Course Overview

CSCI 5250

Database Design

Outline

- Definitions
- History of database processing
- Relational database
- Database life cycle

Definitions

- Database – organized collection of logically related data
  - Designed to (hopefully) meet information needs of organization
  - Designed to be shared by many users

- Database management system (DBMS) – software which supports storing, retrieving, and updating data in a database
  - SQL Server, Oracle, MySQL, PostgreSQL

Definitions

- OLTP – Online Transaction Processing – Transaction oriented application/database

- OLAP– Online Analytical Processing – System used to respond to multi-dimensional analytical queries.
  - ❑ Data cube
  - ❑ BI
  - ❑ Data Warehouse

Data Management Challenges

1. Data accuracy
2. Data security
3. Data organization
4. Data access

Data Management Challenges

1. Data accuracy
   - ❑ Ensuring that data is accurate

   - ❑ Data inaccuracies may occur while:
     - ▪ Capturing the data (e.g., unreliable data source, human error)
     - ▪ Manipulating the data (e.g., using buggy software to process the data)

   - ❑ What to do:
     - ▪ Verify data sources
     - ▪ Be careful with data entry and manipulation
     - ▪ Use DBMS functionality to help with accuracy

Data Management Challenges

2. Data security
   - Keep data safe
   - Ensure only authorized users have access to the specific data needed for their work
   - Protecting data from
     - Theft
     - Destruction (malicious or accidental)
     - Changes (malicious or accidental)

Data Management Challenges

2. Data security
   - Physical security – protecting data against physical corruption or loss
     - Protect the server (e.g., locked room)
     - Use fault-tolerant drives (e.g., RAID)
     - Duplicate the data
       - Mirror the data
       - Regular backups
   - Access security – controlling user and application access to data via authentication and authorization methods
     - Usernames / passwords
     - Privileges (e.g., read/write permissions)

Data Management Challenges

3. Data organization
   - ❑ How data should be structured for use by different applications and user groups
   - ❑ Goal is to minimize data redundancy and ensure quick access to data
     - ▪ Problems with redundant data:
       - ▪ Extra storage space
       - ▪ Data updates more difficult
       - ▪ Potential for data integrity issues
     - ▪ Data standards – help ensure the right data is entered in the same way for every data object

Data Redundancy

Sales file

Customer Number	**Customer Name**	**Customer Address**
2746795	John Jones	123 Elm Street

Accounts Receivable file

Customer Number	**Customer Name**	**Customer Address**
2746795	John Jones	123 Elm Street

Credit file

Customer Number	**Customer Name**	**Customer Address**
2746795	John Jones	123 Elm Street

Sales file

Customer Number	**Customer Name**	**Customer Address**
2746795	John Jones	456 Oak Street

Accounts Receivable file

Customer Number	**Customer Name**	**Customer Address**
2746795	John Jones	456 Oak Street

