

DLBDSPDM01: - Development Stage of the Development of a Data Mart in SQL

INTRODUCTION

SQL is one of the widely used medium for storing data in an organize and structured way, the steps of making these project are: -

- Installation of MySQL workbench on the PC
- Creating of the database in MySQL Workbench
- Creating Tables in the database
- Altering tables for preservation of referential integrity
- Inserting rows of data in tables
- Updating of tables for preservation of referential integrity.

INSTALLATION OF REQUIRED TOOLS

The first step of making of this project is installation of MySQL on a PC.

MySQL Installer 8.0.27

Select Operating System:

Microsoft Windows

Looking for previous GA
versions?

Windows (x86, 32-bit), MSI Installer

(mysql-installer-web-community-8.0.27.1.msi)

8.0.27

2.3M

Download

MD5: 44b7f3e4c1bdcc641621cf3aa31ea18f4 | Signature

Windows (x86, 32-bit), MSI Installer

(mysql-installer-community-8.0.27.1.msi)

8.0.27

470.2M

Download

MD5: 9b7af5c91139659b10b84b1ca357d08f | Signature



We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

CREATION OF THE DATABASE FOLLOWED BY CREATION OF COUNTRY AND AMENITIES AFTER USE TABLES IN THE DATABASE

```
CREATE SCHEMA `datamart`;  
USE datamart;
```

```
CREATE TABLE country (  
    id INT NOT NULL,  
    country VARCHAR(50),  
    continent VARCHAR(50),  
    last_updated DATETIME,  
    PRIMARY KEY(id)  
);
```

```
CREATE TABLE amenities_after_use (  
    id INT UNIQUE NOT NULL,  
    kitchen VARCHAR(50),  
    air_conditioning VARCHAR(50),  
    heating VARCHAR(50),  
    washer VARCHAR(50),  
    dryer VARCHAR(50),  
    internet VARCHAR(50),  
    PRIMARY KEY(id)
```

CREATION OF FACILITIES AFTER USE AND ENTERTAINMENT AFTER USE TABLES

```
26 • CREATE TABLE facilities_after_use (
27     id INT UNIQUE NOT NULL,
28     hot_tub VARCHAR(50),
29     ev_charger VARCHAR(50),
30     pool VARCHAR(50),
31     gym VARCHAR(50),
32     free_parking VARCHAR(50),
33     PRIMARY KEY(id)
34 );
35
36 • CREATE TABLE entertainment_after_use (
37     id INT UNIQUE NOT NULL,
38     tv VARCHAR(50),
39     sound_system VARCHAR(50),
40     projector VARCHAR(50),
41     water_front VARCHAR(50),
42     beach_front VARCHAR(50),
43     PRIMARY KEY(id)
44 );
45
```

CREATION OF SECURITY QUESTIONS AND POST INSPECTION TABLES

```
46
47 • CREATE TABLE security_questions (
48     id INT UNIQUE NOT NULL,
49     nickname VARCHAR(50),
50     pet_name VARCHAR(50),
51     birth_city VARCHAR(50),
52     PRIMARY KEY(id)
53 );
54
55
56 • CREATE TABLE post_inspection (
57     id INT UNIQUE NOT NULL,
58     inspection_date DATETIME,
59     amenities_after_use_id INT UNIQUE NOT NULL,
60     facilities_after_use_id INT UNIQUE NOT NULL,
61     entertainment_after_use_id INT UNIQUE NOT NULL,
62     PRIMARY KEY(id),
63     FOREIGN KEY(amenities_after_use_id) REFERENCES amenities_after_use(id) ON DELETE CASCADE,
64     FOREIGN KEY(facilities_after_use_id) REFERENCES facilities_after_use(id) ON DELETE CASCADE,
65     FOREIGN KEY(entertainment_after_use_id) REFERENCES entertainment_after_use(id) ON DELETE CASCADE
66 );
67
```

CREATION OF LOGIN DETAILS AND LOGIN TABLES

```
68 • Ⓜ CREATE TABLE login_details (
69     id INT UNIQUE NOT NULL,
70     username VARCHAR(50),
71     pass_word VARCHAR(50),
72     last_updated VARCHAR(50),
73     security_questions_id INT UNIQUE NOT NULL,
74     PRIMARY KEY(id),
75     FOREIGN KEY(security_questions_id) REFERENCES security_questions(id) ON DELETE CASCADE
76 );
77
78 • Ⓜ CREATE TABLE login (
79     id INT UNIQUE NOT NULL,
80     email VARCHAR(500),
81     phone_number VARCHAR(50),
82     create_date DATETIME,
83     last_login DATETIME,
84     account_status VARCHAR(50),
85     verified_status VARCHAR(50),
86     login_details_id INT UNIQUE NOT NULL,
87     PRIMARY KEY(id),
88     FOREIGN KEY(login_details_id) REFERENCES login_details(id) ON DELETE CASCADE
89 );
90
```

Activate Winc

CREATION OF STAFF AND DEPENDENTS TABLES

- `CREATE TABLE staff (`
 `id INT UNIQUE NOT NULL,`
 `first_name VARCHAR(50),`
 `last_name VARCHAR(50),`
 `gender VARCHAR(50),`
 `age INT,`
 `title VARCHAR(50),`
 `post_inspection_id INT UNIQUE NOT NULL,`
 `PRIMARY KEY(id),`
 `FOREIGN KEY(post_inspection_id) REFERENCES post_inspection(id) ON DELETE CASCADE`
`);`

- `CREATE TABLE dependents (`
 `id INT UNIQUE NOT NULL,`
 `spouse VARCHAR(3),`
 `children INT,`
 `infants INT,`
 `last_update DATETIME,`
 `PRIMARY KEY(id)`
`);`

CREATION OF ADDRESS(CUSTOMER) AND RULES TABLES

- `CREATE TABLE address (`
 `id INT UNIQUE NOT NULL,`
 `house_no INT,`
 `street VARCHAR(50),`
 `postcode INT,`
 `state VARCHAR(500),`
 `last_update DATETIME,`
 `country_id INT NOT NULL,`
 `PRIMARY KEY(id),`
 `FOREIGN KEY(country_id) REFERENCES country(id)`
`);`

- `CREATE TABLE rules (`
 `id INT UNIQUE NOT NULL,`
 `pets VARCHAR(50),`
 `smoking VARCHAR(50),`
 `PRIMARY KEY(id)`
`);`

Activate Windows

CREATION OF REVIEWS AND ENTERTAINMENT TABLES

```
134
135 • CREATE TABLE reviews (
136     id INT UNIQUE NOT NULL,
137     stars INT,
138     comments VARCHAR(500),
139     recommend VARCHAR(3),
140     PRIMARY KEY(id)
141 );
142
143 • CREATE TABLE entertainment (
144     id INT UNIQUE NOT NULL,
145     tv VARCHAR(3),
146     sound_system VARCHAR(3),
147     projector VARCHAR(3),
148     water_front VARCHAR(3),
149     beach_front VARCHAR(3),
150     PRIMARY KEY(id)
151 );
152
```

CREATION OF FACILITIES AND AMENITIES TABLES

- `CREATE TABLE facilities (`
 `id INT UNIQUE NOT NULL,`
 `hot_tub VARCHAR(3),`
 `ev_charger VARCHAR(3),`
 `pool VARCHAR(3),`
 `gym VARCHAR(3),`
 `free_parking VARCHAR(3),`
 `entertainment_id INT UNIQUE NOT NULL,`
 `PRIMARY KEY(id),`
 `FOREIGN KEY(entertainment_id) REFERENCES entertainment(id) ON DELETE CASCADE`
`);`

- `CREATE TABLE amenities (`
 `id INT UNIQUE NOT NULL,`
 `kitchen VARCHAR(3),`
 `air_conditioning VARCHAR(3),`
 `heating VARCHAR(3),`
 `washer VARCHAR(3),`
 `dryer VARCHAR(3),`
 `internet VARCHAR(3),`
 `facilities_id INT UNIQUE NOT NULL,`
 `PRIMARY KEY(id),`
 `FOREIGN KEY(facilities_id) REFERENCES facilities(id) ON DELETE CASCADE`
`);`

CREATION OF ROOM OWNER ADDRESS AND ROOM OWNER TABLES

- `CREATE TABLE room_owner_address (`
 `id INT UNIQUE NOT NULL,`
 `house_no INT,`
 `street VARCHAR(50),`
 `postcode INT,`
 `state VARCHAR(50),`
 `last_update DATETIME,`
 `country_id INT NOT NULL,`
 `PRIMARY KEY(id),`
 `FOREIGN KEY(country_id) REFERENCES country(id)`
`);`

- `CREATE TABLE room_owner (`
 `id INT UNIQUE NOT NULL,`
 `first_name VARCHAR(50),`
 `last_name VARCHAR(50),`
 `gender VARCHAR(50),`
 `age INT,`
 `room_owner_address_id INT UNIQUE NOT NULL,`
 `PRIMARY KEY(id),`
 `FOREIGN KEY(room_owner_address_id) REFERENCES room_owner_address(id) ON DELETE CASCADE`
`);`

CREATION OF ROOM ADDRESS TABLE



The screenshot shows a MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, etc.), search, and database navigation.
- Status Bar:** Shows "Limit to 1000 rows".
- Query Editor:** Displays the SQL code for creating the `room_address` table. The code is numbered from 201 to 214.

```
201 • CREATE TABLE room_address (
202     id INT UNIQUE NOT NULL,
203     house_no INT,
204     street VARCHAR(50),
205     postcode INT,
206     state VARCHAR(50),
207     country_id INT NOT NULL,
208     room_owner_id INT UNIQUE NOT NULL,
209     amenities_id INT UNIQUE NOT NULL,
210     PRIMARY KEY(id),
211     FOREIGN KEY(room_owner_id) REFERENCES room_owner(id) ON DELETE CASCADE,
212     FOREIGN KEY(country_id) REFERENCES country(id),
213     FOREIGN KEY(amenities_id) REFERENCES amenities(id) ON DELETE CASCADE
214 );
```

CREATION OF ROOM TABLE

```
CREATE TABLE room (
    id INT UNIQUE NOT NULL,
    home_type VARCHAR(50),
    total_bedrooms INT,
    total_bathrooms INT,
    published_date datetime,
    room_address_id INT UNIQUE NOT NULL,
    rules_id INT UNIQUE NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY(room_address_id) REFERENCES room_address(id) ON DELETE CASCADE,
    FOREIGN KEY(rules_id) REFERENCES rules(id) ON DELETE CASCADE
);
```

