



# DATABASE PROJECT

## HOTEL MANAGEMENT DATABASE

GROUP8

**Shahad Hassan Altalhi 444001817**

**Ruba Mohsen Alsaedi 444001785**

**Wajen Naif Almatrafi 444006715**

**Mayas Adil Alakbary 444005008**

# GROUP WORK

**Phase1**

**Phase2**

**Phase3**

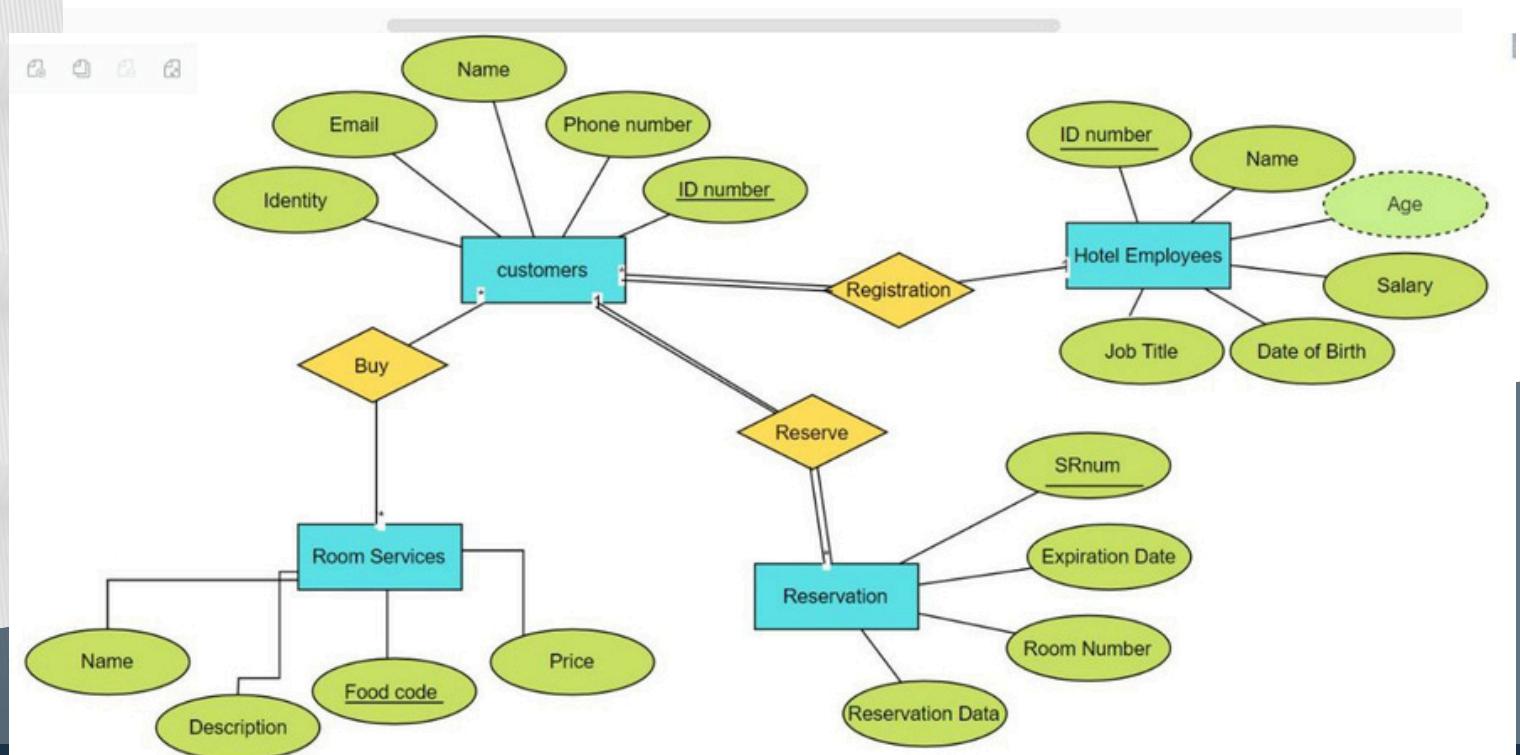
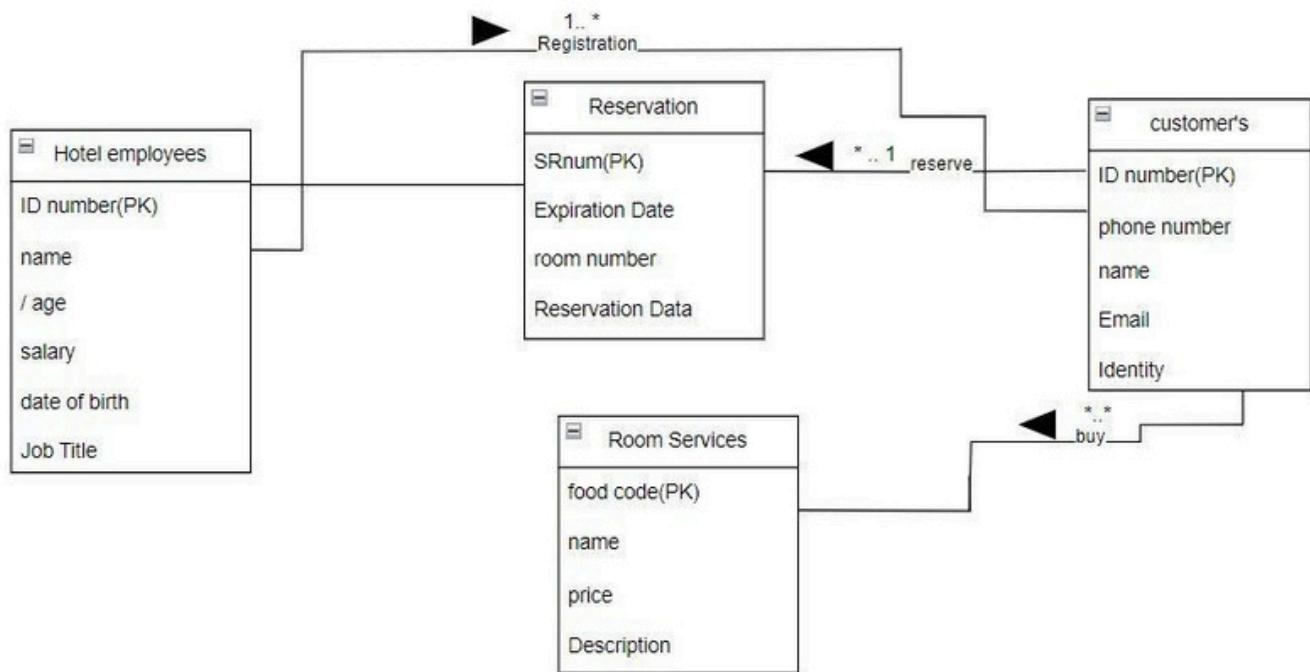
Task	Shahad	Ruba	Wajen	Mayas
Business rule	✓			✓
UML		✓		
Chen's			✓	
Mapping	✓	✓		
Normalizition			✓	✓
Creat tables, CREATE SCHEMA			✓	
insert data, UPDATE, DELETE		✓		
SELECT command	✓			✓

