Fundamentals of Database Systems

Chapter 1: Introduction

eSPIN University

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

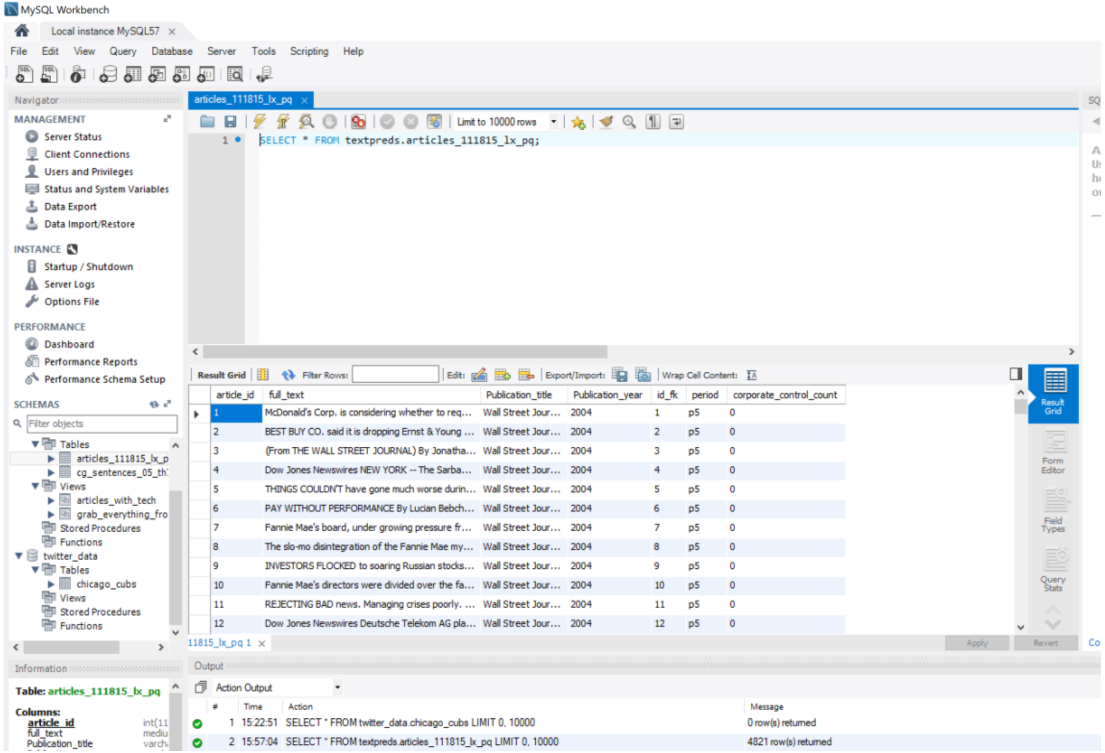
Basic Definitions

- **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)



```
1  SELECT * FROM textpreds.articles_111815_lx_pq;
```

article_id	full_text	Publication_title	Publication_year	id_fk	period	corporate_control_count
1	McDonald's Corp. is considering whether to req...	Wall Street Jour...	2004	1	p5	0
2	BEST BUY CO. said it is dropping Ernst & Young ...	Wall Street Jour...	2004	2	p5	0
3	(From THE WALL STREET JOURNAL) By Jonatha...	Wall Street Jour...	2004	3	p5	0
4	Dow Jones Newswires NEW YORK -- The Sarba...	Wall Street Jour...	2004	4	p5	0
5	THINGS COULDN'T have gone much worse durin...	Wall Street Jour...	2004	5	p5	0
6	PAY WITHOUT PERFORMANCE By Lucian Bebch...	Wall Street Jour...	2004	6	p5	0
7	Fannie Mae's board, under growing pressure fr...	Wall Street Jour...	2004	7	p5	0
8	The slo-mo disintegration of the Fannie Mae my...	Wall Street Jour...	2004	8	p5	0
9	INVESTORS FLOCKED to soaring Russian stocks...	Wall Street Jour...	2004	9	p5	0
10	Fannie Mae's directors were divided over the fa...	Wall Street Jour...	2004	10	p5	0
11	REJECTING BAD news. Managing crises poorly. ...	Wall Street Jour...	2004	11	p5	0
12	Dow Jones Newswires Deutsche Telekom AG pla...	Wall Street Jour...	2004	12	p5	0

