جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY
كلية علوم الحاسب وتقنية المعلومات
College of Computer Science and Information Technology

Database

2024 - second semester

	Name	ID	Role Participated in	Sign
1	Hassan Mohammed Alzourei		(Leader)	

Project Milestone#2

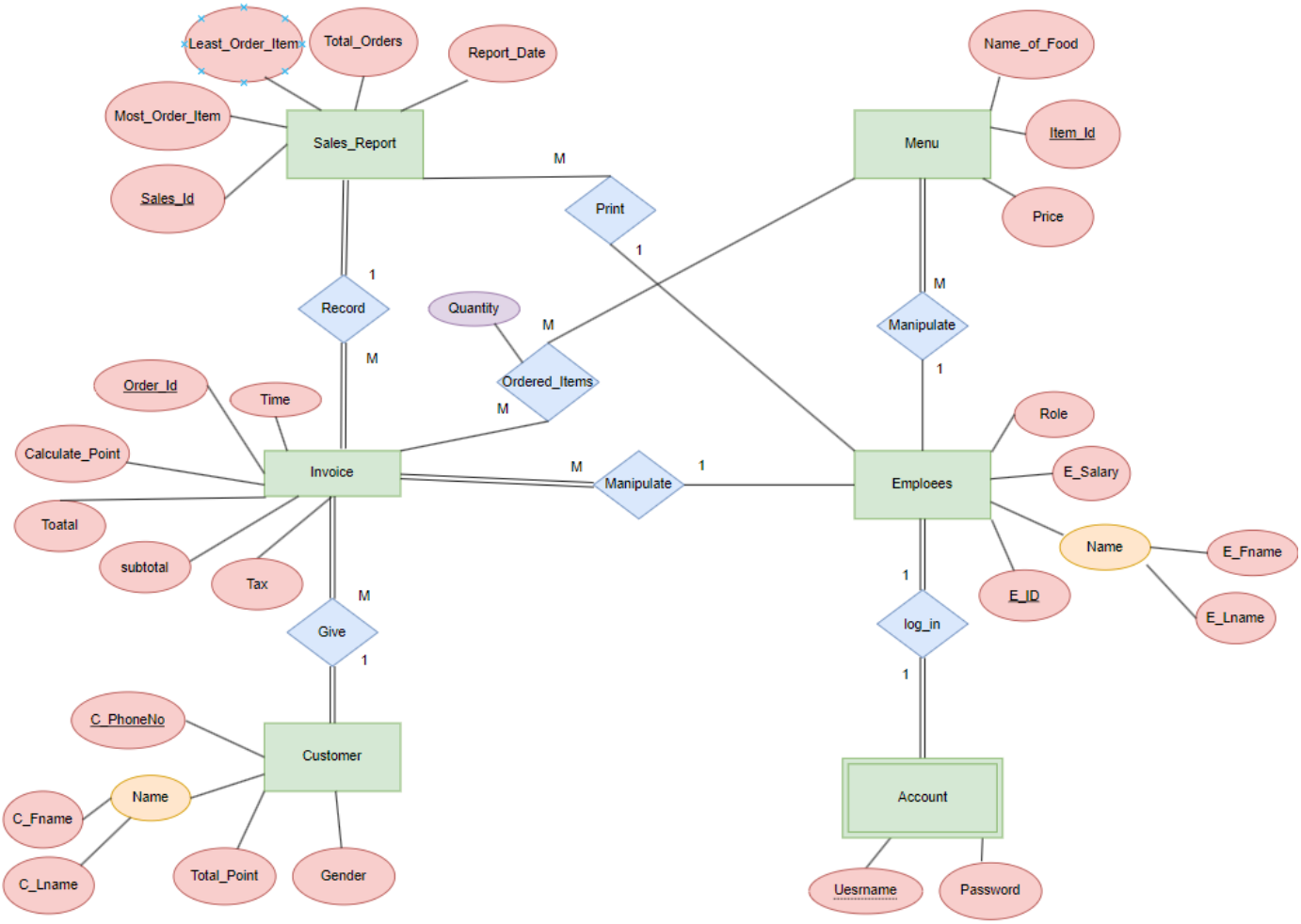
Description

-The restaurant is divided into two parts: one side is managed by the staff taking orders and the other is managed by the manager who manages the staff, the menu content, and can make salary reports.
-The cashier employee takes customer orders and information to complete the order through the system and registers customers who want to earn points to make discounts after a while of orders.
-The customer information will assist us in sending promos and points to them.
-The invoice will have information about the order like the date of order, how many items have been ordered, and information on the customer with points they earned, and so on
-The manager can access sales reports which can also know the most order items, the least order items, and the total orders. The manager can also modify the menu, and it is possible to add items and set a price for each item, or it is possible to delete items.