# PHASE1

## BUSINESS RULES

- The hotel employee reserves rooms for customers.
- Hotel employees' data is name ,date of birth, job number, age(calculated ),and salary.
- The hotel requires that the customer's data be registered, which is the name, phone number, and ID number.
- The hotel system reserves only one room for each customer in each reservation process and records the reservation date, expiration date, and room number, SRnum.
- Customers can purchase from the hotel's restaurant(room service) , and the hotel stores a list of the foods available to it by registering the food code, name, price, and a brief overview of it.

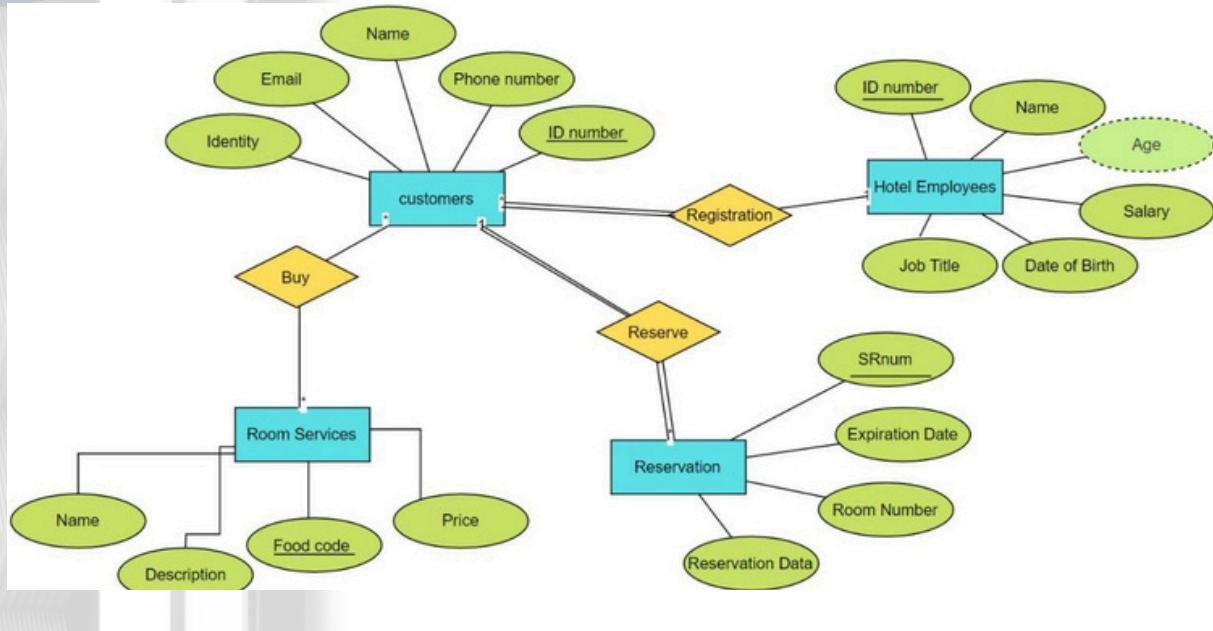
# PHASE1



# PHASE2

## ER-TO-RELATIONAL MAPPING

### STEP 1: MAPPING OF REGULAR ENTITY TYPES



customers

<u>ID number</u>	phone number	Name	email	identity
------------------	--------------	------	-------	----------

Hotel Employee

<u>ID number</u>	Name	salary	Date of birth	job title	Age
------------------	------	--------	---------------	-----------	-----

Room services

<u>Food code</u>	price	Name	description
------------------	-------	------	-------------

Reservation

<u>SRnum</u>	Reservation Date	Room number	Expiration date
--------------	------------------	-------------	-----------------