CREATION OF RESERVATION TABLE

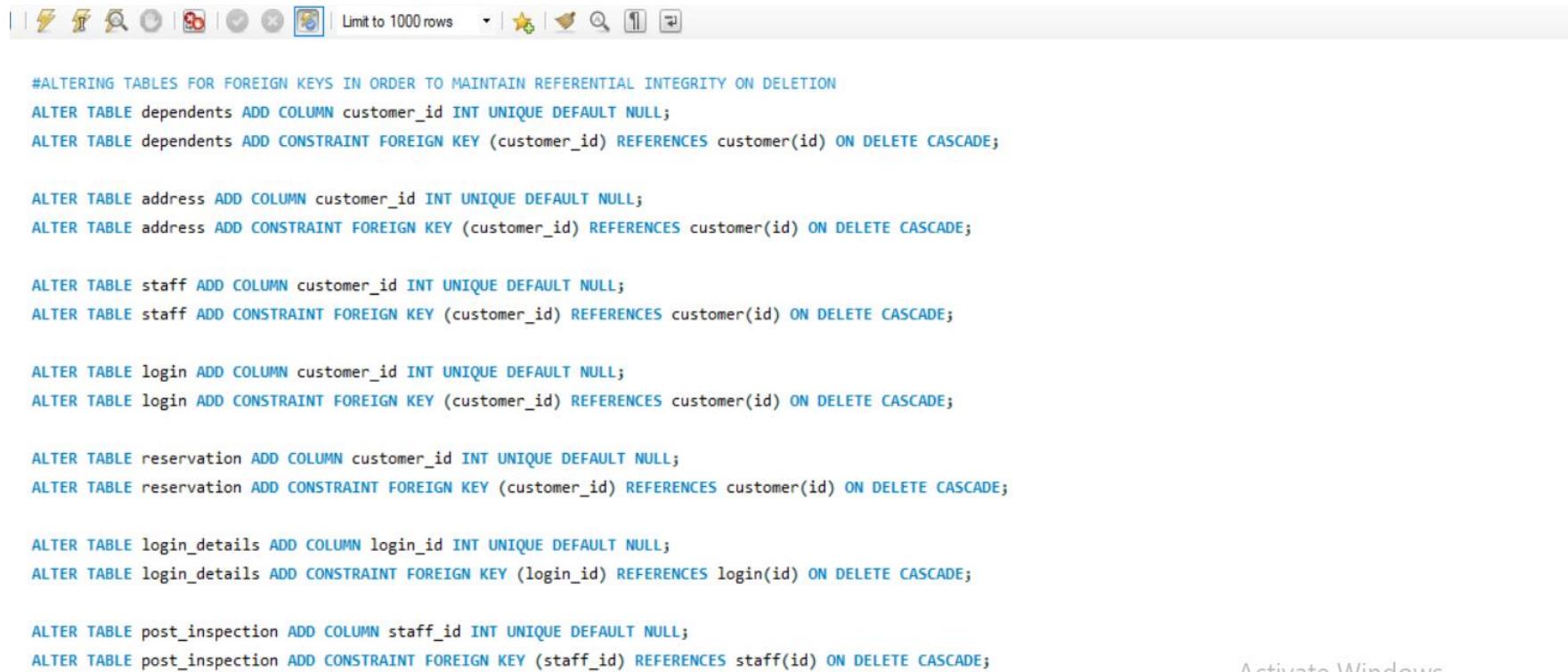
```
--  
243  
244 • CREATE TABLE reservation (  
245     id INT UNIQUE NOT NULL,  
246     reservation_type VARCHAR(50),  
247     booking_date DATETIME,  
248     start_date DATETIME,  
249     end_date DATETIME,  
250     last_update DATETIME,  
251     room_id INT UNIQUE NOT NULL,  
252     payment_id INT UNIQUE NOT NULL,  
253     PRIMARY KEY(id),  
254     FOREIGN KEY(room_id) REFERENCES room(id) ON DELETE CASCADE,  
255     FOREIGN KEY(payment_id) REFERENCES payment(id) ON DELETE CASCADE  
256 );  
257  
258
```

CREATION OF CUSTOMER TABLE

```
257
258
259 • CREATE TABLE customer (
260     id INTEGER UNIQUE NOT NULL,
261     first_name VARCHAR(500),
262     last_name VARCHAR(500),
263     gender VARCHAR(50),
264     age INT,
265     job_title VARCHAR(500),
266     job_industry VARCHAR(500),
267     dependents_id INT UNIQUE NOT NULL,
268     address_id INT UNIQUE NOT NULL,
269     staff_id INT UNIQUE NOT NULL,
270     login_id INT UNIQUE NOT NULL,
271     reservation_id INT UNIQUE NOT NULL,
272     PRIMARY KEY(id),
273     FOREIGN KEY (dependents_id) REFERENCES dependents(id) ON DELETE CASCADE,
274     FOREIGN KEY (address_id) REFERENCES address(id) ON DELETE CASCADE,
275     FOREIGN KEY (staff_id) REFERENCES staff(id) ON DELETE CASCADE,
276     FOREIGN KEY (login_id) REFERENCES login(id) ON DELETE CASCADE,
277     FOREIGN KEY (reservation_id) REFERENCES reservation(id) ON DELETE CASCADE
278 );
279
```

ALTERING TABLES

Tables are altered to add foreign keys in order to maintain referential integrity on deletion of records.



The screenshot shows a MySQL Workbench interface with a query editor containing the following SQL code:

```
#ALTERING TABLES FOR FOREIGN KEYS IN ORDER TO MAINTAIN REFERENTIAL INTEGRITY ON DELETION
ALTER TABLE dependents ADD COLUMN customer_id INT UNIQUE DEFAULT NULL;
ALTER TABLE dependents ADD CONSTRAINT FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE;

ALTER TABLE address ADD COLUMN customer_id INT UNIQUE DEFAULT NULL;
ALTER TABLE address ADD CONSTRAINT FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE;

ALTER TABLE staff ADD COLUMN customer_id INT UNIQUE DEFAULT NULL;
ALTER TABLE staff ADD CONSTRAINT FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE;

ALTER TABLE login ADD COLUMN customer_id INT UNIQUE DEFAULT NULL;
ALTER TABLE login ADD CONSTRAINT FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE;

ALTER TABLE reservation ADD COLUMN customer_id INT UNIQUE DEFAULT NULL;
ALTER TABLE reservation ADD CONSTRAINT FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE;

ALTER TABLE login_details ADD COLUMN login_id INT UNIQUE DEFAULT NULL;
ALTER TABLE login_details ADD CONSTRAINT FOREIGN KEY (login_id) REFERENCES login(id) ON DELETE CASCADE;

ALTER TABLE post_inspection ADD COLUMN staff_id INT UNIQUE DEFAULT NULL;
ALTER TABLE post_inspection ADD CONSTRAINT FOREIGN KEY (staff_id) REFERENCES staff(id) ON DELETE CASCADE;
```

Activate Windows

ALTERING TABLES

```
304 • ALTER TABLE amenities_after_use ADD COLUMN post_inspection_id INT UNIQUE DEFAULT NULL;
305 • ALTER TABLE amenities_after_use ADD CONSTRAINT FOREIGN KEY (post_inspection_id) REFERENCES post_inspection(id) ON DELETE CASCADE;
306
307 • ALTER TABLE facilities_after_use ADD COLUMN post_inspection_id INT UNIQUE DEFAULT NULL;
308 • ALTER TABLE facilities_after_use ADD CONSTRAINT FOREIGN KEY (post_inspection_id) REFERENCES post_inspection(id) ON DELETE CASCADE;
309
310 • ALTER TABLE entertainment_after_use ADD COLUMN post_inspection_id INT UNIQUE DEFAULT NULL;
311 • ALTER TABLE entertainment_after_use ADD CONSTRAINT FOREIGN KEY (post_inspection_id) REFERENCES post_inspection(id) ON DELETE CASCADE;
312
313 • ALTER TABLE security_questions ADD COLUMN login_details_id INT UNIQUE DEFAULT NULL;
314 • ALTER TABLE security_questions ADD CONSTRAINT FOREIGN KEY (login_details_id) REFERENCES login_details(id) ON DELETE CASCADE;
315
316 • ALTER TABLE room ADD COLUMN reservation_id INT UNIQUE DEFAULT NULL;
317 • ALTER TABLE room ADD CONSTRAINT FOREIGN KEY (reservation_id) REFERENCES reservation(id) ON DELETE CASCADE;
318
319 • ALTER TABLE rules ADD COLUMN room_id INT UNIQUE DEFAULT NULL;
320 • ALTER TABLE rules ADD CONSTRAINT FOREIGN KEY (room_id) REFERENCES room(id) ON DELETE CASCADE;
321
322 • ALTER TABLE room_address ADD COLUMN room_id INT UNIQUE DEFAULT NULL;
323 • ALTER TABLE room_address ADD CONSTRAINT FOREIGN KEY (room_id) REFERENCES room(id) ON DELETE CASCADE;
324
325 • ALTER TABLE room_owner ADD COLUMN room_address_id INT UNIQUE DEFAULT NULL;
326 • ALTER TABLE room_owner ADD CONSTRAINT FOREIGN KEY (room_address_id) REFERENCES room_address(id) ON DELETE CASCADE;
---
```

Activate Windows
Go to Settings to activate Windows.

ALTERING TABLES

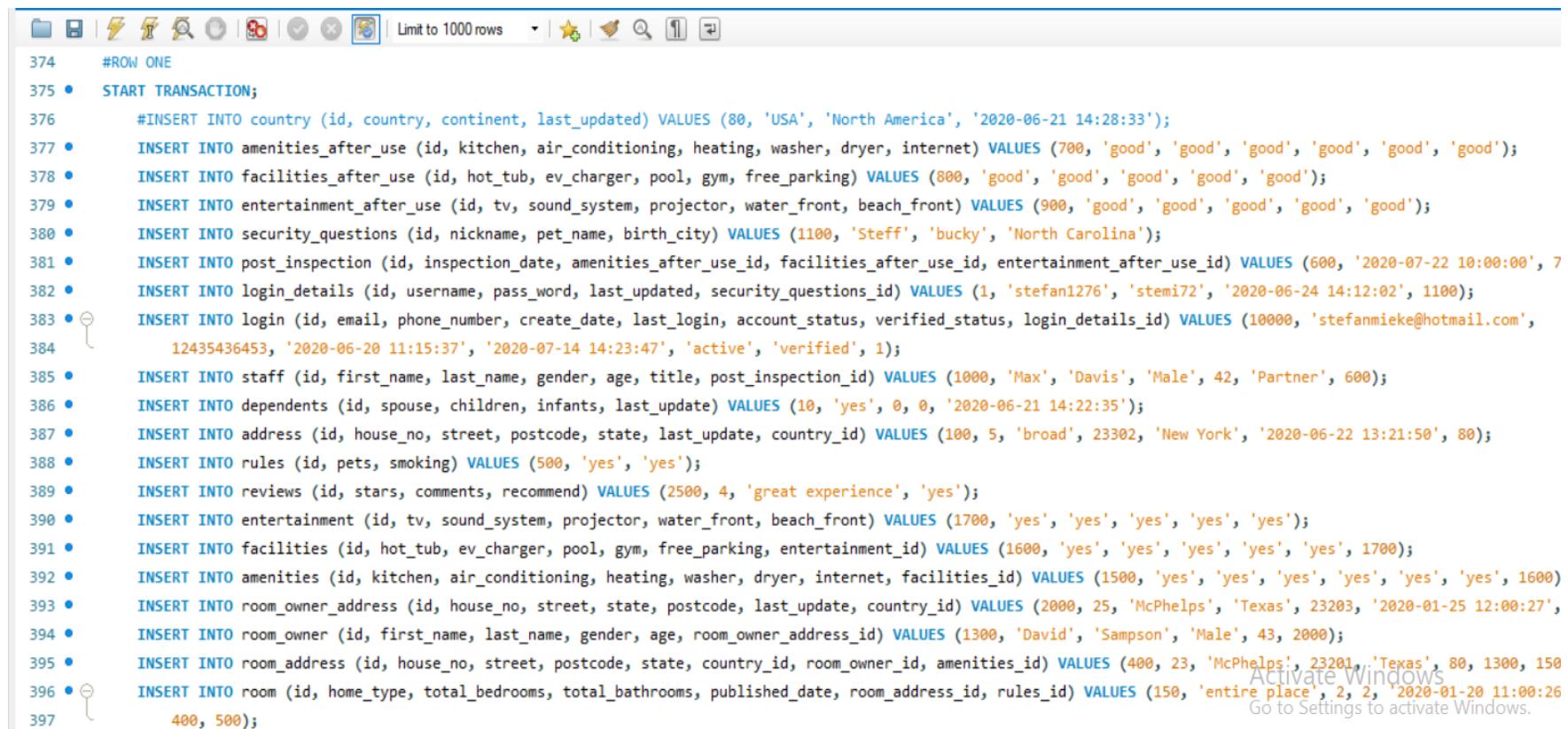
```
327
328 • ALTER TABLE room_owner_address ADD COLUMN room_owner_id INT UNIQUE DEFAULT NULL;
329 • ALTER TABLE room_owner_address ADD CONSTRAINT FOREIGN KEY (room_owner_id) REFERENCES room_owner(id) ON DELETE CASCADE;
330
331 • ALTER TABLE payment ADD COLUMN reservation_id INT UNIQUE DEFAULT NULL;
332 • ALTER TABLE payment ADD CONSTRAINT FOREIGN KEY (reservation_id) REFERENCES reservation(id) ON DELETE CASCADE;
333
334 • ALTER TABLE reviews ADD COLUMN payment_id INT UNIQUE DEFAULT NULL;
335 • ALTER TABLE reviews ADD CONSTRAINT FOREIGN KEY (payment_id) REFERENCES payment(id) ON DELETE CASCADE;
336
337 • ALTER TABLE facilities ADD COLUMN amenities_id INT UNIQUE DEFAULT NULL;
338 • ALTER TABLE facilities ADD CONSTRAINT FOREIGN KEY (amenities_id) REFERENCES amenities(id) ON DELETE CASCADE;
339
340 • ALTER TABLE amenities ADD COLUMN room_address_id INT UNIQUE DEFAULT NULL;
341 • ALTER TABLE amenities ADD CONSTRAINT FOREIGN KEY (room_address_id) REFERENCES room_address(id) ON DELETE CASCADE;
342
343 • ALTER TABLE entertainment ADD COLUMN facilities_id INT UNIQUE DEFAULT NULL;
344 • ALTER TABLE entertainment ADD CONSTRAINT FOREIGN KEY (facilities_id) REFERENCES facilities(id) ON DELETE CASCADE;
345
```

