Distinction project report for Fundamentals of Data Management

Name: Cuong Nguyen Student ID: 102840305 Date prepared: 14/10/2020

*Abstract:

This report is about the design, creation and implementation of a database for a fictional hotel in MySQL. The raw data for this database was created outside and saved in CSV file, which could be imported using phpMyAdmin or commands in the workbench. The report also outlines a variety of use-cases for the database, such as searching bills group by price total, or searching data about feedbacks of customers toward a particular employee. This database stores information on nearly 50 occupancies of 25 different customers, in 28 rooms of the hotel.

*Overview of the Database

This database is intended for collect and store information about activities of Dumas Hotel.

- *Background Information and Main Use:*

Dumas Hotel is a new hotel (in fiction) having a plan to enlarge, so its owners needs methods to effectively manage its own business. To do that, they create a datastore to contain information about their rooms, occupancies, customers, employees, bills, and feedbacks from customers.

Each room has its own room number, room type, bed type, status , price each night and amenities. Each occupancy has occupancy number (for each time a room is used), employee ID, account number , number of days occupied and room number (linked with a room). For each feedbacks, there would have feedback number, occupancy number, employee's opinion and notes (It may or may not have that).

Every customer has its ID card number, account number, full name, phone number and emergency contacts (including contact name and phone number). For employees, they have

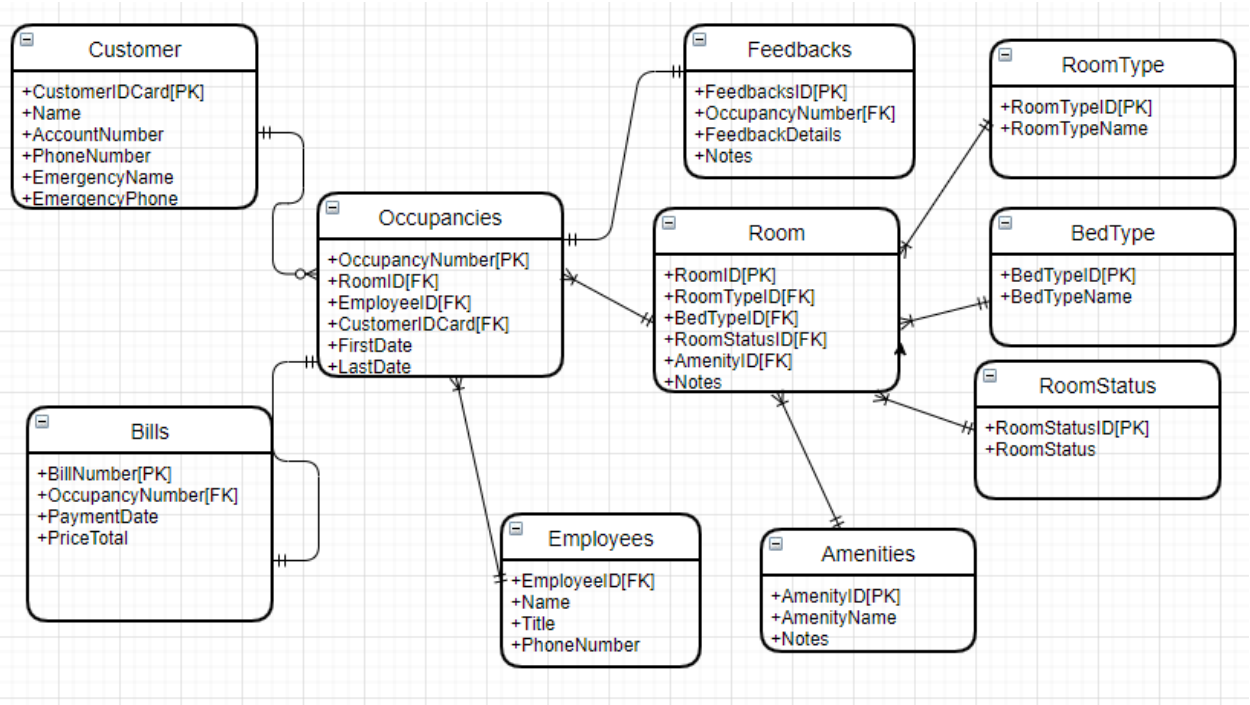
their employee ID, their full name, title and phone number. Finally on bills, that would show bill number, payment date, occupancy number and total price.

- *Entities for the Database:*

This database will include these entities: RoomType, BedType, RoomStatus, Amenities, Room, Occupancies, Customers, Feedbacks, Employees, Bills.

- *Entity Relationship Diagram:*

This is the entity relationship diagram for the database. Note: It has been normalized to the 3NF.



- *Attributes needed*:

+RoomType table:

Attribute	Data Type	Reason
RoomTypeID	TINYINT	There are only 4 room types cited, so higher number is unnecessary.
RoomType	CHAR(10)	These types are standard, suite, king and conference.

+BedType table:

Attribute	Data Type	Reason
BedTypeID	TINYINT	There are only 5 bed types cited(single, double, queen-sized, king-sized, sofa-bed), so higher number is unnecessary.
BedTypeName	CHAR(12)	These types are standard, suite, king and conference.

+RoomStatus table:

Attribute	Data Type	Reason
RoomStatusID	TINYINT	There are only 4 types of room status.
RoomStatus	VARCHAR(20)	These types are occupied, arranging, ready for guests and upgrading.

+Amenities table:

Attribute	Data Type	Reason
AmenityID	TINYINT	There are only 4 types of amenities.
AmenityName	CHAR(12)	These types are standard, intermediate, advanced and pro (none of which is over 12 characters).
Notes	VARCHAR(120)	The description for each type of amenity.

+Room table:

Attribute	Data Type	Reason
RoomID	INT(4) UNSIGNED	For the foresight of the expansion of the hotel.
RoomTypeID	TINYINT	Foreign key of the RoomType table.

BedTypeID	TINYINT	Foreign key of the BedType table.
RoomStatusID	TINYINT	Foreign key of the RoomStatus table.
AmenityID	TINYINT	Foreign key of the Amenities table.
Notes	VARCHAR(100)	The description for each room – could be null.

