Group 10

Barış Can Kayabaşı

Ege Garipağaoğlu

Arda Hasgül

CMPE 232 Relational Databases

Phase 2

Changes we made after phase 1 of the project;

- Since the employee table could store the information belonging to the employee experience table, we removed that table in order to prevent unnecessary operations that cost memory.
- We can reach out to departments of hotels using the employee table, so we do not need another connection between departments and hotels tables because we can already access it.
- We have added a name and surname attribute to the employee table.
- Is_city_center attribute added to hotels table in order to check if the existing hotel takes place in the city center or not.
- We have changed the order of the hotel room and hotel room type tables.
- Phone number, birth date and email address attributes added to the customer table in order to improve the database.
- Some of the varchar's lengths were not big enough to store data, so we incremented them to prevent further errors.
- Removed some of the "not null" requirements as they were necessary.
- Some of the attributes such as phone changed from int to varchar as it is much more useful to do so.
- Used "Date" types of attributes instead of varchar to store date information. In this way we can easily write queries about specific time periods.

ER Diagram and Description of the Tables

We have made quite a few changes, we strongly think that the new version of our database management system is much more realistic and useful now. Because of the changes we made, our ER diagram has changed too.

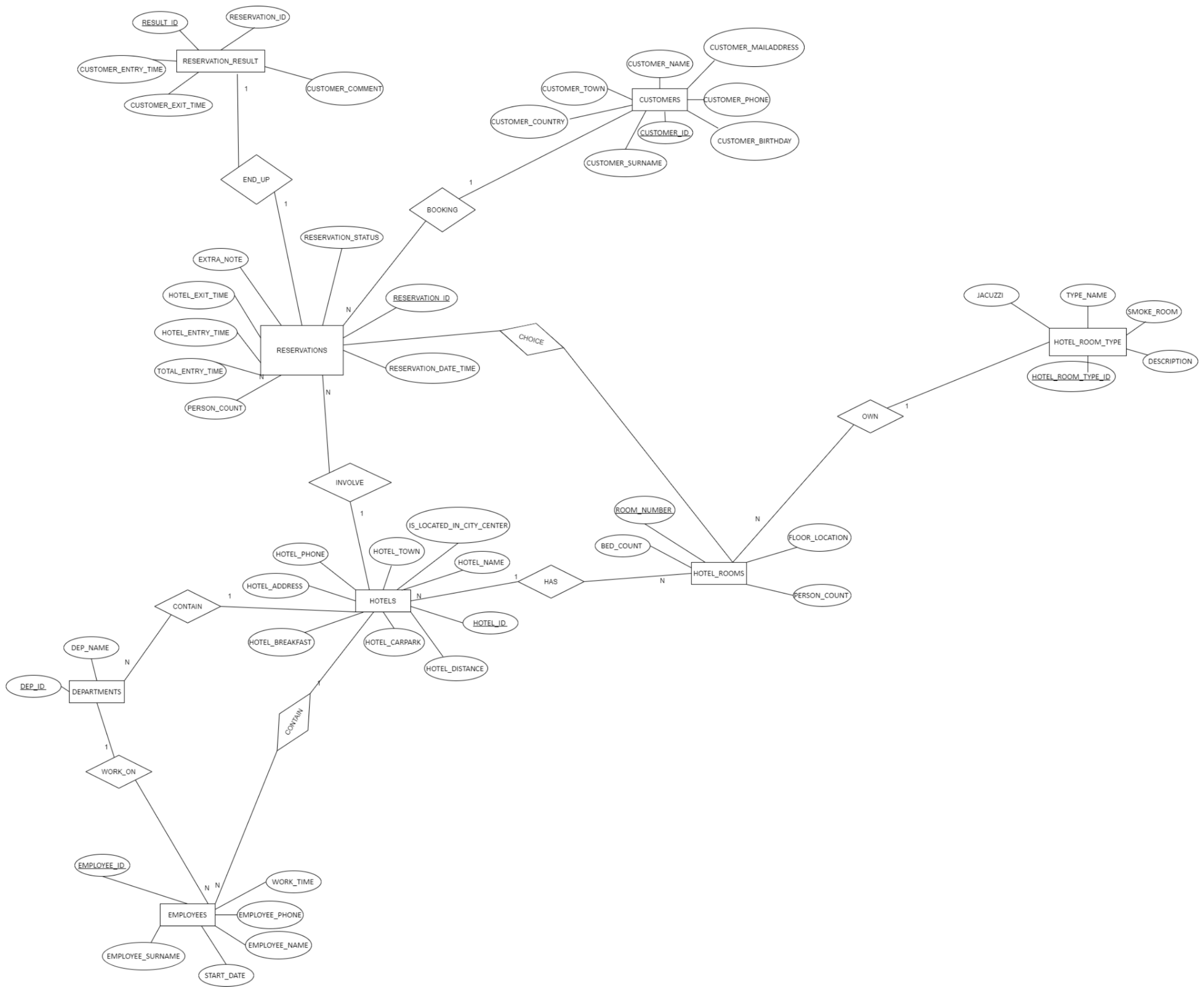


Figure 1: E/R Diagram of Hotel DBMS

Description of Tables

In this part tables in the Railway Database Management System are being explained in a detailed way. There are 2 kinds of tables; Strong and Weak. Strong ones are; Hotels, Hotel_Room_Type, Departments and Customers tables are the strong ones.

1. Hotels: This table has 10 attributes. First one is a unique hotel id. The next 6 attributes hold the information about the hotel's name, the country and city it belongs to, and finally information about contact info, breakfast and car park service availability. Last to 2 attributes are for holding data about city center location. Aim of this table is to provide important information about the hotels that we have in our databases considering they hold significant importance.
