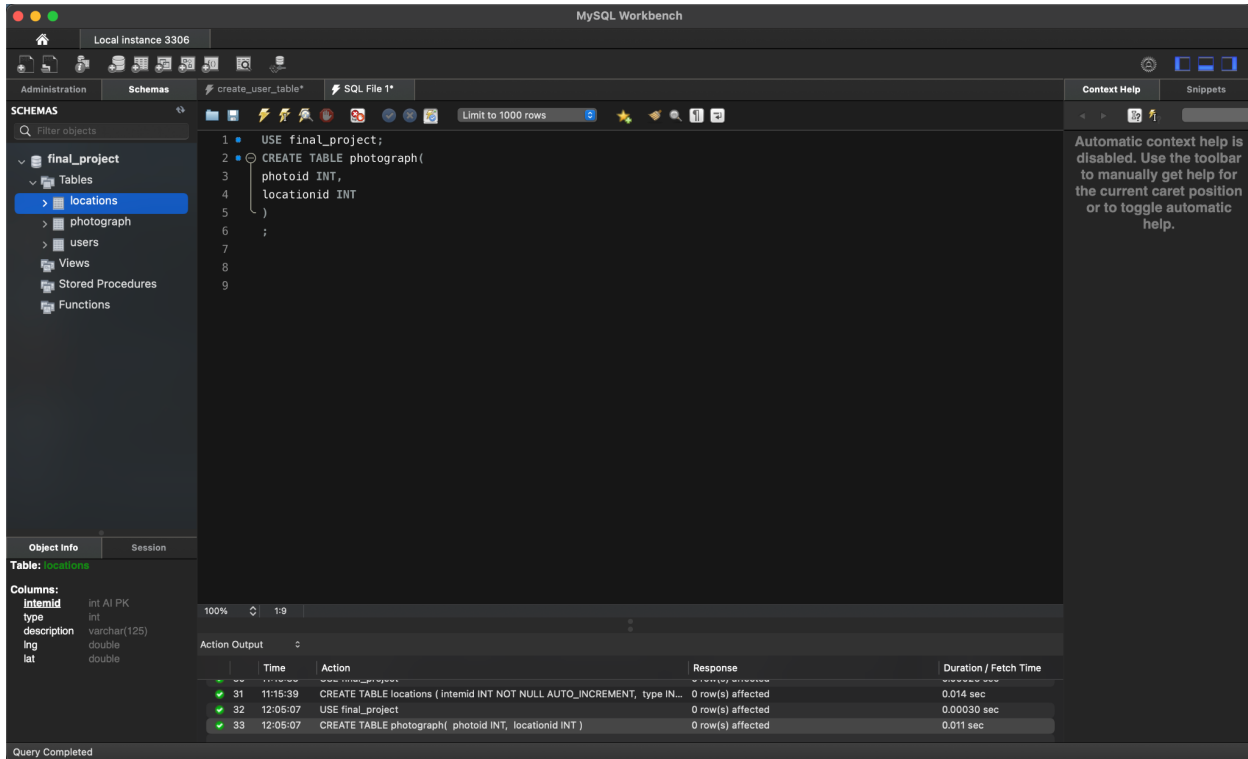
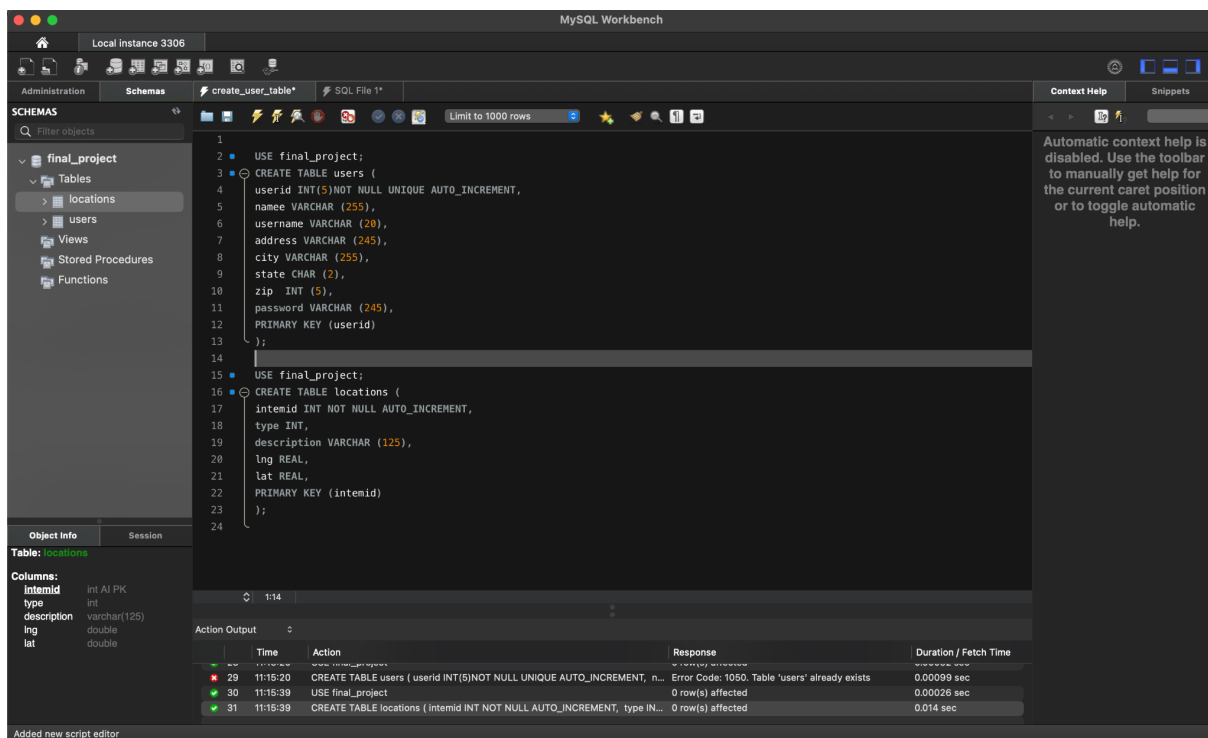