INSERTING DATA INTO COUNTRY TABLE

Data was entered into this table separately because it has a N:M relationship.

```
347      #INSERTING INTO COUNTRY TABLE FIRST BECAUSE IT HAS A CN:CM RELATIONSHIP UNLIKE OTHER TABLES
348 •  START TRANSACTION;
349 •      INSERT INTO country (id, country, continent, last_updated) VALUES (80, 'USA', 'North America', '2020-06-21 14:28:33');
350 •      INSERT INTO country (id, country, continent, last_updated) VALUES (81, 'Canada', 'North America', '2020-07-22 17:38:33');
351 •      INSERT INTO country (id, country, continent, last_updated) VALUES (82, 'Australia', 'North America', '2020-08-22 17:38:33');
352 •      INSERT INTO country (id, country, continent, last_updated) VALUES (83, 'Nigeria', 'Africa', '2020-08-22 19:34:35');
353 •      INSERT INTO country (id, country, continent, last_updated) VALUES (84, 'Germany', 'Europe', '2020-08-22 19:34:35');
354 •      INSERT INTO country (id, country, continent, last_updated) VALUES (85, 'Poland', 'Europe', '2020-08-22 19:34:35');
355 •      INSERT INTO country (id, country, continent, last_updated) VALUES (86, 'France', 'Europe', '2020-08-22 19:34:36');
356 •      INSERT INTO country (id, country, continent, last_updated) VALUES (87, 'Greece', 'Europe', '2020-08-22 19:34:36');
357 •      INSERT INTO country (id, country, continent, last_updated) VALUES (88, 'Spain', 'Europe', '2020-08-22 19:34:37');
358 •      INSERT INTO country (id, country, continent, last_updated) VALUES (89, 'Benin Republic', 'Africa', '2020-08-22 19:34:37');
359 •      INSERT INTO country (id, country, continent, last_updated) VALUES (90, 'Italy', 'Europe', '2020-08-22 19:34:38');
360 •      INSERT INTO country (id, country, continent, last_updated) VALUES (91, 'Kenya', 'Africa', '2020-08-22 19:34:38');
361 •      INSERT INTO country (id, country, continent, last_updated) VALUES (92, 'Ivory Coast', 'Africa', '2020-08-22 19:34:39');
362 •      INSERT INTO country (id, country, continent, last_updated) VALUES (93, 'Algeria', 'Africa', '2020-08-22 19:34:39');
363 •      INSERT INTO country (id, country, continent, last_updated) VALUES (94, 'Togo', 'Africa', '2020-08-22 19:34:40');
364 •      INSERT INTO country (id, country, continent, last_updated) VALUES (95, 'South Africa', 'Africa', '2020-08-22 19:34:40');
365 •      INSERT INTO country (id, country, continent, last_updated) VALUES (96, 'Tanzania', 'Africa', '2020-08-22 19:34:41');
366 •      INSERT INTO country (id, country, continent, last_updated) VALUES (97, 'Russia', 'Europe', '2020-08-22 19:34:42');
367 •      INSERT INTO country (id, country, continent, last_updated) VALUES (98, 'Denmark', 'Europe', '2020-08-22 19:34:42');
368 •      INSERT INTO country (id, country, continent, last_updated) VALUES (99, 'Haiti', 'North America', '2020-08-22 19:34:43');
369 •  COMMIT;
```

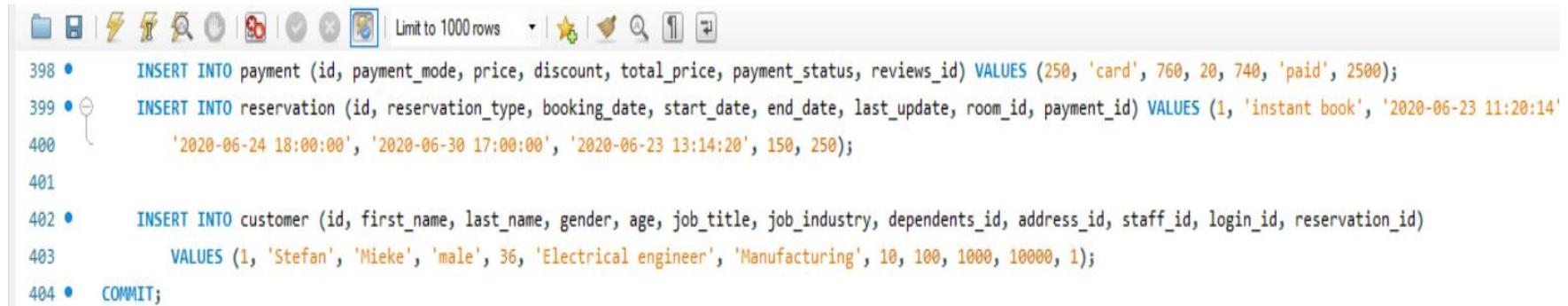
INSERTING INTO FIRST ROWS OF ALL TABLES



The screenshot shows a MySQL Workbench interface with a query editor containing a large block of SQL code. The code consists of approximately 397 numbered lines, starting with '#ROW ONE' and ending with '400, 500);'. The code is used to insert data into various tables such as country, amenities_after_use, facilities_after_use, entertainment_after_use, security_questions, post_inspection, login_details, login, staff, dependents, address, rules, reviews, entertainment, facilities, amenities, room_owner_address, room_owner, room_address, and room. The data inserted includes details like country names, facility types, user information, and review scores. A watermark for 'Activate Windows' is visible in the bottom right corner of the interface.

```
374 #ROW ONE
375 • START TRANSACTION;
376     #INSERT INTO country (id, country, continent, last_updated) VALUES (80, 'USA', 'North America', '2020-06-21 14:28:33');
377 •     INSERT INTO amenities_after_use (id, kitchen, air_conditioning, heating, washer, dryer, internet) VALUES (700, 'good', 'good', 'good', 'good', 'good', 'good');
378 •     INSERT INTO facilities_after_use (id, hot_tub, ev_charger, pool, gym, free_parking) VALUES (800, 'good', 'good', 'good', 'good', 'good');
379 •     INSERT INTO entertainment_after_use (id, tv, sound_system, projector, water_front, beach_front) VALUES (900, 'good', 'good', 'good', 'good', 'good');
380 •     INSERT INTO security_questions (id, nickname, pet_name, birth_city) VALUES (1100, 'Steff', 'bucky', 'North Carolina');
381 •     INSERT INTO post_inspection (id, inspection_date, amenities_after_use_id, facilities_after_use_id, entertainment_after_use_id) VALUES (600, '2020-07-22 10:00:00', 700, 800, 900);
382 •     INSERT INTO login_details (id, username, pass_word, last_updated, security_questions_id) VALUES (1, 'stefan1276', 'stemii72', '2020-06-24 14:12:02', 1100);
383 •     INSERT INTO login (id, email, phone_number, create_date, last_login, account_status, verified_status, login_details_id) VALUES (10000, 'stefanmieke@hotmail.com', 12435436453, '2020-06-20 11:15:37', '2020-07-14 14:23:47', 'active', 'verified', 1);
384     INSERT INTO staff (id, first_name, last_name, gender, age, title, post_inspection_id) VALUES (1000, 'Max', 'Davis', 'Male', 42, 'Partner', 600);
385 •     INSERT INTO dependents (id, spouse, children, infants, last_update) VALUES (10, 'yes', 0, 0, '2020-06-21 14:22:35');
386 •     INSERT INTO address (id, house_no, street, postcode, state, last_update, country_id) VALUES (100, 5, 'broad', 23302, 'New York', '2020-06-22 13:21:50', 80);
387 •     INSERT INTO rules (id, pets, smoking) VALUES (500, 'yes', 'yes');
388 •     INSERT INTO reviews (id, stars, comments, recommend) VALUES (2500, 4, 'great experience', 'yes');
389 •     INSERT INTO entertainment (id, tv, sound_system, projector, water_front, beach_front) VALUES (1700, 'yes', 'yes', 'yes', 'yes', 'yes');
390 •     INSERT INTO facilities (id, hot_tub, ev_charger, pool, gym, free_parking, entertainment_id) VALUES (1600, 'yes', 'yes', 'yes', 'yes', 'yes', 1700);
391 •     INSERT INTO amenities (id, kitchen, air_conditioning, heating, washer, dryer, internet, facilities_id) VALUES (1500, 'yes', 'yes', 'yes', 'yes', 'yes', 'yes', 1600);
392 •     INSERT INTO room_owner_address (id, house_no, street, state, postcode, last_update, country_id) VALUES (2000, 25, 'McPhelps', 'Texas', 23203, '2020-01-25 12:00:27', 2000);
393 •     INSERT INTO room_owner (id, first_name, last_name, gender, age, room_owner_address_id) VALUES (1300, 'David', 'Sampson', 'Male', 43, 2000);
394 •     INSERT INTO room_address (id, house_no, street, postcode, state, country_id, room_owner_id, amenities_id) VALUES (400, 23, 'McPhelps', 23201, 'Texas', 80, 1300, 150);
395 •     INSERT INTO room (id, home_type, total_bedrooms, total_bathrooms, published_date, room_address_id, rules_id) VALUES (150, 'entire place', 2, 2, '2020-01-20 11:00:26', 400, 500);
396 •
397 •
```

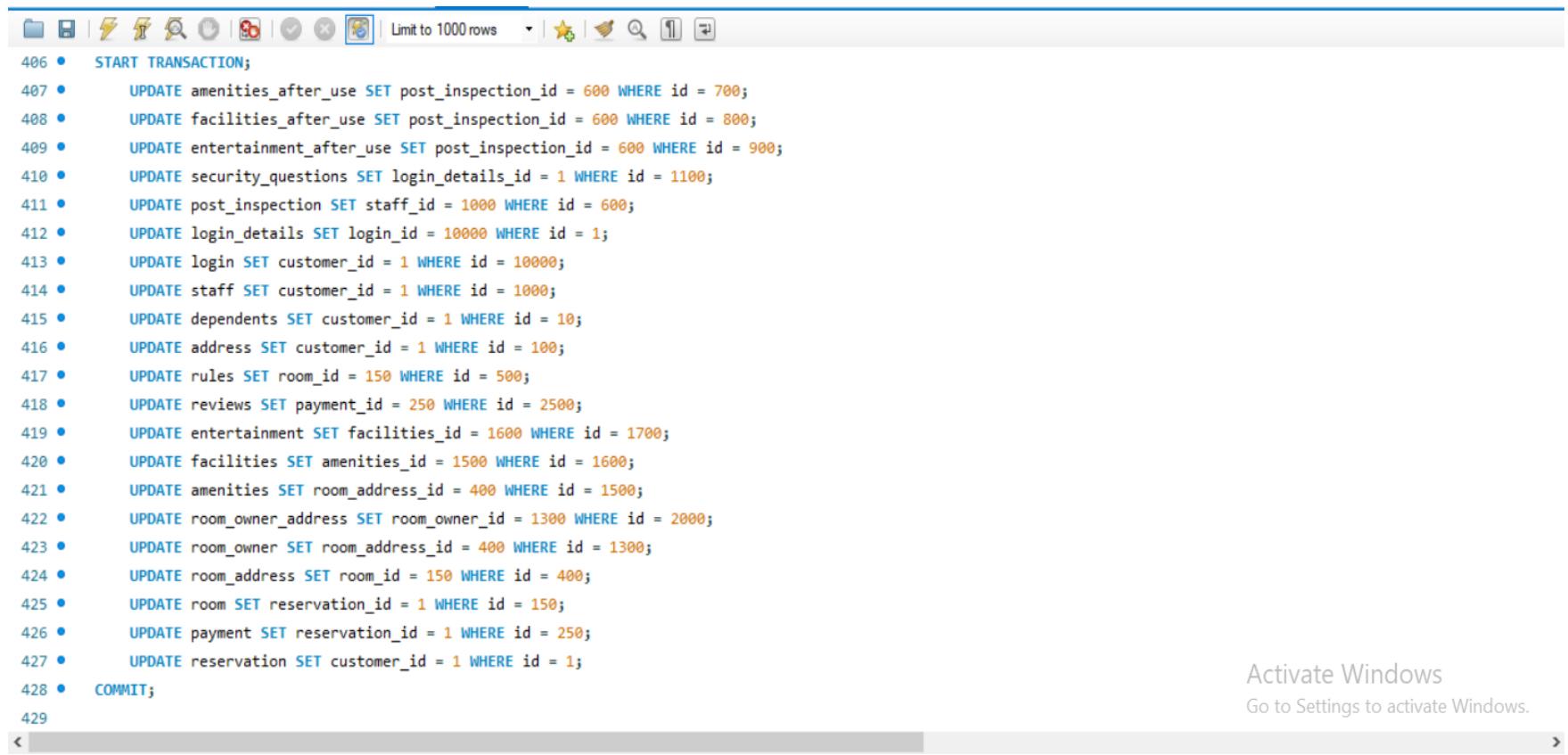
END OF INSERTION INTO FIRST ROW OF ALL TABLES