# ER-TO-RELATIONAL MAPPING

## STEP 2: MAPPING OF WEAK ENTITY TYPES

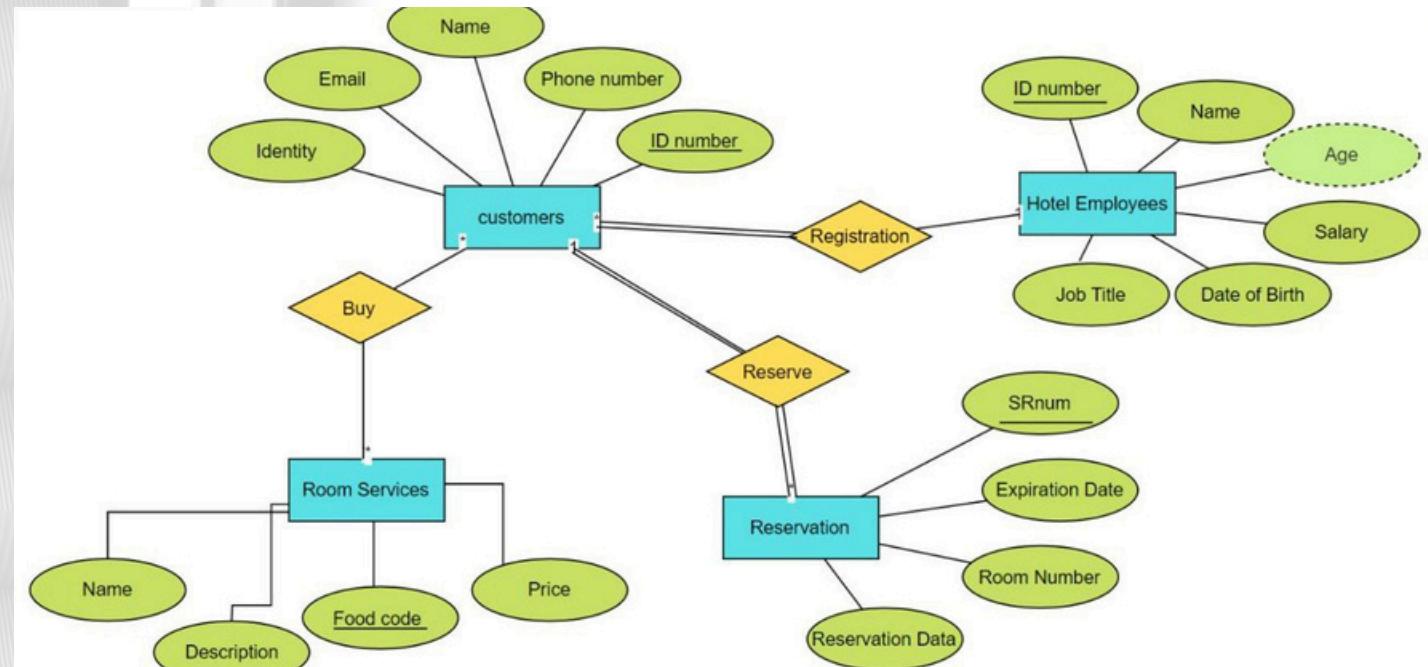
**NONE**

## STEP 3: MAPPING OF BINARY 1:1 RELATIONSHIP TYPES

**NONE**

# ER-TO-RELATIONAL MAPPING

## STEP 4: MAPPING OF BINARY 1:N RELATIONSHIP TYPES



Relationship : reserve (customer&reservation)

customer

ID number	phone number	Name	email	identity
Reservation				
SRnum	Reservation Date	Room number	Expiration date	ID number

Relationship : Registration (customer & Hotel employee)

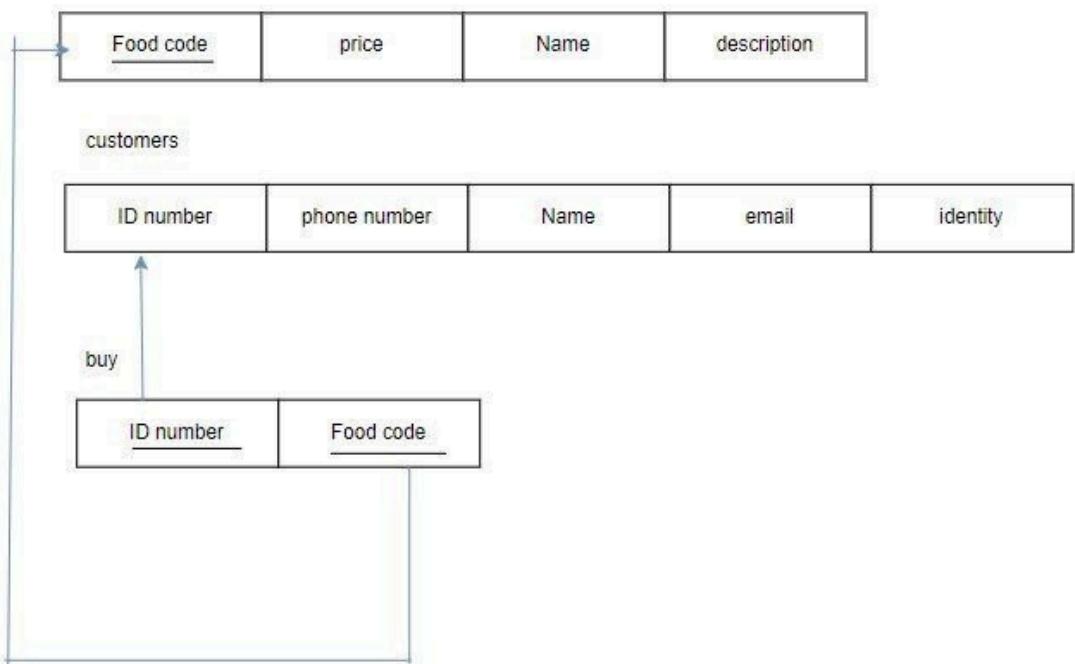
customer

identity	ID number	phone number	Name	email	ID numer_hotel employee
Hotel employee					
ID number	Name	salary	Date of birth	job title	Age