## PROMPT 1



## PROMPT 2



```
USE final_project;
CREATE TABLE photograph(
photoid INT,
locationid INT
)
;
```

```
USE final_project;
CREATE TABLE users (
userid INT(5) NOT NULL UNIQUE AUTO_INCREMENT,
namee VARCHAR (255),
username VARCHAR (20),
address VARCHAR (245),
city VARCHAR (255),
state CHAR (2),
zip INT (5),
password VARCHAR (245),
PRIMARY KEY (userid)
);
```

```
USE final_project;
CREATE TABLE locations (
intemid INT NOT NULL AUTO_INCREMENT,
type INT,
description VARCHAR (125),
lng REAL,
lat REAL,
PRIMARY KEY (intemid)
);
```

## PROMPT 3 - ALTER TABLE

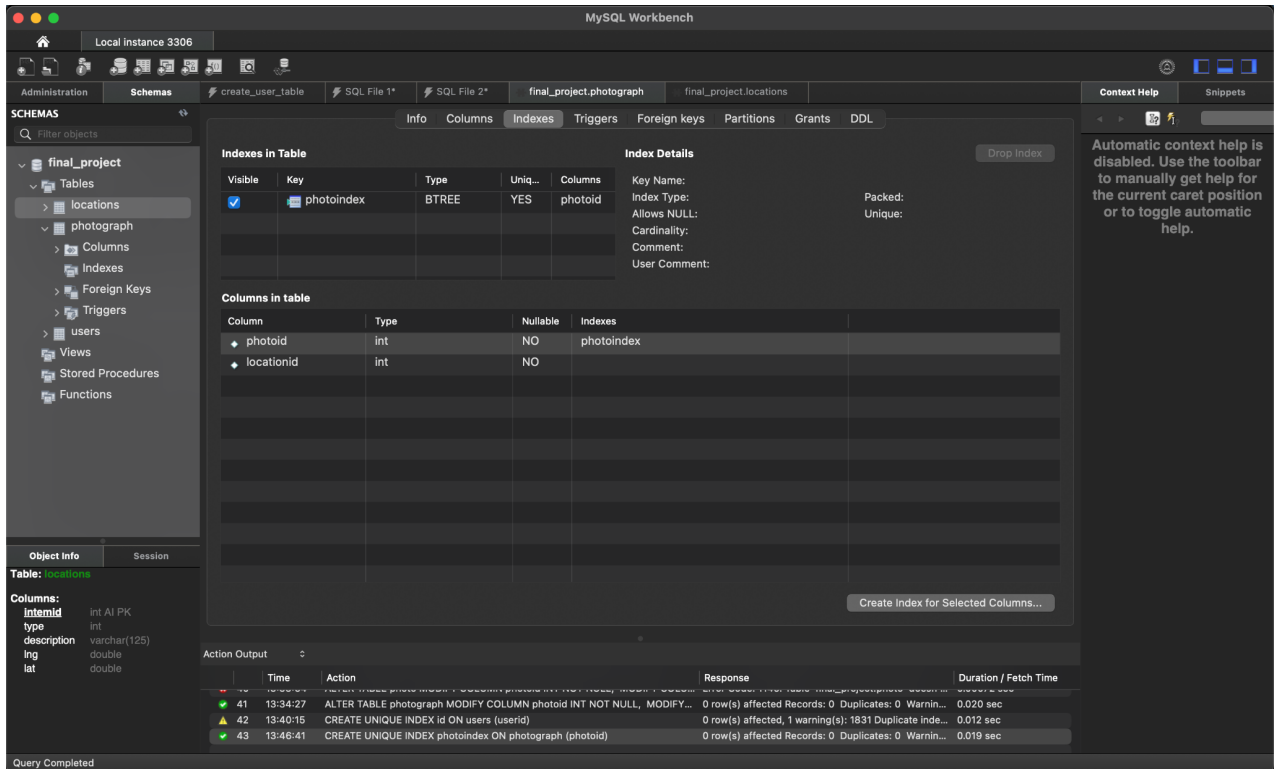
The screenshot shows the MySQL Workbench interface with the 'final\_project' database selected. The SQL editor contains three ALTER TABLE queries to modify columns in the 'locations', 'users', and 'photograph' tables. The 'Object Info' pane on the left shows the 'lat' column in the 'locations' table. The 'Action Output' pane at the bottom shows the execution results of the queries.

```
1
2
3 ALTER TABLE locations
4   MODIFY COLUMN type INT NOT NULL,
5   MODIFY COLUMN description VARCHAR (125) NOT NULL,
6   MODIFY COLUMN lng REAL NOT NULL,
7   MODIFY COLUMN lat REAL NOT NULL;
8
9 ALTER TABLE users
10  MODIFY COLUMN namee VARCHAR (255) NOT NULL,
11  MODIFY COLUMN username VARCHAR (20) NOT NULL,
12  MODIFY COLUMN password VARCHAR (255) NOT NULL;
13
14 ALTER TABLE photograph
15  MODIFY COLUMN photoid INT NOT NULL,
16  MODIFY COLUMN locationid INT NOT NULL;
17
```