The screenshot shows a MySQL Workbench interface with a query editor window. The window title is 'SQL' and it contains the following SQL code:

```
398 •     INSERT INTO payment (id, payment_mode, price, discount, total_price, payment_status, reviews_id) VALUES (250, 'card', 760, 20, 740, 'paid', 2500);
399 •     INSERT INTO reservation (id, reservation_type, booking_date, start_date, end_date, last_update, room_id, payment_id) VALUES (1, 'instant book', '2020-06-23 11:20:14'
400             '2020-06-24 18:00:00', '2020-06-30 17:00:00', '2020-06-23 13:14:20', 150, 250);
401
402 •     INSERT INTO customer (id, first_name, last_name, gender, age, job_title, job_industry, dependents_id, address_id, staff_id, login_id, reservation_id)
403             VALUES (1, 'Stefan', 'Mieke', 'male', 36, 'Electrical engineer', 'Manufacturing', 10, 100, 1000, 10000, 1);
404 •     COMMIT;
```

UPDATING FIRST ROW IN ALL TABLES

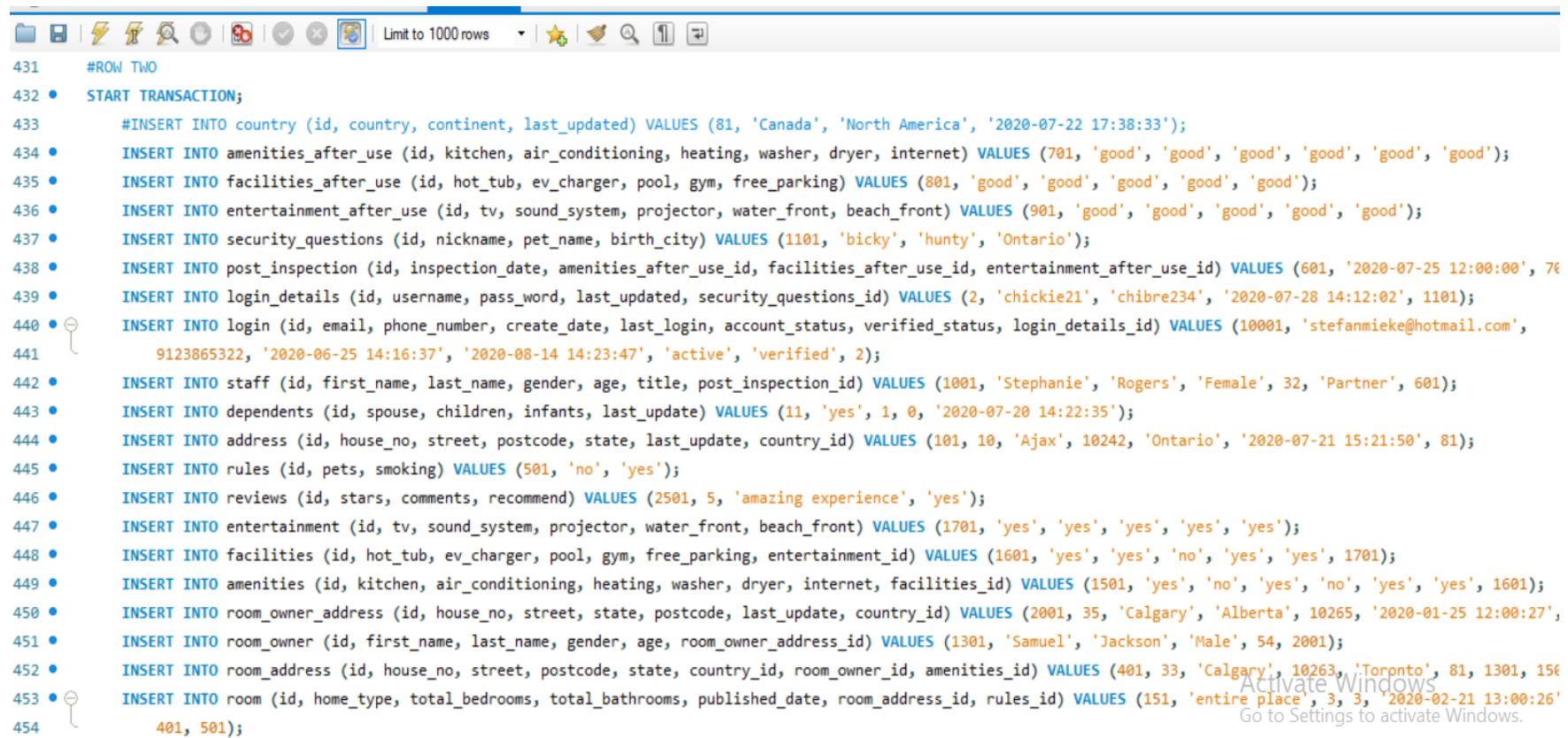


The screenshot shows a MySQL Workbench interface with a transaction log window. The log contains 428 UPDATE statements, each with a line number and a timestamp. The statements are all identical, updating the first row of various tables in the database. The tables include amenities_after_use, facilities_after_use, entertainment_after_use, security_questions, post_inspection, login_details, login, staff, dependents, address, rules, reviews, entertainment, facilities, amenities, room_owner_address, room_owner, room_address, room, payment, and reservation.

```
406 •  START TRANSACTION;
407 •      UPDATE amenities_after_use SET post_inspection_id = 600 WHERE id = 700;
408 •      UPDATE facilities_after_use SET post_inspection_id = 600 WHERE id = 800;
409 •      UPDATE entertainment_after_use SET post_inspection_id = 600 WHERE id = 900;
410 •      UPDATE security_questions SET login_details_id = 1 WHERE id = 1100;
411 •      UPDATE post_inspection SET staff_id = 1000 WHERE id = 600;
412 •      UPDATE login_details SET login_id = 10000 WHERE id = 1;
413 •      UPDATE login SET customer_id = 1 WHERE id = 10000;
414 •      UPDATE staff SET customer_id = 1 WHERE id = 1000;
415 •      UPDATE dependents SET customer_id = 1 WHERE id = 10;
416 •      UPDATE address SET customer_id = 1 WHERE id = 100;
417 •      UPDATE rules SET room_id = 150 WHERE id = 500;
418 •      UPDATE reviews SET payment_id = 250 WHERE id = 2500;
419 •      UPDATE entertainment SET facilities_id = 1600 WHERE id = 1700;
420 •      UPDATE facilities SET amenities_id = 1500 WHERE id = 1600;
421 •      UPDATE amenities SET room_address_id = 400 WHERE id = 1500;
422 •      UPDATE room_owner_address SET room_owner_id = 1300 WHERE id = 2000;
423 •      UPDATE room_owner SET room_address_id = 400 WHERE id = 1300;
424 •      UPDATE room_address SET room_id = 150 WHERE id = 400;
425 •      UPDATE room SET reservation_id = 1 WHERE id = 150;
426 •      UPDATE payment SET reservation_id = 1 WHERE id = 250;
427 •      UPDATE reservation SET customer_id = 1 WHERE id = 1;
428 •  COMMIT;
```

Activate Windows
Go to Settings to activate Windows.

INSERTING INTO SECOND ROW OF ALL TABLES



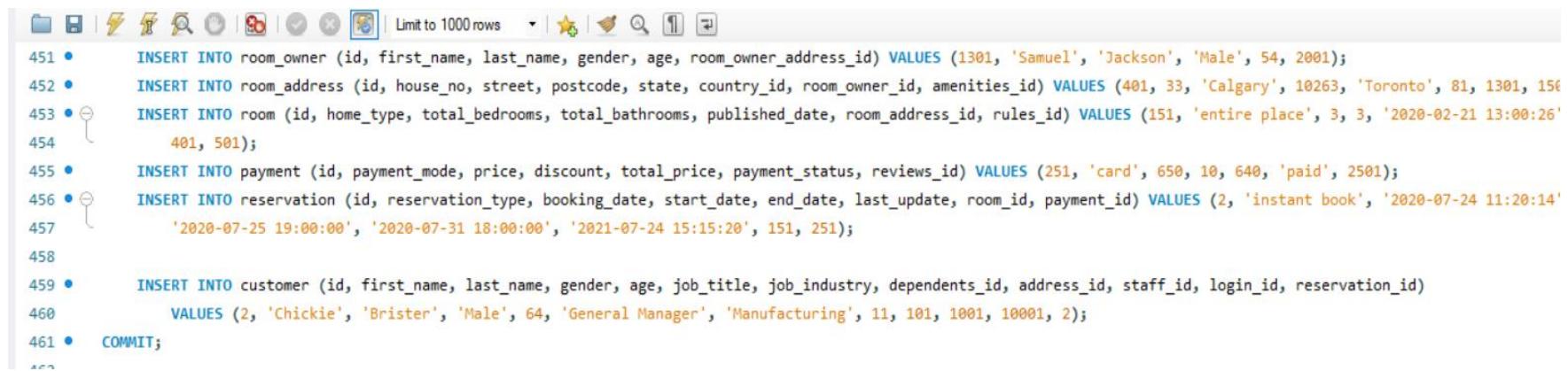
The screenshot shows a MySQL Workbench interface with a query editor containing a large block of SQL code. The code consists of approximately 454 numbered lines, starting with '#ROW TWO' and ending with '501);'. The code is a series of INSERT statements intended to insert data into every table in the database, specifically targeting the second row of each table. The tables mentioned include country, amenities_after_use, facilities, entertainment_after_use, security_questions, post_inspection, login_details, login, staff, dependents, address, rules, reviews, entertainment, facilities, amenities, room_owner_address, room_owner, room_address, and room. The values inserted are mostly 'good' or 'yes' for various attributes like 'air_conditioning', 'heating', 'washer', etc. The code is heavily annotated with numerous comments starting with '#'. The interface includes a toolbar at the top with various icons for file operations, and the bottom right corner features a watermark for 'Activate Windows'.

```
431 #ROW TWO
432 • START TRANSACTION;
433     #INSERT INTO country (id, country, continent, last_updated) VALUES (81, 'Canada', 'North America', '2020-07-22 17:38:33');
434 •     INSERT INTO amenities_after_use (id, kitchen, air_conditioning, heating, washer, dryer, internet) VALUES (701, 'good', 'good', 'good', 'good', 'good', 'good');
435 •     INSERT INTO facilities_after_use (id, hot_tub, ev_charger, pool, gym, free_parking) VALUES (801, 'good', 'good', 'good', 'good', 'good');
436 •     INSERT INTO entertainment_after_use (id, tv, sound_system, projector, water_front, beach_front) VALUES (901, 'good', 'good', 'good', 'good', 'good');
437 •     INSERT INTO security_questions (id, nickname, pet_name, birth_city) VALUES (1101, 'bicky', 'hunty', 'Ontario');
438 •     INSERT INTO post_inspection (id, inspection_date, amenities_after_use_id, facilities_after_use_id, entertainment_after_use_id) VALUES (601, '2020-07-25 12:00:00', 76
439 •     INSERT INTO login_details (id, username, pass_word, last_updated, security_questions_id) VALUES (2, 'chickie21', 'chibre234', '2020-07-28 14:12:02', 1101);
440 •     INSERT INTO login (id, email, phone_number, create_date, last_login, account_status, verified_status, login_details_id) VALUES (10001, 'stefanmieke@hotmail.com',
441         9123865322, '2020-06-25 14:16:37', '2020-08-14 14:23:47', 'active', 'verified', 2);
442 •     INSERT INTO staff (id, first_name, last_name, gender, age, title, post_inspection_id) VALUES (1001, 'Stephanie', 'Rogers', 'Female', 32, 'Partner', 601);
443 •     INSERT INTO dependents (id, spouse, children, infants, last_update) VALUES (11, 'yes', 1, 0, '2020-07-20 14:22:35');
444 •     INSERT INTO address (id, house_no, street, postcode, state, last_update, country_id) VALUES (101, 10, 'Ajax', 10242, 'Ontario', '2020-07-21 15:21:50', 81);
445 •     INSERT INTO rules (id, pets, smoking) VALUES (501, 'no', 'yes');
446 •     INSERT INTO reviews (id, stars, comments, recommend) VALUES (2501, 5, 'amazing experience', 'yes');
447 •     INSERT INTO entertainment (id, tv, sound_system, projector, water_front, beach_front) VALUES (1701, 'yes', 'yes', 'yes', 'yes', 'yes');
448 •     INSERT INTO facilities (id, hot_tub, ev_charger, pool, gym, free_parking, entertainment_id) VALUES (1601, 'yes', 'yes', 'no', 'yes', 'yes', 1701);
449 •     INSERT INTO amenities (id, kitchen, air_conditioning, heating, washer, dryer, internet, facilities_id) VALUES (1501, 'yes', 'no', 'yes', 'no', 'yes', 'yes', 1601);
450 •     INSERT INTO room_owner_address (id, house_no, street, state, postcode, last_update, country_id) VALUES (2001, 35, 'Calgary', 'Alberta', 10265, '2020-01-25 12:00:27',
451 •     INSERT INTO room_owner (id, first_name, last_name, gender, age, room_owner_address_id) VALUES (1301, 'Samuel', 'Jackson', 'Male', 54, 2001);
452 •     INSERT INTO room_address (id, house_no, street, postcode, state, country_id, room_owner_id, amenities_id) VALUES (401, 33, 'Calgary', 10263, 'Toronto', 81, 1301, 156
453 •     INSERT INTO room (id, home_type, total_bedrooms, total_bathrooms, published_date, room_address_id, rules_id) VALUES (151, 'entire place', 3, 3, '2020-02-21 13:00:26'
454         401, 501);
```

