

DataBase Project
Phase 1

ABS Hotel

Teamwork:

Afrah Alsaadi - 439008815

Badriah Alqarni – 439003912

Amaal Asaedi - 438007658

Salha Almasoudi - 439002977

Group work report

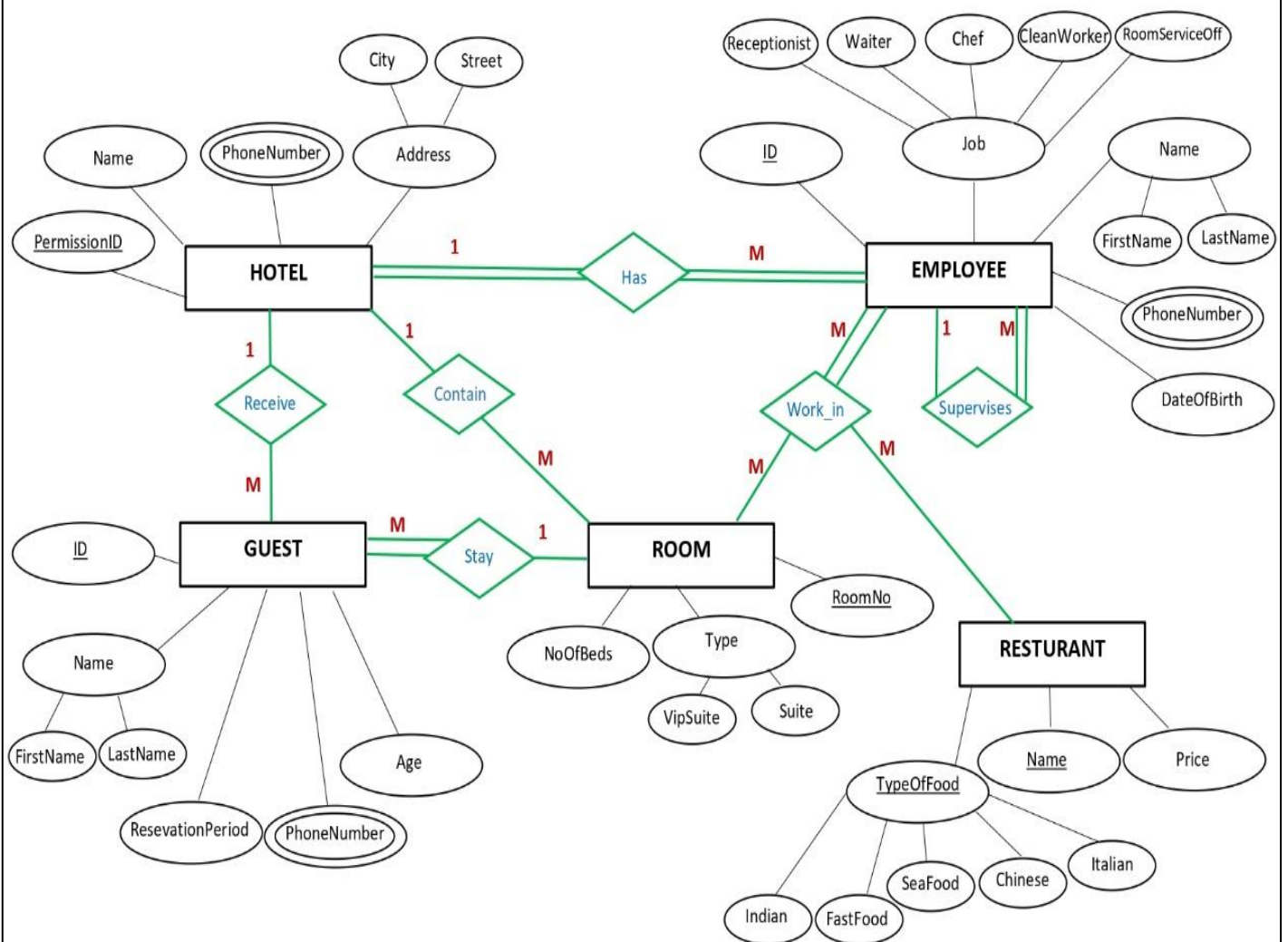
	Afrah 439008815	Badriah 439003912	Amaal 438007658	Salha 439002977
Description	•		•	
UML	○			•
Chen's	○	•		
Phase1 file	•			