**Action Output**

Time	Action	Response	Duration / Fetch Time
39 13:33:35	ALTER TABLE users MODIFY COLUMN namee VARCHAR (255) NOT NULL, M...	0 row(s) affected Records: 0 Duplicates: 0 Warnin...	0.027 sec
40 13:33:54	ALTER TABLE photo MODIFY COLUMN photoid INT NOT NULL, MODIFY COLU...	Error Code: 1146 Table 'final_project.photo' doesn't...	0.00072 sec
41 13:34:27	ALTER TABLE photograph MODIFY COLUMN photoid INT NOT NULL, MODIFY...	0 row(s) affected Records: 0 Duplicates: 0 Warnin...	0.020 sec

Query Completed

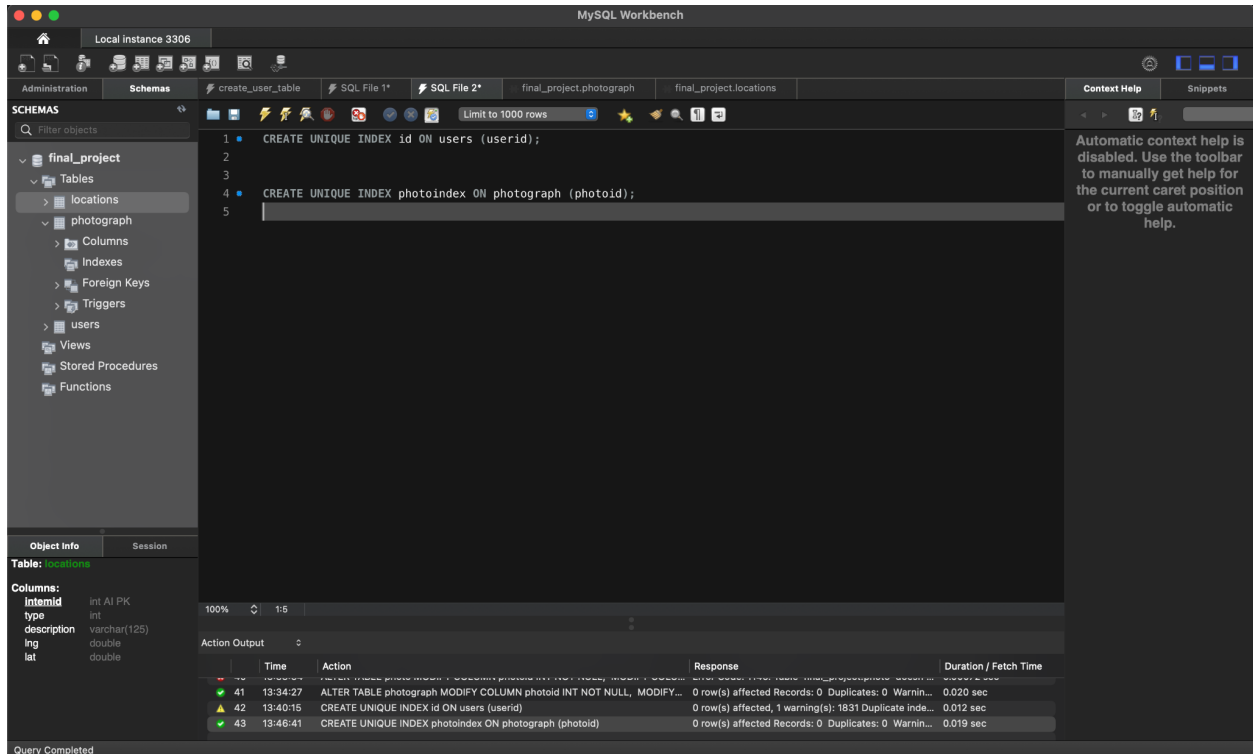


```
ALTER TABLE locations
MODIFY COLUMN type INT NOT NULL,
MODIFY COLUMN description VARCHAR (125) NOT NULL,
MODIFY COLUMN lng REAL NOT NULL,
MODIFY COLUMN lat REAL NOT NULL;
```

```
ALTER TABLE users
MODIFY COLUMN namee VARCHAR (255) NOT NULL,
MODIFY COLUMN username VARCHAR (20) NOT NULL,
MODIFY COLUMN password VARCHAR (255) NOT NULL;
```