+Customer table:

Attribute	Data Type	Reason
CustomerIDCard	INT(8) UNSIGNED	Assuming that every customer has an ID card with 8 digits.
Name	VARCHAR(60)	Name length could vary.
AccountNumber	CHAR(16)	Account number (or in reality, credit card number) only has 15 or 16 digits; (see https://www.experian.com/blogs/ask-experian/how-many-numbers-are-on-a-credit-card/)
PhoneNumber	CHAR(12)	For handling data of phone numbers.
EmergencyName	VARCHAR(50)	Name length in this case could also vary.
EmergencyPhone	CHAR(12)	Same reason with PhoneNumber.

+Employee table:

Attribute	Data Type	Reason
EmployeeID	INT(6) UNSIGNED	Assuming that the ID for employees in the hotel would have 6 digits.
Name	VARCHAR(60)	Name length could vary.
Title	CHAR(4)	Each employee could have titles like Mr, Miss or Ms.
PhoneNumber	INT(10)	For handling data of phone numbers (phone number this case has 10 digits).

+Occupancies table:

Attribute	Data Type	Reason
OccupancyNumber	INT(8) UNSIGNED	Creating more spaces to storing more occupancies in the database in the future
RoomID	INT(4) UNSIGNED	Foreign key from Room table.
EmployeeID	INT(6) UNSIGNED	Foreign key from Employee table.
CustomerIDCard	INT(8) UNSIGNED	Foreign key from Customer table.
FirstDate	DATE	Storing date in MySQL needs this special data type.
LastDate	DATE	The reason is the same as FirstDate attribute.

+Feedbacks table:

Attribute	Data Type	Reason
FeedbacksID	INT(6) UNSIGNED AUTO_INCREMENT	We could have more places to add feedbacks, and adding feedback ID for each feedbacks is unnecessary.
OccupancyNumber	INT(8) UNSIGNED	Foreign keys from Occupancies.
FeedbackDetails	VARCHAR(200)	We could have more spaces to add feedbacks from each customers – which are varied.
Notes	VARCHAR(150)	Noting about feedbacks. This could be varied in length and could be null.

+Bills table:

Attribute	Data Type	Reason
BillNumber	INT(8) UNSIGNED AUTO_INCREMENT	Each occupancy would have its own bill, so that helps to

		synchronize between these two entities.
OccupancyNumber	INT(8) UNSIGNED	Foreign keys from Occupancies.
PaymentDate	DATE	Using DATE to handle date in MySQL.
PriceTotal	INT(8)	Using this to make space for storing price, help data to not being truncated.

- *Typical use-cases of the database:*

This database could be used for a variety of purposes:

+ For searching the number of rooms based on room type, bed type, room status and amenities;

+ For searching data of all rooms, or a particular occupancy;

+ For searching data about a particular employee and customers' opinion about occupancies involving him or her;

+For sorting bills by price total;

+ For providing information about feedbacks of customers, with additional data about customer name, first and last dates;

+For altering information about each of the database entities.

All of them will be discuss in the section about queries used for this database.

*Commands for creating data storage:

- *Creating tables:*

+ For creating RoomType table:

```
MariaDB [DumasHotel]> CREATE TABLE RoomType(
    -> RoomTypeID TINYINT NOT NULL,
    -> RoomTypeName CHAR(10) NOT NULL,
    -> PRIMARY KEY (RoomTypeID)
    -> );
Query OK, 0 rows affected (1.128 sec)
```

+For creating BedType table:

```
MariaDB [DumasHotel]> CREATE TABLE BedType(
    -> BedTypeID TINYINT NOT NULL,
    -> BedTypeName CHAR(12) NOT NULL,
    -> PRIMARY KEY (BedTypeID)
    -> );
Query OK, 0 rows affected (0.042 sec)
```

+For creating RoomStatus table:

```
MariaDB [DumasHotel]> CREATE TABLE RoomStatus(
    -> RoomStatusID TINYINT NOT NULL,
    -> RoomStatus CHAR(12) NOT NULL,
    -> PRIMARY KEY (RoomStatusID)
    -> );
Query OK, 0 rows affected (0.044 sec)
```

```
MariaDB [DumasHotel]> ALTER TABLE RoomStatus MODIFY RoomStatus VARCHAR(20) NOT NULL;
Query OK, 3 rows affected (0.138 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

(I fixed RoomStatus column so that the name could be more suitable to the room status that Dumas Hotel have).

+For creating Amenities table:

```
MariaDB [DumasHotel]> CREATE TABLE Amenities(
    -> AmenityID TINYINT NOT NULL,
    -> AmenityName CHAR(15) NOT NULL,
    -> Notes VARCHAR(80),
    -> PRIMARY KEY (AmenityID)
    -> );
Query OK, 0 rows affected (0.063 sec)
```

```
MariaDB [DumasHotel]> ALTER TABLE Amenities MODIFY Notes VARCHAR(120);
Query OK, 0 rows affected (0.011 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

(I fixed the characteristic of Notes column to VARCHAR(120) so that it would have more space to describe each type of amenities).

+For creating Room table:

```
MariaDB [DumasHotel]> CREATE TABLE Room(
    -> RoomID INT(4) UNSIGNED NOT NULL,
    -> RoomTypeID TINYINT NOT NULL,
    -> BedTypeID TINYINT NOT NULL,
    -> RoomStatusID TINYINT NOT NULL,
    -> AmenityID TINYINT NOT NULL,
    -> Notes VARCHAR(100),
    -> PRIMARY KEY (RoomID),
    -> FOREIGN KEY (RoomTypeID) REFERENCES RoomType(RoomTypeID),
    -> FOREIGN KEY (BedTypeID) REFERENCES BedType(BedTypeID),
    -> FOREIGN KEY (RoomStatusID) REFERENCES RoomStatus(RoomStatusID),
    -> FOREIGN KEY (AmenityID) REFERENCES Amenities(AmenityID)
    -> );
Query OK, 0 rows affected (0.096 sec)
```