- Done the task
- Assisted on the task

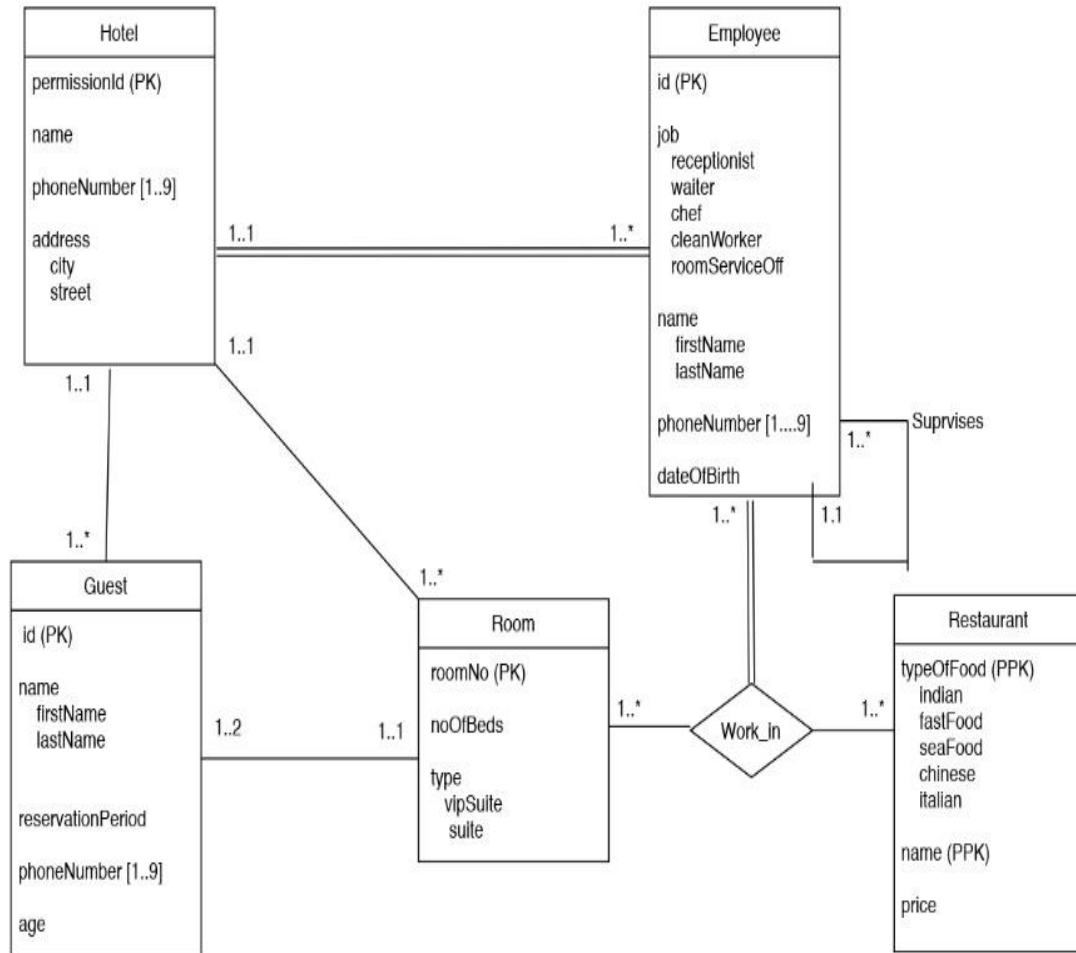
Description

- The ABS hotel receives many of guests, and it contains 100 rooms. Also, it has 70 employees working for it. The identifier for the hotel is Permission ID, and other attributes are name, city, state, and phoneNumber. The phoneNumber consists of nine digits.
- Each employee has an identifier ID, and other attributes firstName, lastName, job, dateOfBirth, and phoneNumber. The phoneNumber consists of nine digits. Each employee has exactly one job to do, and these jobs are like a chef, waiter, clean worker, receptionist, and room service officer. All employees have exactly one supervisor, and the supervisor may supervise more than one employee, but not all the employees are supervisors. All employees work in ABS hotel in different places. Some of them are working in the restaurant and the other working in rooms.
- At most, two guests can stay at one room, however that does not mean all hotel's rooms should be always full of guests . The identifier for the room is RoomNo, and the other attributes are numOfBeds, and Type. The hotel offers two types of rooms: suite, and VIP suite.
- The restaurant has two identifiers name and typeOfFood. The typesOfFood that the restaurant offers are Indian, Chinese, Italian, seaFood and fastFood. Also, the restaurant offers reasonable prices for its types of food.
- A guest has an identifier ID and other attributes. firstName, lastName, phoneNumber, reservation Period and age.

Chen's Notation



UML Notation



DataBase Project
Phase 2

ABS Hotel

Teamwork:

Afrah Alsaadi - 439008815

Badriah Alqarni – 439003912

Amaal Asaedi - 438007658

Salha Almasoudi - 439002977

Group work report

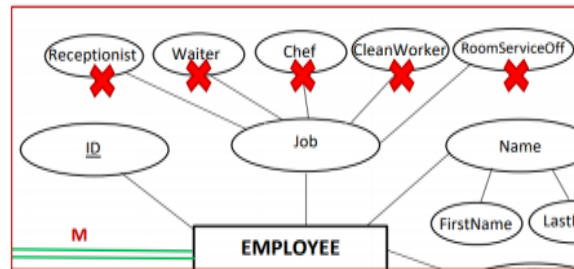
	Afrah 439008815	Badriah 439003912	Amaal 438007658	Salha 439002977
Modify Errors	•	•		
Relational Schema	•	•		
Normalization	•		•	•
Phase2 file		•		

- Done the task ○ Assisted on the task

Modify Errors in phase1

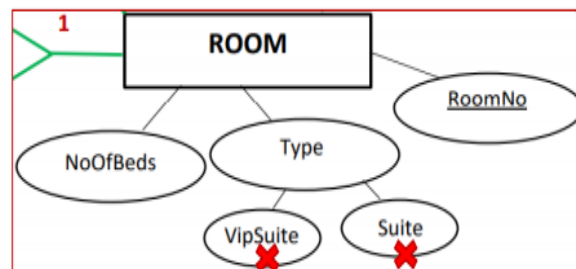
1-The values have been removed from the composite Attribute(**Job**) that belongs to the Entity(**EMPLOYEE**)

because the previously set values are values and the attribute has become simple Attribute and not composite Attribute.



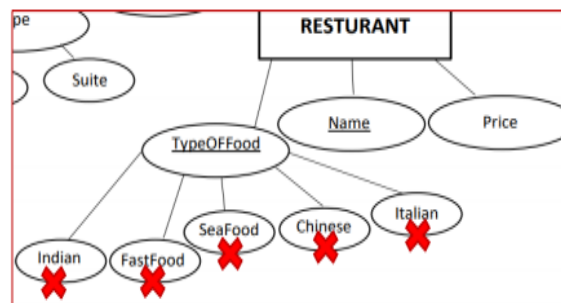
2-The values have been removed from the composite Attribute(**Type**) that belongs to the Entity(**ROOM**)

because the previously set values are values and the attribute has become simple Attribute and not composite Attribute.



3-The values have been removed from the composite Attribute(**TypeOFFood**) that belongs to the Entity(**RESTURANT**)

because the previously set values are values and the attribute has become simple Attribute and not composite Attribute.