Activate Windows

Go to Settings to activate Windows.

END OF INSERTION INTO SECOND ROW OF ALL TABLES

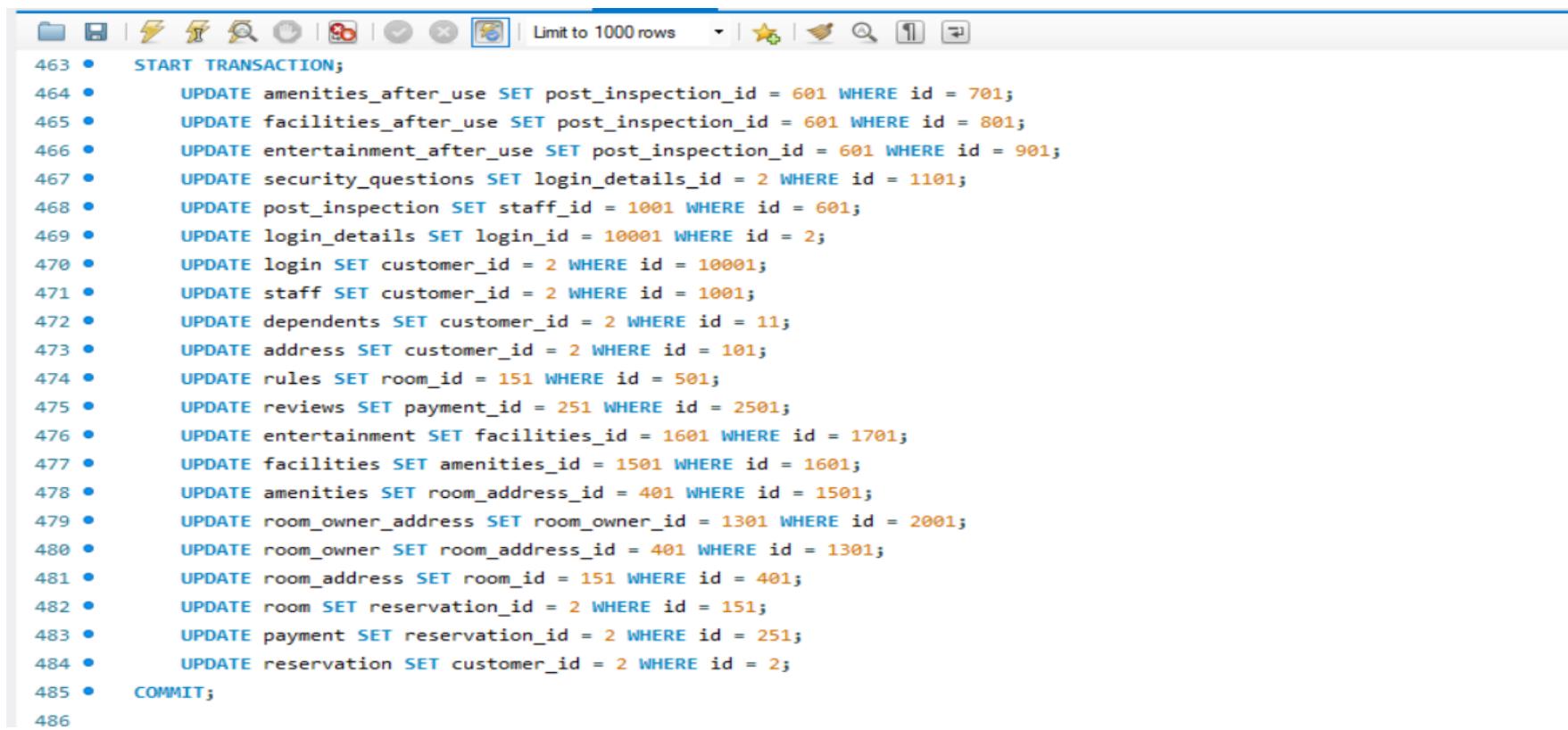


The screenshot shows a MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, etc.), search, and connection management.
- Status Bar:** Shows "Limit to 1000 rows".
- Script Area:** Displays an SQL script with numbered lines (451 to 461) containing various `INSERT` statements and a `COMMIT` command.

```
451 •     INSERT INTO room_owner (id, first_name, last_name, gender, age, room_owner_address_id) VALUES (1301, 'Samuel', 'Jackson', 'Male', 54, 2001);
452 •     INSERT INTO room_address (id, house_no, street, postcode, state, country_id, room_owner_id, amenities_id) VALUES (401, 33, 'Calgary', 10263, 'Toronto', 81, 1301, 156
453 •     INSERT INTO room (id, home_type, total_bedrooms, total_bathrooms, published_date, room_address_id, rules_id) VALUES (151, 'entire place', 3, 3, '2020-02-21 13:00:26'
454     401, 501);
455 •     INSERT INTO payment (id, payment_mode, price, discount, total_price, payment_status, reviews_id) VALUES (251, 'card', 650, 10, 640, 'paid', 2501);
456 •     INSERT INTO reservation (id, reservation_type, booking_date, start_date, end_date, last_update, room_id, payment_id) VALUES (2, 'instant book', '2020-07-24 11:20:14'
457     '2020-07-25 19:00:00', '2020-07-31 18:00:00', '2021-07-24 15:15:20', 151, 251);
458
459 •     INSERT INTO customer (id, first_name, last_name, gender, age, job_title, job_industry, dependents_id, address_id, staff_id, login_id, reservation_id)
460     VALUES (2, 'Chickie', 'Brister', 'Male', 64, 'General Manager', 'Manufacturing', 11, 101, 1001, 10001, 2);
461 •     COMMIT;
```