Credit file

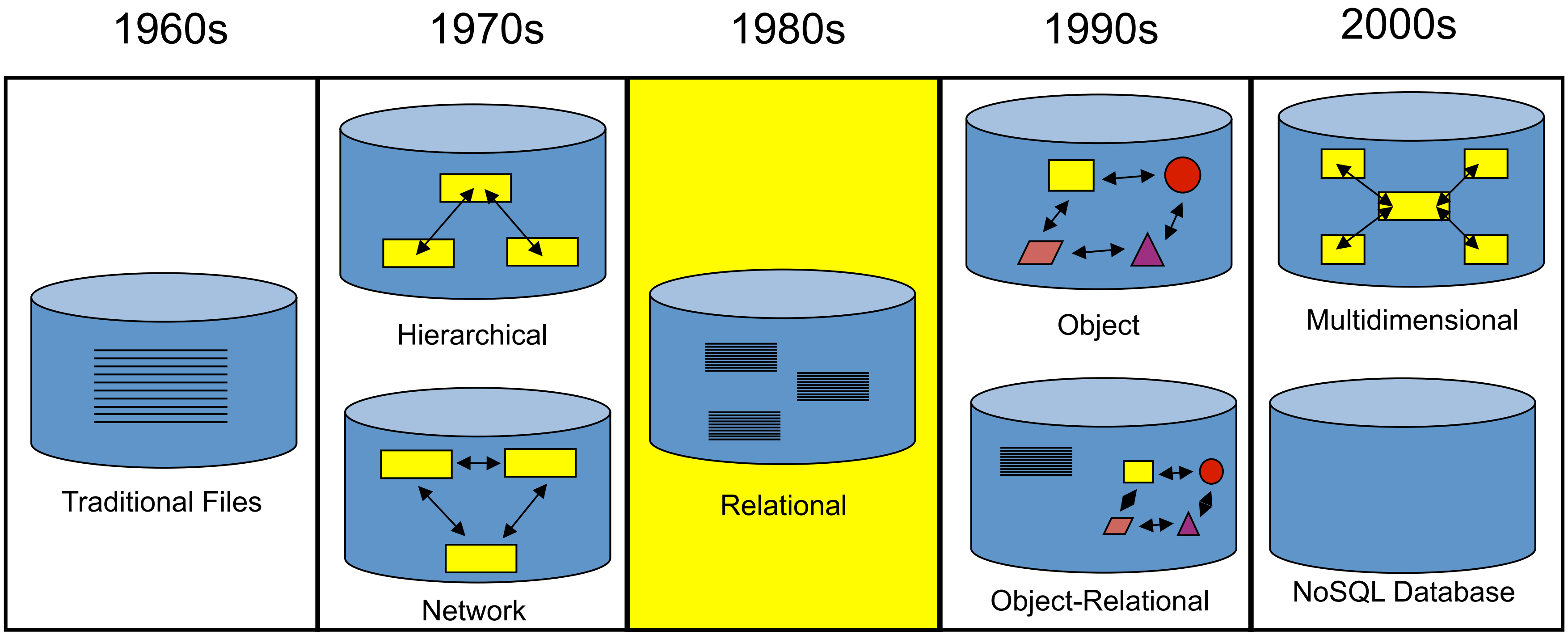
Customer Number	**Customer Name**	**Customer Address**
2746795	John Jones	123 Elm Street

Redundant Data → Inconsistent Data

Data Management Challenges

4. Data access
   - Controlling who accesses which data and how
   - Access may be direct (via queries) or indirect (via applications)
     - Queries are used to retrieve data, insert new data, or manipulate existing data
   - User roles and logins can be used to restrict access
   - Parallelism