Table: articles_111815_lx_pq

Columns:

article_id	int(11
full_text	mediu
Publication_title	varch

```
   #  Time      Action                                                           Message
✓  1  15:22:51  SELECT * FROM twitter_data.chicago_cubs LIMIT 0, 10000             0 row(s) returned
✓  2  15:57:04  SELECT * FROM textpreds.articles_111815_lx_pq LIMIT 0, 10000       4821 row(s) returned
```

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

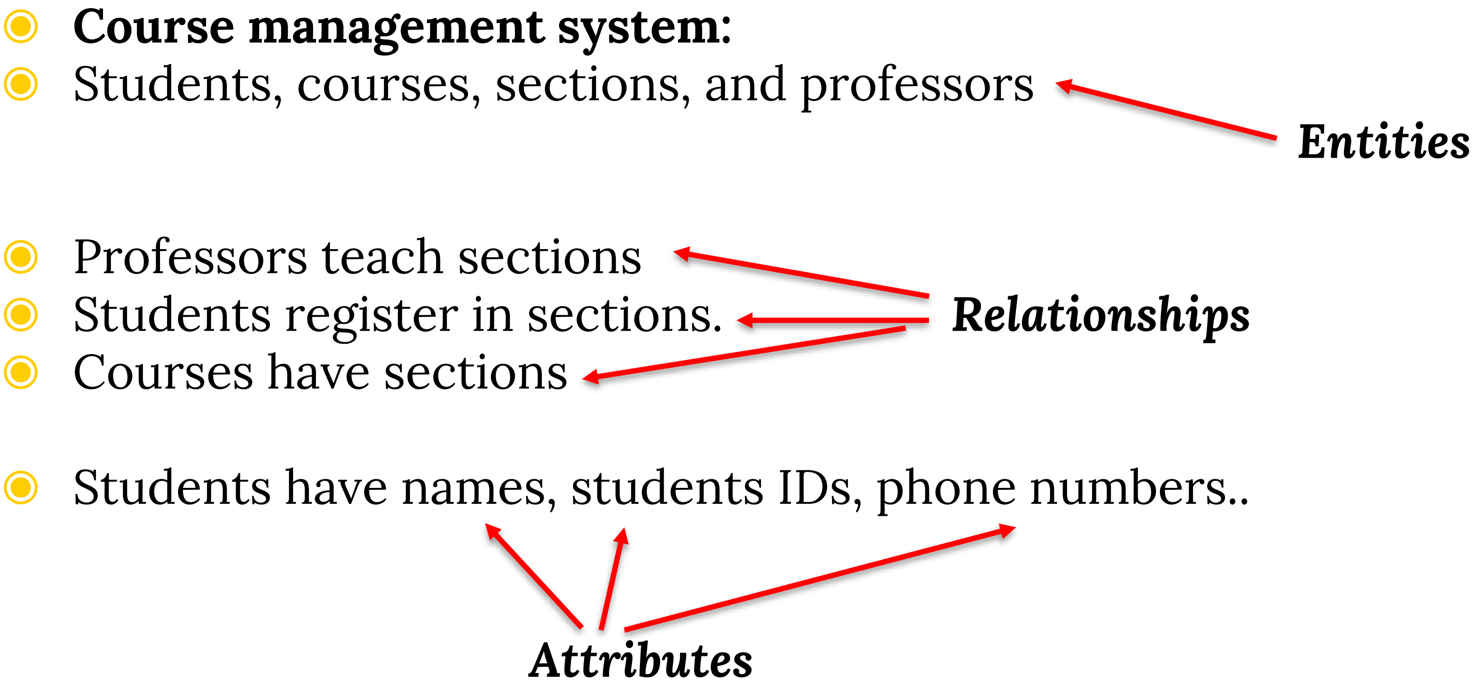
Example 1:

- **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

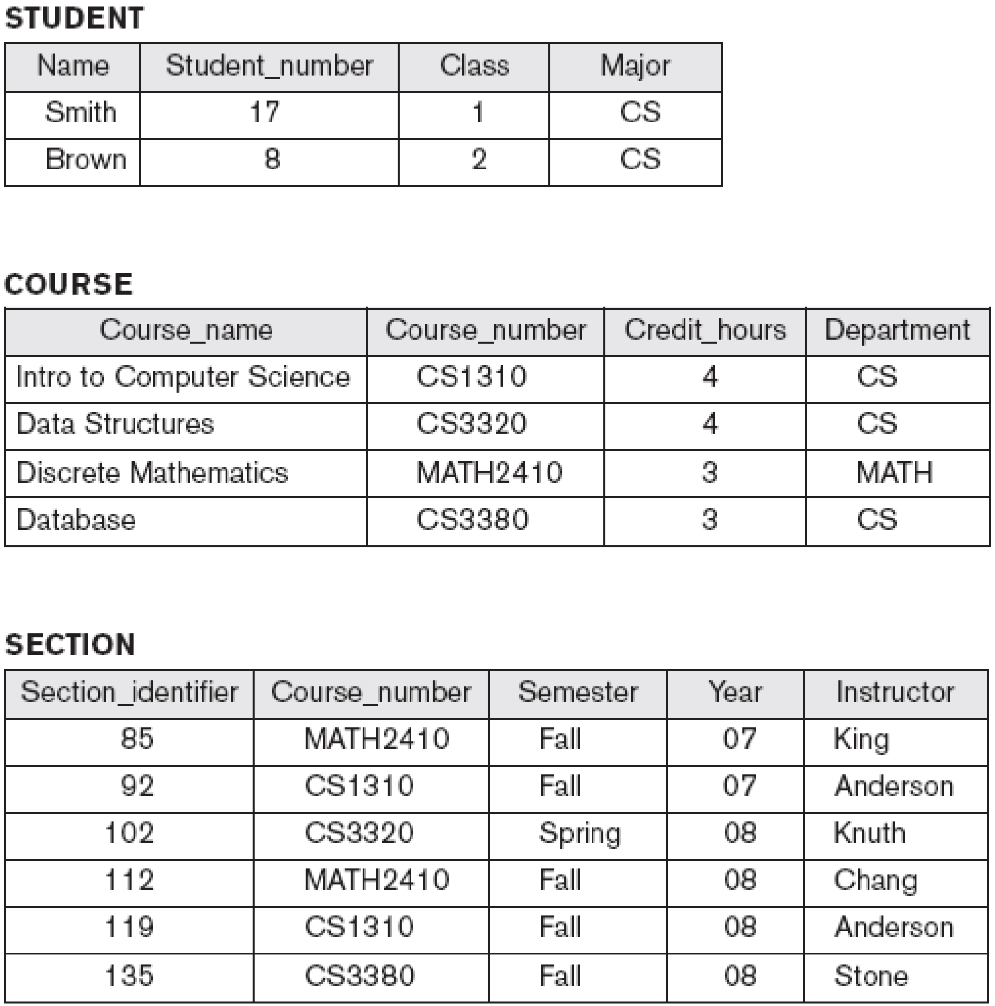
- Professors teach sections
- Students register in sections.
- Courses have sections

