**Hotels.com**

**(CS6360.002 F19) Database Design Final Project**

**By**

**Ali Shariq (sxa190016)**

**Ashika Hande (axh180061)**

**Dnyanesh Tarte (dnt190000)**



**Eric Jonsson School of Engineering and Computer Science**

**The University of Texas at Dallas**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Title** | **Pg.No.** |
| **1** | Requirements | **3** |
| **2** | ER-Diagram | **4** |
| **3** | Relational Schema | **7** |
| **4** | Normalization | **8** |
| **5** | Relational Schema after Normalization | **9** |
| **6** | SQL Commands to create tables | **10** |
| **7** | PL/SQL | **16** |

**Chapter 1**

**Requirements**

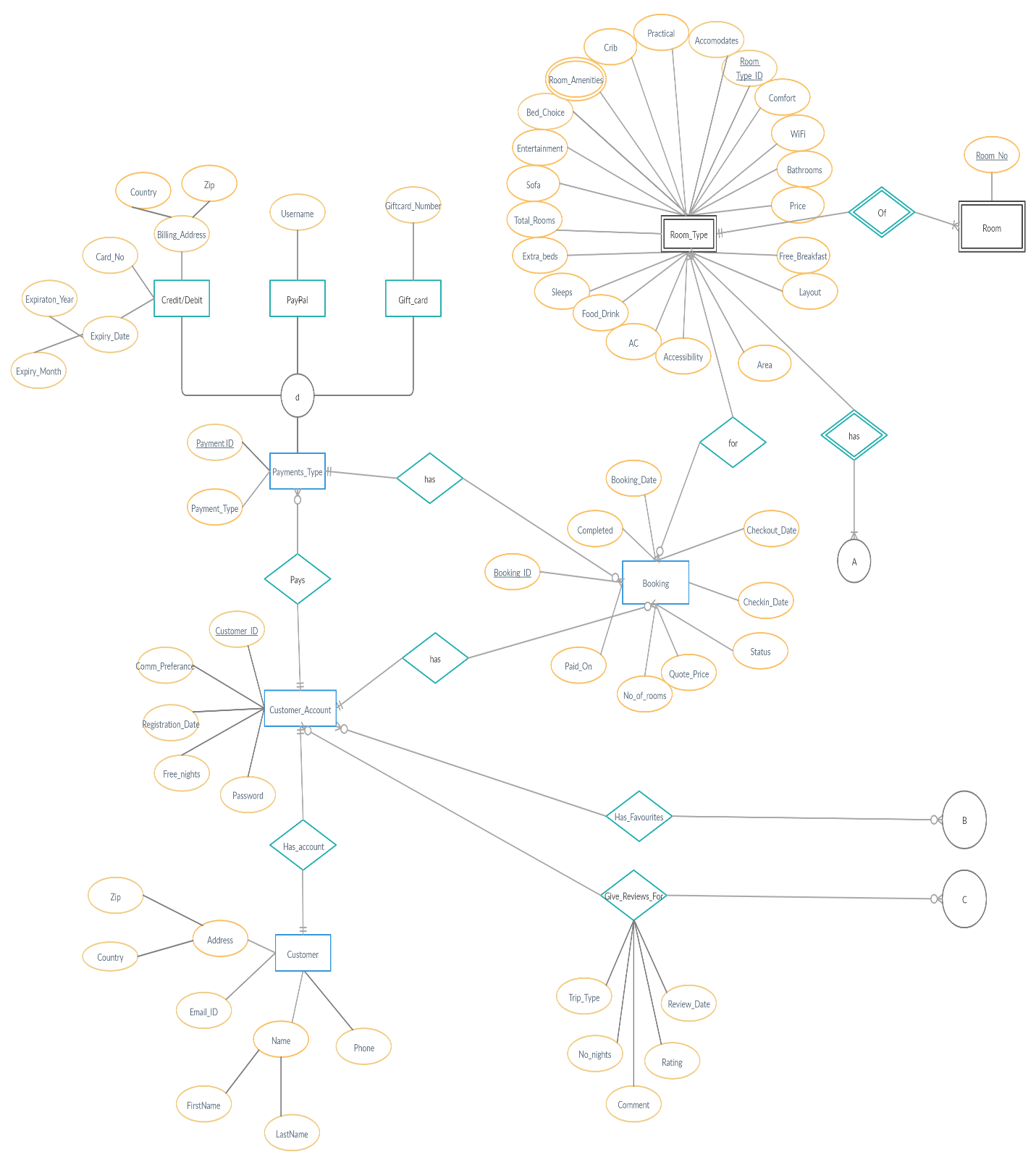
Hotels.com, established in Dallas, Texas in 1991, is a website for booking hotels online all around the world. The company connects customers with hotels in different cities and allows them to choose from a range of hotels. Each hotel has different types of rooms, amenities and places to visit around. Since it is an international company, the website supports multiple languages and currencies along with international payment methods and taxes accordingly. The company also has a robust loyalty program wherein every customer earns ‘free nights’ upon successful check out from previous hotels.

