

UJ Booth



20/03/2023
Database

Instructor:
Dr. Abdullah Alghoson

Participants:

Ahmed Hassan Alhartomi	2140826	2140826@uj.edu.sa Group Leader
Ibrahim Mohammed Alhasil	2140430	2140430@uj.edu.sa
Mohammed Ahmed Alsharif	2140290	2140290@uj.edu.sa

TT8-13719

Introduction:

Because the distance of the university far away as shown in **figure-1**. As result, lack of food services within the campus of the university causing the student to get out of the campus to get snacks or anything during his break time. This problem had wasted a lot of time from me because every time I get hungry between classes I would have to wait for a long break to go get some snacks or food in general.

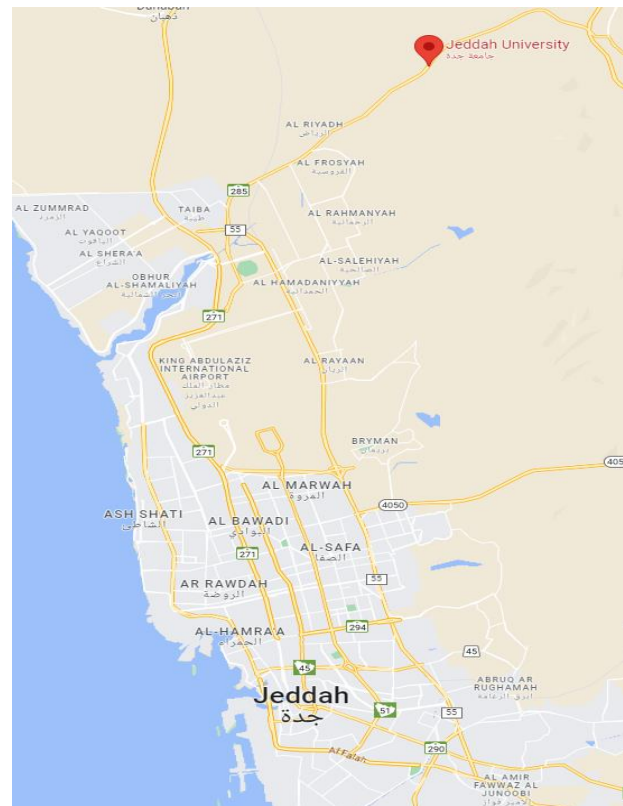


Figure-1

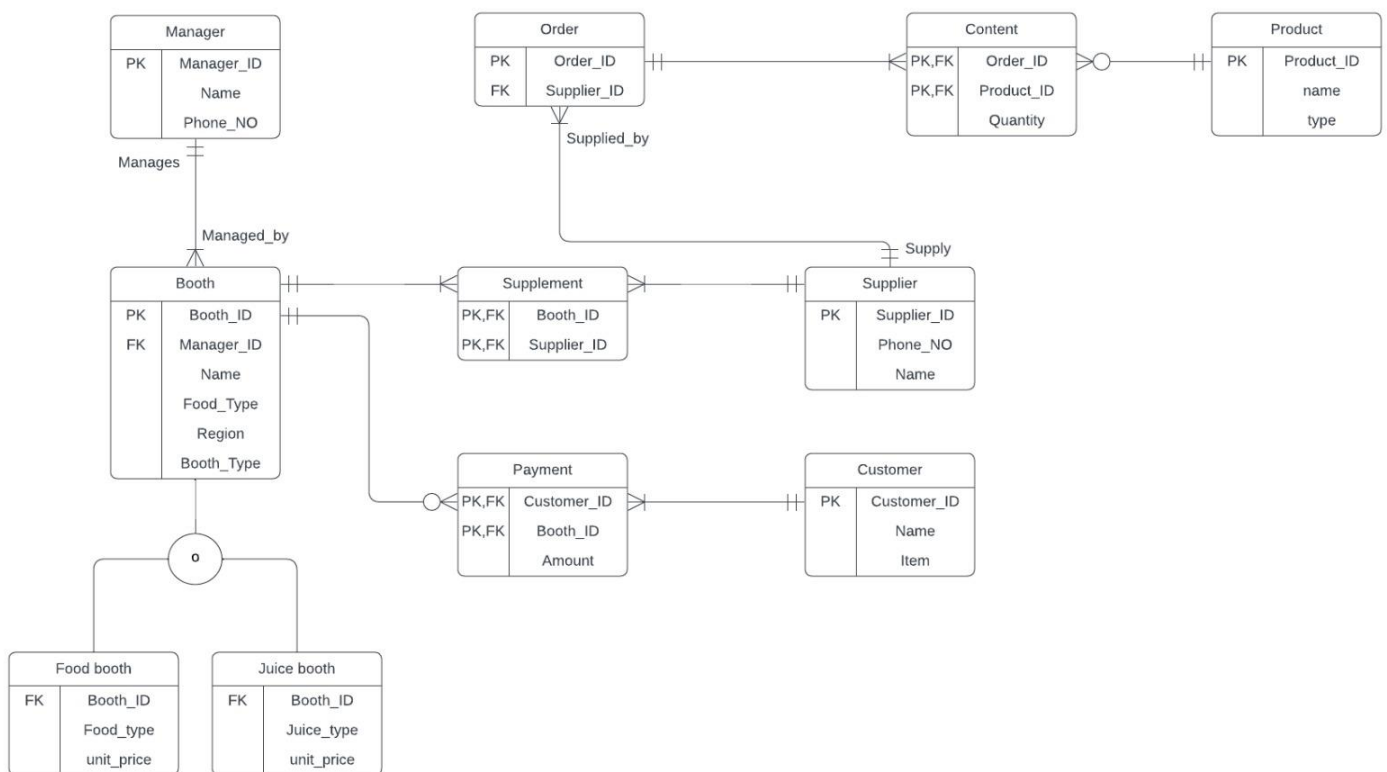
Solution:

signing contracts with companies that provide food services within a booth. For a period. This solution will help the company to attract more customer to their business causing them to gain more money. Also benefits the student to be able to acquire variety types of food.

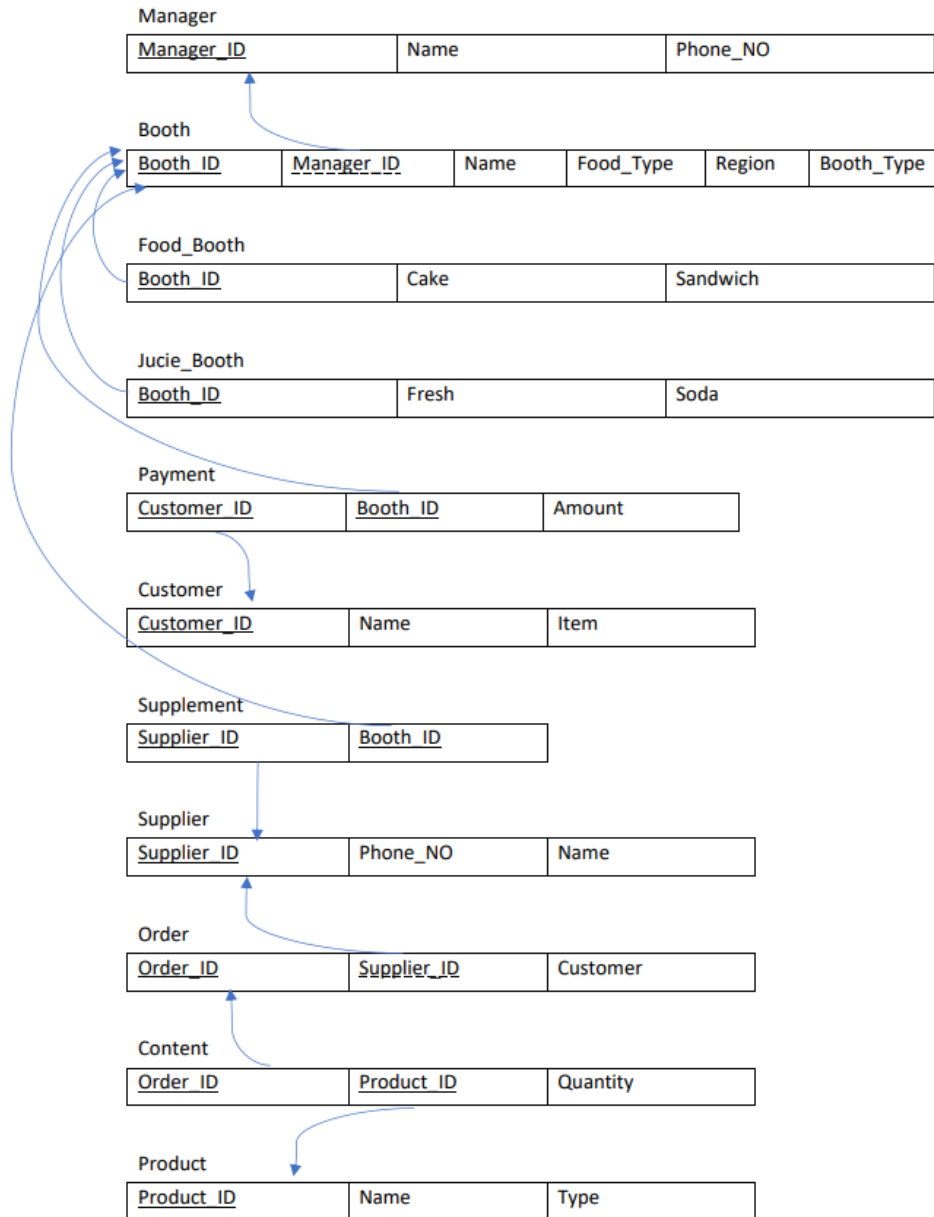
Information needs:

- University address
 - Booth location
 - Name of company that provide the food
 - Food type
 - Number of students
 - Time
-

1-Entite-Relation Diagram:



2-Relation Schema:



3-SQL Statements:

Creating Tables:

```
create table Booth_Manager(
    manager_id varchar2(11) NOT NULL,
    manager_name varchar2(30),
    manager_number number(10),
    constraint pk_manager primary key(manager_id)
);

create table Booth(
    Booth_id varchar2(11) NOT NULL,
    manager_id varchar2(11) NOT NULL,
    Booth_name varchar2(30),
    Booth_type varchar2(20),
    Food_type varchar2(20),
    Region varchar(20),
    constraint pk_booth primary key(Booth_id),
    constraint fk_manager foreign key (manager_id) references Booth_Manager (manager_id)
);

create table Food_Booth(
    Booth_id varchar2(11) NOT NULL,
    Food_type varchar2(20),
    unit_pr number(10),
    constraint pk_booth1 primary key(Booth_id),
    constraint fk_booth1 foreign key (Booth_id) references Booth(Booth_id)
);

create table juice_Booth(
    Booth_id varchar2(11) NOT NULL,
    Juice_type varchar2(20),
    unit_pr number(10),
    constraint pk_booth2 primary key(Booth_id),
    constraint fk_booth2 foreign key (Booth_id) references Booth(Booth_id)
);

create table customer(
    customer_id varchar2(11) not null,
    customer_name varchar2(30),
    customer_item varchar2(20),
    constraint pk_customer primary key(customer_id)
);

create table payment(
    customer_id varchar2(11) not null,
    booth_id varchar2(11) not null,
    amount number(10),

    constraint fk_customer foreign key (customer_id) references customer(customer_id),
    constraint fk_booth3 foreign key (booth_id) references booth(booth_id),
    constraint pk_payment primary key(customer_id,booth_id)
);

create table supplier(
    supplier_id varchar2(11) not null,
    supplier_number number(10),
    supplier_name varchar2(30),
    constraint pk_supplier primary key(supplier_id)
);
```

```

create table supplement(
  supplier_id varchar2(11) not null,
  booth_id varchar2(11) not null,

  constraint fk_supplier foreign key(supplier_id) references supplier(supplier_id),
  constraint fk_booth4 foreign key(booth_id) references booth(booth_id),
  constraint pk_supplement primary key(supplier_id,booth_id)
);

create table Orders(
  order_id varchar2(11) not null,
  supplier_id varchar2(11) not null,
  constraint fk_supplier1 foreign key(supplier_id) references supplier(supplier_id),
  constraint pk_order primary key(order_id)
);

create table product(
  product_id varchar2(11) not null,
  product_name varchar2(30),
  product_type varchar2(20),
  constraint pk_product primary key(product_id)
);

create table order_content(
  order_id varchar2(11) not null,
  product_id varchar2(11) not null,
  quantity number(10),

  constraint fk_order foreign key(order_id) references orders(order_id),
  constraint fk_product foreign key(product_id) references product(product_id),
  constraint pk_content primary key(order_id,product_id)
);