+For creating Customer table:

```
MariaDB [DumasHotel]> CREATE TABLE Customer(
    -> CustomerIDCard INT(8) UNSIGNED NOT NULL,
    -> Name VARCHAR(60) NOT NULL,
    -> AccountNumber INT(16) UNSIGNED NOT NULL,
    -> PhoneNumber INT(10) UNSIGNED NOT NULL,
    -> EmergencyName VARCHAR(50) NOT NULL,
    -> EmergencyPhone INT(10) UNSIGNED NOT NULL,
    -> PRIMARY KEY (CustomerIDCard)
    -> );
Query OK, 0 rows affected (0.129 sec)
```

```
MariaDB [DumasHotel]> ALTER TABLE Customer MODIFY AccountNumber CHAR(16) NOT NULL;
Query OK, 25 rows affected (0.372 sec)
Records: 25  Duplicates: 0  Warnings: 0

MariaDB [DumasHotel]> ALTER TABLE Customer MODIFY PhoneNumber CHAR(12) NOT NULL;
Query OK, 25 rows affected (0.122 sec)
Records: 25  Duplicates: 0  Warnings: 0

MariaDB [DumasHotel]> ALTER TABLE Customer MODIFY EmergencyPhone CHAR(12) NOT NULL;
Query OK, 25 rows affected (0.050 sec)
Records: 25  Duplicates: 0  Warnings: 0
```

(I have to modify AccountNumber , PhoneNumber and EmergencyPhone columns since INT unsigned could not handle a number that is over 4294967295)

+For creating Employee table:

```
MariaDB [DumasHotel]> CREATE TABLE Employee(
    -> EmployeeID INT(6) UNSIGNED NOT NULL,
    -> Name VARCHAR(60) NOT NULL,
    -> Title CHAR(4),
    -> PhoneNumber INT(10) UNSIGNED NOT NULL,
    -> PRIMARY KEY (EmployeeID)
    -> );
Query OK, 0 rows affected (0.044 sec)
```

+For creating Occupancies table:

```
MariaDB [DumasHotel]> CREATE TABLE Occupancies(
    -> OccupancyNumber INT(8) UNSIGNED NOT NULL,
    -> RoomID INT(4) UNSIGNED NOT NULL,
    -> EmployeeID INT(6) UNSIGNED NOT NULL,
    -> CustomerIDCard INT(8) UNSIGNED NOT NULL,
    -> FirstDate DATE NOT NULL,
    -> LastDate DATE NOT NULL,
    -> PRIMARY KEY (OccupancyNumber),
    -> FOREIGN KEY (RoomID) REFERENCES Room(RoomID),
    -> FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
    -> FOREIGN KEY (CustomerIDCard) REFERENCES Customer(CustomerIDCard)
    -> );
Query OK, 0 rows affected (0.051 sec)
```

+For creating Feedbacks table:

```
MariaDB [DumasHotel]> CREATE TABLE Feedbacks(
    -> FeedbacksID INT(6) UNSIGNED AUTO_INCREMENT NOT NULL,
    -> OccupancyNumber INT(8) UNSIGNED,
    -> FeedbackDetails VARCHAR(200) NOT NULL,
    -> Notes VARCHAR(150),
    -> PRIMARY KEY (FeedbacksID),
    -> FOREIGN KEY (OccupancyNumber) REFERENCES Occupancies(OccupancyNumber)
    -> );
Query OK, 0 rows affected (0.104 sec)
```

```
MariaDB [DumasHotel]> ALTER TABLE Feedbacks MODIFY OccupancyNumber INT(8) UNSIGNED NOT NULL;
Query OK, 0 rows affected (0.653 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

(I have to modify OccupancyNumber to ensure consistency with Occupancies table, since this is foreign key)

+For creating Bills table:

```
MariaDB [DumasHotel]> CREATE TABLE Bills(
    -> BillNumber INT(10) UNSIGNED AUTO_INCREMENT NOT NULL,
    -> OccupancyNumber INT(8) UNSIGNED NOT NULL,
    -> PaymentDate DATE NOT NULL,
    -> PriceTotal INT(6) UNSIGNED NOT NULL,
    -> PRIMARY KEY (BillNumber),
    -> FOREIGN KEY (OccupancyNumber) REFERENCES Occupancies(OccupancyNumber)
    -> );
Query OK, 0 rows affected (0.046 sec)
```

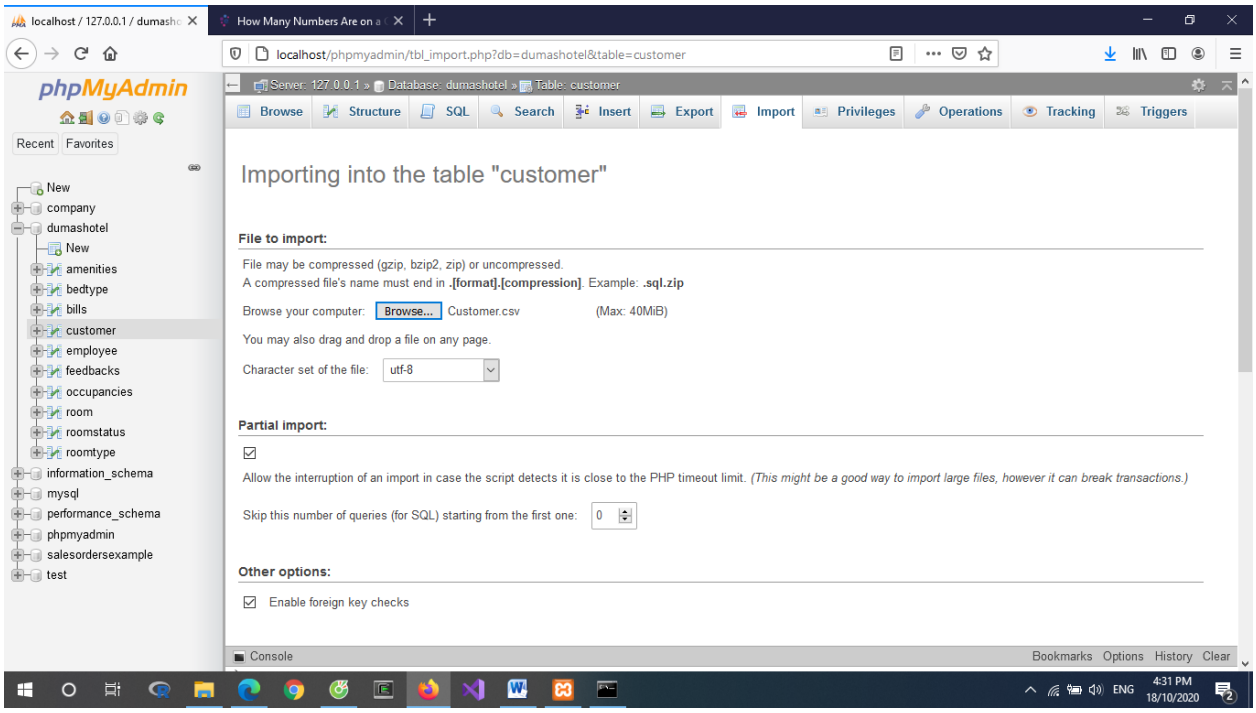
```
MariaDB [DumasHotel]> ALTER TABLE Bills MODIFY PriceTotal INT(8) NOT NULL;
Query OK, 0 rows affected (0.023 sec)
```

- *Adding data into tables:*

There are two ways that I use for adding data into the database.

I created CSV files for each of the tables that I want to create (images of CSV files could be looked at Apprendix), then I used follwing ways:

+ Adding CSV files for each table into the database by using phpMyAdmin:



+Using commands for adding CSV file into Occupancies and Bills tables (credit: https://www.mysqltutorial.org/import-csv-file-mysql-table/)

