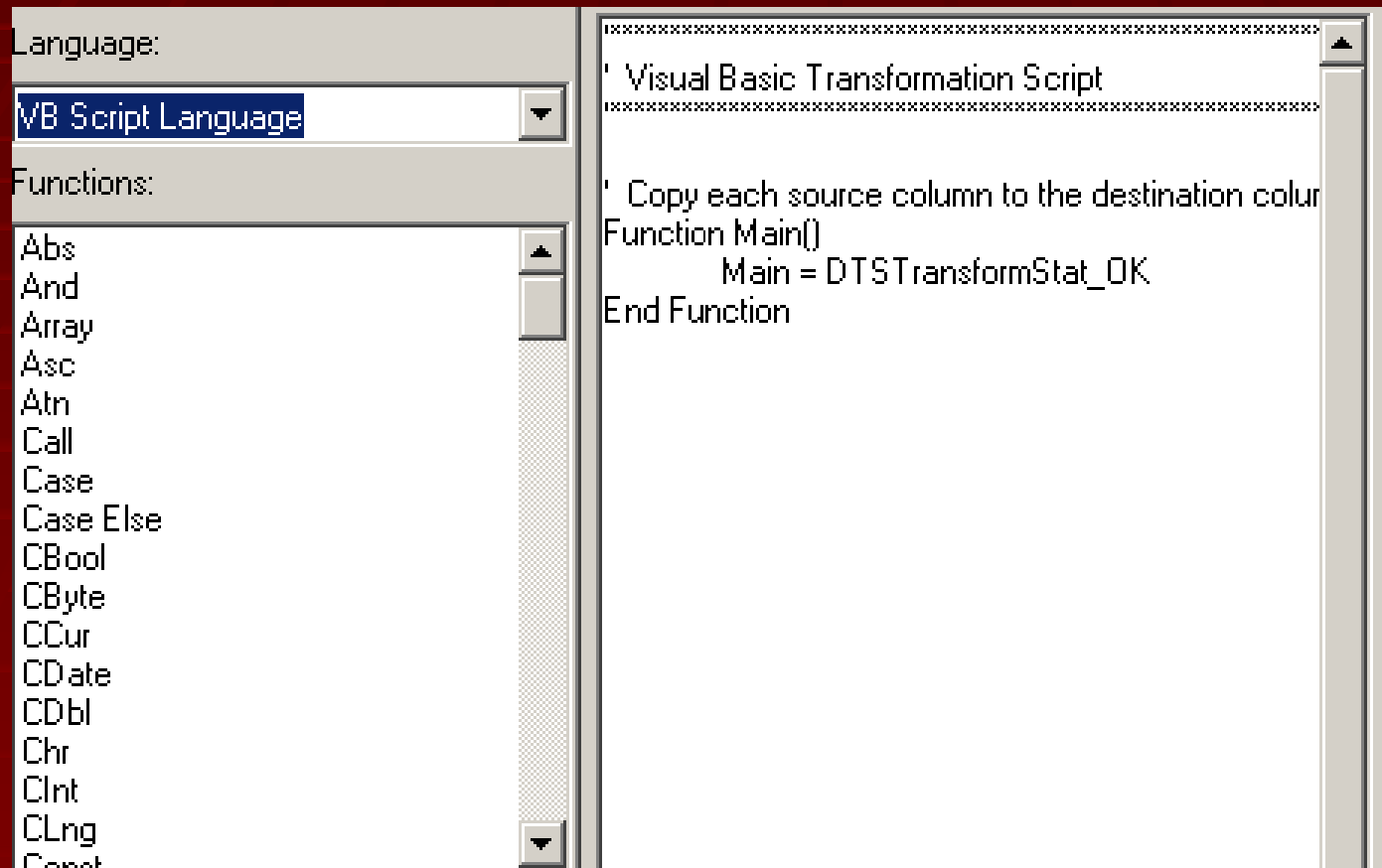
3. DTS Transformations

3. DTS Transformations: Available

3. DTS Transformations: Customized

3. DTS Transformations (Cont.)

■ ActiveX Script

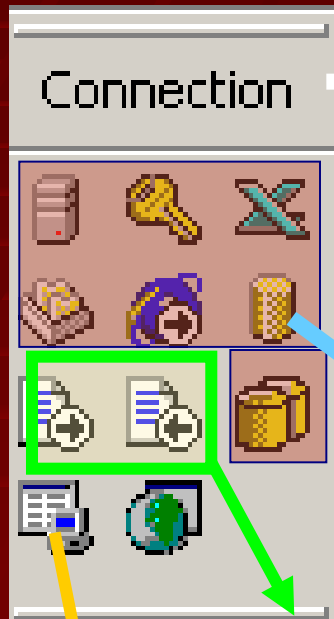


4. DTS Connections

4. DTS Connections (3 *SLIDES.*)

4. DTS Connections (Cont.)

Set of all possible connections in designer, drag the required connection in design area and set its properties

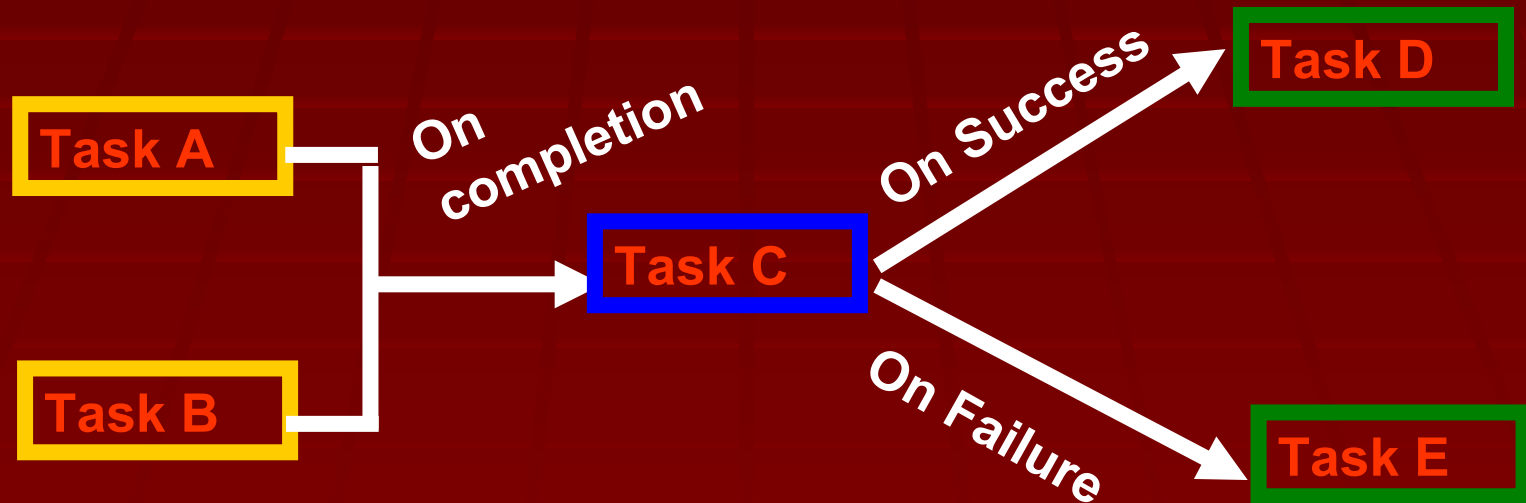


Data source connection

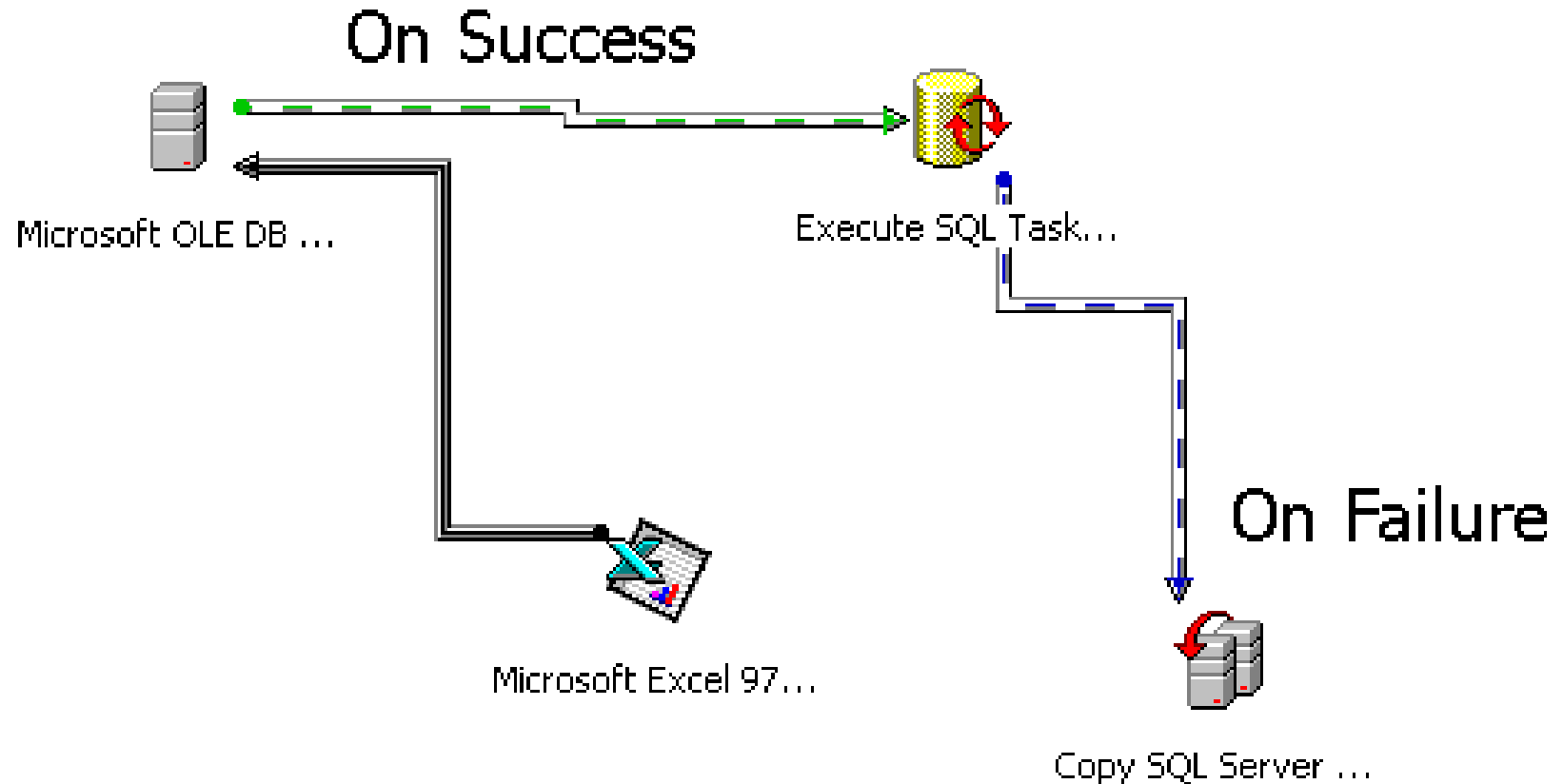
File connection

Data link connection

5. Package workflow



5. Package workflow (DESIGNER)



Virtual University of Pakistan

Data Warehousing

Lab Lect-2

Lab Data Set

Ahsan Abdullah

Assoc. Prof. & Head

Center for Agro-Informatics Research

www.nu.edu.pk/cairindex.asp

FAST National University of Computers & Emerging Sciences, Islamabad

Multi-Campus University

Degree Programs

Disciplines for BS

Disciplines for MS

The need

- **Head Office wants a central data repository for decision support i.e. a DWH**

Students Record Keeping & Mgmt.

Data from Lahore Campus

Data from Lahore Campus: Sample

Lhr_Student_104 - Notepad

File Edit Format View Help

```
SID, St_Name, Father_Name, Gender, Address, Date of Birth, Last Degree  
4000, Sheikh Rashid Sehti, Nazar Anwar Sehti, 0, house # 240 st. #  
4001, Bashir Shakeel Haqqie, Ch. Usman Haqqie, 0, House No. 555 S#  
4002, Shahzor Haseeb, Haseeb Jan, 0, Ho. # 102 St. No. 34 Bhatta Me  
4003, Zeba Ajab, Ajab Muzammil, 1, house no. 416 Street # 30 Shakri  
4004, Arsalan Baig, Aslam Baig, 0, ho. # 731 s no. 25 jalsa qha Gui
```

Lahore: Header of Student Table

- ***SID***
- ***St_Name***
- ***Father_Name***

Lahore: Header of Student Table

- ***Gender***
- ***Address***
- ***[Date of Birth]***
- ***[Reg Date]***

Lahore: Header of Student Table

- **[Reg Status]**
- **[Degree Status]**
- ***[Last Degree]***

Lahore: Header of Course Reg. Table

- ***SID***
- ***Degree***
- ***Semester***
- ***Course***
- ***Marks***
- ***Discipline***

Lahore: Facts About Data

Data from Karachi Campus

Data from Karachi Campus: Sample

</					

Karachi: Header of Student Table

- ***St_ID***
- ***Name***
- ***Father***
- ***DoB***
- ***M/F***
- ***DoReg***
- ***RStatus***
- ***DStatus***
- ***Address***
- ***Qualification***

Karachi: Header of Course Reg. Table

- ***SID:***
- ***Courses***
- ***Score***
- ***Sem***
- ***Disp***

Degree (BS/MS) is missing because separate books are maintained, but the issue is critical while loading data

Karachi: Facts About Data

Data from Islamabad Campus

Data from Islamabad Campus: Sample

Isb_MS_Student : Table

	Roll Num	Name	Father	Date of Birth	Education	Gender
▶		AAmer Zameer	Zameer Badar	8/24/1978	M.Sc	1
	1	Shamim Chohan	Rayyan Kamal	12/13/1977	M.Phil	0
	2	Aga Qudoos Sh	Wajid Qutab-ud	9/13/1979	MSc	1
	3	Eman Hafiz Ras	Hafiz Rasheed I	9/15/1979	BS	0
	4	Mushtaq Ghosh	Ghoshanwar Akbar	8/16/1979	MSc	1

Islamabad: Header of Student Table

- ***Roll Num***
- ***Name***
- ***Father***
- ***Reg Date***
- ***Reg Status***
- ***Degree Status***
- ***Date of Birth***
- ***Education***
- ***Gender***
- ***Address***

Islamabad: Header of Course Reg. Table

- ***Roll Num:***
- ***Course***
- ***Marks***
- ***Discipline***
- ***Session***

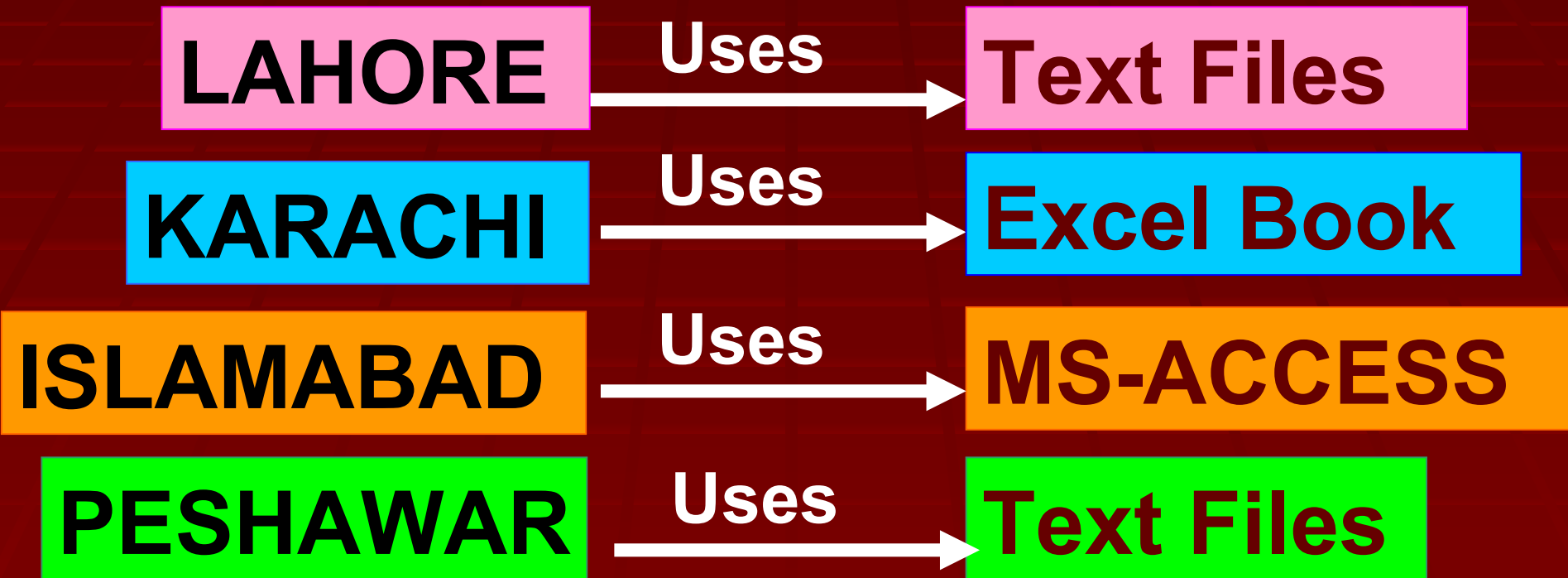
Degree (BS/MS) is missing, whereas same table contains records for both. Only way to differentiate is through discipline attribute.

Islamabad: Facts About Data

Exercise

Problems with Adhoc Approach

Problem-1: Non-Standard data sources



Problem-2: Non-standard attributes

	Identificaion	Gender (M/F)	Degree
Lahore	SID	Gender (0/1)	BS/MS
		4 th Col	
Karachi	St_ID	M/F	Separate books
		5 th Col	
Islamabad	Roll Num	Gender (1/0)	Discipline
		9 th Col	
Peshawar	Reg#		

Problem-3: Non Normalized database

Notepad: Issues

MS-Excel: Issues

MS-Access: Issues

Problem Statement

Data from Peshawar Campus

- Data at Peshawar campus is stored in Text files
- To store data regarding one complete batch 2 text files are used
 - Lhr_Student_batch (Student record)
 - Lhr_Detail_batch (Course Reg. record)
- 22 text files for 11 BS batches
- 8 text files for 4 MS batches

Data from Peshawar Campus: Sample

BS_P_97_Student - Notepad

File Edit Format View Help

```
Reg#,Name,Father,Address,Date of Birth,lastDeg
900,Aneesa Ibrahim,Ibrahim Rahid,Ho. No. 628 St. No. 39
901,Mustamsir Minhas,Fakhar Minhas,H No. 942 st. # 34
902,Rahid Sehti,Jabbar Shabi Sehti,house # 489 s no.30
903,Sundas Haq,Rayyan Umar Haq,h# 803 street no.25 Cha
904,Muhammad Hussain Minhas,Muhammad Hussain Minhas,h# 119 Street # 89
```

Peshawar: Header of Student Table

- ***Reg#:*** Student identity
- ***Name:*** Student name
- ***Father:*** Father name
- ***Address:*** Permanent address
- ***Date of Birth:*** Date of Birth
- ***lastDeg:*** Last degree achieved
- ***Reg Date:*** Date of Enrollment
- ***Reg Status:*** Status of Enrollment (A/T)
- ***Degree Status:*** Status of Degree (C/I)

Peshawar: Header of Course Reg. Table

- **Reg#:**
- **Courses:** Course code
- **Score:** Out of 100
- **Program:** CS/TC/SE/CE
- **Sem:** Fall/Spring
- **Year:** YYYY e.g. 1999

We need to identify semester session (fall04) through combination of Sem and Year

www.vuCtn.com

Good Luck

Regards : Zarva Chaudhary

Admin:

* Zarva Chaudhary *

* Chaudhary Moazzam *

* Laiba Maki *