2. Hotel Room Type: This specific table is created to store information about different types of rooms, such as queen or king room. This table has 6 attributes that hold important information facilities of the room. By this way we have categorized rooms.
3. Departments: If we take a look at the actual structure of hotels, we can see that it consists of departments such as the front office etc. In order to have a realistic database we have added departments table to our DBMS. This table consists of 2 attributes, one to hold a unique department ID and another to hold the name of the department.
4. Customers: Last strong table is the customers table. It stores information about customer objects. It has 8 attributes. One is to store a unique customer id and the other is to store the customer's name,surname, birth date, phone number and location information.

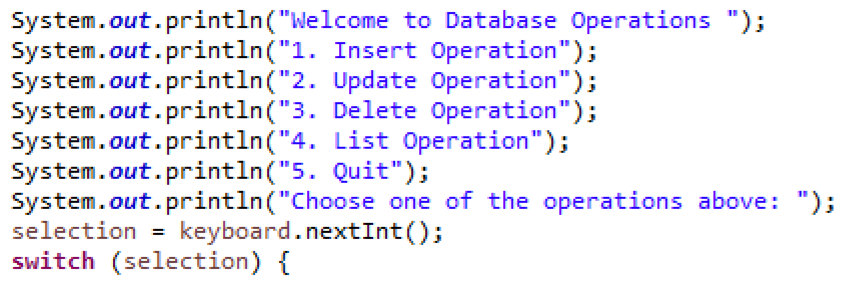
The other tables are the weak ones. It means that they need the presence of another table to form. So they can not exist by themselves. Those tables are;Hotel Rooms, Employees, Reservations and Reservation Results.

1. Hotel Rooms: This table has 6 attributes. First one has a unique Room Number attribute as the primary key. We have foreign keys as type id and hotel id to store information about which type of room it is and what hotel it belongs to.
2. Employees: Employees table has employee id as primary key. Other attributes are for storing information about an employee's name, surname, contact number, work time in years. It has 2 foreign keys to store information about where an employee works at and which department it belongs to.
3. Reservations: This table has 10 attributes. First one is a unique reservation id. Other attributes are for storing information about the reservation that customers made, such as date and price information. Lastly it has a reservation status attribute check as if it is cancelled or not.
4. Reservation Result:This table holds information about when a customer made entry to the hotel reservation and customer’s comments about the hotel.