Relational Schema

STEP 1: Mapping of Regular Entity Types :

EMPLOYEE

<u>ID</u>	job	firstName	lastName	dateOfBirth
-----------	-----	-----------	----------	-------------

HOTEL

<u>permissionID</u>	name	city	street
---------------------	------	------	--------

GUEST

<u>ID</u>	firstName	lastName	resevationPeriod	age
-----------	-----------	----------	------------------	-----

ROOM

<u>roomNo</u>	type	noOfBeds
---------------	------	----------

RESTURANT

<u>name</u>	<u>typeOfFood</u>	price
-------------	-------------------	-------

STEP 2: Mapping of Weak Entity Types :

NONE

Step 3: Mapping of one to one relationship Types :

NONE

STEP 4: Mapping of One to Many Relationships Types :

EMPLOYEE

<u>ID</u>	job	firstName	lastName	dateOfBirth	IDSupervisor	permissionID
-----------	-----	-----------	----------	-------------	--------------	--------------

ROOM

<u>roomNo</u>	type	noOfBeds	permissionID
---------------	------	----------	--------------

GUEST

<u>ID</u>	firstName	lastName	resevationPeriod	age	permissionID	roomNo
-----------	-----------	----------	------------------	-----	--------------	--------

Step 5: Mapping of many to many relationship Types :

NONE

Step 6: Multivalued Attribute mapped into a relation:

HOTEL_Phone

permissionID	<u>phoneNumber</u>
--------------	--------------------

GUEST_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

EMPLOYEE_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

Step 7: Mapping of N-ary relationship types :

Work_in

<u>ID</u>	<u>roomNo</u>	<u>name</u>	<u>typeOfFood</u>
-----------	---------------	-------------	-------------------

Final Relational Schema

EMPLOYEE

<u>ID</u>	job	firstName	lastName	dateOfBirth	IDSupervisor	permissionID
-----------	-----	-----------	----------	-------------	--------------	--------------

HOTEL

<u>permissionID</u>	name	city	street
---------------------	------	------	--------

GUEST

<u>ID</u>	firstName	lastName	resevationPeriod	age	permissionID	roomNo
-----------	-----------	----------	------------------	-----	--------------	--------

ROOM

<u>roomNo</u>	type	noOfBeds	permissionID
---------------	------	----------	--------------

RESTURANT

<u>name</u>	<u>typeOfFood</u>	price
-------------	-------------------	-------

HOTEL_Phone

<u>permissionID</u>	<u>phoneNumber</u>
---------------------	--------------------

GUEST_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

EMPLOYEE_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

Work_in

<u>ID</u>	<u>roomNo</u>	<u>name</u>	<u>typeOfFood</u>
-----------	---------------	-------------	-------------------

Normalization

EMPLOYEE

<u>ID</u>	job	firstName	lastName	dateOfBirth	IDSupervisor	permissionID
-----------	-----	-----------	----------	-------------	--------------	--------------

HOTEL

<u>permissionID</u>	name	city	street
---------------------	------	------	--------

GUEST

<u>ID</u>	firstName	lastName	resevationPeriod	age	permissionID	roomNo
-----------	-----------	----------	------------------	-----	--------------	--------

ROOM

<u>roomNo</u>	type	noOfBeds	permissionID
---------------	------	----------	--------------

RESTURANT

<u>name</u>	<u>typeOfFood</u>	price
-------------	-------------------	-------

HOTEL_Phone

<u>permissionID</u>	<u>phoneNumber</u>
---------------------	--------------------

GUEST_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

EMPLOYEE_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

Work_in

<u>ID</u>	<u>roomNo</u>	<u>name</u>	<u>typeOfFood</u>
-----------	---------------	-------------	-------------------

First Normal Form 1NF:

The Schema is already in 1NF

Second Normal Form 2NF:

EMPLOYEE

<u>ID</u>	job	firstName	lastName	dateOfBirth	IDSupervisor	permissionID
-----------	-----	-----------	----------	-------------	--------------	--------------

HOTEL

<u>permissionID</u>	name	city	street
---------------------	------	------	--------

GUEST

<u>ID</u>	firstName	lastName	resevationPeriod	age	permissionID	roomNo
-----------	-----------	----------	------------------	-----	--------------	--------

ROOM

<u>roomNo</u>	type	noOfBeds	permissionID
---------------	------	----------	--------------

RESTURANT

<u>name</u>	<u>typeOfFood</u>
-------------	-------------------

FOOD_PRICE

<u>name</u>	price
-------------	-------

HOTEL_Phone

<u>permissionID</u>	<u>phoneNumber</u>
---------------------	--------------------

GUEST_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

EMPLOYEE_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

Work_in

<u>ID</u>	<u>roomNo</u>	<u>name</u>	<u>typeOfFood</u>
-----------	---------------	-------------	-------------------

Third Normal Form 3NF:

EMPLOYEE

<u>ID</u>	job	firstName	lastName	dateOfBirth	IDSupervisor	permissionID
-----------	-----	-----------	----------	-------------	--------------	--------------

HOTEL

<u>permissionID</u>	name
---------------------	------

HOTEL_ADDRESS

<u>name</u>	city	street
-------------	------	--------

GUEST

<u>ID</u>	firstName	lastName	resevationPeriod	age	permissionID	roomNo
-----------	-----------	----------	------------------	-----	--------------	--------

ROOM

<u>roomNo</u>	type	noOfBeds	permissionID
---------------	------	----------	--------------

RESTURANT

<u>name</u>	<u>typeOfFood</u>
-------------	-------------------

FOOD_PRICE

<u>name</u>	price
-------------	-------

HOTEL_Phone

<u>permissionID</u>	<u>phoneNumber</u>
---------------------	--------------------

GUEST_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

EMPLOYEE_Phone

<u>ID</u>	<u>phoneNumber</u>
-----------	--------------------

Work_in

<u>ID</u>	<u>roomNo</u>	<u>name</u>	<u>typeOfFood</u>
-----------	---------------	-------------	-------------------

Normalization:

All of the relations confirm to the third normalization form, because:

1. They confirm to the 1NF because:

- a. They have not multi-valued attributes
- b. Each attribute only holds a single-type value

