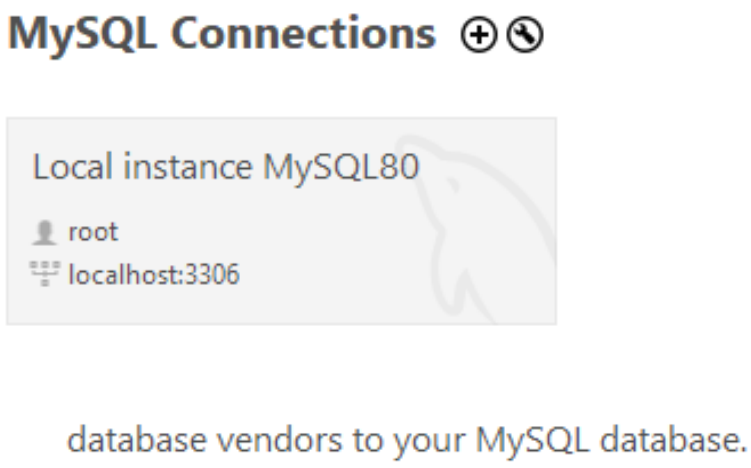
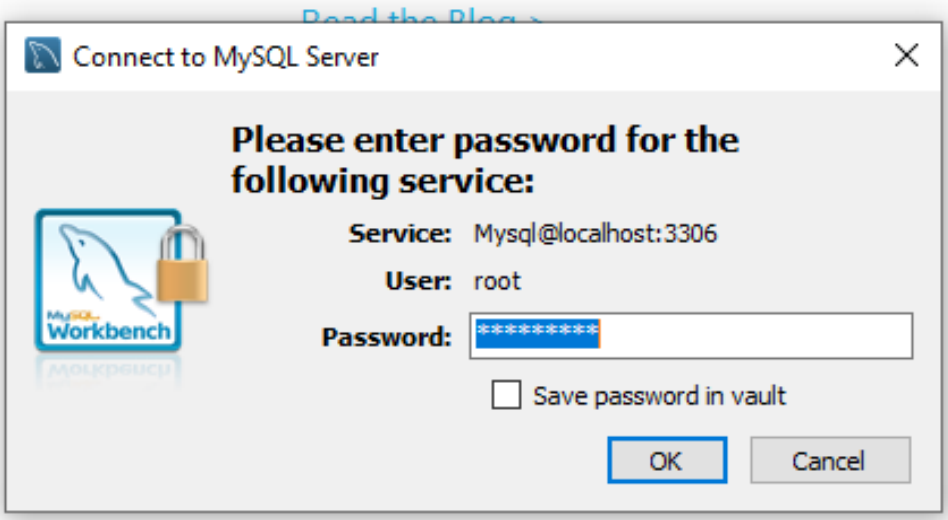
# Final Project Prompts

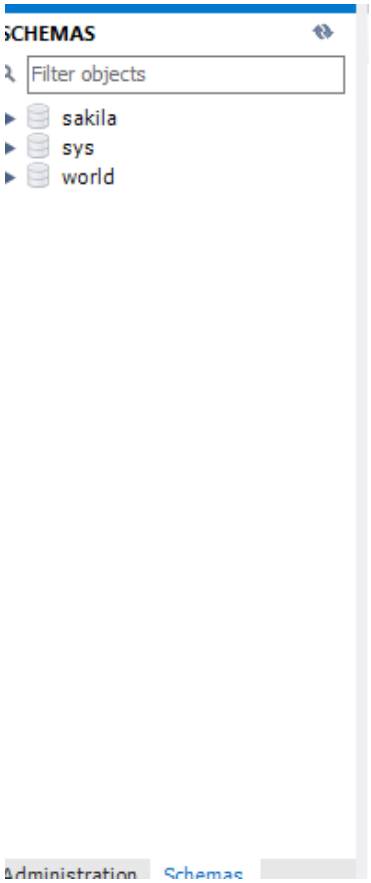
## Prompt 1 - Create the Schema/Database

