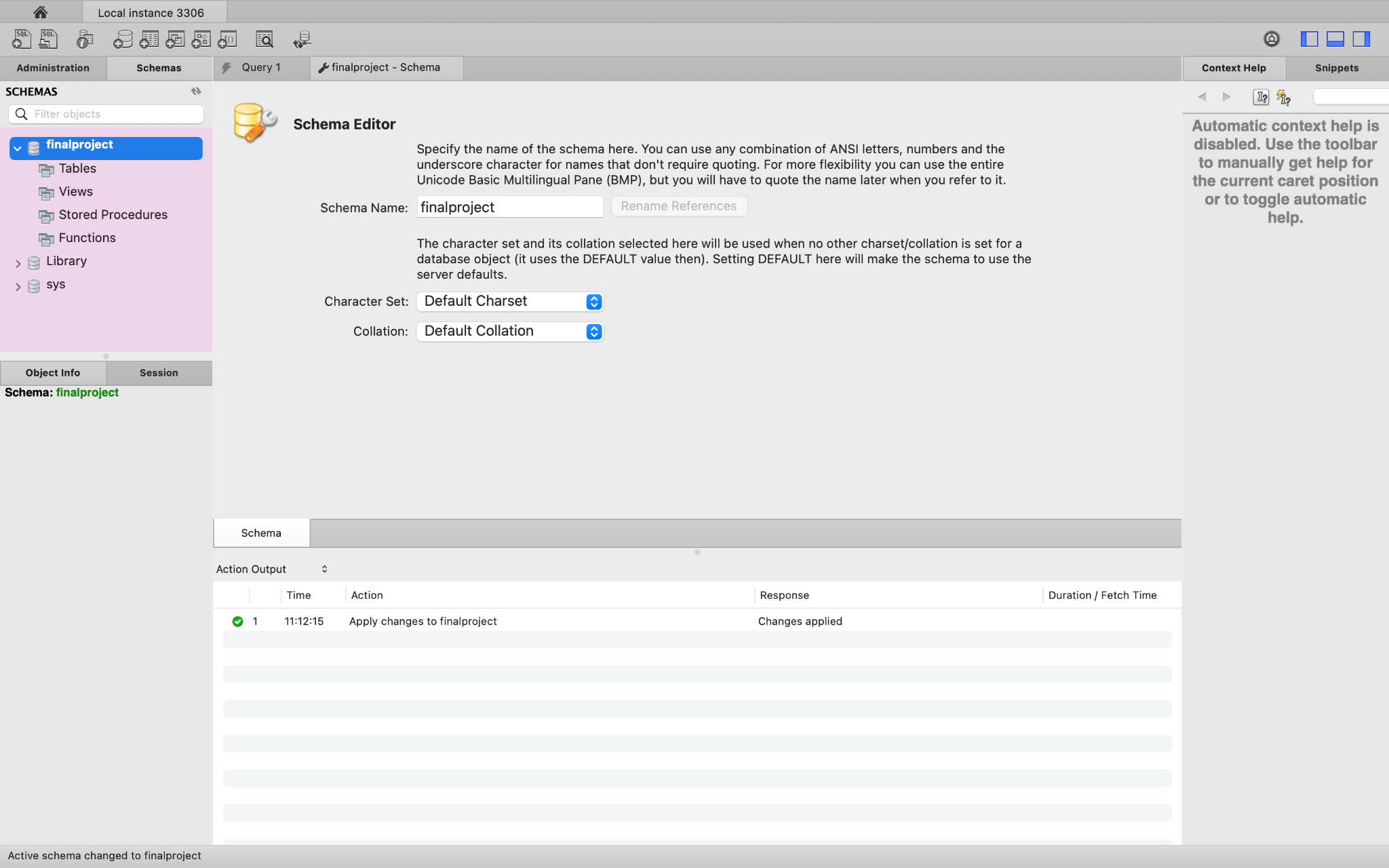
**Database Management - Assignment: Database System**

PROMPT 1 - Create the Schema/Database:



PROMPT 2 - Create Tables:

Users Table:

CREATE TABLE finalproject.users (

userid INT PRIMARY KEY NOT NULL AUTO\_INCREMENT UNIQUE,

name VARCHAR(45) NULL,

username VARCHAR(20) NULL,

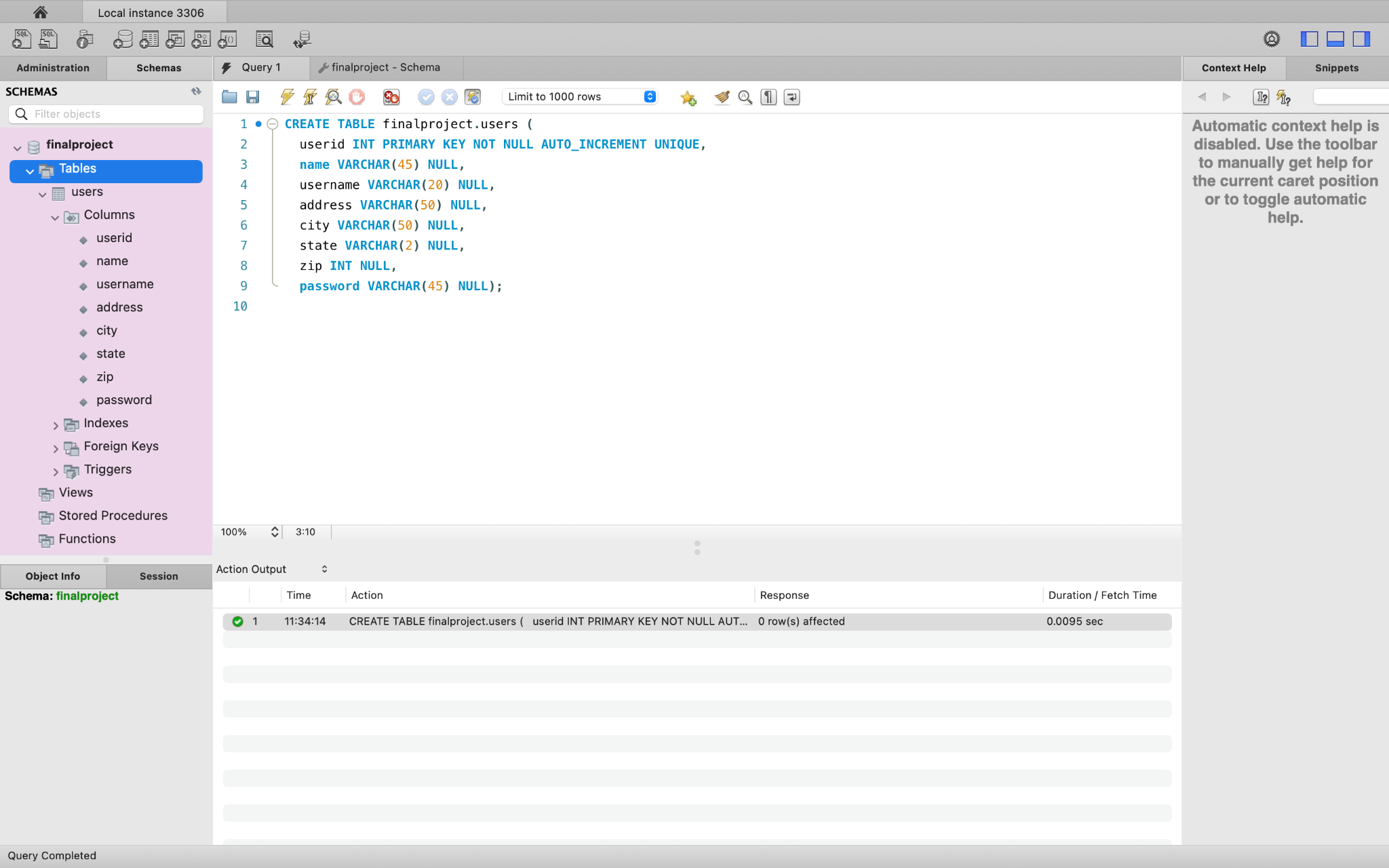
address VARCHAR(50) NULL,

city VARCHAR(50) NULL,

state VARCHAR(2) NULL,

zip INT NULL,

password VARCHAR(45) NULL);



Locations Table:

CREATE TABLE finalproject.locations (

itemid INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

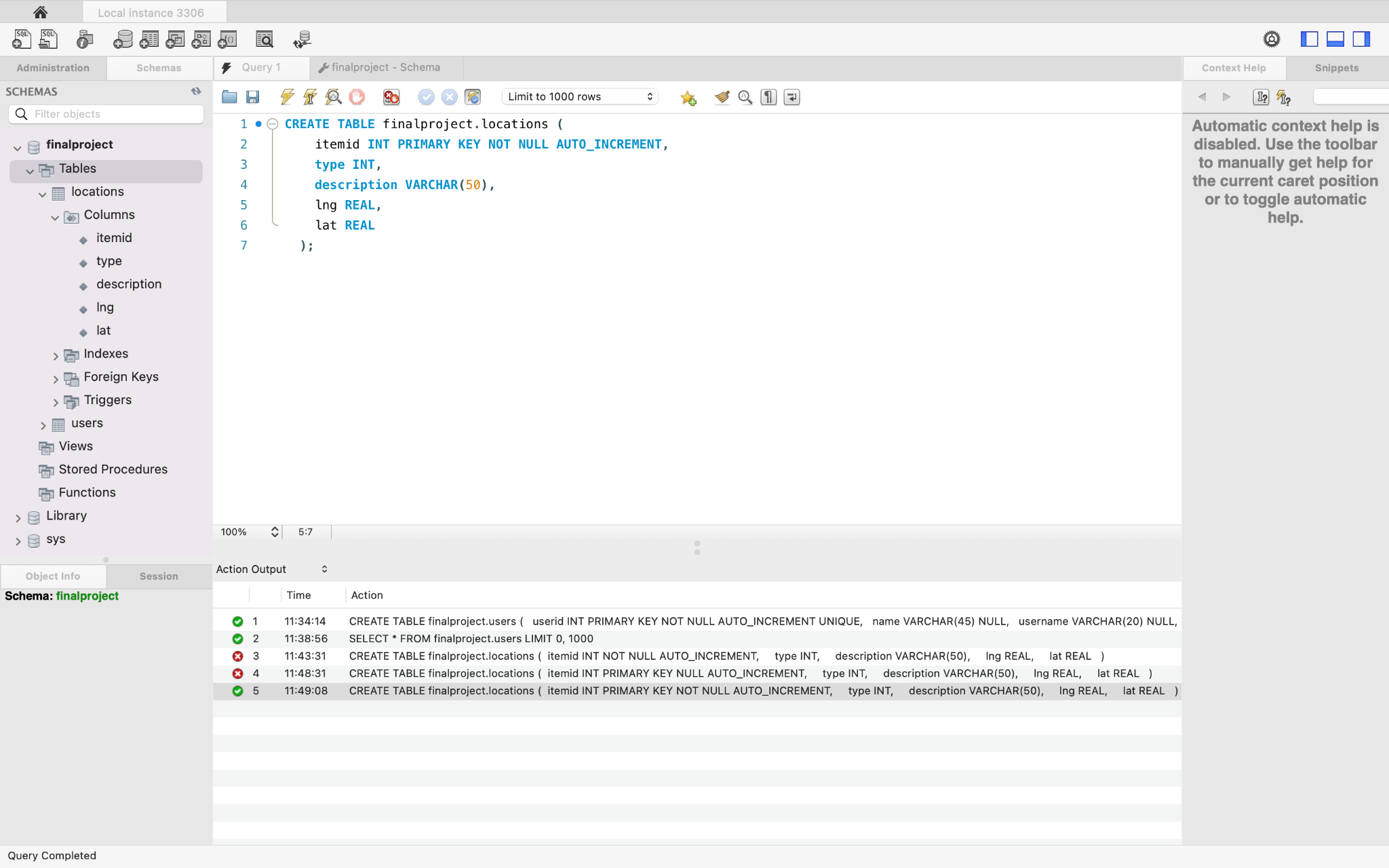
type INT,

description VARCHAR(50),

lng REAL,

lat REAL

);



