More MySQL Commands

Principles of Databases (CS 365)
Fall 2020

Log in to a Specific Database

Let’s log in to the ***music*** database as ***the_user***:

```
mysql -u the-user -p music
```

As *root*, Set Password for Another User

```
SET PASSWORD
FOR 'the-user'@'localhost' = 'the-password';
```

Update Your Own Password

```
SELECT CURRENT_USER();

SET PASSWORD = 'the-password';
```

Select Statements

Retrieve all fields associated with the artist Interpol.

```sql
SELECT *
FROM artist
WHERE artist_name = "Interpol";
```

Select Statements

Retrieve the name of the artist whose ID is 2.

```
SELECT artist_name
FROM artist
WHERE artist_id = 2;
```

Select Statements

Retrieve all albums that *don’t* have an ***album_id*** of 1.

```
SELECT album_name
FROM album
WHERE album_id <> 1;
```

or use the more familiar not (*!*) operator:

```
SELECT album_name
FROM album WHERE
album_id != 1;
```

Select Statements

Retrieve the name of the 2nd track from all albums [1].

```
SELECT track_name
FROM track
WHERE track_id = 2;
```

[1] This assumes that ***track_id*** is entered into the database based on the track list of the album.

Select Statements

Retrieve the name of all artists whose `artist_id`s are less than 5, or the first 4 artists entered into the database.

```sql
SELECT artist_name
FROM artist
WHERE artist_id < 5;
```

Select Statements

Retrieve all artists whose name appears earlier than “U” in the alphabet.

```
SELECT artist_name
FROM artist
WHERE artist_name < 'U';
```

Select Statements

Retrieve all album names that start with a “D”.

```
SELECT album_name
FROM album
WHERE album_name LIKE "D%";
```

Select Statements

Retrieve all album names that start with four chars, followed by a space, followed by the string “on”, followed by another space, and, finally, ending in any character.

```
SELECT album_name
FROM album
WHERE album_name
LIKE "____ on %";
```

Select Statements

Retrieve artists whose names start with an “I” or with an “M” and end in an “s”.

```sql
SELECT artist_name
FROM artist
WHERE artist_name LIKE "I%" OR artist_name LIKE "M%" AND artist_name LIKE "%s";
```

Same as the following:

```sql
SELECT artist_name
FROM artist
WHERE (artist_name LIKE "I%") OR (artist_name LIKE "M%" AND artist_name LIKE "%s");
```

Select Statements

Retrieve the first and last entries in the ***artist*** table.

```
SELECT *
FROM artist
WHERE (artist_id = 1 OR artist_id = (SELECT COUNT(*) FROM artist));
```

Select Statements

Retrieve everything *but* the first and last entries in the ***`artist`*** table.

```sql
SELECT *
FROM artist
WHERE NOT (artist_id = 1 OR artist_id = (SELECT COUNT(*) FROM artist));
```

Select Statements

Retrieve all album names that don’t start with the letter “D”.

```
SELECT album_name
FROM album
WHERE album_name NOT LIKE "D%";
```

Select Statements

Retrieve everything from the ***artist*** table, listed in descending order by ***artist_id***.

```sql
SELECT *
FROM artist
ORDER BY artist_id DESC;
```

Select Statements

Retrieve everything from the ***artist*** table, listed in descending order by ***artist_name***.

```
SELECT *
FROM artist
ORDER BY artist_name DESC;
```

Select Statements with *INNER JOIN*s

Retrieve the albums made by each artist.

```
SELECT artist_name, album_name
FROM artist INNER JOIN album
USING (artist_id);
```