

Abstract on Finalization Phase of Development of a Data Mart using MySQL

Structuring of stored information or data is an essential part of how companies, corporations, organizations etc. work in today's connected world, important information can be inferred from stored data. Various tools have been developed for analysis of stored information in order to make important business and strategic decisions such as Microsoft Power BI. Databases are widely used method of managing stored structured data. This project is about development of a database which will be used to store and manage structured data of rental services such as Airbnb, the development of this data mart project has greatly improved my understanding of databases using MySQL. My understanding of databases by working on this project using MySQL includes but not limited to: -

- ER modelling of database systems
- Database Normalization
- Database creation
- Database population with data.

The development of this project was carried out in the following stages: -

Conception phase: - In this phase, I researched about online rental organization by visiting Airbnb online website to get information about reservation types such as request and instant book. Room types such as entire place, private rooms, shared rooms and Hotel rooms, Locations, Facilities, Rules and so on.

After getting information about rental organizations, I proceeded by designing an Entity Relation (ER) diagram using UML class diagram using an online resource, this diagram displayed tables, columns and relationships between them. This diagram would form the basis on which the working database will be built upon in development phase. The UML diagram contains the following tables: -

- Customer, address, country and dependents tables: - these tables are used to store information about customers names, age, address, dependents and so on.
- Login, login details and security questions tables: - these tables are used to store information about customer login details such as username, email address, password, login history, security questions and so on.
- Reservation, payment and review tables: - these tables are used to store information such as customer's type of reservation, type of apartment, mode of payment, payment status, reviews and so on.
- Room, room address, facilities, amenities, entertainment and rules tables:- these tables are used to store information about type of rooms, total bedrooms and bathrooms, location of room, room rules, facilities and amenities that the room contains such as TV's, sound systems, pools, parking space, beach front and so on.
- Room owner and room owner address tables: - these tables contain information about the owner of rented rooms such as name, sex, age, address and so on
- Staff, post inspection, entertainment after use, facilities after use and amenities after use tables: - these tables contain information about the rental organization staffs assigned to each customer to inspect their rental locations after they check out to take a record of facilities, amenities and entertainment instrument statuses.

Development phase: - In this phase, I started firstly by normalizing the database. Normalization optimizes the structure of a database and the normal form of a database is a measure of this structure. It provides information on how redundant data are stored in a database (Kneuper. pg. 65. 2021). This process led to some modifications from the previous UML class diagram designed from conception phase, these changes consist of: -

- Removal of city table
- Addition of amenities after use table which is used to record data of state of amenities after a customer has checked out
- Addition of facilities after use table which is used to record data of state of amenities after a customer has checked out
- Addition of entertainment after use table which is used to record data of state of amenities after a customer has checked out
- Renaming of attributes in security questions table from first, second and third to nickname, pet name and birth city respectively.
- Adding of more foreign keys in order to maintain referential integrity on deletion of records which means that after deletion, there must be a primary key for every foreign key.

In the process of formulating queries for creation of tables in the database, I discovered that tables with no or less foreign keys needs to be created first before tables with many foreign keys so as to avoid errors when executing queries for tables creation, still not all foreign keys could be added in create table statements which lead to executing of alter table queries in order to add remaining foreign keys after tables has been created.

After tables has been created in the database, insert statements were executed to insert data across tables, I inserted data across tables at once using transactions so as to maintain uniformity across tables, maintenance of referential integrity and in order to have the flexibility of using rollback statement to undo changes when error occurs in the process of inserting data across tables which made data insertion easier and faster, this method of inserting data into the database made error handling while executing queries easier and faster.

While inserting data into tables in the database, I realized that all foreign keys could not be inserted all at once while inserting data, this led to the execution of update statements after insertion of data into the database. I also executed update statements using transactions in order to have access to the flexibility of rollback when error occurs. For example, while executing update statements, I made mistake of repeating of foreign key values into tables, However, this attribute has been restricted by using "UNIQUE" statement which led to errors. A rollback was initiated after which I corrected these values and the query was successfully executed later.

As a submission requirement and in order to infer useful information from data stored in a structured manner, sometimes data need to be selected and joined across tables in a single query. To that effect, test cases were formed by formulating select statement that joins data across tables.

Previous phases of the Development of a Data Mart using MySQL

Conception Phase

In order for organizations to store important data regarding clients, staffs, and services in an organized manner, various means of storing data can be employed such as Comma Separated Values (CSV's), Excel Spreadsheets, databases etc. However, only one will be used for this project which is the database. Database is a great way of storing and organizing data because it stores the data in an efficient manner, access management and functionality to databases is made possible through a database management system (DBMS). MySQL is a type of Relational database management which will be used to store and organize data in a relational manner. The aim of this project is to create a sample database about customers and services of a rental company for holiday destination such as Airbnb.

I am planning to build a data mart that can be used to infer data about clients, browse information about existing reservation, closed reservation, reservation history, customers review and so on. I will start this project by downloading MySQL and MySQL workbench to create the database after completion of schema design through Entity Relationship (ER) diagram.

After surfing Airbnb website for important details about the rental system works, I designed an Entity Relationship model that will be used as a background for designing this database, Entity relationship diagrams are used to display data schemas in a database. ER diagrams contains keys, relationships, entities, and attributes, three popular methods used for ER diagrams are: -

- Chen notation,
- Martin notation, and
- UML class diagram

However, UML class diagram is chosen to model this database schema, it can be observed in the UML diagram below that there are relationships between twenty tables in the UML diagram below, these tables are used to store various information about the reservation system, staffs, customers, available destinations etc. In order to maintain referential integrity, I will make sure that there is an actual primary key for every foreign key when using the designed UML diagram to create the database.

In the UML diagram below: - The **Customer** table will serve as the main table in which other tables relate to, this table will contain data about background information about customers such as name, age, gender, job title, foreign keys referencing other tables etc.

The **Dependent** table will contain data about dependents of the customer which will be included in the reservation such as spouse, children etc.

The **Staff** table will contain information about which staff is in charge of the comfortability of the customer during his reservation such as staff name, rank and foreign key to the post inspection table.

The **Post Inspection** table will contain information about state of facilities and amenities after the guest has checked out of their reservation which will be carried out by the staff.

The **Address** table will contain information about the customers home address, postcode, street etc.

The **Staff, Post Inspection, Entertainment after use, Amenities after use, Facilities after use** tables will contain information about staffs, the customers they were assigned to and data about details about their inspection of facilities, amenities after customers assigned to them has checked out.

The **Country** table will contain information about the country in which the customers reside in, it will also contain information about the country in which the room owners reside in. This table has two relationships attached to it.

The **Log-in** table will contain data about the customers login information such as email, phone number, account status and foreign key to the login details table.

The **Login Details** table will contain information about username, password, last updated date and foreign key to the security details table.

The **Security Questions** table will contain information about customers security questions in case of a situation in which a customer loses their login details.

The **Reservation** table will contain information about available locations such as, reservation type, booking date, start date etc. It will also foreign keys to maintain relationships with room and payment tables.

The **Room** table will contain information about various available room types and other information such as number of bedrooms, number of bathrooms, publish date etc. This table will also contain foreign keys relating to other tables namely room address and rules.

The **Room Address** table will contain information about the address the rooms are located at with street, state and country names. This table will also contain foreign keys relating to room owner and room amenities tables.

The **Room Owner** table will contain information about the owner of the published reservation space such as name, address etc.

The **Rules** table will contain information about dos and don'ts of each available room such as permission of pets and smoking on the property.

The **Payment** table will contain information about the room prices, discounts (if applicable), payment mode, payment status etc. This table will also contain a foreign key related to the reviews table.

The **Amenities** table will contain information about amenities on each property such as kitchen, air conditioning, heater, internet and so on. This table will also contain a foreign key related to the facilities table.

The **Facilities** table will contain information about facilities available on each available space such as hot tub, pool, gym, free parking etc. It will also contain a foreign key to the entertainment table.

The **Entertainment** table will contain information about the availability of entertainment equipment's such as Televisions, Sound systems, beach front, water front's etc.

In the process of normalizing the database during creation using MySQL, more tables may be needed to be included so as to make the database more efficient when managing and querying. The primary keys and foreign keys have deeper background colors to differentiate between other attributes in the tables.

Below is the UML diagram of the Rentals Database



Development Phase

**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.

2

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	8.0.27	2.3M	Download
(mysql-installer-web-community-8.0.27.1.msi)			
	MD5: 44b7f3e4c1bdcc641621cfaa31ea18f4 Signature		
Windows (x86, 32-bit), MSI Installer	8.0.27	470.2M	Download
(mysql-installer-community-8.0.27.1.msi)			
	MD5: 9b7af5c91139659b18b84b1ca357d88f Signature		


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

3

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)
);
```

4

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
```

5

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 );

```

6

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

7

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)
);

```

8

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

9

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 );
---
```

10

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
);

```

Act
Go 1

11

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
);

```

12

CREATION OF ROOM ADDRESS TABLE

```

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 );

```

13

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
);
```

14

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
```

15

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

```

16

ALTERING TABLES

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

```

#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

17

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.

18

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
```

19

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;
370

```

20

INSERTING INTO FIRST ROWS OF ALL TABLES

```

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 18:00:00', 7, 12435436453, '2020-06-20 11:15:37', '2020-07-14 14:23:47', 'active', 'verified', 1);
382 * INSERT INTO login_details (id, username, pass_word, last_updated, security_questions_id) VALUES (1, 'stefan1276', 'stemi72', '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, 'stefanieke@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

```

21

END OF INSERTION INTO FIRST ROW OF ALL TABLES

```

398 * INSERT INTO payment (id, payment_mode, price, discount, total_price, payment_status, reviews_id) VALUES (250, 'card', 750, 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;

```

22

UPDATING FIRST ROW IN ALL TABLES

```

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;
429

```

Activate Windows
Go to Settings to activate Windows.

23

INSERTING INTO SECOND ROW OF ALL TABLES

```

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', 701, 801, 901);
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, 'stefannieke@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 1301);
452 * INSERT INTO room_owner (id, first_name, last_name, gender, age, room_owner_address_id) VALUES (1301, 'Samuel', 'Jackson', 'Male', 54, 2001);
453 * 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, 1501);
454 * 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',
455 401, 501);

```

24

END OF INSERTION INTO SECOND ROW OF ALL TABLES

```

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, 1501);
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', 'Briston', 'Male', 64, 'General Manager', 'Manufacturing', 11, 101, 1001, 10001, 2);
461 * COMMIT;

```

25

UPDATING SECOND ROW OF ALL TABLES

```

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;
486

```

26

INSERTING INTO THIRD ROW OF ALL TABLES

```

488 • FROM 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, 'naddock111', 'smith099', '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', '2020-08-25 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 (182, 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', 3089, '2020-02-28 13:10:31');
507 • INSERT INTO room_owner (id, first_name, last_name, gender, age, room_owner_address_id) VALUES (1302, 'Handy', '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', 3082, '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);
510 •
511 •

```

27

END OF INSERTION INTO THIRD ROW OF ALL TABLES

```

509 * 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, 11
510 * 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'
511 * 402, 502);
512 * INSERT INTO payment (id, payment_mode, price, discount, total_price, payment_status, reviews_id) VALUES (252, 'check', 560, 20, 540, 'paid', 2502);
513 * 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'
514 * '2020-08-04 10:00:00', '2020-08-11 09:00:00', '2021-08-03 12:18:29', 152, 252);
515 *
516 * INSERT INTO customer (id, first_name, last_name, gender, age, job_title, job_industry, dependents_id, address_id, staff_id, login_id, reservation_id)
517 * VALUES (3, 'Nancy', 'Docket', 'Male', 56, 'Financial Analyst', 'Financial Services', 12, 102, 1002, 10002, 3);
518 * COMMIT;

```

28

UPDATING THIRD ROW OF ALL TABLES

```

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.

29

INSERTING INTO FOURTH ROW OF ALL TABLES

```

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 (603, '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, 603, 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 0012311124, '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);

```

30

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;

```

31

UPDATING FORTH ROW OF ALL TABLES

```

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 Settings

32

TEST CASE 1

Get name, spousestatus, 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

```

Result Grid

first name	last name	Spouse Status	children	Street	State	email	creation date	Country Name
Stefan	Melie	yes	0	broad	New York	stefanmelie@hotmail.com	2020-06-20 11:15:37	USA
Chickie	Briester	yes	1	Ajax	Ontario	stefanmelie@hotmail.com	2020-06-25 14:16:37	Canada
Nancy	Doodet	yes	1	Ballarar	Victoria	nancydoodet@gmail.com	2020-06-25 14:16:37	Australia
Olivirole	Emmanuel	yes	0	Ikorodu Road	Lagos	emmanuel23@gmail.com	2020-08-19 14:18:34	Nigeria
Akande	Festus	no	0	Lekki Road	Lagos	akande567@gmail.com	2020-08-29 14:18:34	Nigeria
Olayoke	David	yes	0	Olayole	Ibadan	davidholay@hotmail.com	2020-09-12 14:18:33	Nigeria
Thomas	Forsteel	yes	0	Avenue	New York	thomforsteel@hotmail.com	2020-09-15 14:11:35	USA
Maurilia	Lyste	yes	1	Choiny	Lublin	mauriliayl@gmail.com	2020-09-29 14:33:20	Poland
Heath	Paraday	yes	1	Hauptwache	Frankfurt	farrell345@gmail.com	2020-10-05 14:20:33	Germany

Activate Windows
Go to Settings to activate Win

33

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.room_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
```

First Name	Last Name	Type of Reservation	Room Type	Total Price	stars	comments
Stefan	Meke	instant book	entire place	740	4	great experience
Olusole	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	Gover	request book	Entire Place	780	4	Amazing Experience
Stefan	Dunman	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

34

TEST CASE 3

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

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

Room Type	House No	Street	State	Country Name	Heater	EV Charger
entire place	23	McPhels	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 x

35

References

Prof. Dr. Ralf Kneuper., (2021) DLBCSDMDS01: Database Management and Database Systems Coursebook. IUBH Internationale Hochschule GmbH