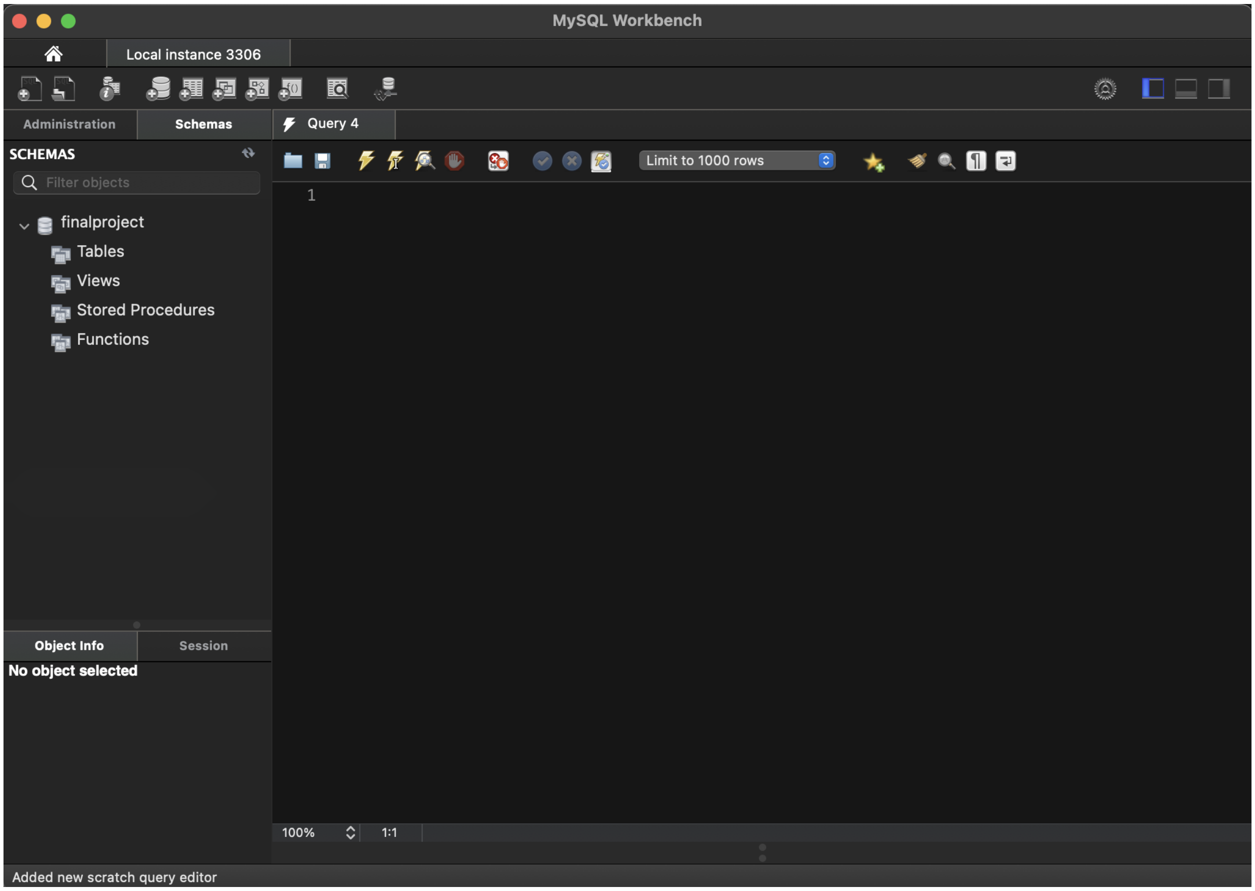
Scott Hurwitz
Comp Sci 303: Data Management Assignment: Database System

Note: the inclusion of screenshots in the document has created some long blank areas. In this case, continue scrolling down the document to find the next entry. Thank you.

Prompt 1:



No object selected

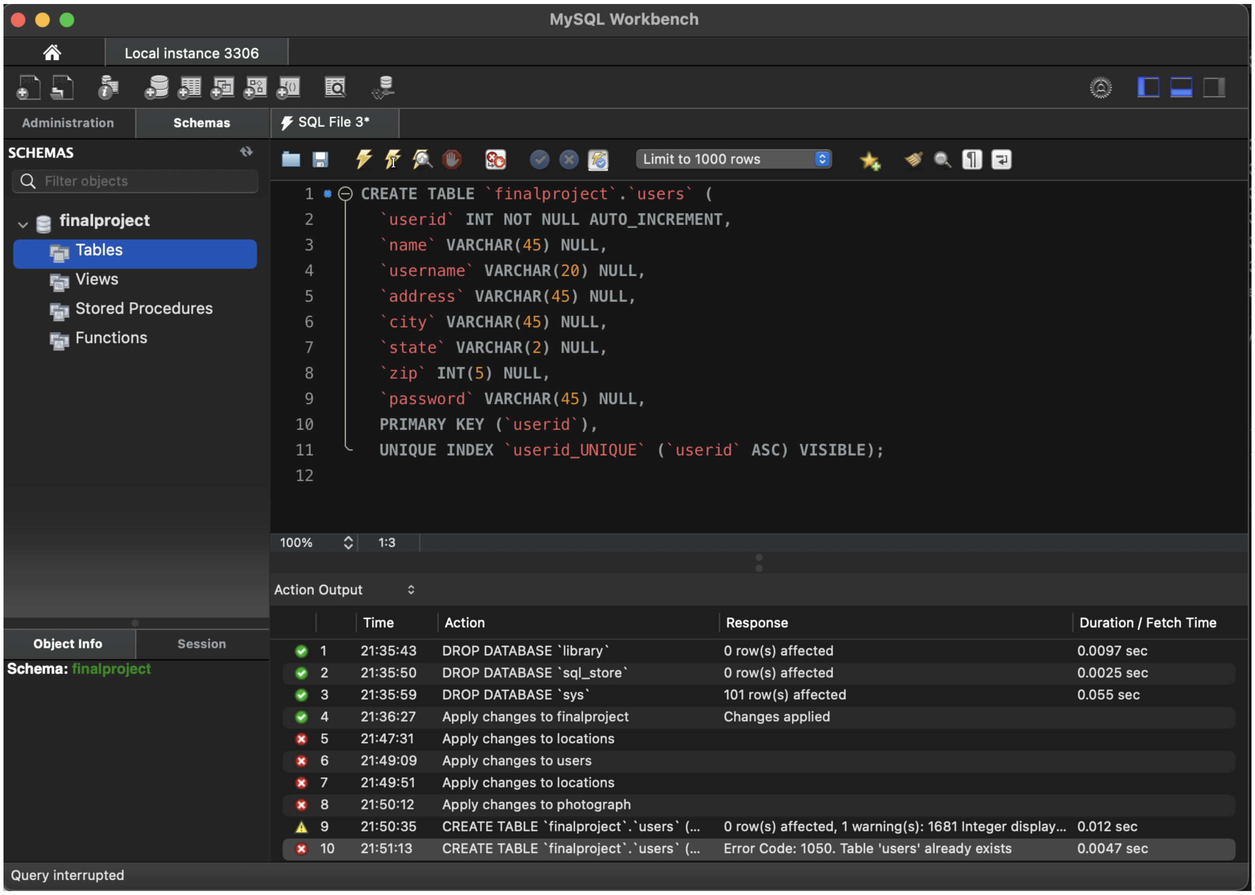
Added new scratch query editor

Prompt 2:

```
CREATE TABLE `finalproject`.`users` (
  `userid` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `username` VARCHAR(20) NULL,
  `address` VARCHAR(45) NULL,
  `city` VARCHAR(45) NULL,
  `state` VARCHAR(2) NULL,
  `zip` INT(5) NULL,
  `password` VARCHAR(45) NULL,
  PRIMARY KEY (`userid`),
  UNIQUE INDEX `userid_UNIQUE` (`userid` ASC) VISIBLE);
```

```
CREATE TABLE `finalproject`.`locations` (
  `itemid` INT NOT NULL AUTO_INCREMENT,
  `type` INT NULL,
  `description` VARCHAR(100) NULL,
  `lng` REAL NULL,
  `lat` REAL NULL);
```

```
CREATE TABLE `finalproject`.`photograph` (
  `photoid` INT NULL,
  `locationid` INT NULL);
```

