Databases? RDBMS : MySQL

Introduction to keys

Super key Candidate key Primary key Foreign key Composite key

```sql
SHOW Databases;

-- Syntax: CREATE DATABASE [IF NOT EXISTS] database_name
CREATE DATABASE IF NOT EXISTS `test`;

-- Delete the database
-- Syntax: DROP DATABASE [IF EXISTS] database_name
DROP DATABASE IF EXISTS `test`;



USE lab_session;
-- Creating tables

-- Syntax for creating a table
-- CREATE TABLE <table_name>
-- (<column_name> <data_type>, <column_name> <data_type>, ...);

-- CREATE TABLE IF NOT EXISTS users (
--     id INT PRIMARY KEY,
--     username VARCHAR(50) NOT NULL UNIQUE,
--     password VARCHAR(255) NOT NULL,
--     email VARCHAR(100) NOT NULL UNIQUE,
--     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
-- );


CREATE TABLE IF NOT EXISTS instructors (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);


-- Data types
-- INT: Integer data type
-- VARCHAR(n): Variable-length string with a maximum length of n
characters
-- TIMESTAMP: Date and time data type


-- Constraints
-- NOT NULL: Ensures that a column cannot have a NULL value
-- UNIQUE: Ensures that all values in a column are unique
```

```sql
USE lab_session;

-- Syntax
-- INSERT INTO <table_name> (<column_name>, <column_name>, ...) VALUES
(<value>, <value>, ...);



-- if you don't specify the column names, you must provide values
-- for all columns in the same order as they are defined in the table.

-- INSERT INTO users VALUES
-- (1,   'john_doe', 'password123', 'john@123', '2023-10-01 10:00:00'),
-- (2,   'jane_smith', 'password456', 'jane@456', '2023-10-02 11:00:00'),
-- (3,   'alice_jones', 'password789', 'alice@789', '2023-10-03
12:00:00'),
-- (4,   'bob_brown', 'password101', 'bob@101', '2023-10-04 13:00:00'),
-- (5,   'charlie_black', 'password202', 'charlie@202', '2023-10-05
14:00:00'),
-- (6,   'dave_white', 'password303', 'dave@303', '2023-10-06 15:00:00'),
-- (7,   'eve_green', 'password404', 'eve@404', '2023-10-07 16:00:00'),
-- (8,   'frank_yellow', 'password505', 'frank@505', '2023-10-08
17:00:00'),
-- (9,   'grace_purple', 'password606', 'grace@606', '2023-10-09
18:00:00'),
-- (10,  'hank_orange', 'password707', 'hank@707', '2023-10-10 19:00:00');


SELECT email, username FROM instructors;

-- if you want to insert data into specific columns,
-- you can specify the column names in the INSERT statement.

-- INSERT INTO instructors (password, username, email) VALUES
-- ('password123', 'rahul', 'john@987'),
-- ('password456', 'naman', 'jane@1001');


-- INSERT INTO users (username, password, email) values
-- ('john_doe', 'password123', 'john@123');

-- SHOW TABLES;



-- Create a database SCALER, and add two tables:
-- courses table with appropriate columns
-- and data types:

-- CourseID
-- CourseName
-- Duration
-- StartDate
-- batches table with columns:

-- BatchId
-- BatchName
-- StartDate



-- CREATE DATABASE SCALER;
-- USE scaler;
-- SHOW TABLES;
-- CREATE TABLE Courses (
--     CourseID INT AUTO_INCREMENT PRIMARY KEY,
--     CourseName VARCHAR(100) NOT NULL,
--     Duration INT,
--     StartDate DATE
-- );

-- CREATE TABLE Batches (
--     BatchID INT AUTO_INCREMENT PRIMARY KEY,
--     BatchName VARCHAR(100) NOT NULL,
--     StartDate DATE
-- );

-- Students
-- s_id | name | psp | b_id
-- 1     | a    | 1   | 1
-- 2     | b    | 1   | 1
-- 3     | c    | 1   | 2


-- Batches
-- b_id | b_name | start_date
-- 1     | b1     | 2023-10-01
-- 2     | b2     | 2023-10-02

-- b_id of students table is a foreign key
-- referencing b_id of batches table

-- when adding a student,
-- the value of b_id must exist in the batches table


-- INSERT INTO students (s_id, name, psp, b_id)
-- VALUES
-- (1, 'a', 1, 4);

-- This will fail because b_id 4 does not exist
-- in the batches table

-- DROP DATABASE IF EXISTS scaler;
-- CREATE DATABASE IF NOT EXISTS scaler;
USE scaler;
```

```sql
-- CREATE TABLE IF NOT EXISTS batches (
--     b_id INT PRIMARY KEY AUTO_INCREMENT,
--     b_name VARCHAR(50) NOT NULL,
--     start_date DATE NOT NULL
-- );

-- Way 1 of adding a foreign key constraint
-- CREATE TABLE IF NOT EXISTS students (
--     s_id INT PRIMARY KEY AUTO_INCREMENT,
--     name VARCHAR(50) NOT NULL,
--     psp INT NOT NULL,
--     b_id INT,
--     -- Adding a foreign key constraint
--     FOREIGN KEY (b_id) REFERENCES batches(b_id)
-- );

-- Way 2 of adding a foreign key constraint
-- ALTER TABLE students
-- ADD CONSTRAINT constraint_name
-- FOREIGN KEY (b_id) REFERENCES batches(b_id);

-- INSERT INTO students (s_id, name, psp, b_id)
-- VALUES
-- (1, 'a', 1, 1),
-- (2, 'b', 1, 1),
-- (3, 'c', 1, 2);

-- Didnt add any data in students table because batches
-- table is empty

INSERT INTO batches (b_id, b_name, start_date)
VALUES
(1, 'b1', '2023-10-01'),
(2, 'b2', '2023-10-02');

SELECT * FROM scaler.batches;


-- INSERT INTO scaler.students (s_id, name, psp, b_id)
-- VALUES
-- (1, 'a', 1, 1),
-- (2, 'b', 1, 1),
-- (3, 'c', 1, 2);

SELECT * FROM scaler.students;

DELETE FROM scaler.batches WHERE b_id = 1;

-- This will fail because there are students with b_id 1
-- DELETE FROM scaler.students WHERE b_id = 2;

ALTER TABLE scaler.students
DROP FOREIGN KEY fk_batches_cascade;
```

```
ALTER TABLE scaler.students
ADD CONSTRAINT fk_batches_cascade
FOREIGN KEY (b_id) REFERENCES scaler.batches(b_id)
ON DELETE SET NULL;



SELECT * FROM sakila.film;

SELECT title, release_year FROM sakila.film
ORDER BY title DESC;
```

What is cascade delete? ON DELETE CASCADE means that if a row in the parent table (batches) is deleted, all corresponding rows in the child table (students) will also be deleted.

ON DELETE SET NULL means that if a row in the parent table (batches) is deleted, the foreign key column in the child table (students) will be set to NULL.

ON DELETE RESTRICT means that if a row in the parent table (batches) is deleted, the deletion will be restricted if there are corresponding rows in the child table (students). This is the default behavior if no action is specified.

ON DELETE NO ACTION means that if a row in the parent table (batches) is deleted, no action will be taken on the child table (students).

Implement ON DELETE CASCADE and ON UPDATE CASCADE on the students table where the batch ID references the batches table. Try implementing ON DELETE SET NULL and ON UPDATE SET NULL in your database.

Sakila Database

Download

Download the zip file containing two essential files:

- schema.sql (Contains table structures)
- data.sql (Contains data for the tables)

Installation:

1. Step 1: Open your MySQL client.
2. Step 2: Run the schema.sql file to create the database and tables. This will create the structure for the Sakila database. Command: source /path/to/sakila-schema.sql

3. Step 3: Run the data.sql file to insert data into the tables. This will populate the tables with sample data. Command: source /path/to/sakila-data.sql

SELECT queries

```sql
SELECT * FROM sakila.film;
```

Task 1 - The marketing team wants an alphabetized list of all actors to design name cards.

```sql
SELECT first_name, last_name
FROM sakila.actor
ORDER BY first_name, last_name;
```

Task - 2: The design team wants the list of movies ordered by their length. The longer movies should appear first on the flyers to attract audiences who love long films.

```sql
SELECT film_id, title, length FROM sakila.film
ORDER BY length DESC;
```