2. They confirm to the 2NF because:

- a. They confirm to the 1NF
- b. There are no non-prime attribute dependencies on part of the primary key

3. They confirm to the 3NF because:

- a. They confirm to the 2NF
- b. There are not transitive dependencies on a super key

DataBase Project
Phase 3

ABS Hotel

Teamwork:

Afrah Alsaadi - 439008815

Badriah Alqarni – 439003912

Amaal Asaedi - 438007658

Salha Almasoudi - 439002977

Group work report

	Afrah <i>439008815</i>	Badriah <i>439003912</i>	Amaal <i>438007658</i>	Salha <i>439002977</i>
CREATE TABLE	●			
INSERT	●			
Update & Delete commands				●
Select commands		●		
Phase3 file		●		

- Done the task
- Assisted on the task

CREATE TABLE HOTEL & EMPLOYEE

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane shows the 'abshotel' database with a list of tables including 'employee', 'employee_phone', 'food_price', 'guest', 'guest_phone', 'hotel', 'hotel_address', 'hotel_phone', 'restaurant', 'room', and 'work_in'. The 'Information' pane at the bottom shows 'No object selected'. The main query window displays the following SQL code:

```
1 CREATE TABLE abshotel.HOTEL
2 (
3     permissionID VARCHAR(5) NOT NULL,
4     hotelName VARCHAR(15) NOT NULL,
5     CONSTRAINT HOTEL_PK PRIMARY KEY (permissionID)
6 );
7
8 CREATE TABLE abshotel.EMPLOYEE
9 (
10     employeeID VARCHAR(5) NOT NULL,
11     fName VARCHAR(20) NOT NULL,
12     lName VARCHAR(20) NOT NULL,
13     job VARCHAR(30) NOT NULL,
14     dateOfBirth VARCHAR(15) NOT NULL,
15     supervisorID VARCHAR(5),
16     permissionID VARCHAR(5) NOT NULL,
17     CONSTRAINT EMPLOYEE_PK PRIMARY KEY (employeeID),
18     CONSTRAINT EMPLOYEE_FK1 FOREIGN KEY (permissionID) REFERENCES HOTEL(permissionID),
19     CONSTRAINT EMPLOYEE_FK2 FOREIGN KEY (supervisorID) REFERENCES EMPLOYEE(employeeID)
20 );
```

CREATE TABLE ROOM

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane shows the 'abshotel' database with a list of tables including 'employee', 'employee_phone', 'food_price', 'guest', 'guest_phone', 'hotel', 'hotel_address', 'hotel_phone', 'restaurant', 'room', and 'work_in'. The 'Information' pane at the bottom shows 'No object selected'. The main query window displays the following SQL code:

```
18 CREATE TABLE abshotel.ROOM
19 (
20     roomNo VARCHAR(5) NOT NULL,
21     roomType VARCHAR(20) NOT NULL,
22     noOfBeds INT(1) NOT NULL,
23     permissionID VARCHAR(5),
24     CONSTRAINT ROOM_PK PRIMARY KEY (roomNo),
25     CONSTRAINT ROOM_FK1 FOREIGN KEY (permissionID) REFERENCES HOTEL(permissionID)
26 );
```

CREATE TABLE GUEST & HOTEL_ADDRESS

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane shows the 'abshotel' database with a list of tables including 'employee', 'employee_phone', 'food_price', 'guest', 'guest_phone', 'hotel', 'hotel_address', 'hotel_phone', 'restaurant', 'room', and 'work_in'. The 'Information' pane at the bottom shows 'No object selected'. The main query window displays the following SQL code:

```
26 CREATE TABLE abshotel.GUEST
27 (
28     guestID VARCHAR(5) NOT NULL,
29     fName VARCHAR(15) NOT NULL,
30     lName VARCHAR(15),
31     resevationPeriod VARCHAR(10) NOT NULL,
32     age INT(2),
33     permissionID VARCHAR(5),
34     roomNo VARCHAR(5),
35     CONSTRAINT Guest_PK PRIMARY KEY (guestID),
36     CONSTRAINT Guest_FK1 FOREIGN KEY (roomNo) REFERENCES ROOM(roomNo),
37     CONSTRAINT Guest_FK2 FOREIGN KEY (permissionID) REFERENCES HOTEL(permissionID)
38 );
39
40 CREATE TABLE abshotel.HOTEL_ADDRESS
41 (
42     hotelName VARCHAR(15),
43     city VARCHAR(15),
44     street VARCHAR(15),
45     CONSTRAINT HOTEL_ADDRESS_PK PRIMARY KEY (hotelName)
46 );
```

CREATE TABLE RESTURANT & FOOD_PRICE

44 • CREATE TABLE abshotel.RESTURANT
 45 (nameOfFood VARCHAR(20) NOT NULL,
 46 typeOfFood VARCHAR(20) NOT NULL,
 47 CONSTRAINT RESTURANT_PK PRIMARY KEY (nameOfFood,typeOfFood)
 48);
 49 • CREATE TABLE abshotel.FOOD_PRICE
 50 (nameOfFood VARCHAR(20) NOT NULL,
 51 price INT(3) NOT NULL ,
 52 CONSTRAINT FOOD_PRICE_PK PRIMARY KEY (nameOfFood)
 53);

CREATE TABLE HOTEL_phone & GUEST_PHONE & EMPLOYEE_PHONE

54 • CREATE TABLE abshotel.HOTEL_phone
 55 (permissionID VARCHAR(5),
 56 phoneNumber VARCHAR(10) ,
 57 CONSTRAINT HOTEL_phone_PK PRIMARY KEY (permissionID,phoneNumber)
 58);
 59 • CREATE TABLE abshotel.GUEST_PHONE
 60 (guestID VARCHAR(5) ,
 61 phoneNumber VARCHAR(10) ,
 62 CONSTRAINT GUEST_PHONE_PK PRIMARY KEY (guestID,phoneNumber),
 63 CONSTRAINT GUEST_PHONE_FK FOREIGN KEY (guestID) REFERENCES GUEST(guestID)
 64);
 65 • CREATE TABLE abshotel.EMPLOYEE_PHONE
 66 (employeeID VARCHAR(5) ,
 67 phoneNumber VARCHAR(10) ,
 68 CONSTRAINT EEMPLOYEE_PHONE_PK PRIMARY KEY (employeeID,phoneNumber),
 69 CONSTRAINT EMPLOYEE_PHONEe_FK FOREIGN KEY (employeeID) REFERENCES EMPLOYEE(employeeID)
 70);