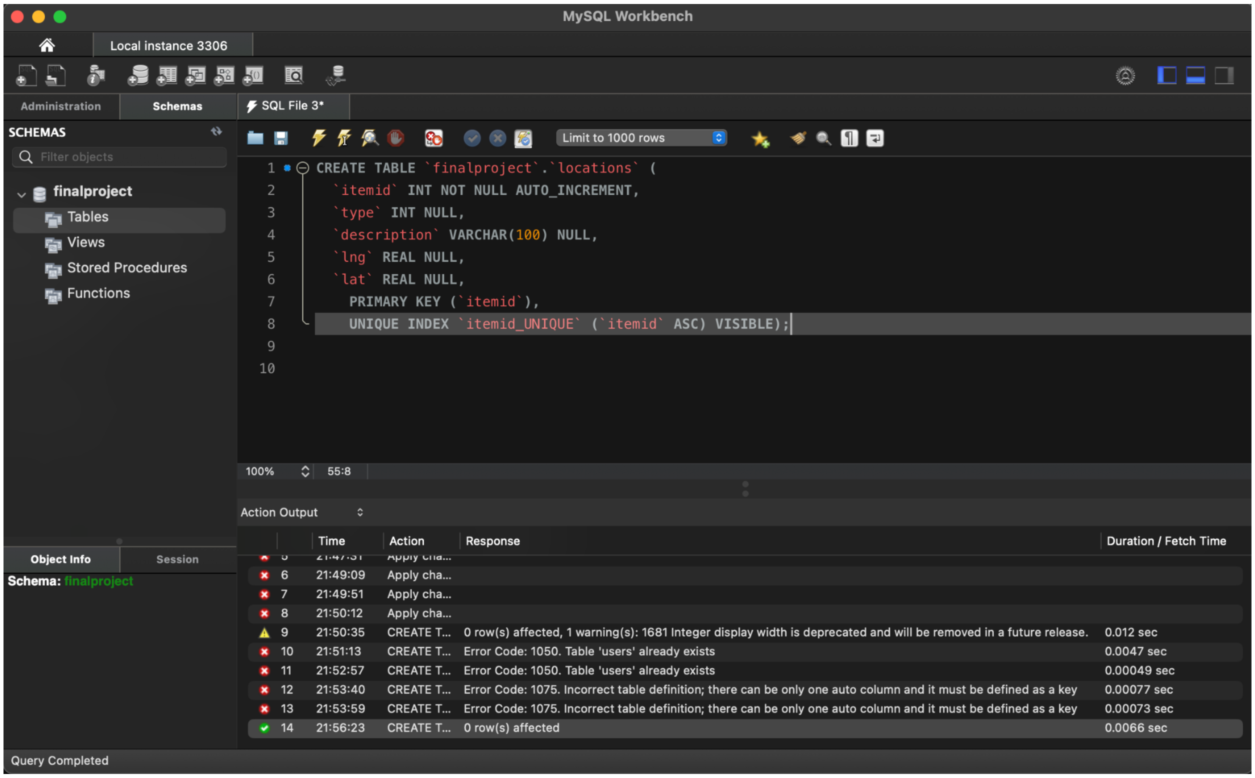


Schema: finalproject

```
CREATE TABLE `finalproject`.`users` (
  `userid` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `username` VARCHAR(20) NULL,
  `address` VARCHAR(45) NULL,
  `city` VARCHAR(45) NULL,
  `state` VARCHAR(2) NULL,
  `zip` INT(5) NULL,
  `password` VARCHAR(45) NULL,
  PRIMARY KEY (`userid`),
  UNIQUE INDEX `userid_UNIQUE` (`userid` ASC) VISIBLE);
```

		Time	Action	Response	Duration / Fetch Time
✓	1	21:35:43	DROP DATABASE `library`	0 row(s) affected	0.0097 sec
✓	2	21:35:50	DROP DATABASE `sql_store`	0 row(s) affected	0.0025 sec
✓	3	21:35:59	DROP DATABASE `sys`	101 row(s) affected	0.055 sec
✓	4	21:36:27	Apply changes to finalproject	Changes applied	
✗	5	21:47:31	Apply changes to locations		
✗	6	21:49:09	Apply changes to users		
✗	7	21:49:51	Apply changes to locations		
✗	8	21:50:12	Apply changes to photograph		
⚠	9	21:50:35	CREATE TABLE `finalproject`.`users` (...	0 row(s) affected, 1 warning(s): 1681 Integer display...	0.012 sec
✗	10	21:51:13	CREATE TABLE `finalproject`.`users` (...	Error Code: 1050. Table 'users' already exists	0.0047 sec

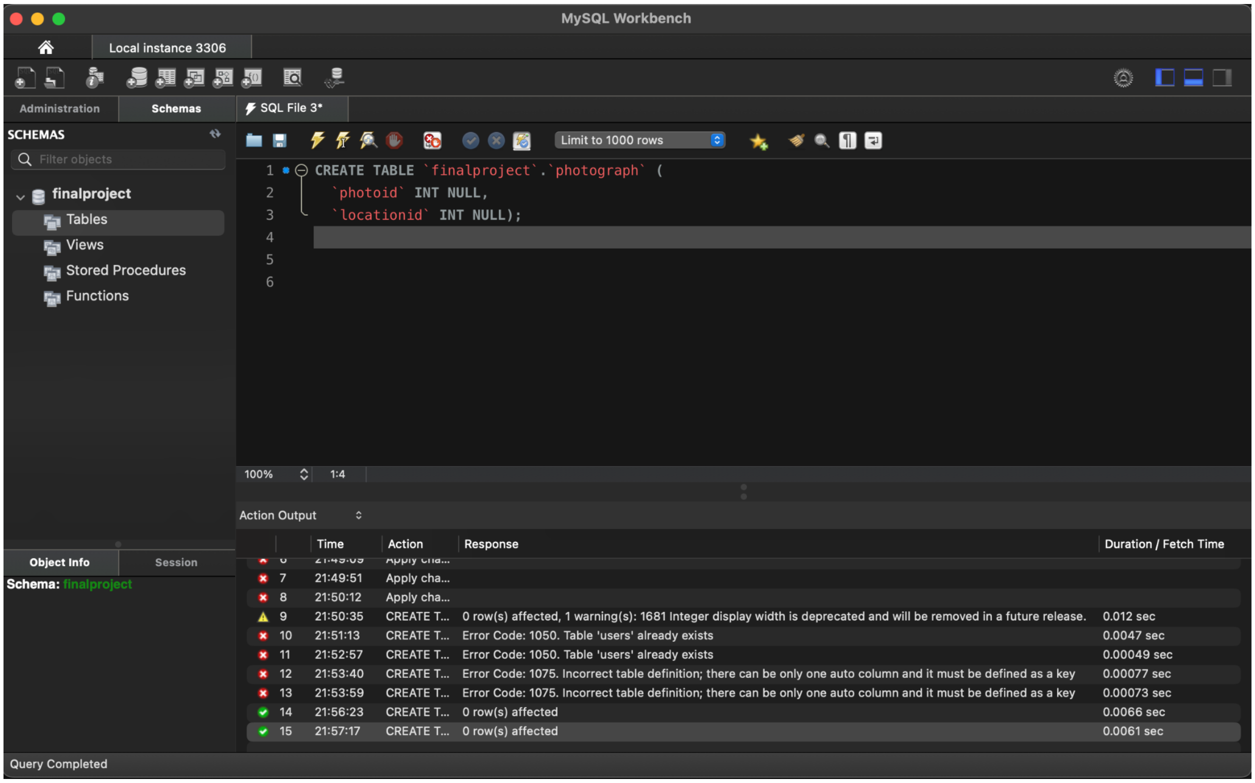
Query interrupted



```sql
CREATE TABLE `finalproject`.`locations` (
  `itemid` INT NOT NULL AUTO_INCREMENT,
  `type` INT NULL,
  `description` VARCHAR(100) NULL,
  `lng` REAL NULL,
  `lat` REAL NULL,
    PRIMARY KEY (`itemid`),
    UNIQUE INDEX `itemid_UNIQUE` (`itemid` ASC) VISIBLE);
```