Explanation of the Java Application Program

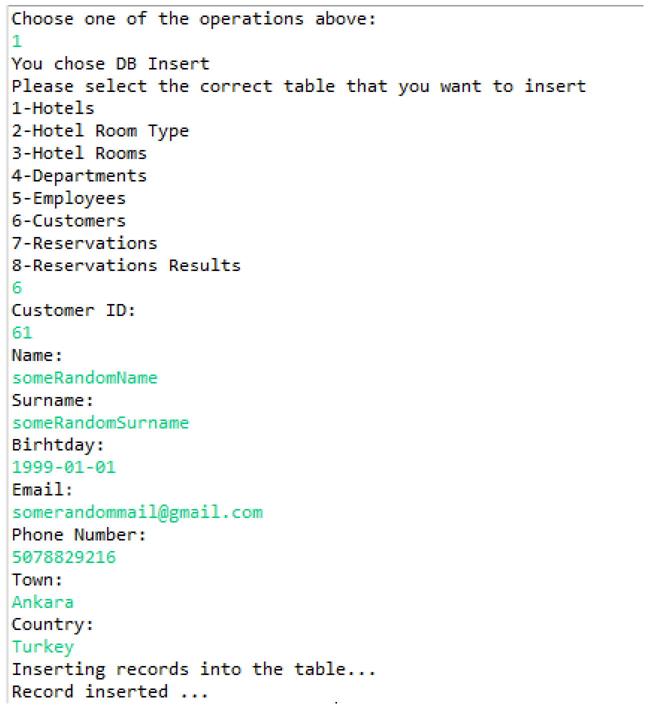
We have written a database application program in Java that accesses our database and does various tasks related to our domain. Since it is an only demo application, it is not fully functional. In order to access our database we need 4 specific pieces of information which are DBC_DRIVER, DB_URL, User and Password. Those variables are needed to connect the school's servers. If connection is successful, we are welcomed by a message which demonstrates a list of operations that we are able to perform. Since it is a demo application, we can only insert, update and delete on a limited number of tables.

When we launch the java application, if a connection is made successfully we are welcomed by the operations list and asked to choose one. If the connection is not successful then "There was an exception in connecting" message is displayed in the console.



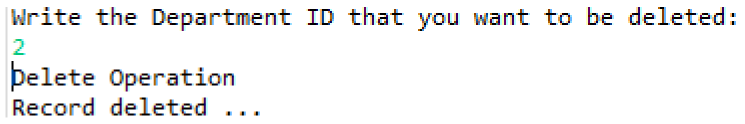
```
System.out.println("Welcome to Database Operations ");
System.out.println("1. Insert Operation");
System.out.println("2. Update Operation");
System.out.println("3. Delete Operation");
System.out.println("4. List Operation");
System.out.println("5. Quit");
System.out.println("Choose one of the operations above: ");
selection = keyboard.nextInt();
switch (selection) {
```

The first option, which is numbered one and is the insert process, is the first option. Since this is just a demonstration, not all of the tables are covered. Only the customer table is covered. We define customer id, name, surname, birthday, email, city and town information to our application then it is inserted to our database management system.

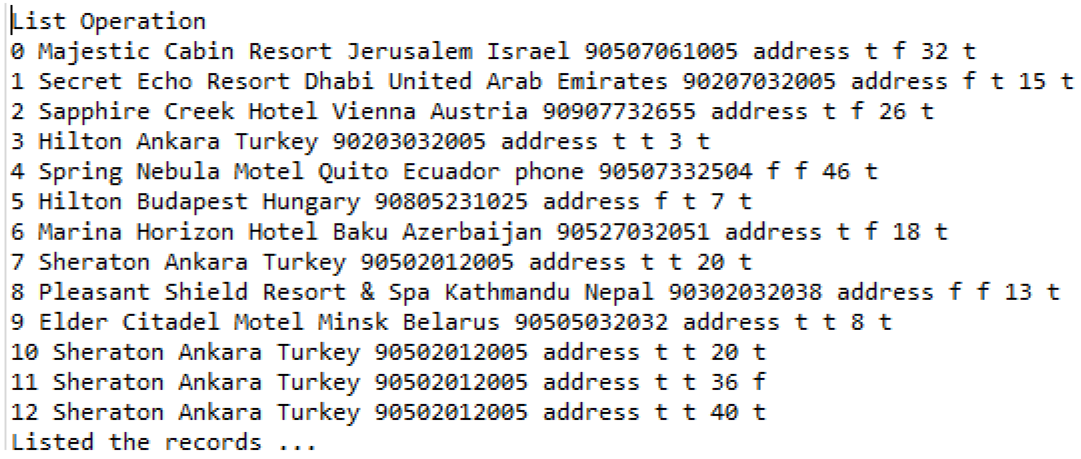


```
Choose one of the operations above:
1
You chose DB Insert
Please select the correct table that you want to insert
1-Hotels
2-Hotel Room Type
3-Hotel Rooms
4-Departments
5-Employees
6-Customers
7-Reservations
8-Reservations Results
6
Customer ID:
61
Name:
someRandomName
Surname:
someRandomSurname
Birhtday:
1999-01-01
Email:
somerandommail@gmail.com
Phone Number:
5078829216
Town:
Ankara
Country:
Turkey
Inserting records into the table...
Record inserted ...
```