Relationships

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

Attributes

Example 1



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

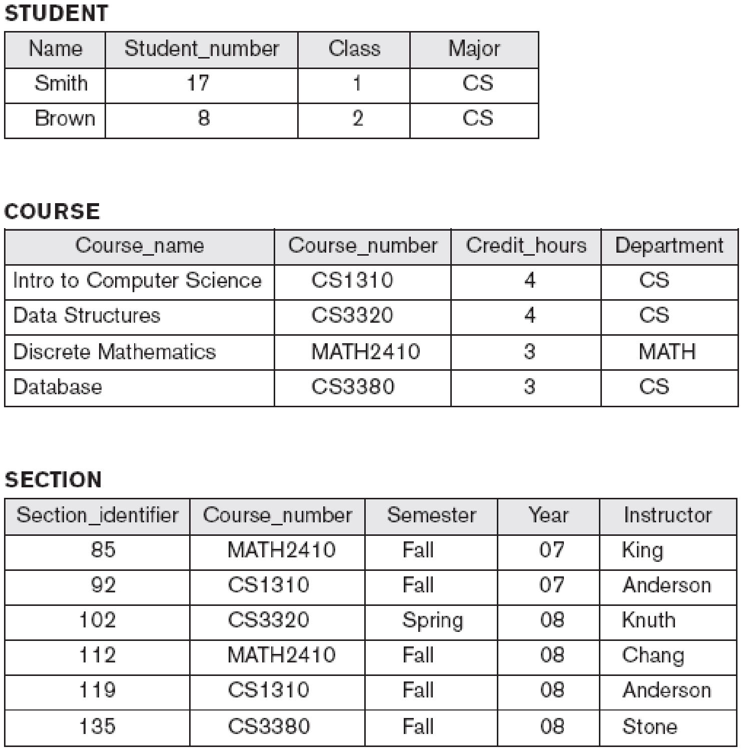
SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

Example 1:

- **Query examples:**

1. Return all students
2. Classes offered in Spring of 08
3. Courses taught in the CS department



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

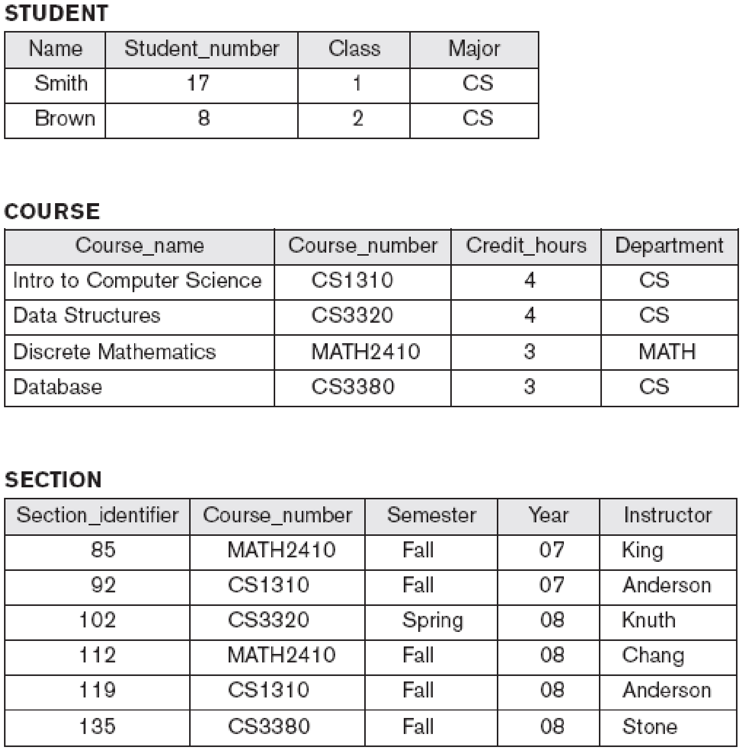
Example 1:

Update examples:

1. Change Smith major to CE

2. Add new math course

3. Change instructor for CS1310 from Anderson to Kruth



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

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

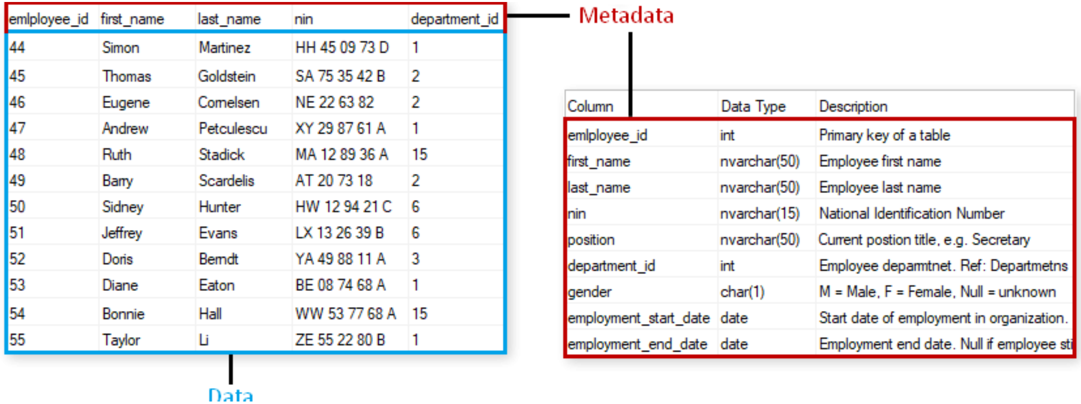
- Examples of queries?
- Examples of updates?