# ER-TO-RELATIONAL MAPPING

## STEP 5: MAPPING OF BINARY M:N RELATIONSHIP TYPES

Relationship : Buy (Customer & Room Service)



# ER-TO-RELATIONAL MAPPING

## STEP 6: MAPPING OF MULTIVALUED ATTRIBUTES

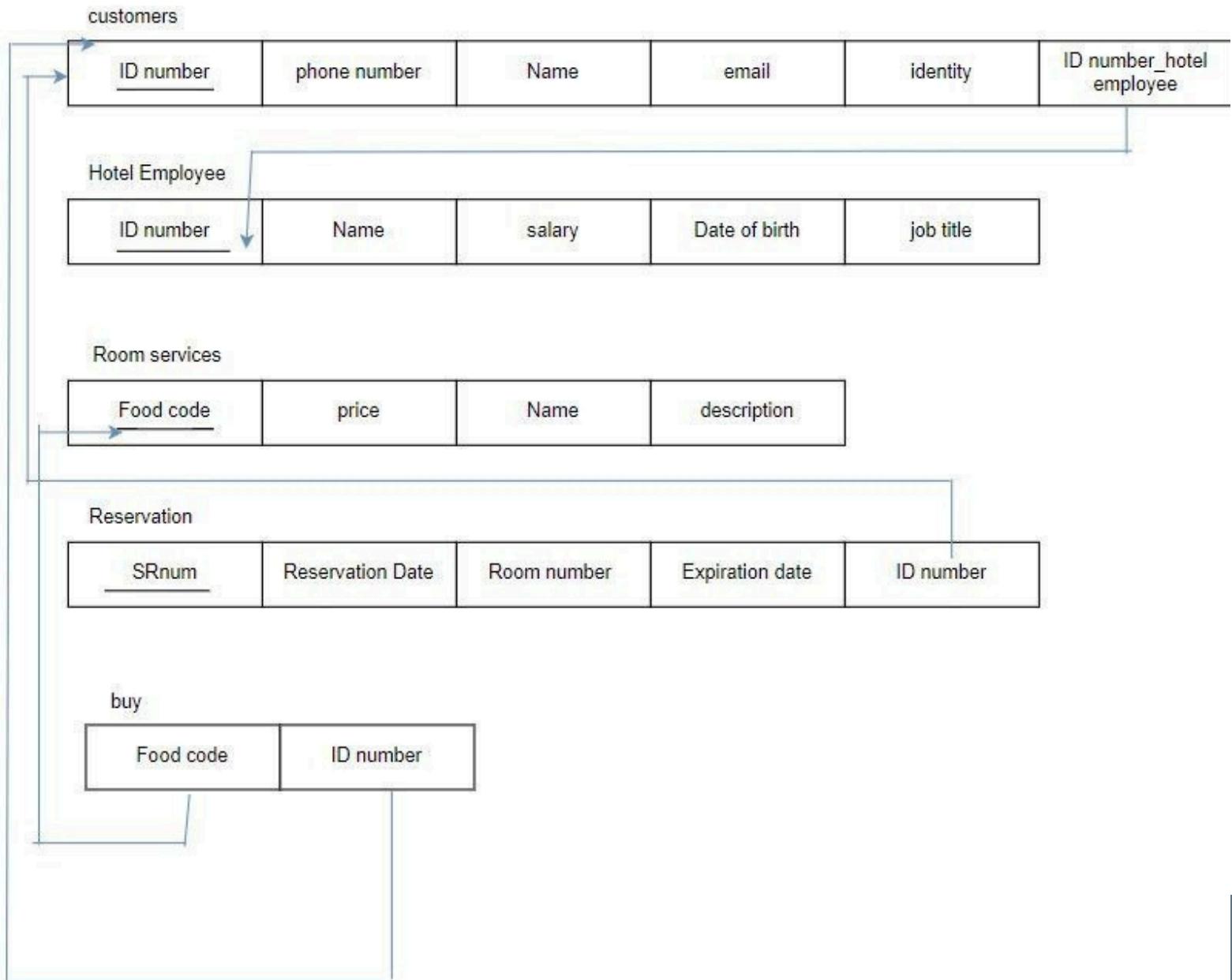
**NONE**

## STEP 7: MAPPING OF N-ARY RELATIONSHIP TYPES

**NONE**

# ER-TO-RELATIONAL MAPPING

## FINAL MAPPING



# PHASE 3

## CREATE SCHEMA

### Create tables

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below the toolbar, a code editor window displays the SQL script for creating the 'hotelmanagement' schema and its tables. The code is as follows:

```
1 •  create schema hotelmanagement;
2 •  CREATE TABLE customers (
3     IDnumber INT AUTO_INCREMENT PRIMARY KEY,
4     phonenumer VARCHAR(20) NOT NULL,
5     Name VARCHAR(50) NOT NULL,
6     email VARCHAR(100) UNIQUE,
7     identity VARCHAR(20) NOT NULL,
8     IDnumber_hotelemployee INT,
9     FOREIGN KEY (IDnumber_hotelemployee) REFERENCES hotelemployee(IDnumber)
10 );
11 •  CREATE TABLE hotelemployee (
12     IDnumber INT AUTO_INCREMENT PRIMARY KEY,
13     Name VARCHAR(50) NOT NULL,
14     salary DECIMAL(10,2) NOT NULL,
15     Date_of_birth DATE NOT NULL,
16     job_title VARCHAR(50) NOT NULL
17 );
18 •  CREATE TABLE roomservices (
19     Food_code INT AUTO_INCREMENT PRIMARY KEY,
20     price DECIMAL(10,2) NOT NULL,
21     Name VARCHAR(50) NOT NULL,
22     description VARCHAR(255)
23 );
24 •  CREATE TABLE reservation (
25     SRnum INT AUTO_INCREMENT PRIMARY KEY,
26     Reservation_Date DATE NOT NULL,
27     Room_number INT NOT NULL,
28     Expiration_date DATE NOT NULL,
29     IDnumber INT,
30     FOREIGN KEY (IDnumber) REFERENCES customers(IDnumber)
31 );
32 •  CREATE TABLE buy (
33     Food_code INT,
34     IDnumber INT,
35     PRIMARY KEY (Food_code, IDnumber),
36     FOREIGN KEY (Food_code) REFERENCES roomservices(Food_code),
37     FOREIGN KEY (IDnumber) REFERENCES customers(IDnumber)
38 );
39
```

At the bottom of the screen, the 'Tables' section of the 'hotelmanagement' schema is visible, showing the five tables: buy, customers, hotelemployee, reservation, and roomservices.

# PHASE 3

## INSERT,UPDATE,DELETE :

The image shows two separate sessions in MySQL Workbench. Both sessions are connected to the 'hotelmanagement' schema.

**Session 1 (Top):**

- Query: `INSERT INTO hotelemployee VALUES(1, 'sara', 1000, '2001-10-12', 'Receptionist'), (2, 'seba', 2000, '1999-10-25', 'Maintenance Supervisor'), (3, 'lama', 5000, '1990-08-13', 'Reservation Agent'), (4, 'mary', 87200, '2000-03-12', 'Restaurant Waiter'), (5, 'ran', 12000, '1998-05-16', 'Room Service Attendant');`
- Output:

IDnumber	Name	salary	Date_of_birth	job_title
1	sara	1000.00	2001-10-12	Receptionist
2	seba	2000.00	1999-10-25	Maintenance Supervisor
3	lama	5000.00	1990-08-13	Reservation Agent
4	mary	87200.00	2000-03-12	Restaurant Waiter
5	ran	12000.00	1998-05-16	Room Service Attendant

- Action Output:

#	Time	Action	Message	Duration / Fetch
42	01:12:06	CREATE TABLE hotelemployee ( IDnumber INT AUTO_INCREMENT PRIMARY KEY, Name VARCHAR(50), salary DECIMAL(10,2), Date_of_birth DATE, job_title VARCHAR(50) ) ENGINE=InnoDB;	0 row(s) affected	0.047 sec
43	01:12:06	CREATE TABLE customers ( IDnumber INT AUTO_INCREMENT PRIMARY KEY, Name VARCHAR(50), address VARCHAR(100), city VARCHAR(50), state VARCHAR(2), zip_code INT, phone INT, email VARCHAR(100) ) ENGINE=InnoDB;	0 row(s) affected	0.063 sec
44	01:12:06	CREATE TABLE roomservices ( Food_code INT AUTO_INCREMENT PRIMARY KEY, Service_name VARCHAR(50), Price DECIMAL(10,2) ) ENGINE=InnoDB;	0 row(s) affected	0.047 sec
45	01:12:06	CREATE TABLE reservation ( SNum INT AUTO_INCREMENT PRIMARY KEY, RoomNumber INT, Check_in DATE, Check_out DATE, Price DECIMAL(10,2) ) ENGINE=InnoDB;	0 row(s) affected	0.047 sec
46	01:12:06	CREATE TABLE buy ( Food_code INT, IDnumber INT, PRIMARY KEY(Food_code, IDnumber) ) ENGINE=InnoDB;	0 row(s) affected	0.047 sec
47	01:12:06	INSERT INTO hotelemployee VALUES(1, 'sara', 1000, '2001-10-12', 'Receptionist'), (2, 'seba', 2000, '1999-10-25', 'Maintenance Supervisor'), (3, 'lama', 5000, '1990-08-13', 'Reservation Agent'), (4, 'mary', 87200, '2000-03-12', 'Restaurant Waiter'), (5, 'ran', 12000, '1998-05-16', 'Room Service Attendant');	5 rows affected. Duration: 0.047 sec. Warnings: 0	0.047 sec

**Session 2 (Bottom):**

- Query: `SELECT * FROM hotelemployee;`
- Output:

IDnumber	Name	salary	Date_of_birth	job_title
1	sara	1000.00	2001-10-12	Receptionist
2	seba	2000.00	1999-10-25	Maintenance Supervisor
3	lama	5000.00	1990-08-13	Reservation Agent
4	mary	87200.00	2000-03-12	Restaurant Waiter
5	ran	12000.00	1998-05-16	Room Service Attendant

- Action Output:

#	Time	Action	Message	Duration / Fetch
44	01:12:06	CREATE TABLE roomservices ( Food_code INT AUTO_INCREMENT PRIMARY KEY, Service_name VARCHAR(50), Price DECIMAL(10,2) ) ENGINE=InnoDB;	0 row(s) affected	0.047 sec
45	01:12:06	CREATE TABLE reservation ( SNum INT AUTO_INCREMENT PRIMARY KEY, RoomNumber INT, Check_in DATE, Check_out DATE, Price DECIMAL(10,2) ) ENGINE=InnoDB;	0 row(s) affected	0.047 sec
46	01:12:06	CREATE TABLE buy ( Food_code INT, IDnumber INT, PRIMARY KEY(Food_code, IDnumber) ) ENGINE=InnoDB;	0 row(s) affected	0.047 sec

# PHASE 3

MySQL Workbench

Local instance MySQL80 ×

Edit View Query Database Server Tools Scripting Help

SQL File 4\* × hotelemployee

1 • UPDATE hotelemployee  
2 SET salary = 9000  
3 WHERE IDnumber = 1;  
4  
5 • SELECT \* FROM hotelemployee;