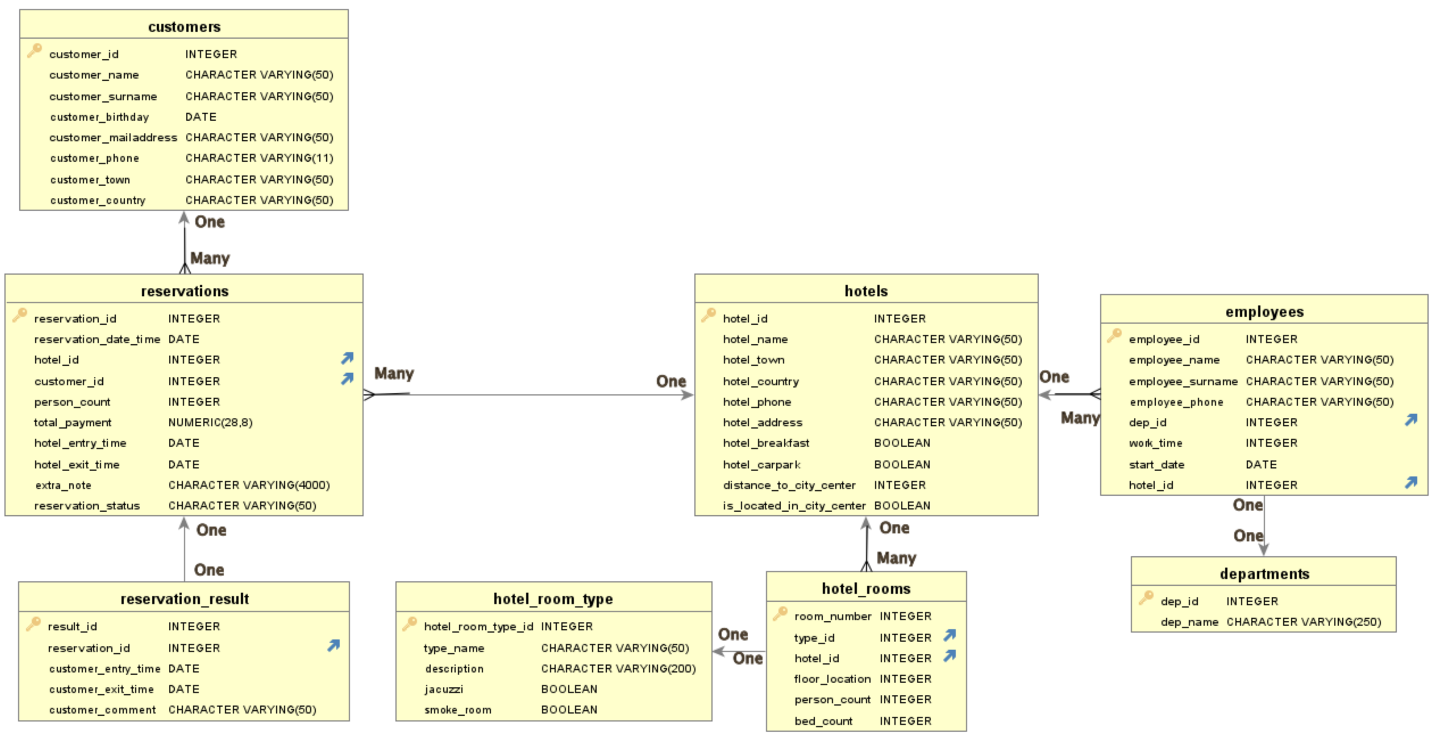
Third option is delete operation. Department table is covered for delete operation. We only need to give the ID information of the department then it is removed by the application.


```
Write the Department ID that you want to be deleted:
2
Delete Operation
Record deleted ...
```