CREATE TABLE WORK_IN

71 • CREATE TABLE abshotel.WORK_IN
 72 (employeeID VARCHAR(5),
 73 roomNo VARCHAR(5) ,
 74 nameOfFood VARCHAR(20) ,
 75 typeOfFood VARCHAR(20) ,
 76 CONSTRAINT WORK_IN_PK PRIMARY KEY (employeeID,roomNo,nameOfFood,typeOfFood)
 77);

INSERT HOTEL

Filter objects	79	#HOTEL
▼ abshotel	80	• INSERT INTO abshotel.HOTEL (permissionID,hotelName)
▼ Tables	81	VALUES ("G9051","ABS HOTEL 1");
employee	82	• INSERT INTO abshotel.HOTEL (permissionID,hotelName)
employee_phone	83	VALUES ("G9052","ABS HOTEL 2");
food_price	84	• INSERT INTO abshotel.HOTEL (permissionID,hotelName)
guest	85	VALUES ("G9053","ABS HOTEL 3");
guest_phone	86	• INSERT INTO abshotel.HOTEL (permissionID,hotelName)
hotel	87	VALUES ("G9054","ABS HOTEL 4");
hotel_address	88	• INSERT INTO abshotel.HOTEL (permissionID,hotelName)
hotel_phone	89	VALUES ("G9055","ABS HOTEL 5");
restaurant	90	
room		
work_in		
Views		

INSERT EMPLOYEE

Filter objects	91	#EMPLOYEE
▼ abshotel	92	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,permissionID)
▼ Tables	93	VALUES ("E1001" , "Noha", "Saleh", "Waiter", "2-2-1990", "G9051");
employee	94	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,supervisorID,permissionID)
employee_phone	95	VALUES ("E1002" , "Ahmad", "Zain", "Waiter", "22-9-1988", "E1001", "G9051");
food_price	96	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,permissionID)
guest	97	VALUES ("E1003" , "Saleh", "Waleed", "Chef", "21-11-1993", "G9052");
guest_phone	98	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,supervisorID,permissionID)
hotel	99	VALUES ("E1004" , "Bader", "salem", "Chef", "1-12-1989", "E1003", "G9052");
hotel_address	100	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,permissionID)
hotel_phone	101	VALUES ("E1009" , "Majed", "Khaled", "Resceptionist", "17-7-1996", "G9053");
restaurant	102	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,permissionID)
room	103	VALUES ("E1006" , "Mathyo", "Keven", "CleanWorker", "28-10-1998", "G9054");
work_in	104	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,supervisorID,permissionID)
Views	105	VALUES ("E1005" , "Joe", "Maikel", "CleanWorker", "3-8-1986", "E1006", "G9054");
Stored Procedures	106	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,permissionID)
Functions	107	VALUES ("E1007" , "Jamse", "Joe", "RoomServiceOfficer", "30-11-1997", "G9055");
Administration	108	• INSERT INTO abshotel.EMPLOYEE (employeeID ,fName, lName,job,dateOfBirth,supervisorID,permissionID)
Schemas	109	VALUES ("E1008" , "Malkom", "Kareem", "RoomServiceOfficer", "2-1-1985", "E1007", "G9055");
Information	110	
No object selected		

INSERT ROOM

Filter objects	111	#ROOM
▼ abshotel	112	• INSERT INTO abshotel.ROOM(roomNo,roomType,noOfBeds,permissionID)
▼ Tables	113	VALUES ("F30", "Suite", 2, "G9051");
employee	114	• INSERT INTO abshotel.ROOM(roomNo,roomType,noOfBeds,permissionID)
employee_phone	115	VALUES ("F39", "VIP Suite", 1, "G9052");
food_price	116	• INSERT INTO abshotel.ROOM(roomNo,roomType,noOfBeds,permissionID)
guest	117	VALUES ("F37", "Suite", 2, "G9054");
guest_phone	118	• INSERT INTO abshotel.ROOM(roomNo,roomType,noOfBeds,permissionID)
hotel	119	VALUES ("F38", "VIP Suite", 1, "G9053");
hotel_address	120	• INSERT INTO abshotel.ROOM(roomNo,roomType,noOfBeds,permissionID)
hotel_phone	121	VALUES ("F31", "Suite", 2, "G9055");
restaurant	122	
room		
work_in		
Views		

INSERT RESTURANT

```

123 #RESTURANT
124 • INSERT INTO abshotel.RESTURANT(nameOfFood,typeOfFood)
125 VALUES ("Pitza", "Italian");
126 • INSERT INTO abshotel.RESTURANT(nameOfFood,typeOfFood)
127 VALUES ("Spring Rolls", "Chinese");
128 • INSERT INTO abshotel.RESTURANT(nameOfFood,typeOfFood)
129 VALUES ("Sea Fruit Salad", "Sea Food");
130 • INSERT INTO abshotel.RESTURANT(nameOfFood,typeOfFood)
131 VALUES ("Chicken Burger", "Fast Food");
132 • INSERT INTO abshotel.RESTURANT(nameOfFood,typeOfFood)
133 VALUES ("Biryani", "Indian");
134