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

IDnumber	Name	salary	Date_of_birth	job_title
1	sara	9000.00	2001-10-12	Receptionist
2	seba	2000.00	1999-10-25	Maintenance Supervisor
3	lama	5000.00	1990-08-13	Reservation Agent
*	HOLe	HOLe	HOLe	HOLe

hotelemployee 3 ×

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL File 4\* × hotelemployee hotelmanagement - Schema

1 • DELETE FROM hotelemployee  
2 WHERE IDnumber = 5;  
3  
4 • SELECT \* FROM hotelemployee;

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

IDnumber	Name	salary	Date_of_birth	job_title
1	sara	9000.00	2001-10-12	Receptionist
2	seba	2000.00	1999-10-25	Maintenance Supervisor
3	lama	5000.00	1990-08-13	Reservation Agent
4	mary	87200.00	2000-03-12	Restaurant Waiter
*	HOLe	HOLe	HOLe	HOLe

hotelemployee 6 ×

# PHASE 3

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Q. Filter objects

coursedb

hotelmanagement

firstlabs

lab2

Administration Schemas

SQL File 4\* | hotelemployee | customers

1 • INSERT INTO customers

VALUES(100,'05673','walaee','wala100@gmail.com',11873,1),

(200,'05643','joood','joood@gmail.com',1158,2),

(300,'05797','salf','salf90@gmail.com',1024,3),

(400,'05797','sama','sama7@gmail.com',1198,4),

(500,'05797','remas','remas5@gmail.com',1144,1);

7

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Q. Filter objects

coursedb

hotelmanagement

firstlabs

lab2

Administration Schemas

SQL File 4\* | hotelemployee | customers

1 • DELETE from customers

2 WHERE IDnumber = 500;

3

4 • SELECT \* FROM customers

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

IDnumber	phonenumer	Name	email	identity	IDnumber_hotelemployee
100	05673	wala	wala100@gmail.com	11873	1
200	05643	joood	joood@gmail.com	1158	2
300	05797	salf	salf90@gmail.com	1024	3
400	05553	sama	sama7@gmail.com	1198	4
*	*	*	*	*	*

# PHASE 3

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

course db

Tables

- course
- instructor
- teachingassistant

Views

Stored Procedures

Functions

firstlabs

hotellemployee

Tables

- buy
- customers
- hotellemployee
- reservation
- roomservices

Views

Stored Procedures

Functions

lab2

SQL File 4\* × hotellemployee customers

1 • UPDATE customers  
2 SET phonenumber = '05553'  
3 WHERE IDnumber = 400;

