DDL Commands

Data Definition Language (DDL) commands allow us to define and manage a schema.

DDL Command	DESCRIPTION	SYNTAX
CREATE	Create a table and its columns together with their datatype.	CREATE TABLE
ALTER	Modify column names and add or delete a column.	ALTER TABLE
RENAME	Change the name of the table.	RENAME TABLE
TRUNCATE	Remove data from a table without deleting the table.	TRUNCATE TABLE
DROP	Delete the table with its data.	DROP TABLE

CREATE command

When I need to create a new table, I use the CREATE TABLE command as seen below:

```
CREATE TABLE actor (
    actor_id varchar(100),
    first_name varchar(100),
    last_name varchar(100),
    last_update varchar(100)
);
```

ALTER command

Using ALTER, we can update a table without having to create another table and delete the old one.

ADD command in ALTER

The ADD COLUMN command in ALTER TABLE is used to add a new column to an existing table. We can also define its data type, constraints, and default value when adding it.

```
ALTER TABLE actor ADD nationality varchar(100);
```

CHANGE command in ALTER

The CHANGE COLUMN command in ALTER TABLE is used to rename a column and optionally change its data type

```
ALTER TABLE actor CHANGE COLUMN last_name family_name varchar(100);
```

MODIFY command in ALTER

The MODIFY COLUMN command in ALTER TABLE is used to change the data type of an existing column.

```
ALTER TABLE actor MODIFY COLUMN last_update TIMESTAMP;
```

DROP command in ALTER

The DROP COLUMN statement is used to permanently remove a column from an existing table. Once dropped, the column and its data cannot be recovered unless restored from a backup.

```
ALTER TABLE actor DROP COLUMN last_update;
```

RENAME TABLE command in SQL

The RENAME TABLE statement in SQL is used to change the name of an existing table without modifying its structure or data.

```
RENAME TABLE actor TO actor_names;
```

TRUNCATE command in SQL

The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

```
TRUNCATE TABLE actor;
```

DROP command in SQL

The DROP TABLE statement is used to drop an existing table in a database.

```
DROP TABLE actor;
```

Joining

Creating Tables with Foreign Key:

```
CREATE TABLE movie (
   movie_id INT PRIMARY KEY,
   title VARCHAR(255),
   release_date DATE
);

CREATE TABLE actor (
   actor_id INT PRIMARY KEY,
   first_name VARCHAR(100),
   last_name VARCHAR(100),
   movie_id INT,
   FOREIGN KEY (movie_id) REFERENCES movie(movie_id)
);
```

Inserting Sample Data into the Tables:

Inserting Data into movie Table:

```
INSERT INTO movie (movie_id, title, release_date)
VALUES
(1, 'Titanic', '1997-12-19'),
(2, 'Interstellar', '2014-11-07'),
(3, 'The Revenant', '2015-12-25');
```

Inserting Data into actor Table:

```
INSERT INTO actor (actor_id, first_name, last_name, movie_id)
VALUES
(1, 'Leonardo', 'DiCaprio', 1),
(2, 'Kate', 'Winslet', 1),
(3, 'Matthew', 'McConaughey', 2),
(4, 'Anne', 'Hathaway', 2),
(5, 'Tom', 'Hanks', NULL);
```

INNER JOIN

An INNER JOIN returns rows that have matching values in both tables. It will exclude rows from either table if no match is found.

```
SELECT actor.first_name, actor.last_name, movie.title
FROM actor
INNER JOIN movie ON actor.movie_id = movie.movie_id;
```

first_name	last_name	title
Leonardo	DiCaprio	Titanic
Kate	Winslet	Titanic
Matthew	McConaughey	Interstellar
Anne	Hathaway	Interstellar

LEFT JOIN (LEFT OUTER JOIN)

The LEFT JOIN returns all rows from the left table (actor), and the matched rows from the right table (movie). If no match is found in the right table, NULL is returned.

```
SELECT actor.first_name, actor.last_name, movie.title
FROM actor
LEFT JOIN movie ON actor.movie_id = movie.movie_id;
```

first_name	last_name	title
Leonardo	DiCaprio	Titanic
Kate	Winslet	Titanic
Matthew	McConaughey	Interstellar
Anne	Hathaway	Interstellar
Tom	Hanks	NULL

RIGHT JOIN (RIGHT OUTER JOIN)

The RIGHT JOIN returns all rows from the right table (movie), and the matched rows from the left table (actor). If no match is found in the left table, NULL is returned.

```
SELECT actor.first_name, actor.last_name, movie.title
FROM actor
RIGHT JOIN movie ON actor.movie_id = movie.movie_id;
```

first_name	last_name	title
Leonardo	DiCaprio	Titanic
Kate	Winslet	Titanic
Matthew	McConaughey	Interstellar
Anne	Hathaway	Interstellar
NULL	NULL	The Revenant

FULL OUTER JOIN (Simulated in MySQL using UNION)

MySQL doesn't support FULL OUTER JOIN directly, but you can simulate it using a combination of LEFT JOIN and RIGHT JOIN with a UNION.

```
SELECT actor.first_name, actor.last_name, movie.title
FROM actor
LEFT JOIN movie ON actor.movie_id = movie.movie_id
UNION
SELECT actor.first_name, actor.last_name, movie.title
FROM actor
RIGHT JOIN movie ON actor.movie_id = movie.movie_id;
```

first_name	last_name	title
Leonardo	DiCaprio	Titanic
Kate	Winslet	Titanic
Matthew	McConaughey	Interstellar
Anne	Hathaway	Interstellar
Tom	Hanks	NULL
NULL	NULL	The Revenant

CROSS JOIN

The CROSS JOIN returns the Cartesian product of two tables, meaning every row from the left table is combined with every row from the right table.

```
SELECT actor.first_name, actor.last_name, movie.title
FROM actor
CROSS JOIN movie;
```

first_name	last_name	title
Leonardo	DiCaprio	Titanic
Leonardo	DiCaprio	Interstellar
Leonardo	DiCaprio	The Revenant
Kate	Winslet	Titanic
Kate	Winslet	Interstellar
Kate	Winslet	The Revenant
Matthew	McConaughey	Titanic
Matthew	McConaughey	Interstellar
Matthew	McConaughey	The Revenant
Anne	Hathaway	Titanic
Anne	Hathaway	Interstellar
Anne	Hathaway	The Revenant
Tom	Hanks	Titanic
Tom	Hanks	Interstellar
Tom	Hanks	The Revenant

SELF JOIN

A SELF JOIN is a join where a table is joined with itself. You can use aliases to differentiate between the two instances of the same table.

For example, let's say we want to find actors who acted in the same movie (i.e., co-actors).

```
SELECT a.first_name, a.last_name, b.first_name AS co_actor_first_name, b.last_name AS co_actor_last_name
FROM actor a
JOIN actor b ON a.movie_id = b.movie_id AND a.actor_id != b.actor_id;
```

first_name	last_name	co_actor_first_name	co_actor_last_name
Leonardo	DiCaprio	Kate	Winslet
Kate	Winslet	Leonardo	DiCaprio
Matthew	McConaughey	Anne	Hathaway
Anne	Hathaway	Matthew	McConaughey