System Constraints:

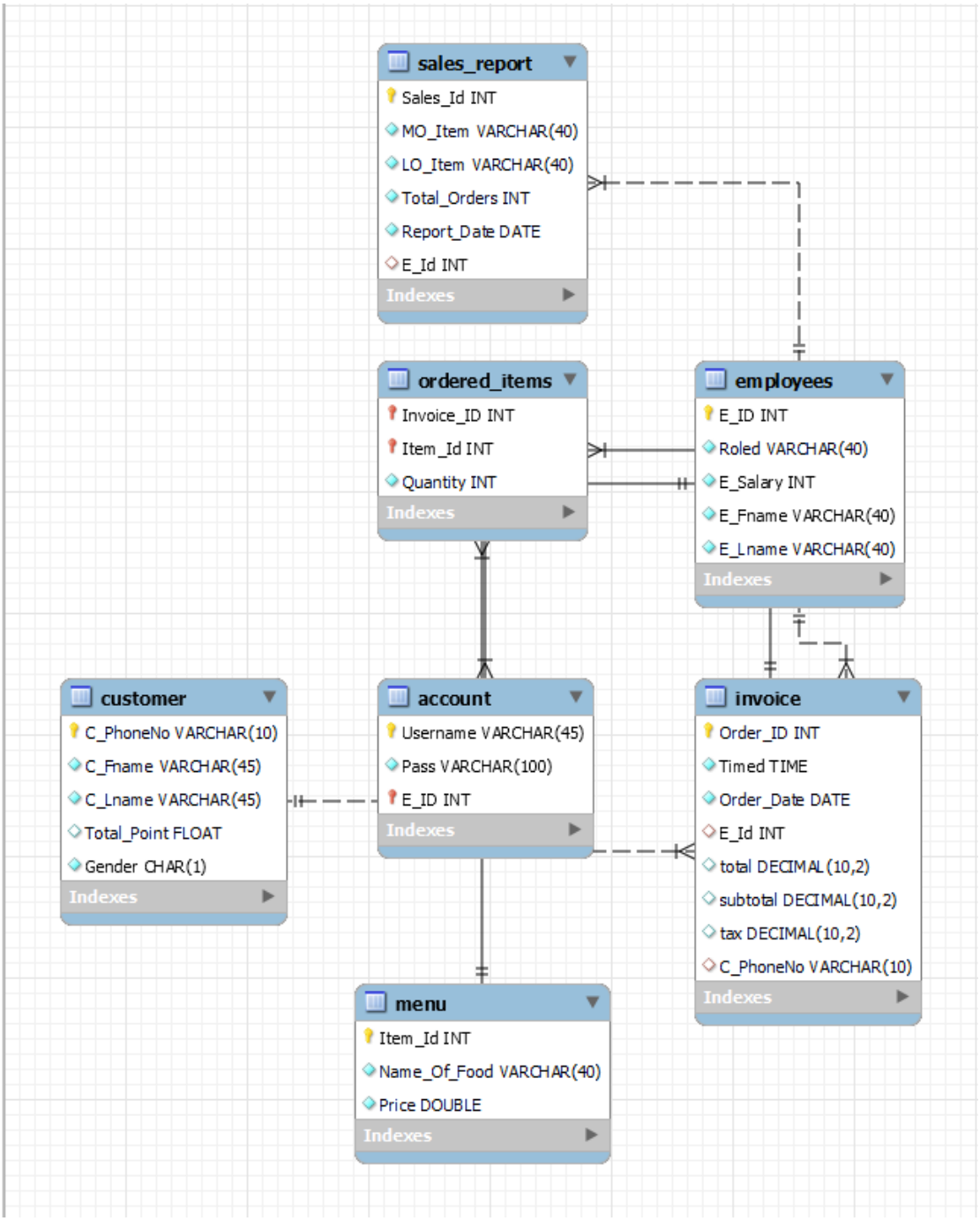
- Each employee can have one account
- Only one Employee can be the manager
- Manager can print multiple Sales reports
- Employee can create Manipulate Invoices
- Customer phone number is a primary key
- Each menu item has a limit

