**Topic:**

**Hotel Management System**

**Introduction :**

Our project is based on Hotel Data Management. This is designed to store, process, retrieve and analyse information concerned the hotel. This project aims at maintaining all the information pertaining to customers, different groups of workers available in hotel and helps the management of hotel to manage these data in a better way.

Hotel database management system was designed to simplify front desk office tasks, enhance management functions, to improve the experience of customers and workers who are performing their job in the hotel. The project's aim is to develop an application system to minimize the manual work of the hotel, by this hotels can improve their efficiency and effectively manage operations such as check-in, check out and the availability of rooms in hotel.

The main objective of the entire activity is to automate the process of day to day activities of Hotel. Using this system you can manage room activities and keep track of admission of a New Customer. Using this system you can check rooms according to customer’s need and can assign rooms easily to customer. Check-out of a customer and updation of releasing the room information in the system is very easy.

**Data Requirements:**

Customer is an entity type which has many attributes like customer id (Cus\_id), Name, DOB, Age, Mobile Number and Address. Every customer has a unique Customer Id (Cus\_id) which is the primary key of customer entity type. Key cannot have null value. Mob is a multi-valued attribute and Age is the derived attribute which is derived from DOB. There are two other attributes which are Name and Address.

Hotel is an entity type which have Name, Hotel\_id, Address, Contact and Area as their attributes. Hotel has its own id (Hotel\_id this is the primary key) which is unique and cannot be null. Area tells us in how many acres the hotel has as their own property. Address tells about the location of hotel.

Rooms is an entity type which has room\_id, type and status as their attributes. Here room\_id is primary key which is unique and cannot be null. Type tells us about the type of rooms present in the hotel whether they are (AC or non-AC rooms). And the status tells us whether the rooms are empty or any customer stays in the room.

Restaurant has its own Restaurant Id (res\_id) which is unique and it is primary key of the restaurant entity which cannot be null. It has also another attribute which is Name.

Workers is an entity type which has Name, DOB, Age, Mobile No. (Mob) as a multivalued attribute, Workers have their own Id (Worker\_id) which is unique and it is primary key, Age is the derived attribute and also it has another attribute which is Address. Workers entity is a super-class which has Manager, Cleaning staff, Restaurant staff and security staff as their subclasses which means worker can be from cleaning staff or worker can be from restaurant staff or worker can be from security or worker can be Manager also.

Every hotel has their own parking system where their workers and customers can park their vehicles and every parking area has its own capacity of parking the vehicles. As Parking entity has no primary key so it is weak entity type.

Item is another type of an entity which has Name, Cost and type as their attributes. Here the type specifies whether the dish prepared by the restaurant is either the vegetarian dish or non-vegetarian dish. And all the dishes have their particular name so the name here is the primary key (Name) of the entity item.

Payment is another type of an entity which has Transaction Id, Amount and Mode as their attributes. Here Transaction Id is unique for every customer so Trans\_id is the primary key. Also here Mode is single valued attribute.

**Relationships between the entities:**

### Customer stays in Hotel (M-1):

Many customer stays in one of the hotel so there is (M-1) relationship between them. And many Customers who come in the hotel is not necessary that he/she stays in the hotel so there is partial participation from both sides.

### Customer parks their vehicles in the parking (M-1):

Many customers park their vehicles in the one parking of the hotel so there is (M-1) relationship between them. And many Customer who come in the hotel is not necessary that he/she parks their vehicle in the hotel so there is partial participation from both the sides.

### Customer lives in the rooms (1-M):

In one hotel many customers stays at one time so there is (1-M) relationship between them. Customer who

comes in the hotel is not necessary that he/she stays in the hotel definitely also one customer can book more than one room so there is partial participation from both the sides.

### Customer pays their payment (1-1):

Every Customer pays their own bill so the relationship becomes (1-1). And there is total participation from both the sides because it is necessary that every customer stays in the hotel or order something have to pay the bill so there is total participation from both sides.

### Customer orders item (N-M):

Customer can order one or more items and also many customers can order same item so there is N-M relationship. But there is not the total participation from both the sides because it is necessary that every customer order all the items that are prepared by the restaurants in the hotel so there is partial participation from both sides.

### Hotel contains rooms (1-M):

Every Hotel contain the rooms means one hotel contains number of rooms and also all the rooms of the hotel are present inside the hotel so there is (1-M) relationship between them. So there is total participation from both the sides because it is necessary that every room which comes under the hotel need to be present inside the hotel so there is total participation from both sides.

### Hotel have restaurant (1-M):

Every Hotel have restaurant means one hotel contains number of restaurants so there is (1-M) relationship between them. Also it is not necessary that all the types of restaurant are present in one hotel it may or may not be possible in one hotel. So there is total participation from the restaurant side.

### Hotel contain parking area (1-1):

Mostly one hotel has one parking area so there is (1-1) relationship between them. Every Hotel has its own parking area it is not necessary because some hotel does not have their own parking area. So there is partial participation from both sides.

### Workers works in the hotel (M-1):

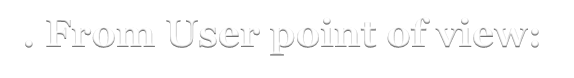
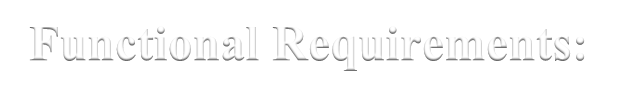
Every Hotel has a team of workers means many workers worked in one hotel means they have (M-1). In this case their total participation from both the sides because it is necessary that every worker who worked in the hotel is comes under workers category in the hotel so there is total participation from both sides.

Also we can say that in a hotel there is different type of workers like the cleaning staff workers who works only the work related to the cleaning perspective or the restaurant staff workers who works only the work related to the restaurant. Also a manager who manages all the works and the security staff which will take the duty related to the safety purpose. Also there is receptionist who works on the reception.

### Restaurant prepares the item (1-M):

One restaurant prepares many items so their is (1-M) relationship between them. But it is not necessary that all items are prepared by one restaurant because Vegetarian restaurant cannot prepares the non-veg dishes also the restaurant which prepares the Asian food cannot prepares the Italian food so there is partial participation from both the sides.

**Functional Requirements:**



1. **From User point of view:**

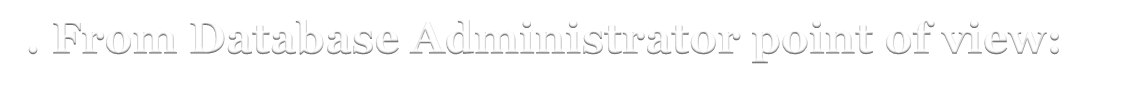
The user will be able to see the details of customers like the details of customers available in the database and also about the details of the workers of the hotel who worked in the hotel. There should be a list of the items that are available at that particular time and that list will be updated regularly. Also there should be the list of the rooms that shows how many rooms are empty and how many rooms are full this will be updated regularly.

So that in case if customer wants to change the room in hotel he/she will be able to change the room without any difficulty. The user will also be able to see clearly about how many items are prepared by the restaurants and also about the no. of vehicles that are parked in the hotel parking area.

The basic functions that user will performs:

* 1. View the details of the workers.
  2. View the data of the customers.
  3. View status of the rooms.
  4. View whether hotels has empty rooms or not.

## From Database Administrator point of view:



Administrator is in charge of creating the website which is used to access the database. Administrator has all the privileges of the user but the administrator has the authority to add and remove the data from the database which the user cannot do.

The administrator should be allowed to enter the details of the customers present in the hotel. He/She have the authority to enter details of how many rooms are available. He/She will make sure that the status of the room should be timely updated. Also the list of customers who check-in and check-out will be handled by the administrator.

If a particular room is not available for the customers the administrator should update that and thus should be able to arrange the backup rooms as soon as possible. If a customer or worker is no more related to the hotel the administrator should delete the records of the customers, or if there are any changes in the workers details the administrator should update them.

The basic functions administrator will performs:

1. Display different list e.g. Vehicles, rooms list, workers list, customers list etc.
2. Display the no. of rooms available.
3. Display the no. of customers that stayed in the hotel.
4. View the customer and worker information.
5. Update the room status after every check-in and check-out of the customers.

### Insertion of Data:

1. When a new customer comes in the hotel so the details of customers should be inserted like the Name, Address, Customer\_Id, Mobile Number, DOB.
2. During the parking of the vehicle the details of the customer’s vehicle or worker’s vehicle is inserted.
3. If new worker joined the hotel the data of the worker is inserted like its Name, Address, Mobile number, DOB and the worker\_id is allotted to that worker.
4. When customer order something the no. of dishes he/she is ordered the quantity of that dish is inserted in the final bill also the type of dish ordered is inserted.
5. Whenever a new room is built in the hotel the type of room details is inserted and also the new room\_id is allotted to that room.

### Deletion of Data:

1. Delete the old data of the customer if he/she is no longer stays in the hotel.
2. Delete the details of the worker if worker resigned the job long time ago.
3. Delete the data of the room if it is not in the condition of giving to the customers.
4. Delete the details of item from the menu if it no longer prepared by the restaurant.
5. Delete the old data about the vehicles that has parked in the past.

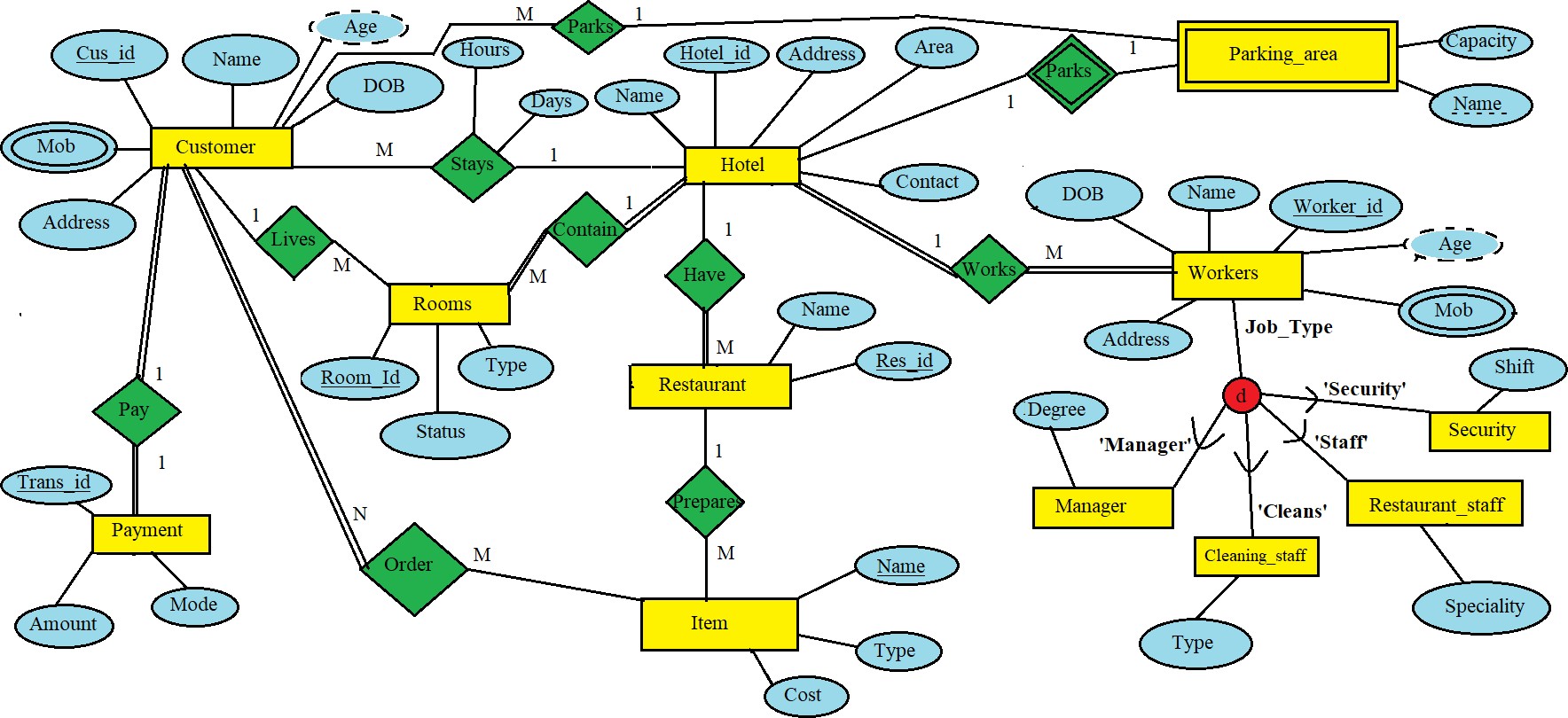
### Updation of Data:

1. Update about the status of the rooms after any customer check-in or check-out the hotel.
2. Update the Mobile No. of the worker if he/she changed their mobile number.
3. Update the details of the item if new items are added in the list.
4. Update the customer details if something entered incorrect during the insertion of data.
5. Update the name of the hotel (in rare circumstances).
6. Update the room details of customers if customer changed their room due to any inconvenience.

### Retrieval of Data:

1. Retrieve the data of the customers who are staying in the hotel.
2. Retrieve the data of the workers who are working in the hotels.
3. Retrieve the data of the rooms that are booked by the customers.
4. Retrieve the data about the restaurant that a hotel has.
5. Retrieve the data about the no. of items ordered by the customers.
6. Check the payment details by which mode customer pays the bill.
7. Retrieve the data about how many customers and workers are in the hotel at present time.
8. Retrieve the data about the dishes that are prepared in particular day.

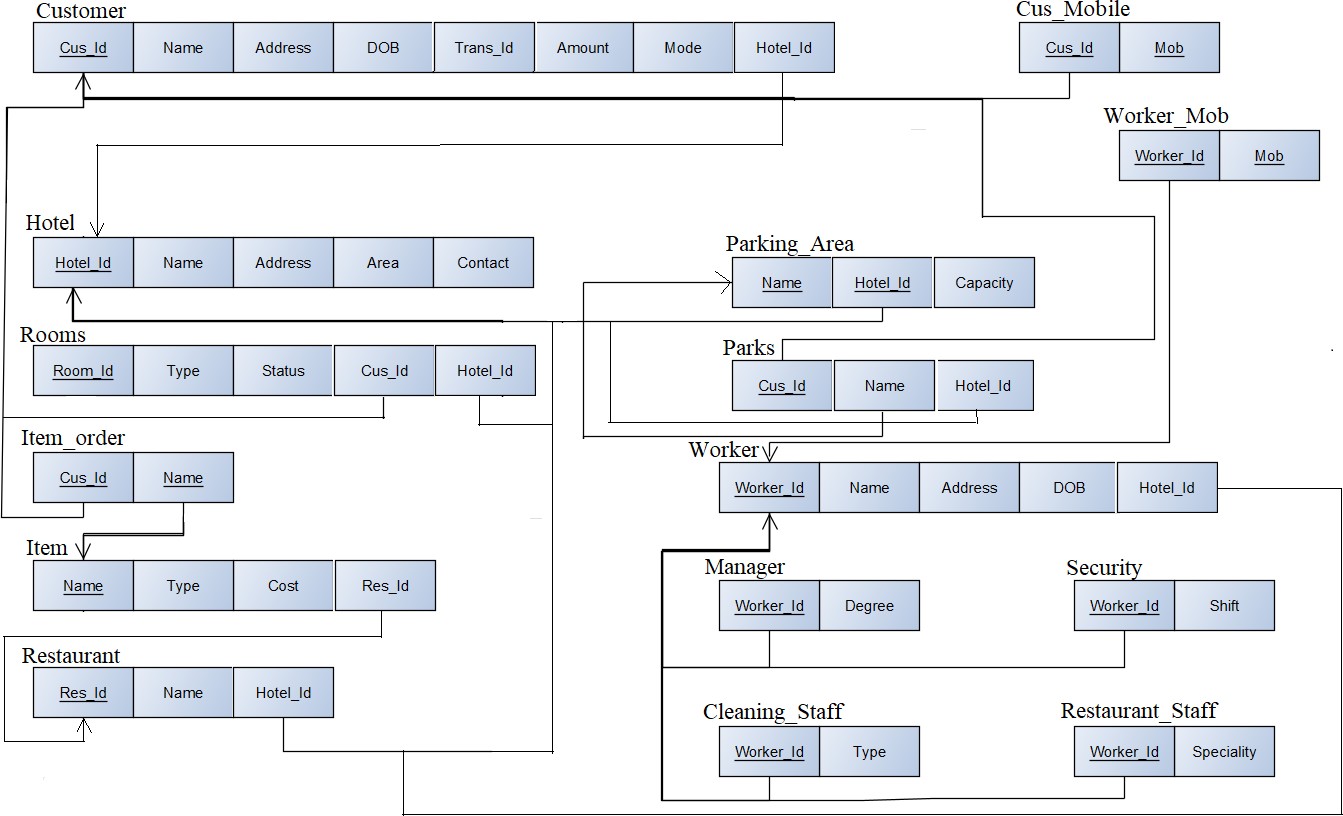
**EER Model along with Key constraints, Participation constraints and Cardinality constraints:**



**Topic:**

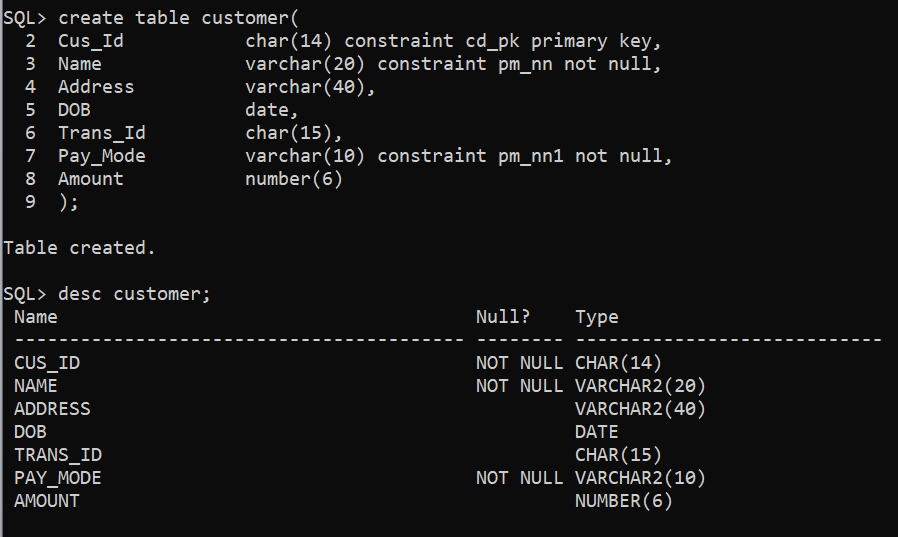
**Hotel Management System**

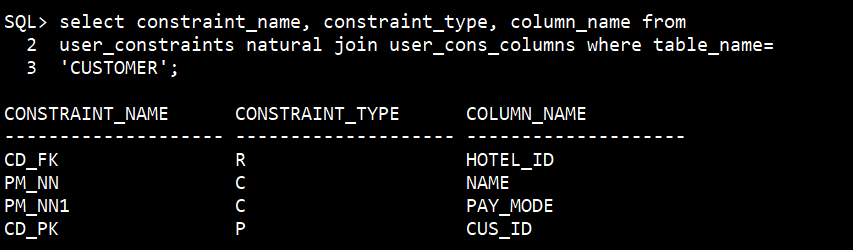
**Relational Database Diagram of EER Model**



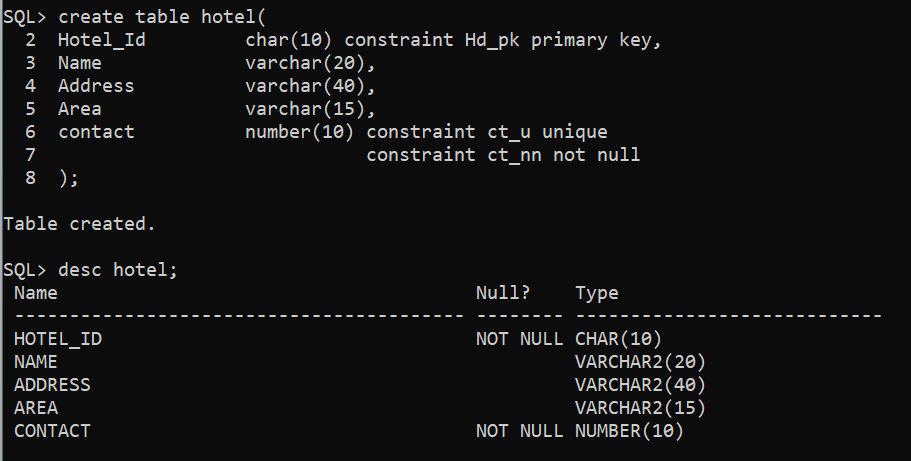
* 1. **Creating the tables :-**
* **Customer and Payment**
* **Hotel**
* **Rooms**
* **Item**
* **Item\_order**
* **Restaurant**
* **Cus\_mobile**
* **Parking\_area**
* **Parks**
* **Worker**
* **Worker\_mob**
* **Manager**
* **Security**
* **Cleaning\_Staff**
* **Restaurant\_Staff**