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.



emlployee_id	first_name	last_name	nin	department_id
44	Simon	Martinez	HH 45 09 73 D	1
45	Thomas	Goldstein	SA 75 35 42 B	2
46	Eugene	Cornelsen	NE 22 63 82	2
47	Andrew	Petculescu	XY 29 87 61 A	1
48	Ruth	Stadick	MA 12 89 36 A	15
49	Barry	Scardelis	AT 20 73 18	2
50	Sidney	Hunter	HW 12 94 21 C	6
51	Jeffrey	Evans	LX 13 26 39 B	6
52	Doris	Berndt	YA 49 88 11 A	3
53	Diane	Eaton	BE 08 74 68 A	1
54	Bonnie	Hall	WW 53 77 68 A	15
55	Taylor	Li	ZE 55 22 80 B	1

Column	Data Type	Description
emlployee_id	int	Primary key of a table
first_name	nvarchar(50)	Employee first name
last_name	nvarchar(50)	Employee last name
nin	nvarchar(15)	National Identification Number
position	nvarchar(50)	Current postion title, e.g. Secretary
department_id	int	Employee depamtnet. Ref: Departmetns
gender	char(1)	M = Male, F = Female, Null = unknown
employment_start_date	date	Start date of employment in organization.
employment_end_date	date	Employment end date. Null if employee sti

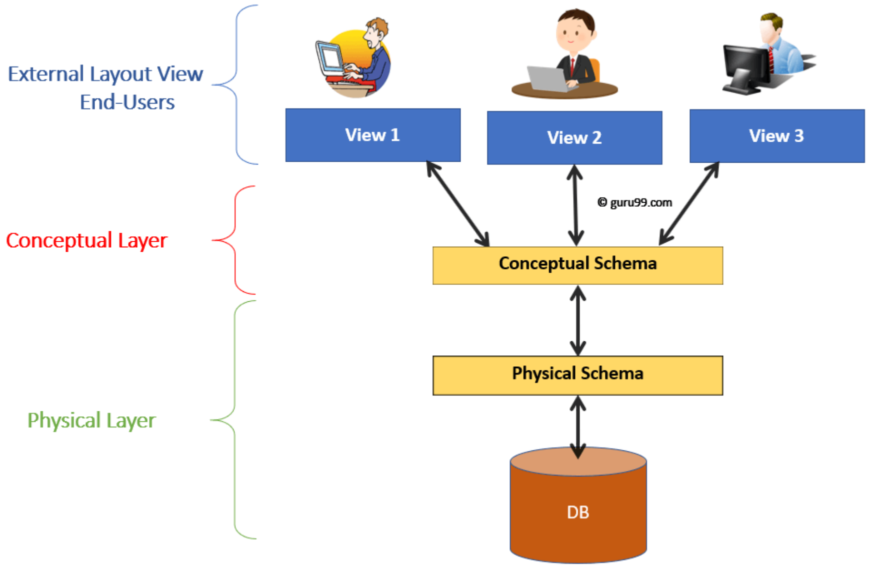
Data Independence

- *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.



