

Relational Databases

Manas Jyoti Das, PhD
Computer Science
SIUE

What is a database

- A collection of data
- A collection of related pieces of data
- In this lecture we mostly talk about text data
- A maintained collection of data
- Desirable properties of a DBMS: easy to insert, search, find duplicate, delete, link with other pieces of data, concurrent users, etc.
- Example: SIUE- Student data, faculty data, subject (course) data, staff data, etc.

Why not store data in simple files?

- Same data may be present in many files
- If need to change one piece of information that may scattered around various files, one need to change all the files
- Problem to maintain consistency
- Creating a database file system in a OS
- It will be hard to query about a piece of information with OS file system (e.g. find all the employees above age 25, who are from computer science department and is a resident of Illinois)
- In DBMS, there is a data access language called SQL (Structured Query Language)

E/R Model in brief

- Three important terms: entity, relationship and attributes
- University database:
 - entity : students, courses, faculty, departments, etc
 - Relationship:
 - grading relationship between student and faculty or TA
 - Enrolment relationship between student and course
 - Employment relationship between faculty and department, etc.
 - Attribute:
 - Name, 800_number, address, etc.
 - Name, Employment_ID, salary, etc.

Data models

- Conceptual data model
 - A high level description of the database
 - This will help in understanding the requirements (what kind of data do we really need)
- Representational data model
 - It helps in creating the logical representation of the data.
 - How data are related to each other
- Physical data model
 - Details about the data, its type
 - Format of the data

Representational level

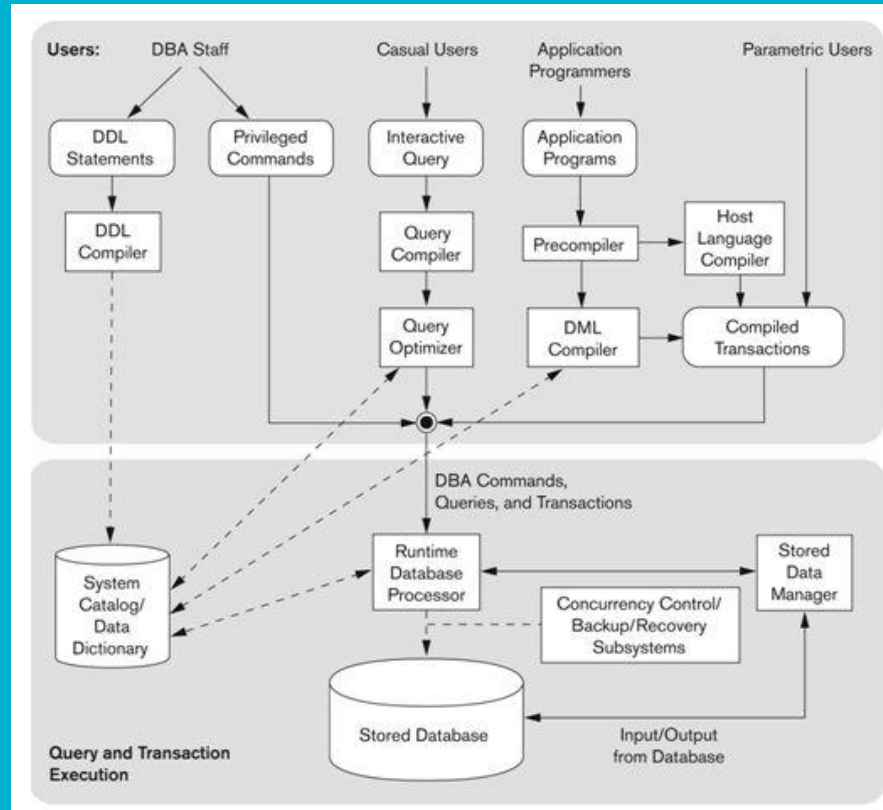
The diagram illustrates a database table structure. The table is titled 'Student' and has seven columns representing attributes: S_Name, 800_Num, Birth_date, Year_Join, Program, Dept, and Addr. The first row contains specific data for a student named Bob. The second row contains ellipses, indicating more data tuples. Annotations with arrows point to the table title, the column headers, and the first data row.

Student						
S_Name	800_Num	Birth_date	Year_Join	Program	Dept	Addr
Bob	800809999	09/12/2003	2021	Under_graduate	CS	xyz
....

RDBMS

- Relational database management system
- Schema: table name, data type and attribute names for the table, constraints, etc.
- Database definition: Setting up the structure of the table, with datatype and constraints, etc.
- Database populating: Input data or storing data

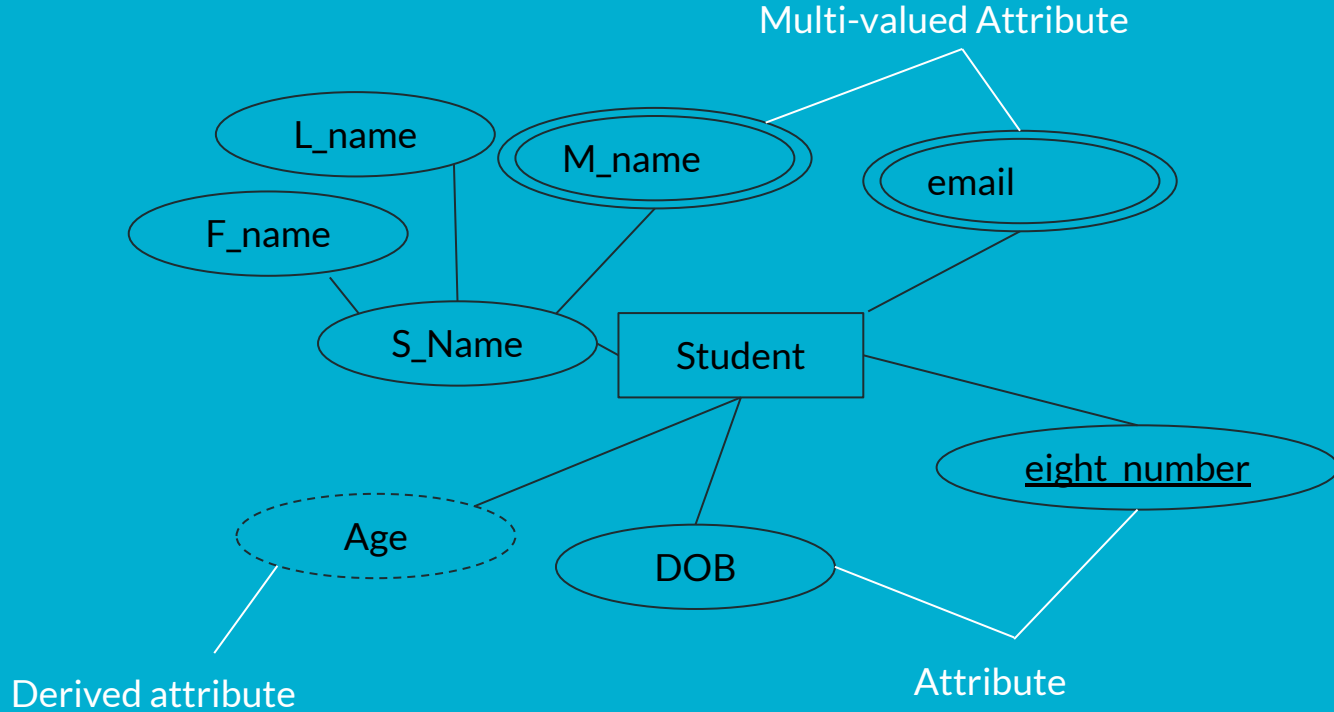
RDBMS architecture



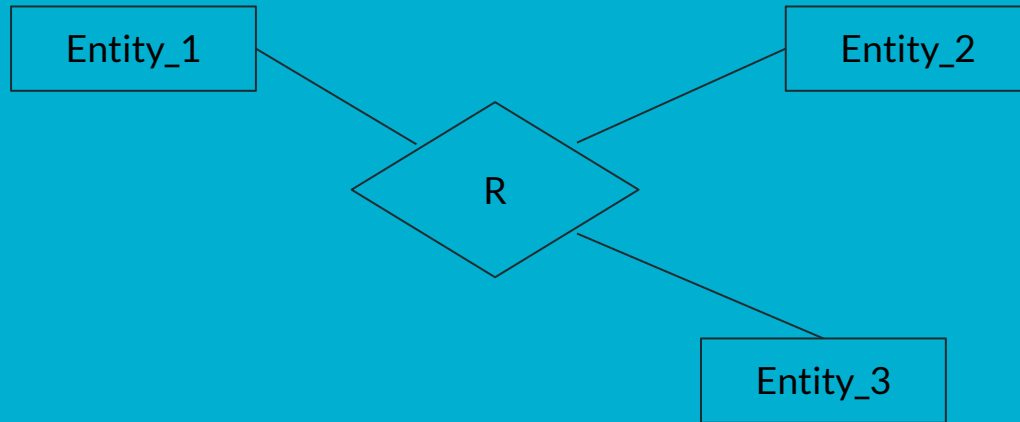
Types of attributes

- Simple attribute: Having atomic value, 800_number, phone_number, Name?
- Composite attribute: Having several components in the value
 - Address: Street address, city, state, and zip code
 - Order: Order ID, customer ID, product ID, order quantity, and order date
 - Customer: First name, last name, email address, and phone number
- Derived attributes
 - Age: Derived from the date_of_birth attribute
 - Average customer rating: Derived from the customer_rating attribute in the product_reviews table
 - Shipping cost: Derived from the shipping_weight attribute in the orders table and the shipping_rates table
- Single valued
- Multi Valued

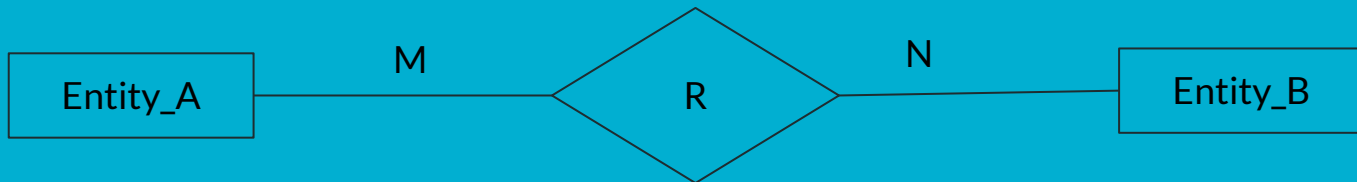
ER Diagram



Notation for relationship



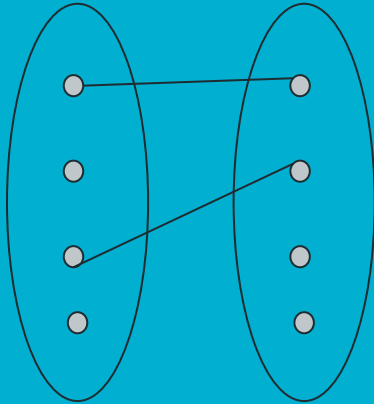
Relationships



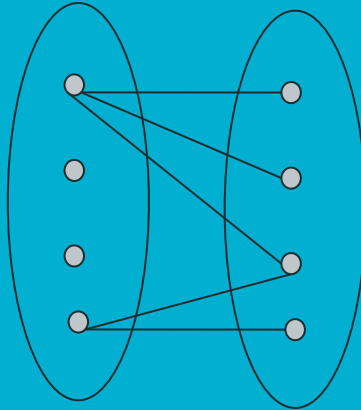
- A **one-to-one relationship (1:1)** means that entity A is associated with only one entity B and vice versa. Ex: One department chair heads one department, and one department can have only one department chair.
- A **one-to-many relationship (1:M)** means that entity A is associated with one or more B entities, and one B entity is associated with only one A entity. Ex: One course can have many classes (sections), but a single class is created from a single course.
- A **many-to-many relationship (M:N)** means that one A entity is associated with many B entities, and one B entity is associated with many A entities. Ex: One student enrolls in many classes, and a single class has many enrolled students.

Relationships

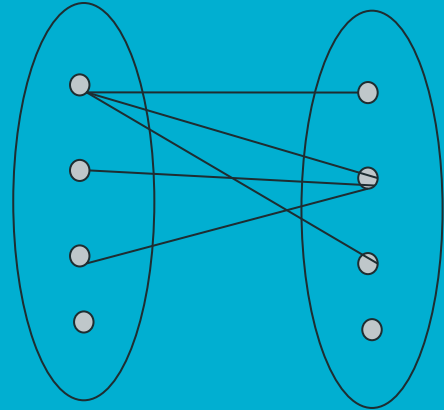
One-to-One



One-to-Many

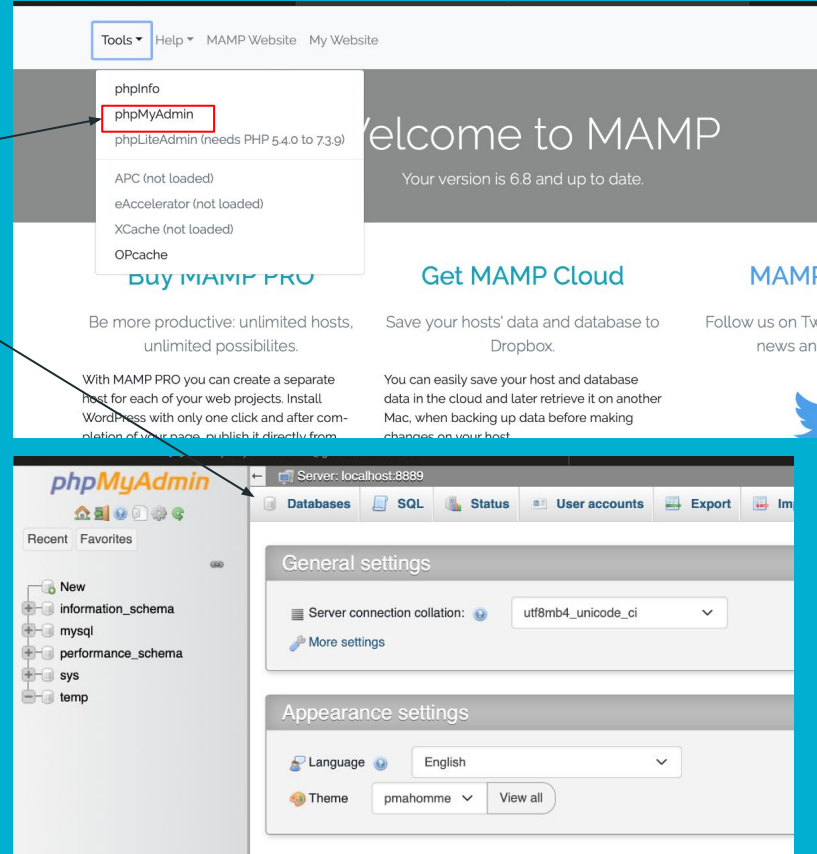


Many-to-Many



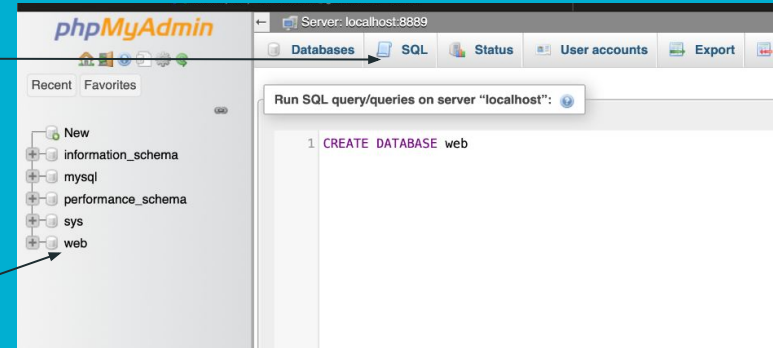
MAMP and database

- Click on phpMyAdmin
- After that click on Databases



Creating a database

- Click on the SQL tab
- Create a database (not a table)
- Inside of a database you can create tables
- To create a database, the command is:
 - CREATE DATA web
 - Here web is the name of the database, you can name it appropriately
 - After creation of the database it will be shown here



Keys

- **Primary Key (PK):** A primary key is a unique identifier for each record in a table. A primary key can consist of one or multiple columns
- **Super Key:** A super key is a set of one or more columns that can be used to uniquely identify records within a table. It may contain more columns than necessary to uniquely identify records, and it may include other non-unique columns as well
- **Candidate Key:** A candidate key is a minimal super key, meaning it is a super key with the fewest possible columns required to uniquely identify records. In a table, there can be multiple candidate keys, and one of them is chosen to be the primary key
- **Foreign Key (FK):** A foreign key is a column or set of columns in one table that is used to establish a link between the data in two tables. It typically refers to the primary key of another table, creating a relationship between them
- Candidate keys are used to normalize databases, which means putting the data into a logical and efficient format
- Alternate keys can be used to create indexes on tables, which can improve the performance of queries

Create a table

- An example:

```
CREATE TABLE student (  
  id INT NOT NULL AUTO_INCREMENT,  
  name VARCHAR(255) NOT NULL,  
  email VARCHAR(255) NOT NULL,  
  PRIMARY KEY (id)  
);
```

- In MAMP to create a table select the database. Which is towards the left of phpMyAdmin page. In the previous slide we created a database called “web”

Data Types in MySql

- Numeric data types:
 - INT, TINYINT, BIGINT, etc.
- Floating point:
 - FLOAT, DOUBLE, REAL, etc.
- String:
 - CHAR, VARCHAR, TEXT, LONGTEXT, etc.
- Date and time:
 - DATE, TIME, DATETIME
- Many more

Inserting data into a table

- In the query box write of MAMP:

```
INSERT INTO student (name, email) VALUES ('John Doe', 'john.doe@example.com');
```

```
INSERT INTO student (name, email) VALUES ('Jane Doe', 'jane.doe@example.com');
```

- To see the inserted data:

```
SELECT * FROM student
```

Selecting a row from the table

- Based on some criteria you want to query the table
- The rows of the table which does matches with the query will be given as a output
- example:

```
SELECT * FROM users WHERE name='John doe'
```

- Not case sensitive

```
SELECT * FROM users WHERE name LIKE 'John%'
```

—