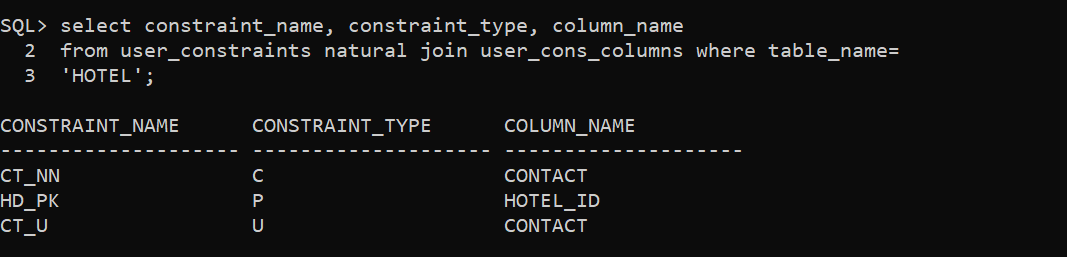
**Customer and Payment**



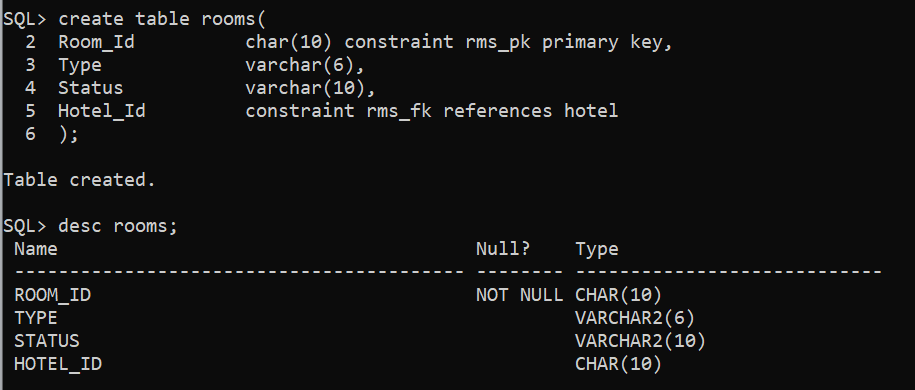


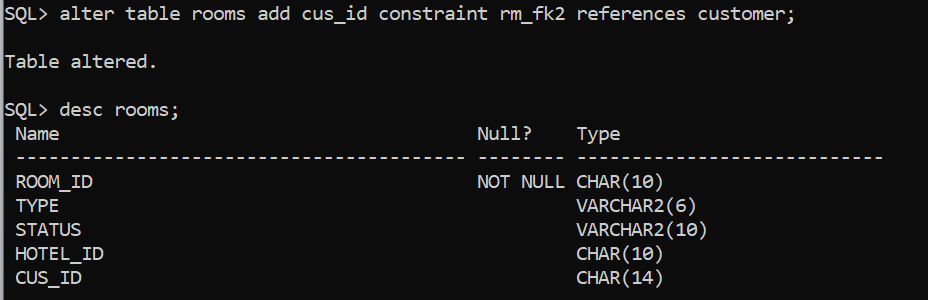
**Hotel**





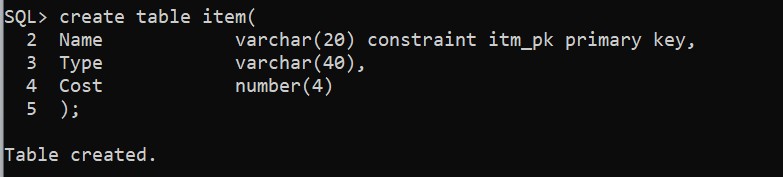
**Rooms**



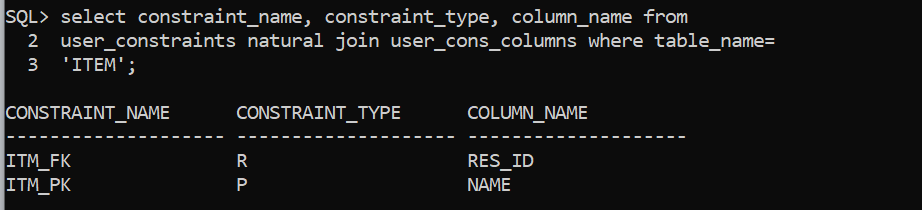




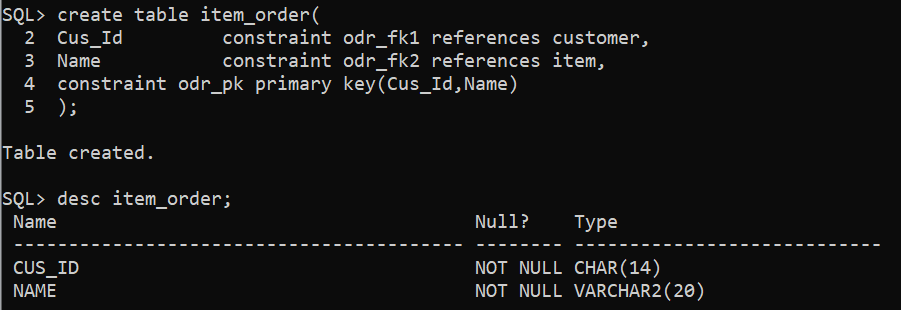
**Item**

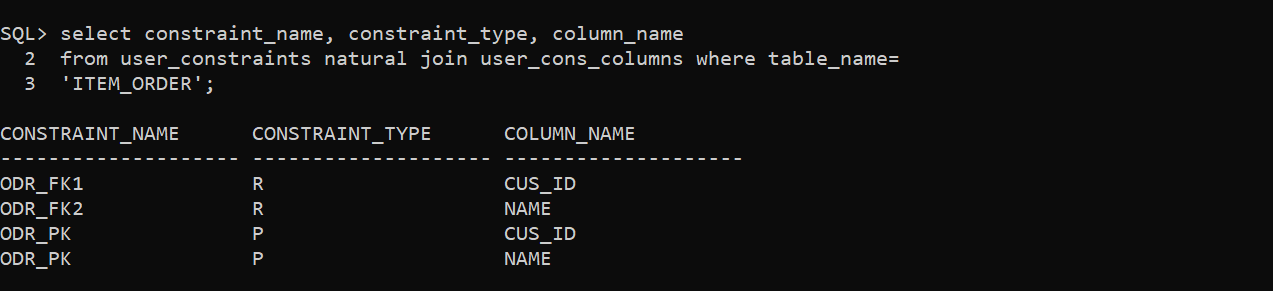




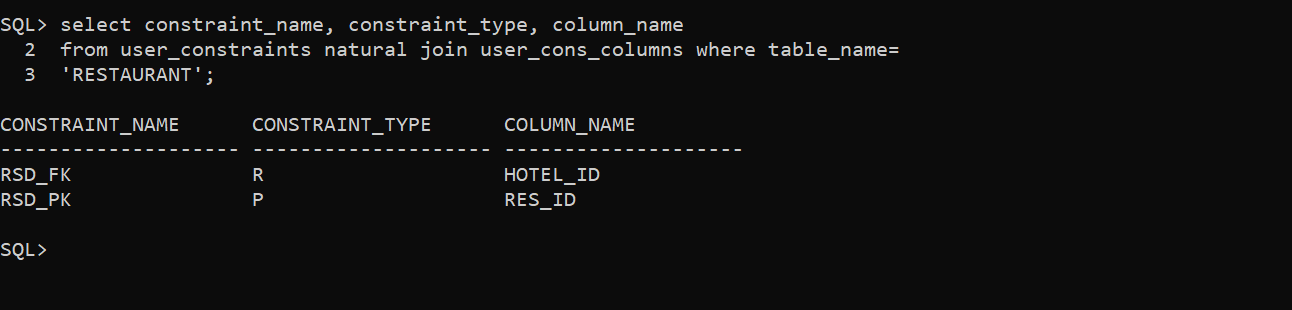
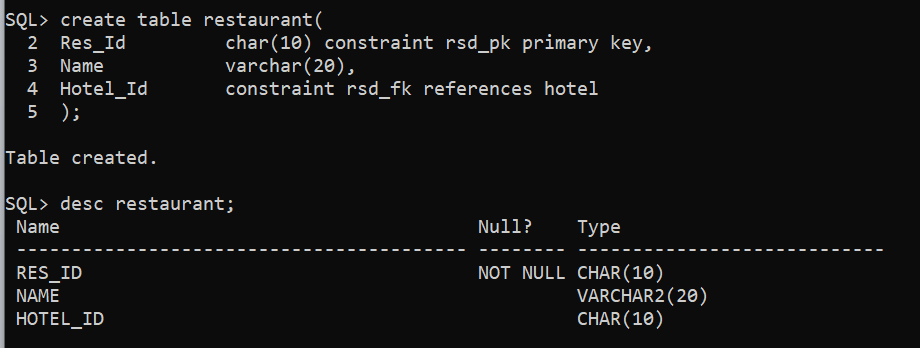


**Item\_Order**

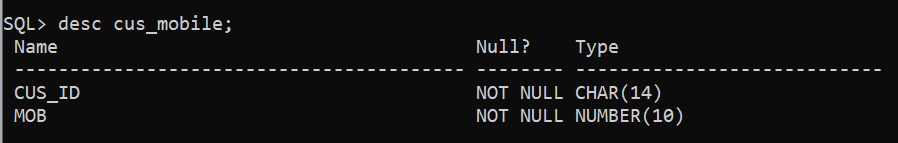




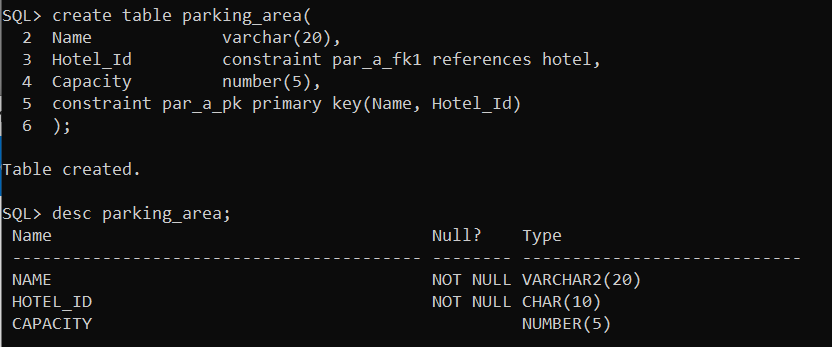
**Restaurant**



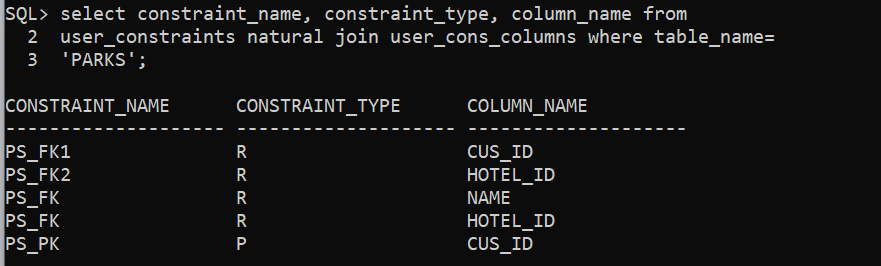
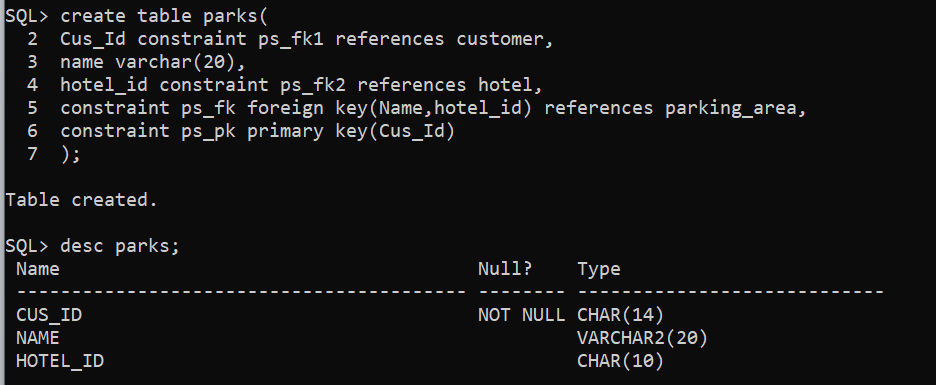
**Cus\_mobile**



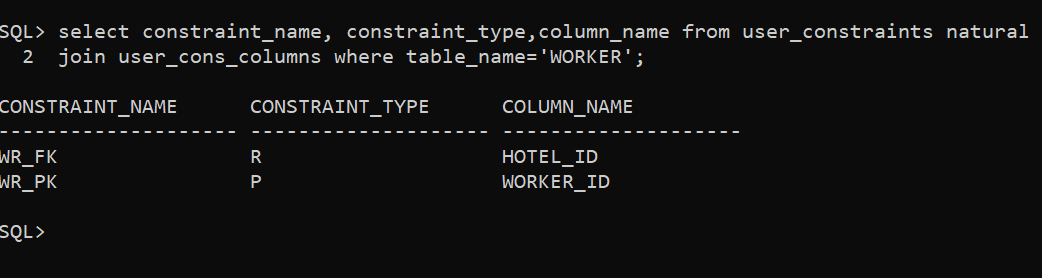
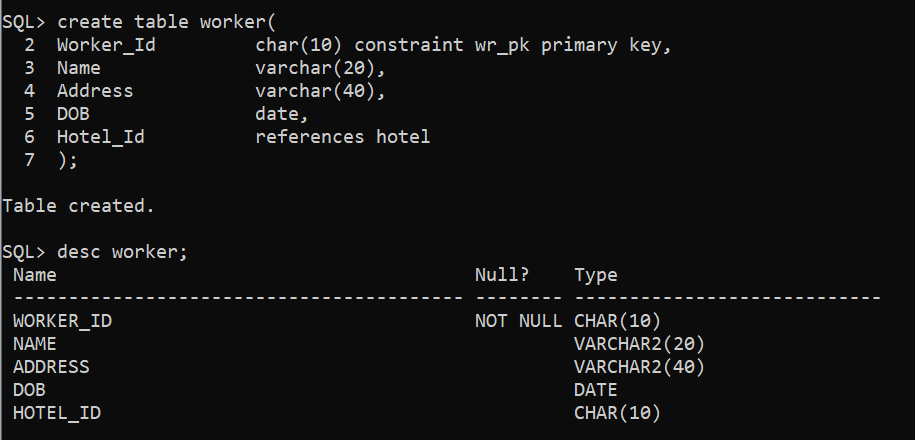
**Parking\_Area**



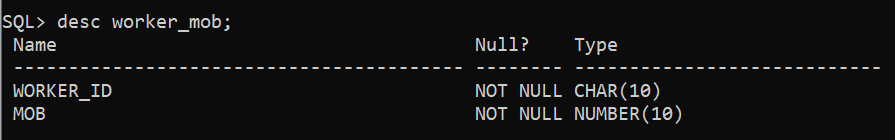
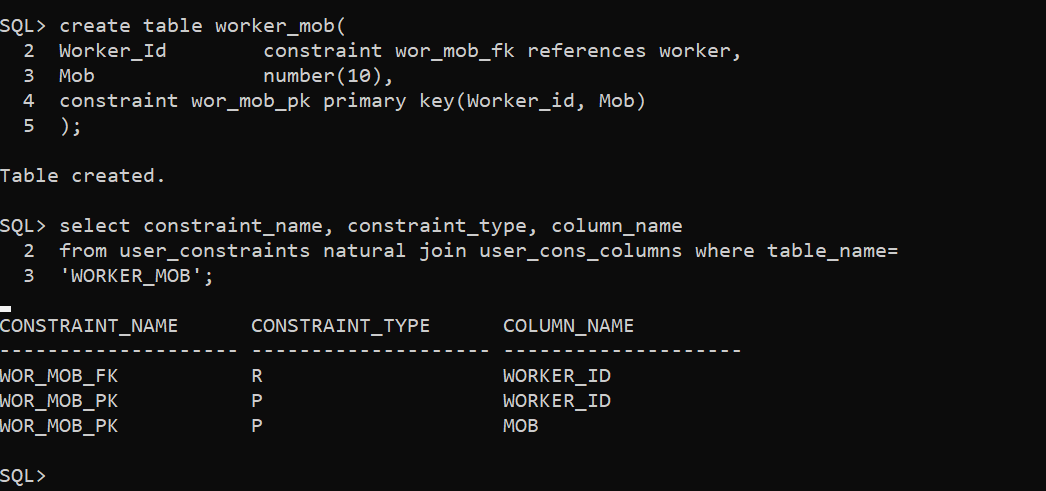
**Parks**



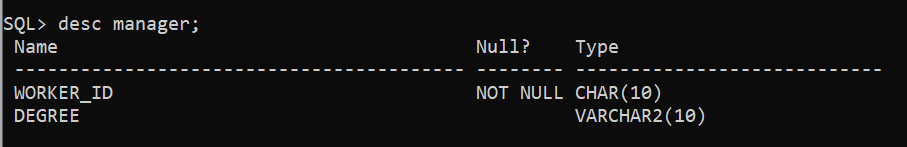
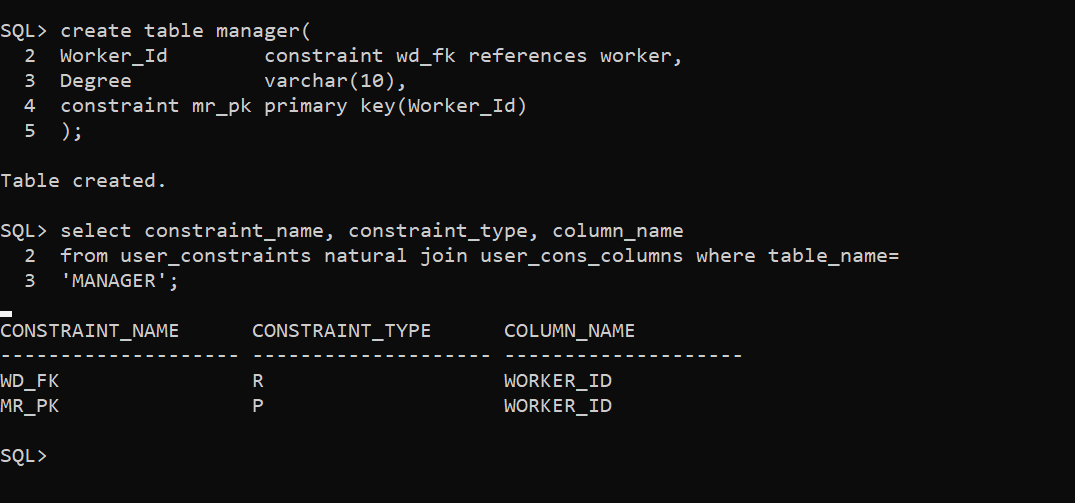
**Worker**



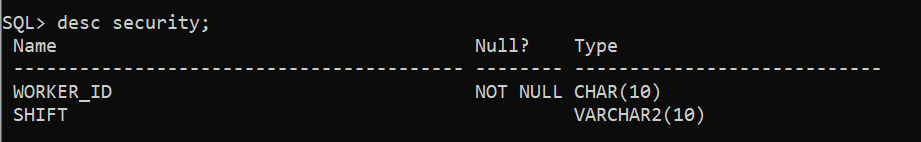
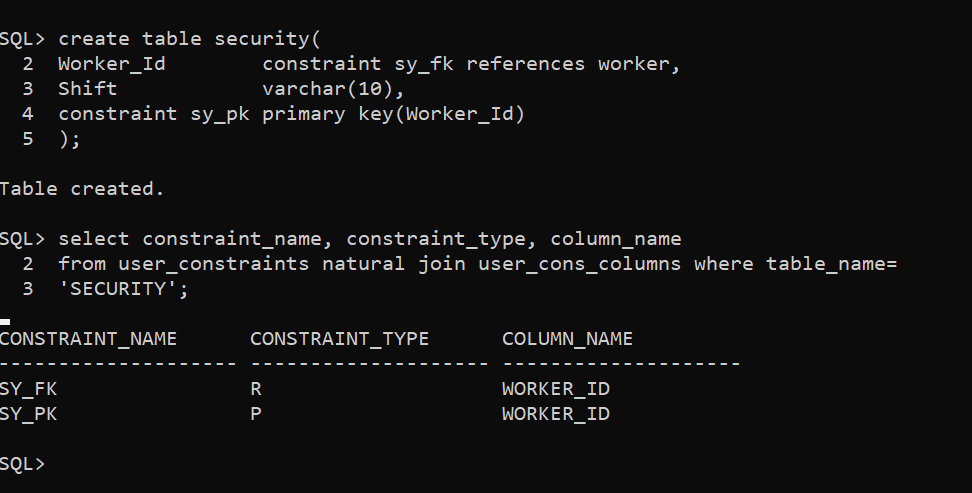
**Worker\_Mob**