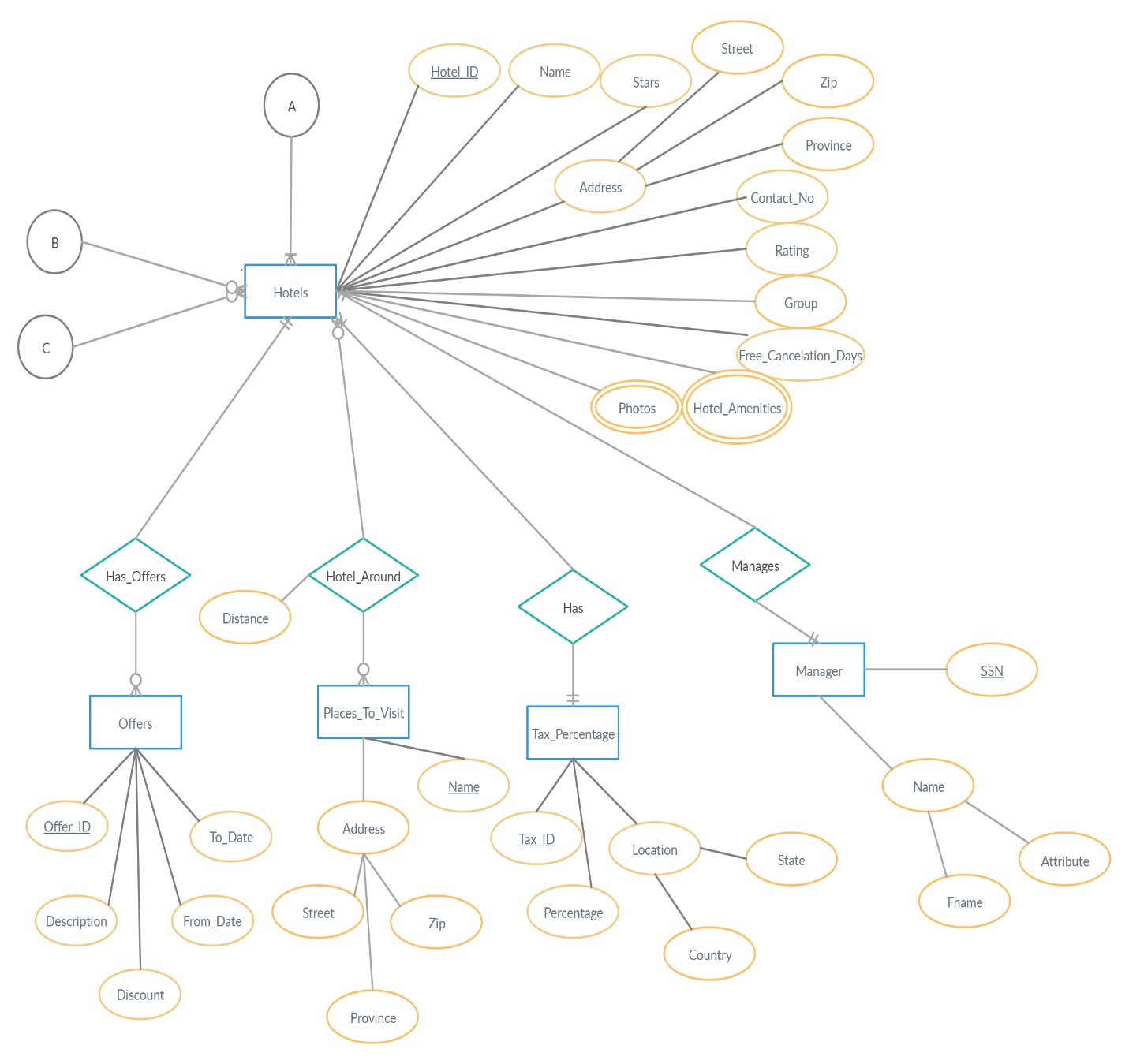
Main entities are:

* Customer: Any individual or a group of individuals who wishes to book a hotel room for a particular date range. A customer can have favorite hotels.
* Hotel: Any individual hotel or a group of hotels in a particular location. Every hotel has a fixed number of rooms of a particular room type. Each room is associated with some amenities particular to that room like A/C, TV etc. There are also some shared amenities like swimming pool, parking, bar etc which are shared by all the customers.
* Payment Type: Since it is an international website, it supports multiple payment methods, credit/debit card, PayPal or gift cards.
* Tax Percentage: The tax depends on the location of the hotel.
* Booking: Stores the booking details by a customer for a hotel. It is connected to the payment type entity to identify the payment method, customer entity to identify the customer and the room type entity to identify the type of room booked in that particular hotel. The customer gets to know the room type but not the room number.
* Amenities: There are 2 types of amenities, hotel amenities like swimming pool, parking, bar etc which are shared by all the customers and room amenities available for a particular room in a hotel like A/C, TV etc.
* Room Type: A hotel can have multiple rooms of multiple kinds.
* Offers: The website gives various offers on hotel bookings which have a start date and an expiry date. An offer is applicable to only one hotel but every hotel can have multiple offers.
* Reviews: A customer can write reviews about one of more hotels.