```
MariaDB [DumasHotel]> LOAD DATA INFILE 'd:/Occupancies.csv'
    -> INTO TABLE Occupancies
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 ROWS;
Query OK, 49 rows affected (0.115 sec)
Records: 49  Deleted: 0  Skipped: 0  Warnings: 0
```

```
MariaDB [DumasHotel]> LOAD DATA INFILE 'd:/Bills.csv'
    -> INTO TABLE Bills
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 ROWS;
```

*Queries needed for the database:

- *Grouping queries*:

These queries show data about different types of rooms in the hotel.

+Rooms by type:

```
MariaDB [DumasHotel]> SELECT RoomTypeName AS 'Room type', COUNT(RoomID) AS 'Number of rooms in each type'
    -> FROM Room r
    -> JOIN RoomType t ON r.RoomTypeID = t.RoomTypeID
    -> GROUP BY RoomTypeName;
+-----------+-----------------------------+
| Room type | Number of rooms in each type |
+-----------+-----------------------------+
| King      |                          13 |
| Queen     |                           8 |
| Standard  |                           4 |
| Suite     |                           3 |
+-----------+-----------------------------+
4 rows in set (0.019 sec)
```

+Rooms by amenities.Notice here that alias are needed for Notes column, because Room and Amenities table have the same column in this case.

```
MariaDB [DumasHotel]> SELECT AmenityName AS 'Amenity type', COUNT(RoomID) AS 'Number of rooms in each type', a.Notes AS 'Amenity details'
    -> FROM Room r
    -> JOIN Amenities a ON r.AmenityID = a.AmenityID
    -> GROUP BY AmenityName
    -> ORDER BY COUNT(RoomID) DESC;
+--------------+-----------------------------+-----------------------------------------------------------------------------------------+
| Amenity type | Number of rooms in each type | Amenity details                                                                         |
+--------------+-----------------------------+-----------------------------------------------------------------------------------------+
| Intermediate |                          11 | Basic amenities plus a kitchenette, hair dryer, towel and bathtub                       |
| Advanced     |                           9 | Intermediate amenities and its own balcony, looking to the sea or to the swimming pool  |
| Pro          |                           6 | Advanced amenities, plus could be served in room by chefs and calling meals in any time |
| Standard     |                           2 | Basic amenities(30-inch TV, free Wi-Fi, refrigerator and personal items)                |
+--------------+-----------------------------+-----------------------------------------------------------------------------------------+
4 rows in set (0.057 sec)
```

+ Bills by price total (in descending order and over 2500 dollars), with data about occupancy number, room number, room type, first date, last date:

```
MariaDB [DumasHotel]> SELECT BillNumber AS "Bill number", b.OccupancyNumber AS "Occupancy ID", o.RoomID AS "Room number", RoomTypeName AS "RoomType", FirstDate AS "Firs
t date", LastDate AS "Last date", PriceTotal AS 'Price total (in dollars)'
    -> FROM Bills b
    -> JOIN Occupancies o ON b.OccupancyNumber = o.OccupancyNumber
    -> JOIN Room r ON o.RoomID = r.RoomID
    -> JOIN RoomType t ON r.RoomTypeID = t.RoomTypeID
    -> WHERE PriceTotal > 2500
    -> GROUP BY PriceTotal DESC;
```

```
+-------------+--------------+-------------+----------+------------+------------+--------------------------+
| Bill number | Occupancy ID | Room number | RoomType | First date | Last date  | Price total (in dollars) |
+-------------+--------------+-------------+----------+------------+------------+--------------------------+
|    10081288 |       100045 |         305 | King     | 2020-02-21 | 2020-03-01 |                     5600 |
|    10081249 |       100006 |         404 | King     | 2018-06-24 | 2018-06-30 |                     3600 |
|    10081289 |       100046 |         301 | King     | 2020-02-26 | 2020-03-03 |                     3400 |
|    10081253 |       100010 |         103 | Queen    | 2019-01-22 | 2019-02-03 |                     3200 |
|    10081274 |       100031 |         204 | King     | 2019-10-11 | 2019-10-18 |                     3150 |
|    10081271 |       100028 |         602 | King     | 2019-09-03 | 2019-09-06 |                     3090 |
|    10081281 |       100038 |         601 | King     | 2020-01-04 | 2020-01-11 |                     3050 |
|    10081254 |       100011 |         305 | King     | 2019-01-23 | 2019-01-29 |                     3000 |
|    10081251 |       100008 |         302 | Queen    | 2018-07-05 | 2018-07-11 |                     2800 |
|    10081263 |       100020 |         103 | Queen    | 2019-06-07 | 2019-06-14 |                     2650 |
|    10081278 |       100035 |         503 | Queen    | 2019-12-21 | 2019-12-25 |                     2600 |
|    10081257 |       100014 |         101 | Standard | 2019-03-03 | 2019-03-08 |                     2550 |
+-------------+--------------+-------------+----------+------------+------------+--------------------------+
12 rows in set (0.009 sec)
```

+ Feedbacks (grouping by LastDate in descending order), including data about customer name, occupancy number, first date, last date and feedback details:

```
MariaDB [DumasHotel]> SELECT FeedbacksID AS 'Feedback number', f.OccupancyNumber AS 'Occupancy number', c.Name as 'Customer name', FirstDate AS 'First date', LastDate A
S 'Last date', FeedbackDetails AS 'Feedback details'
    -> FROM Feedbacks f
    -> LEFT JOIN Occupancies o ON f.OccupancyNumber = o.OccupancyNumber
    -> JOIN Customer c ON o.CustomerIDCard = c.CustomerIDCard
    -> GROUP BY LastDate DESC;
```

The result is like this:

Feedback number	Occupancy number	Customer name	First date	Last date	Feedback details
10278	100049	Addia Biddles	2020-06-09	2020-06-18	This should be renovate more!
10277	100048	Lonni Andree	2020-04-21	2020-04-27	Everything are wonderful here!
10276	100047	Dan Buterton	2020-03-09	2020-03-12	A balcon should be added for my room?
10275	100046	James Rodriguez	2020-02-26	2020-03-03	Well, just staying here these days and I am grateful!
10274	100045	Stavros Crocetti	2020-02-21	2020-03-01	This is good, isn't it?
10273	100044	Carce Gott	2020-02-07	2020-02-10	Replenish the water in the swimming pool please?
10271	100042	Dan Buterton	2020-02-01	2020-02-06	This is well constructed!
10270	100041	Ulla Stanislaw	2020-01-15	2020-01-21	I will definitely come back to this hotel!
10269	100040	Manas Kavi	2020-01-13	2020-01-18	Build a kitchen, please!
10267	100038	Stavros Crocetti	2020-01-04	2020-01-11	Now we need to consider more before returning to the hotel?
10268	100039	James Rodriguez	2020-01-07	2020-01-09	Cleaning our room more should resolve every problem!
10266	100037	Holli Mountstephen	2020-01-01	2020-01-06	New plan to figure out though?
10264	100035	Manas Kavi	2019-12-21	2019-12-25	Can you renovate this room?
10263	100034	Orbadiah Vinick	2019-11-20	2019-12-04	Enlarge this room would be nice though?
10262	100033	Tyrus Duckels	2019-11-17	2019-11-19	More employees is better!
10261	100032	Krista Wilding	2019-10-15	2019-10-21	Beautiful, hope to see everyone here!
10260	100031	Millisent Bramah	2019-10-11	2019-10-18	Just gorgeous, and clean!
10259	100030	Nissy Renney	2019-10-03	2019-10-11	It is mostly Ok, except that services could be better though?
10258	100029	Neilla Copcott	2019-09-04	2019-09-19	I should tell everyone about this wonderful hotel!
10257	100028	Nissy Renney	2019-09-03	2019-09-06	Should this room has more amenities?
10256	100027	Twyla Cremer	2019-08-19	2019-09-05	Swimming pool here is the best!
10255	100026	Aaron Aucutt	2019-08-17	2019-08-29	Hopefully everything could be ameliorated by the next time?
10254	100025	Norma Peinke	2019-08-17	2019-08-20	Amenities here are the best in this city!
10253	100024	Millisent Bramah	2019-08-08	2019-08-17	New trees can be planted near the entrance, please?
10252	100023	Nathanial Sulter	2019-07-21	2019-07-25	Employees are well-mannered, and things are good here!
10251	100022	Nathanial Sulter	2019-06-15	2019-06-24	Food and services are pretty good!
10249	100020	Carce Gott	2019-06-07	2019-06-14	Everything are wonderful here, I can't wait to see it again!
10248	100019	Kit Folds	2019-05-26	2019-06-04	Thanks for perfect service for us!
10247	100018	Lonni Andree	2019-05-22	2019-06-01	This is amazing, I should tell to more people about ir!
10246	100017	Twyla Cremer	2019-04-30	2019-05-04	Amenities are quite good at here.
10244	100015	Aaron Aucutt	2019-03-05	2019-03-10	The employees are nice and sympathy!
10243	100014	Sandro Stang-Gjertsen	2019-03-03	2019-03-08	This room should be enlarged.
10242	100013	Neilla Copcott	2019-02-25	2019-03-04	The lamp should be fixed off, it's quite dark!
10241	100012	Doti Dorwood	2019-01-28	2019-02-04	Can you renovate this room? It's quite old!
10239	100010	Kit Folds	2019-01-22	2019-02-03	Three words: Price too high!
10240	100011	Nikolos Bezants	2019-01-23	2019-01-29	Beds in here is the best!
10265	100036	Holli Mountstephen	2019-12-25	2018-12-29	Just enhance the service is good for that.
10238	100009	Adey Follitt	2018-07-15	2018-07-21	Best rooms in the city, hopefully we could visit it more!
10237	100008	Orbadiah Vinick	2018-07-05	2018-07-11	Better to have a bigger TV.
10236	100007	Jacqueline Hunnicutt	2018-07-02	2018-07-06	Make our room cleaner and shinier
10235	100006	Jacqueline Hunnicutt	2018-06-24	2018-06-30	Nice rooms, with nice views. I highly support it!
10234	100005	Nikolos Bezants	2018-06-18	2018-06-22	Employees should be more polite
10233	100004	Norma Peinke	2018-05-13	2018-05-24	The bath needs to be clearer and we should have more amenities.
10232	100003	Krista Wilding	2018-05-11	2018-05-16	The waiter is quite good, and service is the best!
10245	100016	Doti Dorwood	2019-03-28	2018-03-31	Price is not so suitable with us here?
10231	100002	Ulla Stanislaw	2018-03-06	2018-03-12	It's beautiful, hope to revisit it many times!
10230	100001	Tyrus Duckels	2018-02-04	2018-02-11	It's quite nice and clean here!

- *Accessing data :*

+ For all rooms in the hotel, grouping by RoomID (in ascending order):