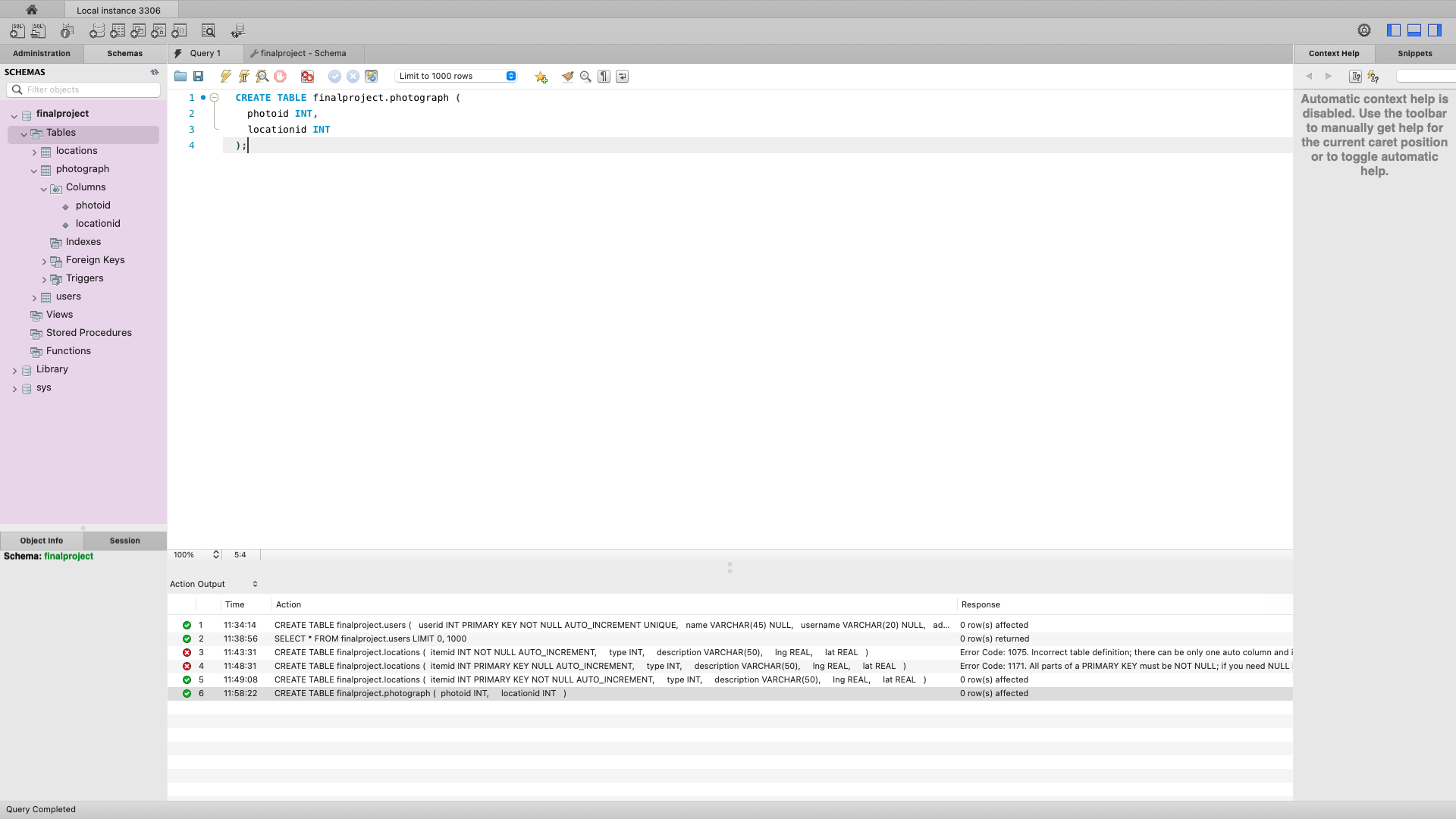
Photograph Table:

CREATE TABLE finalproject.photograph (

photoid INT,

locationid INT

);



PROMPT 3 - Alter Tables:

ALTER TABLE finalproject.locations

MODIFY COLUMN type INT NOT NULL,

MODIFY COLUMN description VARCHAR(50) NOT NULL,

MODIFY COLUMN lng REAL NOT NULL,

MODIFY COLUMN lat REAL NOT NULL;

ALTER TABLE finalproject.users

MODIFY COLUMN name VARCHAR(45) NOT NULL,

MODIFY COLUMN username VARCHAR(20) NOT NULL,

MODIFY COLUMN password VARCHAR(45) NOT NULL;

ALTER TABLE finalproject.photograph

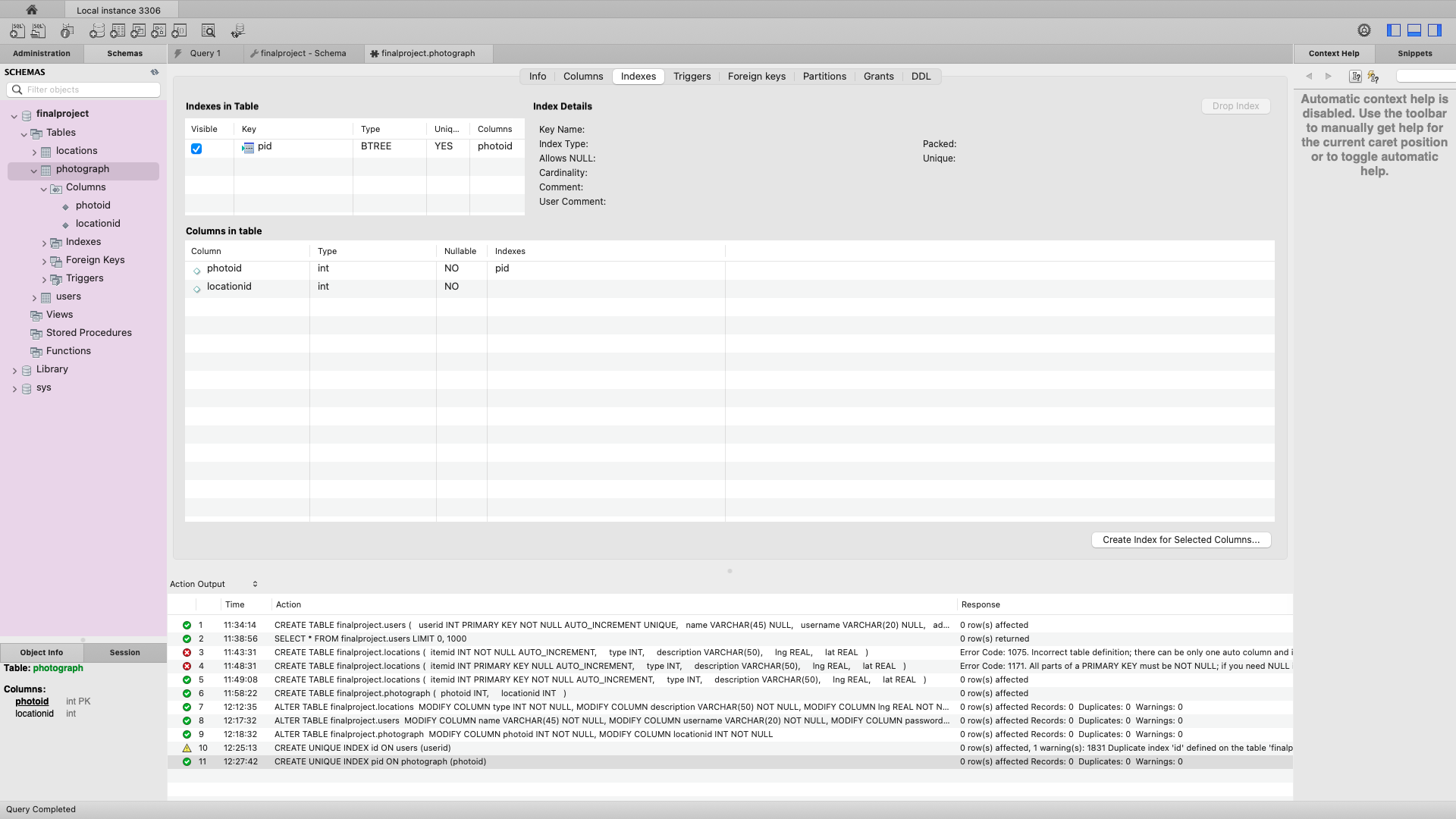
MODIFY COLUMN photoid INT NOT NULL,

MODIFY COLUMN locationid INT NOT NULL;