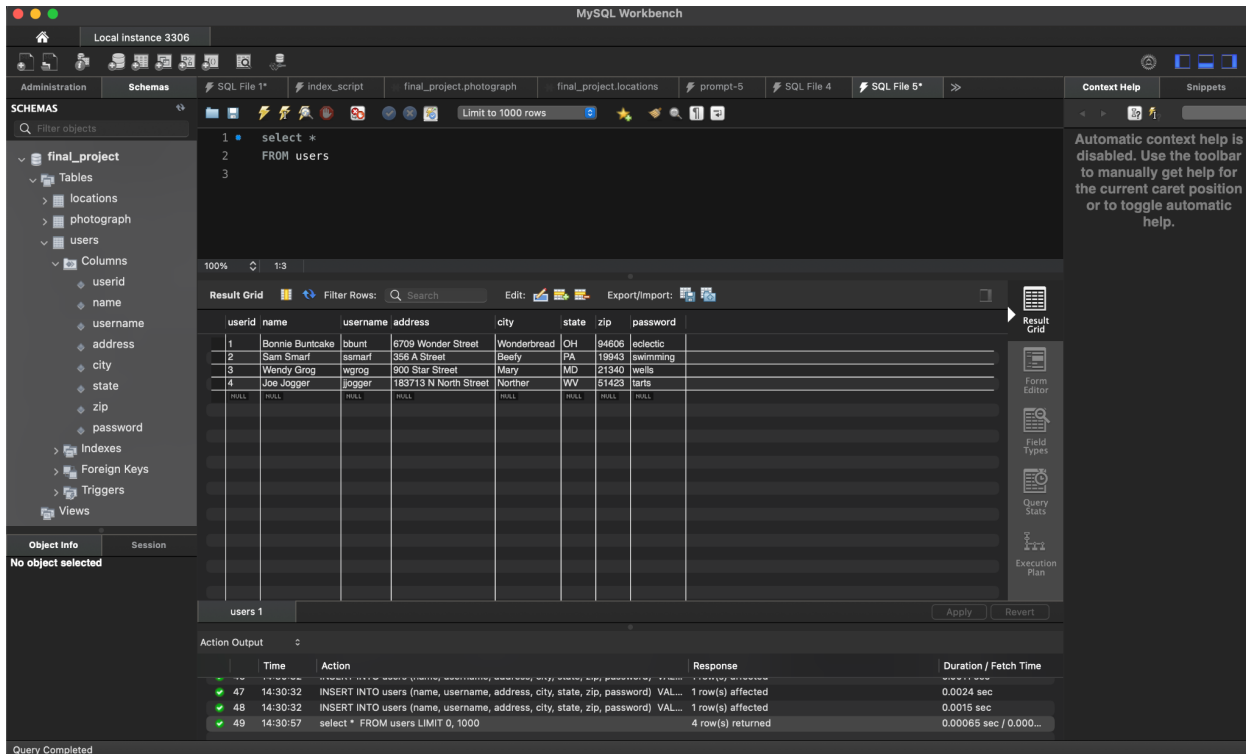
```
ALTER TABLE photograph
MODIFY COLUMN photoid INT NOT NULL,
MODIFY COLUMN locationid INT NOT NULL;
```

## Prompt 4 - Create Index



CREATE UNIQUE INDEX id ON users (userid);  
CREATE UNIQUE INDEX photoindex ON photograph (photoid);

## PROMPT 5

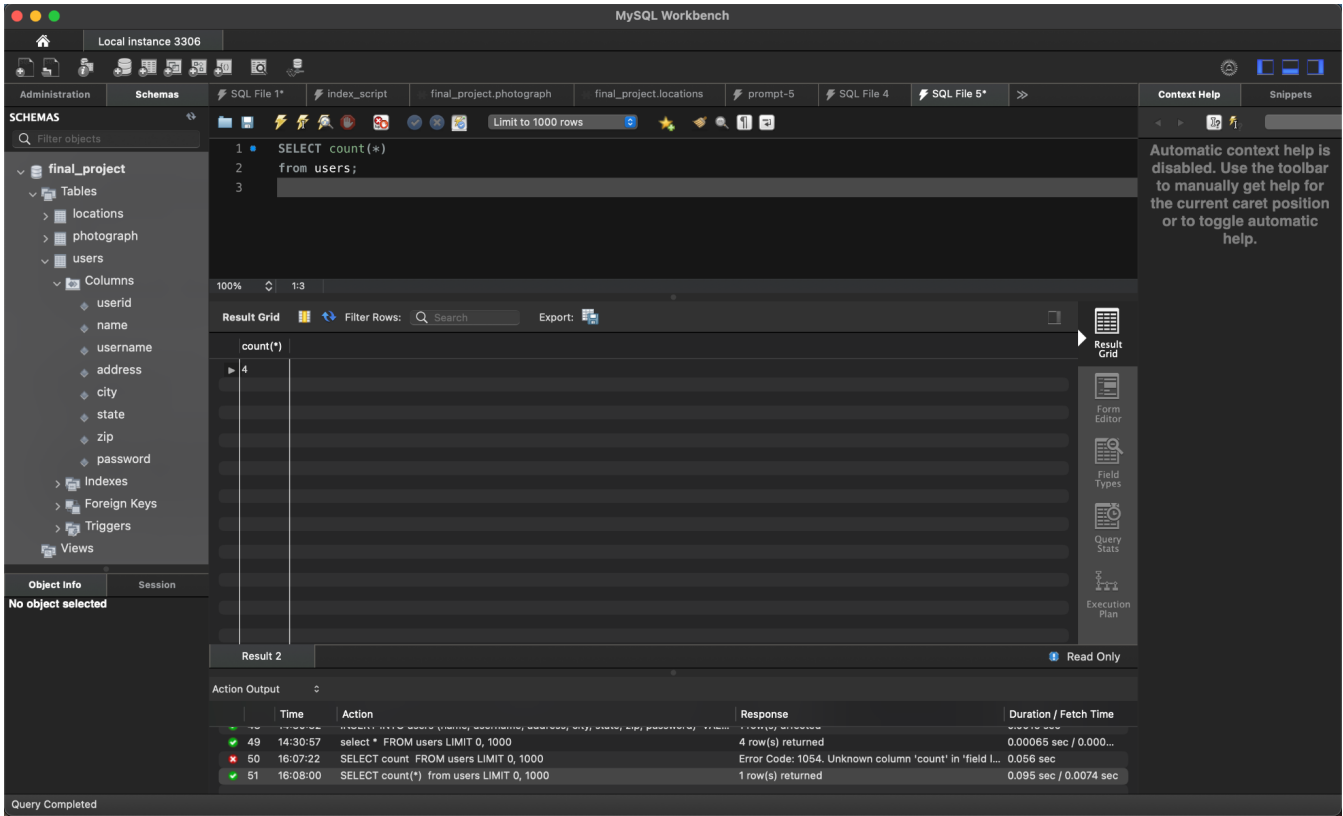


```
INSERT INTO users (name, username, address, city, state, zip, password)
VALUES ('Bonnie Buntcake', 'bbunt', '6709 Wonder Street','Wonderbread','OH','94606','eclectic'
)
;
INSERT INTO users (name, username, address, city, state, zip, password)
VALUES ('Sam Smarf', 'ssmarf', '356 A Street','Beefy','PA','19943','swimming')

;
INSERT INTO users (name, username, address, city, state, zip, password)
VALUES ('Wendy Grog','wgrog','900 Star Street','Mary','MD','21340','wells')

;
INSERT INTO users (name, username, address, city, state, zip, password)
VALUES ('Joe Jogger','jjogger','183713 N North Street','Norther','WV','51423','tarts')
;
```

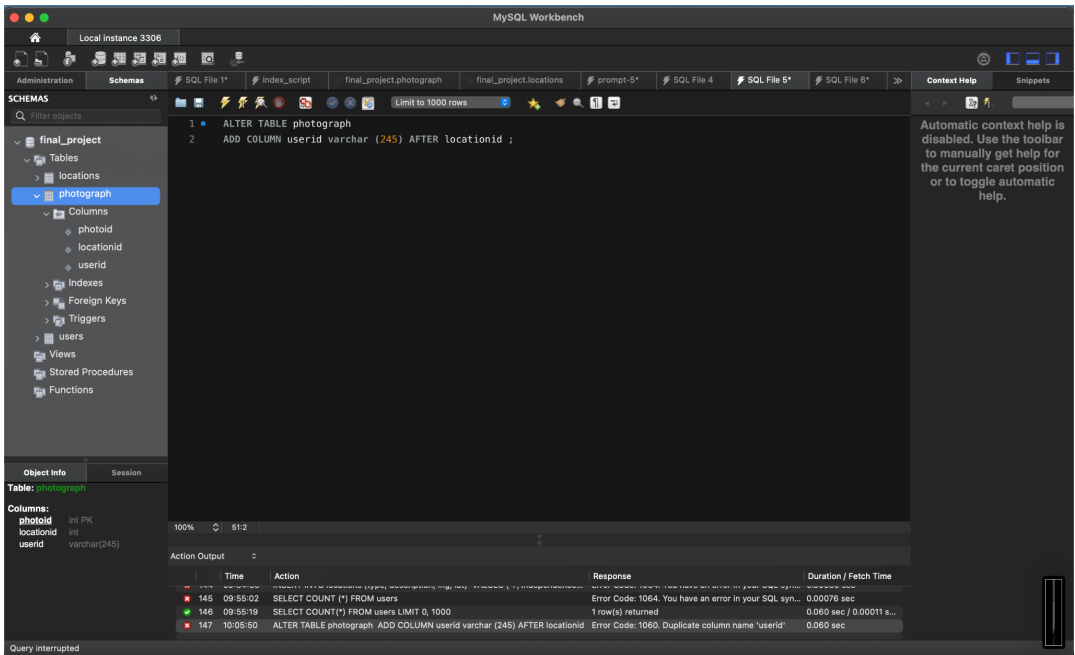
PROMPT 6



SELECT COUNT(\*)  
FROM users  
;

PROMPT 7

ALTER TABLE photograph  
ADD COLUMN userid varchar (245) AFTER locationid



## PROMPT 8

In order to ensure integrity for the userid column it would need to be made into a foreign key. The foreign key would be tied to the primary key userid column in the users table. This would ensure that the data link between the two would not be lost. Foreign and Primary keys are integral to relational databases and show the relationships between tables. Since we have already created the user id column in the photograph table. We would need to use ALTER TABLE, ADD FOREIGN KEY and REFERENCES. This would be the code we would need to use to add the relationship between the users table and the photograph