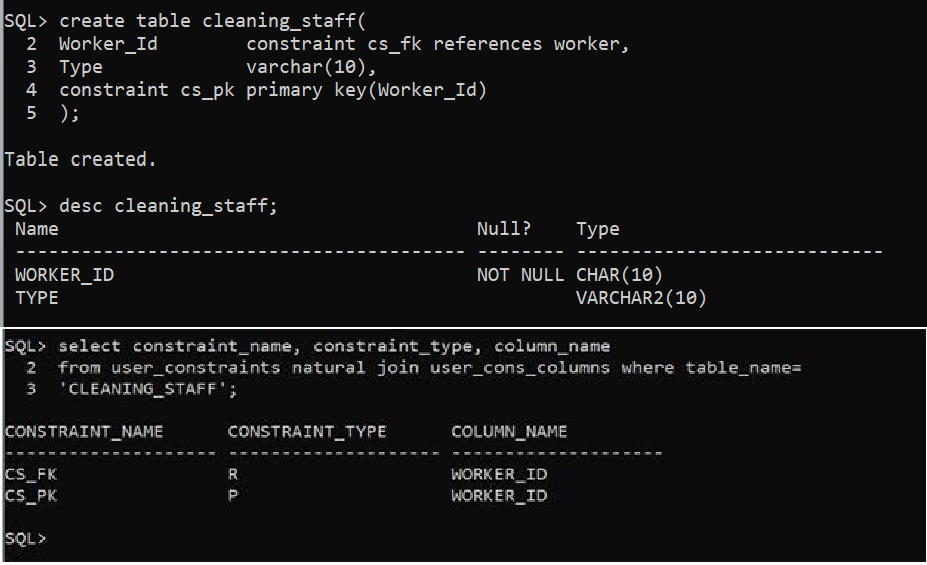
**Manager**



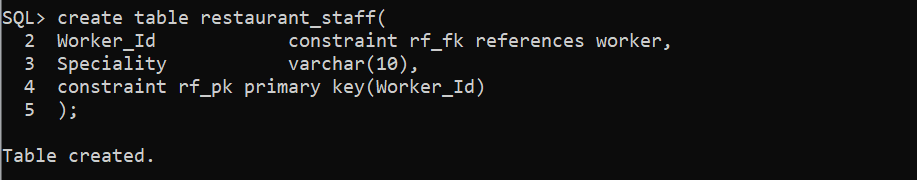
**Security**

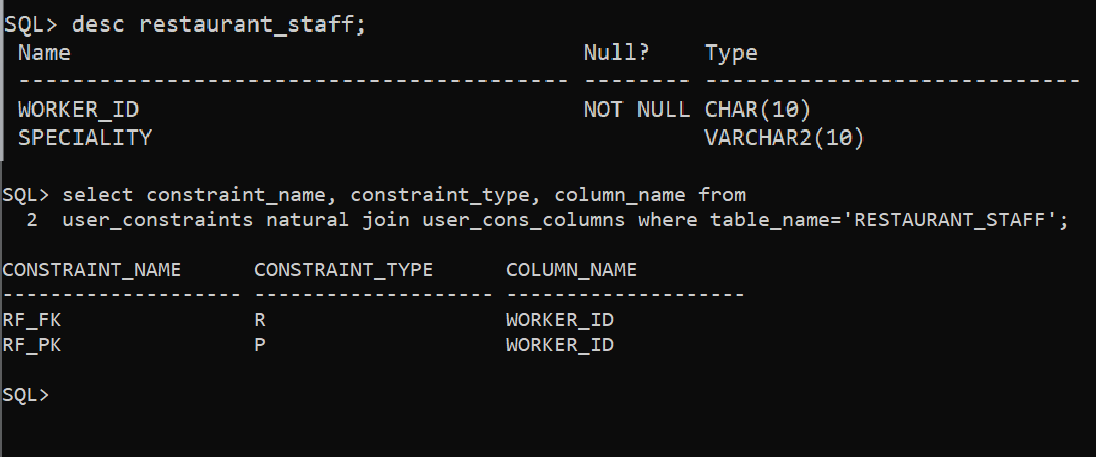


**Cleaning\_staff**



**Restaurant\_Staff**

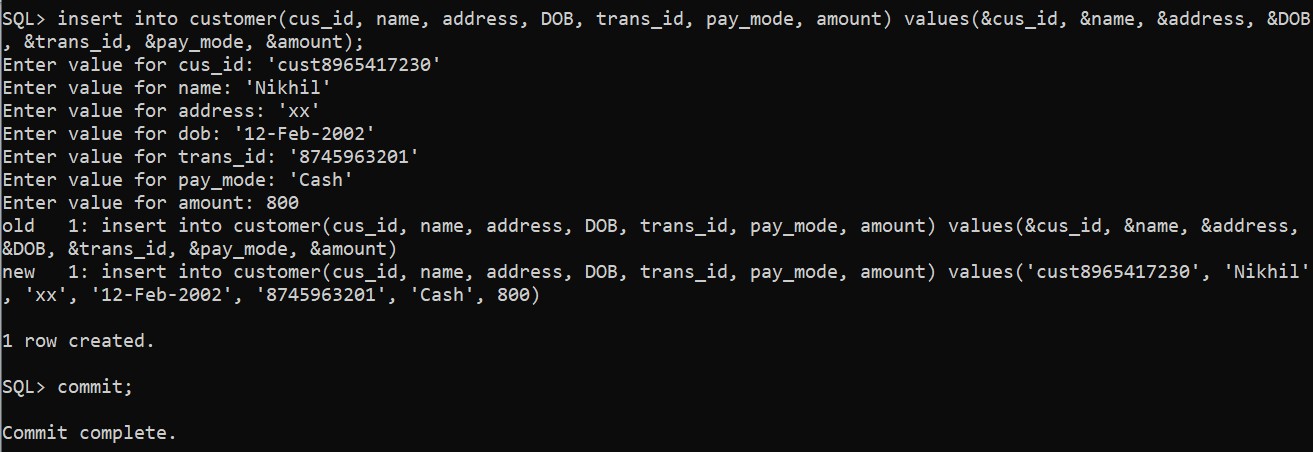


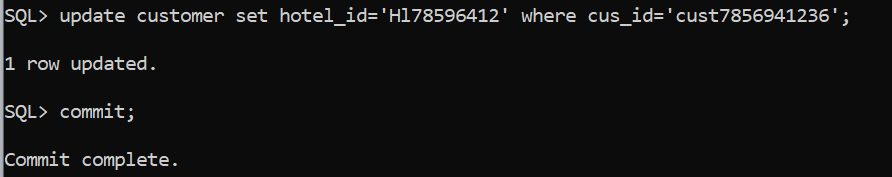
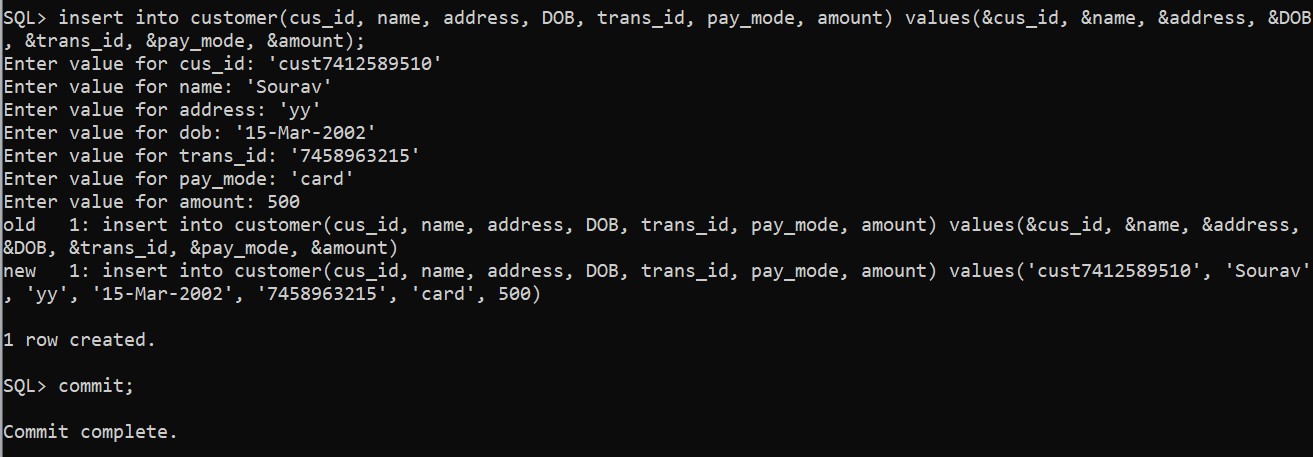


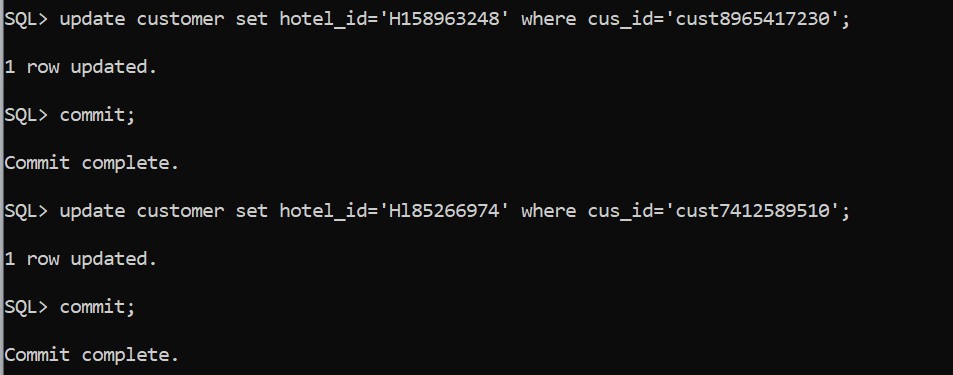
**Insertion of data in Customer table :-**

**Customer**



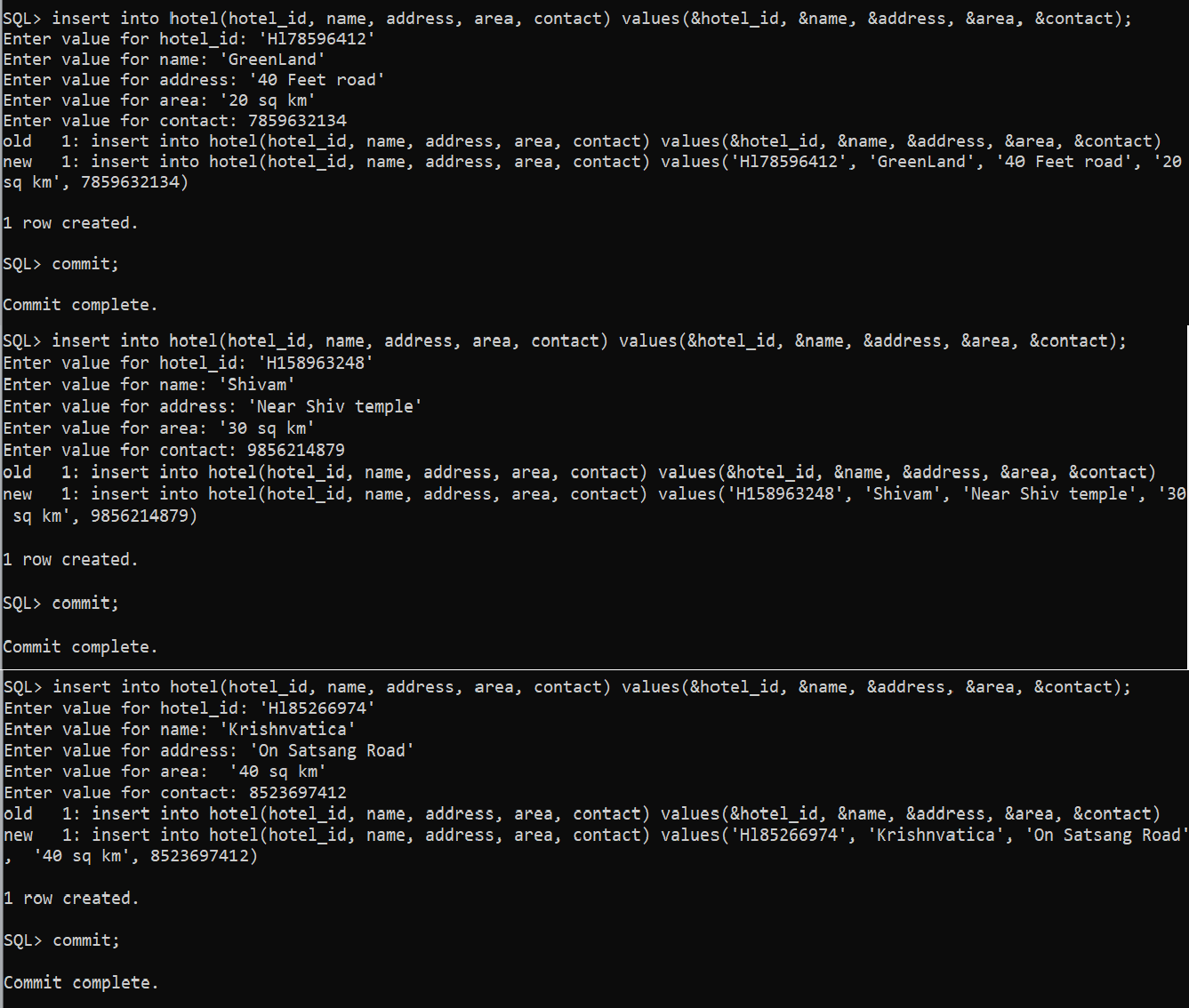






## Displaying the inserted data of Customer table

**Insertion of data in Hotel table :- Hotel**

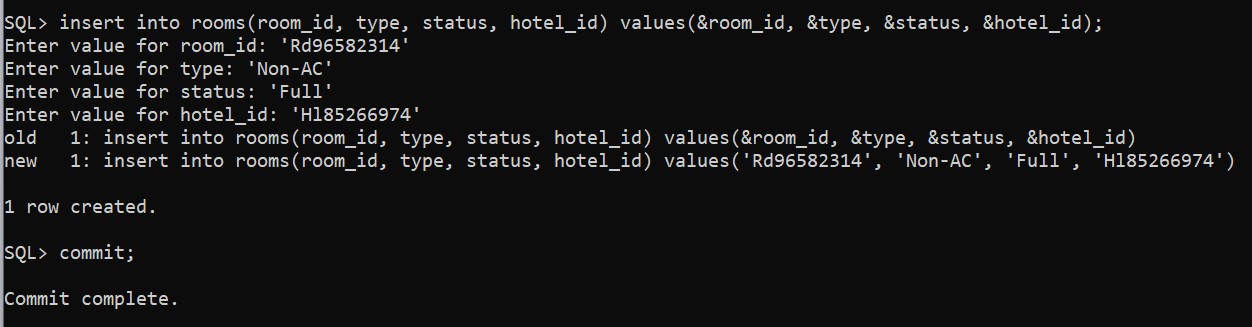


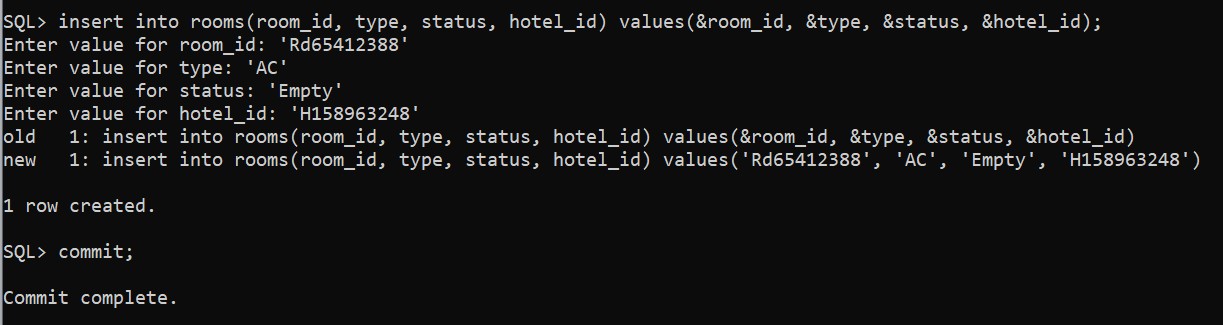
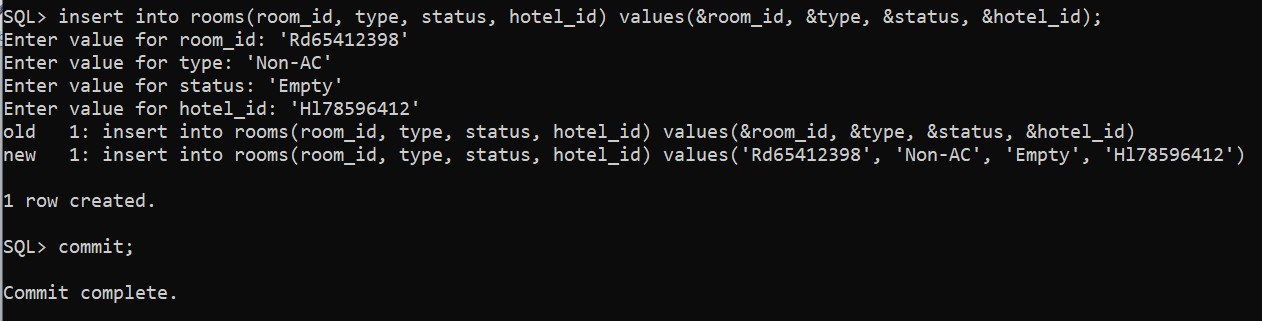
## Displaying the inserted data of Hotel table

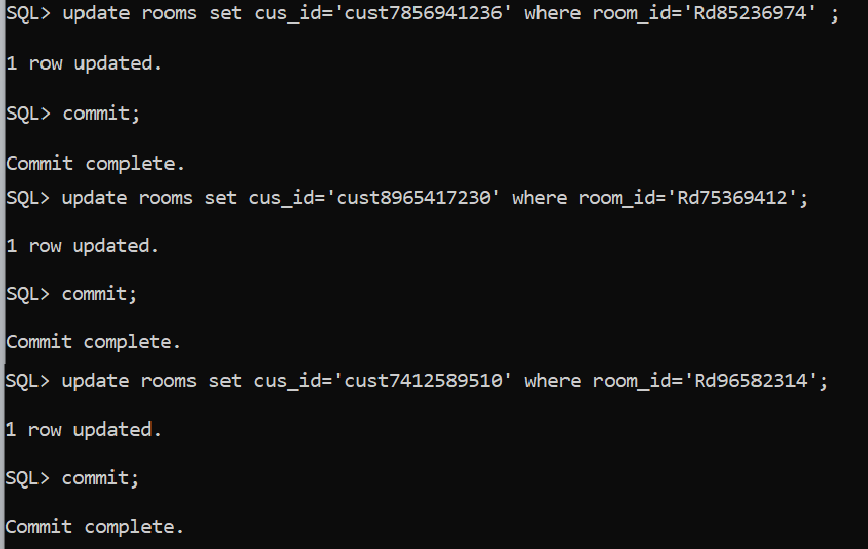
**Insertion of data in Rooms table :-**

**Rooms**

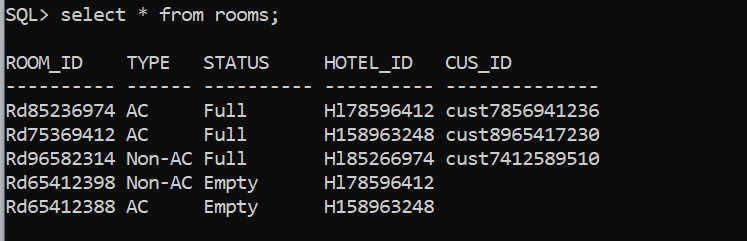






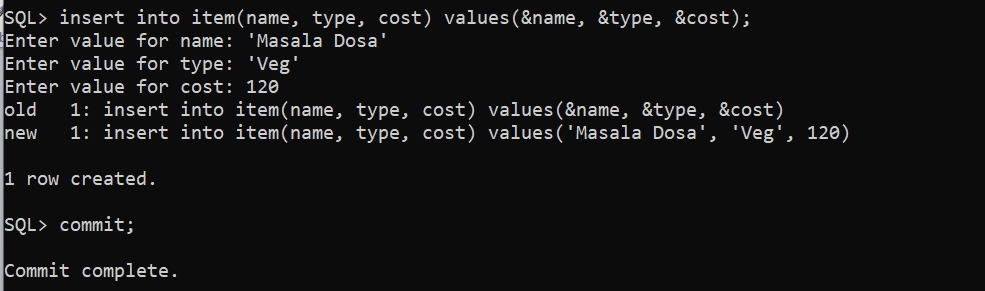
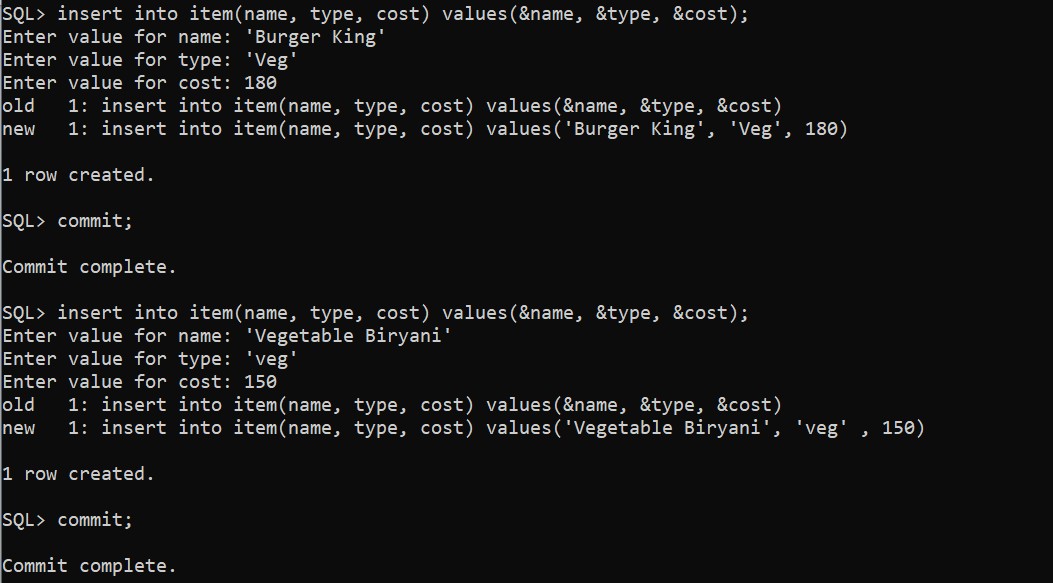


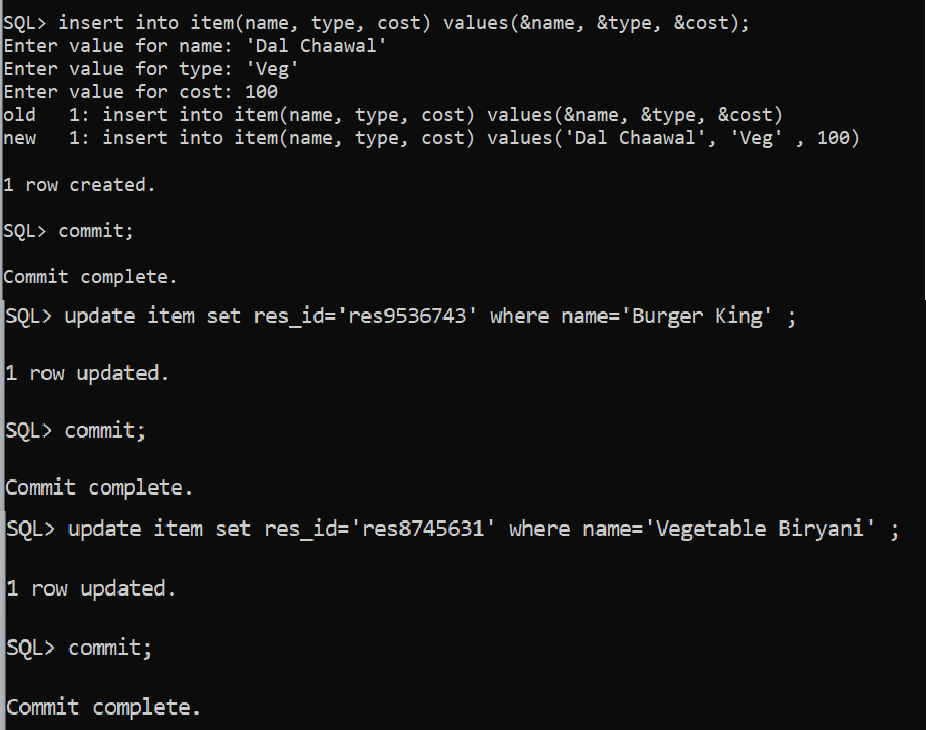
## Displaying the inserted data of Rooms table