PROMPT 4 - Create Index:

CREATE UNIQUE INDEX id ON users (userid);

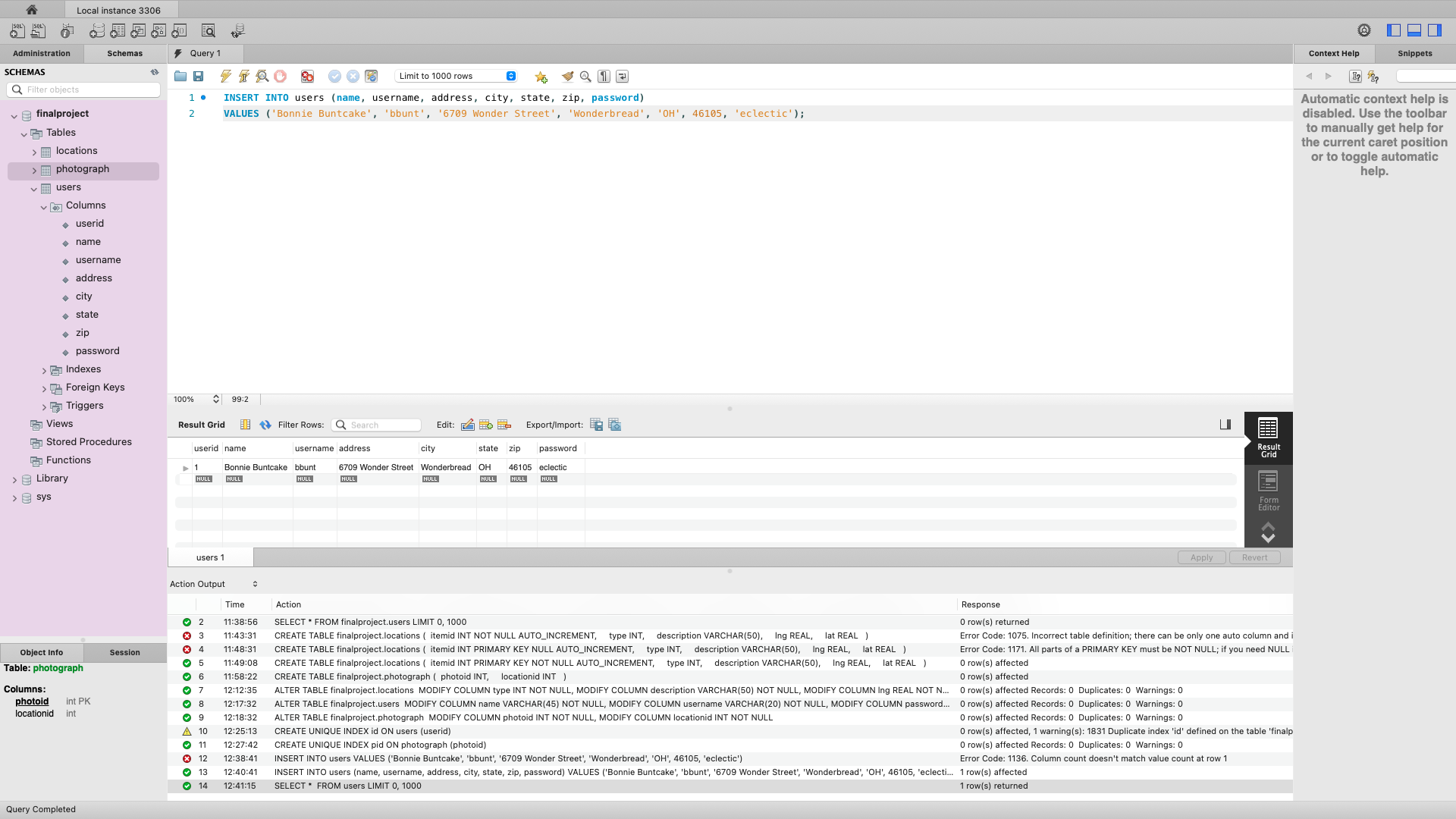
CREATE UNIQUE INDEX pid ON photograph (photoid);



PROMPT 5 - Enter Data:

INSERT INTO users (name, username, address, city, state, zip, password)

VALUES ('Bonnie Buntcake', 'bbunt', '6709 Wonder Street', 'Wonderbread', 'OH', 46105, 'eclectic');



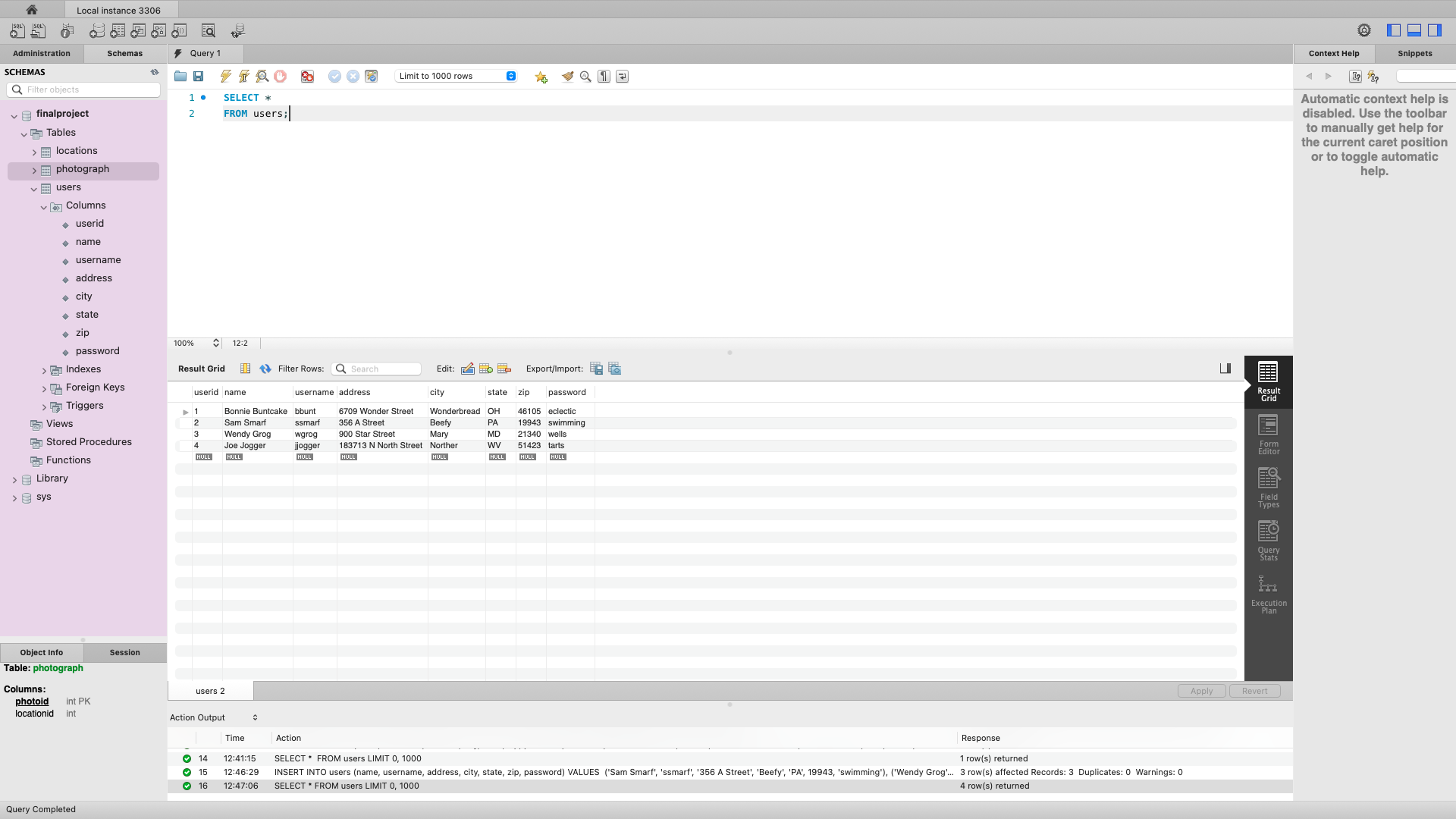
INSERT INTO users (name, username, address, city, state, zip, password)