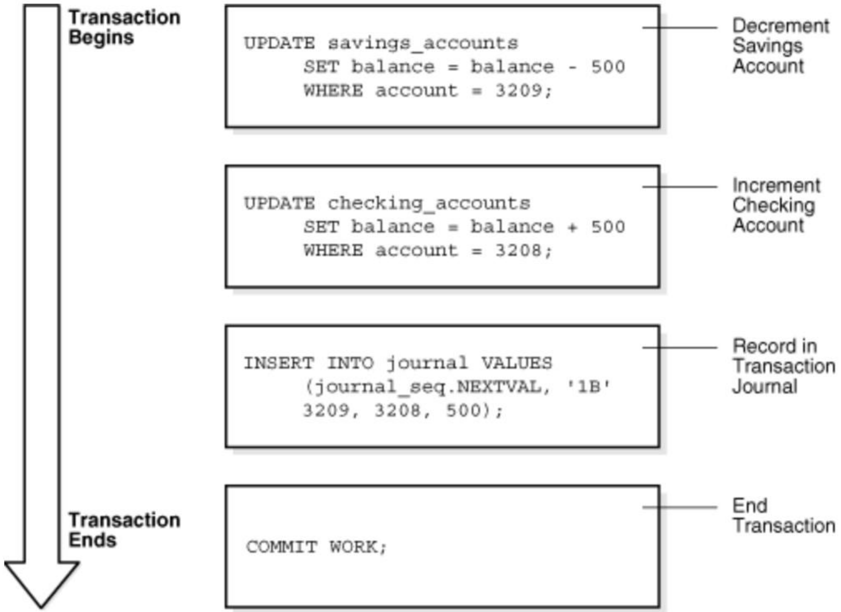
Sharing of data and multi-user transaction processing

- 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)

Figure 10-1 A Banking Transaction



```
UPDATE savings_accounts
      SET balance = balance - 500
      WHERE account = 3209;
```

```
UPDATE checking_accounts
      SET balance = balance + 500
      WHERE account = 3208;
```

```
INSERT INTO journal VALUES
      (journal_seq.NEXTVAL, '1B'
      3209, 3208, 500);
```

```
COMMIT WORK;
```

Transaction Begins
Transaction Ends
Decrement Savings Account
Increment Checking Account
Record in Transaction Journal
End Transaction

Description of "Figure 10-1 A Banking Transaction"

Transactions

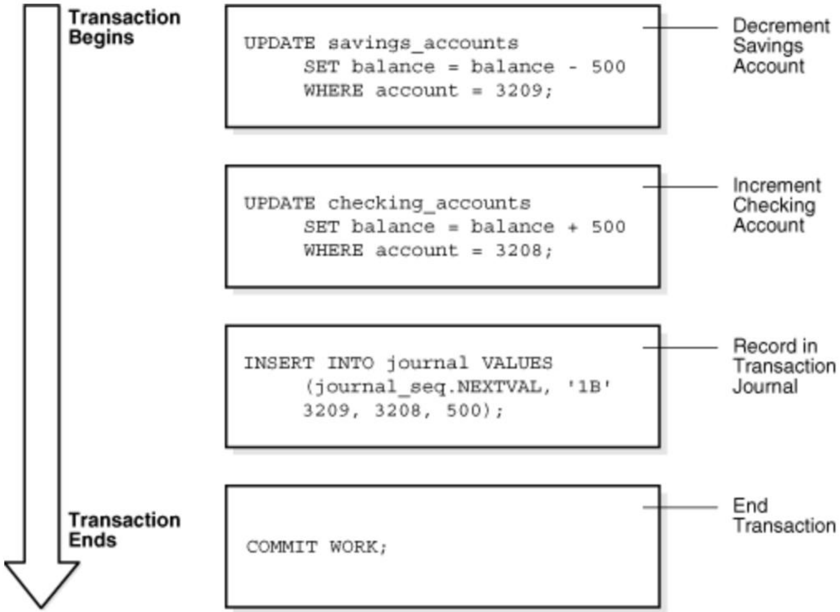
- Unit of work (lines of code)

- **Two properties:**

1. **Isolation property:** Each transaction is independent from the other.