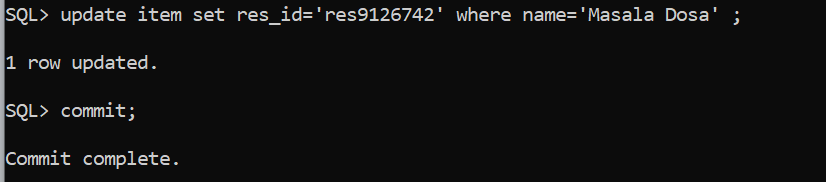


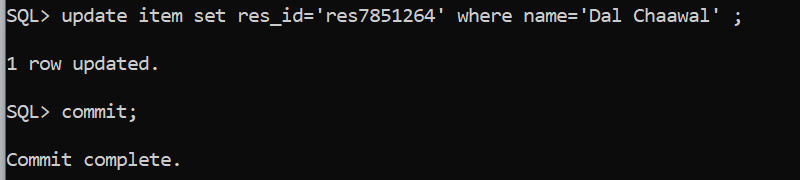
**Insertion of data in Item table :-**

**Item**







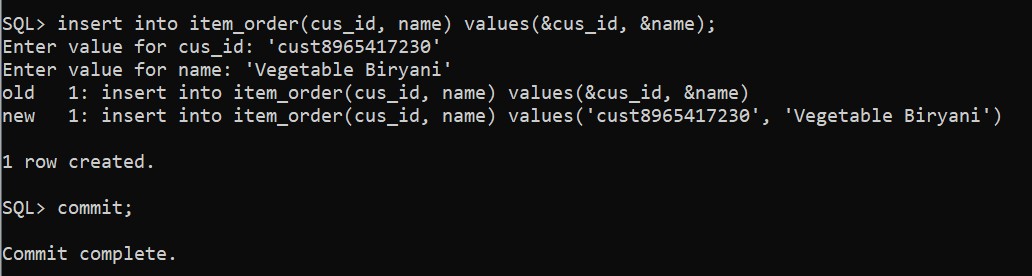


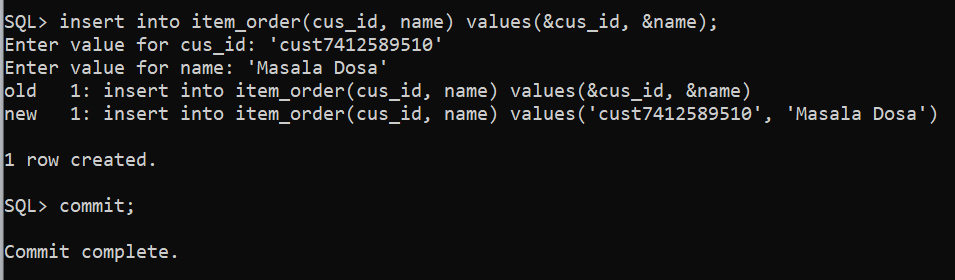
## Displaying the inserted data of Item table

**Insertion of data in Item\_order table :-**

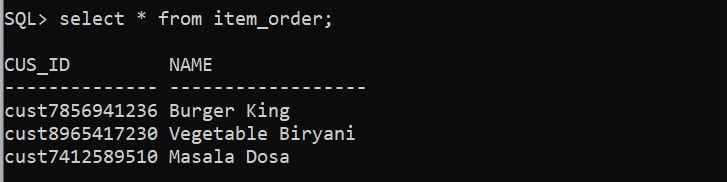
**Item\_order**





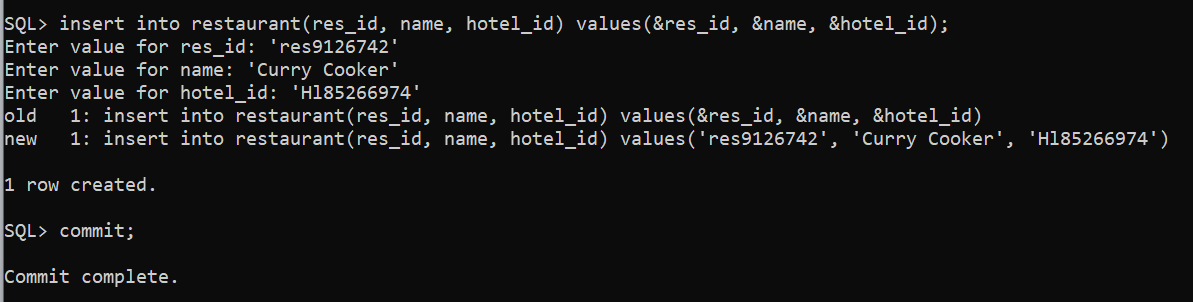
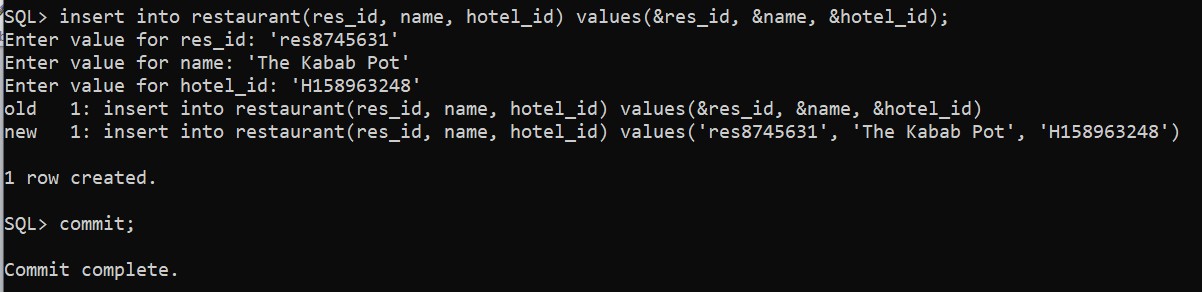


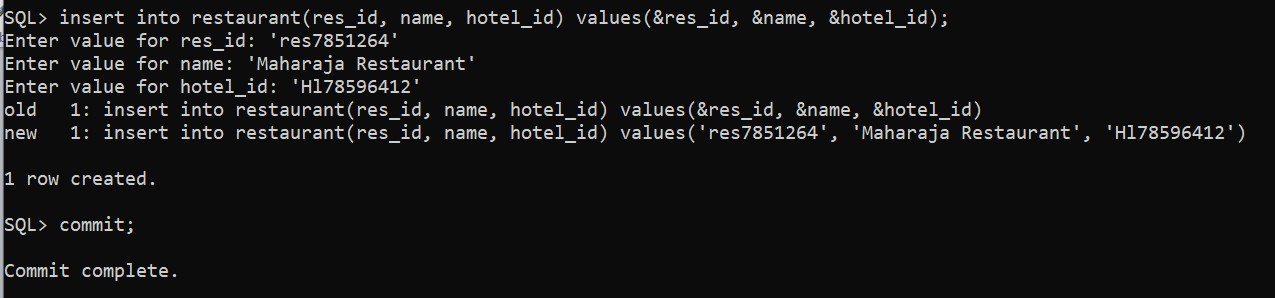
## Displaying the inserted data of Item\_order table

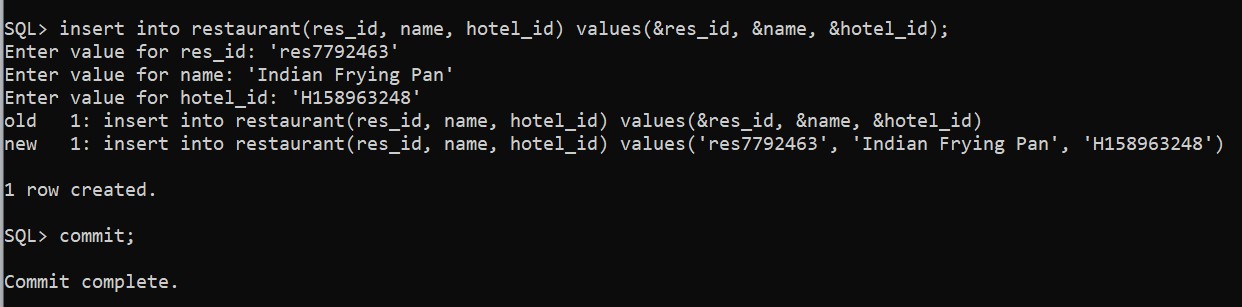


**Insertion of data in Restaurant table :-**

**Restaurant**





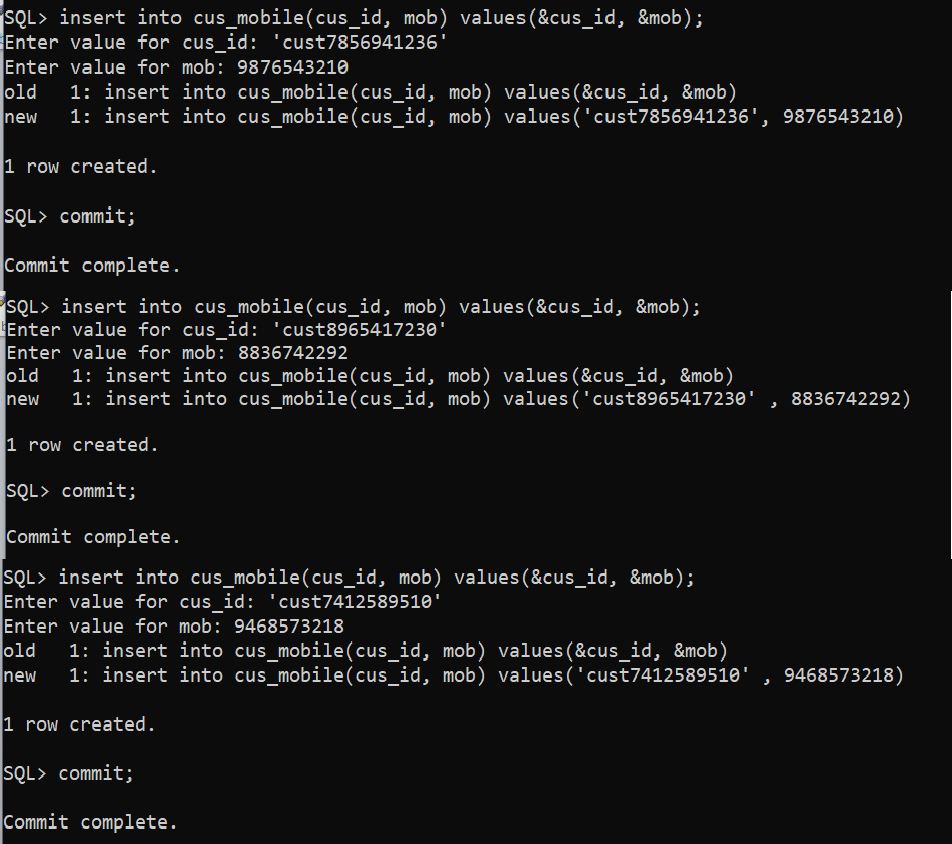


## Displaying the inserted data of Restaurant table



**Insertion of data in Cus\_mobile table :-**

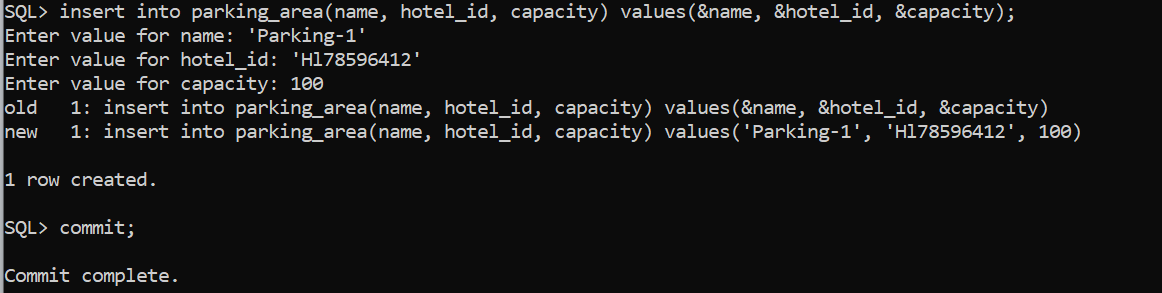
**Cus\_mobile**



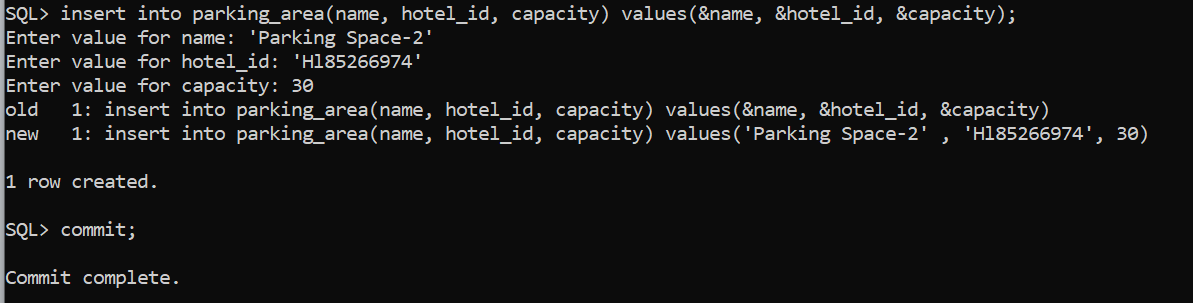
## Displaying the inserted data of Cus\_mobile table

**Insertion of data in Parking\_area table :-**

**Parking\_area**

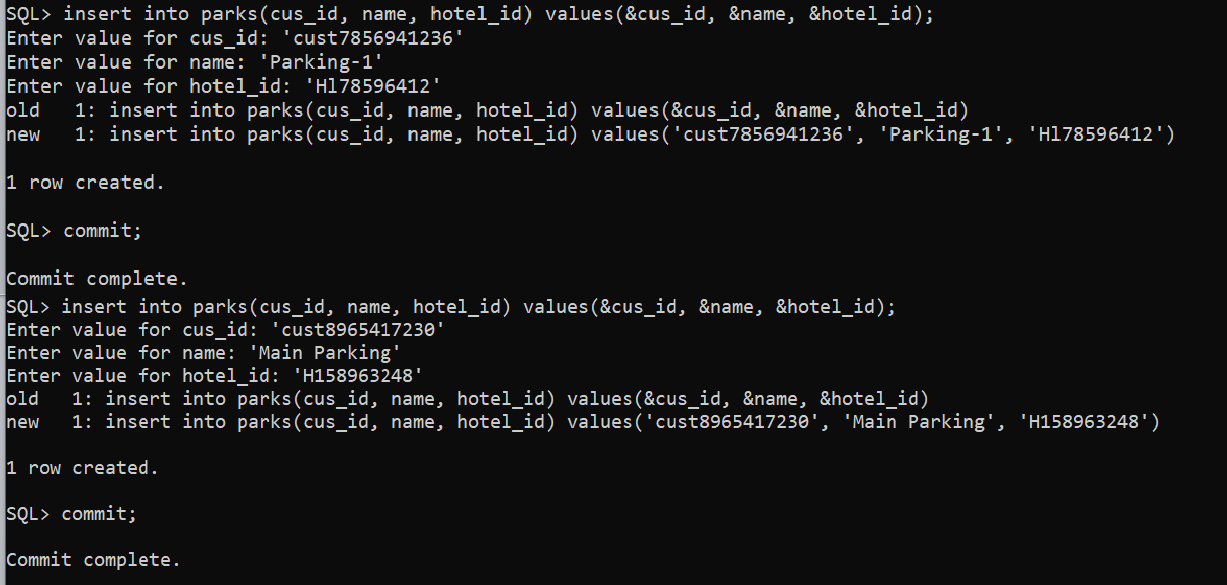


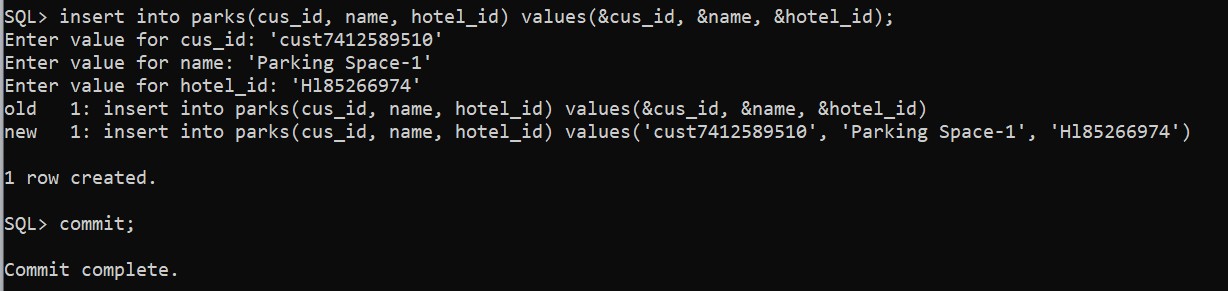




## Displaying the inserted data of Parking\_area table

**Insertion of data in Parks table :-**

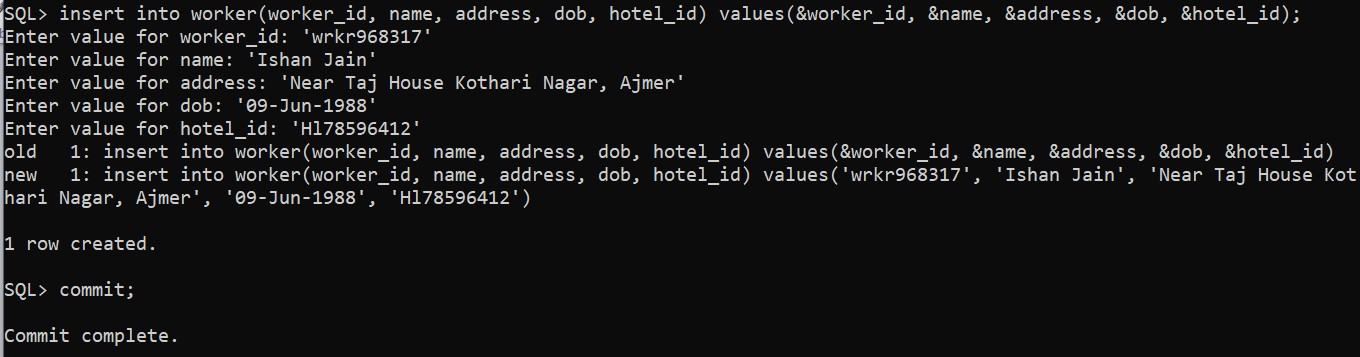


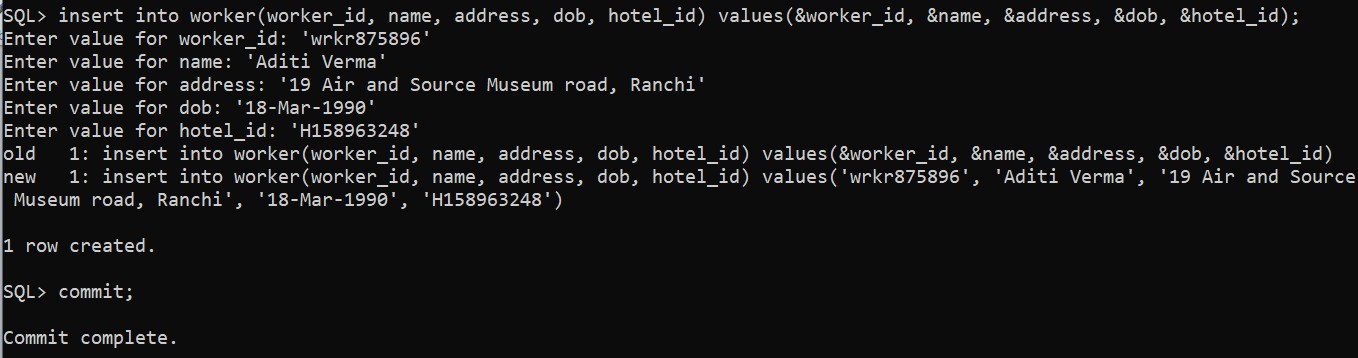


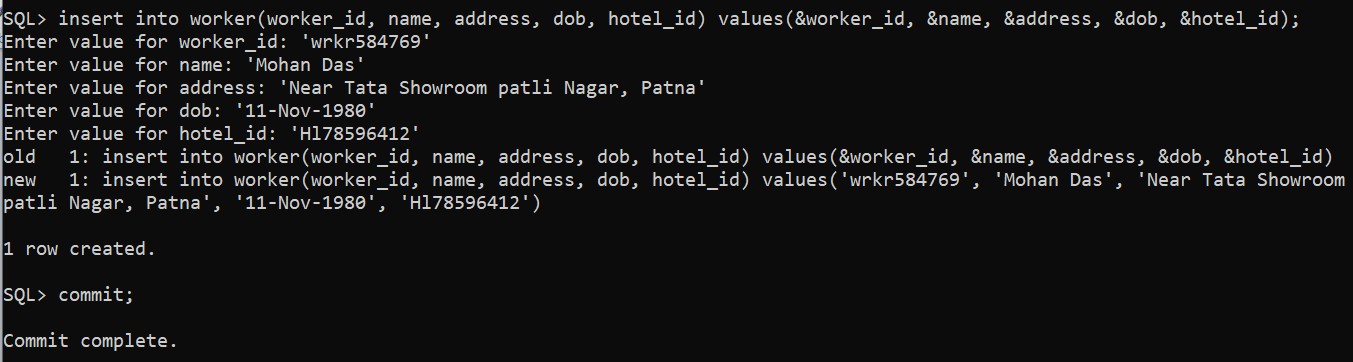
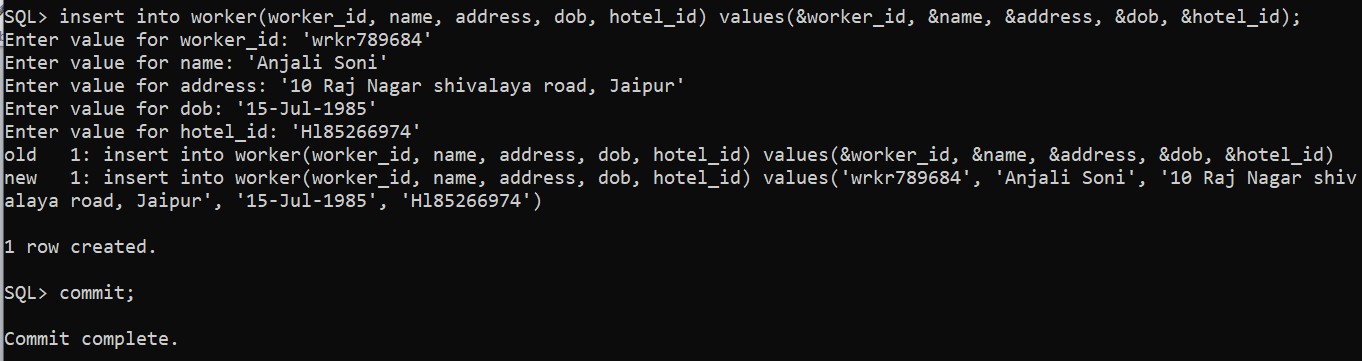
## Displaying the inserted data of Parks table