VALUES

('Sam Smarf', 'ssmarf', '356 A Street', 'Beefy', 'PA', 19943, 'swimming'),

('Wendy Grog', 'wgrog', '900 Star Street', 'Mary', 'MD', 21340, 'wells'),

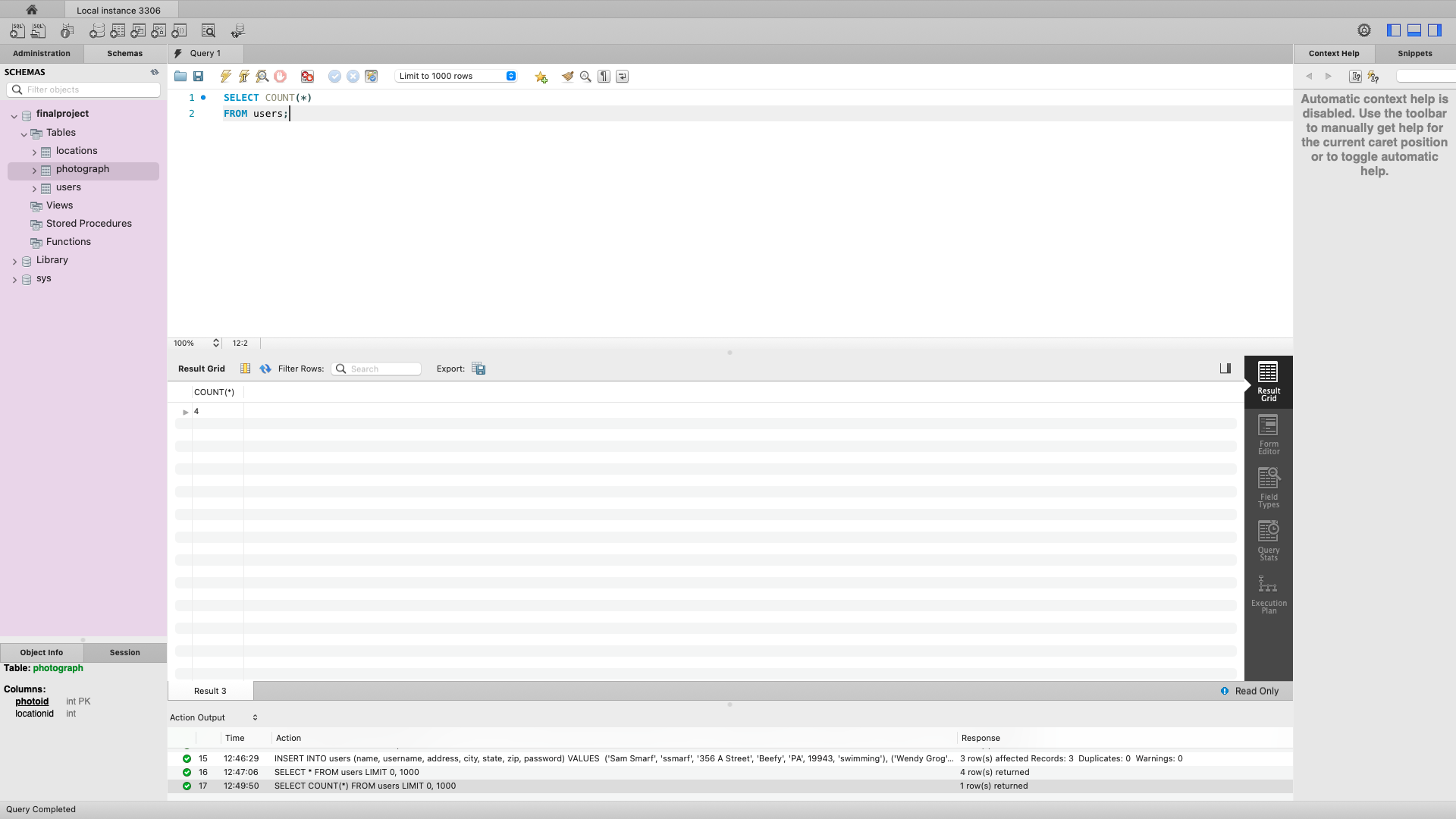
('Joe Jogger', 'jjogger', '183713 N North Street', 'Norther', 'WV', 51423, 'tarts');



PROMPT 6 - Count Rows:

SELECT COUNT(\*)