```

Inserting Data in tables:

Booth Manager:

```

INSERT INTO booth_manager
VALUES ('1','Ahmed',0536053410);
INSERT INTO booth_manager
VALUES ('2','Hassan',0536053421);
INSERT INTO booth_manager
VALUES ('3','Mohammed',0536053432);
INSERT INTO booth_manager
VALUES ('4','Khaled',0536053443);
INSERT INTO booth_manager
VALUES ('5','Saeed',0536053454);
INSERT INTO booth_manager
VALUES (6,'Abdullah',536053434);

```

Booth:

```

INSERT INTO booth
VALUES ('1','1','Foody','cake','Food_booth','New Building');
INSERT INTO booth
VALUES ('2','3','AkIk','baked','Food_booth','New Building');
INSERT INTO BOOTH (BOOTH_ID, MANAGER_ID, BOOTH_NAME, BOOTH_TYPE, FOOD_TYPE, REGION) VALUES ('3',
'2', '2G0', 'Food_booth', 'fast food', 'New Building');
INSERT INTO BOOTH (BOOTH_ID, MANAGER_ID, BOOTH_NAME, BOOTH_TYPE, FOOD_TYPE, REGION) VALUES ('4',
'5', 'HPYCOOK', 'Food_booth', 'Healthy Food', 'New Building');
INSERT INTO BOOTH (BOOTH_ID, MANAGER_ID, BOOTH_NAME, BOOTH_TYPE, FOOD_TYPE, REGION) VALUES ('5',
'4', 'COFY', 'jucie_booth', 'Coffee', 'New Building' );

```

Food Booth:

```

INSERT INTO FOOD_BOOTH(BOOTH_ID, FOOD_TYPE, UNIT_PR) VALUES ('1', 'CAKE' ,150);
INSERT INTO FOOD_BOOTH(BOOTH_ID, FOOD_TYPE, UNIT_PR) VALUES ('2', 'Baked' ,15);
INSERT INTO FOOD_BOOTH(BOOTH_ID, FOOD_TYPE, UNIT_PR) VALUES ('3', 'Fast Food' ,32);
INSERT INTO FOOD_BOOTH(BOOTH_ID, FOOD_TYPE, UNIT_PR) VALUES ('4', 'Healthy Food' ,40);

```

Juice Booth:

```

INSERT INTO JUICE_BOOTH(BOOTH_ID, JUICE_TYPE, UNIT_PR) VALUES ('1', 'V60', 30);
INSERT INTO JUICE_BOOTH(BOOTH_ID, JUICE_TYPE, UNIT_PR) VALUES ('2', 'Tea', 1.5);
INSERT INTO JUICE_BOOTH(BOOTH_ID, JUICE_TYPE, UNIT_PR) VALUES ('3', 'Soft Drinks', 3);
INSERT INTO JUICE_BOOTH(BOOTH_ID, JUICE_TYPE, UNIT_PR) VALUES ('4', 'Orange Juice', 12);
INSERT INTO JUICE_BOOTH(BOOTH_ID, JUICE_TYPE, UNIT_PR) VALUES ('5', 'Coffee', 25);

```

Supplier:

```

INSERT INTO SUPPLIER (SUPPLIER_ID, SUPPLIER_NUMBER, SUPPLIER_NAME) VALUES ('4', '555555504',
'2FAST');
INSERT INTO SUPPLIER (SUPPLIER_ID, SUPPLIER_NUMBER, SUPPLIER_NAME) VALUES ('5', '555555515',
'COHE');
INSERT INTO SUPPLIER (SUPPLIER_ID, SUPPLIER_NUMBER, SUPPLIER_NAME) VALUES ('3', '555555503', 'Food
order delivery');
INSERT INTO SUPPLIER (SUPPLIER_ID, SUPPLIER_NUMBER, SUPPLIER_NAME) VALUES ('2', '555555502',
'JuiceDel');
INSERT INTO SUPPLIER (SUPPLIER_ID, SUPPLIER_NUMBER, SUPPLIER_NAME) VALUES ('1', '555555501',
'FoodDel');

```


Customer:

```

INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ITEM) VALUES ('501', 'Mazen', 'Burger Meal + Pepsi');
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ITEM) VALUES ('502', 'Abdullah', 'Cake');
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ITEM) VALUES ('503', 'Ammar', 'V60');
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ITEM) VALUES ('505', 'Ibrahim', 'Cold brew');
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ITEM) VALUES ('506', 'Abdulmajeed', '3 Salads');

```

Payment:

```

INSERT INTO PAYMENT (CUSTOMER_ID, BOOTH_ID, AMOUNT) VALUES ('501', '3', 37);
INSERT INTO PAYMENT (CUSTOMER_ID, BOOTH_ID, AMOUNT) VALUES ('502', '1', 50);
INSERT INTO PAYMENT (CUSTOMER_ID, BOOTH_ID, AMOUNT) VALUES ('503', '5', 25);
INSERT INTO PAYMENT (CUSTOMER_ID, BOOTH_ID, AMOUNT) VALUES ('505', '5', 25);
INSERT INTO PAYMENT (CUSTOMER_ID, BOOTH_ID, AMOUNT) VALUES ('506', '4', 120);

```

Product:

```

INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_NAME, PRODUCT_TYPE) VALUES ('1', 'Orange juice', 'juice');
INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_NAME, PRODUCT_TYPE) VALUES ('2', 'Apple juice', 'juice');
INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_NAME, PRODUCT_TYPE) VALUES ('3', 'Mango juice', 'juice');
INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_NAME, PRODUCT_TYPE) VALUES ('4', 'English Cake', 'cake');
INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_NAME, PRODUCT_TYPE) VALUES ('5', 'cup Cake', 'cake');
INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_NAME, PRODUCT_TYPE) VALUES ('6', 'cookies', 'cake');

```

Orders:

```

INSERT INTO ORDERS(ORDER_ID, SUPPLIER_ID) VALUES ('1', '1');
INSERT INTO ORDERS(ORDER_ID, SUPPLIER_ID) VALUES ('2', '3');
INSERT INTO ORDERS(ORDER_ID, SUPPLIER_ID) VALUES ('3', '2');
INSERT INTO ORDERS(ORDER_ID, SUPPLIER_ID) VALUES ('4', '2');
INSERT INTO ORDERS(ORDER_ID, SUPPLIER_ID) VALUES ('5', '2');

```

Order Content:

```

INSERT INTO ORDER_CONTENT(ORDER_ID, PRODUCT_ID, QUANTITY) VALUES ('1', '5', '10');
INSERT INTO ORDER_CONTENT(ORDER_ID, PRODUCT_ID, QUANTITY) VALUES ('2', '4', '5');
INSERT INTO ORDER_CONTENT(ORDER_ID, PRODUCT_ID, QUANTITY) VALUES ('3', '1', '10');
INSERT INTO ORDER_CONTENT(ORDER_ID, PRODUCT_ID, QUANTITY) VALUES ('4', '2', '10');
INSERT INTO ORDER_CONTENT(ORDER_ID, PRODUCT_ID, QUANTITY) VALUES ('5', '3', '15');

```

Supplement:

```

INSERT INTO SUPPLEMENT (SUPPLIER_ID, BOOTH_ID) VALUES ('1', '1');
INSERT INTO SUPPLEMENT (SUPPLIER_ID, BOOTH_ID) VALUES ('1', '4');
INSERT INTO SUPPLEMENT (SUPPLIER_ID, BOOTH_ID) VALUES ('5', '5');
INSERT INTO SUPPLEMENT (SUPPLIER_ID, BOOTH_ID) VALUES ('4', '3');
INSERT INTO SUPPLEMENT (SUPPLIER_ID, BOOTH_ID) VALUES ('3', '2');

```

Queries:

```
1)
select booth_id,Manager_id,booth_name
from booth
where booth_id = 1;

2)
select customer_id,customer_name
from customer
where customer_item is not null
order by customer_name;

3)
SELECT product_id
FROM order_content INNER JOIN orders ON order_content.order_id = orders.order_id
where supplier_id = 1;

4)
select customer_id, sum(amount) as Total from payment group by customer_id;

5)
select customer_name,customer_item from customer
where customer_name like 'I%' or customer_name like 'A%' order by customer_name desc;

6)
SELECT MANAGER_NAME FROM BOOTH_MANAGER WHERE MANAGER_ID IN (SELECT MANAGER_ID FROM BOOTH WHERE
REGION = 'New Building');
```

Procedure:

1) a PARAMETER based SELECT QUERY stored procedure which return records based on parameters:

```

create or replace procedure getProductInfo(
    p_id in product.product_id%type,
    p_name out product.product_name%type,
    p_type out product.product_type%type
)
is
begin
    select product_name,product_type
    into p_name,p_type
    from product
    where product_id = p_id;
end;

```

used after creation procedure:

```

declare
    p_name product.product_name%type;
    p_type product.product_type%type;
begin
    getProductInfo(1,p_name,p_type);
    dbms_output.put_line('Prodcut Name:' ||p_name );
    dbms_output.put_line('Prodcut Type:' ||p_type );
end;

```

2)UPDATE query based stored procedure:

```

CREATE OR REPLACE PROCEDURE updateManager(
    m_id IN booth.manager_id%TYPE,
    b_id IN booth.BOOTH_ID%TYPE
)
IS
BEGIN

    UPDATE booth SET manager_id = m_id where booth_id = b_id;

    COMMIT;

END;

```

used after creation procedure:

```

execute updateManager(6,5);

```

5-Results:

Tables after population:

Booth Manager:

MANAGER_ID	MANAGER_NAME	MANAGER_NUMBER
1	Ahmed	536053410
2	Hassan	536053421
3	Mohammed	536053432
4	Khaled	536053443
5	Saeed	536053454
6	Abdullah	536053434

Booth:

BOOTH_ID	MANAGER_ID	BOOTH_NAME	BOOTH_TYPE	FOOD_TYPE	REGION
1	1	Foody	cake	Food_booth	New Building
2	3	Aklk	baked	Food_booth	New Building
3	2	2G0	Food_booth	fast food	New Building
4	5	HPYCOOK	Food_booth	Healthy Food	New Building
5	6	COFy	jucie_booth	Coffee	New Building

Food Booth:

BOOTH_ID	FOOD_TYPE	UNIT_PR
1	CAKE	150
2	Baked	15
3	Fast Food	32
4	Healthy Food	40

Juice Booth:

BOOTH_ID	JUICE_TYPE	UNIT_PR
1	V60	30
2	Tea	2
3	Soft Drinks	3
4	Orange Juice	12
5	Coffee	25

Customer:

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ITEM
501	Mazen	Burger Meal + Pepsi
502	Abdullah	Cake
503	Ammar	V60
505	Ibrahim	Cold brew
506	Abdulmajeed	3 Salads

Payment:

CUSTOMER_ID	BOOTH_ID	AMOUNT
501	3	37
502	1	50
503	5	25
505	5	25
506	4	120

Orders:

ORDER_ID	SUPPLIER_ID
1	1
2	3
3	2
4	2
5	2

Product:

PRODUCT_ID	PRODUCT_NAME	PRODUCT_TYPE
1	Orange juice	juice
2	Apple juice	juice
3	Mango juice	juice
4	English Cake	cake
5	cup Cake	cake
6	cookies	cake

Order Content:

ORDER_ID	PRODUCT_ID	QUANTITY
1	5	10
2	4	5
3	1	10
4	2	10
5	3	15

Supplement:

SUPPLIER_ID	BOOTH_ID
1	1
1	4
3	2
4	3
5	5

Supplier:

SUPPLIER_ID	SUPPLIER_NUMBER	SUPPLIER_NAME
4	555555504	2FAST
5	555555515	COHE
3	555555503	Food order delivery
2	555555502	JuiceDel
1	555555501	FoodDel

Queries:

1) Query which show the details of booth with specified Booth ID:

```
1 select booth_id,Manager_id,booth_name
2 from booth
3 where booth_id = 1
```

BOOTH_ID	MANAGER_ID	BOOTH_NAME
1	1	Foody

Download CSV

2) Query which show the details of customers who have items (or buy product), and order result by customers name:

```
1 select customer_id,customer_name
2 from customer
3 where customer_item is not null
4 order by customer_name
```

CUSTOMER_ID	CUSTOMER_NAME
502	Abdullah
506	Abdulmajeed
503	Ammar
505	Ibrahim
501	Mazen

Download CSV

5 rows selected.

3) Query which join two tables by order ID and show the product ID by specified supplier ID:

```
1 SELECT product_id
2 FROM order_content INNER JOIN orders ON order_content.order_id = orders.order_id
3 where supplier_id = 1
```

PRODUCT_ID

5

Download CSV

4) Query which sum the total amount of customer and group result by customer ID:

```
1 select customer_id, sum(amount) as Total from payment group by customer_id
2
```

CUSTOMER_ID	TOTAL
501	37
502	50
506	120
503	25
505	25

Download CSV

5 rows selected.

5)Query which customers name and customers items, where customer name starts with I or A, and order result in descending order and by customer name:

```
1 select customer_name,customer_item from customer
2 where customer_name like 'I%' or customer_name like 'A%' order by customer_name desc
```

CUSTOMER_NAME	CUSTOMER_ITEM
Ibrahim	Cold brew
Ammar	V60
Abdulmajeed	3 Salads
Abdullah	Cake

Download CSV

4 rows selected.

6)Query which show the name of managers who manages booth in the new building:

```
1 SELECT MANAGER_NAME FROM BOOTH_MANAGER WHERE MANAGER_ID IN (SELECT MANAGER_ID FROM BOOTH WHERE REGION = 'New Building')
2
```

MANAGER_NAME
Ahmed
Mohammed
Hassan
Saeed
Abdullah

Download CSV

5 rows selected.

Procedures:

I) Procedure returns the name and type of product by specified product ID:

Procedure Creation:

```
1  create or replace procedure getProductInfo(  
2      p_id in product.product_id%type,  
3      p_name out product.product_name%type,  
4      p_type out product.product_type%type  
5  )  
6  is  
7      begin  
8          select product_name,product_type  
9              into p_name,p_type  
10             from product  
11             where product_id = p_id;  
12  
13  end;
```

Procedure created.

Procedure Execution:

```
1  declare  
2      p_name product.product_name%type;  
3      p_type product.product_type%type;  
4  begin  
5      getProductInfo(1,p_name,p_type);  
6      dbms_output.put_line('Prodcut Name:' ||p_name );  
7      dbms_output.put_line('Prodcut Type:' ||p_type );  
8  end;
```

Statement processed.
Prodcut Name:Orange juice
Prodcut Type:juice

2) Procedure updates the manager of the booth:

Procedure Creation:

```
1 CREATE OR REPLACE PROCEDURE updateManager(  
2     m_id IN booth.manager_id%TYPE,  
3     b_id IN booth.BOOTH_ID%TYPE  
4 )  
5 IS  
6 BEGIN  
7  
8     UPDATE booth SET manager_id = m_id where booth_id = b_id;  
9  
10    COMMIT;  
11  
12 END;
```

Procedure created.

Procedure Execution:

```
1 begin  
2     updateManager(6,5);  
3 end;
```

Statement processed.

Index:

- 1- Introduction & Information: 2
- 2- ER-Diagram: 4
- 3- Relational Schema: 5
- 4- SQL Statements: 6
- 5- Results: 12