**Chapter 2**

**ER Diagram**



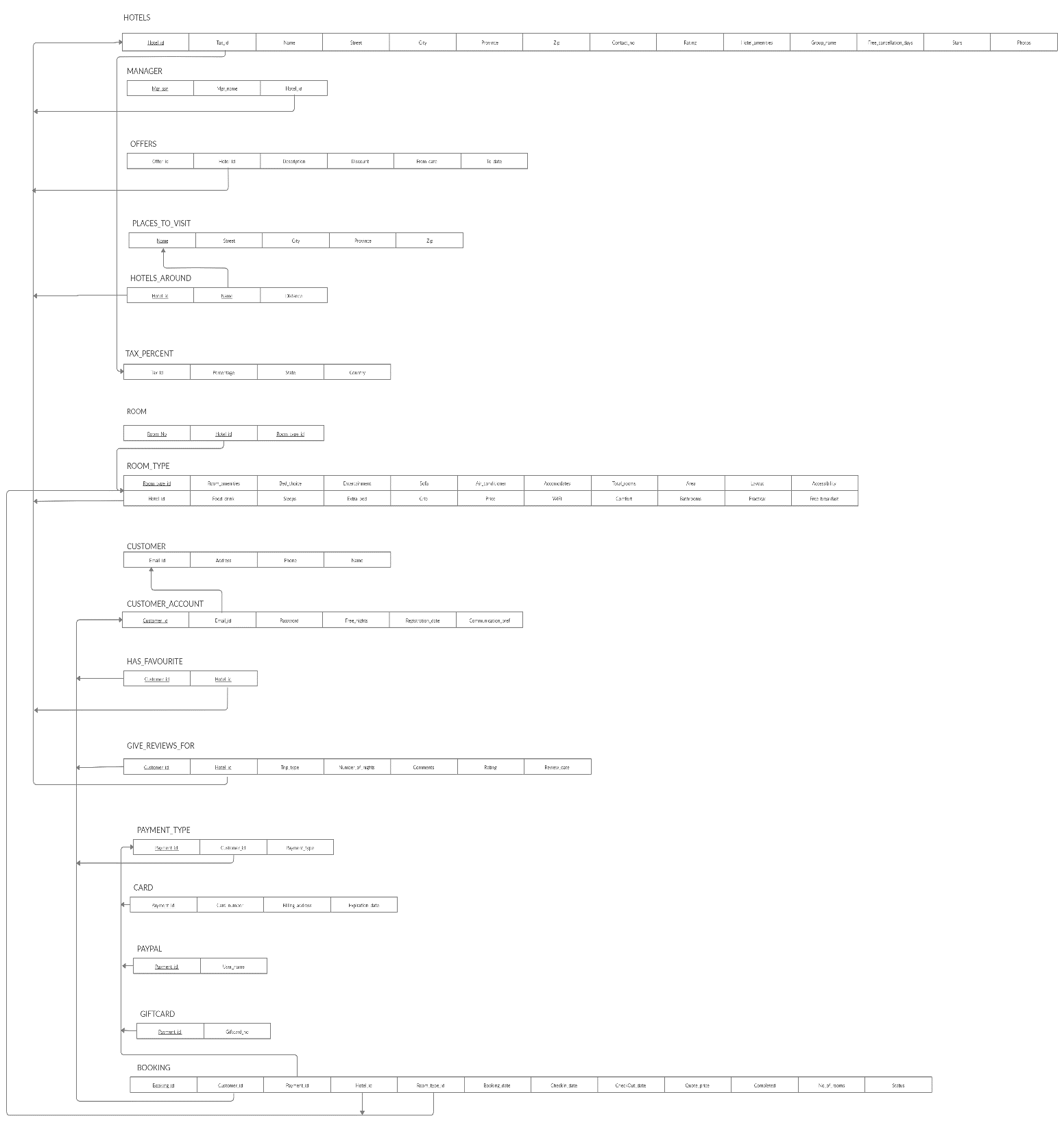


**Summary of the relationships in the ER:**

1. 1:1 Relationships
2. Customer and Customer\_Account
3. Hotels and Manager
4. 1:N Relationships
5. Hotels and Offers
6. Hotels and Tax\_percentage
7. Customer\_Account and Payments\_Type
8. Room\_Type and Room
9. Room\_Type and Hotels
10. Room\_Type and Booking
11. Payments\_Type and Booking
12. Customer\_Account and Booking
13. M:N Relationships
14. Hotels and Places\_to\_visit
15. Customer\_Account and Hotels (Has\_Favourites)
16. Customer\_Account and Hotels (Give\_reviews\_for)