2. **Atomicity property:** Either execute all the lines successfully or none

Figure 10-1 A Banking Transaction



```
UPDATE savings_accounts
    SET balance = balance - 500
    WHERE account = 3209;
```

```
UPDATE checking_accounts
    SET balance = balance + 500
    WHERE account = 3208;
```

```
INSERT INTO journal VALUES
    (journal_seq.NEXTVAL, '1B'
    3209, 3208, 500);
```

```
COMMIT WORK;
```

Transaction Begins
Decrement Savings Account
Increment Checking Account
Record in Transaction Journal
Transaction Ends
End Transaction

Description of "Figure 10-1 A Banking Transaction"

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

PLAYER

Online_ID	Date_of_Birth	Country

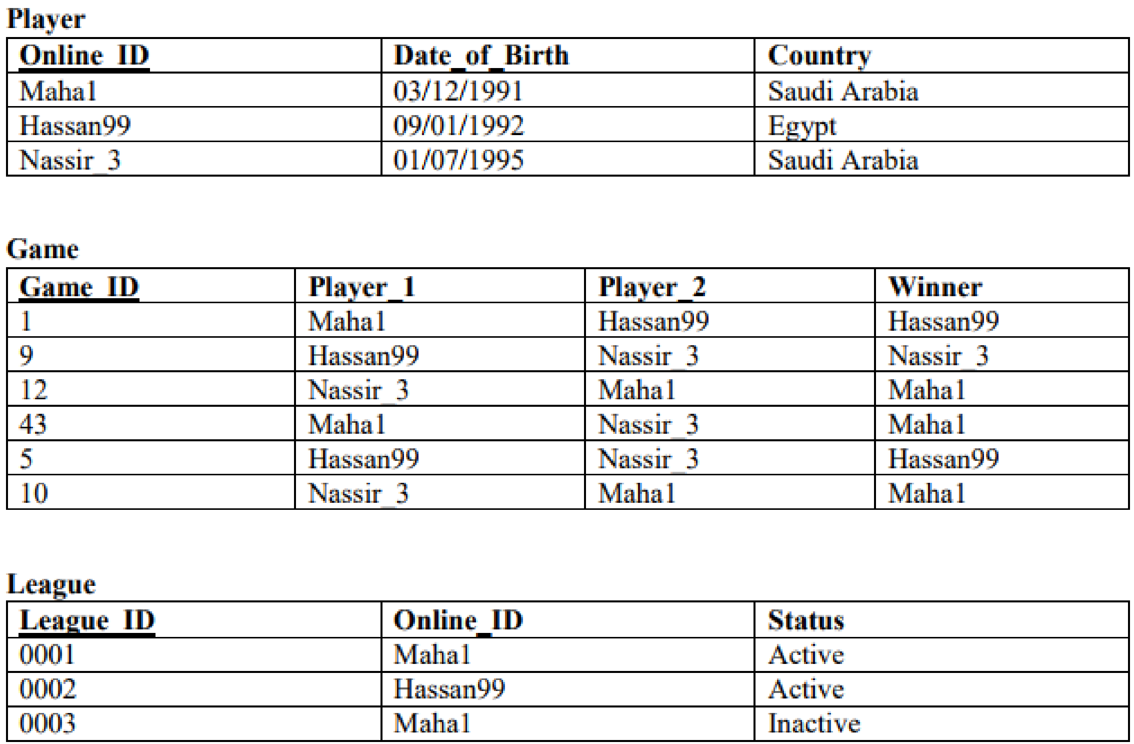
GAME

Game_ID	Player_1	Player_2	Winner

LEAGUE

League_ID	Online_ID	Status

Database State: Example



Player

Online_ID	**Date_of_Birth**	**Country**
Maha1	03/12/1991	Saudi Arabia
Hassan99	09/01/1992	Egypt
Nassir_3	01/07/1995	Saudi Arabia

Game

Game_ID	**Player_1**	**Player_2**	**Winner**
1	Maha1	Hassan99	Hassan99
9	Hassan99	Nassir_3	Nassir_3
12	Nassir_3	Maha1	Maha1
43	Maha1	Nassir_3	Maha1
5	Hassan99	Nassir_3	Hassan99
10	Nassir_3	Maha1	Maha1

League

League_ID	**Online_ID**	**Status**
0001	Maha1	Active
0002	Hassan99	Active
0003	Maha1	Inactive