Action Output

		Time	Action	Response	Duration / Fetch Time
✗	5	21:47:31	Apply cha...		
✗	6	21:49:09	Apply cha...		
✗	7	21:49:51	Apply cha...		
✗	8	21:50:12	Apply cha...		
⚠	9	21:50:35	CREATE T...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.012 sec
✗	10	21:51:13	CREATE T...	Error Code: 1050. Table 'users' already exists	0.0047 sec
✗	11	21:52:57	CREATE T...	Error Code: 1050. Table 'users' already exists	0.00049 sec
✗	12	21:53:40	CREATE T...	Error Code: 1075. Incorrect table definition; there can be only one auto column and it must be defined as a key	0.00077 sec
✗	13	21:53:59	CREATE T...	Error Code: 1075. Incorrect table definition; there can be only one auto column and it must be defined as a key	0.00073 sec
✓	14	21:56:23	CREATE T...	0 row(s) affected	0.0066 sec



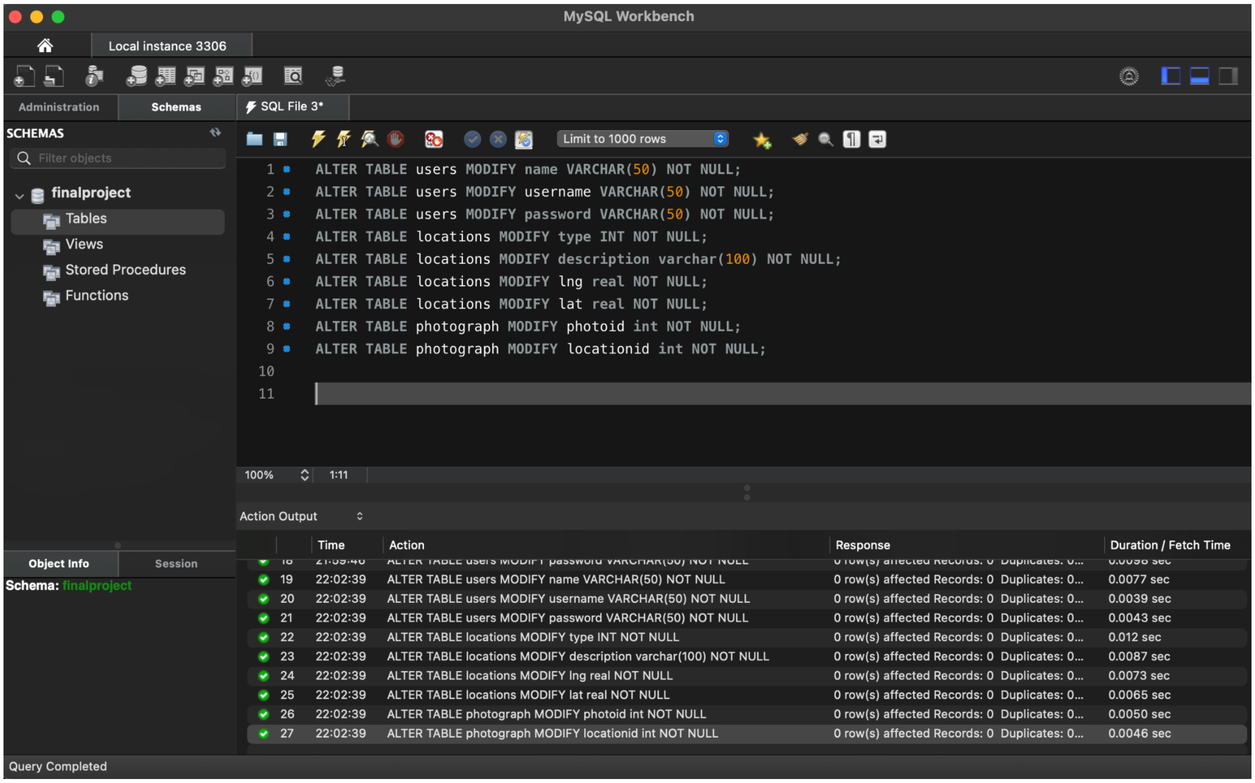
```sql
CREATE TABLE `finalproject`.`photograph` (
  `photoid` INT NULL,
  `locationid` INT NULL);
```

Action Output

		Time	Action	Response	Duration / Fetch Time
✗	6	21:49:09	Apply cha...		
✗	7	21:49:51	Apply cha...		
✗	8	21:50:12	Apply cha...		
⚠	9	21:50:35	CREATE T...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.012 sec
✗	10	21:51:13	CREATE T...	Error Code: 1050. Table 'users' already exists	0.0047 sec
✗	11	21:52:57	CREATE T...	Error Code: 1050. Table 'users' already exists	0.00049 sec
✗	12	21:53:40	CREATE T...	Error Code: 1075. Incorrect table definition; there can be only one auto column and it must be defined as a key	0.00077 sec
✗	13	21:53:59	CREATE T...	Error Code: 1075. Incorrect table definition; there can be only one auto column and it must be defined as a key	0.00073 sec
✓	14	21:56:23	CREATE T...	0 row(s) affected	0.0066 sec
✓	15	21:57:17	CREATE T...	0 row(s) affected	0.0061 sec

Prompt 3:
ALTER TABLE users MODIFY name VARCHAR(50) NOT NULL;
ALTER TABLE users MODIFY username VARCHAR(50) NOT NULL;

```
ALTER TABLE users MODIFY password VARCHAR(50) NOT NULL;
ALTER TABLE locations MODIFY type INT NOT NULL;
ALTER TABLE locations MODIFY description varchar(100) NOT NULL;
ALTER TABLE locations MODIFY lng real NOT NULL;
ALTER TABLE locations MODIFY lat real NOT NULL;
ALTER TABLE photograph MODIFY photoid int NOT NULL;
ALTER TABLE photograph MODIFY locationid int NOT NULL;
```