```

INSERT GUEST

```

135 #GUEST
136 • INSERT INTO abshotel.GUEST(guestID,fName,lName,resevationPeriod,age,permissionID,roomNo)
137 VALUES ("U3999", "Adam", "show", "4 Days", 55, "G9051", "F30");
138 • INSERT INTO abshotel.GUEST(guestID,fName,lName,resevationPeriod,age,permissionID,roomNo)
139 VALUES ("U7989", "Abdullah", "Saleh", "2 Days", 35, "G9052", "F39");
140 • INSERT INTO abshotel.GUEST(guestID,fName,lName,resevationPeriod,age,permissionID,roomNo)
141 VALUES ("U4009", "Htem", "Albert", "9 Days", 28, "G9053", "F37");
142 • INSERT INTO abshotel.GUEST(guestID,fName,lName,resevationPeriod,age,permissionID,roomNo)
143 VALUES ("U6876", "Sam", "Simbson", "7 Days", 55, "G9054", "F38");
144 • INSERT INTO abshotel.GUEST(guestID,fName,lName,resevationPeriod,age,permissionID,roomNo)
145 VALUES ("U1178", "Sharlok", "Jems", "1 Day", 55, "G9055", "F31");
146

```

INSERT HOTEL_ADDRESS

```

147 #HOTEL_ADDRESS
148 • INSERT INTO abshotel.HOTEL_ADDRESS (hotelName,city,street)
149 VALUES ( "ABS HOTEL 1","Makkah","AL-HAJE ");
150 • INSERT INTO abshotel.HOTEL_ADDRESS (hotelName,city,street)
151 VALUES ("ABS HOTEL 2", "Jeddah", "AL-aziziah ");
152 • INSERT INTO abshotel.HOTEL_ADDRESS (hotelName,city,street)
153 VALUES ("ABS HOTEL 3", "Riyad", "AL-aziziah");
154 • INSERT INTO abshotel.HOTEL_ADDRESS (hotelName,city,street)
155 VALUES ("ABS HOTEL 4", "Dammam", "AL-aziziah ");
156 • INSERT INTO abshotel.HOTEL_ADDRESS (hotelName,city,street)
157 VALUES ("ABS HOTEL 5", "Madenah", "AL-aziziah ");
158

```

INSERT FOOD_PRICE

```

159 #FOOD_PRICE
160 • INSERT INTO abshotel.FOOD_PRICE(nameOfFood,price)
161 VALUES ("Pitza",30);
162 • INSERT INTO abshotel.FOOD_PRICE(nameOfFood,price)
163 VALUES ("Spring Rolls",20);
164 • INSERT INTO abshotel.FOOD_PRICE(nameOfFood,price)
165 VALUES ("Sea Fruit Salad",40);
166 • INSERT INTO abshotel.FOOD_PRICE(nameOfFood,price)
167 VALUES ("Chicken Burger",28);
168 • INSERT INTO abshotel.FOOD_PRICE(nameOfFood,price)
169 VALUES ("Biryani",60);

```

INSERT HOTEL_phone

```

171 #HOTEL_phone
172 • INSERT INTO abshotel.HOTEL_phone(permissionID,phoneNumber)
173 VALUES ("G9051" , "0555-6789");
174 • INSERT INTO abshotel.HOTEL_phone(permissionID,phoneNumber)
175 VALUES ("G9052" , "0534-2189");
176 • INSERT INTO abshotel.HOTEL_phone(permissionID,phoneNumber)
177 VALUES ("G9053" , "0876-4221");
178 • INSERT INTO abshotel.HOTEL_phone(permissionID,phoneNumber)
179 VALUES ("G9054" , "0345-1339");
180 • INSERT INTO abshotel.HOTEL_phone(permissionID,phoneNumber)
181 VALUES ("G9055" , "0788-7932");
182

```

INSERT GUEST_PHONE

```

183 #GUEST_PHONE
184 • INSERT INTO abshotel.GUEST_PHONE(guestID,phoneNumber)
185 VALUES ("U3999" , "0564-3229");
186 • INSERT INTO abshotel.GUEST_PHONE(guestID,phoneNumber)
187 VALUES ("U7989" , "0564-3777");
188 • INSERT INTO abshotel.GUEST_PHONE(guestID,phoneNumber)
189 VALUES ("U4009" , "0585-6229");
190 • INSERT INTO abshotel.GUEST_PHONE(guestID,phoneNumber)
191 VALUES ("U6876" , "0598-7359");
192 • INSERT INTO abshotel.GUEST_PHONE(guestID,phoneNumber)
193 VALUES ("U1178" , "0543-2980");
194 • INSERT INTO abshotel.GUEST_PHONE(guestID,phoneNumber)
195 VALUES ("U4009" , "0567-5342");
196

```

INSERT EMPLOYEE_PHONE

```

197 #EMPLOYEE_PHONE
198 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
199 VALUES("E1001" , "0555-5432");
200 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
201 VALUES("E1002" , "0511-1114");
202 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
203 VALUES("E1003" , "0554-3212");
204 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
205 VALUES("E1004" , "0587-5422");
206 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
207 VALUES("E1005" , "0543-2002");
208 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
209 VALUES("E1006" , "0590-0875");
210 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
211 VALUES("E1007" , "0522-3322");
212 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
213 VALUES("E1008" , "0555-5332");
214 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
215 VALUES("E1001" , "0577-7772");
216 • INSERT INTO abshotel.EMPLOYEE_PHONE(employeeID,phoneNumber)
217 VALUES("E1009" , "0555-5552");
218

```

INSERT WORK_IN

```

218
219 #WORK_IN
220 • INSERT INTO abshotel.WORK_IN(employeeID,roomNo,nameOfFood,typeOfFood)
221   VALUES("E1001","F30","Pitza","Italian");
222 • INSERT INTO abshotel.WORK_IN(employeeID,roomNo,nameOfFood,typeOfFood)
223   VALUES("E1002","F39","Spring Rolls","Chinese");
224 • INSERT INTO abshotel.WORK_IN(employeeID,roomNo,nameOfFood,typeOfFood)
225   VALUES("E1008","F37","Sea Fruit Salad","Sea Food");
226 • INSERT INTO abshotel.WORK_IN(employeeID,roomNo,nameOfFood,typeOfFood)
227   VALUES("E1009","F38","Chicken Burger","Fast Food");
228 • INSERT INTO abshotel.WORK_IN(employeeID,roomNo,nameOfFood,typeOfFood)
229   VALUES("E1006","F31","Biryani","Indian");
230
  