**Chapter 3**

**Relational Schema**

****

**Chapter 4**

**Normalization**

**Violations for 1 NF:**

Hotels (Hotel\_id🡪 Photos, Hotel\_Amenities)

Room\_Type (Room\_Type\_id🡪Room\_Amenities)

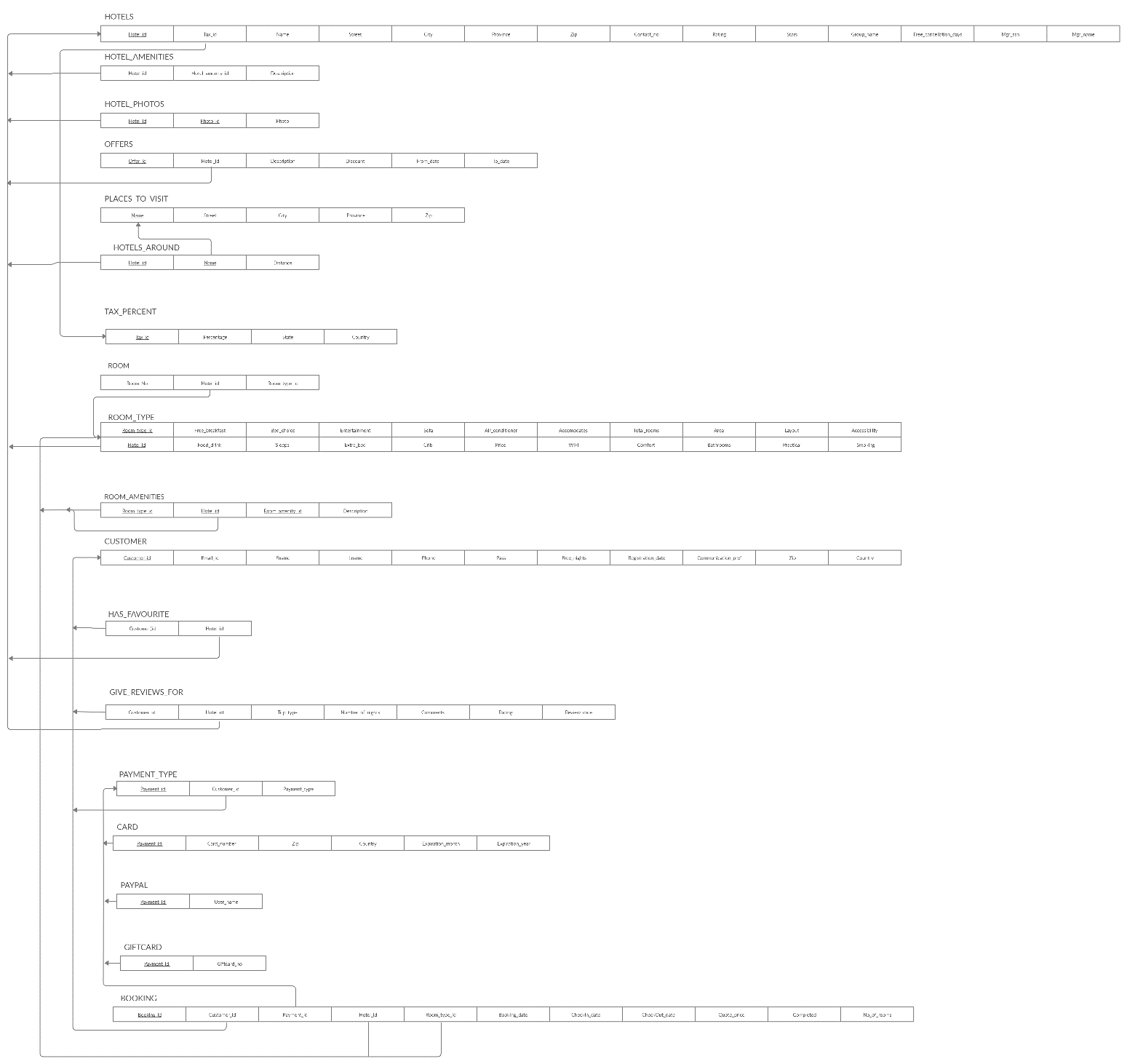
The above attributes have multiple values.

Hence, a separate table for each has been created.

There are no further violations in the relational schema. The following relational schema is in 3NF.