ER Diagram


Least_Order_Item
Total_Orders
Report_Date
Most_Order_Item
Sales_Report
Sales_Id
M
Print
1
Name_of_Food
Item_Id
Menu
Price
1
Record
M
Quantity
M
Ordered_Items
M
M
Manipulate
1
Order_Id
Time
Calculate_Point
Invoice
M
Manipulate
1
Role
E_Salary
Emploees
Toatal
subtotal
Tax
M
Give
1
Name
E_Fname
E_Lname
E_ID
1
log_in
1
C_PhoneNo
Customer
Name
C_Fname
C_Lname
Total_Point
Gender
Account
Uesrname
Password

Relation Schema

Relational Schema Mapping



Employee

E_ID	Roled	E_Salary	E_Fname	E_Lname
1000	Manager	12000	Hassan	Al-Zourei
1001	Cashier	7000	Abdulrahman	Al-Mejna
1002	Cashier	7500	Abdullah	Al-Battat
1003	Cashier	7250	Ahmed	Al-Shaikh
1004	Cashier	6500	Ali	Al-Khars
1005	Cashier	6900	Baqer	Al-Haddad
1006	Cashier	2500	Hassan	Mohsen
1007	Cashier	7500	Hassan	zayer
1008	Cashier	5000	Ali	Hassan
1009	Cashier	5000	Ali	mohammed
1010	Cashier	5555	aj	def
1012	Cashier	85000	Hassan	zoury
1013	Cashier	85000	Ali	zoury
1014	Cashier	1555	mod	mod

Account

Username	Pass	E_ID
Abdullah	am12345678	1002
Abdulrahman	DH99!@#123	1001
Ahmed	123456aa	1003
Ali	qwerty123	1004
AliHassan12	AliHassan12	1009
Baqer	abc123123	1005
Hassan	Hassan123456	1000
HMohsen123	Password123	1006
mod12345	mod12345	1014
zayer12345	zayer12345	1007

Customer

C_PhoneNo	C_Fname	C_Lname	Total_Point	Gender
0501232318	Ahmed	Yousef	138	M
0502144737	Abdulrhman	Mohsen	437	M
0512345678	Mohammed	Ahmed	2151	M
0545678901	Noor	Mohammed	180	F
0546420000	Ali	Hassan	1392.6	M
0546885516	Abdullah	Mohammed	120	M
0555926390	Mohammed	Hamad	200	M
0556789012	Yasmine	Khalid	170	F
0567890123	Omar	Saeed	220	M
0578901234	Layla	Hassan	190	F
0589012345	Rida	Nasser	210	M
0590123456	Sara	Amin	230	F

Invoice

Order_ID	Timed	Order_Date	E_Id	total	subtotal	tax	C_PhoneNo
1	17:30:30	2024-05-11	1005	13.80	12.00	1.80	NULL
2	17:36:01	2024-05-11	1005	29.90	26.00	3.90	0502144737
3	17:36:27	2024-05-11	1005	27.60	24.00	3.60	NULL
4	17:40:20	2024-05-11	1004	200.10	174.00	26.10	0512345678
5	23:42:20	2024-05-11	1004	211.60	184.00	27.60	0546420000
6	23:44:10	2024-05-11	1004	221.38	192.50	28.88	0546420000
7	23:51:13	2024-05-11	1009	75.90	66.00	9.90	NULL
8	00:07:10	2024-05-12	1009	346.15	301.00	45.15	0546420000
9	00:07:33	2024-05-12	1009	62.10	54.00	8.10	0546420000
10	00:14:35	2024-05-12	1009	200.10	174.00	26.10	0546420000
11	11:13:36	2024-05-12	1007	82.80	72.00	10.80	NULL
12	11:19:18	2024-05-12	1007	80.50	70.00	10.50	NULL

Sales_Report

Sales_Id	MO_Item	LO_Item	Total_Orders	Report_Date	E_Id
609874619	Chicken Burger	Green Tea	95	2024-03-13	1000
609875293	Chocolate Cake	Chicken Noodles	95	2024-04-01	1000
609875778	Chicken Burger	Green Tea	109	2024-03-07	1000
609876168	Chicken Burger	Green Tea	104	2024-04-01	1000
609876302	Chicken Burger	Green Tea	97	2024-03-22	1000
609876386	Chicken Burger	Rainbow	91	2024-04-01	1000
609877557	Chicken Burger	Green Tea	63	2024-04-02	1000
609878179	Chocolate Cake	Chicken Noodles	114	2024-03-25	1000
609878283	Chocolate Cake	Green Tea	86	2024-04-01	1000
609878571	Chicken Burger	Rainbow	106	2024-04-02	1000
609878572	Chicken Noodles	7up	3	2024-05-11	1000
609878573	Chicken Noodles	7up	4	2024-05-11	1000
609878574	Vegetarian Pizza	7up	7	2024-05-12	1000
609878575	Vegetarian Pizza	Chicken Burger	3	2024-05-12	1000

Menu

Item_Id	Name_Of_Food	Price
1	Chicken Noodles	25
2	Chicken Burger	18
3	Vegetarian Pizza	15
4	Chocolate Cake	50
5	Fruit Cake	60
6	Strawberry Cake	55
7	Rainbow	35
8	Chocolate Coffee	18
9	Cappuccino Coffee	17
10	Cold Coffee	13
11	Green Tea	6
12	Orange Juice	8
13	Coke	4
14	7up	2
15	Water	0.5

Ordered_items

Invoice_ID	Item_Id	Quantity
1	14	3
2	10	2
3	12	3
5	1	1
6	1	3
7	2	2
8	2	2
9	2	3
10	1	2
11	2	2
12	8	3

SQL Query Statements

SQL Query Statements (5 marks)

Write all SQL Queries required in your system to achieve all requirements (screenshot the result for each query)

Insert

```
-- Insert:
insert into Menu values
(16,'Pasta',32);
select * from Menu;
```

Item_Id	Name_Of_Food	Price
1	Chicken Noodles	25
2	Chicken Burger	18
3	Vegetarian Pizza	15
4	Chocolate Cake	50
5	Fruit Cake	60
6	Strawberry Cake	55
7	Rainbow	35
8	Chocolate Coffee	18
9	Cappuccino Coffee	17
10	Cold Coffee	13
11	Green Tea	6
12	Orange Juice	8
13	Coke	4
14	7up	2
15	Water	0.5
16	Pasta	32
NULL	NULL	NULL

Delete

```
-- Delete:
delete from menu
where item_id = 16;
select * from Menu;
```

Item_Id	Name_Of_Food	Price
1	Chicken Noodles	25
2	Chicken Burger	18
3	Vegetarian Pizza	15
4	Chocolate Cake	50
5	Fruit Cake	60
6	Strawberry Cake	55
7	Rainbow	35
8	Chocolate Coffee	18
9	Cappuccino Coffee	17
10	Cold Coffee	13
11	Green Tea	6
12	Orange Juice	8
13	Coke	4
14	7up	2
15	Water	0.5
NULL	NULL	NULL

Update

```
-- Update:
update menu
set Price =50
where Item_Id = 6;
select * from Menu;
```

Item_Id	Name_Of_Food	Price
1	Chicken Noodles	25
2	Chicken Burger	18
3	Vegetarian Pizza	15
4	Chocolate Cake	50
5	Fruit Cake	60
6	Strawberry Cake	50
7	Rainbow	35
8	Chocolate Coffee	18
9	Cappuccino Coffee	17
10	Cold Coffee	13
11	Green Tea	6
12	Orange Juice	8
13	Coke	4
14	7up	2
15	Water	0.5
NULL	NULL	NULL

Like and Not Like

```
306 •   select Name_Of_Food as FoodWithAAtEnd
307     from Menu
308     where Name_Of_Food like '%a';
```

FoodWithAAtEnd
Vegetarian Pizza
Green Tea

```
312 •   select Name_Of_Food as AllWithoutQ
313     from Menu
314     where Name_Of_Food not like '%q%';
```

AllWithoutQ
Chicken Noodles
Chicken Burger
Vegetarian Pizza
Chocolate Cake
Fruit Cake
Strawberry Cake
Rainbow
Chocolate Coffee
Cappuccino Coffee
Cold Coffee
Green Tea
Orange Juice
Coke
7up
Water

Green Tea

Between

```
select Price ,Name_Of_Food as high_cost
from Menu
where Price Between 40 and 100;
```

Price	high_cost
50	Chocolate Cake
60	Fruit Cake
50	Strawberry Cake

In

```
select*
from customer
where  Total_Point in (100,200,300,400,500);
```

C_PhoneNo	C_Fname	C_Lname	Total_Point	Gender
0555926390	Mohammed	Hamad	200	M
NULL	NULL	NULL	NULL	NULL

Order by

```
318  select Menu.Name_Of_Food , Menu.price as LowToHigh
319  from Menu
320  order by Price asc ;
```

Name_Of_Food	LowToHigh
Water	0.5
7up	2
Coke	4
Green Tea	6
Orange Juice	8
Cold Coffee	13
Vegetarian Pizza	15
Cappuccino Coffee	17
Chicken Burger	18
Chocolate Coffee	18
Chicken Noodles	25
Rainbow	35
Chocolate Cake	50
Strawberry Cake	50
Fruit Cake	60

Is null

```
380 •   select C_PhoneNo,C_Fname, C_Lname
381     from customer
382     where C_Fname is null;
```

C_PhoneNo	C_Fname	C_Lname
0555926390	NULL	Hamad
0589012345	NULL	Nasser
NULL	NULL	NULL

Intersect

NOTE// MySQL does not support Intersect directly .

```
SELECT
    CONCAT(SR1.MO_Item, ' - ', SR2.LO_Item) AS Best_State,
    SUM(SR1.Total_Orders + SR2.Total_Orders) AS Total_Sales
FROM
    Sales_Report AS SR1
JOIN
    Sales_Report AS SR2 ON SR1.MO_Item <> SR2.MO_Item AND SR1.LO_Item <> SR2.LO_Item
GROUP BY
    Best_State
ORDER BY
    Total_Sales DESC
LIMIT 3;
```

Best_State	Total_Sales
Chicken Burger - Chicken Noodles	2793
Chocolate Cake - Green Tea	1981
Chocolate Cake - Rainbow	1181

Except

NOTE// MySQL does not support Except directly .

```sql
-- NOTE// In MySQL, the Except keyword is not directly supported :
SELECT
    CONCAT(SR1.MO_Item, ' - ', SR1.LO_Item) AS Best_State,
    SUM(SR1.Total_Orders) AS Total_Sales
FROM
    Sales_Report AS SR1
LEFT JOIN
    Sales_Report AS SR2 ON SR1.MO_Item = SR2.MO_Item AND SR1.LO_Item = SR2.LO_Item AND SR1.Sales_Id != SR2.Sales_Id
WHERE
    SR2.Sales_Id IS NULL
GROUP BY
    Best_State
ORDER BY
    Total_Sales DESC
LIMIT 1;
```

	Best_State	Total_Sales
▶	Chocolate Cake - Green Tea	86

Union

```sql
-- Union:
select C_PhoneNo ,Total_Point
from customer
union
select E_ID, Roled
from  employees   ;
```

	C_PhoneNo	Total_Point
▶	0501232318	138
	0502144737	437
	0512345678	2151
	0545678901	180
	0546420000	1392.5999755859375
	0546885516	120
	0555926390	200
	0556789012	170
	0567890123	220
	0578901234	190
	0589012345	210
	0590123456	230
	1000	Manager
	1001	Cashier
	1002	Cashier
	1003	Cashier
	1004	Cashier
	1005	Cashier
	1006	Cashier
	1007	Cashier
	1008	Cashier
	1009	Cashier
	1010	Cashier
	1012	Cashier
	1013	Cashier
	1014	Cashier

Aggregate Functions

```
381 •   SELECT MAX(price)
382     FROM Menu;
```

	MAX(price)
▶	60

```
384 •   SELECT min(price)
385     FROM Menu;
```

	min(price)
▶	0.5

```
•   select count(C_PhoneNo)
    from customer;
```

	count(c_Fname)
▶	12

```
•   select avg(e_salary)
    from employees;
```

	avg(e_salary)
▶	17447.1429

```
360 •   select sum(Price)
361     from Menu;
```

	sum(Price)
▶	321.5

Aggregate Functions + Group By + Having

```
397     -- Aggregate Functions + Group By + Having:
398 •   select c_Lname, c_Fname
399     from customer
400     group by c_Lname, c_Fname
401     having avg(Total_Point) < 200
402     order by c_Lname ;
```

c_Lname	c_Fname
Hassan	Layla
Khalid	Yasmine
Mohammed	Abdullah
Mohammed	Noor
Yousef	Ahmed

NATURAL JOIN

```
•   SELECT *
    FROM employees
    NATURAL JOIN invoice ;
```

E_ID	Roled	E_Salary	E_Fname	E_Lname	Order_ID	Timed	Order_Date	total	subtotal	tax	C_PhoneNo
1005	Cashier	6900	Baqer	Al-Haddad	1	17:30:30	2024-05-11	13.80	12.00	1.80	NULL
1005	Cashier	6900	Baqer	Al-Haddad	2	17:36:01	2024-05-11	29.90	26.00	3.90	0502144737
1005	Cashier	6900	Baqer	Al-Haddad	3	17:36:27	2024-05-11	27.60	24.00	3.60	NULL
1004	Cashier	6500	Ali	Al-Khars	4	17:40:20	2024-05-11	200.10	174.00	26.10	0512345678
1004	Cashier	6500	Ali	Al-Khars	5	23:42:20	2024-05-11	211.60	184.00	27.60	0546420000
1004	Cashier	6500	Ali	Al-Khars	6	23:44:10	2024-05-11	221.38	192.50	28.88	0546420000
1009	Cashier	5000	Ali	mohammed	7	23:51:13	2024-05-11	75.90	66.00	9.90	NULL
1009	Cashier	5000	Ali	mohammed	8	00:07:10	2024-05-12	346.15	301.00	45.15	0546420000
1009	Cashier	5000	Ali	mohammed	9	00:07:33	2024-05-12	62.10	54.00	8.10	0546420000
1009	Cashier	5000	Ali	mohammed	10	00:14:35	2024-05-12	200.10	174.00	26.10	0546420000
1007	Cashier	7500	Hassan	zayer	11	11:13:36	2024-05-12	82.80	72.00	10.80	NULL
1007	Cashier	7500	Hassan	zayer	12	11:19:18	2024-05-12	80.50	70.00	10.50	NULL

Full OUTER JOIN

NOTE// MySQL does not support Full OUTER JOINdirectly .

```sql
-- FULL OUTER JOIN :
-- NOTE// MySQL does not support FULL OUTER JOIN directly .
--        However, you can simulate a full outer join using a combination of
--        LEFT OUTER JOIN, RIGHT OUTER JOIN, and UNION. Here's how you can achieve a full outer join in MySQL:
SELECT *
FROM invoice
LEFT OUTER JOIN customer ON invoice.C_PhoneNo = customer.C_PhoneNo
UNION
SELECT *
FROM invoice
RIGHT OUTER JOIN customer ON invoice.C_PhoneNo = customer.C_PhoneNo
WHERE invoice.C_PhoneNo IS NULL;
```

Order_ID	Timed	Order_Date	E_Id	total	subtotal	tax	C_PhoneNo	C_PhoneNo	C_Fname	C_Lname	Total_Point	Gender
1	17:30:30	2024-05-11	1005	13.80	12.00	1.80	NULL	NULL	NULL	NULL	NULL	NULL
2	17:36:01	2024-05-11	1005	29.90	26.00	3.90	0502144737	0502144737	Abdulrhman	Mohsen	437	M
3	17:36:27	2024-05-11	1005	27.60	24.00	3.60	NULL	NULL	NULL	NULL	NULL	NULL
4	17:40:20	2024-05-11	1004	200.10	174.00	26.10	0512345678	0512345678	Mohammed	Ahmed	2151	M
5	23:42:20	2024-05-11	1004	211.60	184.00	27.60	0546420000	0546420000	Ali	Hassan	1392.6	M
6	23:44:10	2024-05-11	1004	221.38	192.50	28.88	0546420000	0546420000	Ali	Hassan	1392.6	M
7	23:51:13	2024-05-11	1009	75.90	66.00	9.90	NULL	NULL	NULL	NULL	NULL	NULL
8	00:07:10	2024-05-12	1009	346.15	301.00	45.15	0546420000	0546420000	Ali	Hassan	1392.6	M
9	00:07:33	2024-05-12	1009	62.10	54.00	8.10	0546420000	0546420000	Ali	Hassan	1392.6	M
10	00:14:35	2024-05-12	1009	200.10	174.00	26.10	0546420000	0546420000	Ali	Hassan	1392.6	M
11	11:13:36	2024-05-12	1007	82.80	72.00	10.80	NULL	NULL	NULL	NULL	NULL	NULL
12	11:19:18	2024-05-12	1007	80.50	70.00	10.50	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0501232318	Ahmed	Yousef	138	M
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0545678901	Noor	Moham...	180	F
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0546885516	Abdullah	Moham...	120	M
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0555926390	NULL	Hamad	200	M
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0556789012	Yasmine	Khalid	170	F
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0567890123	Omar	Saeed	220	M
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0578901234	Layla	Hassan	190	F
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0589012345	NULL	Nasser	210	M
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0590123456	Sara	Amin	230	F

Exist

```sql
329 •   select * from  invoice
330     where exists
331     (select* from employees  where invoice.E_Id =  employees.E_ID );
```

Order_ID	Timed	Order_Date	E_Id	total	subtotal	tax	C_PhoneNo
1	17:30:30	2024-05-11	1005	13.80	12.00	1.80	NULL
2	17:36:01	2024-05-11	1005	29.90	26.00	3.90	0502144737
3	17:36:27	2024-05-11	1005	27.60	24.00	3.60	NULL
4	17:40:20	2024-05-11	1004	200.10	174.00	26.10	0512345678
5	23:42:20	2024-05-11	1004	211.60	184.00	27.60	0546420000
6	23:44:10	2024-05-11	1004	221.38	192.50	28.88	0546420000
7	23:51:13	2024-05-11	1009	75.90	66.00	9.90	NULL
8	00:07:10	2024-05-12	1009	346.15	301.00	45.15	0546420000
9	00:07:33	2024-05-12	1009	62.10	54.00	8.10	0546420000
10	00:14:35	2024-05-12	1009	200.10	174.00	26.10	0546420000
11	11:13:36	2024-05-12	1007	82.80	72.00	10.80	NULL
12	11:19:18	2024-05-12	1007	80.50	70.00	10.50	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Nested Query

```
440     -- Nested Query:
441     select E_ID,E_Salary,E_Fname,E_Lname
442     from employees
443     where E_Salary <
444     (select avg(E_Salary) from employees);
```

E_ID	E_Salary	E_Fname	E_Lname
1000	12000	Hassan	Al-Zourei
1001	7000	Abdulrahman	Al-Mejna
1002	7500	Abdullah	Al-Battat
1003	7250	Ahmed	Al-Shaikh
1004	6500	Ali	Al-Khars
1005	6900	Baqer	Al-Haddad
1006	2500	Hassan	Mohsen
1007	7500	Hassan	zayer
1008	5000	Ali	Hassan
1009	5000	Ali	mohammed
1010	5555	aj	def
1014	1555	mod	mod
NULL	NULL	NULL	NULL

Trigger

```
-- Trigger :
delimiter //

CREATE TRIGGER changepoint
BEFORE UPDATE ON customer
FOR EACH ROW
BEGIN
    IF NEW.Total_Point BETWEEN 100 AND 120 THEN
        SET NEW.Total_Point = NEW.Total_Point + 20;
    END IF;
END//

delimiter ;
```

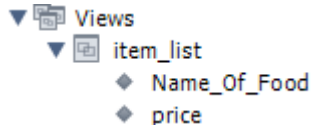
C_PhoneNo	C_Fname	C_Lname	Total_Point	Gender
0501232318	Ahmed	Yousef	138	M
0502144737	Abdulrhman	Mohsen	437	M
0512345678	Mohammed	Ahmed	2151	M
0545678901	Noor	Mohammed	180	F
0546420000	Ali	Hassan	1392.6	M
0546885516	Abdullah	Mohammed	90	M
0555926390	Mohammed	Hamad	200	M
0556789012	Yasmine	Khalid	170	F
0567890123	Omar	Saeed	220	M
0578901234	Layla	Hassan	190	F
0589012345	Rida	Nasser	140	M
0590123456	Sara	Amin	230	F
NULL	NULL	NULL	NULL	NULL

```
UPDATE `restaurant`.`customer` SET `Total_Point` = '100' WHERE (`C_PhoneNo` = '0546885516');
```

C_PhoneNo	C_Fname	C_Lname	Total_Point	Gender
0501232318	Ahmed	Yousef	138	M
0502144737	Abdulrhman	Mohsen	437	M
0512345678	Mohammed	Ahmed	2151	M
0545678901	Noor	Mohammed	180	F
0546420000	Ali	Hassan	1392.6	M
0546885516	Abdullah	Mohammed	120	M
0555926390	Mohammed	Hamad	200	M
0556789012	Yasmine	Khalid	170	F
0567890123	Omar	Saeed	220	M
0578901234	Layla	Hassan	190	F
0589012345	Rida	Nasser	140	M
0590123456	Sara	Amin	230	F
NULL	NULL	NULL	NULL	NULL

View

```
353 •   create view item_list as
354     select Name_Of_Food , price
355     from Menu ;
356 •   SELECT * FROM restaurant project.item list;
```



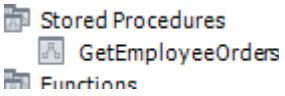
Name_Of_Food	price
Chicken Noodles	25
Chicken Burger	18
Vegetarian Pizza	15
Chocolate Cake	50
Fruit Cake	60
Strawberry Cake	50
Rainbow	35
Chocolate Coffee	18
Cappuccino Coffee	17
Cold Coffee	13
Green Tea	6
Orange Juice	8
Coke	4
7up	2
Water	0.5

CREATE PROCEDURE

```
-- CREATE PROCEDURE
DELIMITER //

CREATE PROCEDURE GetEmployeeOrders()
BEGIN
    SELECT e.E_ID, e.E_Fname, e.E_Lname, i.Order_ID, i.Order_Date
    FROM Employees e
    INNER JOIN Invoice i ON e.E_ID = i.E_Id;
END;
//
DELIMITER ;
```

```
CALL GetEmployeeOrders();
```



	E_ID	E_Fname	E_Lname	Order_ID	Order_Date
▶	1005	Baqer	Al-Haddad	1	2024-05-11
	1005	Baqer	Al-Haddad	2	2024-05-11
	1005	Baqer	Al-Haddad	3	2024-05-11
	1004	Ali	Al-Khars	4	2024-05-11
	1004	Ali	Al-Khars	5	2024-05-11
	1004	Ali	Al-Khars	6	2024-05-11
	1009	Ali	mohammed	7	2024-05-11
	1009	Ali	mohammed	8	2024-05-12
	1009	Ali	mohammed	9	2024-05-12
	1009	Ali	mohammed	10	2024-05-12
	1007	Hassan	zayer	11	2024-05-12
	1007	Hassan	zayer	12	2024-05-12

Function

```
532     DELIMITER //
533
534 •   CREATE FUNCTION salary_bonus(salary float ,total_orders INT)
535     RETURNS FLOAT deterministic
536     BEGIN
537         DECLARE total FLOAT;
538
539         IF total_orders > 99 THEN
540             SET total =salary * 0.05;
541         ELSE
542             SET total = 0;
543         END IF;
544
545         RETURN total   ;
546     END;
547     //
548     DELIMITER ;

550 •   SELECT salary_bonus(1000, 100);
```

	salary_bonus(1000, 100)
▶	50