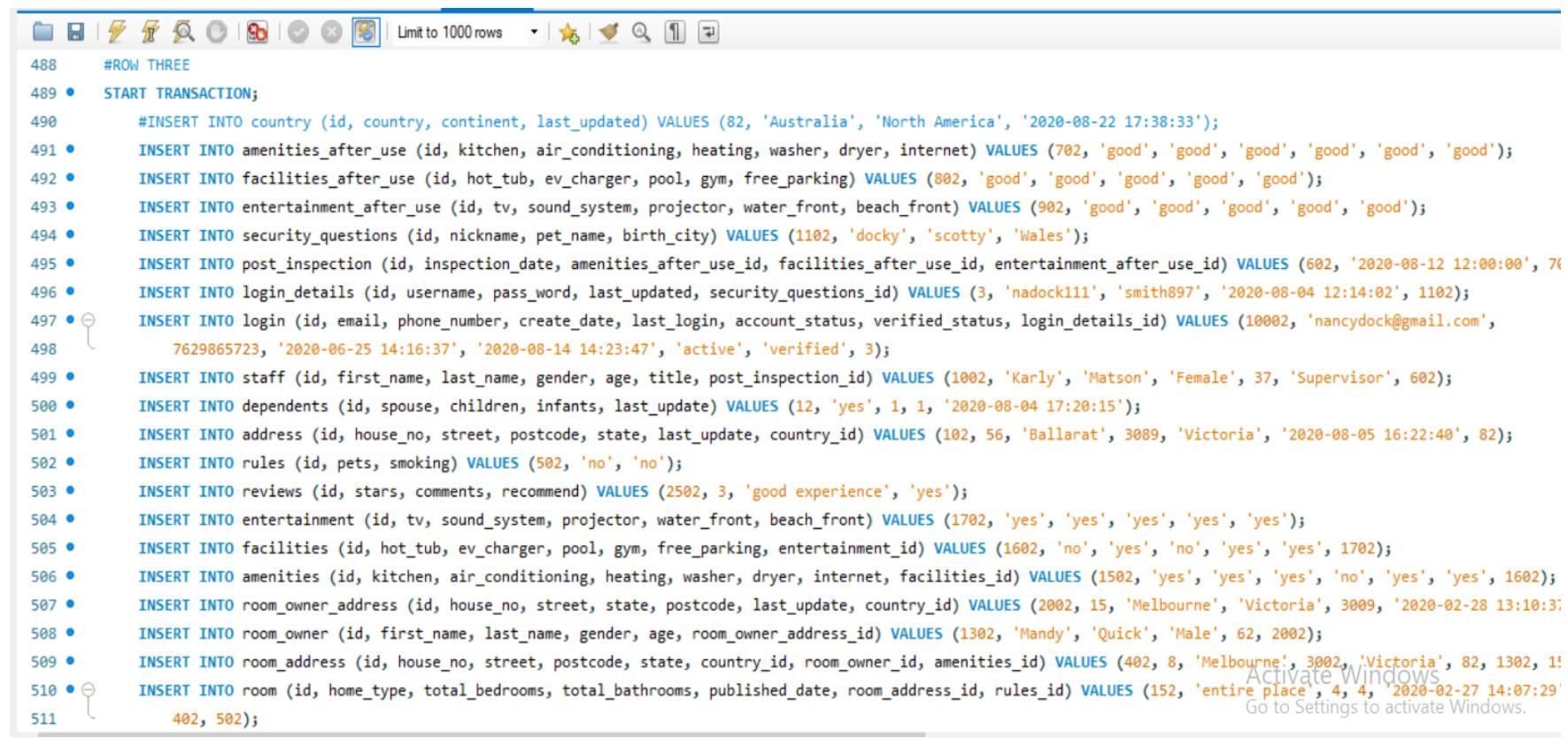
UPDATING SECOND ROW OF ALL TABLES



The screenshot shows a MySQL Workbench interface with a transaction log window. The log contains 486 lines of SQL code, each starting with a line number and a bullet point. The code consists primarily of UPDATE statements targeting various tables such as amenities_after_use, facilities_after_use, entertainment_after_use, security_questions, post_inspection, login_details, login, staff, dependents, address, rules, reviews, entertainment, facilities, amenities, room_owner_address, room_owner, room_address, room, payment, and reservation. The WHERE clauses in these statements often specify conditions like id = 2 or id = 1001, indicating they are updating the second row of each respective table. The log also includes a START TRANSACTION; statement at line 463 and a COMMIT; statement at line 485.

```
463 • START TRANSACTION;
464 •     UPDATE amenities_after_use SET post_inspection_id = 601 WHERE id = 701;
465 •     UPDATE facilities_after_use SET post_inspection_id = 601 WHERE id = 801;
466 •     UPDATE entertainment_after_use SET post_inspection_id = 601 WHERE id = 901;
467 •     UPDATE security_questions SET login_details_id = 2 WHERE id = 1101;
468 •     UPDATE post_inspection SET staff_id = 1001 WHERE id = 601;
469 •     UPDATE login_details SET login_id = 10001 WHERE id = 2;
470 •     UPDATE login SET customer_id = 2 WHERE id = 10001;
471 •     UPDATE staff SET customer_id = 2 WHERE id = 1001;
472 •     UPDATE dependents SET customer_id = 2 WHERE id = 11;
473 •     UPDATE address SET customer_id = 2 WHERE id = 101;
474 •     UPDATE rules SET room_id = 151 WHERE id = 501;
475 •     UPDATE reviews SET payment_id = 251 WHERE id = 2501;
476 •     UPDATE entertainment SET facilities_id = 1601 WHERE id = 1701;
477 •     UPDATE facilities SET amenities_id = 1501 WHERE id = 1601;
478 •     UPDATE amenities SET room_address_id = 401 WHERE id = 1501;
479 •     UPDATE room_owner_address SET room_owner_id = 1301 WHERE id = 2001;
480 •     UPDATE room_owner SET room_address_id = 401 WHERE id = 1301;
481 •     UPDATE room_address SET room_id = 151 WHERE id = 401;
482 •     UPDATE room SET reservation_id = 2 WHERE id = 151;
483 •     UPDATE payment SET reservation_id = 2 WHERE id = 251;
484 •     UPDATE reservation SET customer_id = 2 WHERE id = 2;
485 •     COMMIT;
```

INSERTING INTO THIRD ROW OF ALL TABLES



```
488 • #ROW THREE
489 • START TRANSACTION;
490 •     #INSERT INTO country (id, country, continent, last_updated) VALUES (82, 'Australia', 'North America', '2020-08-22 17:38:33');
491 •     INSERT INTO amenities_after_use (id, kitchen, air_conditioning, heating, washer, dryer, internet) VALUES (702, 'good', 'good', 'good', 'good', 'good', 'good');
492 •     INSERT INTO facilities_after_use (id, hot_tub, ev_charger, pool, gym, free_parking) VALUES (802, 'good', 'good', 'good', 'good', 'good');
493 •     INSERT INTO entertainment_after_use (id, tv, sound_system, projector, water_front, beach_front) VALUES (902, 'good', 'good', 'good', 'good', 'good');
494 •     INSERT INTO security_questions (id, nickname, pet_name, birth_city) VALUES (1102, 'dicky', 'scotty', 'Wales');
495 •     INSERT INTO post_inspection (id, inspection_date, amenities_after_use_id, facilities_after_use_id, entertainment_after_use_id) VALUES (602, '2020-08-12 12:00:00', 702, 802, 902);
496 •     INSERT INTO login_details (id, username, pass_word, last_updated, security_questions_id) VALUES (3, 'nadock111', 'smith897', '2020-08-04 12:14:02', 1102);
497 •     INSERT INTO login (id, email, phone_number, create_date, last_login, account_status, verified_status, login_details_id) VALUES (10002, 'nancydock@gmail.com', '7629865723', '2020-06-25 14:16:37', '2020-08-14 14:23:47', 'active', 'verified', 3);
498 •     INSERT INTO staff (id, first_name, last_name, gender, age, title, post_inspection_id) VALUES (1002, 'Karly', 'Matson', 'Female', 37, 'Supervisor', 602);
499 •     INSERT INTO dependents (id, spouse, children, infants, last_update) VALUES (12, 'yes', 1, 1, '2020-08-04 17:20:15');
500 •     INSERT INTO address (id, house_no, street, postcode, state, last_update, country_id) VALUES (102, 56, 'Ballarat', 3089, 'Victoria', '2020-08-05 16:22:40', 82);
501 •     INSERT INTO rules (id, pets, smoking) VALUES (502, 'no', 'no');
502 •     INSERT INTO reviews (id, stars, comments, recommend) VALUES (2502, 3, 'good experience', 'yes');
503 •     INSERT INTO entertainment (id, tv, sound_system, projector, water_front, beach_front) VALUES (1702, 'yes', 'yes', 'yes', 'yes', 'yes');
504 •     INSERT INTO facilities (id, hot_tub, ev_charger, pool, gym, free_parking, entertainment_id) VALUES (1602, 'no', 'yes', 'no', 'yes', 'yes', 1702);
505 •     INSERT INTO amenities (id, kitchen, air_conditioning, heating, washer, dryer, internet, facilities_id) VALUES (1502, 'yes', 'yes', 'yes', 'no', 'yes', 'yes', 1602);
506 •     INSERT INTO room_owner_address (id, house_no, street, state, postcode, last_update, country_id) VALUES (2002, 15, 'Melbourne', 'Victoria', 3009, '2020-02-28 13:10:31');
507 •     INSERT INTO room_owner (id, first_name, last_name, gender, age, room_owner_address_id) VALUES (1302, 'Mandy', 'Quick', 'Male', 62, 2002);
508 •     INSERT INTO room_address (id, house_no, street, postcode, state, country_id, room_owner_id, amenities_id) VALUES (402, 8, 'Melbourne', 3002, 'Victoria', 82, 1302, 1502);
509 •     INSERT INTO room (id, home_type, total_bedrooms, total_bathrooms, published_date, room_address_id, rules_id) VALUES (152, 'entire place', 4, 4, '2020-02-27 14:07:29', 402, 502);
```

Activate Windows
Go to Settings to activate Windows.

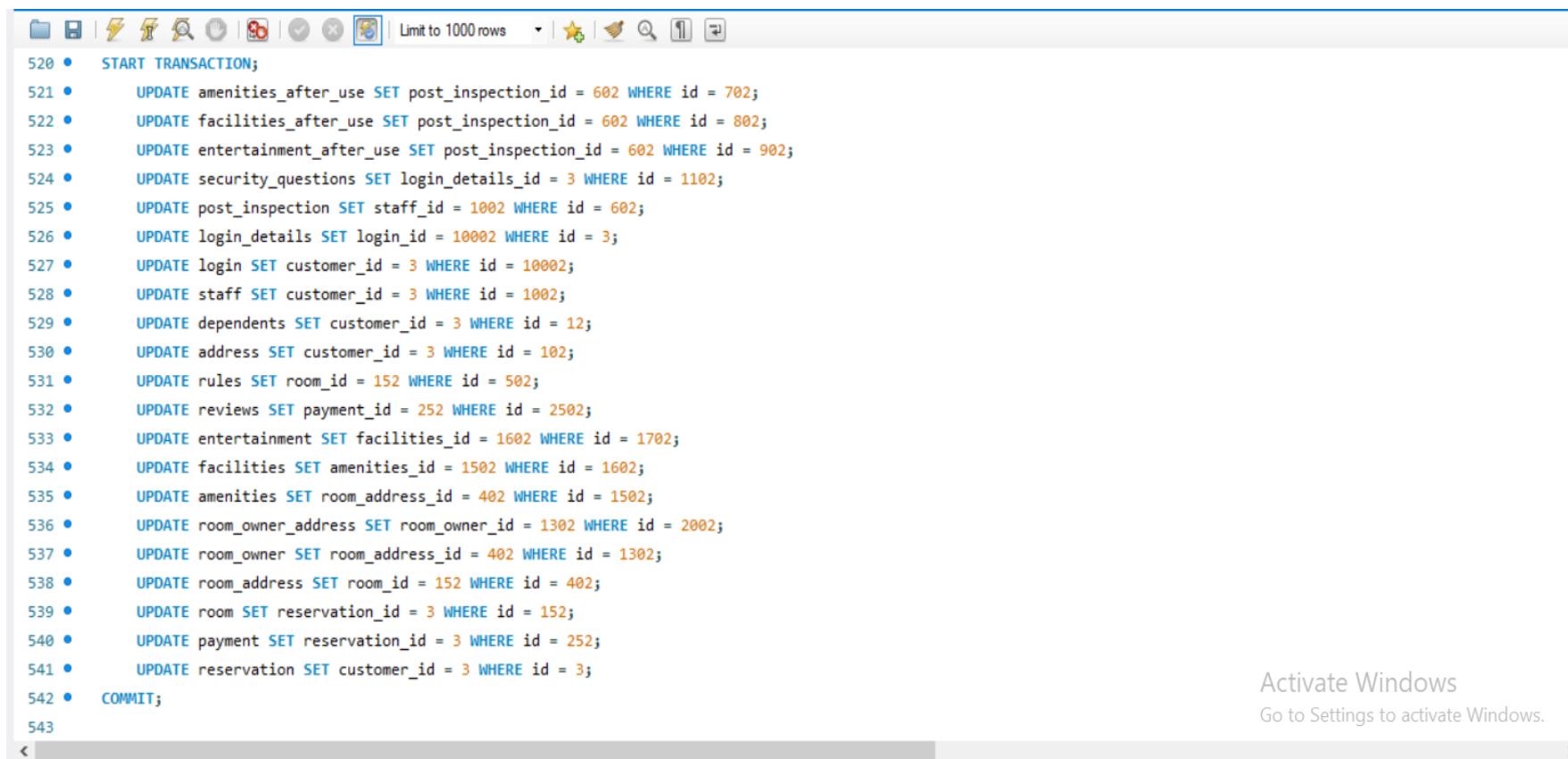
END OF INSERTION INTO THIRD ROW OF ALL TABLES



The screenshot shows a MySQL query editor window with the following details:

- Toolbar:** Includes icons for file operations, search, and refresh.
- Status Bar:** Shows "Limit to 1000 rows".
- Script Content:** Displays numbered SQL statements (509 to 518) for inserting data into various tables. The statements include:
 - INSERT INTO room_address (id, house_no, street, postcode, state, country_id, room_owner_id, amenities_id) VALUES (402, 8, 'Melbourne', 3002, 'Victoria', 82, 1302, 15)
 - INSERT INTO room (id, home_type, total_bedrooms, total_bathrooms, published_date, room_address_id, rules_id) VALUES (152, 'entire place', 4, 4, '2020-02-27 14:07:29', 402, 502);
 - INSERT INTO payment (id, payment_mode, price, discount, total_price, payment_status, reviews_id) VALUES (252, 'check', 560, 20, 540, 'paid', 2502);
 - INSERT INTO reservation (id, reservation_type, booking_date, start_date, end_date, last_update, room_id, payment_id) VALUES (3, 'instant book', '2020-08-02 11:27:34', '2020-08-04 10:00:00', '2020-08-11 09:00:00', '2021-08-03 12:18:29', 152, 252);
 - INSERT INTO customer (id, first_name, last_name, gender, age, job_title, job_industry, dependents_id, address_id, staff_id, login_id, reservation_id) VALUES (3, 'Nancy', 'Docket', 'Male', 56, 'Financial Analyst', 'Financial Services', 12, 102, 1002, 10002, 3);
- Line Numbers:** Shows line numbers 509 through 518 on the left side of the code area.
- Bottom Status:** Shows "COMMIT;" as the final command.