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

course db

Tables

- course
- instructor
- teachingassistant

Views

Stored Procedures

Functions

firstlabs

hotellemployee

Tables

- buy
- customers
- hotellemployee
- reservation
- roomservices

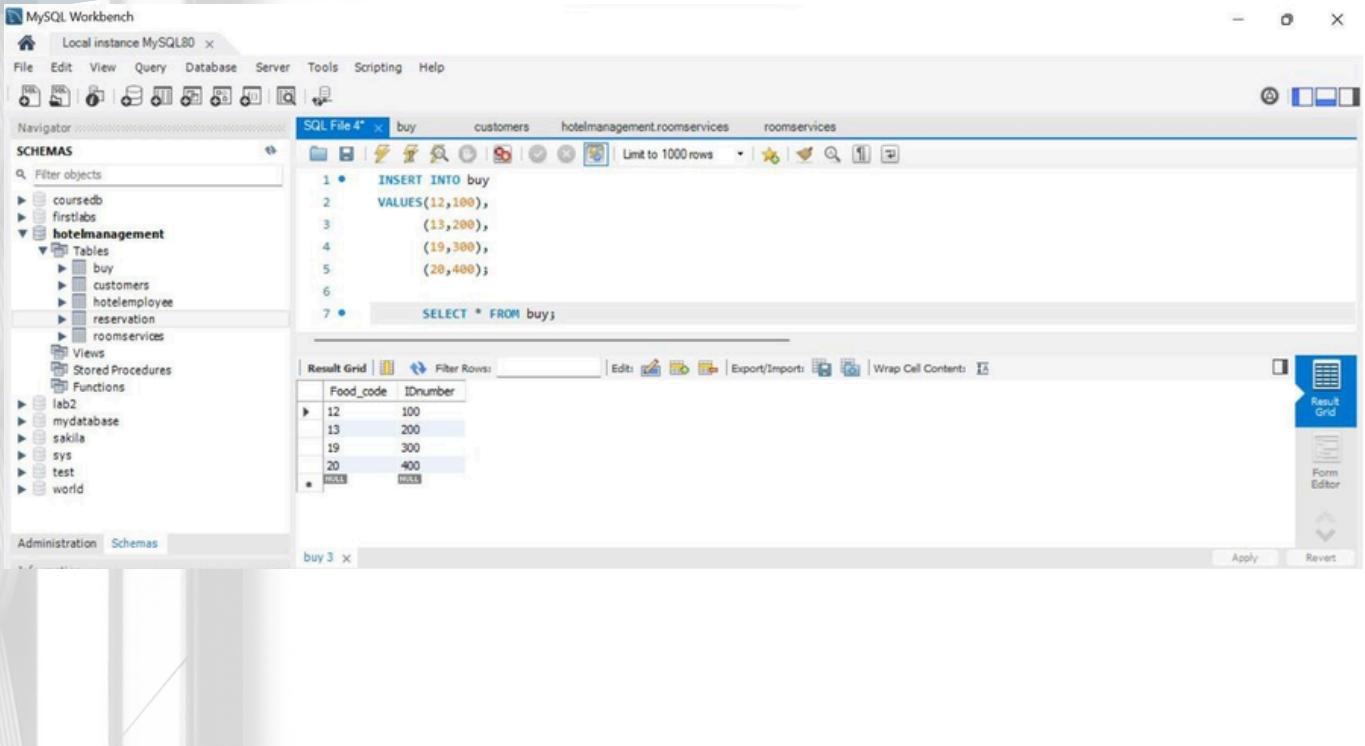
Views

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

IDNumber	phonenumber	Name	Email	identity	IDNumber_hotellemployee
100	05673	walaa	walaa100@gmail.com	11873	1
200	05643	jood	jood@gmail.com	1158	2
300	05797	salf	salf90@gmail.com	1024	3
400	05553	sama	sama7@gmail.com	1198	4

Result Grid Form

# PHASE 3



MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas SQL File 4\* buy customers hotelmanagement.roomservices roomservices

SCHEMAS

Filter objects

coursesdb  
firstlabs  
hotelmanagement  
Tables: buy, customers, hotelemployee, reservation, roomservices  
Views  
Stored Procedures  
Functions  
lab2  
mydatabase  
sakila  
sys  
test  
world

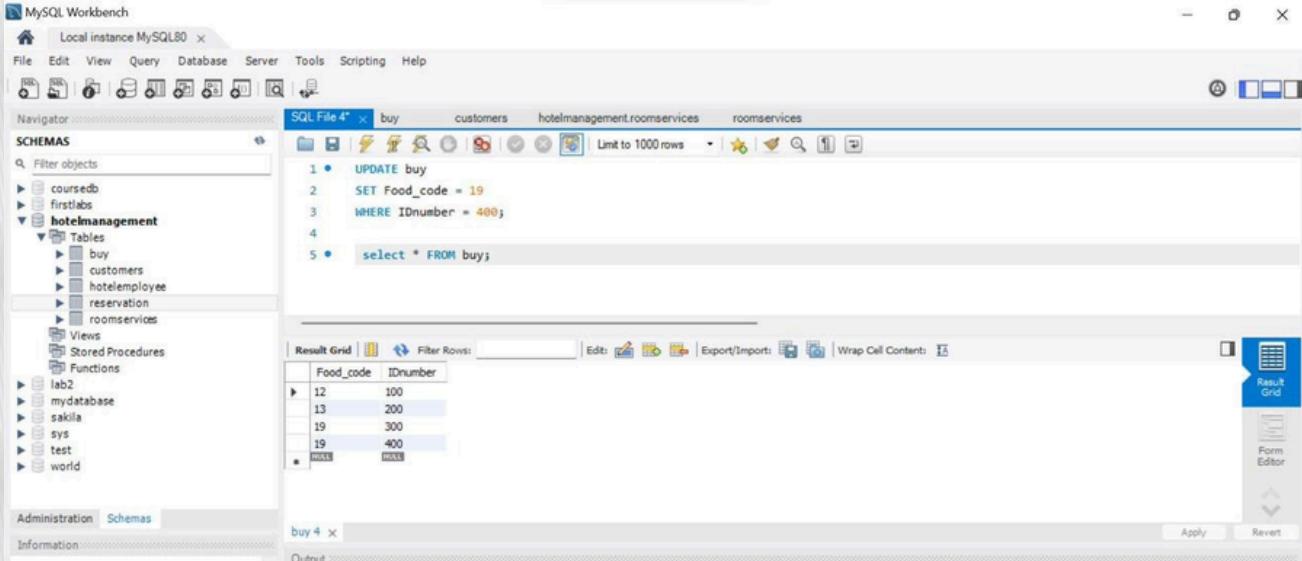
SQL File 4\*

```
1 • INSERT INTO buy
2     VALUES(12,100),
3             (13,200),
4             (19,300),
5             (20,400);
6
7 • SELECT * FROM buy;
```

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

Food_code	IDnumber
12	100
13	200
19	300
20	400
*	NULL
*	NULL

Administration Schemas buy 3 x Apply Revert



MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas SQL File 4\* buy customers hotelmanagement.roomservices roomservices

SCHEMAS

Filter objects

coursesdb  
firstlabs  
hotelmanagement  
Tables: buy, customers, hotelemployee, reservation, roomservices  
Views  
Stored Procedures  
Functions  
lab2  
mydatabase  
sakila  
sys  
test  
world

SQL File 4\*

```
1 • UPDATE buy
2     SET Food_code = 19
3     WHERE IDnumber = 400;
4
5 • select * FROM buy;
```

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

Food_code	IDnumber
12	100
13	200
19	300
19	400
*	NULL
*	NULL

Administration Schemas buy 4 x Information Output Apply Revert

# PHASE 3

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

coursedb  
firstlab  
hotelmanagement  
Tables: buy, customers, hotelemployee, reservation, roomservices  
Views  
Stored Procedures  
Functions  
lab2  
mydatabase  
sakila  
sys  
test  
world

SQL File 4\* × roomservices

```
1 • INSERT INTO roomservices VALUES (12,25,'pasta','healthy'),  
2     (13,30,'pizza','pizza with vegetables'),  
3     (19,10,'Sushi','flavorful and bite-sized'),  
4     (20,8,'Chicken Shawarma','spiced, and savory');  
5  
6 • SELECT * FROM roomservices;
```

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

Food_code	price	Name	description
12	25.00	pasta	healthy
13	30.00	pizza	pizza with vegetables
19	10.00	Sushi	Sushi flavorful and bite-sized
20	8.00	Chicken Shawarma	spiced, and savory
*	NULL	NULL	NULL

roomservices 1 ×

Output: Apply Revert

This screenshot shows the MySQL Workbench interface with a SQL editor window titled 'SQL File 4\* × roomservices'. The editor contains two statements: an INSERT query and a SELECT query. The SELECT query retrieves all rows from the roomservices table. Below the editor is a 'Result Grid' table with four columns: Food\_code, price, Name, and description. The data in the table matches the inserted values. The MySQL Workbench interface includes a Navigator pane on the left listing databases and tables, and a toolbar at the top.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

coursedb  
firstlab  
hotelmanagement  
Tables: buy, customers, hotelemployee, reservation, roomservices  
Views  
Stored Procedures  
Functions  
lab2  
mydatabase  
sakila  
sys  
test  
world

SQL File 4\* × roomservices

```
1 • UPDATE roomservices  
2     SET price = 20  
3     WHERE Food_code = 12;  
4  
5  
6 • SELECT * FROM roomservices;
```

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

Food_code	price	Name	description
12	20.00	pasta	healthy
13	30.00	pizza	pizza with vegetables
19	10.00	Sushi	flavorful and bite-sized
20	8.00	Chicken Shawarma	spiced, and savory
*	NULL	NULL	NULL

roomservices 2 ×

Output: Apply Revert

This screenshot shows the MySQL Workbench interface with a SQL editor window titled 'SQL File 4\* × roomservices'. It contains an UPDATE statement setting the price to 20 for food code 12, followed by a SELECT statement. The SELECT query retrieves all rows from the roomservices table. Below the editor is a 'Result Grid' table with the same four columns as before. The data has been updated, with the price for food code 12 now being 20.00. The MySQL Workbench interface is identical to the first screenshot, including the Navigator pane and toolbar.

# PHASE 3

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

SQL File 4\* hotelemployee customers reservation

1 • INSERT INTO reservation

```
VALUES(01,'2024-4-12',90,'2024-4-20',100),  
(02,'2024-5-10',91,'2024-5-15',200),  
(03,'2024-6-25',92,'2024-5-30',300),  
(04,'2024-8-9',93,'2024-8-19',400);
```

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

Information

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- coursesdb
- firsttbs
- hotelmanagement
- lab2
- mydatabase
- sakila
- sys

Administration Schemas

JANUARY 2024

# PHASE 3

## SELECT command :

Where :

```
1 • SELECT Name,job_title,salary  
2   from hotelemployee  
3 WHERE salary > 200;  
4  
5  
6  
7
```

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

Name	job_title	salary
sara	Receptionist	9000.00
seba	Maintenance Supervisor	2000.00
lama	Reservation Agent	5000.00
mary	Restaurant Waiter	87200.00

Order by :

```
1 • SELECT Name,phonenumber  
2   from customers  
3 ORDER BY IDnumber DESC;  
4  
5  
6  
7  
8
```

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

Name	phonenumber
sama	05553
salf	05797
jood	05643
walaa	05673

# PHASE 3

## SELECT command :

### group by :

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following query is written:

```
1 •  SELECT MAX(salary) as 'employee salary' , job_title
2   from hotelemployee
3   group by job_title;
```

The result grid displays the output of the query:

employee salary	job_title
9000.00	Receptionist
2000.00	Maintenance Supervisor
5000.00	Reservation Agent
87200.00	Restaurant Waiter

### WHERE

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following query is written:

```
1 •  SELECT *
2   FROM reservation
3   WHERE IDnumber = (SELECT IDnumber FROM customers WHERE Name = 'jood');
```

The result grid displays the output of the query:

SRnum	Reservation_Date	Room_number	Expiration_date	IDnumber
2	2024-05-10	91	2024-05-15	200
HULL	HULL	HULL	HULL	HULL

# PHASE 3

## SELECT command :

### Having

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a query editor window containing the following SQL code:

```
1 •  SELECT count(IDnumber) , job_title
2   from hotelemployee
3   group by job_title
4   having MIN(salary)>10;
```

Below the query editor is a results grid titled "Result Grid". It has two columns: "count(IDnumber)" and "job\_title". The data is as follows:

count(IDnumber)	job_title
1	Receptionist
1	Maintenance Supervisor
1	Reservation Agent
1	Restaurant Waiter

### order by

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a query editor window containing the following SQL code:

```
1 •  SELECT salary , job_title, IDnumber
2   from hotelemployee
3   order by IDnumber desc;
```

Below the query editor is a results grid titled "Result Grid". It has three columns: "salary", "job\_title", and "IDnumber". The data is as follows:

salary	job_title	IDnumber
87200.00	Restaurant Waiter	4
5000.00	Reservation Agent	3
2000.00	Maintenance Supervisor	2
9000.00	Receptionist	1
NULL	NULL	NULL

# PHASE 3

## LECT command :

### Join operation

SQL File 1\* x

```
1 • SELECT c.Name, r.Reservation_Date
2 FROM customers c
3 JOIN reservation r ON c.IDnumber = r.IDnumber;
```

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

	Name	Reservation_Date
▶	walaa	2024-04-12
	jood	2024-05-10
	salf	2024-06-25

Result 3 x

SQL File 1\* x

```
1 SELECT e.Name AS Employee_Name, c.Name AS Customer_Name, r.Reservation_Date
2 FROM hotelemployee e
3 JOIN customers c ON e.IDnumber = c.IDnumber_hotelemployee
4 JOIN reservation r ON c.IDnumber = r.IDnumber;
```

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

	Employee_Name	Customer_Name	Reservation_Date
▶	sara	walaa	2024-04-12
	seba	jood	2024-05-10
	lama	salf	2024-06-25

Result 2 x

Output:

