CHAPTER 9

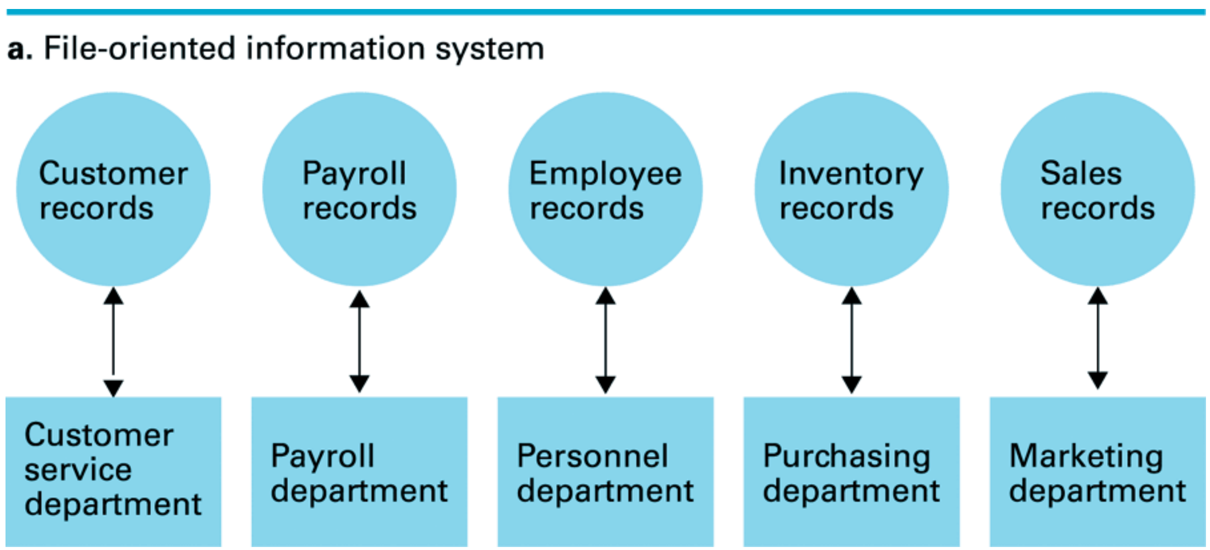
Database Structures

- (Large) integrated collections of data that can be accessed quickly

Lecture prepared and delivered by
Dr Syed Khaldoon Khurshid

9.1: Historical Perspective

- Originally: departments of large organizations stored all data separately in *flat files*

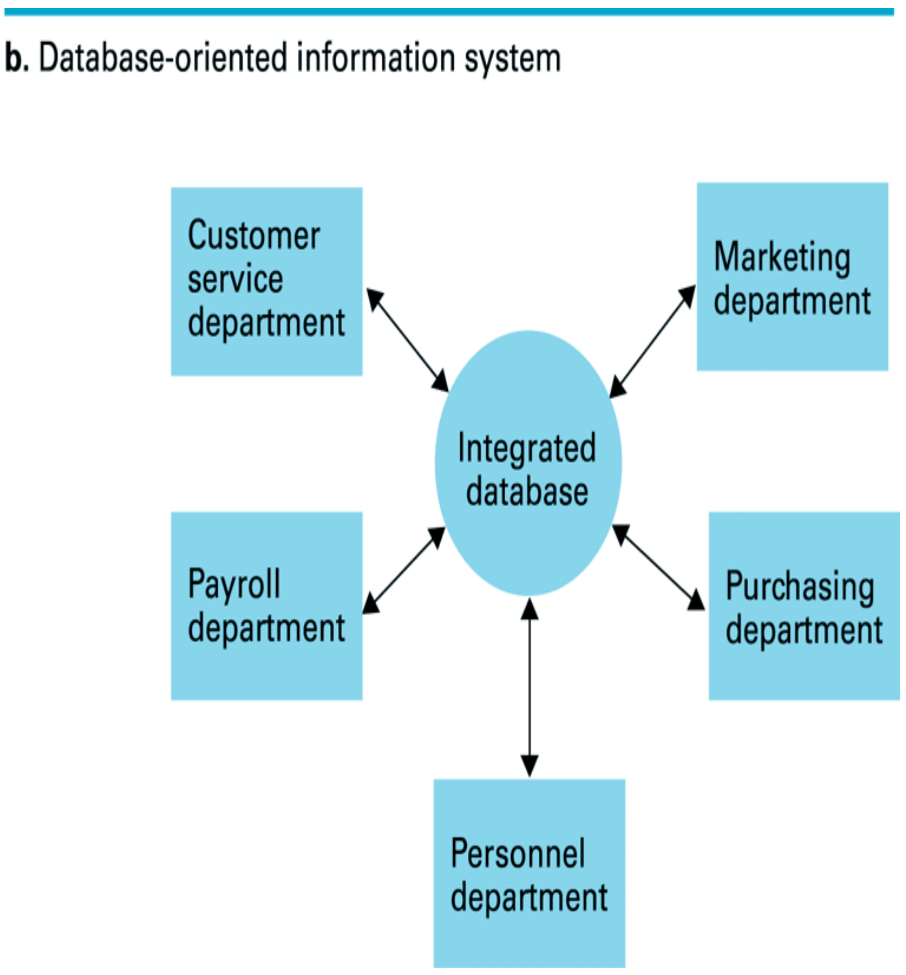


- Problems: redundancy & inconsistencies

9.1: Integrated Database System

- Better approach: integrate all data in a single system, to be accessed by all departments.
  - Schema and Subschema

Example: University student and faculty records



9.1: Disadvantages of Data Integration

- Control of access to sensitive data?
- Misinterpretation of integrated data?
- What about the **right** to hold/collect/interpret data?

File System

- Database is evolved from the File Systems.
- Understand the characteristics of the file system.
- Data management limitations by File system.
- Eliminations of the short comings of the file system by DBMS.

Basic Definitions

- Data: raw facts
  - Not processed yet to reveal their meaning
  - Constitute building blocks of information
  - For Examples:
    - Online Surveys
    - Online Data Entry Forms
    - Excel Sheets
    - Reports Forms
- Record keeping with the raw facts
  - Example: Students
    - » Pass 90%
    - » Fail 10 %
    - » Quick Answers

Basic Definitions (Continued...)

- Information: is produced by processing data and reveals meaning of data
  - Good, timely, relevant information key to decision making
  - Good decision making key to organizational survival
  - Example: Informed decisions to meet student grading record
  - Raw data: Storage, Processing and presentation
- Complex formatting: is required when working with complex data types such as sounds, videos ‘or’ images.

Basic Definitions (Continued...)

- Knowledge: the body of the information and facts about a specific subjects
- New Knowledge can be derived from Old Knowledge.

Basic Definitions (Continued...)

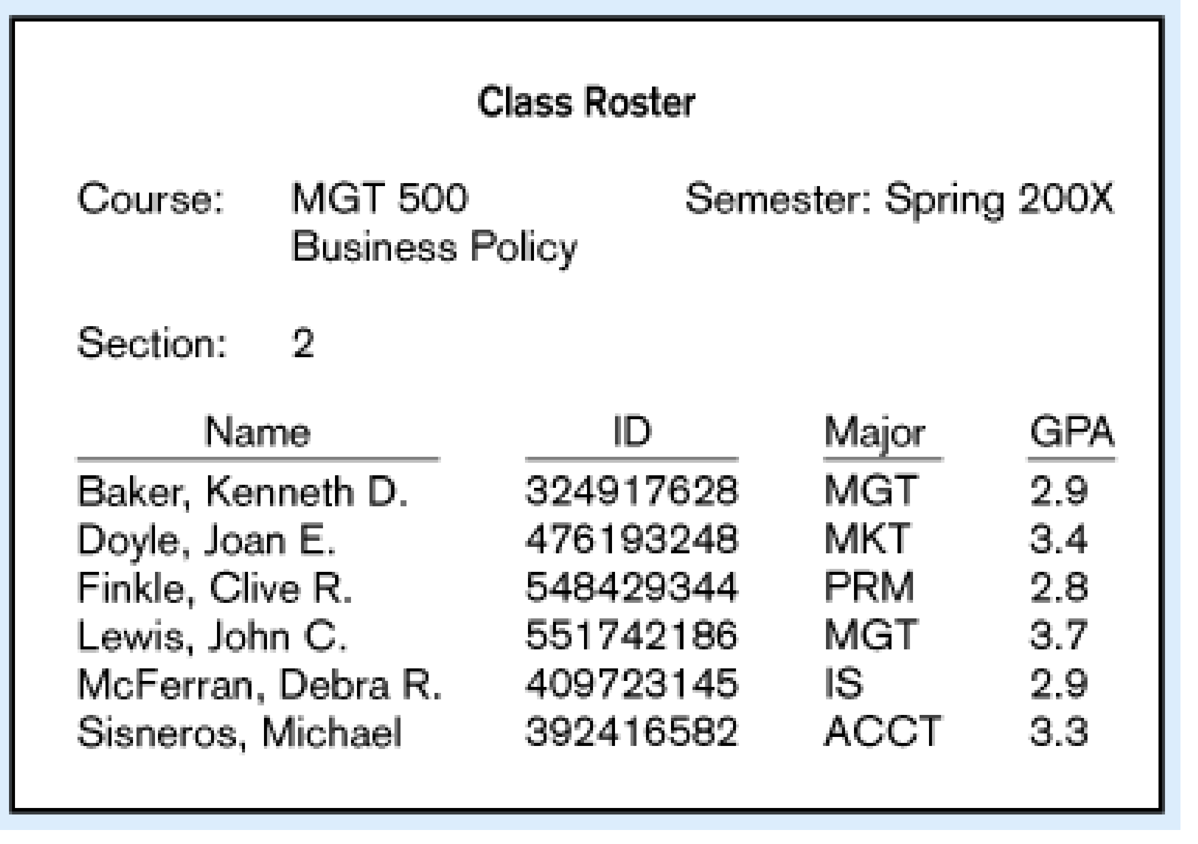
- Data Management is a discipline that focuses on the proper generation, storage and retrieval of data.
- Efficient Data Management requires computer DB.

Basic Definitions (Continued...)

- Database: shared, integrated computer structure housing:
  - End user data
  - Metadata
- Metadata provides a description of the data characteristics and set of relationships that link the data within the Database.
  - Structural metadata is data about the containers of data.
  - Descriptive metadata uses individual instances of application data or the data content.

An Example

- Converting data to information



Class Roster

Course: MGT 500 Business Policy

Semester: Spring 200X

Section: 2

Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

An Example (Cont'd)

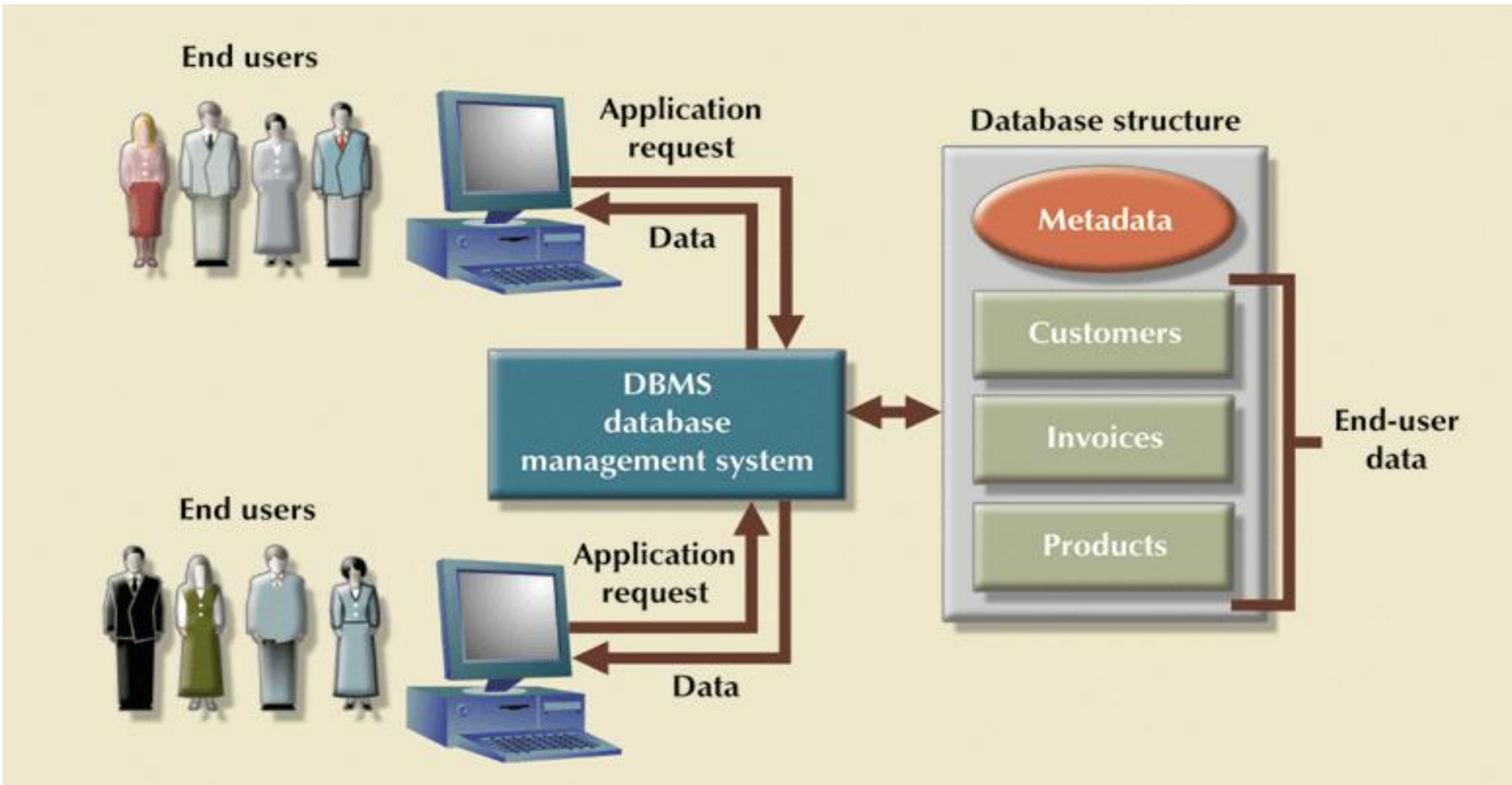
- Metadata

Data Item		Value			
Name	**Type**	**Length**	**Min**	**Max**	**Description**
Course	Alphanumeric	30			Course ID and name
Section	Integer	1	1	9	Section number
Semester	Alphanumeric	10			Semester and year
Name	Alphanumeric	30			Student name
ID	Integer	9			Student ID (SSN)
Major	Alphanumeric	4			Student major
GPA	Decimal	3	0.0	4.0	Student grade point average

What is a Database Management System (DBMS)

- A collection of programs that manages the database structure and controls access to the data stored in the database
  - Possible to share data among multiple applications or users
    - Example: bank and its ATM machines
  - Makes data management more efficient and effective
    - End users have better access to more and better-managed data
- DBMS hides much of the database's internal complexity from application program and End user

DBMS Manages Interaction



Advantages of the DBMS

- Improved data sharing
  – Shared among users and applications
- Better Data Integration
  – Different User’s views into single data Repository
    - Repository: can be a place where multiple DBs or files are located for distribution over the network.
- Minimized Data inconsistency
  – Different versions of the same data.
    - Example: Product ID and Product Number in different departments

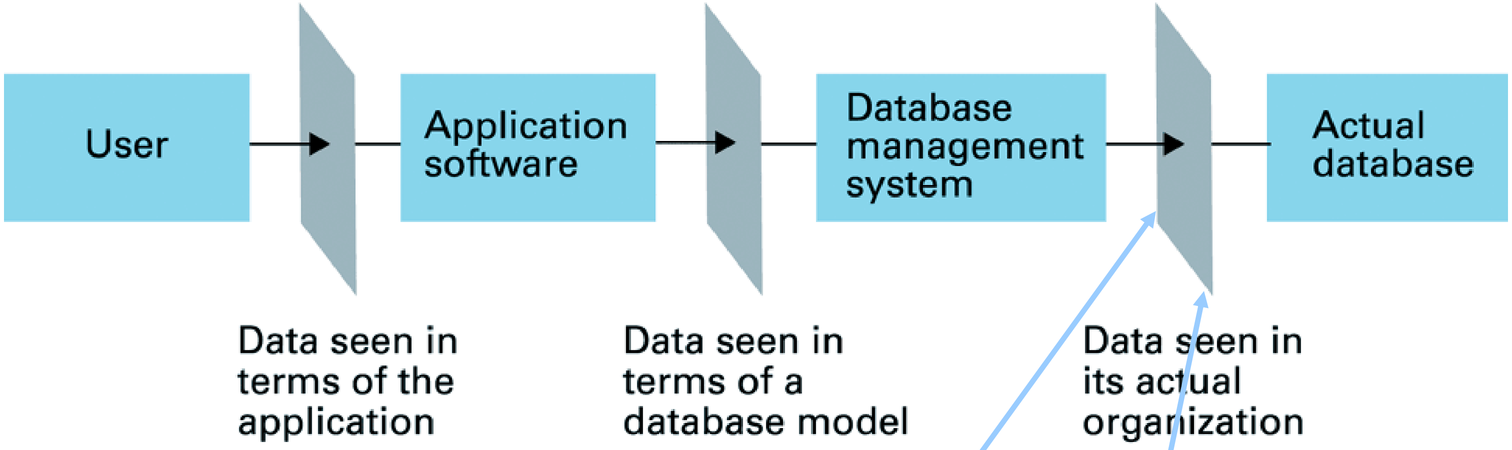
Advantages of the DBMS

- Improved Data access
  - Quick answers to the ad hoc queries
  - Query is a complete question: a specific request for data manipulation (read or update data)
  - DBMS sends back an Answer (Query result set) to the application
- Improved Decision Making
  - Better managed data and improved data access ->to better quality information ->better decisions
- Increased End User Productivity

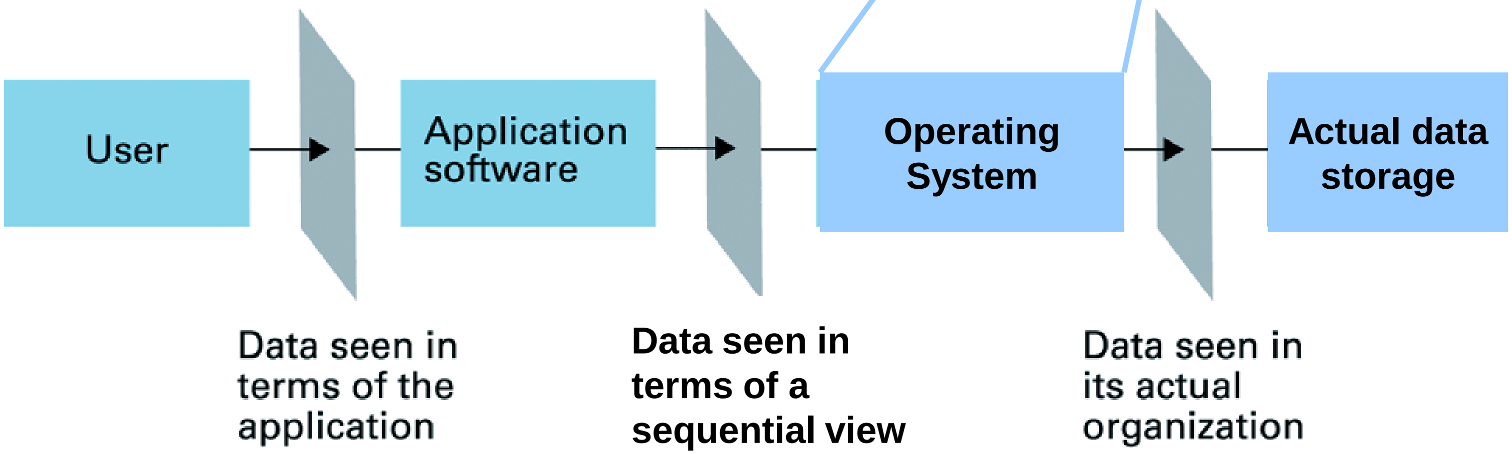
Types of the Databases

- Single User Database: Runs on a personal Computer
- Multiuser Database: less than 50 workgroup DB, more than 50 Enterprise DB
- Location wise:
  - Single site: Centralized DB
  - Several sites: Distributed DB
- Function wise: Operational/transactional/production
  - Time Sensitive information gathered
  - Support a company 's day to day operations

9.2: Conceptual Database Layers



- Compare:



9.3: The Relational Model

- Relational Model
  - shows data as being stored in rectangular tables, called *relations*, e.g.:

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

  - row in a relation is called '*tuple*'
  - column in a relation is called '*attribute*'

9.3: Issues of Relational Design

- So, *relations* make up a relational database…
- … but this is not so straightforward:

Empl Id	Name	Address	SSN	Job Id	Job Title	Skill Code	Dept	Start Date	Term Date
25X15	Joe E. Baker	33 Nowhere St.	111223333	F5	Floor manager	FM3	Sales	9-1-2001	9-30-2002
25X15	Joe E. Baker	33 Nowhere St.	111223333	D7	Dept. head	K2	Sales	10-1-2002	*
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999	F5	Floor manager	FM3	Sales	10-1-2001	*
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S25X	Secretary	T5	Personnel	3-1-1999	4-30-2001
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S25Z	Secretary	T6	Accounting	5-1-2001	*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

- Problem: more than one concept combined in single relation

9.3: Redesign by extraction of 3 concepts

EMPLOYEE relation

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

JOB relation

Job Id	Job Title	Skill Code	Dept
S25X	Secretary	T5	Personnel
S26Z	Secretary	T6	Accounting
F5	Floor manager	FM3	Sales
•	•	•	•
•	•	•	•
•	•	•	•

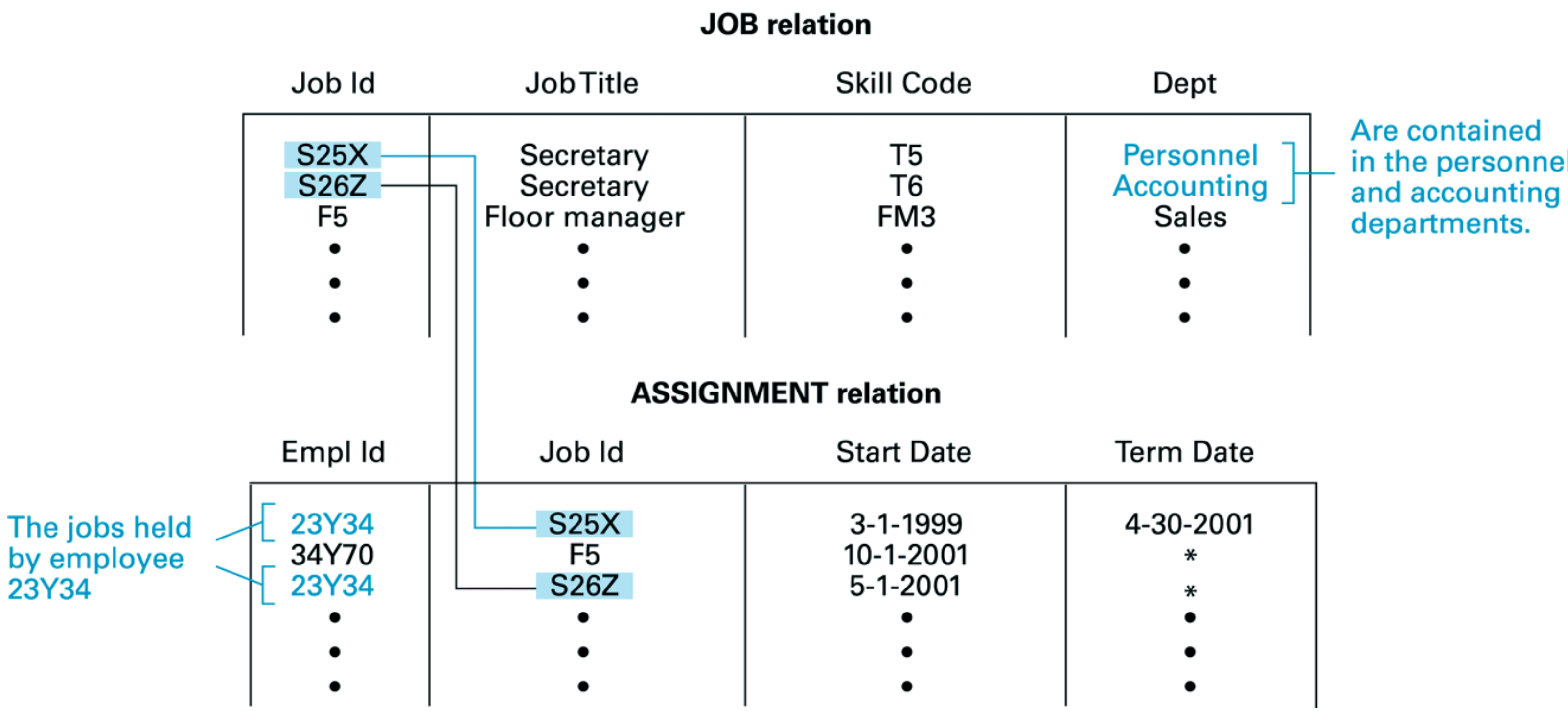
ASSIGNMENT relation

Empl Id	Job Id	Start Date	Term Date
23Y34	S25X	3-1-1999	4-30-2001
34Y70	F5	10-1-2001	*
25X15	S26Z	5-1-2001	*
•	•	•	•
•	•	•	•
•	•	•	•

Any information obtained by combining information from multiple relations

9.3: Example:

- Finding all departments in which employee 23Y34 has worked:



9.3: Relational Operations

- Extracting information from a relational database by way of *relational operations*
  - Most important ones:
    - (1) extract tuples (rows) : SELECT
    - (2) extract attributes (columns) : PROJECT
    - (3) combine relations : JOIN

- Such operations on relations produce other relations
  - so: they can be used in combination, to create complex database requests (or '*queries*')

9.3: The SELECT operation

EMPLOYEE relation

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
⋮	⋮	⋮	⋮

NEW ← SELECT from EMPLOYEE where EmplId = “34Y70”

NEW relation

Empl Id	Name	Address	SSN
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999

9.3: The JOIN operation

ASSIGNMENT relation

Empl Id	Job Id	Start Date	Term Date
23Y34	S25X	3-1-1999	4-30-2001
34Y70	F5	10-1-2001	*
25X15	S26Z	5-1-2001	*
•	•	•	•
•	•	•	•
•	•	•	•

JOB relation

Job Id	Job Title	Skill Code	Dept
S25X	Secretary	T5	Personnel
S26Z	Secretary	T6	Accounting
F5	Floor manager	FM3	Sales
•	•	•	•
•	•	•	•
•	•	•	•

NEW1 ← JOIN ASSIGNMENT and JOB where ASSIGNMENT. JobId = JOB.JobId

NEW1 relation

ASSIGNMENT Empl Id	ASSIGNMENT Job Id	ASSIGNMENT StartDate	ASSIGNMENT TermDate	JOB Job Id	JOB JobTitle	JOB SkillCode	JOB Dept
23Y34	S25X	3-1-1999	4-30-2001	S25X	Secretary	T5	Personnel
34Y70	F5	10-1-2001	*	F5	Floor manager	FM3	Sales
25X15	S26Z	5-1-2001	*	S26Z	Secretary	T6	Accounting
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

Chapter 9 - Database Structures: Conclusions

- Database Structures:
  - (large) integrated collections of data that can be accessed quickly
- Database Management System
  - provides high-level view of actual data storage (database model)
- Relational Model most often used
  - relational operations: SELECT, PROJECT, JOIN, …
  - high-level language for database access: SQL

Data Science Relation

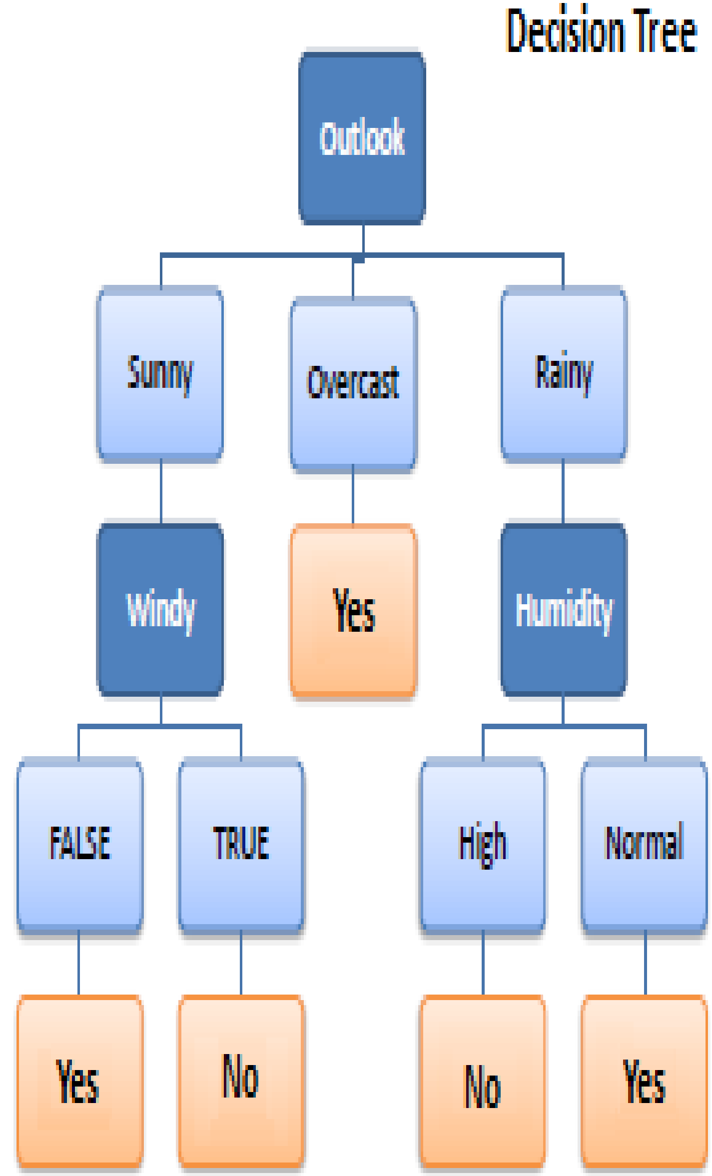
(with Example)

Decision Tree - Classification

- Data Science > Predicting the Future > Modeling > Classification > Decision Tree
- Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**.
- A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

Predictors | Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



Decision Tree to Decision Rules

A decision tree can easily be transformed to a set of rules by mapping from the root node to the leaf nodes one by one.

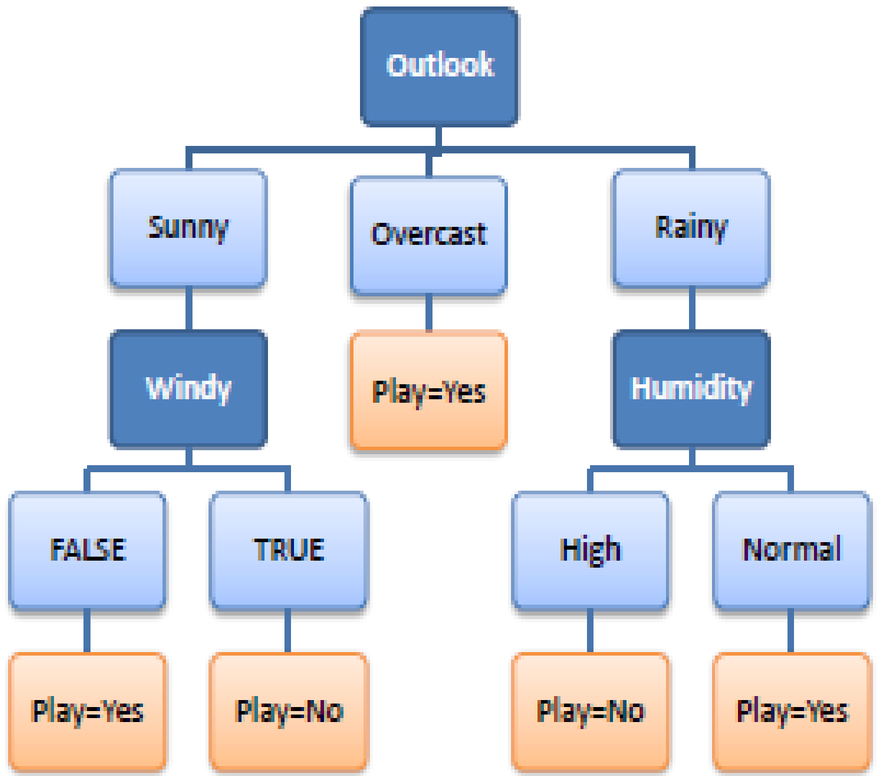
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes




Any
Questions