```
MariaDB [DumasHotel]> SELECT RoomID AS 'Room number', RoomTypeName AS 'Room type', BedTypeName AS 'Bed type', RoomStatus AS 'Room status', AmenityName AS 'Amenity type'
, r.Notes AS 'Room notes'
    -> FROM Room r
    -> JOIN RoomType t ON r.RoomTypeID = t.RoomTypeID
    -> JOIN BedType b ON r.BedTypeID = b.BedTypeID
    -> JOIN RoomStatus s ON r.RoomStatusID = s.RoomStatusID
    -> JOIN Amenities a ON r.AmenityID = a.AmenityID
    -> GROUP BY RoomID ASC;
+-------------+-----------+------------+-----------------+-------------+-----------------------------------------------+
| Room number | Room type | Bed type   | Room status     | Amenity type | Room notes                                    |
+-------------+-----------+------------+-----------------+-------------+-----------------------------------------------+
|         101 | Standard  | Single     | Occupied        | Standard    | Amenities needed to be upgraded               |
|         102 | King      | Double     | Ready for guests | Advanced    | NULL                                          |
|         103 | Queen     | Single     | Ready for guests | Intermediate | NULL                                          |
|         104 | Standard  | Single     | Arranging       | Intermediate | NULL                                          |
|         105 | Queen     | Queen-sized | Occupied        | Intermediate | NULL                                          |
|         201 | King      | King-sized | Upgrading       | Intermediate | Upgrades will finish in November 2020         |
|         202 | Suite     | Queen-sized | Ready for guests | Pro         | NULL                                          |
|         203 | King      | King-sized | Ready for guests | Pro         | NULL                                          |
|         204 | King      | Sofa-bed   | Upgrading       | Standard    | NULL                                          |
|         205 | Queen     | Queen-sized | Ready for guests | Intermediate | NULL                                          |
|         301 | King      | King-sized | Upgrading       | Advanced    | Renovation will finish in December 2020       |
|         302 | Queen     | Single     | Arranging       | Advanced    | NULL                                          |
|         303 | King      | Double     | Ready for guests | Pro         | NULL                                          |
|         304 | Standard  | Double     | Upgrading       | Intermediate | NULL                                          |
|         305 | King      | Queen-sized | Ready for guests | Advanced    | This room needs to be upgraded after October 2020 |
|         401 | Suite     | Sofa-bed   | Ready for guests | Pro         | NULL                                          |
|         402 | King      | King-sized | Ready for guests | Intermediate | NULL                                          |
|         403 | Queen     | Queen-sized | Arranging       | Advanced    | NULL                                          |
|         404 | King      | Double     | Ready for guests | Advanced    | NULL                                          |
|         405 | King      | Double     | Occupied        | Advanced    | NULL                                          |
|         501 | Queen     | Queen-sized | Arranging       | Intermediate | NULL                                          |
|         502 | King      | Double     | Ready for guests | Intermediate | NULL                                          |
|         503 | Queen     | Single     | Occupied        | Intermediate | This room needs to be renovated in beds type  |
|         504 | Standard  | Double     | Ready for guests | Intermediate | NULL                                          |
|         505 | Queen     | Queen-sized | Ready for guests | Advanced    | NULL                                          |
|         601 | King      | Double     | Upgrading       | Advanced    | Upgrading will finish in January 2021         |
|         602 | King      | Sofa-bed   | Ready for guests | Pro         | NULL                                          |
|         603 | Suite     | Sofa-bed   | Arranging       | Pro         | NULL                                          |
+-------------+-----------+------------+-----------------+-------------+-----------------------------------------------+
28 rows in set (0.010 sec)
```

+About a particular occupancy:

```
MariaDB [DumasHotel]> SELECT OccupancyNumber AS "Occupancy ID", o.RoomID AS "Room number", RoomTypeName AS "Room type", BedTypeName AS "Bed type", AmenityName AS "Ameni
ty type"
    -> FROM Occupancies o
    -> JOIN Room r ON o.RoomID = r.RoomID
    -> JOIN RoomType t ON r.RoomTypeID = t.RoomTypeID
    -> JOIN BedType b ON r.BedTypeID = b.BedTypeID
    -> JOIN Amenities a ON r.AmenityID = a.AmenityID
    -> WHERE OccupancyNumber = '100021';
+--------------+-------------+-----------+-------------+--------------+
| Occupancy ID | Room number | Room type | Bed type    | Amenity type |
+--------------+-------------+-----------+-------------+--------------+
|       100021 |         105 | Queen     | Queen-sized | Intermediate |
+--------------+-------------+-----------+-------------+--------------+
1 row in set (0.004 sec)
```

+ About an employee and customer feedbacks for him, to check that whether feedbacks are toward this employee or not:

```
MariaDB [DumasHotel]> SELECT o.OccupancyNumber AS 'Occupancy number', o.EmployeeID AS 'Employee ID', e.Name AS 'Employee name', c.Name AS 'Customer that are served by t
his employee', FeedbackDetails AS 'Feedbacks'
    -> FROM Occupancies o
    -> LEFT JOIN Employee e ON o.EmployeeID = e.EmployeeID
    -> JOIN Customer c ON o.CustomerIDCard = c.CustomerIDCard
    -> JOIN Feedbacks f ON o.OccupancyNumber = f.OccupancyNumber
    -> WHERE o.EmployeeID = '101110';
```

Here is the result:

```
+-----------------+------------+---------------+-----------------------------------------+-----------------------------------------------------------+
| Occupancy number | Employee ID | Employee name  | Customer that are served by this employee | Feedbacks                                                 |
+-----------------+------------+---------------+-----------------------------------------+-----------------------------------------------------------+
|          100007 |      101110 | James Pasikena | Jacqueline Hunnicutt                    | Make our room cleaner and shinier                         |
|          100009 |      101110 | James Pasikena | Adey Follitt                            | Best rooms in the city, hopefully we could visit it more! |
|          100037 |      101110 | James Pasikena | Holli Mountstephen                      | New plan to figure out though?                            |
|          100047 |      101110 | James Pasikena | Dan Buterton                            | A balcon should be added for my room?                     |
+-----------------+------------+---------------+-----------------------------------------+-----------------------------------------------------------+
4 rows in set (0.001 sec)
```

- *Altering data:*

+ For a customer:

```
MariaDB [DumasHotel]> UPDATE Customer
    -> SET Name = "James Rodriguez", EmergencyName = "Carlos Pereira"
    -> WHERE CustomerIDCard = '10419721';
Query OK, 1 row affected (0.019 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

The result could be checked like this:

```
MariaDB [DumasHotel]> SELECT * FROM Customer WHERE CustomerIDCard = '10419721';
+----------------+-----------------+---------------+-------------+----------------+----------------+
| CustomerIDCard | Name            | AccountNumber | PhoneNumber | EmergencyName  | EmergencyPhone |
+----------------+-----------------+---------------+-------------+----------------+----------------+
|       10419721 | James Rodriguez | 3.56E+15      | 5239426570  | Carlos Pereira | 4733404432     |
+----------------+-----------------+---------------+-------------+----------------+----------------+
1 row in set (0.001 sec)
```

+ For an employee:

```
MariaDB [DumasHotel]> UPDATE Employee
    -> SET Title = "Ms.", PhoneNumber = "242156790"
    -> WHERE Name = "Gajah Namani";
Query OK, 1 row affected (0.019 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

The result for this query is like this:

```
MariaDB [DumasHotel]> SELECT * FROM Employee WHERE Name = "Gajah Namani";
+------------+--------------+-------+-------------+
| EmployeeID | Name         | Title | PhoneNumber |
+------------+--------------+-------+-------------+
|     101115 | Gajah Namani | Ms.   |   242156790 |
+------------+--------------+-------+-------------+
1 row in set (0.001 sec)
```

*__References__:

- Jason Steele, How Many Numbers Are on a Credit Card? , Experian, last accessed 15 October 2020, https://www.experian.com/blogs/ask-experian/how-many-numbers-are-on-a-credit-card/

-MySQLTutorial, Import CSV File Into MySQL Table, last accessed 16 October 2020, https://www.mysqltutorial.org/import-csv-file-mysql-table/

*Apprendix: CSV Tables.

RoomTypeID	RoomTypeName
1	Standard
2	Queen
3	King
4	Suite

- CSV file for RoomType table.

BedTypeID	BedTypeName
1	Single
2	Double
3	Queen-sized
4	King-sized
5	Sofa-bed

- CSV file for BedType table.

RoomStatusID	RoomStatus
1	Occupied
2	Arranging
3	Ready for guests
4	Upgrading

- CSV file for RoomStatus table

AmenityID	AmenityName	Notes
1	Standard	Basic amenities(30-inch TV, free Wi-Fi, refrigerator and personal items)
2	Intermediate	Basic amenities plus a kitchenette, hair dryer, towel and bathtub
3	Advanced	Intermediate amenities and its own balcony, looking to the sea or to the swimming pool
4	Pro	Advanced amenities, plus could be served in room by chefs and calling meals in any time

- CSV file for Amenities table.

RoomID	RoomTypeID	BedTypeID	RoomStatusID	AmenityID	Notes
101	1	1	1	1	Amenities needed to be upgraded
102	3	2	3	3	NULL
103	2	1	3	2	NULL
104	1	1	2	2	NULL
105	2	3	1	2	NULL
201	3	4	4	2	Upgrades will finish in November 2020
202	4	3	3	4	NULL
203	3	4	3	4	NULL
204	3	5	4	1	NULL
205	2	3	3	2	NULL
301	3	4	4	3	Renovation will finish in December 2020
302	2	1	2	3	NULL
303	3	2	3	4	NULL
304	1	2	4	2	NULL
305	3	3	3	3	This room needs to be upgraded after October 2020
401	4	5	3	4	NULL
402	3	4	3	2	NULL
403	2	3	2	3	NULL
404	3	2	3	3	NULL
405	3	2	1	3	NULL
501	2	3	2	2	NULL
502	3	2	3	2	NULL
503	2	1	1	2	This room needs to be renovated in beds type
504	1	2	3	2	NULL
505	2	3	3	3	NULL
601	3	2	4	3	Upgrading will finish in January 2021
602	3	5	3	4	NULL
603	4	5	2	4	NULL

-CSV file for Room table

CustomerIDCar	Name	AccountNumber	PhoneNumber	EmergencyName	EmergencyPhone
12655481	Manas Kavi	6.30E+15	9219981233	Jaka Navi	9452321212
10419721	Elysee Gillham	3.56E+15	5239426570	Delilah Caldes	4733404432
12823723	Addia Biddles	6.76E+15	2498083626	Walt Junkinson	3876982268
20348224	Krista Wilding	6.39E+15	3191433585	Judi Tunkin	4563737644
14994408	Millisent Bramah	4.03E+15	3666066716	Ilka Wray	9061611806
12567646	Neilla Copcott	5.10E+15	1759614239	Annabela Gretton	6954029450
14449457	Jacqueline Hunnicutt	2.01E+15	2018751938	Cory Mercer	1262896974
12926387	Nathanial Sulter	3.57E+15	5994708959	Gunilla Iacovacci	8179794297
12134943	Ulla Stanislaw	2.02E+15	7845584896	Carlo Riep	2748591511
18293083	Lonni Andree	5.01E+15	1735059447	Oberon Veighey	6699394050
18059604	Carce Gott	5.10E+15	4267506617	Milo Allgood	2383557489
17383880	Stavros Crocetti	3.54E+15	1407531553	Jobi Steiner	8205697951
18270465	Nissy Renney	5.11E+15	5888097751	Tiffi Patterfield	8411378562
15685710	Norma Peinke	3.58E+15	8676722905	Katharina Coweuppe	9476475776
20071611	Nikolos Bezants	5.10E+15	7162534255	Hiram Stammirs	7466953975
15852779	Aaron Aucutt	3.59E+15	1872576353	Kevin Althrop	4946968750
17472400	Tyrus Duckels	3.55E+15	7014585848	Arluene Glusby	4434917308
20543913	Kit Folds	3.54E+15	2597176886	Alisun Inglish	4047890933
24040614	Adey Follitt	5.10E+15	2876768502	Alli Stratten	7213835174
10330509	Holli Mountstephen	6.76E+15	8863917933	Ricoriki Tomaskov	8353328380
23655489	Doti Dorwood	5.53E+15	1753588673	Ki Glowinski	5681169943
12837434	Orbadiah Vinick	2.02E+15	4299934844	Ernst Gilling	1618146625
23028056	Twyla Cremer	6.39E+15	8587196942	Garrett Memmory	1852283370
10435012	Dan Buterton	5.60E+15	2587626884	Irwinn Durrad	1565922438
19274376	Sandro Stang-Gjertse	2.01E+15	5891169594	Jamil Webling	2755972508

- CSV file for Customer table. Notice here is all account number is stored in type like (2.01E+15), that is format for large numbers stored in CSV file.

OccupancyNumber	RoomID	EmployeeID	CustomerIDCard	FirstDate	LastDate
100001	402	101115	17472400	2018-02-04	2018-02-11
100002	203	101101	12134943	2018-03-06	2018-03-12
100003	203	101106	20348224	2018-05-11	2018-05-16
100004	401	101107	15685710	2018-05-13	2018-05-24
100005	105	101113	20071611	2018-06-18	2018-06-22
100006	404	101103	14449457	2018-06-24	2018-06-30
100007	205	101110	14449457	2018-07-02	2018-07-06
100008	302	101109	12837434	2018-07-05	2018-07-11
100009	201	101110	24040614	2018-07-15	2018-07-21
100010	103	101103	20543913	2019-01-22	2019-02-03
100011	305	101108	20071611	2019-01-23	2019-01-29
100012	405	101115	23655489	2019-01-28	2019-02-04
100013	505	101103	12567646	2019-02-25	2019-03-04
100014	101	101112	19274376	2019-03-03	2019-03-08
100015	501	101101	15852779	2019-03-05	2019-03-10
100016	202	101103	23655489	2019-03-28	2018-03-31
100017	204	101111	23028056	2019-04-30	2019-05-04
100018	102	101104	18293083	2019-05-22	2019-06-01
100019	202	101112	20543913	2019-05-26	2019-06-04
100020	103	101101	18059604	2019-06-07	2019-06-14
100021	105	101111	24040614	2019-06-07	2019-06-14
100022	603	101104	12926387	2019-06-15	2019-06-24
100023	502	101111	12926387	2019-07-21	2019-07-25
100024	302	101104	14994408	2019-08-08	2019-08-17
100025	403	101111	15685710	2019-08-17	2019-08-20
100026	501	101113	15852779	2019-08-17	2019-08-29
100027	201	101102	23028056	2019-08-19	2019-09-05
100028	602	101107	18270465	2019-09-03	2019-09-06
100029	402	101115	12567646	2019-09-04	2019-09-19
100030	102	101108	18270465	2019-10-03	2019-10-11
100031	204	101112	14994408	2019-10-11	2019-10-18
100032	104	101115	20348224	2019-10-15	2019-10-21
100033	205	101113	17472400	2019-11-17	2019-11-19
100034	403	101105	12837434	2019-11-20	2019-12-04
100035	503	101111	12655481	2019-12-21	2019-12-25
100036	303	101101	10330509	2019-12-25	2018-12-29
100037	101	101110	10330509	2020-01-01	2020-01-06
100038	601	101103	17383880	2020-01-04	2020-01-11
100039	504	101109	10419721	2020-01-07	2020-01-09
100040	401	101115	12655481	2020-01-13	2020-01-18
100041	304	101108	12134943	2020-01-15	2020-01-21
100042	301	101115	10435012	2020-02-01	2020-02-06
100043	104	101108	12823723	2020-02-04	2020-02-06
100044	404	101112	18059604	2020-02-07	2020-02-10
100045	305	101102	17383880	2020-02-21	2020-03-01
100046	301	101106	10419721	2020-02-26	2020-03-03
100047	304	101110	10435012	2020-03-09	2020-03-12
100048	405	101113	18293083	2020-04-21	2020-04-27
100049	303	101101	12823723	2020-06-09	2020-06-18

- CSV file for Occupancies table.

BillNumber	OccupancyNumber	PaymentDate	PriceTotal
10081244	100001	2018-02-10	2000
10081245	100002	2018-03-12	2800
10081246	100003	2018-05-16	2500
10081247	100004	2018-05-24	2600
10081248	100005	2018-06-21	1600
10081249	100006	2018-06-30	3600
10081250	100007	2018-07-04	1600
10081251	100008	2018-07-11	2800
10081252	100009	2018-07-21	2100
10081253	100010	2019-02-07	3200
10081254	100011	2019-01-29	3000
10081255	100012	2019-02-04	1800
10081256	100013	2019-03-07	1600
10081257	100014	2019-03-08	2550
10081258	100015	2019-03-10	900
10081259	100016	2019-03-31	1500
10081260	100017	2019-05-04	1480
10081261	100018	2019-06-01	2100
10081262	100019	2019-06-04	1780
10081263	100020	2019-06-14	2650
10081264	100021	2019-06-14	1650
10081265	100022	2019-06-24	1900
10081266	100023	2019-07-25	1870
10081267	100024	2019-08-17	980
10081268	100025	2019-08-20	1300
10081269	100026	2019-08-29	1640
10081270	100027	2019-09-05	1880
10081271	100028	2019-09-05	3090
10081272	100029	2019-09-19	3150
10081273	100030	2019-10-11	2800
10081274	100031	2019-10-18	3150
10081275	100032	2019-10-21	2400
10081276	100033	2019-11-19	1000
10081277	100034	2019-12-04	2450
10081278	100035	2019-12-25	2600
10081279	100036	2019-12-29	1800
10081280	100037	2020-01-06	2100
10081281	100038	2020-01-10	3050
10081282	100039	2020-01-09	850
10081283	100040	2020-01-18	3050
10081284	100041	2020-01-21	2400
10081285	100042	2020-02-05	2500
10081286	100043	2020-02-06	1500
10081287	100044	2020-02-09	1350
10081288	100045	2020-03-01	5600
10081289	100046	2020-03-03	3400
10081290	100047	2020-03-10	960
10081291	100048	2020-04-26	2400
10081292	100049	2020-06-17	3150

- CSV file for Bills table.

EmployeeID	Name	Title	PhoneNumber
101101	Ratana Manas	Mr	125234098
101102	Lukas Gehrig	Mr	167982129
101103	Katherine Goodall	Ms.	491028345
101104	George Danson	Mr	324125678
101105	Trang Nguyen	Ms	407892192
101106	Henry Bayley	Mr	443243452
101107	Georgy Lubakov	Mr	423123890
101108	Alicia Madena	Ms.	129086777
101109	Frederick Lubaki	Mr	459019342
101110	James Pasikena	Mr	678980127
101111	Manira Gurik	Ms.	432909213
101112	Lyndon McDonald	Mr	231245551
101113	Evita Patterson	Ms.	876567980
101114	Elizabeth Lubiki	Ms.	543445555
101115	Gajah Namani	Mr.	201253490

- CSV file for Employee table.