Last one is list operation. Hotels and customers tables are covered for the list operation. The query we have written is executed when list operation is requested by the user. The below image demonstrates the sample list operation. We also have Customers table, the purpose is to show that insert function is working properly, we can insert customer object then list customer table


```
List Operation
0 Majestic Cabin Resort Jerusalem Israel 90507061005 address t f 32 t
1 Secret Echo Resort Dhabi United Arab Emirates 90207032005 address f t 15 t
2 Sapphire Creek Hotel Vienna Austria 90907732655 address t f 26 t
3 Hilton Ankara Turkey 90203032005 address t t 3 t
4 Spring Nebula Motel Quito Ecuador phone 90507332504 f f 46 t
5 Hilton Budapest Hungary 90805231025 address f t 7 t
6 Marina Horizon Hotel Baku Azerbaijan 90527032051 address t f 18 t
7 Sheraton Ankara Turkey 90502012005 address t t 20 t
8 Pleasant Shield Resort & Spa Kathmandu Nepal 90302032038 address f f 13 t
9 Elder Citadel Motel Minsk Belarus 90505032032 address t t 8 t
10 Sheraton Ankara Turkey 90502012005 address t t 20 t
11 Sheraton Ankara Turkey 90502012005 address t t 36 f
12 Sheraton Ankara Turkey 90502012005 address t t 40 t
Listed the records ...
```

Final Schema Of Tables



customers

customer_id	INTEGER
customer_name	CHARACTER VARYING(50)
customer_surname	CHARACTER VARYING(50)
customer_birthday	DATE
customer_mailaddress	CHARACTER VARYING(50)
customer_phone	CHARACTER VARYING(11)
customer_town	CHARACTER VARYING(50)
customer_country	CHARACTER VARYING(50)

reservations

reservation_id	INTEGER
reservation_date_time	DATE
hotel_id	INTEGER
customer_id	INTEGER
person_count	INTEGER
total_payment	NUMERIC(28,8)
hotel_entry_time	DATE
hotel_exit_time	DATE
extra_note	CHARACTER VARYING(4000)
reservation_status	CHARACTER VARYING(50)

hotels

hotel_id	INTEGER
hotel_name	CHARACTER VARYING(50)
hotel_town	CHARACTER VARYING(50)
hotel_country	CHARACTER VARYING(50)
hotel_phone	CHARACTER VARYING(50)
hotel_address	CHARACTER VARYING(50)
hotel_breakfast	BOOLEAN
hotel_carpark	BOOLEAN
distance_to_city_center	INTEGER
is_located_in_city_center	BOOLEAN

employees

employee_id	INTEGER
employee_name	CHARACTER VARYING(50)
employee_surname	CHARACTER VARYING(50)
employee_phone	CHARACTER VARYING(50)
dep_id	INTEGER
work_time	INTEGER
start_date	DATE
hotel_id	INTEGER

departments

dep_id	INTEGER
dep_name	CHARACTER VARYING(250)

reservation_result

result_id	INTEGER
reservation_id	INTEGER
customer_entry_time	DATE
customer_exit_time	DATE
customer_comment	CHARACTER VARYING(50)

hotel_room_type

hotel_room_type_id	INTEGER
type_name	CHARACTER VARYING(50)
description	CHARACTER VARYING(200)
jacuzzi	BOOLEAN
smoke_room	BOOLEAN

hotel_rooms

room_number	INTEGER
type_id	INTEGER
hotel_id	INTEGER
floor_location	INTEGER
person_count	INTEGER
bed_count	INTEGER

Figure 2:Schema Of Hotel DBMS