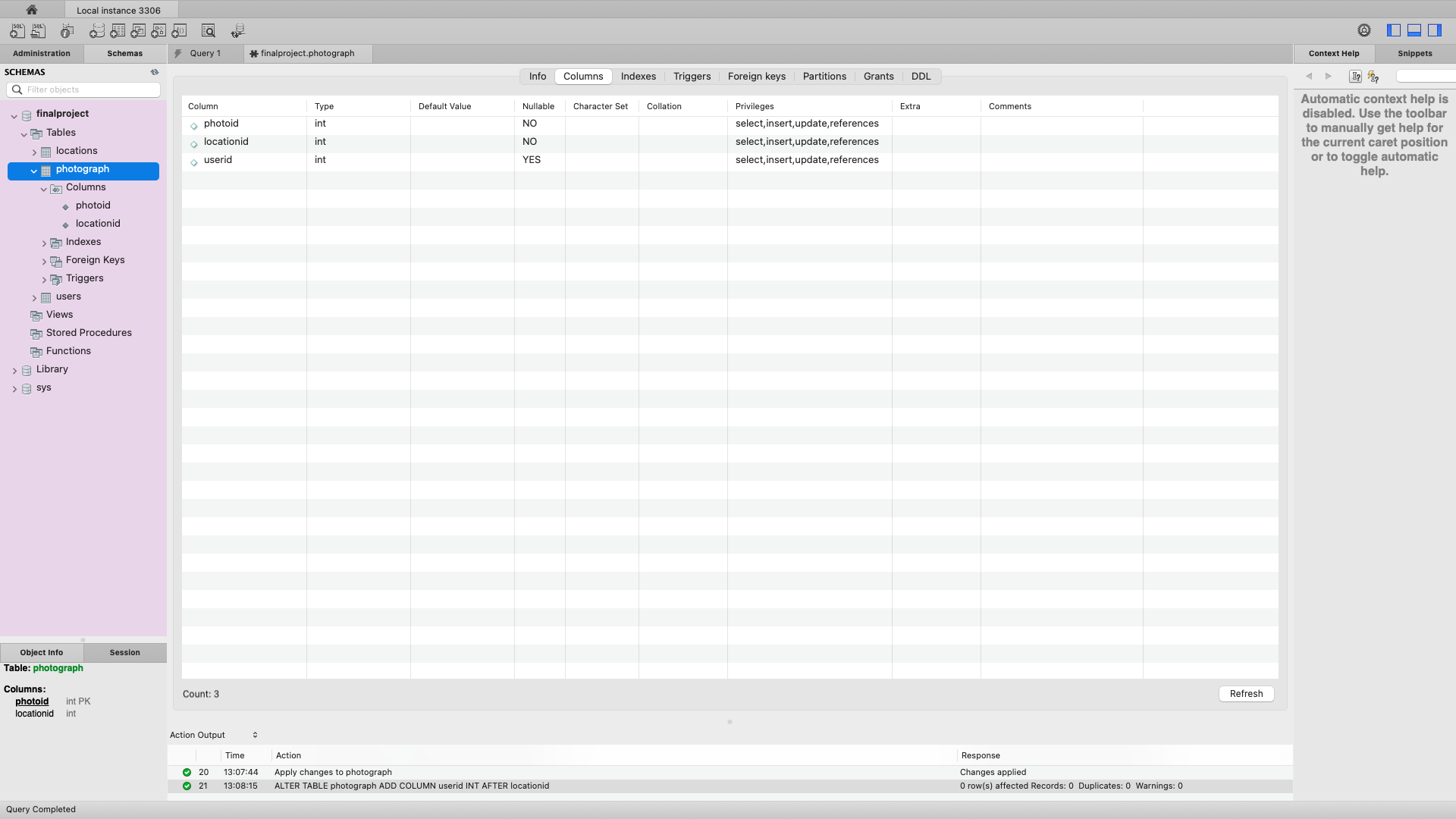
FROM users;



PROMPT 7 - Add Column:

ALTER TABLE photograph ADD COLUMN userid INT AFTER locationid;

(Table Inspector view of the columns tab for the photograph table showing the addition of the userid column)



PROMPT 8 - Issue with New Column:

ALTER TABLE photograph MODIFY userid INT NOT NULL;

ALTER TABLE photograph ADD CONSTRAINT userid\_fk FOREIGN KEY (userid) REFERENCES users (userid);

The new userid column in the photograph table is still incorrect because constraints have not been added to it to ensure data integrity. First, it needs to be NOT NULL. Second, the userid in the photograph table needs to be linked to the userid in the users table. In the users table, the userid is the PRIMARY KEY, so in the photograph table, we need to make userid a FOREIGN KEY to link to the tables together. By doing this, we ensure that the userid remains unique and data redundancy is reduced because we will not unintentionally have two different userid’s for the same person.

Prompt 9 - Location and Photograph Table Updates:

INSERT INTO locations (type, description, lng, lat)