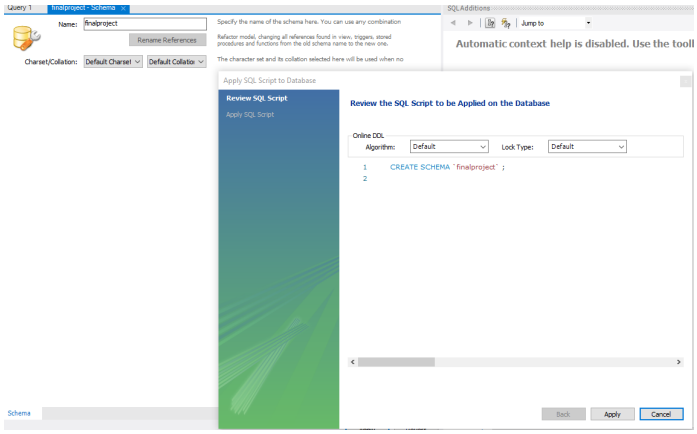
01.
02. Launch your local instance and login:



03. When MySQL Workbench loads, the **Schemas** displays on the left-hand side of the window.




04. In the Schemas panel of MySQL, **right-click** and select **Create Schema**. Note, you may not have any existing schemas listed; that is OK.
05. Let's call it 'finalproject'. Type the name in and then click the Apply button at the lower-right corner of the window.
06. Click Apply at the next screen:
07. Then click Finish
08.
09. 

**10. Label this screen capture 'Prompt 1'**

SCHEMAS

Filter objects

- finalproject
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration | Schemas