History of Database Models

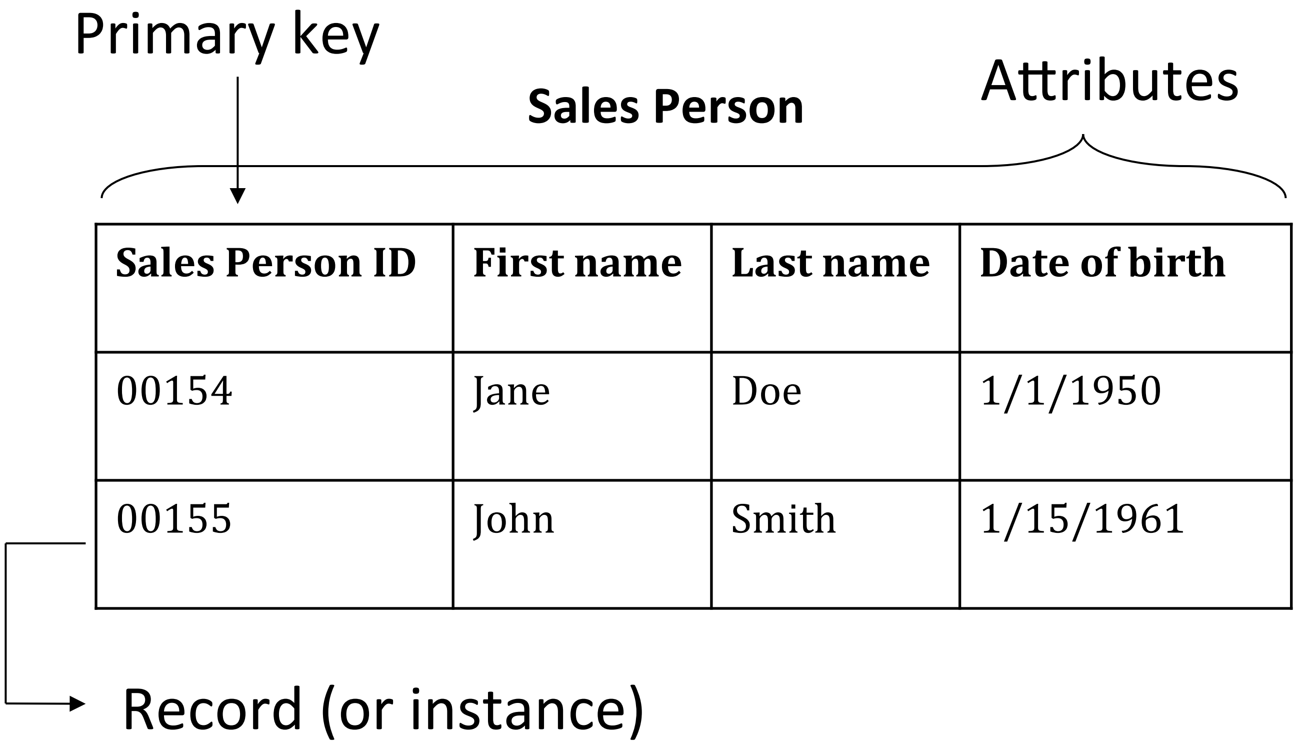


Relational Database

- One or more tables represent an entity
  - Each column represents an attribute (or field)
  - Each row (or tuple) represents a data instance (or record)
- Primary key – One (or more) attribute(s) that uniquely identifies each instance
- Tables are connected through logical relationships
  - Relationships are implemented using foreign keys

- Entity:
  - Person, thing, place, event (i.e., noun)
- Attributes of an entity:
  - Facts, characteristics, properties of entities

Relational Database



Primary key

Sales Person

Attributes

Sales Person ID	First name	Last name	Date of birth
00154	Jane	Doe	1/1/1950
00155	John	Smith	1/15/1961

Record (or instance)

Relational Database

Sales Person

Sales Person ID	First name	Last name	Date of birth
00154	Jane	Doe	1/1/1950
00155	John	Smith	1/15/1961

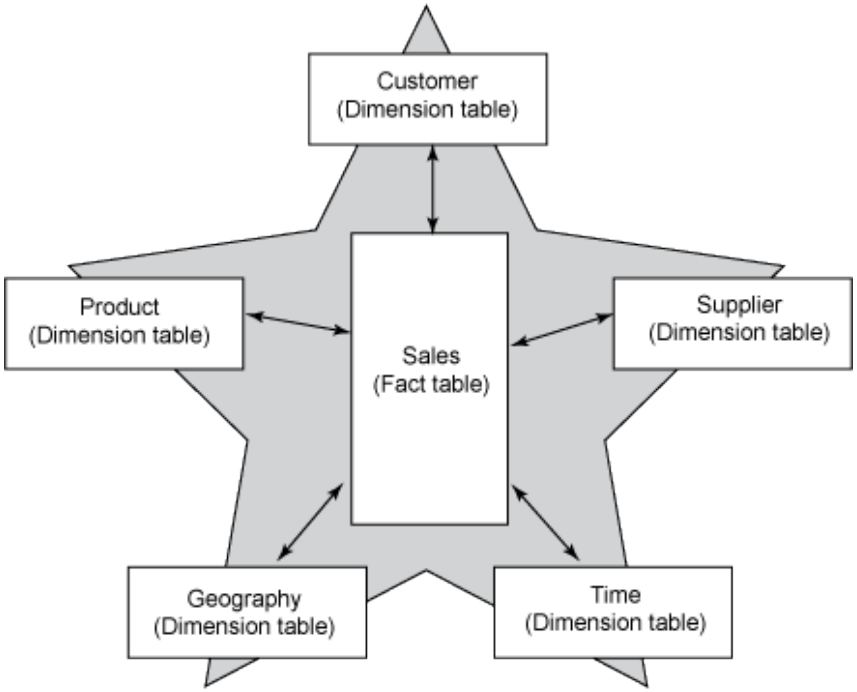
Order

Foreign Key

Order ID	Customer ID	Order Total	Sales Person ID
99384	01232	$5,145.09	00154
99578	15778	$9,543.23	00155