**Chapter 5**

**Relational Schema after Normalization**

****

**Chapter 6**

**SQL Commands to create tables**

CREATE TABLE Customer(

Customer\_id varchar2(10),

Email\_id varchar2(20) NOT NULL,

Fname varchar(10) NOT NULL,

Lname varchar2(10) NOT NULL,

Phone int NOT NULL,

Pass varchar2(8) NOT NULL,

Free\_Nights int DEFAULT 0,

Registration\_Date date NOT NULL,

Communication\_Pref varchar2(20),

Zip int NOT NULL, CHECK(zip BETWEEN 10000 AND 99999),

Country varchar2(20) NOT NULL,

PRIMARY KEY(Customer\_id)

);

CREATE TABLE Payment\_type(

Payment\_id varchar2(10) NOT NULL,

Customer\_id varchar2(10) ,

Payment\_type varchar2(10) NOT NULL,

PRIMARY KEY (Payment\_id),

FOREIGN KEY (Customer\_id ) REFERENCES Customer(Customer\_id ) ON DELETE CASCADE

);

CREATE TABLE Card(

Payment\_id varchar2(10) NOT NULL,

Card\_number varchar2(16) UNIQUE,

Zip int NOT NULL,

Country varchar2(10) NOT NULL,

Expiration\_month int NOT NULL,

Expiration\_year int NOT NULL,

PRIMARY KEY (Payment\_id),

FOREIGN KEY (Payment\_id) REFERENCES Payment\_type(Payment\_id) ON DELETE CASCADE

);

CREATE TABLE Paypal(

Payment\_id varchar2(10),

User\_name varchar2(20) UNIQUE,

PRIMARY KEY (Payment\_id),

FOREIGN KEY (Payment\_id) REFERENCES Payment\_type(Payment\_id) ON DELETE CASCADE

);

CREATE TABLE Giftcard(

Payment\_id varchar2(10) ,

Giftcard\_no int UNIQUE,

PRIMARY KEY (Payment\_id),

FOREIGN KEY (Payment\_id) REFERENCES Payment\_type(Payment\_id) ON DELETE CASCADE

);

CREATE TABLE Hotels(

Hotel\_id varchar2(10),

Tax\_id varchar2(10) NOT NULL,

Name varchar2(10) NOT NULL,

Street varchar2(10) NOT NULL,

City varchar2(10) NOT NULL,

Province varchar2(10) NOT NULL,

Zip varchar2(10) NOT NULL,

Contact int NOT NULL,

Rating int DEFAULT 0,

Stars int NOT NULL,

Group\_name varchar2(10) ,

Free\_cancellation\_days int,

Mgr\_ssn varchar2(10) UNIQUE,

Mgr\_name varchar2(50),

PRIMARY KEY (Hotel\_id),

FOREIGN KEY (Tax\_id ) REFERENCES Tax\_percentage(Tax\_id) ON DELETE SET NULL);

CREATE TABLE Hotel\_Amenities(

Hotel\_id varchar2(10),

Hotel\_amenity\_id varchar2(10),

Description varchar2(30),

PRIMARY KEY (Hotel\_id, Hotel\_amenity\_id ),

FOREIGN KEY (Hotel\_id) REFERENCES Hotels (Hotel\_id) ON DELETE CASCADE

);

CREATE TABLE Hotel\_Photos(

Hotel\_id varchar2(10),

Photo\_id varchar2(10),

Photo blob,

PRIMARY KEY (Hotel\_id, Photo\_id),

FOREIGN KEY (Hotel\_id) REFERENCES Hotels(Hotel\_id) ON DELETE CASCADE

);

CREATE TABLE Has\_favourite(

Customer\_id varchar2(10),

Hotel\_id varchar2(10) ,

PRIMARY KEY (Customer\_id  , Hotel\_id ),

FOREIGN KEY (Customer\_id ) REFERENCES Customer(Customer\_id ) ON DELETE CASCADE ,

FOREIGN KEY (Hotel\_id ) REFERENCES Hotels(Hotel\_id ) ON DELETE CASCADE

);

CREATE TABLE Room\_Type(

Room\_type\_id varchar2(10),

Hotel\_id varchar2(10) NOT NULL,

Accommodates int NOT NULL,

Bed\_choice char NOT NULL,

Total\_rooms int NOT NULL,

Area int NOT NULL,

Layout varchar2(10) NOT NULL,

Accessibility varchar2(20) NOT NULL,

Free\_breakfast char NOT NULL,

Practical varchar(20) NOT NULL,

Bathrooms int NOT NULL,

Comfort varchar2(20) NOT NULL,

WiFi char NOT NULL ,

Price int NOT NULL,

Crib char NOT NULL ,

Entertainment varchar(20) NOT NULL,

Sofa char  NOT NULL,

Extra\_bed char NOT NULL ,

Sleeps int NOT NULL,

Food\_drink char NOT NULL,

Air\_conditioner char NOT NULL,

Smoking char NOT NULL,

PRIMARY KEY (Room\_type\_id, Hotel\_id),

FOREIGN KEY (Hotel\_id) REFERENCES Hotels(Hotel\_id) ON DELETE CASCADE

);