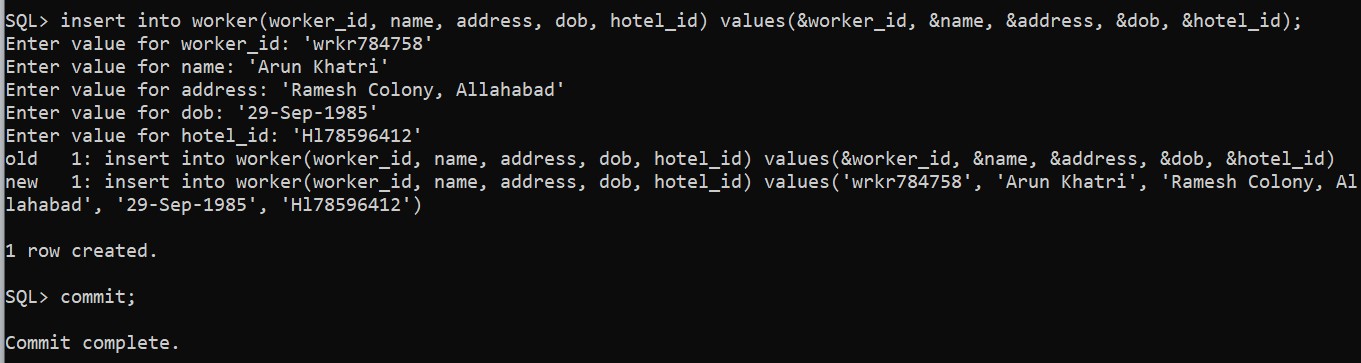
**Insertion of data in Worker table :-**

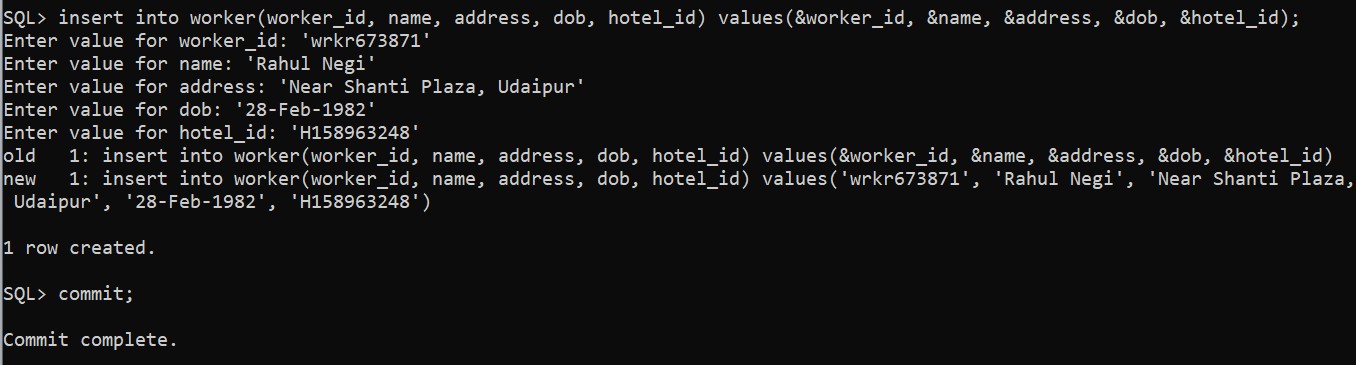
**Worker**





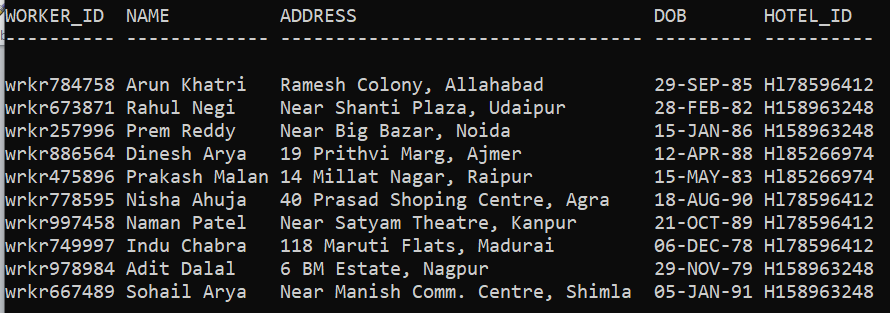


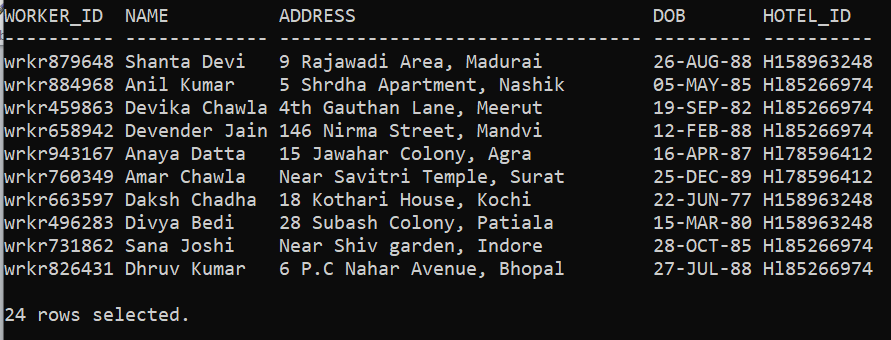




## Displaying the inserted data of Worker table

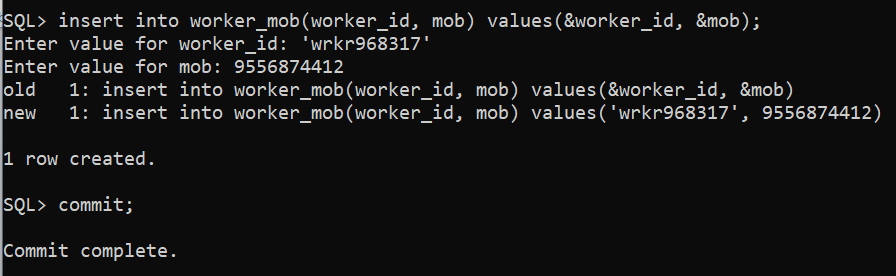


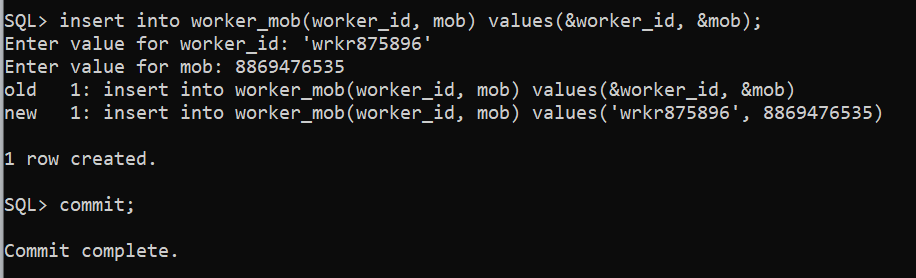


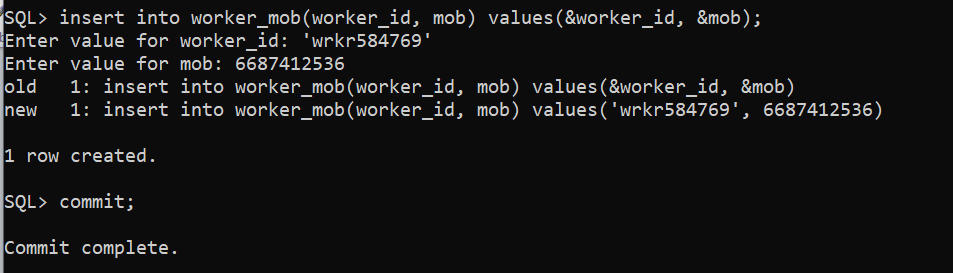


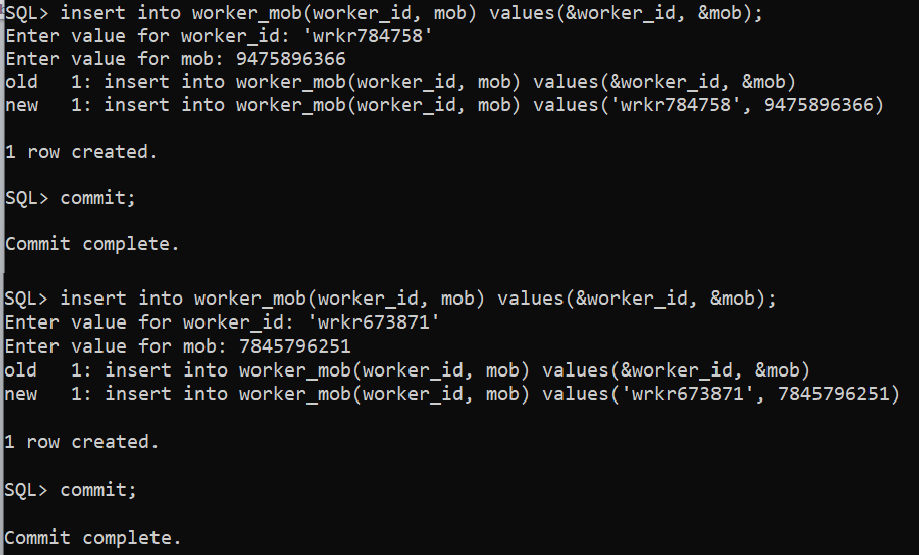
**Insertion of data in Worker\_mob table :-**

**Worker\_mob**

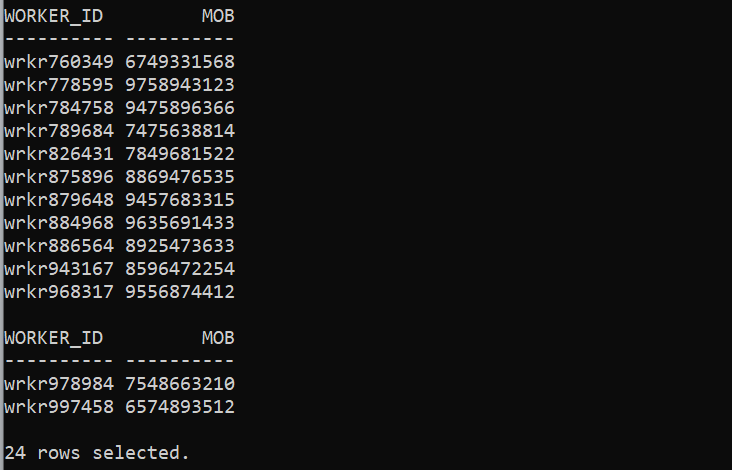
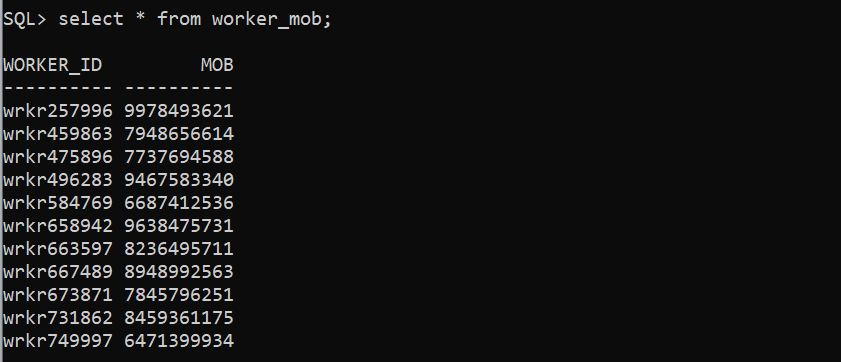






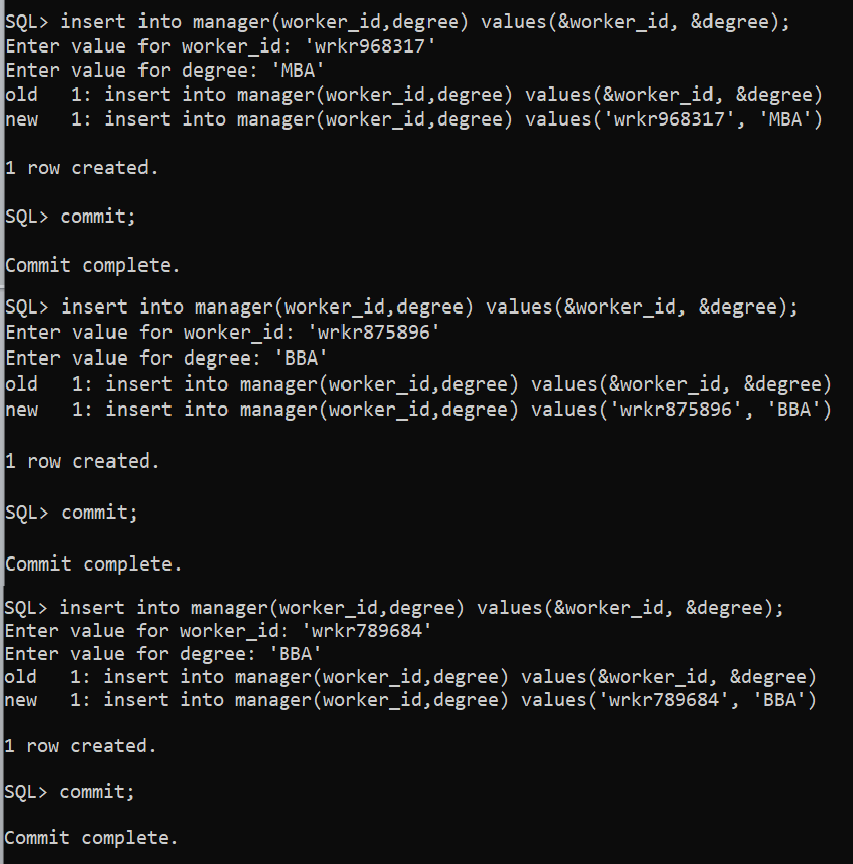


**Displaying the inserted data of Worker\_mob table**

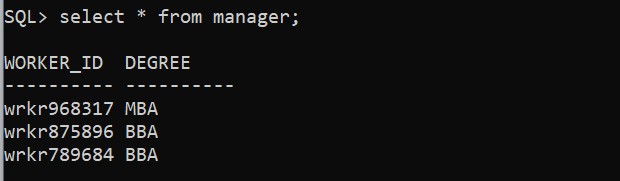


## Insertion of data in Manager table :-

**Manager**

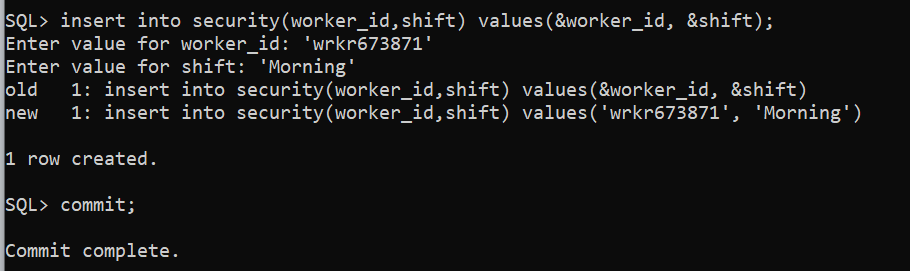


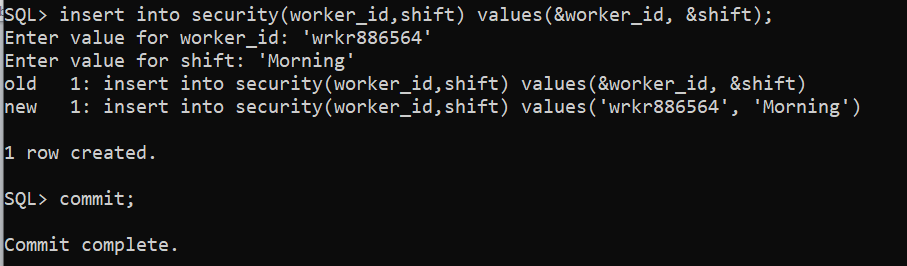
**Displaying the inserted data of Manager table**

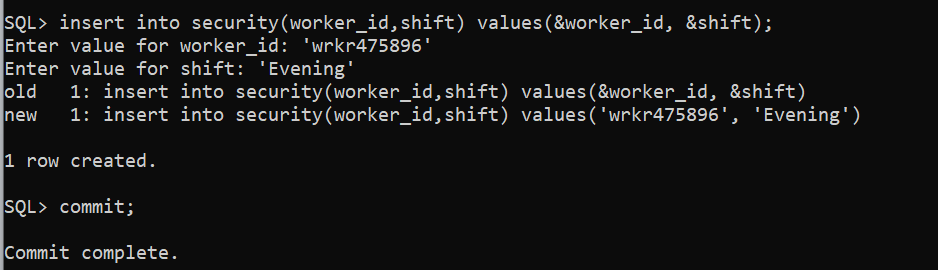


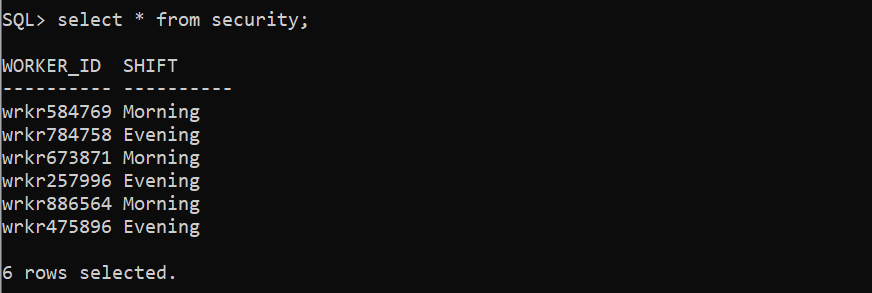
**Security**

## Insertion of data in Security table :-



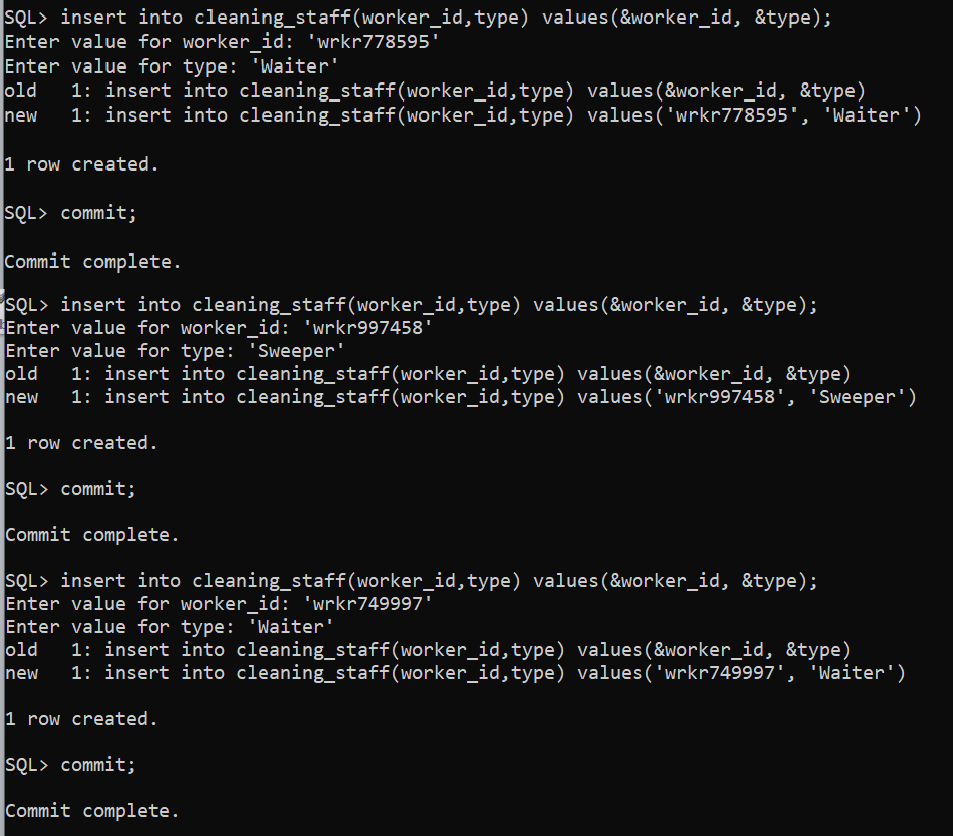


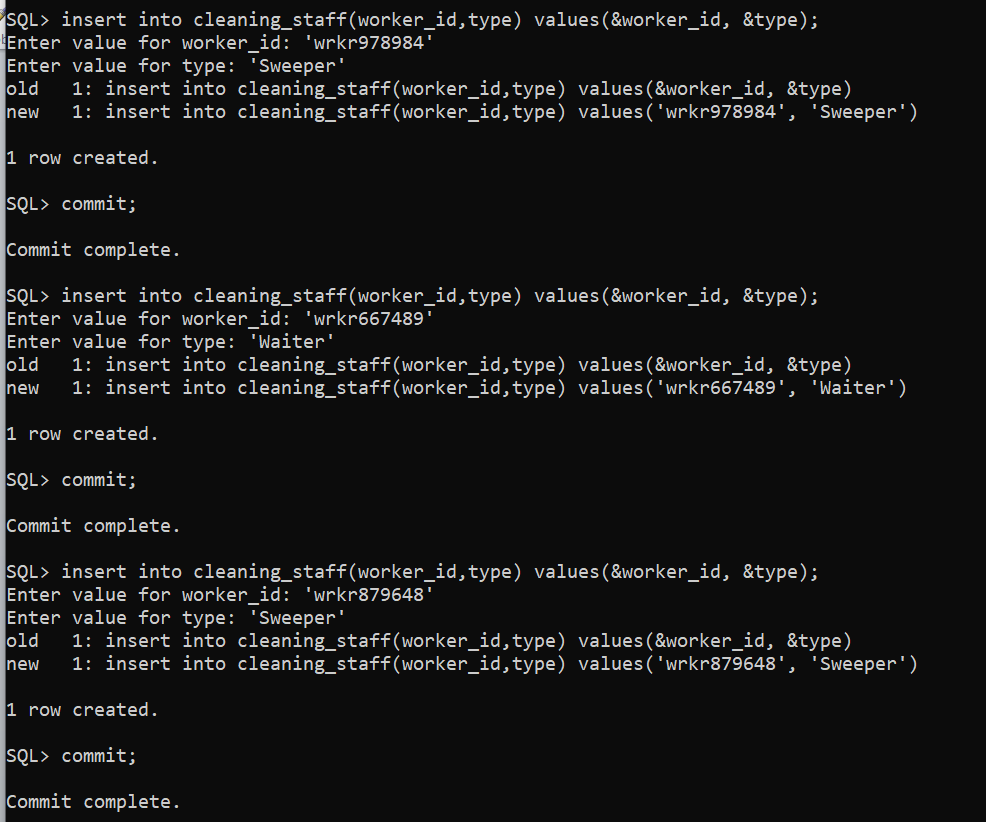


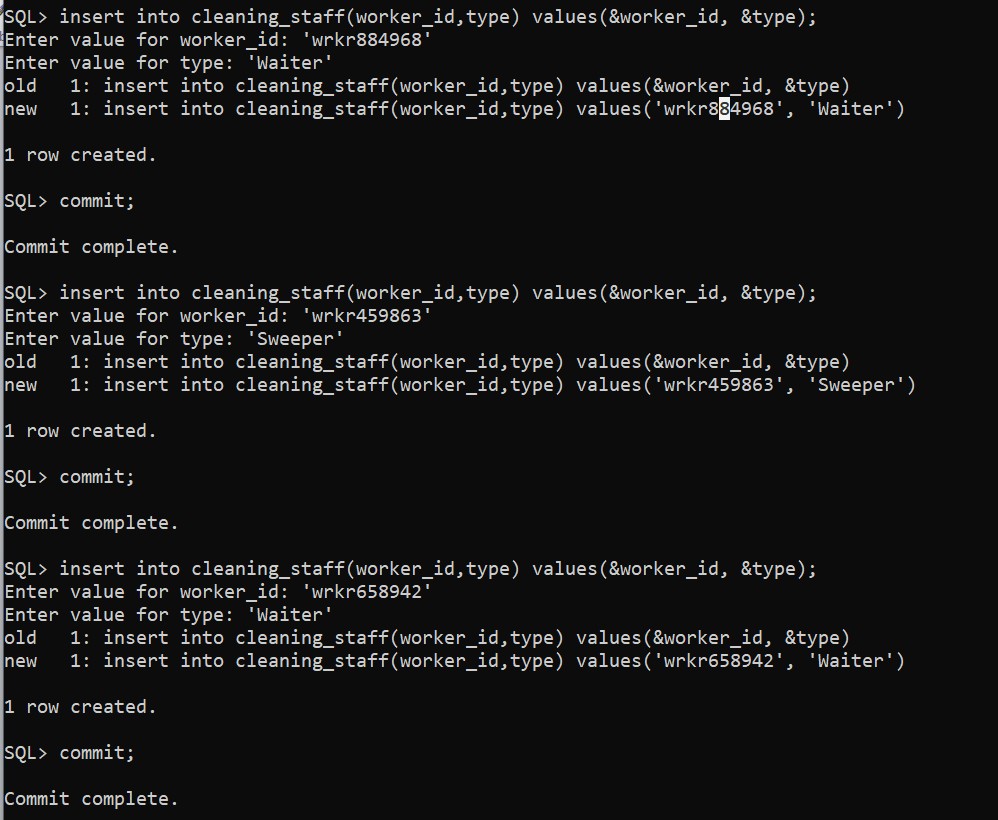
**Displaying the inserted data of Security table**