UPDATING THIRD ROW OF ALL TABLES

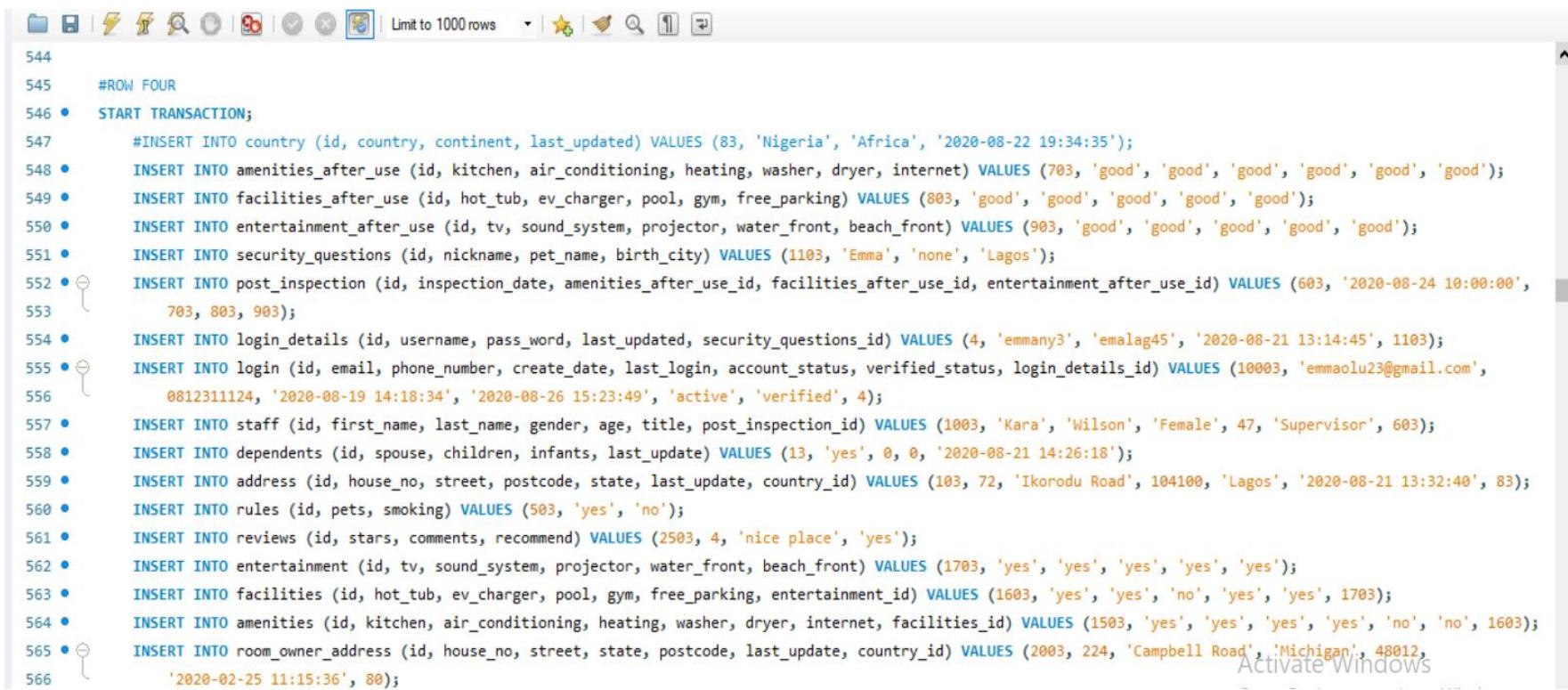


The screenshot shows a MySQL Workbench interface with a transaction log window. The log contains a series of UPDATE statements, each with a line number and a timestamp. The statements are all targeting the third row of various tables, indicated by the WHERE id = 3 condition in each query. The tables affected include amenities_after_use, facilities_after_use, entertainment_after_use, security_questions, post_inspection, login_details, login, customer, staff, dependents, address, rules, reviews, entertainment, facilities, amenities, room_address, room_owner_address, room_owner, room_address, room, payment, reservation, and reservation. The log also includes START TRANSACTION and COMMIT commands.

Line Number	Timestamp	Query
520	•	START TRANSACTION;
521	•	UPDATE amenities_after_use SET post_inspection_id = 602 WHERE id = 702;
522	•	UPDATE facilities_after_use SET post_inspection_id = 602 WHERE id = 802;
523	•	UPDATE entertainment_after_use SET post_inspection_id = 602 WHERE id = 902;
524	•	UPDATE security_questions SET login_details_id = 3 WHERE id = 1102;
525	•	UPDATE post_inspection SET staff_id = 1002 WHERE id = 602;
526	•	UPDATE login_details SET login_id = 10002 WHERE id = 3;
527	•	UPDATE login SET customer_id = 3 WHERE id = 10002;
528	•	UPDATE staff SET customer_id = 3 WHERE id = 1002;
529	•	UPDATE dependents SET customer_id = 3 WHERE id = 12;
530	•	UPDATE address SET customer_id = 3 WHERE id = 102;
531	•	UPDATE rules SET room_id = 152 WHERE id = 502;
532	•	UPDATE reviews SET payment_id = 252 WHERE id = 2502;
533	•	UPDATE entertainment SET facilities_id = 1602 WHERE id = 1702;
534	•	UPDATE facilities SET amenities_id = 1502 WHERE id = 1602;
535	•	UPDATE amenities SET room_address_id = 402 WHERE id = 1502;
536	•	UPDATE room_owner_address SET room_owner_id = 1302 WHERE id = 2002;
537	•	UPDATE room_owner SET room_address_id = 402 WHERE id = 1302;
538	•	UPDATE room_address SET room_id = 152 WHERE id = 402;
539	•	UPDATE room SET reservation_id = 3 WHERE id = 152;
540	•	UPDATE payment SET reservation_id = 3 WHERE id = 252;
541	•	UPDATE reservation SET customer_id = 3 WHERE id = 3;
542	•	COMMIT;
543		

Activate Windows
Go to Settings to activate Windows.

INSERTING INTO FOURTH ROW OF ALL TABLES



The screenshot shows a MySQL Workbench interface with a query editor containing a large block of SQL code. The code consists of 566 numbered lines, starting with a comment '#ROW FOUR' and followed by multiple INSERT statements into various tables. The tables include country, amenities_after_use, facilities_after_use, entertainment_after_use, security_questions, post_inspection, login_details, login, staff, dependents, address, rules, reviews, entertainment, facilities, amenities, and room_owner_address. The values inserted into these tables represent the fourth row of data for each respective table. The interface includes a toolbar at the top with various icons for database management.

```
544
545  #ROW FOUR
546 • START TRANSACTION;
547      INSERT INTO country (id, country, continent, last_updated) VALUES (83, 'Nigeria', 'Africa', '2020-08-22 19:34:35');
548 •      INSERT INTO amenities_after_use (id, kitchen, air_conditioning, heating, washer, dryer, internet) VALUES (703, 'good', 'good', 'good', 'good', 'good', 'good');
549 •      INSERT INTO facilities_after_use (id, hot_tub, ev_charger, pool, gym, free_parking) VALUES (803, 'good', 'good', 'good', 'good', 'good');
550 •      INSERT INTO entertainment_after_use (id, tv, sound_system, projector, water_front, beach_front) VALUES (903, 'good', 'good', 'good', 'good', 'good');
551 •      INSERT INTO security_questions (id, nickname, pet_name, birth_city) VALUES (1103, 'Emma', 'none', 'Lagos');
552 •      INSERT INTO post_inspection (id, inspection_date, amenities_after_use_id, facilities_after_use_id, entertainment_after_use_id) VALUES (603, '2020-08-24 10:00:00',
553          703, 803, 903);
554 •      INSERT INTO login_details (id, username, pass_word, last_updated, security_questions_id) VALUES (4, 'emmany3', 'emalag45', '2020-08-21 13:14:45', 1103);
555 •      INSERT INTO login (id, email, phone_number, create_date, last_login, account_status, verified_status, login_details_id) VALUES (10003, 'emmaolu23@gmail.com',
556          0812311124, '2020-08-19 14:18:34', '2020-08-26 15:23:49', 'active', 'verified', 4);
557 •      INSERT INTO staff (id, first_name, last_name, gender, age, title, post_inspection_id) VALUES (1003, 'Kara', 'Wilson', 'Female', 47, 'Supervisor', 603);
558 •      INSERT INTO dependents (id, spouse, children, infants, last_update) VALUES (13, 'yes', 0, 0, '2020-08-21 14:26:18');
559 •      INSERT INTO address (id, house_no, street, postcode, state, last_update, country_id) VALUES (103, 72, 'Ikorodu Road', 104100, 'Lagos', '2020-08-21 13:32:40', 83);
560 •      INSERT INTO rules (id, pets, smoking) VALUES (503, 'yes', 'no');
561 •      INSERT INTO reviews (id, stars, comments, recommend) VALUES (2503, 4, 'nice place', 'yes');
562 •      INSERT INTO entertainment (id, tv, sound_system, projector, water_front, beach_front) VALUES (1703, 'yes', 'yes', 'yes', 'yes', 'yes');
563 •      INSERT INTO facilities (id, hot_tub, ev_charger, pool, gym, free_parking, entertainment_id) VALUES (1603, 'yes', 'yes', 'no', 'yes', 'yes', 1703);
564 •      INSERT INTO amenities (id, kitchen, air_conditioning, heating, washer, dryer, internet, facilities_id) VALUES (1503, 'yes', 'yes', 'yes', 'yes', 'no', 'no', 1603);
565 •      INSERT INTO room_owner_address (id, house_no, street, state, postcode, last_update, country_id) VALUES (2003, 224, 'Campbell Road', 'Michigan', 48012,
566          '2020-02-25 11:15:36', 80);
```