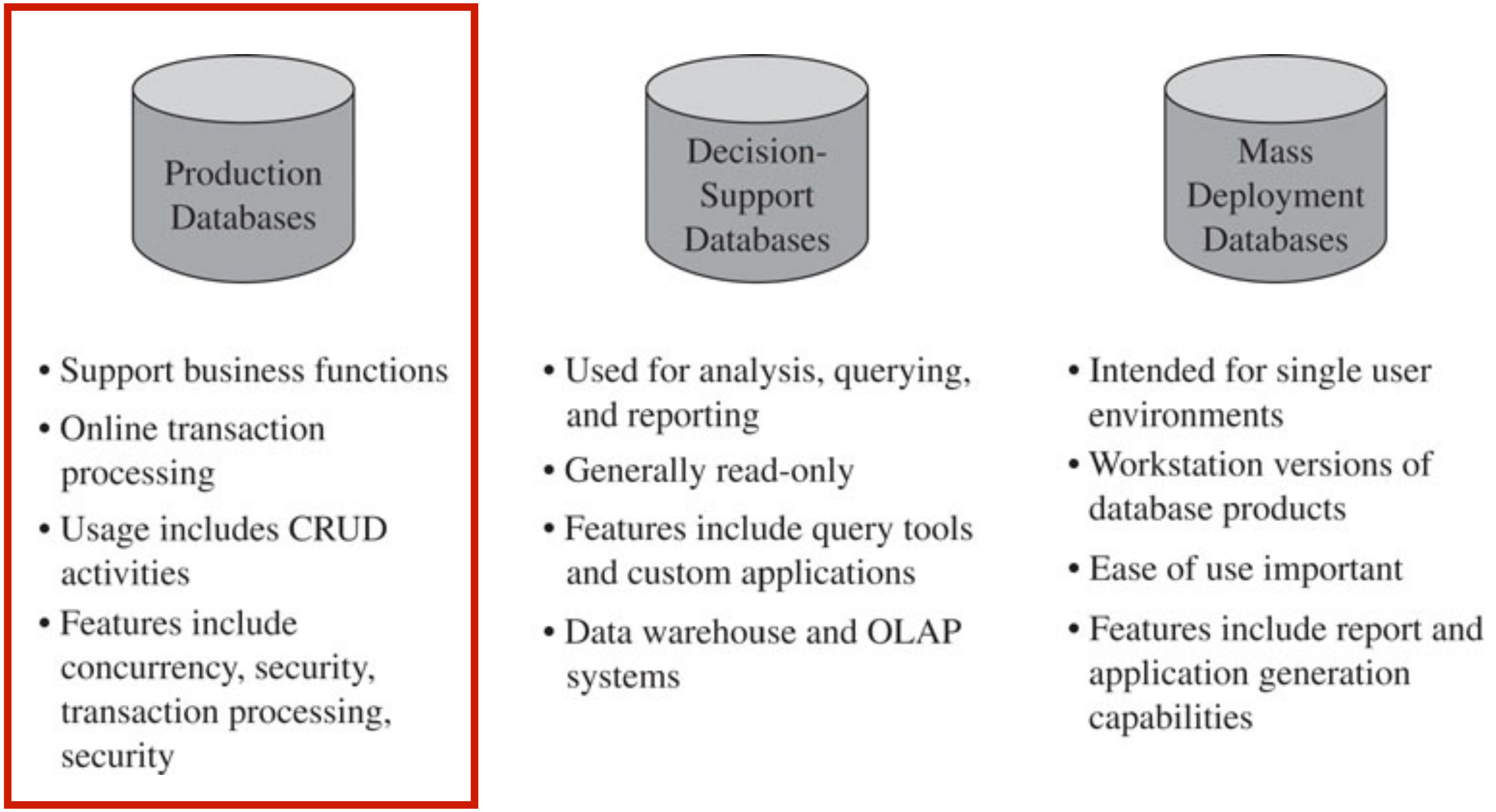
Multidimensional Database

- Data warehousing
- Organizes data using star schema, snowflakes schema, etc.
- Designed for quick access and computations on large data volumes



Types of Databases

- How are they used?



Production Databases

- Support business functions
- Online transaction processing
- Usage includes CRUD activities
- Features include concurrency, security, transaction processing, security

Transactional Databases

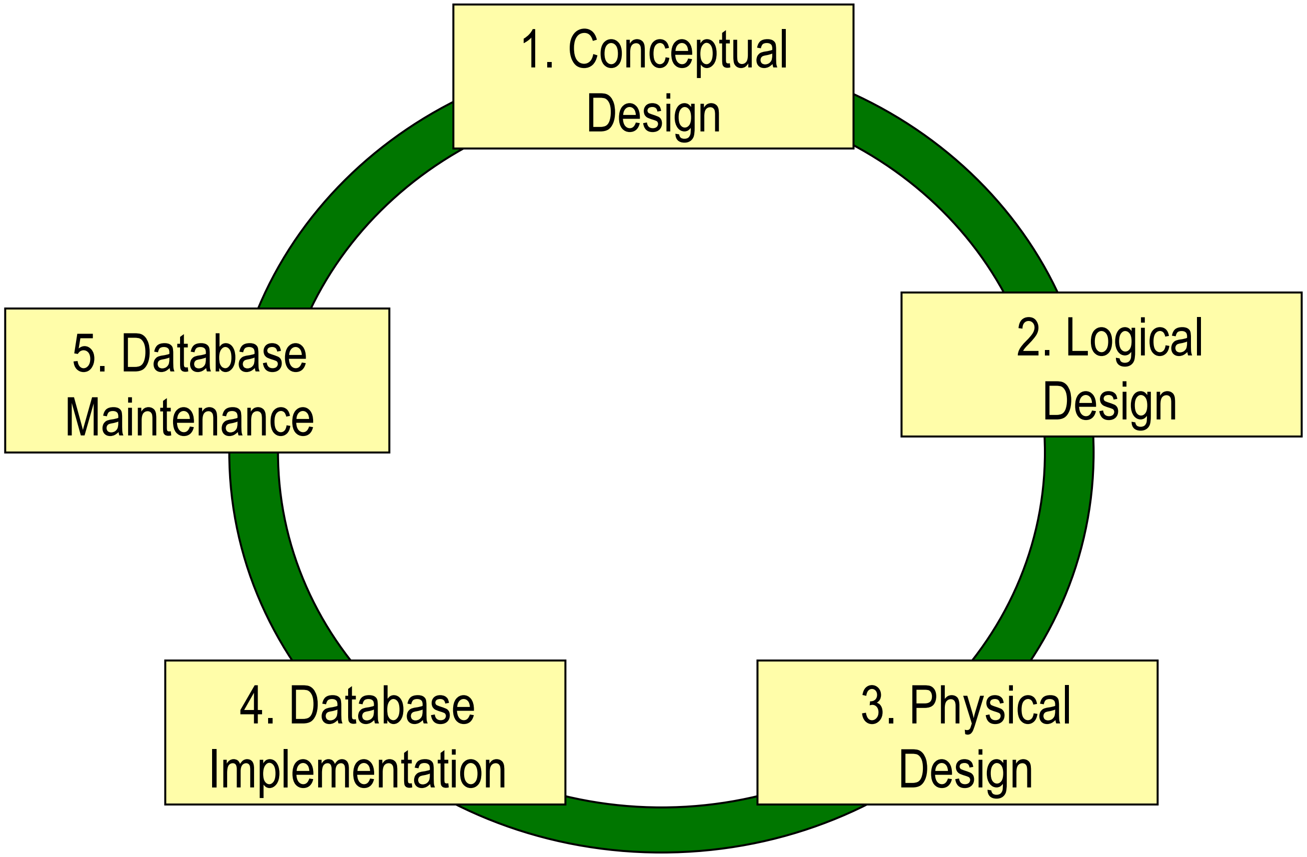
Decision-Support Databases

- Used for analysis, querying, and reporting
- Generally read-only
- Features include query tools and custom applications
- Data warehouse and OLAP systems

Mass Deployment Databases

- Intended for single user environments
- Workstation versions of database products
- Ease of use important
- Features include report and application generation capabilities

Database Life Cycle



DBLC Phases

1. Conceptual design:
    - ❑ Interview users, or look at problem description to identify:
        - Entities
        - Attributes
        - Relationships
    - ❑ Output: Entity-relationship diagram (ERD)

DBLC Phases

2. Logical design:
   - Map ERD to relational tables
   - Normalize tables as needed
   - Output: Relational tables

DBLC Phases

3. Physical design:
   - ❑ Specify data types, referential integrity, etc.
   - ❑ Output: Implementable tables with complete meta-data

DBLC Phases

4. Database implementation:
   - ❑ Select storage mechanisms, access methods, data update strategy, database distribution strategy (if needed)
   - ❑ Create database tables with fields and primary keys
   - ❑ Specify relationships between tables via foreign keys
   - ❑ Load/import data into the above tables
   - ❑ Create user documentation
   - ❑ Output: Fully functional relational database and user documentation

DBLC Phases

5. Database maintenance:
   - ❑ Modify data structure as needed (indefinitely)
   - ❑ Modify application programs as needed
   - ❑ Monitor database performance in real time
   - ❑ Perform routine database upgrades

Next Lecture

- Data modeling
- Project