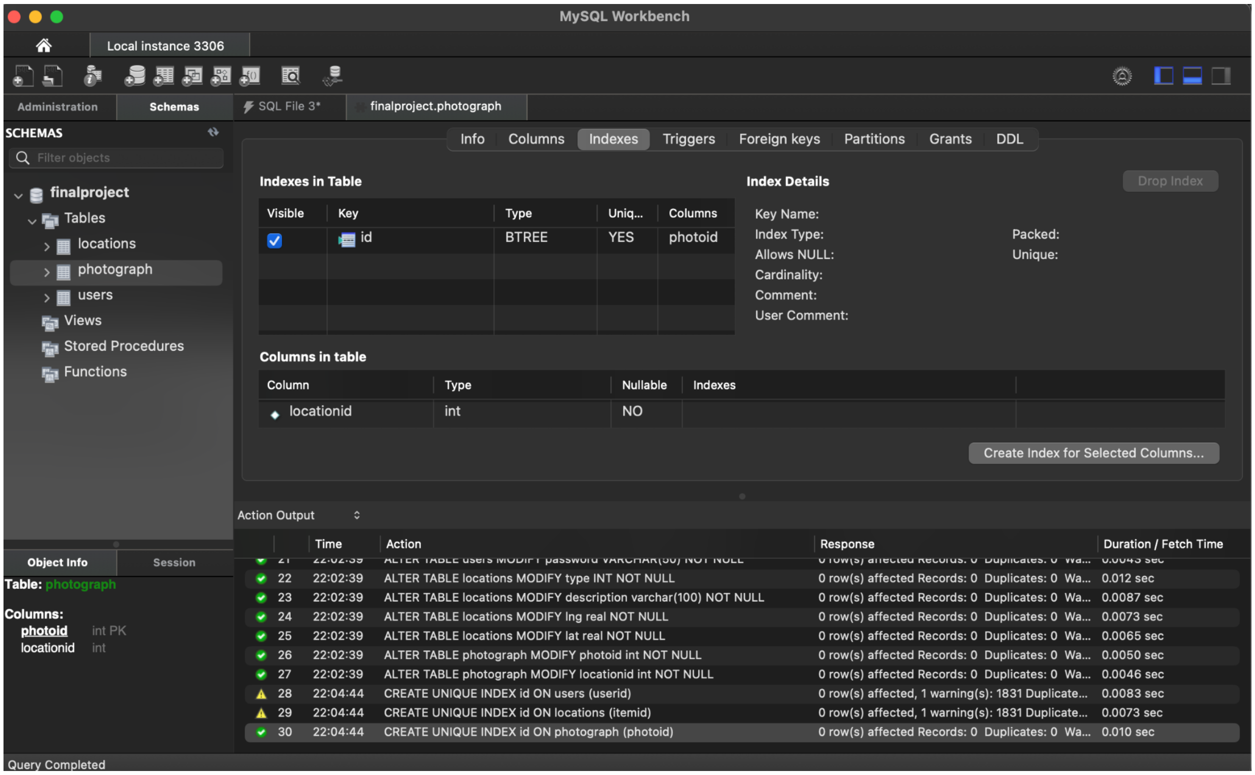
Schema: finalproject

```
ALTER TABLE users MODIFY name VARCHAR(50) NOT NULL;
ALTER TABLE users MODIFY username VARCHAR(50) NOT NULL;
ALTER TABLE users MODIFY password VARCHAR(50) NOT NULL;
ALTER TABLE locations MODIFY type INT NOT NULL;
ALTER TABLE locations MODIFY description varchar(100) NOT NULL;
ALTER TABLE locations MODIFY lng real NOT NULL;
ALTER TABLE locations MODIFY lat real NOT NULL;
ALTER TABLE photograph MODIFY photoid int NOT NULL;
ALTER TABLE photograph MODIFY locationid int NOT NULL;
```

		Time	Action	Response	Duration / Fetch Time
✓	18	21:59:46	ALTER TABLE users MODIFY password VARCHAR(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0098 sec
✓	19	22:02:39	ALTER TABLE users MODIFY name VARCHAR(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0077 sec
✓	20	22:02:39	ALTER TABLE users MODIFY username VARCHAR(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0039 sec
✓	21	22:02:39	ALTER TABLE users MODIFY password VARCHAR(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0043 sec
✓	22	22:02:39	ALTER TABLE locations MODIFY type INT NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.012 sec
✓	23	22:02:39	ALTER TABLE locations MODIFY description varchar(100) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0087 sec
✓	24	22:02:39	ALTER TABLE locations MODIFY lng real NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0073 sec
✓	25	22:02:39	ALTER TABLE locations MODIFY lat real NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0065 sec
✓	26	22:02:39	ALTER TABLE photograph MODIFY photoid int NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0050 sec
✓	27	22:02:39	ALTER TABLE photograph MODIFY locationid int NOT NULL	0 row(s) affected Records: 0 Duplicates: 0...	0.0046 sec

Prompt 4:

```
CREATE UNIQUE INDEX id ON users (userid);
CREATE UNIQUE INDEX id ON locations (itemid);
CREATE UNIQUE INDEX id ON photograph (photoid);
```



Indexes in Table

Visible	Key	Type	Uniq...	Columns
☑	id	BTREE	YES	photoid

Index Details

Key Name:
Index Type:
Allows NULL:
Cardinality:
Comment:
User Comment:
Packed:
Unique:

Columns in table

Column	Type	Nullable	Indexes
locationid	int	NO	

Table: photograph

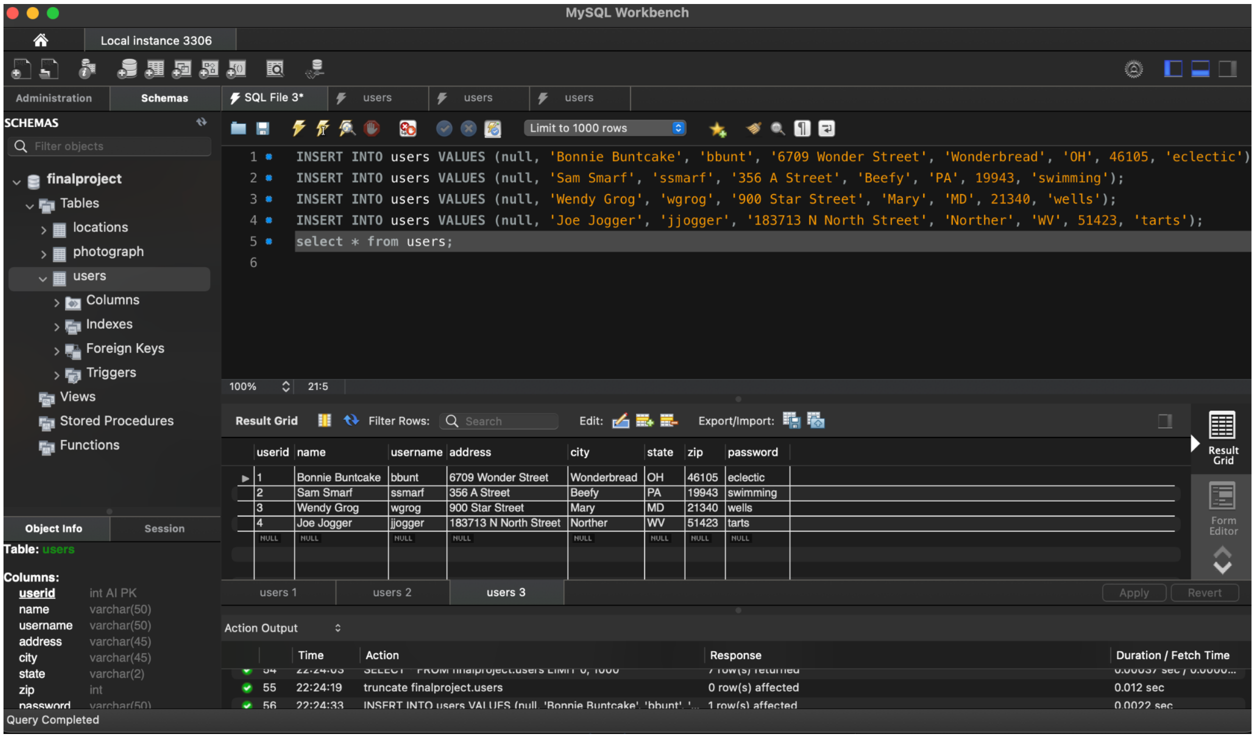
Columns:
photoid int PK
locationid int

Action Output

		Time	Action	Response	Duration / Fetch Time
✓	21	22:02:39	ALTER TABLE users MODIFY password VARCHAR(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Wa...	0.0043 sec
✓	22	22:02:39	ALTER TABLE locations MODIFY type INT NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Wa...	0.012 sec
✓	23	22:02:39	ALTER TABLE locations MODIFY description varchar(100) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Wa...	0.0087 sec
✓	24	22:02:39	ALTER TABLE locations MODIFY lng real NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Wa...	0.0073 sec
✓	25	22:02:39	ALTER TABLE locations MODIFY lat real NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Wa...	0.0065 sec
✓	26	22:02:39	ALTER TABLE photograph MODIFY photoid int NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Wa...	0.0050 sec
✓	27	22:02:39	ALTER TABLE photograph MODIFY locationid int NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Wa...	0.0046 sec
⚠	28	22:04:44	CREATE UNIQUE INDEX id ON users (userid)	0 row(s) affected, 1 warning(s): 1831 Duplicate...	0.0083 sec
⚠	29	22:04:44	CREATE UNIQUE INDEX id ON locations (itemid)	0 row(s) affected, 1 warning(s): 1831 Duplicate...	0.0073 sec
✓	30	22:04:44	CREATE UNIQUE INDEX id ON photograph (photoid)	0 row(s) affected Records: 0 Duplicates: 0 Wa...	0.010 sec

Prompt 5:

INSERT INTO users VALUES (null, 'Bonnie Buntcake', 'bbunt', '6709 Wonder Street', 'Wonderbread', 'OH', 46105, 'eclectic');
INSERT INTO users VALUES (null, 'Sam Smarf', 'ssmarf', '356 A Street', 'Beefy', 'PA', 19943, 'swimming');
INSERT INTO users VALUES (null, 'Wendy Grog', 'wgrog', '900 Star Street', 'Mary', 'MD', 21340, 'wells');
INSERT INTO users VALUES (null, 'Joe Jogger', 'jjogger', '183713 N North Street', 'Norther', 'WV', 51423, 'tarts');
select * from users;



```
INSERT INTO users VALUES (null, 'Bonnie Buntcake', 'bbunt', '6709 Wonder Street', 'Wonderbread', 'OH', 46105, 'eclectic')
INSERT INTO users VALUES (null, 'Sam Smarf', 'ssmarf', '356 A Street', 'Beefy', 'PA', 19943, 'swimming');
INSERT INTO users VALUES (null, 'Wendy Grog', 'wgrog', '900 Star Street', 'Mary', 'MD', 21340, 'wells');
INSERT INTO users VALUES (null, 'Joe Jogger', 'jjogger', '183713 N North Street', 'Norther', 'WV', 51423, 'tarts');
select * from users;
```

userid	name	username	address	city	state	zip	password
1	Bonnie Buntcake	bbunt	6709 Wonder Street	Wonderbread	OH	46105	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

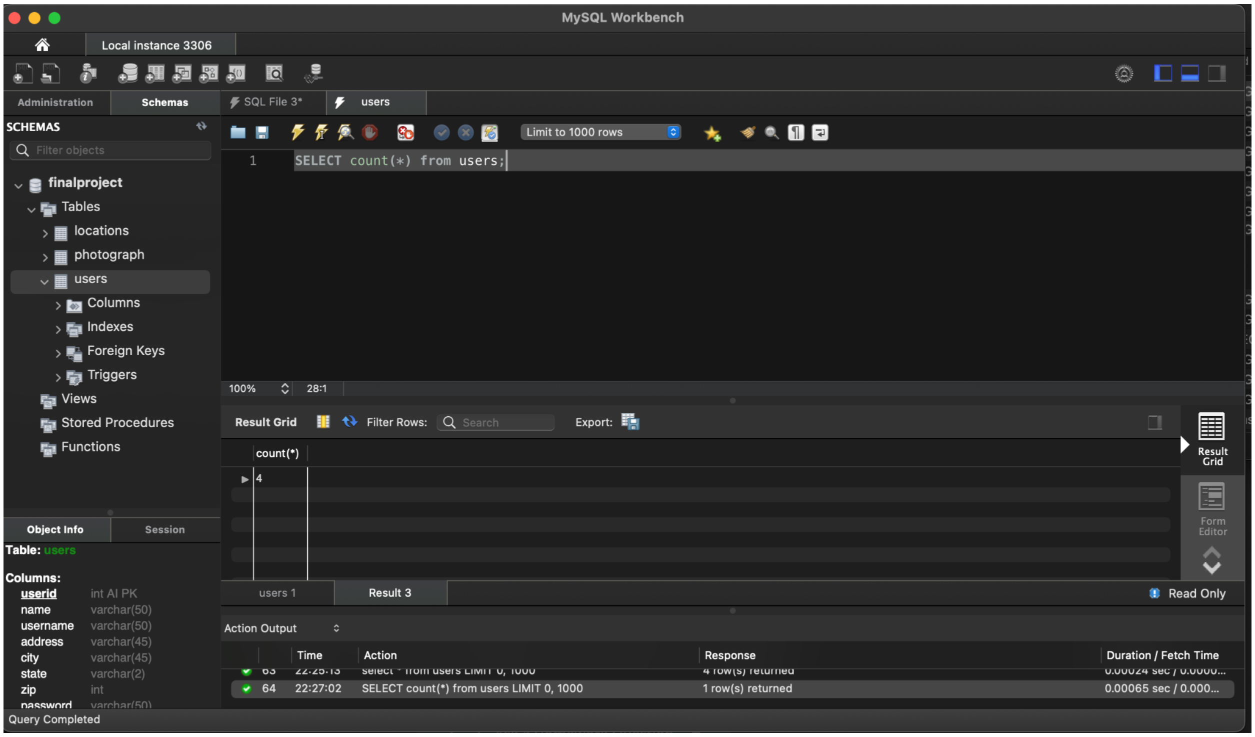
```
     Time      Action                                                    Response            Duration / Fetch Time
 54  22:24:03  SELECT * FROM finalproject.users LIMIT 0, 1000            7 row(s) returned   0.00037 sec / 0.0000...
 55  22:24:19  truncate finalproject.users                               0 row(s) affected   0.012 sec
 56  22:24:33  INSERT INTO users VALUES (null, 'Bonnie Buntcake', 'bbunt', '...  1 row(s) affected   0.0022 sec
```

Table: users

Columns:

userid	int AI PK
name	varchar(50)
username	varchar(50)
address	varchar(45)
city	varchar(45)
state	varchar(2)
zip	int
password	varchar(50)

Prompt 6:



```
SELECT count(*) from users;
```

count(*)
4

```
     Time      Action                                     Response            Duration / Fetch Time
 63  22:25:13  select * from users LIMIT 0, 1000          4 row(s) returned   0.00024 sec / 0.0000...
 64  22:27:02  SELECT count(*) from users LIMIT 0, 1000   1 row(s) returned   0.00065 sec / 0.000...
```

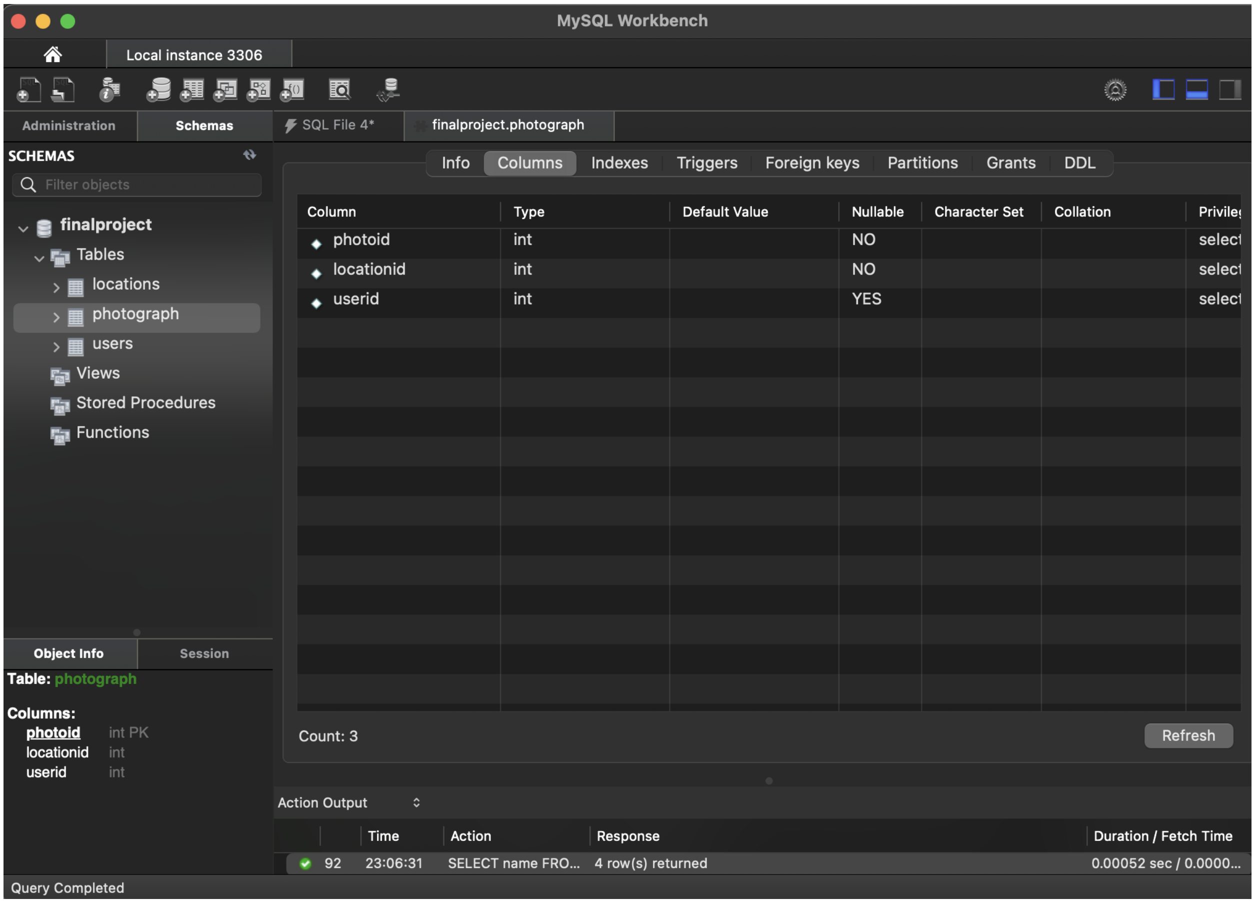
Table: users

Columns:

userid	int AI PK
name	varchar(50)
username	varchar(50)
address	varchar(45)
city	varchar(45)
state	varchar(2)
zip	int
password	varchar(50)

Prompt 7:

ALTER TABLE photograph ADD COLUMN user.id int AFTER locationid;



Column	Type	Default Value	Nullable	Character Set	Collation	Privile
photoid	int		NO			select
locationid	int		NO			select
userid	int		YES			select

Count: 3

Table: photograph

Columns:

photoid	int PK
locationid	int
userid	int

```
     Time      Action              Response            Duration / Fetch Time
92   23:06:31  SELECT name FRO...  4 row(s) returned   0.00052 sec / 0.0000...
```

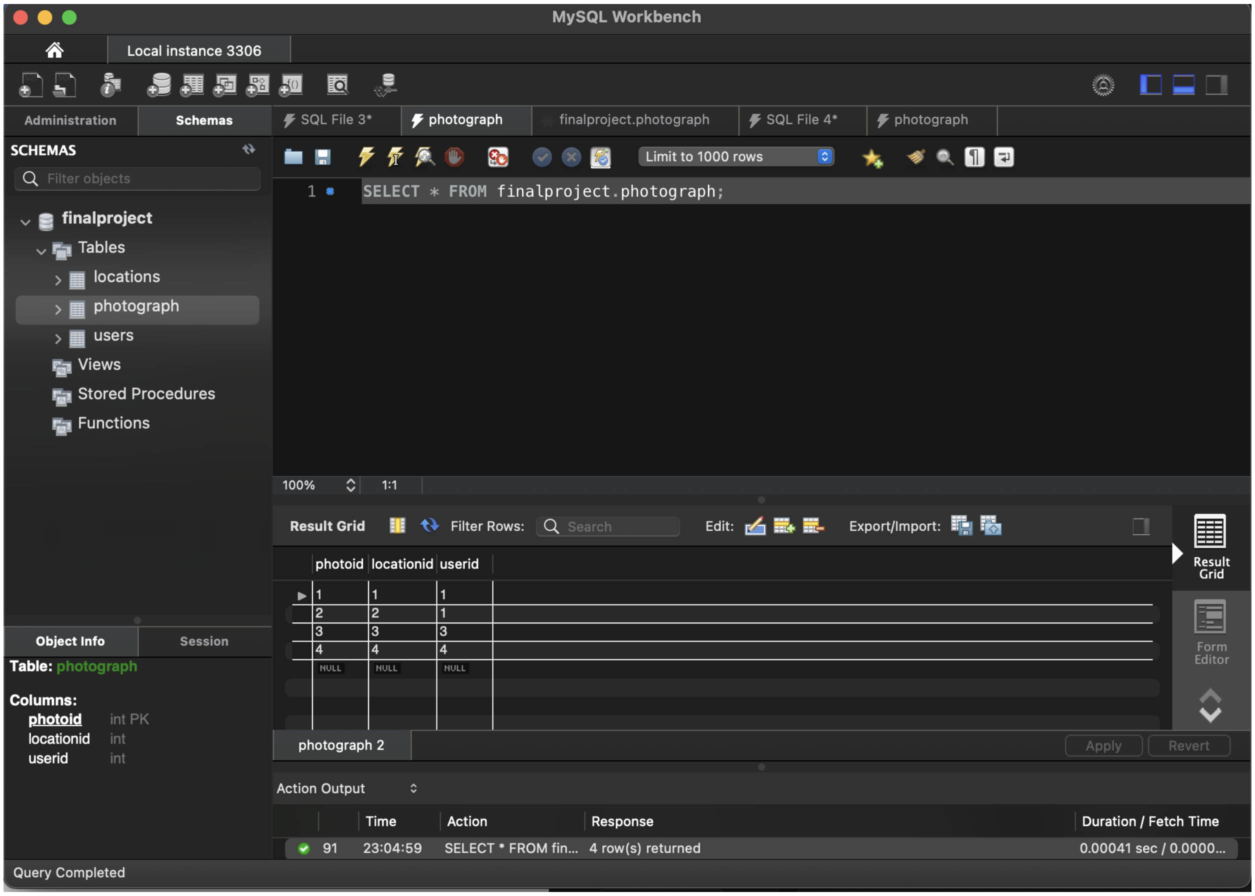
Query Completed

Prompt 8:

In order to ensure the accuracy and consistency of our data we should alter the user.id column to be not null. This is because in our database this column allows us to identify which users have taken photos, and what user the photos belong to. This column acts as a foreign key/primary key relationship between the photograph and users tables. By using a not null constraint here will make sure that we are following the business rules determined by the database administrator and app developer. This is especially important when dealing with photographs because they are intellectual property of the photographer, and thus they need to be readily related in the database in order to ensure compliance with copyright law.

Prompt 9:

```
INSERT INTO locations VALUES (null,1, 'Independence Hall', 794.35, 651.43);
INSERT INTO locations VALUES (null, 2, '6709 Wonder Street', 323.41, 412.22);
INSERT INTO locations VALUES (null, 1, 'sunrise', 221.45, 132.43);
INSERT INTO locations VALUES (null,2, '356 A Street', 123.32, 222.43);
INSERT INTO locations VALUES (null,1, 'Mountains', 34.12, 87.99);
INSERT INTO locations VALUES (null,2, '900 Star Street', 1071.9, 206.45);
INSERT INTO locations VALUES (null,1, 'Moonrise', 816.2, 111.2);
INSERT INTO locations VALUES (null,2, '183714 N North Street', 176.11, 11.176);
INSERT INTO photograph VALUES (1, 1, 1);
INSERT INTO photograph VALUES (2, 2, 1);
INSERT INTO photograph VALUES (3, 3, 3);
INSERT INTO photograph VALUES (4, 4, 4);
```



```
SELECT * FROM finalproject.photograph;
```

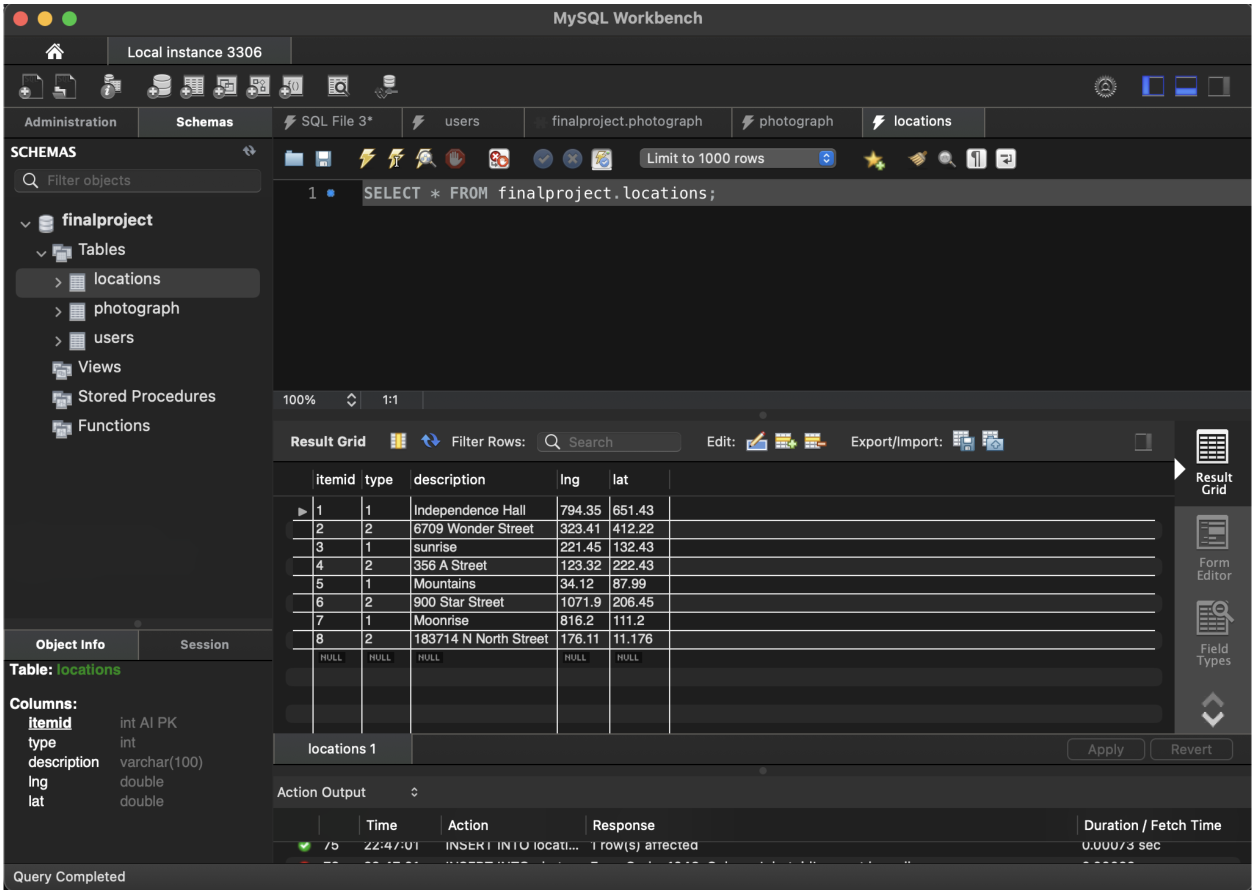
photoid	locationid	userid
1	1	1
2	2	1
3	3	3
4	4	4
NULL	NULL	NULL

Table: photograph

Columns:

photoid	int PK
locationid	int
userid	int

	Time	Action	Response	Duration / Fetch Time
91	23:04:59	SELECT * FROM fin...	4 row(s) returned	0.00041 sec / 0.0000...



```
SELECT * FROM finalproject.locations;
```

itemid	type	description	lng	lat
1	1	Independence Hall	794.35	651.43
2	2	6709 Wonder Street	323.41	412.22
3	1	sunrise	221.45	132.43
4	2	356 A Street	123.32	222.43
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	176.11	11.176
NULL	NULL	NULL	NULL	NULL

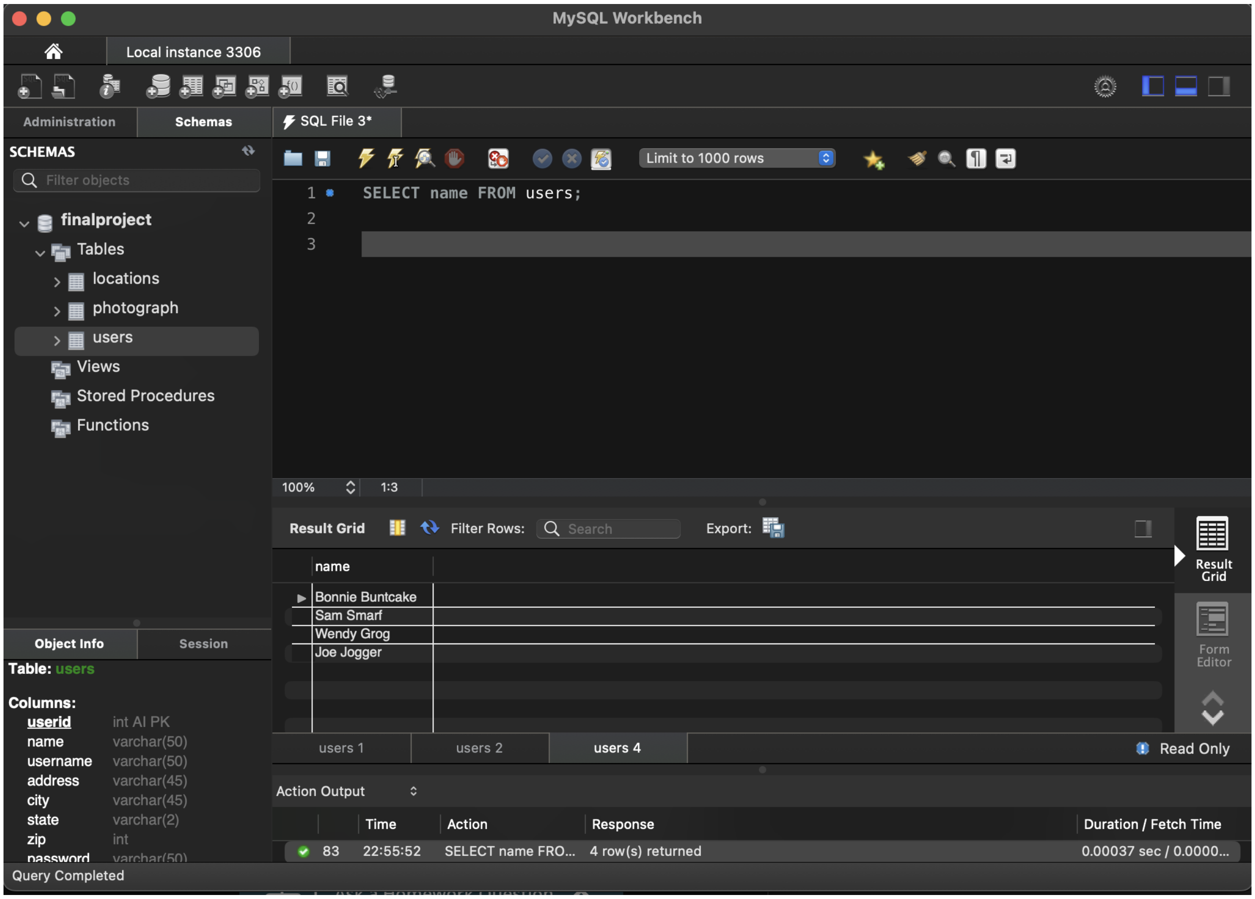
Table: locations

Columns:

itemid	int AI PK
type	int
description	varchar(100)
lng	double
lat	double

		Time	Action	Response	Duration / Fetch Time
✓	75	22:47:01	INSERT INTO locati...	1 row(s) affected	0.00073 sec

Prompt 10:



```
SELECT name FROM users;
```

```
name
Bonnie Buntcake
Sam Smarf
Wendy Grog
Joe Jogger
```

Table: users

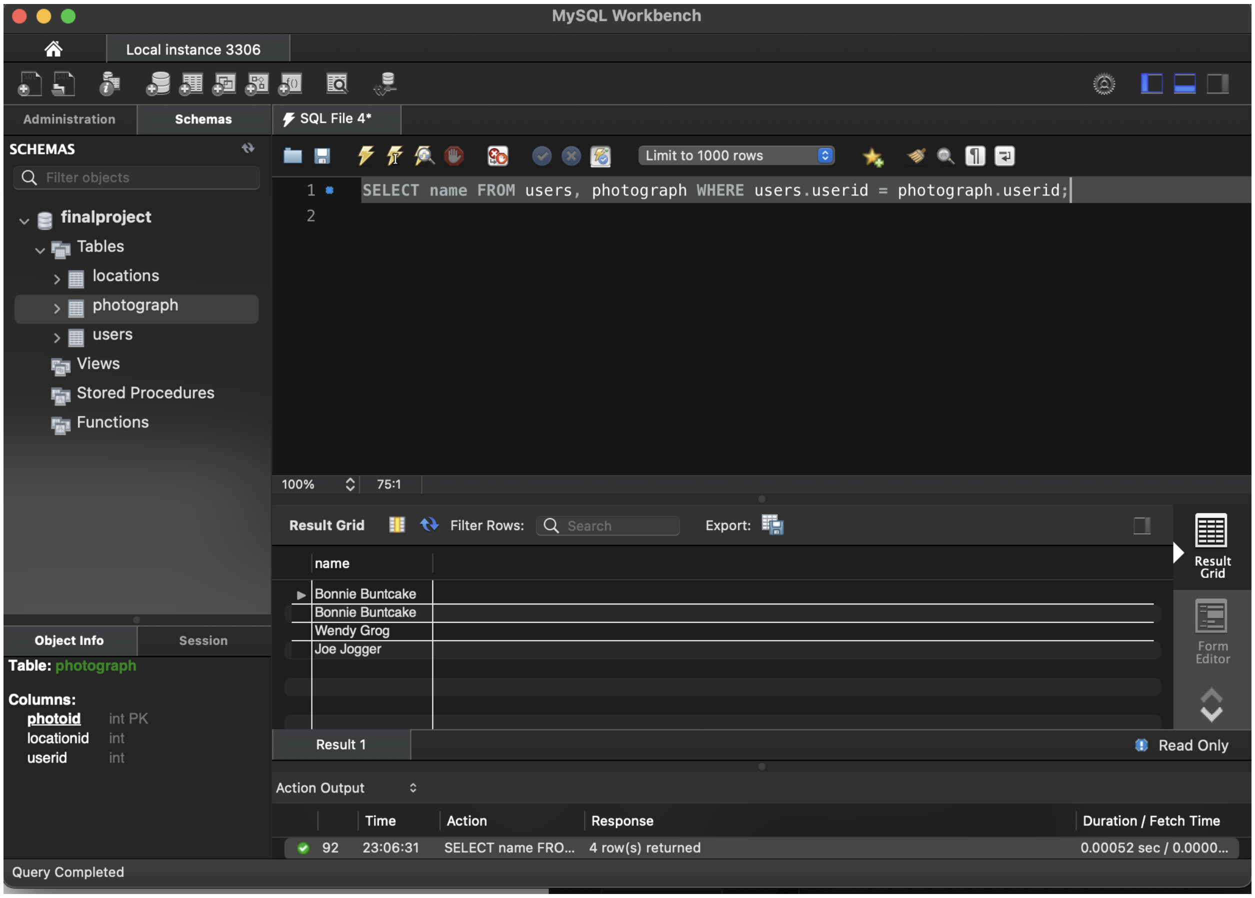
Columns:

userid	int AI PK
name	varchar(50)
username	varchar(50)
address	varchar(45)
city	varchar(45)
state	varchar(2)
zip	int
password	varchar(50)

```
    Time      Action               Response            Duration / Fetch Time
83  22:55:52  SELECT name FRO...   4 row(s) returned   0.00037 sec / 0.0000...
```

Prompt 11:

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



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

name
Bonnie Buntcake
Bonnie Buntcake
Wendy Grog
Joe Jogger

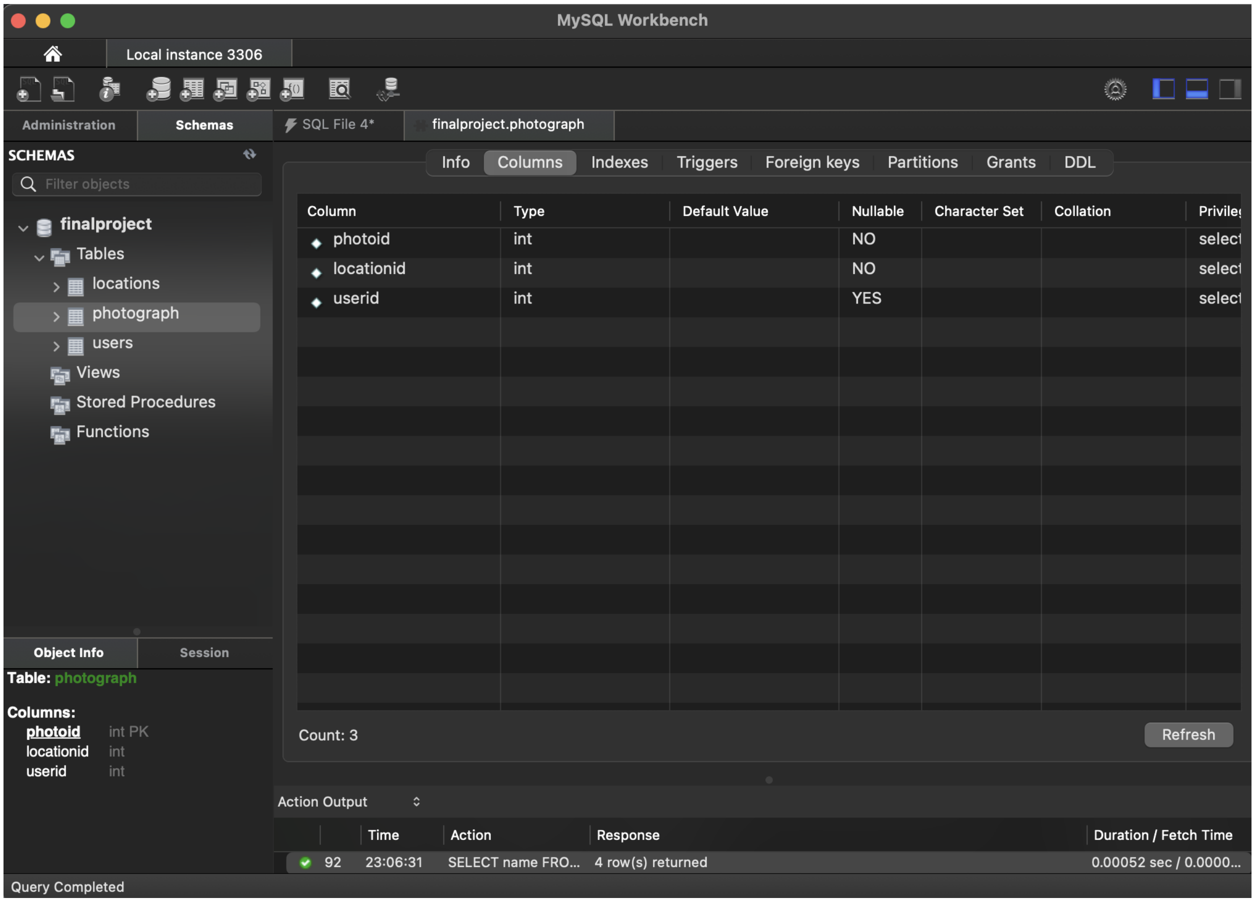
Table: photograph

Columns:
photoid int PK
locationid int
userid int

	Time	Action	Response	Duration / Fetch Time
92	23:06:31	SELECT name FRO...	4 row(s) returned	0.00052 sec / 0.0000...

Prompt 12:

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



Column	Type	Default Value	Nullable	Character Set	Collation	Privileg
◆ photoid	int		NO			select
◆ locationid	int		NO			select
◆ userid	int		YES			select

Count: 3

Table: photograph

Columns:

photoid	int PK
locationid	int
userid	int

		Time	Action	Response	Duration / Fetch Time
✓	92	23:06:31	SELECT name FRO...	4 row(s) returned	0.00052 sec / 0.0000...

Query Completed