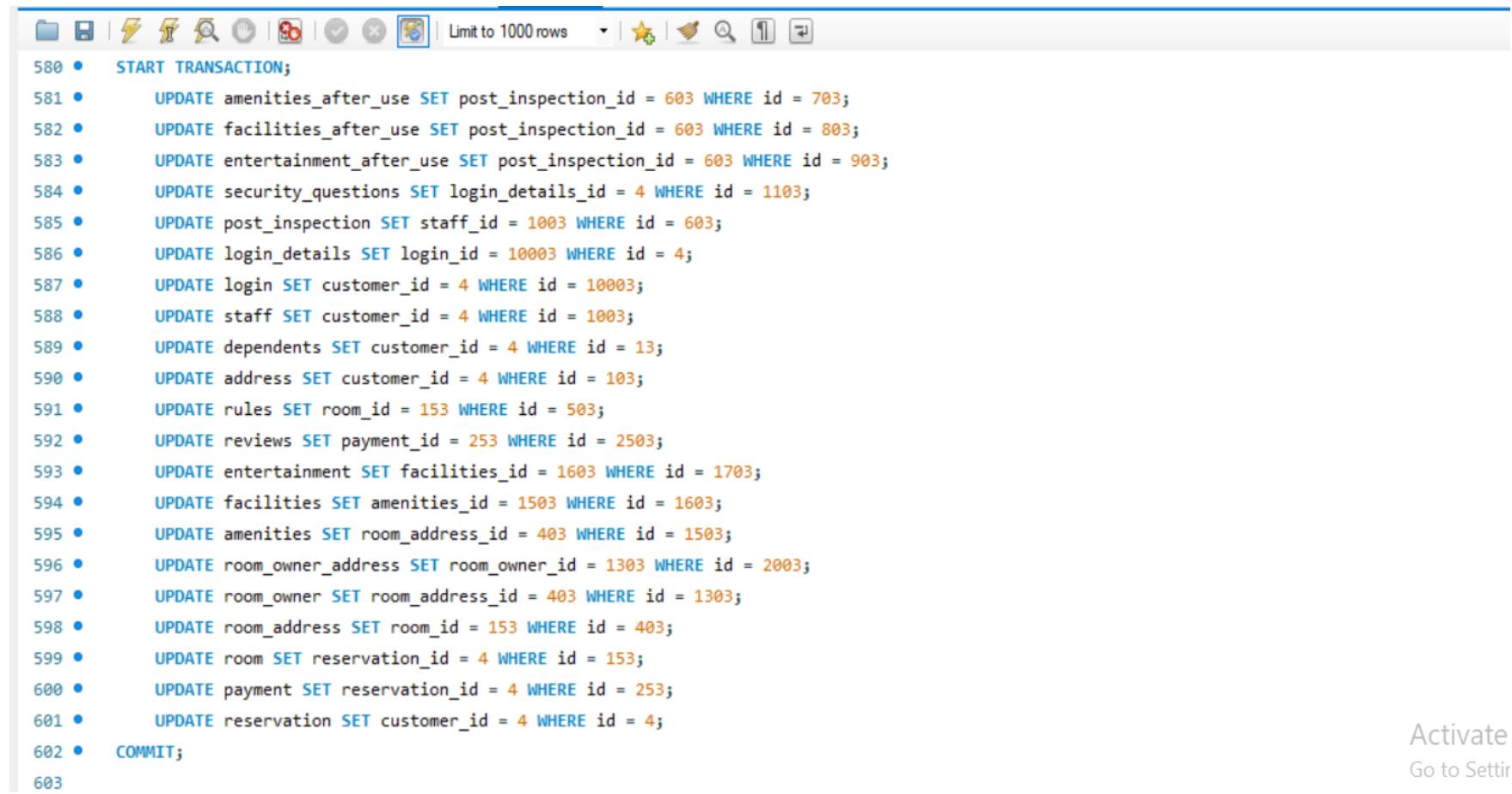
Activate Windows

END OF INSERTION INTO FORTH ROW OF ALL TABLES

This process of inserting and updating was continued for the remaining sixteen(16) entities in each tables.

```
567 •     INSERT INTO room_owner (id, first_name, last_name, gender, age, room_owner_address_id) VALUES (1303, 'Chris', 'Daniels', 'Male', 52, 2003);
568 •     INSERT INTO room_address (id, house_no, street, postcode, state, country_id, room_owner_id, amenities_id) VALUES (403, 234, 'Campbell Road', 48067, 'Michigan',
569           80, 1303, 1503);
570 •     INSERT INTO room (id, home_type, total_bedrooms, total_bathrooms, published_date, room_address_id, rules_id) VALUES (153, 'entire place', 2, 2, '2020-01-26 12:13:20',
571           403, 503);
572 •     INSERT INTO payment (id, payment_mode, price, discount, total_price, payment_status, reviews_id) VALUES (253, 'card', 700, 00, 700, 'paid', 2503);
573 •     INSERT INTO reservation (id, reservation_type, booking_date, start_date, end_date, last_update, room_id, payment_id) VALUES (4, 'instant book', '2020-08-10 13:29:39',
574           '2020-08-16 20:00:00', '2020-08-23 19:00:00', '2021-08-11 17:15:21', 153, 253);
575
576 •     INSERT INTO customer (id, first_name, last_name, gender, age, job_title, job_industry, dependents_id, address_id, staff_id, login_id, reservation_id)
577           VALUES (4, 'Oluwole', 'Emmanuel', 'Male', 26, 'Telesales Agent', 'Financial Services', 13, 103, 1003, 10003, 4);
578 •     COMMIT;
```

UPDATING FORTH ROW OF ALL TABLES



The screenshot shows a MySQL Workbench interface with a transaction log window. The log contains 603 numbered statements, all of which are UPDATE commands. These statements are distributed across various tables such as amenities_after_use, facilities_after_use, entertainment_after_use, security_questions, post_inspection, login_details, login, staff, dependents, address, rules, reviews, entertainment, facilities, amenities, room_address, room_owner_address, room_owner, room_address, room, payment, and reservation. Each statement updates a specific row where the id matches a value from another table like post_inspection_id or customer_id. The log is preceded by a START TRANSACTION; statement and followed by a COMMIT; statement. A 'Limit to 1000 rows' dropdown is visible at the top of the log window.

```
580 • START TRANSACTION;
581 •     UPDATE amenities_after_use SET post_inspection_id = 603 WHERE id = 703;
582 •     UPDATE facilities_after_use SET post_inspection_id = 603 WHERE id = 803;
583 •     UPDATE entertainment_after_use SET post_inspection_id = 603 WHERE id = 903;
584 •     UPDATE security_questions SET login_details_id = 4 WHERE id = 1103;
585 •     UPDATE post_inspection SET staff_id = 1003 WHERE id = 603;
586 •     UPDATE login_details SET login_id = 10003 WHERE id = 4;
587 •     UPDATE login SET customer_id = 4 WHERE id = 10003;
588 •     UPDATE staff SET customer_id = 4 WHERE id = 1003;
589 •     UPDATE dependents SET customer_id = 4 WHERE id = 13;
590 •     UPDATE address SET customer_id = 4 WHERE id = 103;
591 •     UPDATE rules SET room_id = 153 WHERE id = 503;
592 •     UPDATE reviews SET payment_id = 253 WHERE id = 2503;
593 •     UPDATE entertainment SET facilities_id = 1603 WHERE id = 1703;
594 •     UPDATE facilities SET amenities_id = 1503 WHERE id = 1603;
595 •     UPDATE amenities SET room_address_id = 403 WHERE id = 1503;
596 •     UPDATE room_owner_address SET room_owner_id = 1303 WHERE id = 2003;
597 •     UPDATE room_owner SET room_address_id = 403 WHERE id = 1303;
598 •     UPDATE room_address SET room_id = 153 WHERE id = 403;
599 •     UPDATE room SET reservation_id = 4 WHERE id = 153;
600 •     UPDATE payment SET reservation_id = 4 WHERE id = 253;
601 •     UPDATE reservation SET customer_id = 4 WHERE id = 4;
602 •     COMMIT;
603
```

Activate
Go to Settir

TEST CASE 1

Get name, spouse status, address and account creation date of customers.

```

26
27 •  SELECT customer.first_name AS 'first name', customer.last_name AS 'last name',
28      dependents.spouse AS 'Spouse Status', dependents.children AS 'children',
29      address.street AS 'Street', address.state AS 'State',
30      login.email AS 'email', login.create_date AS 'creation date',
31      country.country AS 'Country Name'
32  FROM customer
33  JOIN dependents ON customer.dependents_id = dependents.id
34  JOIN address ON customer.address_id = address.id
35  JOIN login ON customer.login_id = login.id
36  JOIN country ON address.country_id = country.id
37
38

```

The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons for database management. Below the toolbar is a query editor window containing the provided SQL code. The code is a SELECT statement that joins five tables: customer, dependents, address, login, and country. It retrieves columns such as first name, last name, spouse status, children, street, state, email, creation date, and country name. The result grid below the code shows 10 rows of data corresponding to the query results. The data includes names like Stefan, Mieke, Chickie, Brister, Nancy, Docket, Oluwole, Emmanuel, Akande, Festus, Olayoke, David, Thomas, Forsteel, Maurita, Lyste, Heath, and Faraday, along with their respective details like address, email, and creation dates. A message at the bottom right of the grid says "Activate Windows Go to Settings to activate Win".

first name	last name	Spouse Status	children	Street	State	email	creation date	Country Name
Stefan	Mieke	yes	0	broad	New York	stefarmieke@hotmail.com	2020-06-20 11:15:37	USA
Chickie	Brister	yes	1	Ajax	Ontario	stefarmieke@hotmail.com	2020-06-25 14:16:37	Canada
Nancy	Docket	yes	1	Ballarat	Victoria	nancydock@gmail.com	2020-06-25 14:16:37	Australia
Oluwole	Emmanuel	yes	0	Ikorodu Road	Lagos	emmaolu23@gmail.com	2020-08-19 14:18:34	Nigeria
Akande	Festus	no	0	Lekki Road	Lagos	akanke5676@gmail.com	2020-08-29 14:18:34	Nigeria
Olayoke	David	yes	0	Oluoyele	Ibadan	davidholay@hotmail.com	2020-09-12 14:18:33	Nigeria
Thomas	Forsteel	yes	0	Avenue	New York	thomfhoral@hotmail.com	2020-09-15 14:11:35	USA
Maurita	Lyste	yes	1	Choiny	Lublin	mauricelynn@gmail.com	2020-09-29 14:33:20	Poland
Heath	Faraday	yes	1	Hauptwache	Frankfurt	farrell345@gmail.com	2020-10-05 14:20:33	Germany

Result 1 ×

Activate Windows
Go to Settings to activate Win

TEST CASE 2

Get name of customers, their type of reservation, who paid a total price of over 700 dollars and gave a rated their service as high.

```

27 #Get name of customers, type of room and reservation, total price and reviews of customers who paid more than 400 dollars and with highest rating
28 SELECT customer.first_name AS 'First Name', customer.last_name AS 'Last Name',
29     reservation.reservation_type AS 'Type of Reservation',
30     room.home_type AS 'Room Type',
31     payment.total_price AS 'Total Price',
32     reviews.stars AS 'stars', reviews.comments AS 'comments'
33 FROM customer
34 JOIN reservation ON reservation.customer_id = customer.id
35 JOIN room ON room.reservation_id = reservation.id
36 JOIN payment ON payment.id = reservation.payment_id
37 JOIN reviews ON reviews.payment_id = payment.id
38 WHERE payment.total_price >= 700 AND reviews.stars >= 4

```

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

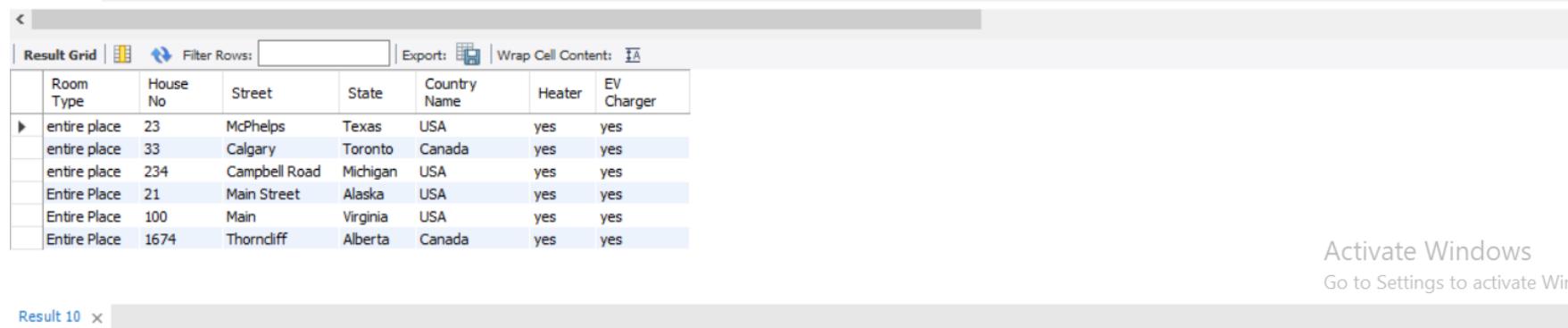
First Name	Last Name	Type of Reservation	Room Type	Total Price	stars	comments
Stefan	Mieke	instant book	entire place	740	4	great experience
Oluwole	Emmanuel	instant book	entire place	700	4	nice place
Olayoke	David	request book	Hotel room	850	5	great and relaxing experience
Heath	Faraday	instant book	Entire Place	999	4	Nice
Gordan	Gower	request book	Entire Place	780	4	Amazing Experience
Stefan	Dunnan	instant book	Entire Place	730	5	Exquisite place
Shane	Michael	instant book	Entire Place	820	4	Amazing Place

Activate Windows
Go to Settings to activate Win

TEST CASE 3

Get type of and address of rooms in USA and Canada that are equipped with Heater and EV Charger.

```
--  
27      #Get type and address of rooms in USA and Canada that is equipped with EV charger and heater  
28 •  SELECT room.home_type AS 'Room Type',  
29      room_address.house_no AS 'House No', room_address.street AS 'Street', room_address.state AS 'State',  
30      country.country AS 'Country Name',  
31      amenities.heating AS 'Heater',  
32      facilities.ev_charger AS 'EV Charger'  
33      FROM room  
34      JOIN room_address ON room_address.room_id = room.id  
35      JOIN country ON country.id = room_address.country_id  
36      JOIN amenities ON amenities.room_address_id = room_address.id  
37      JOIN facilities ON facilities.amenities_id = amenities.id  
38      WHERE country IN ('USA', 'Canada') AND amenities.heating = 'yes' AND facilities.ev_charger = 'yes'
```



The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid', 'Filter Rows:', 'Export:', and 'Wrap Cell Content:'. The grid has columns: Room Type, House No, Street, State, Country Name, Heater, and EV Charger. The data shows various room types across different countries and states, all equipped with both a Heater and an EV Charger.

Room Type	House No	Street	State	Country Name	Heater	EV Charger
entire place	23	McPhelps	Texas	USA	yes	yes
entire place	33	Calgary	Toronto	Canada	yes	yes
entire place	234	Campbell Road	Michigan	USA	yes	yes
Entire Place	21	Main Street	Alaska	USA	yes	yes
Entire Place	100	Main	Virginia	USA	yes	yes
Entire Place	1674	Thorndiff	Alberta	Canada	yes	yes

Activate Windows
Go to Settings to activate Win

Result 10 ×