Information

**Schema: finalproject**

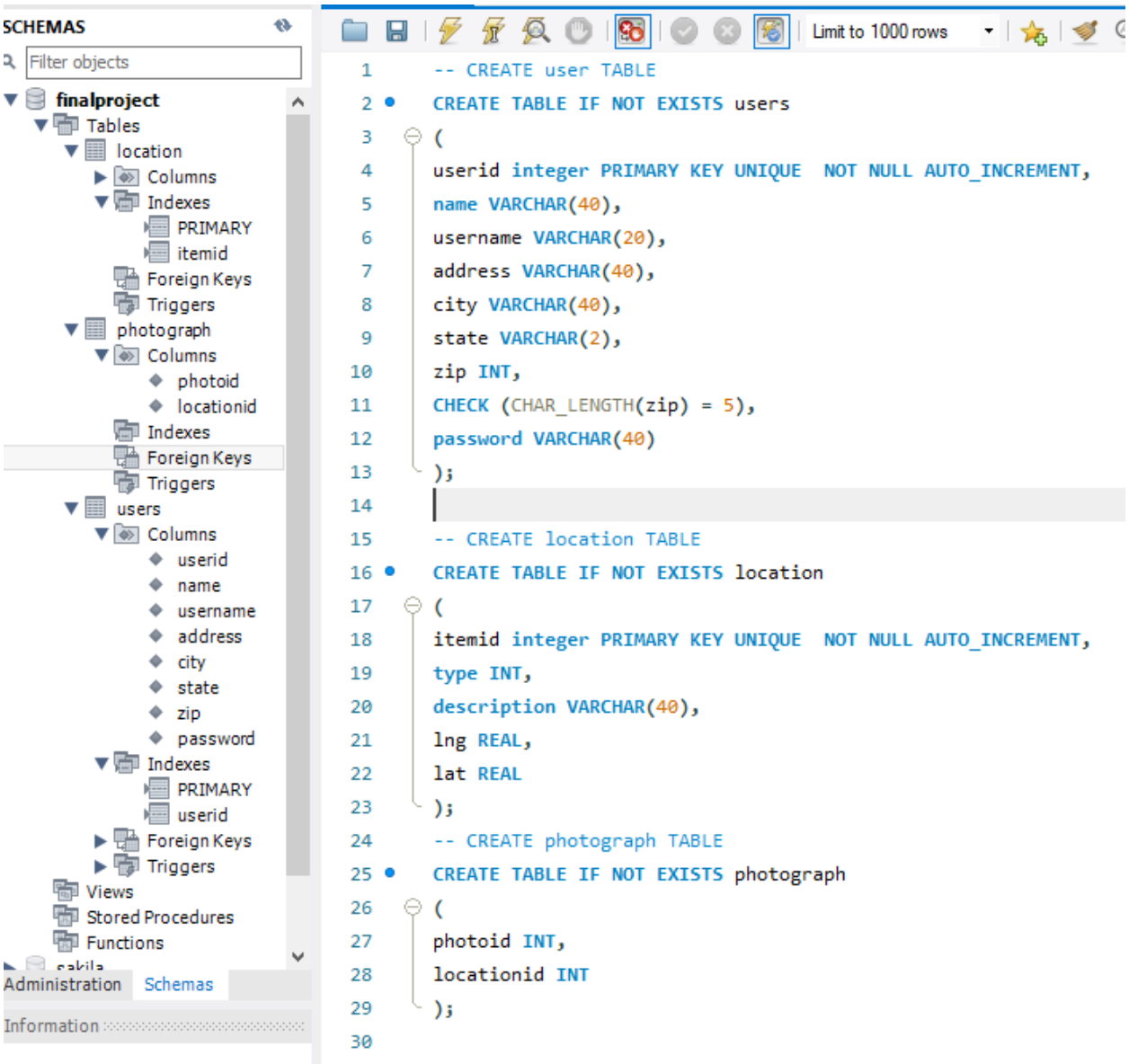
1

Output

Action Output

| # | Time | Action |
|---|---|---|
| 1 | 22:55:18 | Apply changes to finalprojec |
| 2 | 22:57:09 | Apply changes to finalprojec |

## 11. Prompt 2 - Create Tables



## 12. Prompt 3 - Alter Tables

```sql
-- Prompt 3 - Alter Tables
ALTER TABLE location MODIFY type INT NOT NULL;
ALTER TABLE location MODIFY description VARCHAR(40) NOT NULL;
ALTER TABLE location MODIFY lng REAL NOT NULL;
ALTER TABLE location MODIFY lat REAL NOT NULL;
ALTER TABLE users MODIFY name VARCHAR(40) NOT NULL;
ALTER TABLE users MODIFY userid VARCHAR(40) NOT NULL;
ALTER TABLE users MODIFY password VARCHAR(40) NOT NULL;
ALTER TABLE photograph MODIFY photoid INT NOT NULL;
ALTER TABLE photograph MODIFY locationid INT NOT NULL;
```

## 13. Prompt 4 - Create Index

```sql
-- Prompt 4 - Create Index
CREATE UNIQUE INDEX id ON users (userid);
CREATE UNIQUE INDEX photo_id ON users (photoid);
```

## 14. Prompt 5 - Enter Data

```sql
INSERT INTO users (userid, name, username, address, city, state, zip, password)
VALUES
(1, "Bonnie Buntcake", "bbunt", "6709 Wonder Street", "Wonderbread", "OH", 46106, "eclectic"),
(2, "Sam Smarf", "ssmarf", "356 A Street", "Beefy", "PA", 19943, "swimming"),
(3, "Wendy Grog", "wgrog", "900 Star Street", "Mary", "MD", 21340, "wells"),
(4, "Joe Jogger", "jjogger", "183713 N North Street", "Norther", "WV", 51423, "tarts");

SELECT * FROM users;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| userid | name | username | address | city | state | zip | password |
|---|---|---|---|---|---|---|---|
| 1 | Bonnie Buntcake | bbunt | 6709 Wonder Street | Wonderbread | OH | 46106 | eclectic |
| 2 | Sam Smarf | ssmarf | 356 A Street | Beefy | PA | 19943 | swimming |
| 3 | Wendy Grog | wgrog | 900 Star Street | Mary | MD | 21340 | wells |
| 4 | Joe Jogger | jjogger | 183713 N North Street | Norther | WV | 51423 | tarts |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

## 15. Prompt 6 - Count Rows

```
-- Prompt 6 - Count Rows
SELECT count(*)
FROM users;
```

Result Grid | Filter Rows:

| count(*) |
|---|
| 4 |

## 16. Prompt 7 - Add Column

```
ALTER TABLE photograph ADD COLUMN userid INT NOT NULL AFTER locationid;
```

Output

Action Output

| # | Time | Action | |
|---|---|---|---|
| 23 | 00:01:29 | ALTER TABLE location MODIFY lat REAL NOT NULL | 0 |
| 24 | 00:02:24 | ALTER TABLE photograph MODIFY locationid INT NOT NULL | 0 |
| 25 | 00:20:04 | INSERT INTO users (userid, name, username, address, city, state, zip, password) VALUES (1,"Bonnie Buntcake","bbunt", "6709 Wonder Street","Wo... | 4 |
| 26 | 00:22:16 | SELECT * FROM users LIMIT 0, 1000 | 4 |
| 27 | 00:26:01 | SELECT count(*) FROM users LIMIT 0, 1000 | 1 |
| 28 | 00:33:20 | ALTER TABLE photograph ADD COLUMN userid INT NOT NULL AFTER locationid | 0 |

**–verify**

```
SELECT * FROM photograph;
```

Result Grid | Filter Rows:

| photoid | locationid | userid |
|---|---|---|

## 17. Prompt 8 - Issue with New Column

This statement will work in the program, however due to the constraints that have been established the primary key will take precedence and all that will be entered will register as the userid. So the photo will be taken by a user in the user FROM the user TABLE. The Constraint must be assigned to establish a connect between the user and the photograph table using a FOREIGN Key constraint in the users TABLE. The statement was originally an INTEGER then changed to a VARCHAR(40) now back to a VARCHAR(40) and the the

```
ALTER TABLE photograph
ADD FOREIGN KEY (userid)
REFERENCES users(userid);
```

```
--  Prompt 8 - Issue with New Column
ALTER TABLE photograph MODIFY userid VARCHAR(40) NOT NULL;
ALTER TABLE photograph
ADD FOREIGN KEY (userid)
REFERENCES users(userid);
```

## 18. Prompt 9 - Location and Photograph Table Updates

```
-- Prompt 9 - Location and Photograph Table Updates
INSERT INTO location (type, description,    lng,    la
VALUES
(1, "Independence Hall",    794.35, 651.43),
(2, "6709 Wonder Street",   323.41, 412.22),
(1, "Sunrise",              221.45, 132.43),
(2, "356 A Street",         123.32, 222.43),
(1, "Mountains",            34.12,  87.99),
(2, "900 Star Street",      1071.9, 206.45),
(1, "Moonrise",             816.2,  111.2),
(2, "183714 N North Street", 76.11, 11.176);

SELECT *
FROM location;
```

Result Grid | Filter Rows: | Edit: | Export/I

| itemid | type | description | lng | lat |
|---|---|---|---|---|
| 5 | 1 | Mountains | 34.12 | 87.99 |
| 6 | 2 | 900 Star Street | 1071.9 | 206.45 |
| 7 | 1 | Moonrise | 816.2 | 111.2 |
| 8 | 2 | 183714 N North Street | 76.11 | 11.176 |
| NULL | NULL | NULL | NULL | NULL |

ocation 6

Output

Action Output

| # | Time | Action |
|---|---|---|
| 39 | 01:06:36 | ALTER TABLE photograph ADD FOREIGN KEY (userid) REFERENCE |
| 40 | 01:27:09 | INSERT INTO locations (itemid,type,description,lng,lat) VALUES (1,"Ind |
| 41 | 01:27:54 | INSERT INTO location (itemid,type,description,lng,lat) VALUES (1,"Inde |
| 42 | 01:30:28 | INSERT INTO location (type,description,lng,lat) VALUES (1,"Independe |

```
SELECT *
FROM location;

INSERT INTO photograph(photoid, locationid, userid)
VALUES
(1, 11, 1),
(2, 9,  1),
(3, 3,  3),
(4, 8,  4);

SELECT *
FROM photograph;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| photoid | locationid | userid |
|---|---|---|
| 1 | 11 | 1 |
| 2 | 9 | 1 |
| 3 | 3 | 3 |
| 4 | 8 | 4 |

19. **Prompt 10 - Users**

```
-- Prompt 10 - Users
SELECT name FROM users;
```

Result Grid | Filter Rows:

| name |
|---|
| Bonnie Buntcake |
| Sam Smarf |
| Wendy Grog |
| Joe Jogger |

## 20. Prompt 11 - Who's Taking Pictures?

```
-- Prompt 10 - Users
SELECT DISTINCT name FROM users;

-- Prompt 11 - Who's Taking Pictures?
SELECT DISTINCT name
 FROM users AS u,  photograph AS p
 WHERE u.userid IN(p.userid);
```

Result Grid | Filter Rows: | Export:

| name |
|---|
| Bonnie Buntcake |
| Wendy Grog |
| Joe Jogger |

## 21. Prompt 12 - Unique Names

```

-- Prompt 12 - Unique Names
SELECT DISTINCT name
 FROM users AS u,  photograph AS p
 WHERE u.userid IN(p.userid);
```

Result Grid | Filter Rows: | Export: | Wra

| name |
|---|
| Bonnie Buntcake |
| Wendy Grog |
| Joe Jogger |

Bibliography

Database Management - Assignment: Database System. (2018, April 23). Retrieved from https://study.com/academy/lesson/database-management-assignment-database-system.html.

Github: https://github.com/mensahTribeWeb/DB_Fundamentals_Com_Sci_303.git