VALUES

(1, 'Independence Hall', 794.35, 651.43),

(2, '6709 Wonder Street', 323.41, 412.22),

(1, 'Sunrise', 221.45, 132.43),

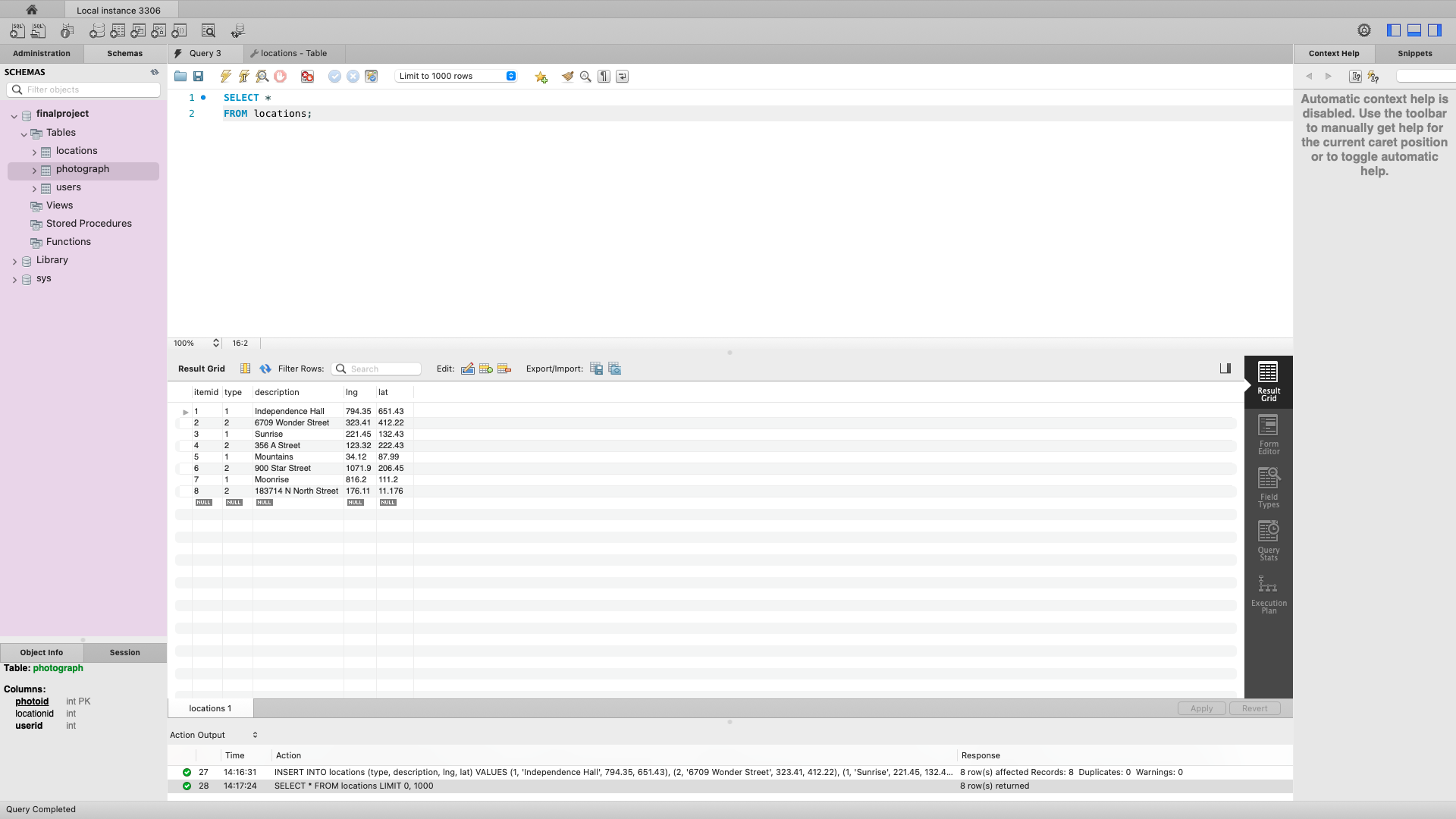
(2, '356 A Street', 123.32, 222.43),

(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', 176.11, 11.176);



INSERT INTO photograph (photoid, locationid, userid)