CREATE TABLE Room\_amenities(

Room\_type\_id varchar2(10),

Hotel\_id varchar(10),

Room\_amenity\_id varchar2(10),

Description varchar2(50),

PRIMARY KEY (Room\_type\_id, Hotel\_id, Room\_amenity\_id ),

FOREIGN KEY (Room\_type\_id, Hotel\_id) REFERENCES Room\_type (Room\_type\_id, Hotel\_id) ON DELETE CASCADE

);

CREATE TABLE Room(

Room\_no int,

Room\_type\_id varchar(10),

Hotel\_id varchar(10),

PRIMARY KEY (Room\_no, Room\_type\_id, Hotel\_id),

FOREIGN KEY (Room\_type\_id, Hotel\_id) REFERENCES Room\_type (Room\_type\_id, Hotel\_id) ON DELETE CASCADE

);

CREATE TABLE Booking(

Booking\_id VARCHAR(10),

Customer\_id VARCHAR(10),

Hotel\_id VARCHAR(10),

Payment\_id VARCHAR(10),

Room\_type\_id VARCHAR(10),

Booking\_date DATE NOT NULL,

CheckIn\_date DATE NOT NULL,

CheckOut\_date DATE NOT NULL,

Quote\_price INT NOT NULL,

Completed CHAR NOT NULL,

No\_of\_rooms int NOT NULL,

Status VARCHAR(10),

PRIMARY KEY (Booking\_id),

FOREIGN KEY (Customer\_id) REFERENCES Customer(Customer\_id),

FOREIGN KEY (Room\_type\_id, Hotel\_id) REFERENCES Room\_type (Room\_type\_id, Hotel\_id) ON DELETE CASCADE,

FOREIGN KEY (Payment\_id) REFERENCES Payment\_type(Payment\_id) ON DELETE SET NULL

);

CREATE TABLE Give\_reviews\_for(

Customer\_id varchar2(20),

Hotel\_id varchar2(20),

Trip\_type varchar2(10),

Number\_of\_nights int,

Comments varchar2(40),

Rating int CHECK(Rating BETWEEN 1 AND 10) NOT NULL,

Review\_date date NOT NULL,

PRIMARY KEY (Customer\_id , Hotel\_id),

FOREIGN KEY (Customer\_id ) REFERENCES Customer(Customer\_id) ON DELETE CASCADE,

FOREIGN KEY (Hotel\_id ) REFERENCES Hotels(Hotel\_id) ON DELETE CASCADE

);

CREATE TABLE Offers(

Offer\_id varchar2(10),

Hotel\_id varchar2(10),

Description varchar2(40),

Discount int NOT NULL,

From\_date date NOT NULL,

To\_date date NOT NULL,

PRIMARY KEY (Offer\_id ),

FOREIGN KEY(Hotel\_id ) REFERENCES  Hotels(Hotel\_id) ON DELETE CASCADE);

CREATE TABLE Places\_to\_visit(

Name varchar2(20) NOT NULL,

Street varchar2(100) NOT NULL,

City varchar2(10) NOT NULL,

Province varchar2(10) NOT NULL,

Zip int NOT NULL,

PRIMARY KEY (Name)

);

CREATE TABLE Hotels\_around(

Hotel\_id varchar2(10),

Name varchar2(20),

Distance float NOT NULL,

PRIMARY KEY (Hotel\_id , Name),

FOREIGN KEY (Hotel\_id ) REFERENCES  Hotels(Hotel\_id ) ON DELETE CASCADE,

FOREIGN KEY (Name ) REFERENCES  Places\_to\_visit(Name ) ON DELETE CASCADE

);

CREATE TABLE tax\_percentage(

Tax\_id varchar2(10) NOT NULL,

Percentage float NOT NULL,

State varchar2(20) NOT NULL,

Country varchar2(20) NOT NULL,

PRIMARY KEY (Tax\_id )

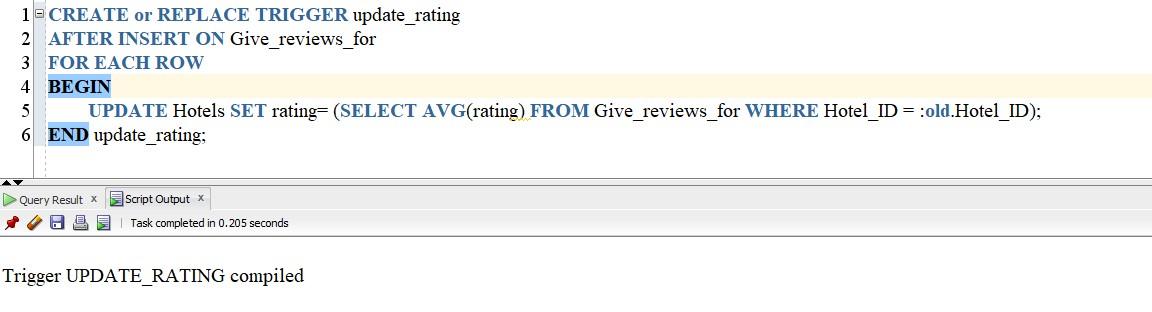
);

**Chapter 7**

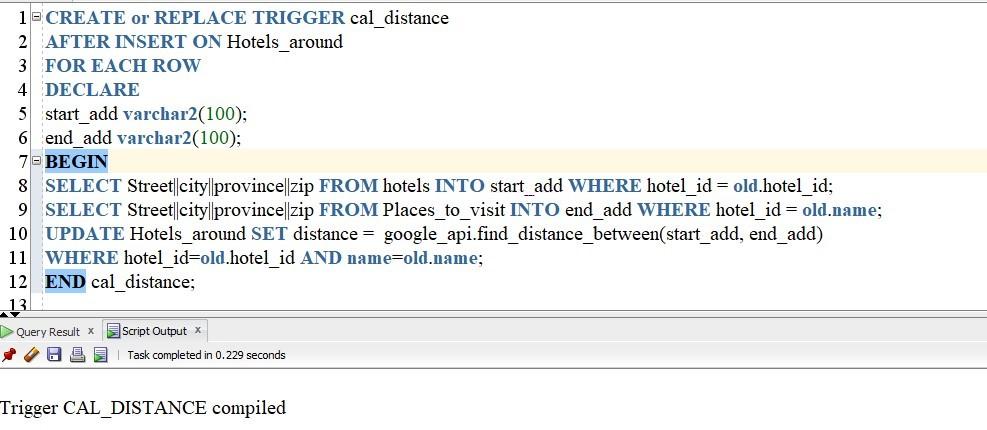
**PL\SQL**

* **Triggers**

1. Trigger to update the rating of a hotel to the average rating after a customer gives review for the hotel:

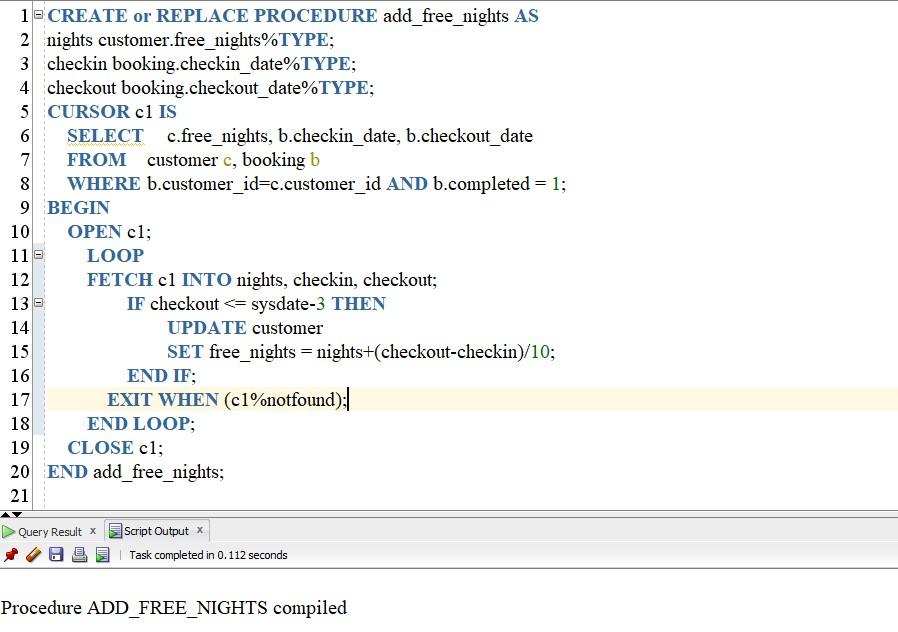


1. Trigger to calculate the distance of a place from the hotel after a new place has been inserted around the hotel. The following trigger shall make use of google map API. Input to the API function are the location of the hotel and location of the place:

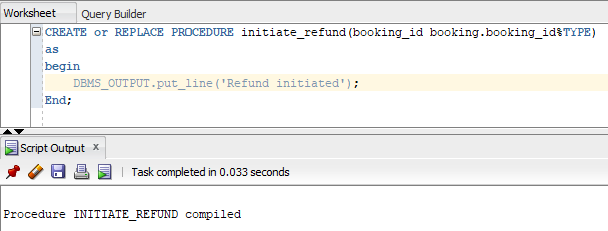
****

* **Procedures**

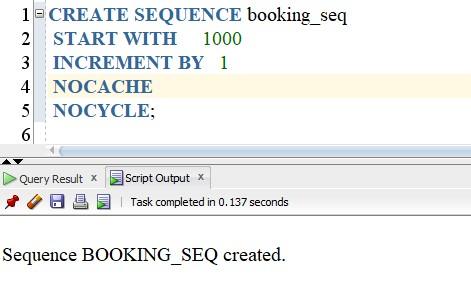
1. Procedure to add number of free nights to the customer account 72 hours each new booking. This procedure can be scheduled to be run every hour or every night:



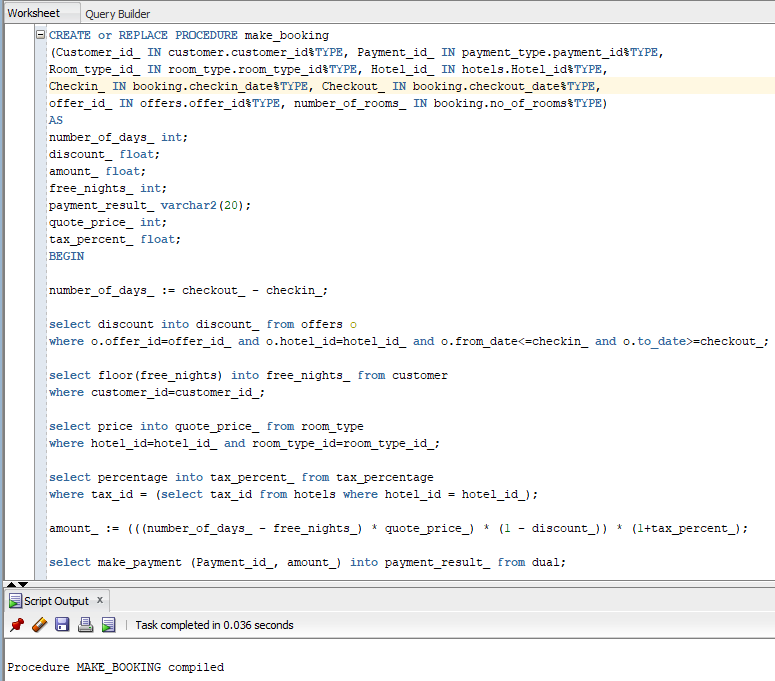
1. A dummy procedure to imitate initiating refund for a booking which was not successful:

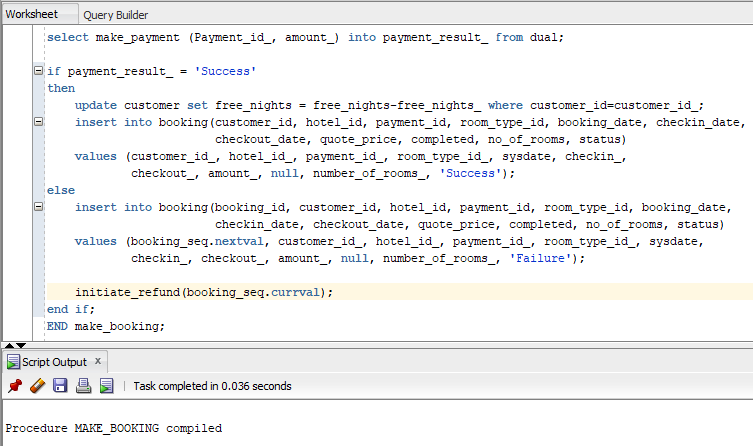


1. Sequence to generate unique booking id for every entry into the booking table. This sequence is required for the below procedure:



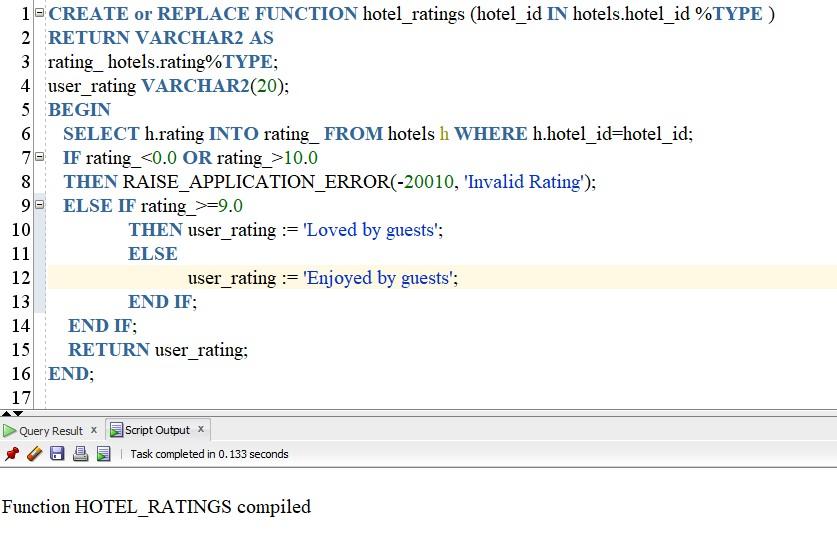
The below procedure makes a booking if payment is successful and initiates refund if there was a problem with the booking:





* **Functions**

1. Function to fetch user readable hotel rating based on average of previous ratings:



2) A dummy function to imitate making payment for a booking and sending the status of the transaction to the Make\_booking procedure:

