**Fundamentals of Database Systems**

**Chapter 1: Introduction**

**Overview**

◉ The world is increasingly **driven by data…**

◉ This class teaches the basics of how to use & manage data.

**Why is this class important**

◉ Database is needed in almost any software application

◉ Data-intensive tools and applications are becoming increasingly popular

◉ Data analytics, business intelligence and data science

**Basic Definitions**

◉ **Database:** A collection of related data

◉ **Data:** Known facts that can be recorded and have an implicit meaning

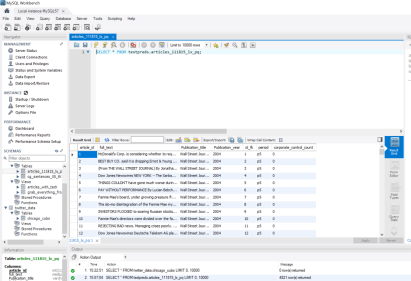
◉ **Database Management System (DBMS):** A software package/ system to store and manage databases

**Basic Definitions**

◉ For Example: MySQL, SQL-Server, Oracle, Microsoft Access, etc. are popular commercial DBMS used in different applications. DBMS allows users the following tasks:

• **Data Definition:** It helps in creation, modification and removal of definitions that define the organization of data in database. • **Data Updation:** It helps in insertion, modification and deletion of the actual data in the database.

• **Data Retrieval:** It helps in retrieval of data from the database which can be used by applications for various purposes.

**Database Management System (DBMS)**

**Database Management System (DBMS)**

◉ **Typical functionalities include:**

1. Define a database (tables, datatypes, constraints, and structures) 2. Retrieve from/query a database

3. Update a database (insert, modify or delete)

4. Keeping the data valid

5. Allowing multiple users and applications to access and share the database.

◉ Other functionalities include preventing unauthorized access and displaying and visualizing the data.

**Database**

◉ **Models a real-world enterprise:**

◉ **1. Entities:** Entities are specific things or objects in the mini-world that are represented in the database

◉ **2. Attributes:** Properties used to describe an entity

◉ **3. Relationships:** Relates two or more distinct entities with a specific meaning

**Query and Update**

◉ **1. Query:** Retrieve from tables **2. Update:** Change in tables

**Example 1:**

◉ **Course management system:**

◉ Students, courses, sections, and professors

◉ Professors teach sections

◉ Students register in sections.

◉ Courses have sections

◉ Students have names, students IDs, phone numbers..

**Example 1:**

◉ **Course management system:**

◉ Students, courses, sections, and professors 

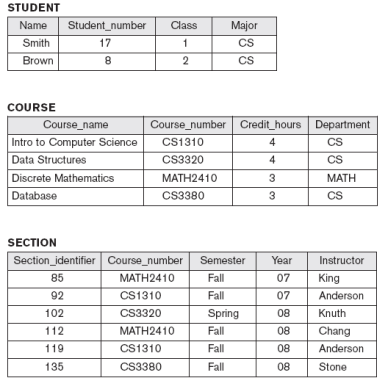
***Entities***

◉ Professors teach sections 

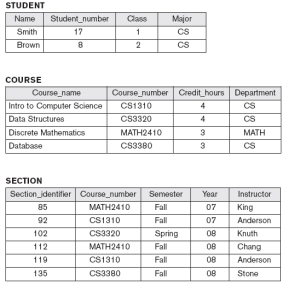
◉ Students register in sections. ***Relationships*** ◉ Courses have sections 

◉ Students have names, students IDs, phone numbers.. 

***Attributes***

**Example 1**

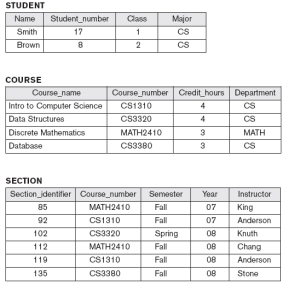
**Example 1:**

◉ **Query examples:** 

1. Return all students

2. Classes offered in Spring of 08 3. Courses taught in the CS department

**Example 1:**

◉ **Update examples:** 

1. Change Smith major to CE 2. Add new math course

3. Change instructor for CS1310 from Anderson to Kruth

**Exercise 1:**

◉ **Management system for final year projects** ◉ What are the entities in the system? ◉ What are the attributes for each entity? ◉ What are the relationships

◉ Examples of queries?

◉ Examples of updates?

**Exercise 2:**

◉ **Management system for car repair shop** ◉ What are the entities in the system? ◉ What are the attributes for each entity? ◉ What are the relationships

◉ Examples of queries?

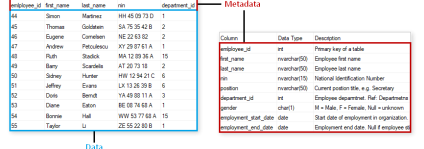
◉ Examples of updates?

***Part 2: Important Concepts***

**Self-describing nature of a database system**

◉ The DBMS has the information related to the databases. ◉ Therefore, you can use the DBMS to find all the information about the database without having to use external resources.

◉ **Meta-data:** data about data. Information describe the tables in the database.



**Data Independence**

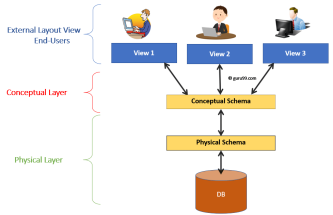
◉ *Very important: One of the main reasons for using DBMSs* ◉ Applications don’t care about how the data is structured and stored

◉ **Insulation between programs and data:** You can change how the data is stored and organized without having to change the programs the access the data

**Support of multiple views of the data**

◉ Different users view the database in various ways. ◉ The manager of the car repair shop view different information

than the car mechanic.



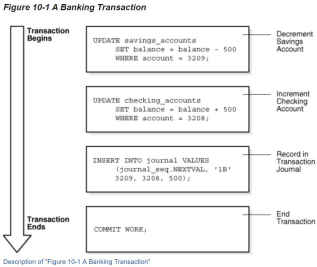
**Sharing of data and multi-user transaction processing**

◉ Allowing a number of users to retrieve from and to update the database at the same time.

◉ Concurrency control within the DBMS guarantees that each transaction is correctly executed or aborted

◉ Transaction: Unit of work (lines of code)

**Transactions**

◉ Unit of work (lines of code)

**Transactions**

◉ Unit of work (lines of code) 

◉ **Two properties:**

**1. Isolation property:** Each transaction is

independent from the other.

**2. Atomicity property:** Either execute all the

lines successfully or none

**Actors on the Scene**

◉ Important people

**1. Database Administrator (DBA):** Manages the database **2. Database Designer:** Specify the data and structures **3. End User:** Users who access the database

**4. Systems analysts:** Determine the requirements for end users

**5. Application programmers:** Write code to allow end user to access database based on the requirements from systems analysts.

**Actors on the Scene**

◉ For the car repair shop example:

**1. Database Administrator (DBA):** Make sure DB is running with no issues. Grant access to database if needed.

**2. Database Designer:** Specify the tables and the attributes needed for each table

**3. End User:** Person at the front desk. Maybe mechanics , Admin **4. Systems analysts:** Specify what the end user needs to see **5. Application programmers:** Write the code for the end user

**Some of the Advantages of using the database approach**

◉ **Control redundancy:**

- No need to repeat the same information

- Instead of saving a customer info every single time, save once and reference

◉ **Restrict unauthorized access:**

- Ensure access to DB is limited to authorized users

◉ **Efficient query processing:**

- Faster to search and find what you’re looking for.

**Some of the Advantages of using the database approach**

◉ **Provide backup and recovery:**

- Take copies of the database and use if necessary

◉ **Provide multiple user interfaces**

- Different ways of accessing and viewing the data

◉ **Enforcing integrity constraints:**

- For example: no two citizens can have the same national ID

**Schema and State**

◉ **Database Schema:** The description of a database. Includes descriptions of the database structure, data types, and the constraints on the database.

◉ **Schema Diagram:** An illustrative display of (most aspects of) a database schema.

◉ **Database State:** The actual data stored in a database at a *particular moment*in time. This includes the collection of all the data in the database. ◉ Also called database instance

**Schema and State**

◉ **Database State:** Refers to the content of a database at a moment in time.

◉ **Initial Database State:** Refers to the database state when it is initially loaded into the system.

◉ **Valid State:** A state that satisfies the structure and constraints of the database.

**Schema and State: Distinction**

• The **database schema** changes very infrequently.

• The **database state** changes every time the database is updated.

**Database Schema: Example**

****

**Database State: Example**

****