Introduction to Mysql

Created	@May 2, 2022 8:45 PM
Class	
Type	
Materials	
Reviewed	☐

In earlier days, data was generally stored in files. Still a lot of people use excel sheets for representing data.

This file based system was having a lot of problems:

- Data Redundancy
- Data inconsistency
- Difficult data access
- Security Problems
- Difficulty in concurrent access

Database

> A database is a shared collection of related data. By data, we mean known facts that can be recorded and have implicit meaning. For example: Names, Telephone numbers, addresses of the students.

DBMS (Database management system)

DBMS is a software that helps us to use this approach of the new mechanism database.

DBMS helps in the following operations:

- Defining data
- Creating data
- Manipulating data

- Reading data

Example: MySQL, PgSQL, Sqlite, etc.

SQL

SQL (Structured query language), it is a language using which we can perform operation on different dbms. So every dbms might have different syntax for the SQL. (Although most of the syntax is similar). We will try to understand mysql, in the lecture.

RDBMS (Relational Database Management System)

In this kind of DBMS we store data in the form of tables.

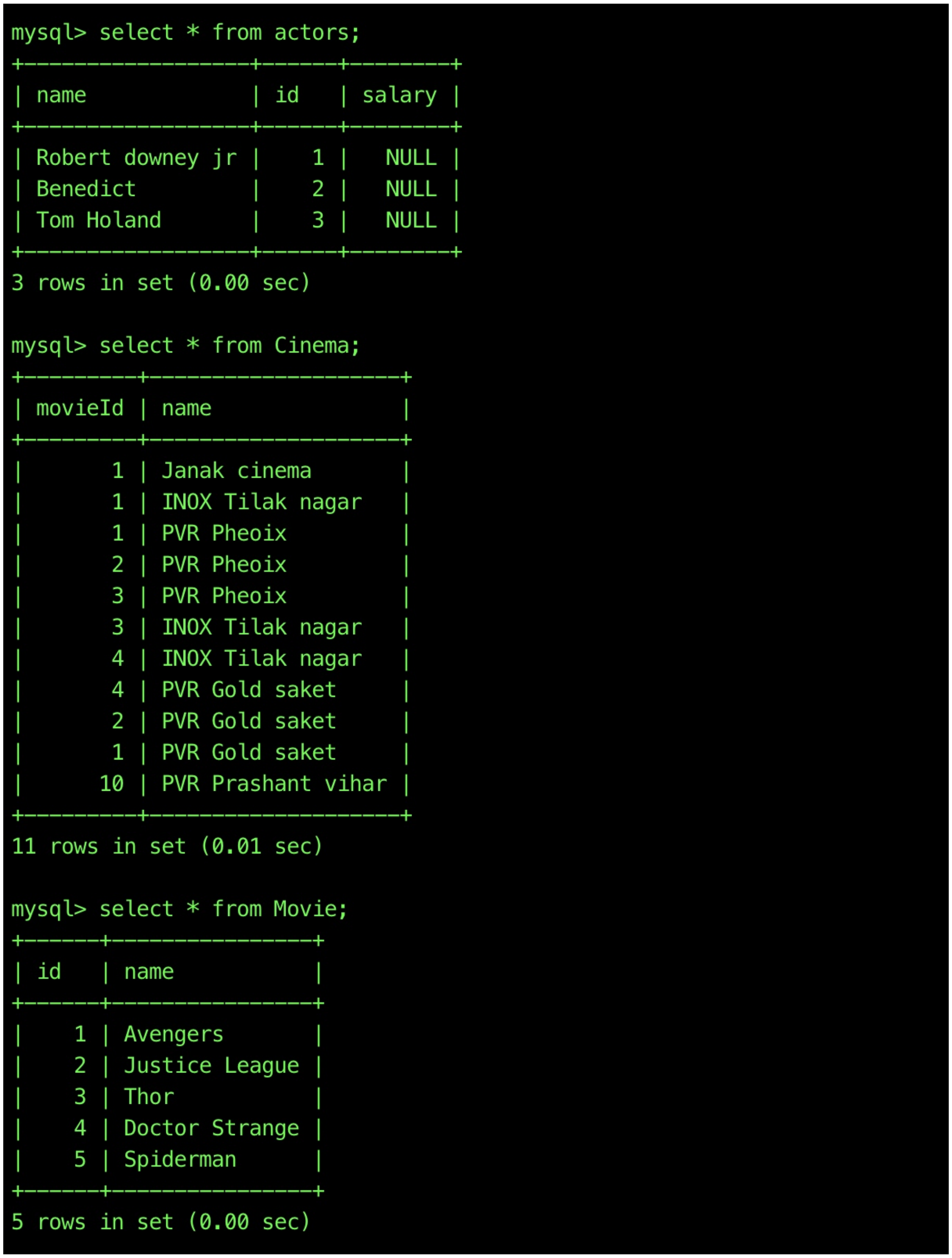
Roll No	Student Name
1	Sanket
2	Archit

Course id	Course Name
1	Physics
2	Web development

The tables are going to represent real life entities. Example: Students, Courses, etc.

The columns of the tables will represent the properties of the entities.

The rows of the tables will represent actual data entry in the table.



```
mysql> select * from actors;
+------------------+------+--------+
| name             | id   | salary |
+------------------+------+--------+
| Robert downey jr |    1 |   NULL |
| Benedict         |    2 |   NULL |
| Tom Holand       |    3 |   NULL |
+------------------+------+--------+
3 rows in set (0.00 sec)

mysql> select * from Cinema;
+---------+--------------------+
| movieId | name               |
+---------+--------------------+
|       1 | Janak cinema       |
|       1 | INOX Tilak nagar   |
|       1 | PVR Pheoix         |
|       2 | PVR Pheoix         |
|       3 | PVR Pheoix         |
|       3 | INOX Tilak nagar   |
|       4 | INOX Tilak nagar   |
|       4 | PVR Gold saket     |
|       2 | PVR Gold saket     |
|       1 | PVR Gold saket     |
|      10 | PVR Prashant vihar |
+---------+--------------------+
11 rows in set (0.01 sec)

mysql> select * from Movie;
+------+----------------+
| id   | name           |
+------+----------------+
|    1 | Avengers       |
|    2 | Justice League |
|    3 | Thor           |
|    4 | Doctor Strange |
|    5 | Spiderman      |
+------+----------------+
5 rows in set (0.00 sec)
```

The above image represents example of tables that we can store in RDBMS.

MySQL is a RDBMS.

- You can have multiple databases.
- In each database you will be storing related data in the form of multiple tables.
- Tables will be actually storing the data.

SQL Syntax:

How to create database ?

```
CREATE DATABASE University; // CREATE DATABASE <Name of the database>
```

How to list all the databases ?

```
SHOW DATABASES;
```

How to delete a database?

```
DROP DATABASE ecomdb; // DROP DATABASE <Name of the database>
```

How to start using a database ?

```
USE University; // USE <Name of the database>
```

NOTE: Strings in MySQL are Varchar

How to create a table ?

```
CREATE TABLE Student (ID INT, AGE INT, NAME VARCHAR(20));
```

How to list all the tables ?

```
SHOW TABLES;
```

How to insert data in a table ?

```
INSERT INTO STUDENT (ID, NAME, AGE) VALUES (1, "Sanket", 24);
```

How to check properties of a table and their type ?

```
DESC Student; // DESC <Name of the table>
```

How to display data of the table ?

```
SELECT * FROM Student; // If you want to fetch all the columns
SELECT ID, NAME FROM Student; // If you want to fetch some columns not all
// SELECT <Columns to fetch> FROM <Name of the table>
```

How to update data in a table ?

```
UPDATE Student SET AGE = 22 WHERE ID = 2;
// Update <Table name> SET <Column name> = <New Value> WHERE <Column name> = <value>;
```

How to delete data from a table ?

```
DELETE FROM Student WHERE ID = 3;
```

How to filter the records form a table ?

```
SELECT * FROM Student WHERE AGE >= 24;
```

In Mysql we can also use logical operators

```
SELECT * FROM Student WHERE AGE >=24 OR AGE <= 20;;
```

Like Query for string matching

Prefix matching

```
SELECT * FROM STUDENT WHERE NAME LIKE "K%"; // returns all the names that start with k
```

Suffix Matching

```
SELECT * FROM STUDENT WHERE NAME LIKE "%t";// returns all the names which ends with t
```

Substring matching

```
SELECT * FROM STUDENT WHERE NAME LIKE "%an%" ;// returns all the names which has an in between somewhere
```

Few more queries

```
SELECT Count(*) FROM Student;
SELECT Count(*) FROM Student WHERE NAME LIKE "%an%" ;
SELECT SUM(AGE) FROM Student;
SELECT SUM(AGE) FROM Student  WHERE NAME LIKE "%an%" ;—
```

Data Modelling

For building any application, we have to logically think how our data should look like in the database.

For example: We have a table of customer who bought some product from our website.

Customer ID	
Customer Name	
Phone Number	
Product Id	
Product Name	
Price	

One customer can buy multiple products, so there will multiple rows for a customer.

Now assume, the phone number of the customer changed, so in order to update the phone number we need to go to all the rows where the details of this customer is present and then update all of them

So here we can directly see that its not an ideal design.

E R Diagrams

Entity Relation Diagram. What is entity ? So any real life object represented by our tables.

Relation means relationship between two entities.

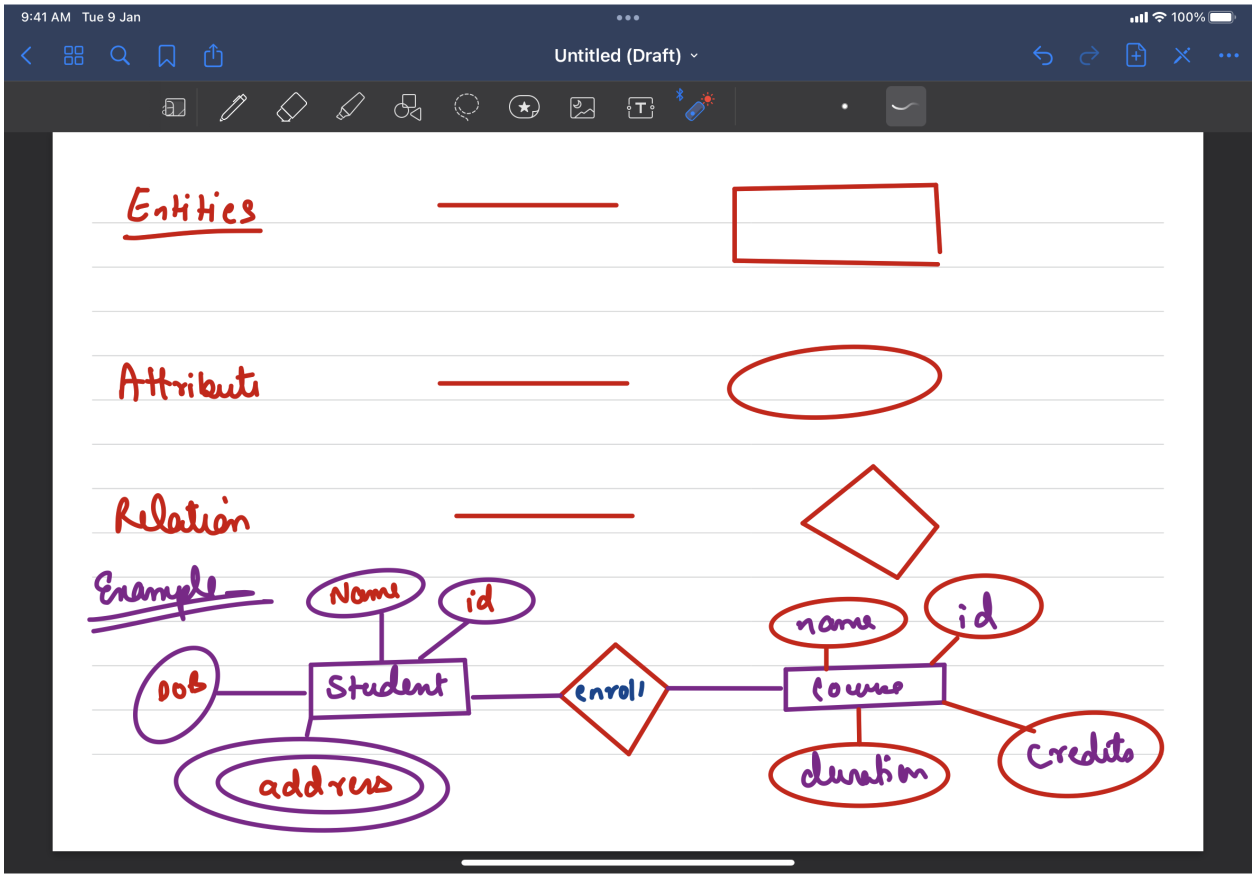
Example: Table: Student, Table: Course → A student enrolls in many courses.

Table: Project → A student has one project and a project belongs to one student.

So this diagram, helps us to first visualise how the relationships between the entities look like and then we can prepare our database.

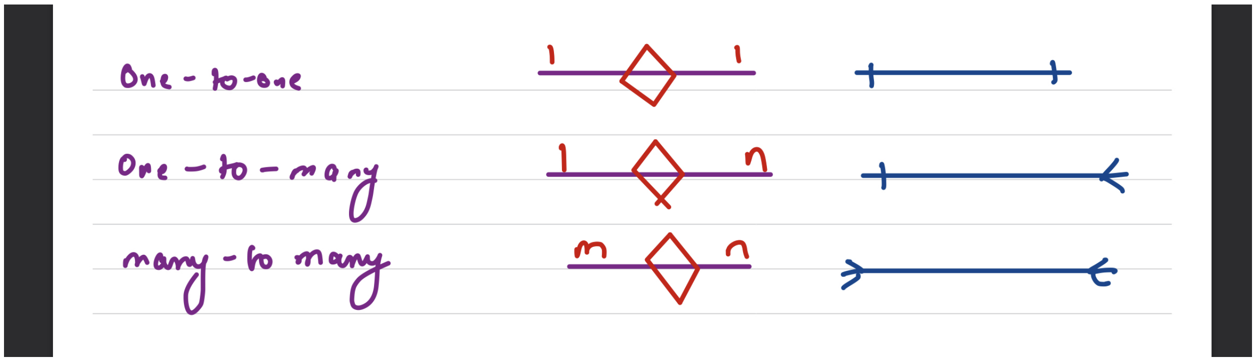
In order to prepare E R Diagrams, we need to first understand few terminologies:

- **Entity:** Any real life object. Example: Student
- **Attribute:** Property of an entity. Example: Age of a student
  - **Single:** These attributes have a single value. Example: Phone number
  - **Composite:** An attribute where value of the attribute can be further sub divided into meaningful values. Example: Address - (House No, Street, City, State, Pin Code)
  - **Derived:** An attribute like age, should be a derived attribute that means we don't need to specifically store it, we can derive it from other attribute like D.O.B
- **Relationship:** Relation between different entities. Example: Student has a project.
-



Types of relations

- one to one : A student has a project and a project belongs to a student
- many to many: A student has many courses, and each course has many students
- one to many: A department will be having many employees, but an employee belongs to one department.



Task:

Draw an ER Diagram for a flight booking system

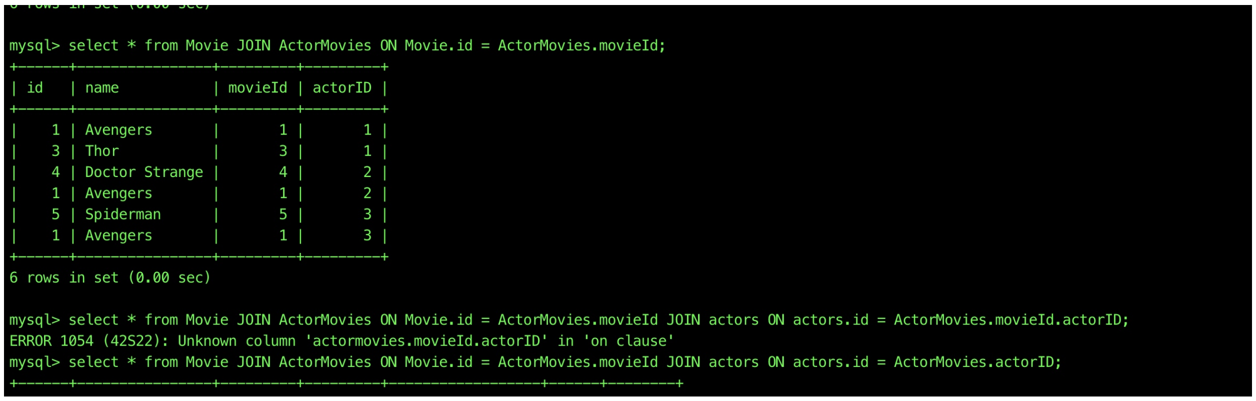
Follow the given requirements:

- **Entities:**
  - Airplane
  - Flight
  - Passenger
  - Booking
- **Attributes:**
  - Airplane: model number, registration number, capacity etc.
  - An airplane flight has unique flight number, departure airport, arrival airport, departure date, arrival date, departure time, arrival time, etc.
  - Passenger: Name, email, address etc
- **Relations:**
  - **Each flight is carries by a single airplane**
  - Airline has one or more airplanes
  - Passenger can book a seat in a flight for a tour.
  - You can add few more relations if you want.

Task:

Design an ER diagram for a basic twitter application that can support following features

- User can tweet
- User can comment on a tweet
- User can like a tweet or a comment
- User can comment on a comment
- User can follow other users



```
6 rows in set (0.00 sec)

mysql> select * from Movie JOIN ActorMovies ON Movie.id = ActorMovies.movieId;
+------+----------------+---------+---------+
| id   | name           | movieId | actorID |
+------+----------------+---------+---------+
|    1 | Avengers       |       1 |       1 |
|    3 | Thor           |       3 |       1 |
|    4 | Doctor Strange |       4 |       2 |
|    1 | Avengers       |       1 |       2 |
|    5 | Spiderman      |       5 |       3 |
|    1 | Avengers       |       1 |       3 |
+------+----------------+---------+---------+
6 rows in set (0.00 sec)

mysql> select * from Movie JOIN ActorMovies ON Movie.id = ActorMovies.movieId JOIN actors ON actors.id = ActorMovies.movieId.actorID;
ERROR 1054 (42S22): Unknown column 'actormovies.movieId.actorID' in 'on clause'
mysql> select * from Movie JOIN ActorMovies ON Movie.id = ActorMovies.movieId JOIN actors ON actors.id = ActorMovies.actorID;
+------+----------------+---------+---------+------------------+------+--------+
```