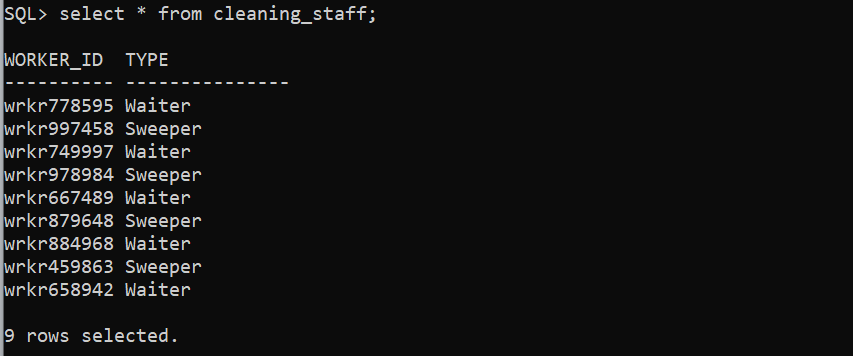
## Insertion of data in Cleaning\_staff table :-

**Cleaning\_staff**



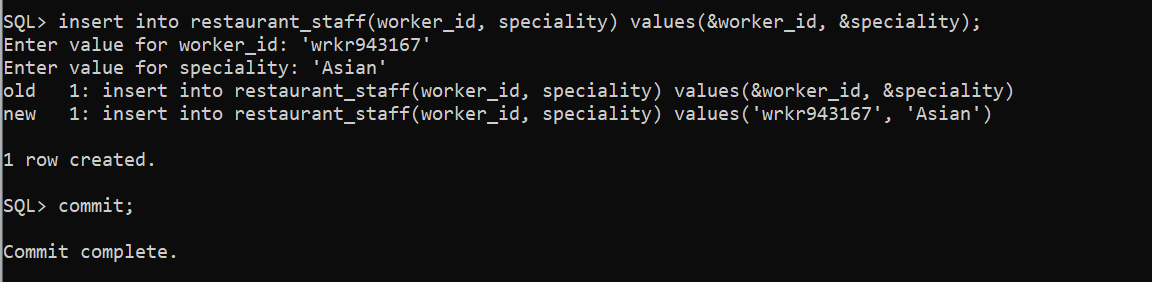


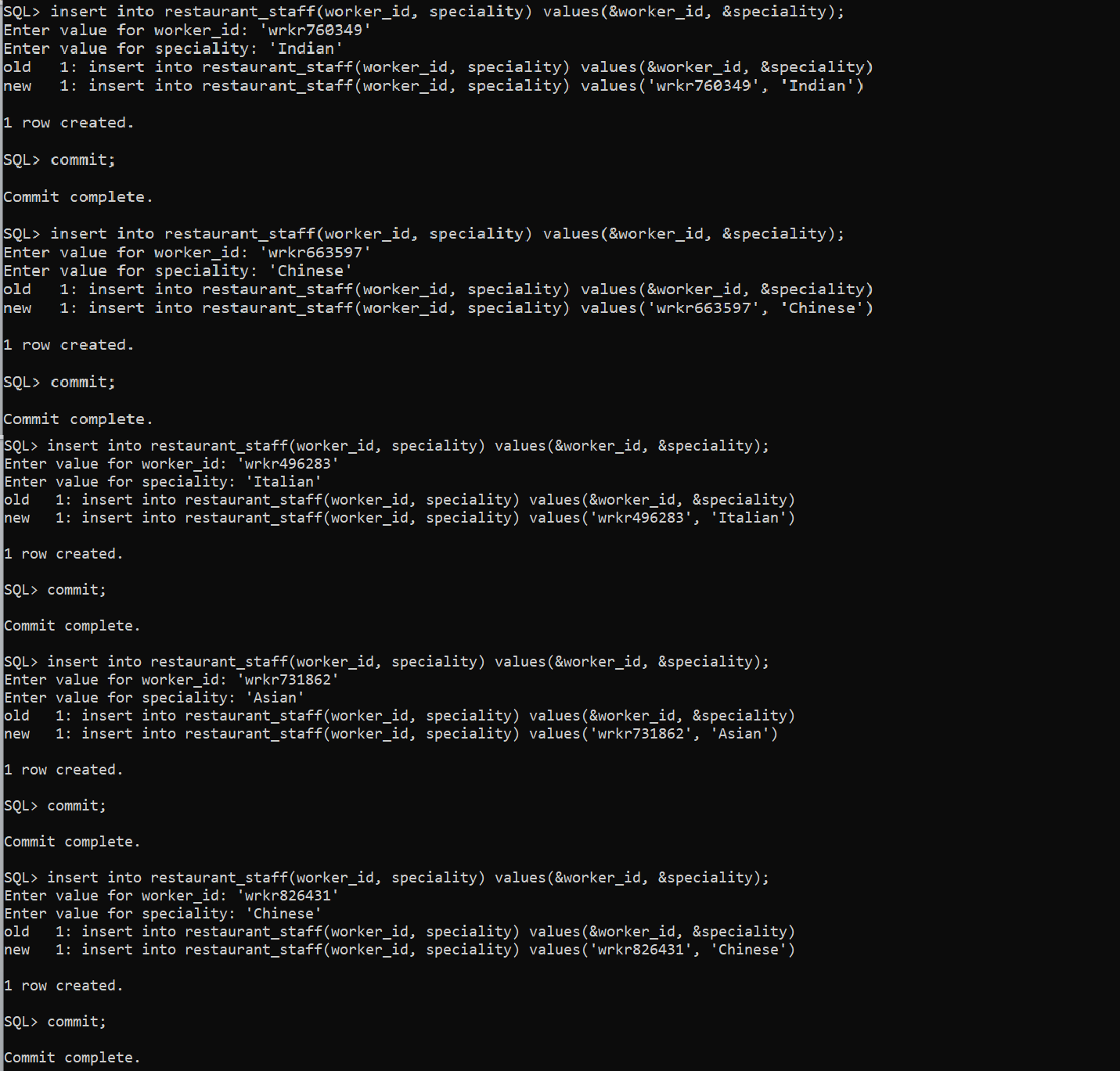


**Displaying the inserted data of Cleaning\_staff table**

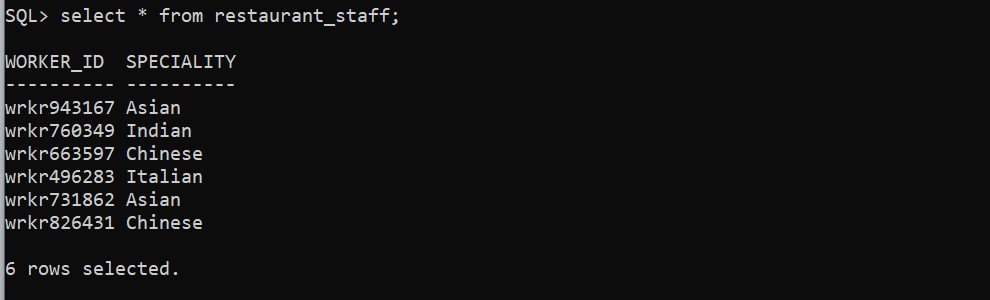
## Insertion of data in Restaurant\_staff table :-

**Restaurant\_staff**



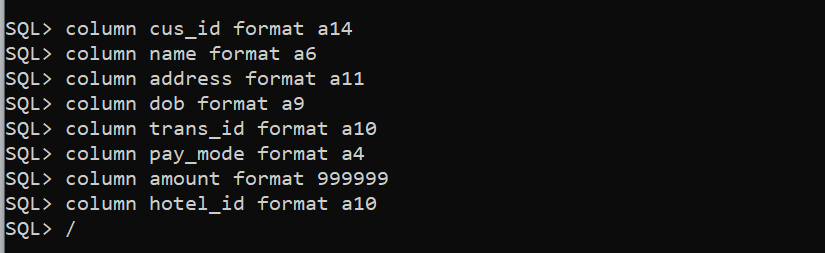


**Displaying the inserted data of Restaurant\_staff table**

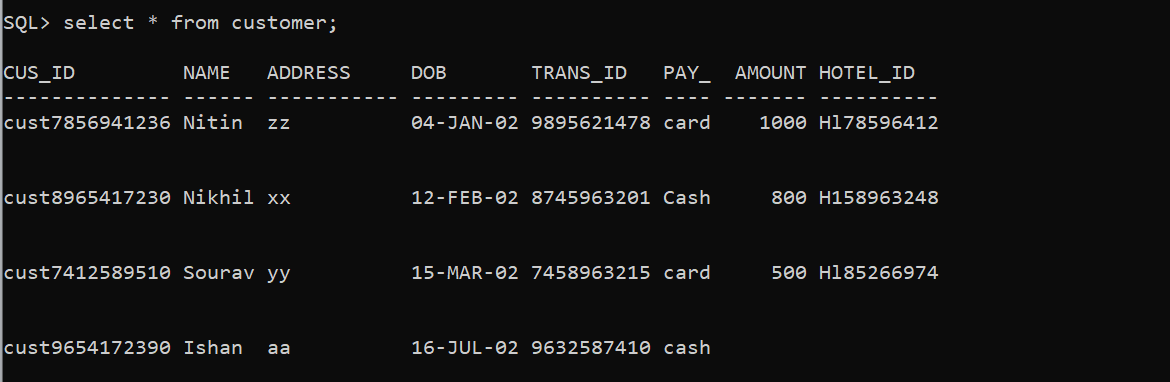


**Q6.** Write down the necessary SQL statements for implementation of functional requirements. There should be at least four different scenarios of removal of old data, four different scenarios for modification of existing data and eight different scenarios for data retrieval through SQL delete, update and select statement respectively.

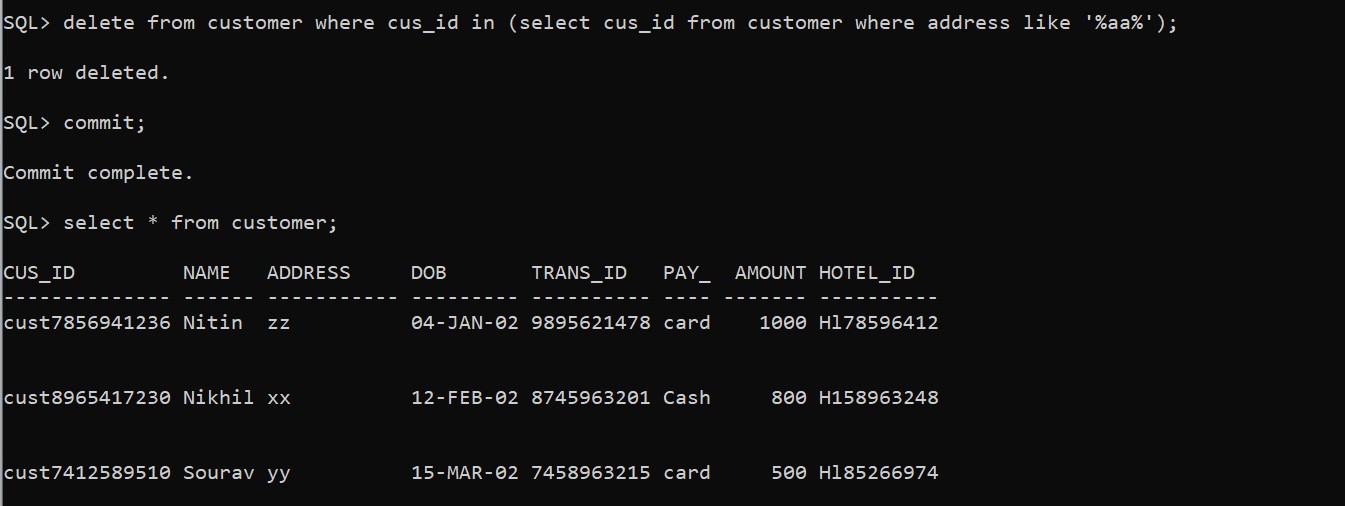
### Delete Statements :-

1. Delete the data of the customer if he/she has ‘aa’ in their address.

Before deleting the data :-



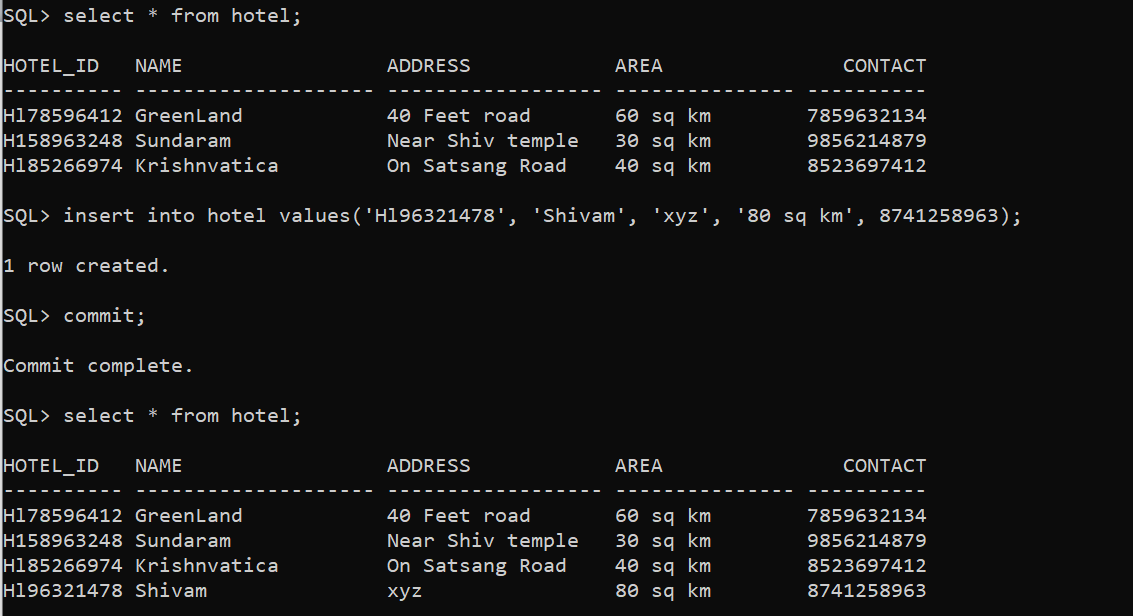
After deleting the data :-



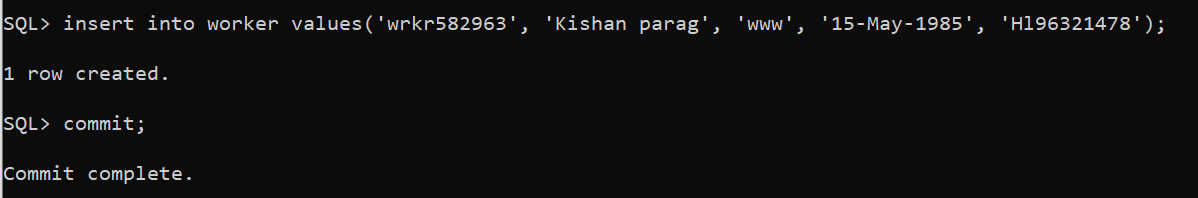
1. Delete the details of the worker who are working in hotel\_id='Hl96321478'.

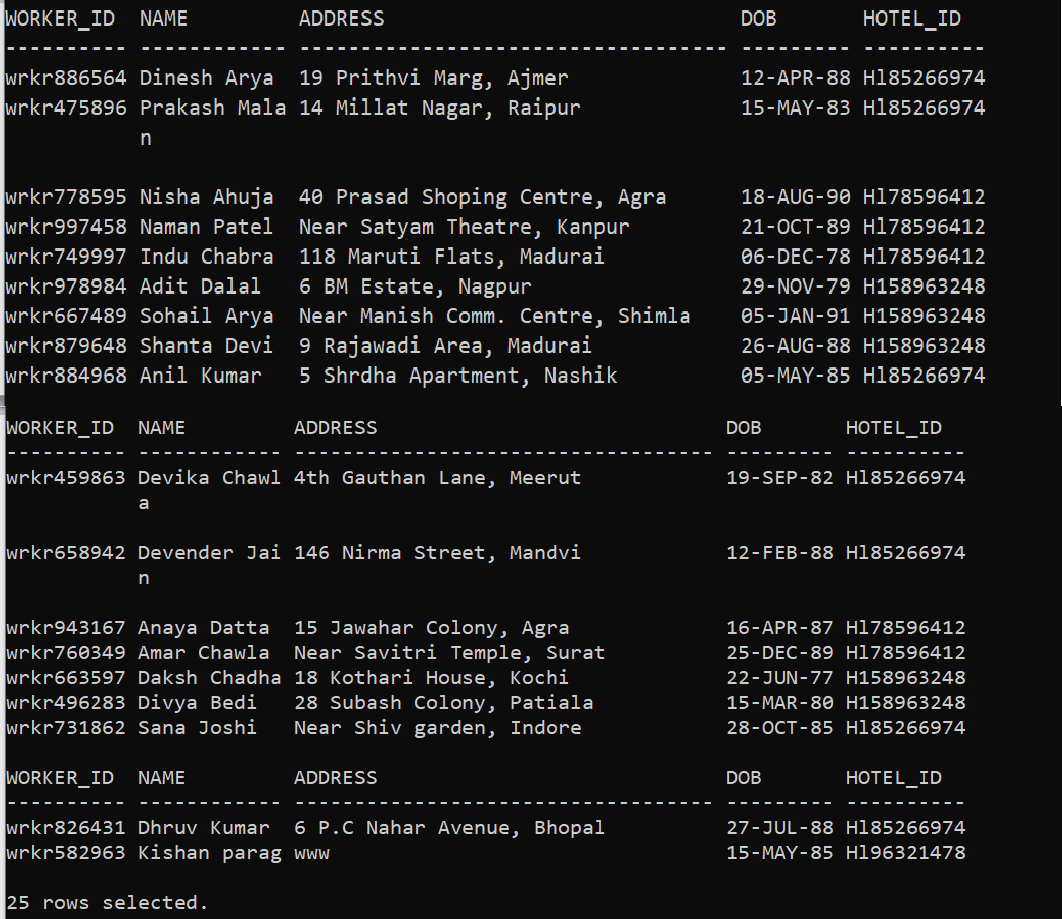
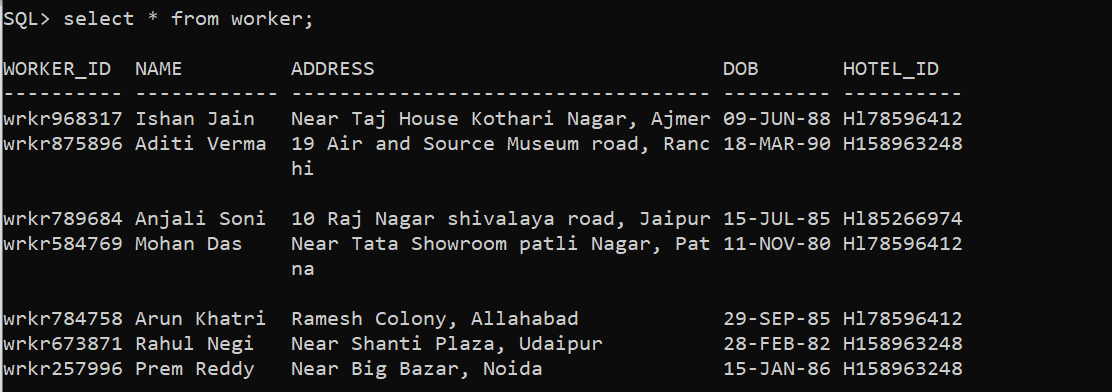
Before deleting the data :-

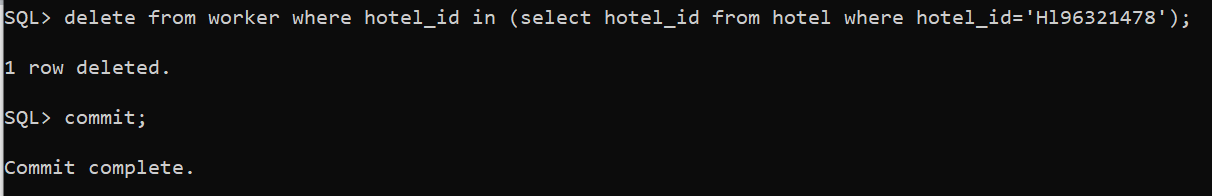
We have no data of that hotel\_id so we are inserting the details of that particular hotel to show the result of the query.