```

All Tables Are Before commands

SCHEMAS

Filter objects

abshotel

Tables

employee

employee_phone

food_price

guest

guest_phone

hotel

hotel_address

hotel_phone

restaurant

room

work_in

Views

Stored Procedures

Administration Schemas

Information

1 • SELECT * FROM abshotel.employee;

Result Grid

employeeID	fName	lName	job	dateOfBirth	supervisorID	permissionID
E1001	Noha	Saleh	Waiter	2-2-1990	NULL	G9051
E1002	Ahmad	Zain	Waiter	22-9-1988	E1001	G9051
E1003	Saleh	Waleed	Chef	21-11-1993	NULL	G9052
E1004	Bader	salem	Chef	1-12-1989	E1003	G9052
E1005	Joe	Maikel	CleanWorker	3-8-1986	E1006	G9054
E1006	Mathyo	Keven	CleanWorker	28-10-1998	NULL	G9054
E1007	Jamse	Joe	RoomServiceOfficer	30-11-1997	NULL	G9055
E1008	Malkom	Kareem	RoomServiceOfficer	2-1-1985	E1007	G9055
E1009	Majed	Khaled	Receptionist	17-7-1996	NULL	G9053

SCHEMAS

Filter objects

abshotel

Tables

employee

employee_phone

food_price

guest

guest_phone

hotel

hotel_address

hotel_phone

restaurant

room

work_in

Views

Stored Procedures

Administration Schemas

Information

1 • SELECT * FROM abshotel.employee_phone;

Result Grid

employeeID	phoneNumber
E1002	0511-1114
E1003	0554-3212
E1004	0587-5422
E1005	0543-2002
E1006	0590-0875
E1007	0522-3322
E1008	0555-5332
E1009	0555-5552

SCHEMAS

Filter objects

abshotel

Tables

employee

employee_phone

food_price

guest

guest_phone

hotel

hotel_address

hotel_phone

restaurant

room

work_in

Views

Stored Procedures

Administration Schemas

Information

1 • SELECT * FROM abshotel.food_price;

Result Grid

nameOfFood	price
Biryani	60
Chicken Burger	28
Pitza	30
Sea Fruit Salad	40
Spring Rolls	20
NULL	NULL

SCHEMAS

Filter objects

abshotel

- Tables
 - employee
 - employee_phone
 - food_price
 - guest**
 - guest_phone
 - hotel
 - hotel_address
 - hotel_phone
 - restaurant
 - room
 - work_in
- Views
- Stored Procedures
- Functions

Administration Schemas

1 • SELECT * FROM abshotel.guest;

Limit to 1000 rows

Result Grid

	guestID	fName	lName	reservationPeriod	age	permissionID	roomNo
▶	U1178	Sharlok	Jems	1 Day	55	G9055	F31
	U3999	Adam	show	4 Days	55	G9051	F30
	U4009	Htem	Albert	9 Days	28	G9053	F37
	U6876	Sam	Simbson	7 Days	55	G9054	F38
	U7989	AbdUllah	Saleh	2 Days	35	G9052	F39

SCHEMAS

Filter objects

abshotel

- Tables
 - employee
 - employee_phone
 - food_price
 - guest
 - guest_phone**
 - hotel
 - hotel_address
 - hotel_phone
 - restaurant
 - room
 - work_in
- Views
- Stored Procedures
- Functions

Administration Schemas

1 • SELECT * FROM abshotel.guest_phone;

Limit to 1000 rows

Result Grid

	guestID	phoneNumber
▶	U1178	0543-2980
	U3999	0564-3229
	U4009	0567-5342
	U4009	0585-6229
	U6876	0598-7359
	U7989	0564-3777
	NULL	NULL

SCHEMAS

Filter objects

abshotel

- Tables
 - employee
 - employee_phone
 - food_price
 - guest
 - guest_phone
 - hotel**
 - hotel_address
 - hotel_phone
 - restaurant
 - room
 - work_in
- Views
- Stored Procedures
- Functions

Administration Schemas

1 • SELECT * FROM abshotel.hotel;

Limit to 1000 rows

Result Grid

	permissionID	hotelName
▶	G9051	ABS HOTEL 1
	G9052	ABS HOTEL 2
	G9053	ABS HOTEL 3
	G9054	ABS HOTEL 4
	G9055	ABS HOTEL 5

SCHEMAS

Filter objects

abshotel

- Tables
 - employee
 - employee_phone
 - food_price
 - guest
 - guest_phone
 - hotel
 - hotel_address**
 - hotel_phone
 - restaurant
 - room
 - work_in
- Views
- Stored Procedures
- Functions

Administration Schemas

1 • SELECT * FROM abshotel.hotel_address;

Limit to 1000 rows

Result Grid

	hotelName	city	street
▶	ABS HOTEL 1	Makkah	AL-HAJE
	ABS HOTEL 2	Jeddah	AL-aziziah
	ABS HOTEL 3	Riyad	AL-aziziah
	ABS HOTEL 4	Dammam	AL-aziziah
	ABS HOTEL 5	Madenah	AL-aziziah

SCHEMAS

Filter objects

abshotel

- Tables
 - employee
 - employee_phone
 - food_price
 - guest
 - guest_phone
 - hotel
 - hotel_address
 - hotel_phone**
 - restaurant
 - room
 - work_in
- Views
- Stored Procedures
- Functions

Administration Schemas

1 • SELECT * FROM abshotel.hotel_phone;

Result Grid

permissionID	phoneNumber
G9051	0555-6789
G9052	0534-2189
G9053	0876-4221
G9054	0345-1339
G9055	0788-7932

SCHEMAS

Filter objects

abshotel

- Tables
 - employee
 - employee_phone
 - food_price
 - guest
 - guest_phone
 - hotel
 - hotel_address
 - hotel_phone
 - restaurant**
 - room
 - work_in
- Views
- Stored Procedures
- Functions

Administration Schemas

1 • SELECT * FROM abshotel.restaurant;

Result Grid

nameOfFood	typeOfFood
Biryani	Indian
Chicken Burger	Fast Food
Pitza	Italian
Sea Fruit Salad	Sea Food
Spring Rolls	Chinese

SCHEMAS

Filter objects

abshotel

- Tables
 - employee
 - employee_phone
 - food_price
 - guest
 - guest_phone
 - hotel
 - hotel_address
 - hotel_phone
 - restaurant
 - room**
 - work_in
- Views
- Stored Procedures
- Functions

Administration Schemas

1 • SELECT * FROM abshotel.room;

Result Grid

roomNo	roomType	noOfBeds	permissionID
F30	Suite	2	G9051
F31	Suite	2	G9055
F37	Suite	2	G9054
F38	VIP Suite	1	G9053
F39	VIP Suite	1	G9052

SCHEMAS

Filter objects

abshotel

- Tables
 - employee
 - employee_phone
 - food_price
 - guest
 - guest_phone
 - hotel
 - hotel_address
 - hotel_phone
 - restaurant
 - room
 - work_in**
- Views
- Stored Procedures
- Functions

Administration Schemas

1 • SELECT * FROM abshotel.work_in;

Result Grid

employeeID	roomNo	nameOfFood	typeOfFood
E1001	F30	Pitza	Italian
E1002	F39	Spring Rolls	Chinese
E1006	F31	Biryani	Indian
E1008	F37	Sea Fruit Salad	Sea Food
E1009	F38	Chicken Burger	Fast Food

Update & Delete commands

```
264 • update food_price
265 set price=price*1.05
266 where price <60;
```

Table After command

	nameOfFood	price
►	Biryani	60
	Chicken Burger	30
	Pitza	34
	Sea Fruit Salad	44
	Spring Rolls	22

```
268
269 • delete from hotel_address
270 where city='Dammam';
```

Table After command

hotelName	city	street
► ABS HOTEL 1	Makkah	AL-HAJE
ABS HOTEL 2	Jeddah	AL-aziziah
ABS HOTEL 3	Riyad	AL-aziziah
ABS HOTEL 5	Madenah	AL-aziziah

All commands

```

271
272 • select hotel.permissionID,hotelName,employeeID,fName,lName,job
273     from hotel,employee
274     where hotel.permissionID=employee.permissionID
275     order by hotel.permissionID desc;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	permissionID	hotelName	employeeID	fName	lName	job
▶	G9055	ABS HOTEL 5	E1007	Jamse	Joe	RoomServiceOfficer
	G9055	ABS HOTEL 5	E1008	Malkom	Kareem	RoomServiceOfficer
	G9054	ABS HOTEL 4	E1005	Joe	Maikel	CleanWorker
	G9054	ABS HOTEL 4	E1006	Mathyo	Keven	CleanWorker
	G9053	ABS HOTEL 3	E1009	Majed	Khaled	Resceptionist
	G9052	ABS HOTEL 2	E1003	Saleh	Waleed	Chef
	G9052	ABS HOTEL 2	E1004	Bader	salem	Chef
	G9051	ABS HOTEL 1	E1001	Noha	Saleh	Waiter
	G9051	ABS HOTEL 1	E1002	Ahmad	Zain	Waiter

```

276
277 • select guestID,fName,lName,age
278     from guest
279     where age<=(select avg(age) from guest);
280
281

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	guestID	fName	lName	age
▶	U4009	Htem	Albert	28
	U7989	AbdUllah	Saleh	35
*	NULL	NULL	NULL	NULL

```

272 • select guestID,resevationPeriod
273     from guest
274     where resevationPeriod >=6;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	guestID	resevationPeriod
▶	U4009	9 Days
	U6876	7 Days

```

277 • select roomNo,roomType,noOfBeds
278     from room
279     where roomType='Suite' AND noOfBeds=2 ;

```

Result Grid | Filter Rows: | Edit: | Export/Import: |

	roomNo	roomType	noOfBeds
▶	F30	Suite	2
	F31	Suite	2
	F37	Suite	2

```

282 • select phoneNumber,guestID
283     from guest_phone
284     order by guestID desc;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

	phoneNumber	guestID
▶	0564-3777	U7989
	0598-7359	U6876
	0585-6229	U4009
	0567-5342	U4009
	0564-3229	U3999
	0543-2980	U1178

```

287 • select roomType,count(permissionID)as count
288 from room
289 group by roomType
290 having count(permissionID)>=1;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

roomType	count
Suite	3
VIP Suite	2

```

301 • select *
302 from room
303 where roomNo =(select roomNo
304 from guest
305 where resevationPeriod<=1
306 and age>=55);
307
308 • select job,count(employeeID)as NumEmployees

```

Result Grid | Filter Rows: | Edit: | Export/

roomNo	roomType	noOfBeds	permissionID
F31	Suite	2	G9055

```

308 • select job,count(employeeID)as NumEmployees
309 from employee
310 group by job
311 order by employeeID asc;
312

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

job	NumEmployees
Waiter	2
Chef	2
CleanWorker	2
RoomServiceOfficer	2
Resceptionist	1

```

312
313 • select nameOfFood,price
314 from food_price
315 where price between 22 And 40;
316

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

nameOfFood	price
Chicken Burger	30
Pitza	34
Spring Rolls	22
NULL	NULL

```

317
318 • select typeOfFood
319 from resturant
320 where nameOfFood='Biryani' OR nameOfFood='Sea Fruit Salad';

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

typeOfFood
Indian
Sea Food

DataBase Project
Phase 4

ABS Hotel

Teamwork:

Afrah Alsaadi - 439008815

Badriah Alqarni – 439003912

Salha Almasoudi - 439002977

Group work report

	Afrah 439008815	Badriah 439003912	Salha 439002977
PowerPoint presentation	•		•
pdf file containing all the phases		•	

- Done the task ○ Assisted on the task