## PROMPT 9

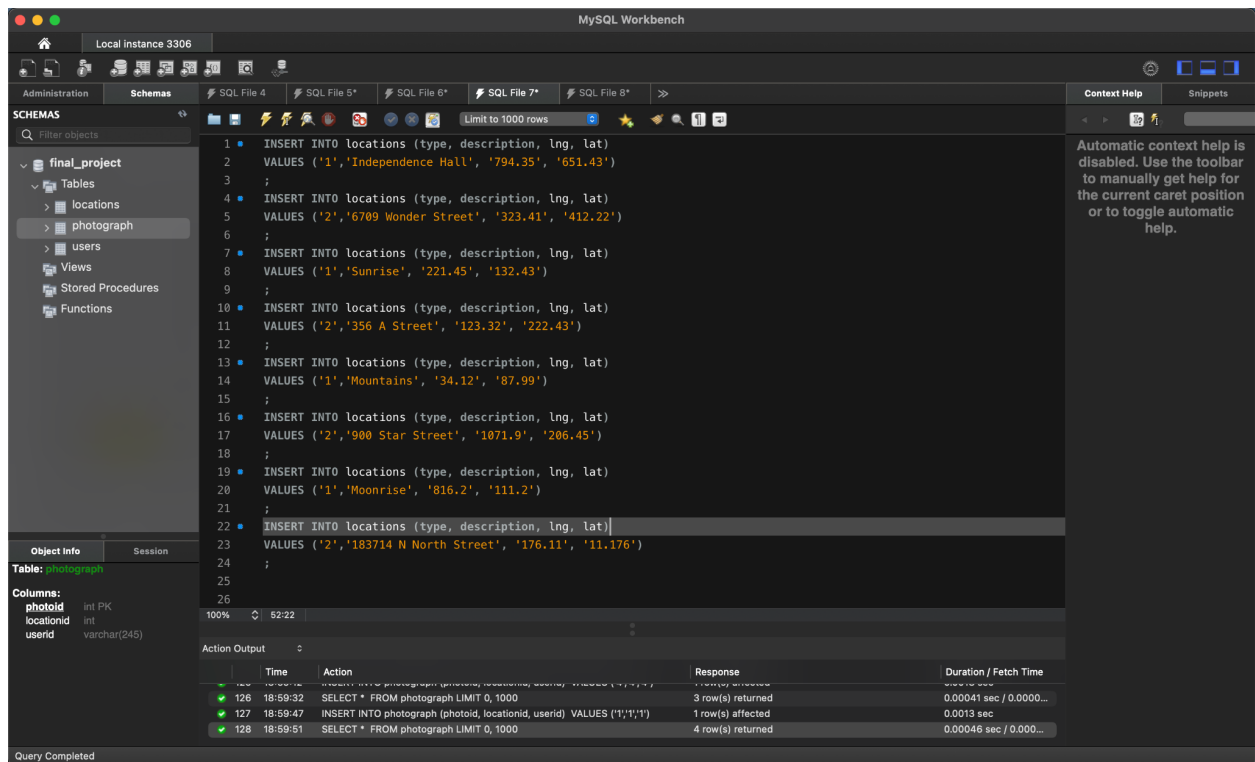
```
INSERT INTO locations (type, description, lng, lat)
VALUES ('1','Independence Hall', '794.35', '651.43')
;
INSERT INTO locations (type, description, lng, lat)
VALUES ('2','6709 Wonder Street', '323.41', '412.22')
;
INSERT INTO locations (type, description, lng, lat)
VALUES ('1','Sunrise', '221.45', '132.43')
;
INSERT INTO locations (type, description, lng, lat)
VALUES ('2','356 A Street', '123.32', '222.43')
;
INSERT INTO locations (type, description, lng, lat)
VALUES ('1','Mountains', '34.12', '87.99')
;
INSERT INTO locations (type, description, lng, lat)
VALUES ('2','900 Star Street', '1071.9', '206.45')
;
INSERT INTO locations (type, description, lng, lat)
VALUES ('1','Moonrise', '816.2', '111.2')
;
INSERT INTO locations (type, description, lng, lat)
VALUES ('2','183714 N North Street', '176.11', '11.176')
;
INSERT INTO photograph (photoid, locationid, userid)
VALUES ('1','1','1')
;
INSERT INTO photograph (photoid, locationid, userid)
VALUES ('2','2','1')
;
```

```
INSERT INTO photograph (photoid, locationid, userid)
VALUES ('3','3','3')
```

```
;
```

```
INSERT INTO photograph (photoid, locationid, userid)
VALUES ('4','4','4')
```

```
;
```