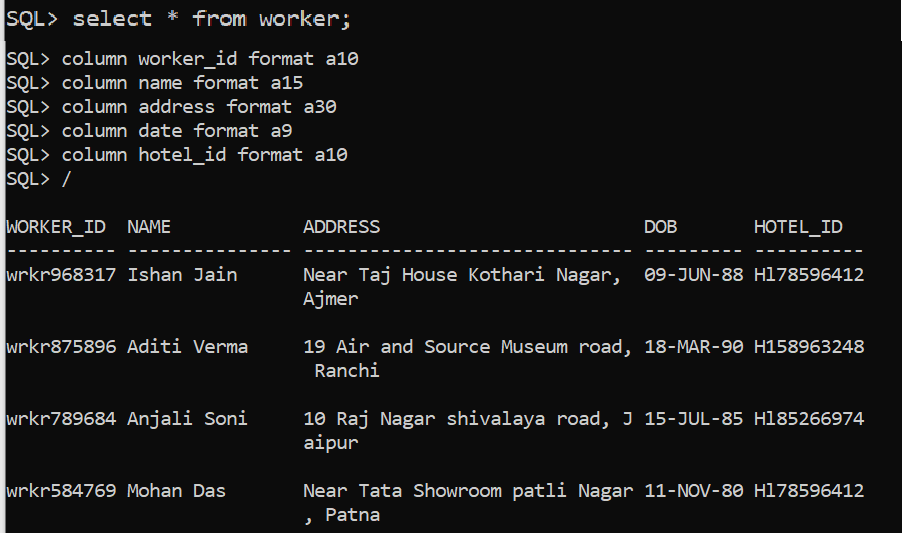
We have no worker data of that hotel\_id so we are inserting the details of worker to show the result of the query.

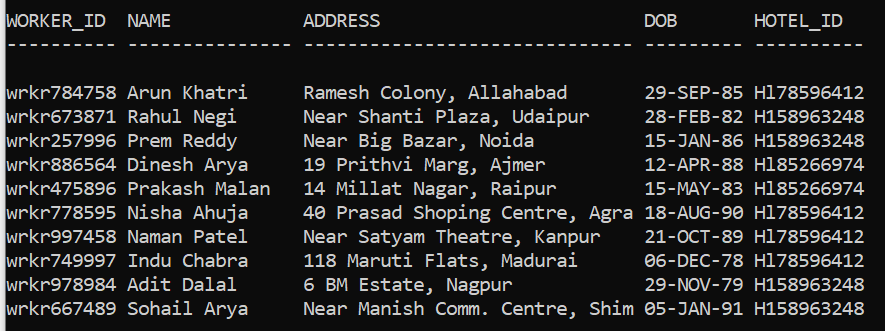


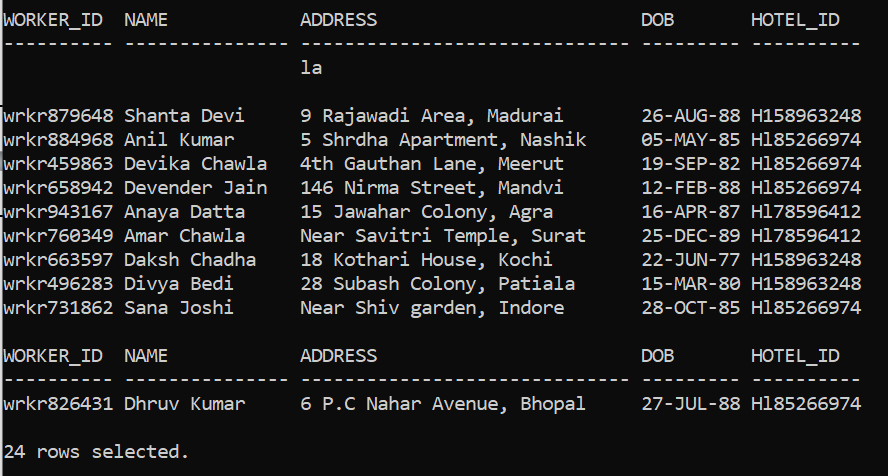




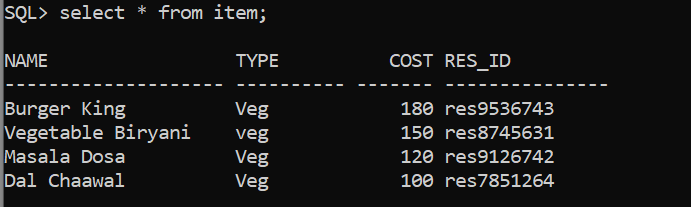
After deleting the data of the worker in that particular hotel\_id.



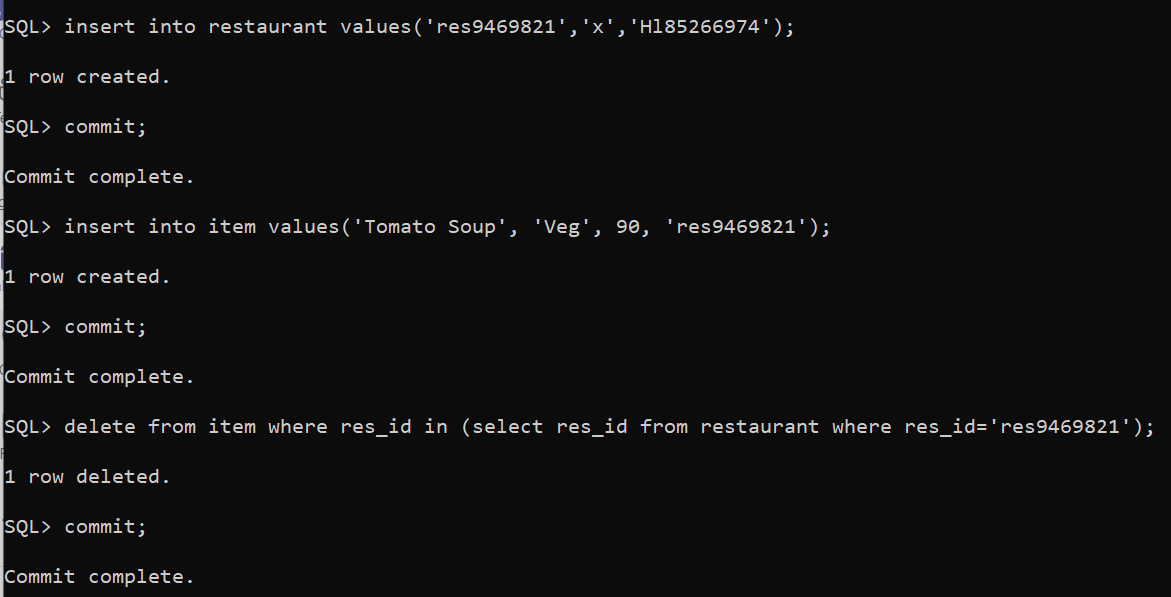




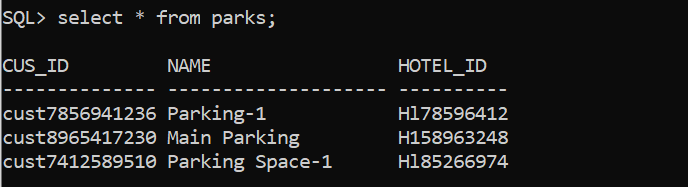
1. Delete the details of item from the menu whose name is ‘Tomato Soup’. Before deleting the data :-



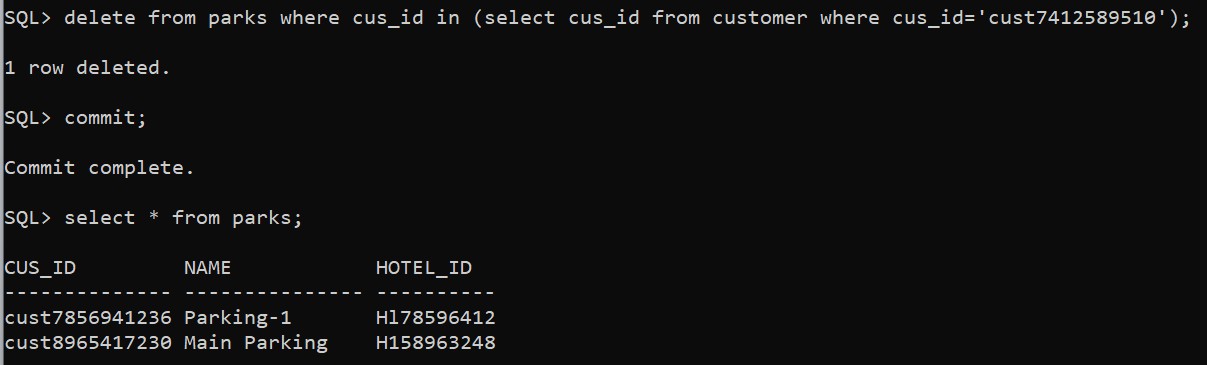
After deleting the data :-



1. Delete the data of the parking area where the customer whose cus\_id is ‘cust7412589510’ . Before deleting the data :-

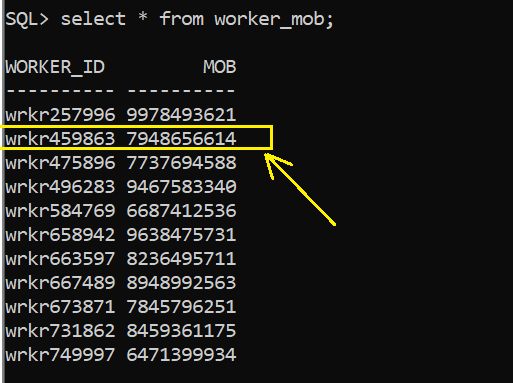
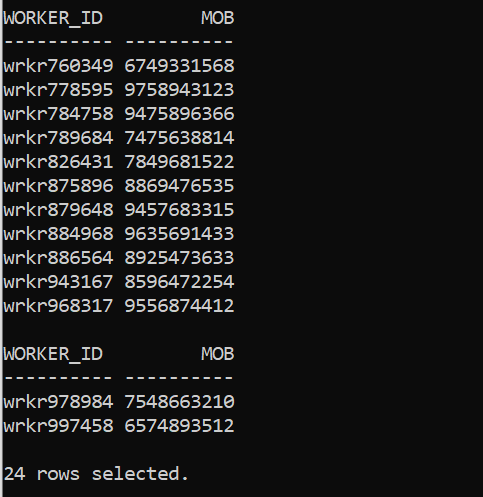


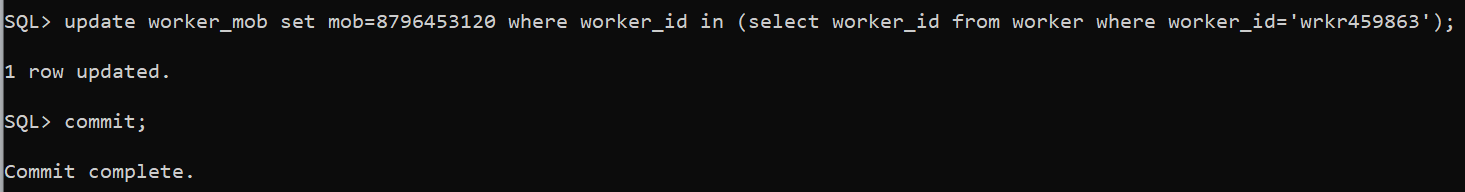
After deleting the data :-



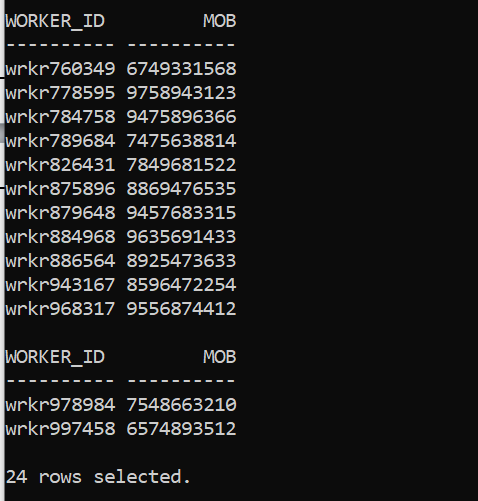
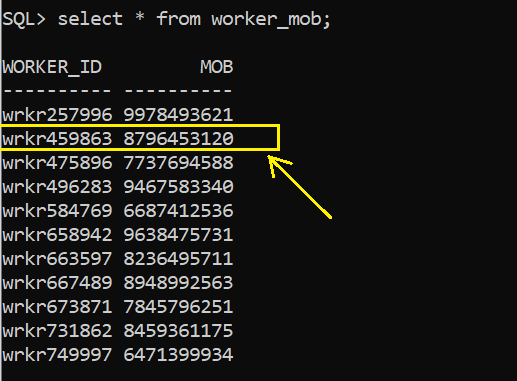
### Updation Statements :-

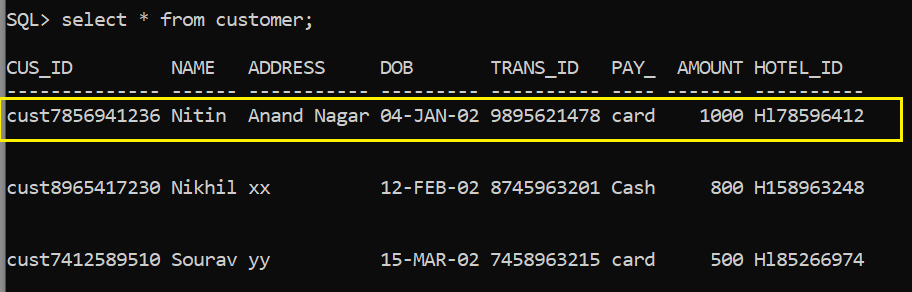
1. Update the Mobile No. of the worker if he/she changed their mobile number. Before updating the data :-

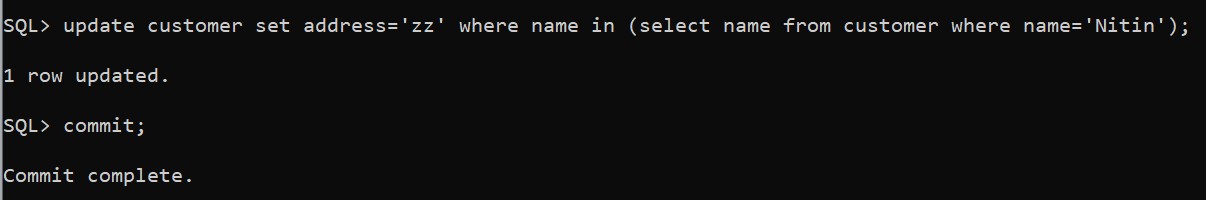
 



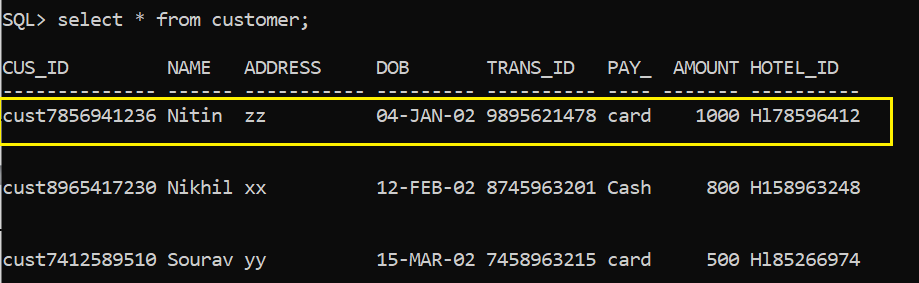
After updating the data :-



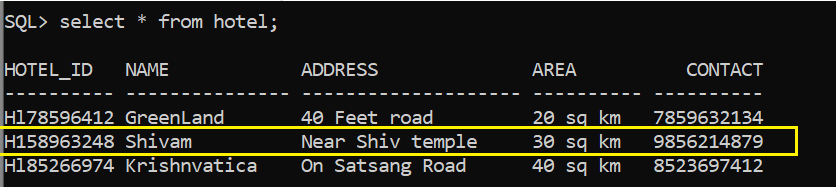
1. Update the customer details if something entered incorrect during the insertion of data. (If address is wrong) Before updating the data :-

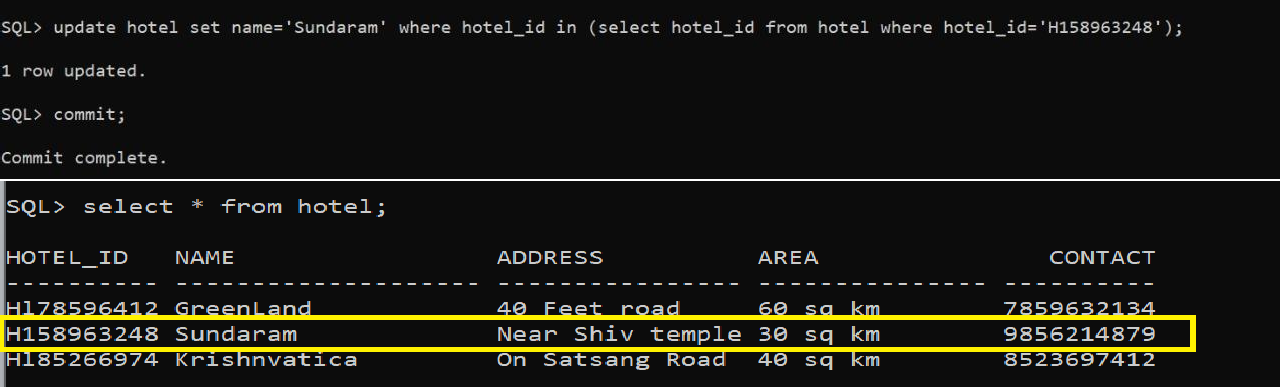


After updating the data :-

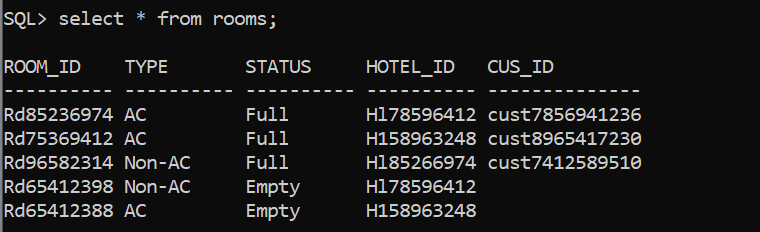


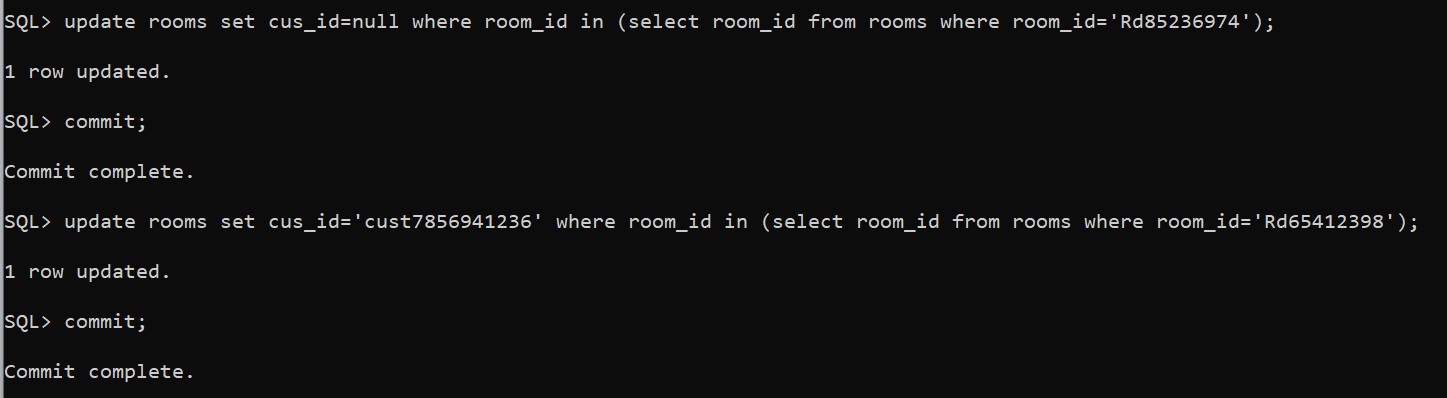
1. Update the name of the hotel (in rare circumstances). Before updating the data :-



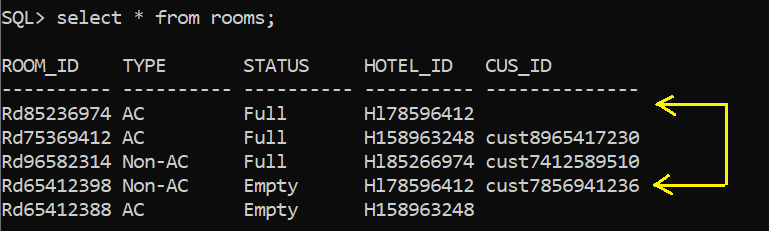
After updating the data :-

1. Update the room details if customer changed their room due to any inconvenience. Before updating the data :-

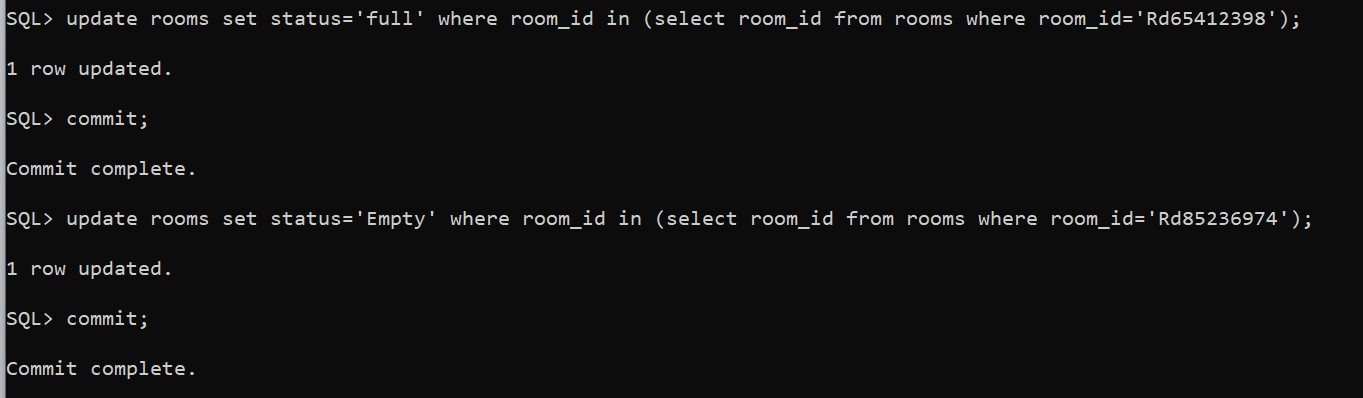




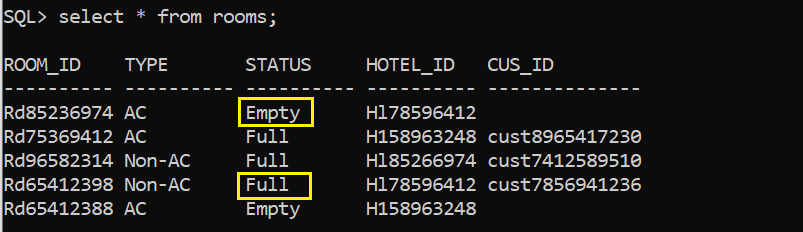
After updating the data :-



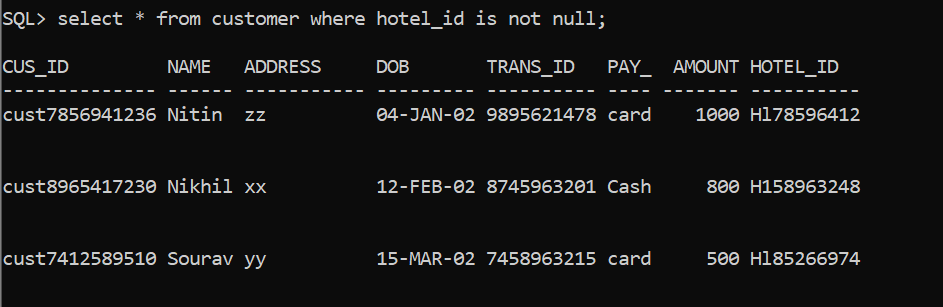
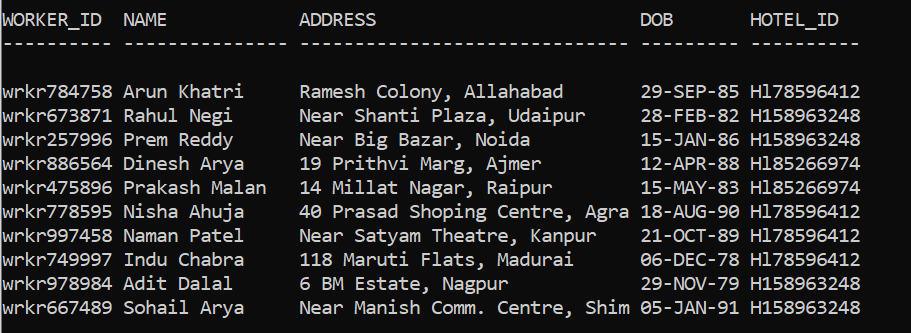
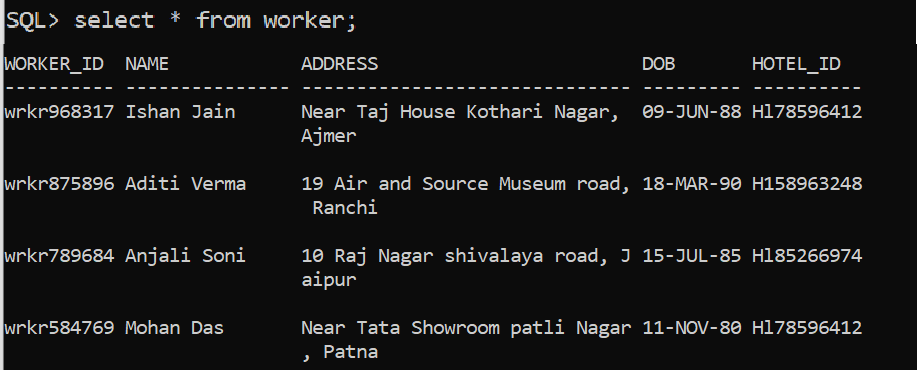
If rooms are changed then the status of rooms also be changed



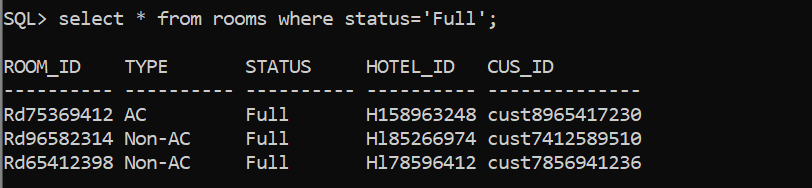
After updating the data :-

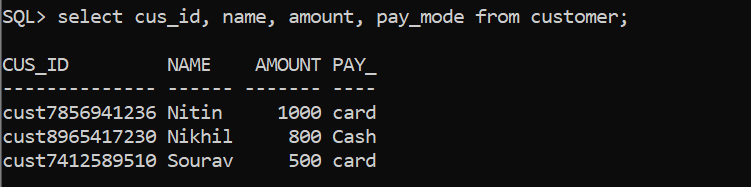


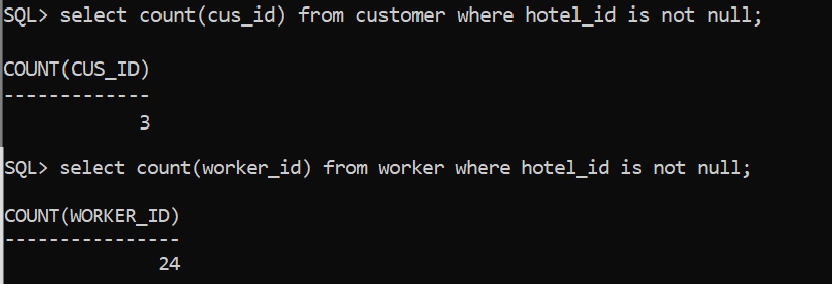
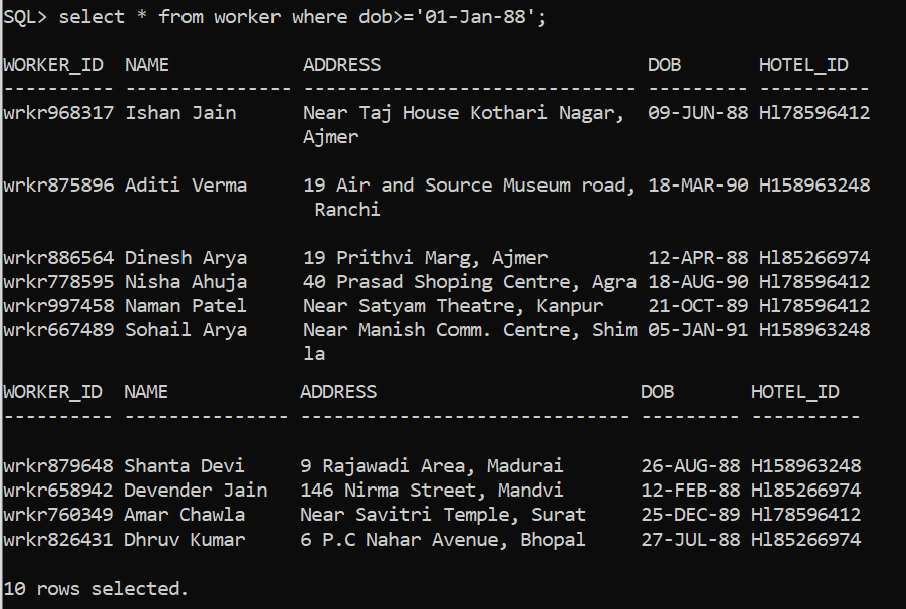
Retrieval Statements

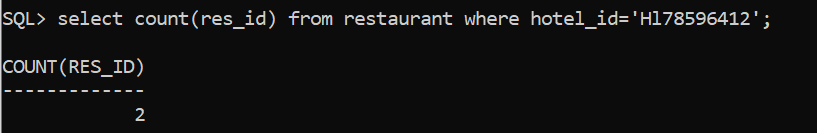
1. Retrieve the data of the customers who are staying in the hotel.
2. Retrieve the data of the workers who are working in the hotels.

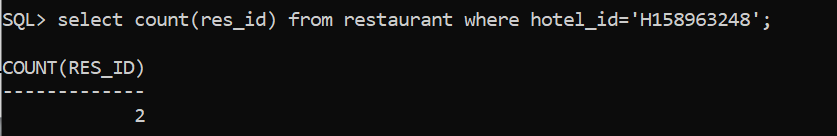


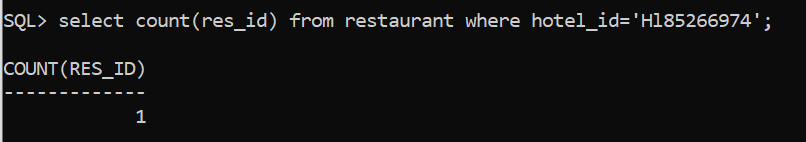
1. Retrieve the data of the rooms that are booked by the customers.
2. Retrieve the data of restaurant along with their hotel id.
3. Retrieve cus id, name, amount and the mode of payment by which customer pay their bill.



1. Count how many customers and workers are in the hotel at present time.
2. Retrieve the data of workers was born on or after 1st January 1988.
3. Retrieve the no. of restaurant a particular hotel has.

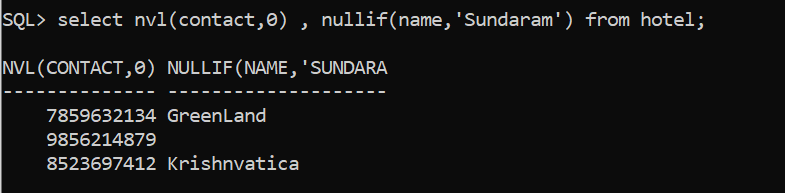




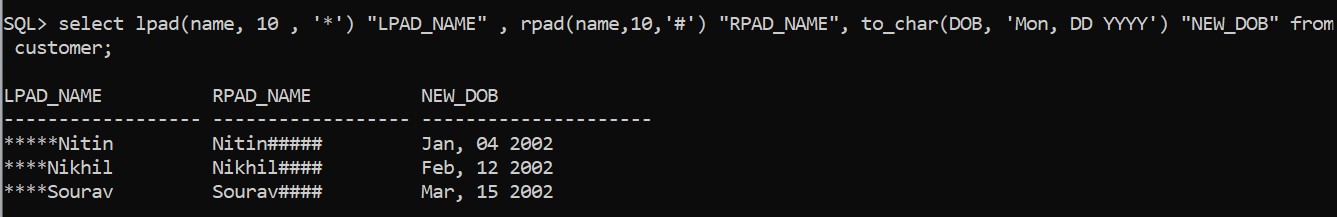


* 1. The select statements must include

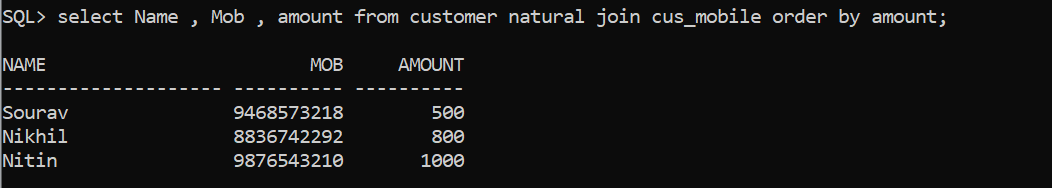
1. one query showing the usage of nvl function and nullif function,
2. one query showing usage of left and right padding character and date value formatting
3. one join query involving order by clause,
4. one uncorrelated nested query,
5. one correlated nested query,
6. one query involving one of the set operators,
7. one query involving group by, having and where clause and
8. one query involving (left or right or full) outer join.
9. Query involving *nvl* function and *nullif* function in one statement

* Display zero if any hotel has no contact and also replace the string with empty string if any hotel has name ‘Sundaram’.

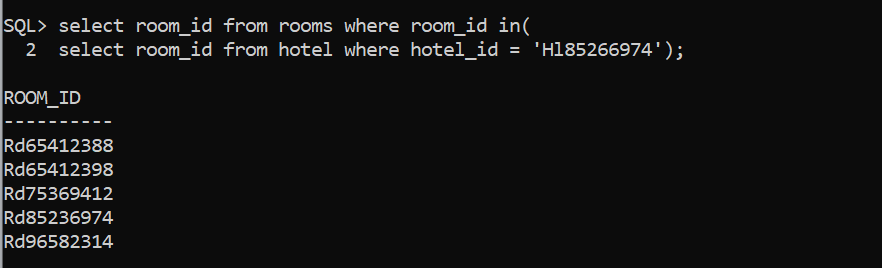
1. Query showing usage of left and right padding character and date formatting

* Display the customer name with left and right padding character with date formatting of DOB.

1. Join query involving order by clause

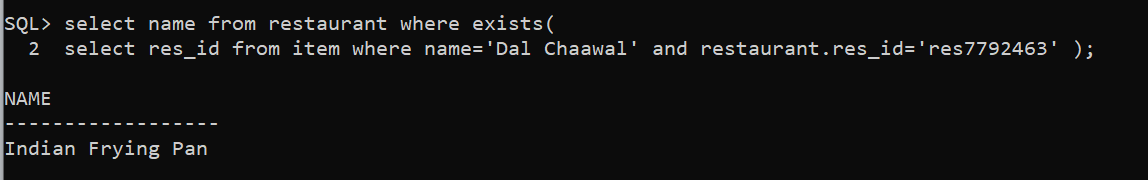
* Display name, mobile no. of customer also amount that customer have to pay in increasing order of amount.

1. Uncorrelated nested query

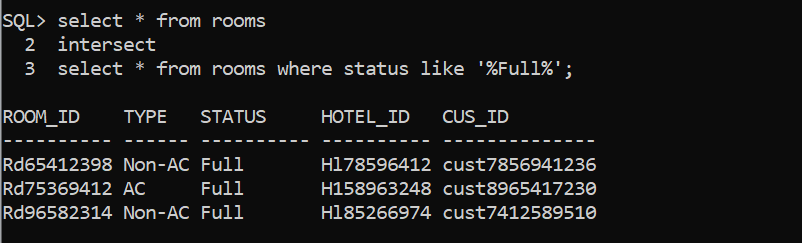
* Display all rooms id of any particular hotel.

.

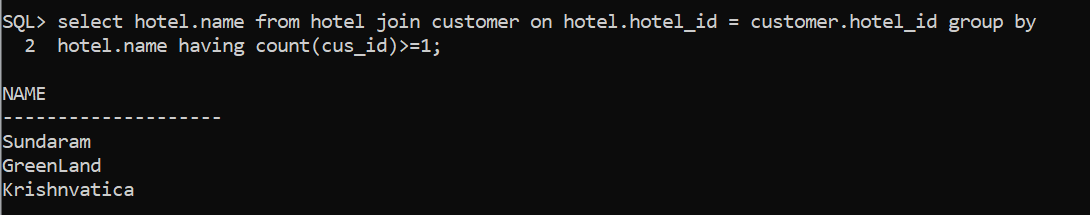
1. Correlated nested query

* Display the name of the restaurant which prepares Dal Chaawal.

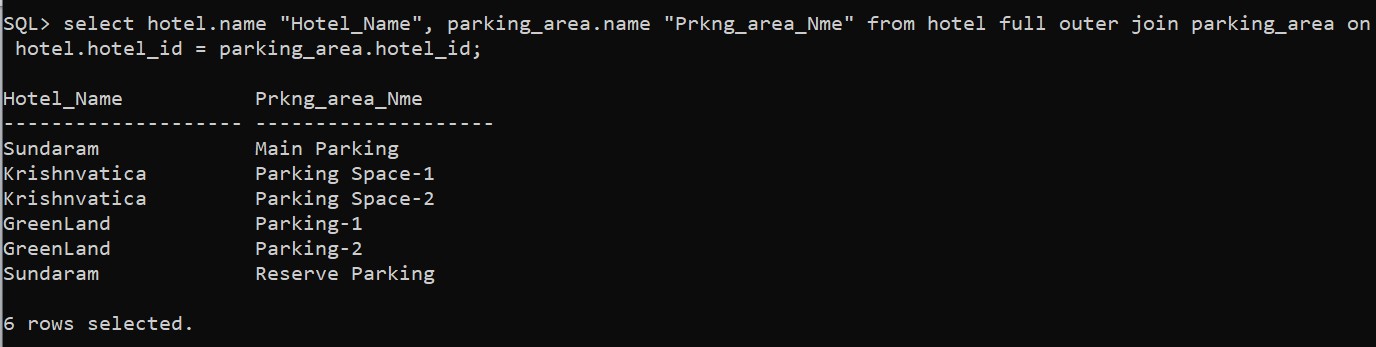
1. Query involving one of the set operators

* Display the details of the rooms that are full.

1. Query involving group by, having and where clause

* Display the names of the hotel that have atleast one customer.

1. Query involving (left or right or full) outer join

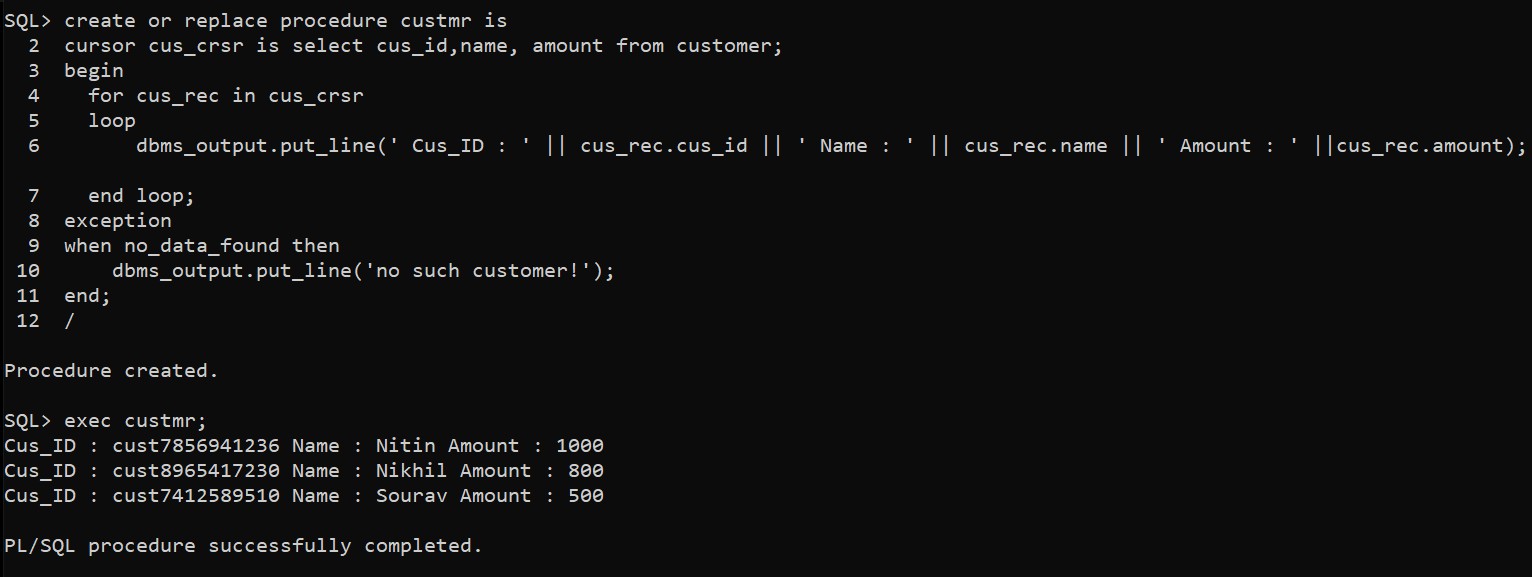
* Display the name of parking area with their respective hotel name.

Q7**.** Define and implement one PL/SQL function involving cursor and one PL/SQL procedure involving cursor and a trigger for the database under consideration.

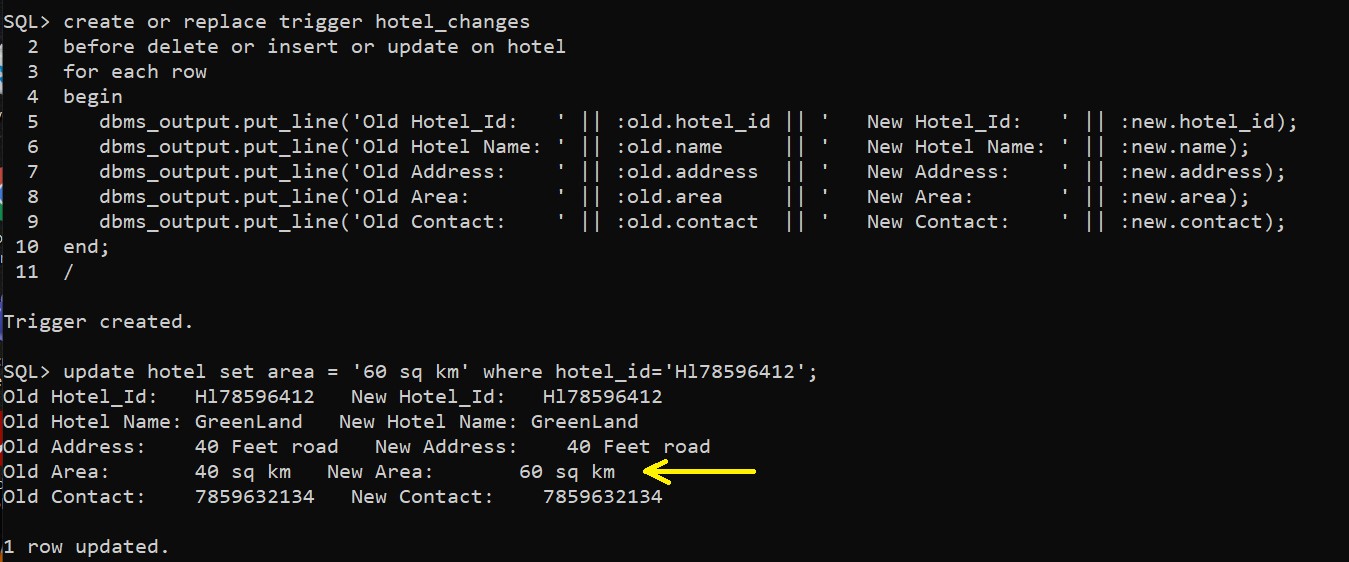
* + Function involving Cursor

Display the no. of workers using function involving cursor.

# Procedure involving Cursor

Display the cus\_id, name and amount of the customer using procedure involving cursor.

# Trigger

Display the changes when any changes are done in the hotel table using trigger.