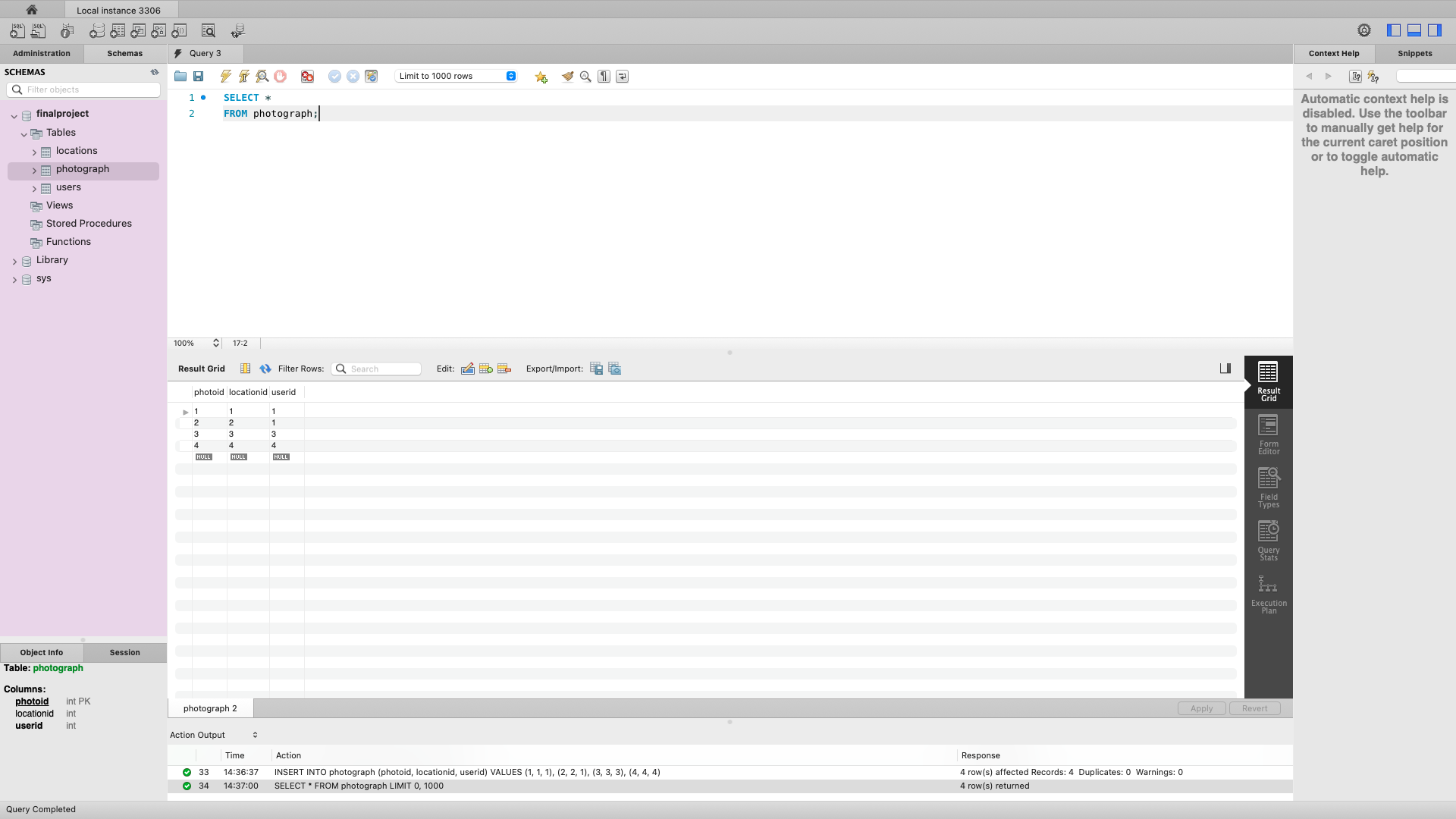
VALUES

(1, 1, 1),

(2, 2, 1),

(3, 3, 3),

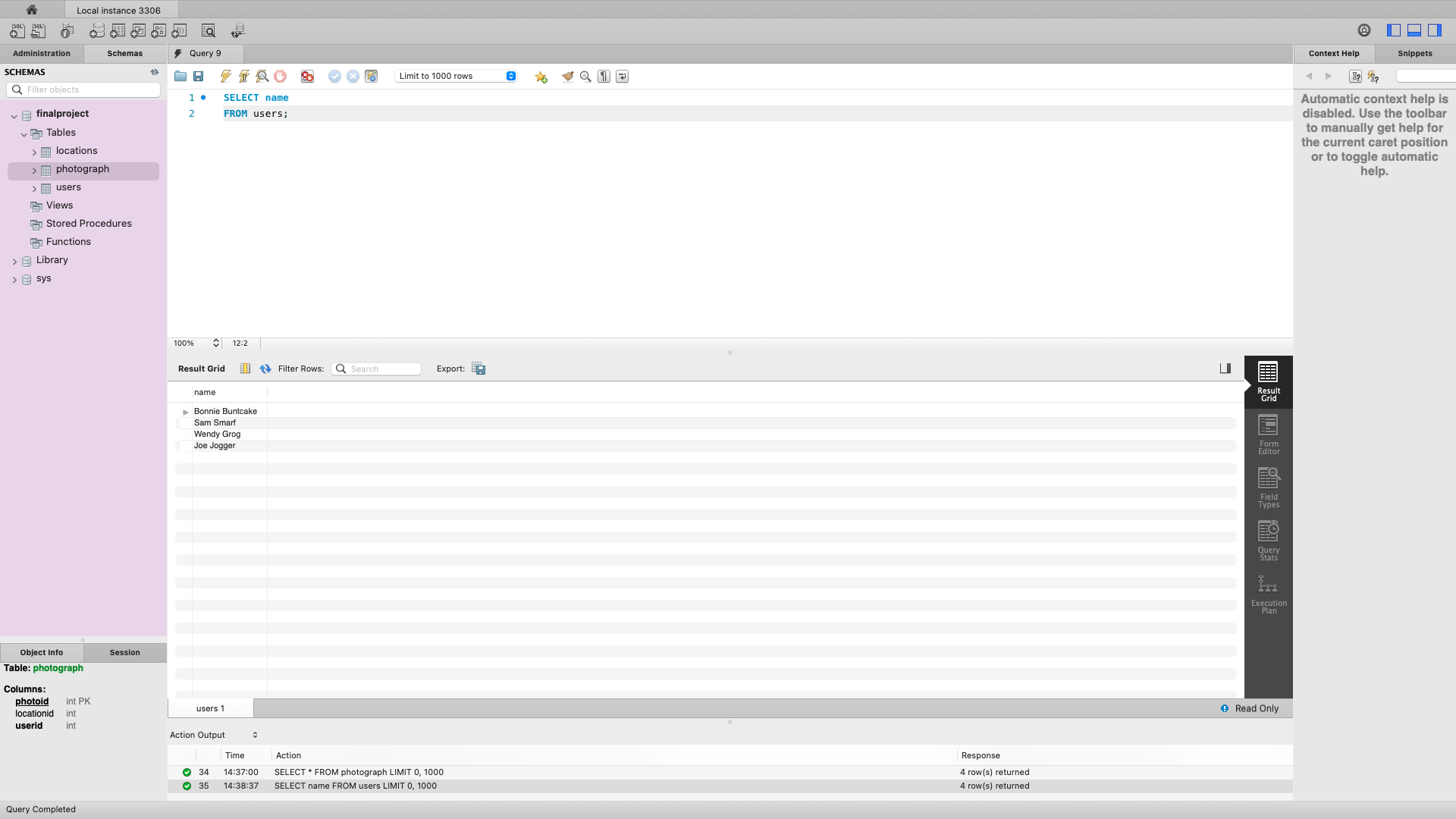
(4, 4, 4);



PROMPT 10 - Users:

SELECT name

FROM users;

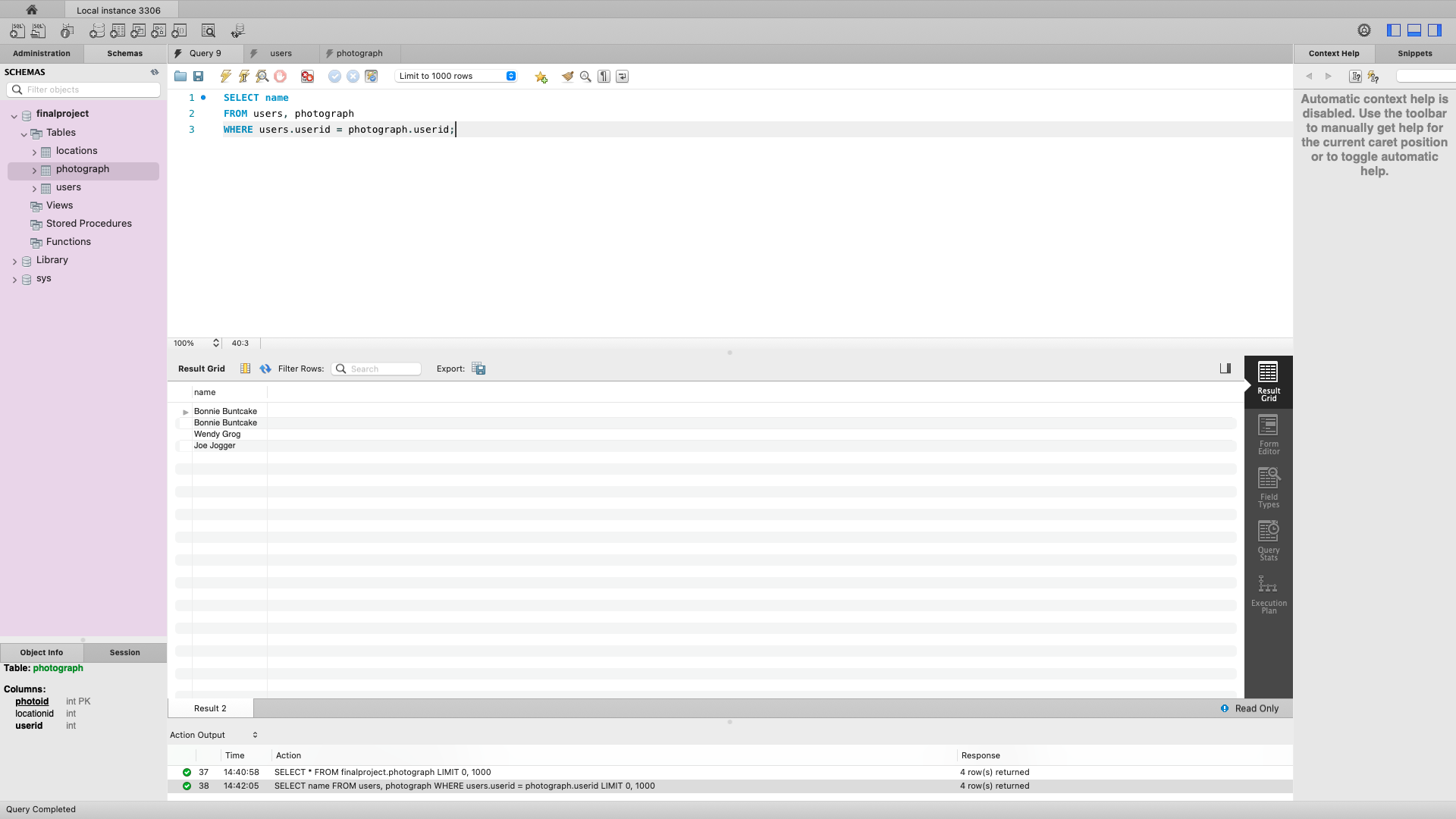


PROMPT 11 - Who’s Taking Pictures?

SELECT name

FROM users, photograph

WHERE users.userid = photograph.userid;



PROMPT 12 - Unique Names:

SELECT DISTINCT name

FROM users, photograph

WHERE users.userid = photograph.userid;