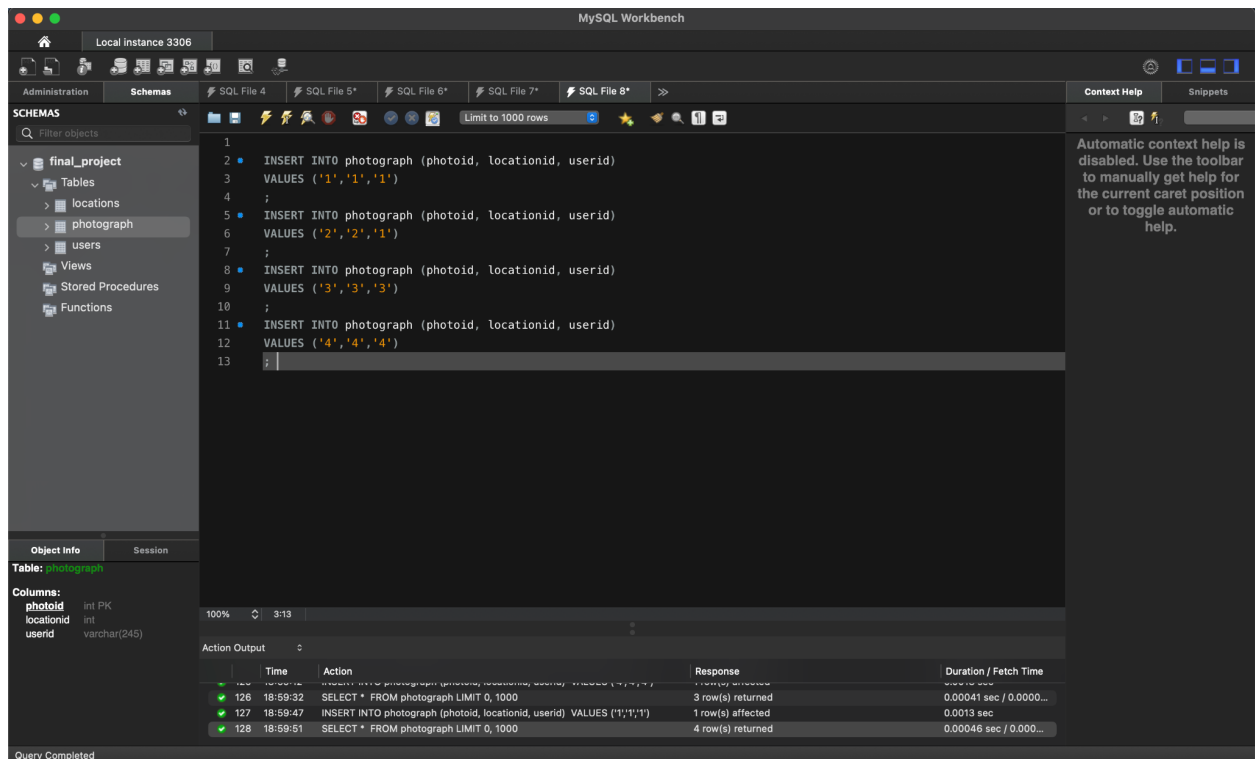
MySQL Workbench interface showing a SQL query editor with multiple INSERT statements for the 'locations' table. The query is executed, and the Action Output pane shows the results of the execution.

Query Editor Content:

```
1 INSERT INTO locations (type, description, lng, lat)
2 VALUES ('1','Independence Hall', '794.35', '651.43')
3 ;
4 INSERT INTO locations (type, description, lng, lat)
5 VALUES ('2','6709 Wonder Street', '323.41', '412.22')
6 ;
7 INSERT INTO locations (type, description, lng, lat)
8 VALUES ('1','Sunrise', '221.45', '132.43')
9 ;
10 INSERT INTO locations (type, description, lng, lat)
11 VALUES ('2','356 A Street', '123.32', '222.43')
12 ;
13 INSERT INTO locations (type, description, lng, lat)
14 VALUES ('1','Mountains', '34.12', '87.99')
15 ;
16 INSERT INTO locations (type, description, lng, lat)
17 VALUES ('2','900 Star Street', '1071.9', '206.45')
18 ;
19 INSERT INTO locations (type, description, lng, lat)
20 VALUES ('1','Moonrise', '816.2', '111.2')
21 ;
22 INSERT INTO locations (type, description, lng, lat)
23 VALUES ('2','183714 N North Street', '176.11', '11.176')
24 ;
25 ;
26
```

Action Output:

	Time	Action	Response	Duration / Fetch Time
✓	18:59:32	SELECT * FROM photograph LIMIT 0, 1000	3 row(s) returned	0.00041 sec / 0.0000...
✓	18:59:47	INSERT INTO photograph (photoid, locationid, userid) VALUES ('1','1','1')	1 row(s) affected	0.0013 sec
✓	18:59:51	SELECT * FROM photograph LIMIT 0, 1000	4 row(s) returned	0.00046 sec / 0.000...



MySQL Workbench interface showing a SQL query editor with multiple INSERT statements for the 'photograph' table. The query is executed, and the Action Output pane shows the results of the execution.

Query Editor Content:

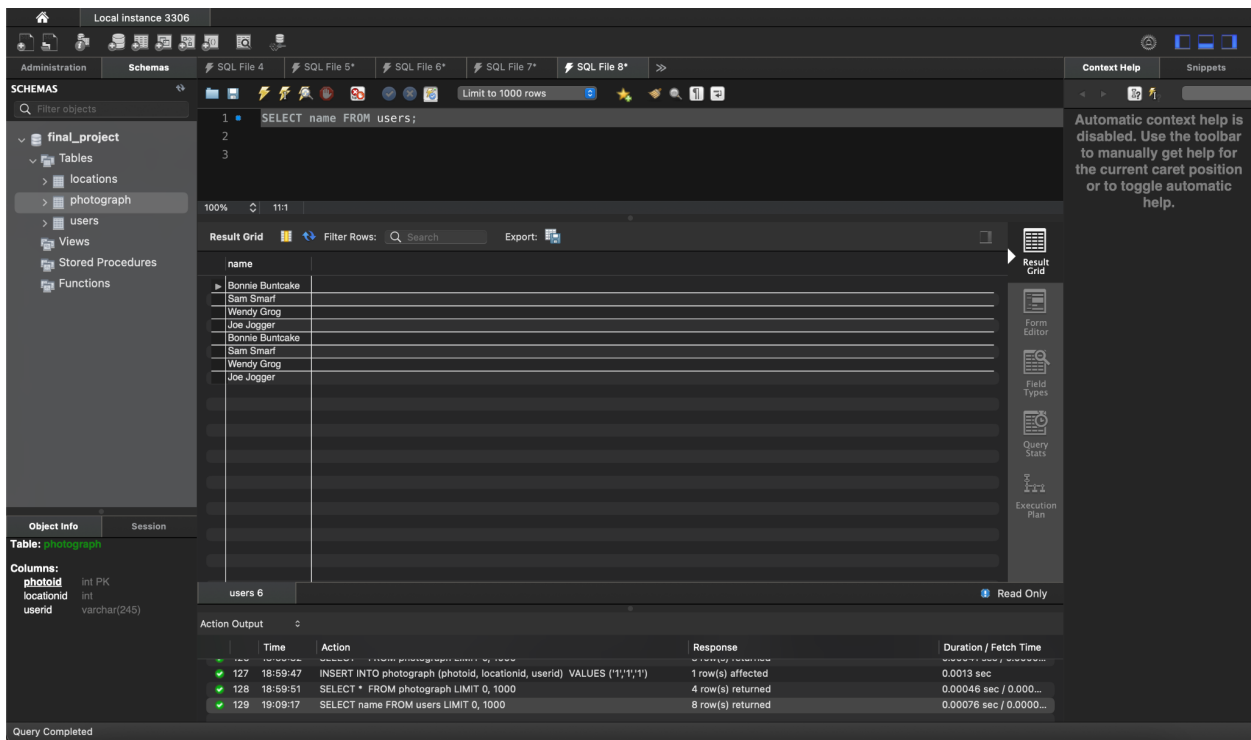
```
1
2 INSERT INTO photograph (photoid, locationid, userid)
3 VALUES ('1','1','1')
4 ;
5 INSERT INTO photograph (photoid, locationid, userid)
6 VALUES ('2','2','1')
7 ;
8 INSERT INTO photograph (photoid, locationid, userid)
9 VALUES ('3','3','3')
10 ;
11 INSERT INTO photograph (photoid, locationid, userid)
12 VALUES ('4','4','4')
13 ;
14
```

Action Output:

	Time	Action	Response	Duration / Fetch Time
✓	18:59:32	SELECT * FROM photograph LIMIT 0, 1000	3 row(s) returned	0.00041 sec / 0.0000...
✓	18:59:47	INSERT INTO photograph (photoid, locationid, userid) VALUES ('1','1','1')	1 row(s) affected	0.0013 sec
✓	18:59:51	SELECT * FROM photograph LIMIT 0, 1000	4 row(s) returned	0.00046 sec / 0.000...



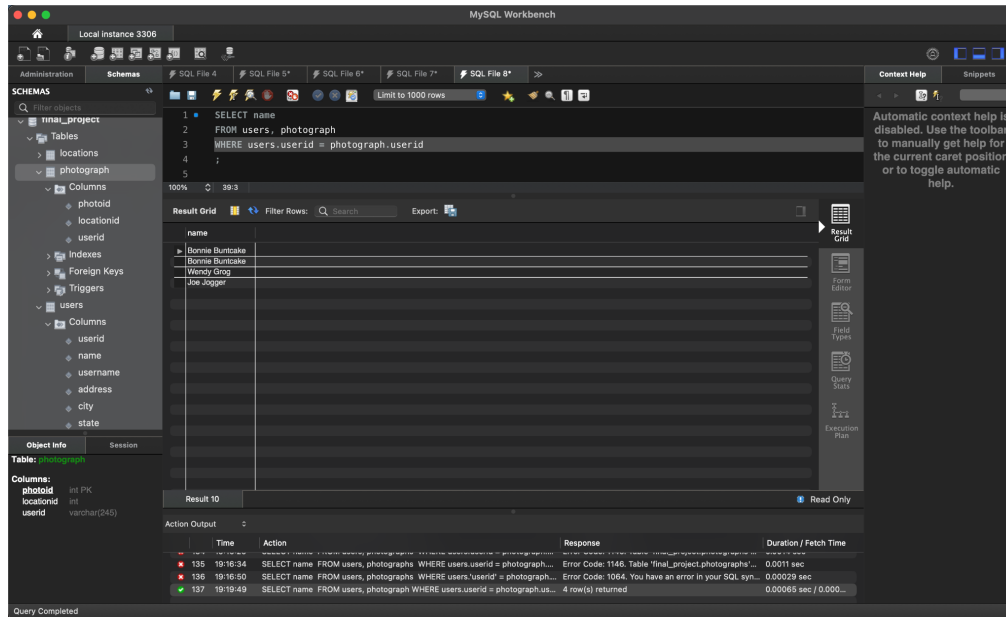
PROMPT 10



SELECT name  
FROM users, photograph  
WHERE users.userid = photograph.userid ;

## PROMPT 11

```
SELECT name
FROM users, photograph
WHERE users.userid = photograph.userid ;
```



## PROMPT 12

```
SELECT DISTINCT name
FROM users, photograph
WHERE users.userid = photograph.userid
```

