

IN THE NAME OF ALLAH, THE MOST GREATEST THE MOST MERCIFUL

**INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**FINAL LAB PROJECT**

**CANTEEN AUTOMATION SYSTEM**

**Name** : Sanjida Afrin  
**Student ID** : C201248  
**Email ID** : c201248@ugrad.iiuc.ac.bd  
**Session** : Autumn-21  
**Semester** : 4th  
**Section** : 4BF  
**Course Code** : CSE-2424  
**Course Title** : Database Management System Lab  
**Submission Date** : 24-05-22

**Submitted To:**

**Mohammad Aman Ullah**

**Associate professor**

**Department of CSE, IIUC**

---

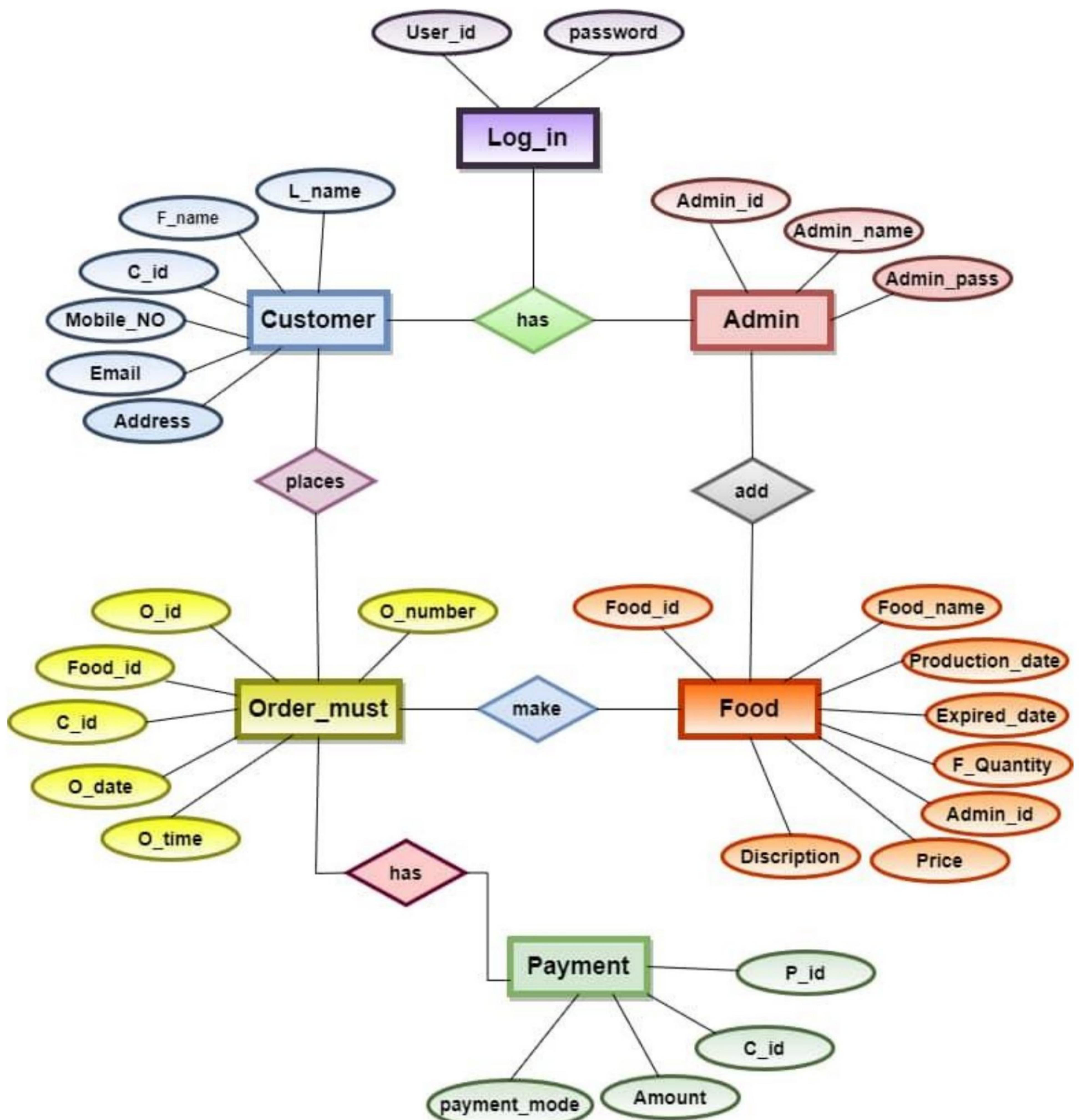
Project Title :

## CANTEEN AUTOMATION SYSTEM

**Abstract :** The purpose of "**Canteen Automation System**" is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling the requirements, so that their valuable data can be stored for a longer period with easy accessing and manipulation of the same. The required hardware and software are freely available and easy to work with. Canteen Automation System using Android, as described above, can lead to error free, secure, reliable and faster to manage the system. It can assist the user to focus on their other activities rather than focusing on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by data that is not relevant, while being able to reach the data. The aim is to automate its existing system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data can be stored for a longer period with easy accessing and manipulation of the same.

**Introduction :** Lots of time is spent in queues at a college canteen. The proposed software is effective in cutting the amount of time spent in the queue to send orders straight into the kitchen, placing orders before and with the option to use a card payment system that reduces time spent in tendering changes. This time can be used for any other purpose that must be relevant. Cash payment in canteens is nowadays the only option for making the payment. The situation was well adapted for small canteens. For some large canteens, credit card facilities are offered but this is very unusual. The major downside of cash payment system is that the user seems to want to carry the cash all time. And he / she has to pay the correct sum else there might be issues in providing the balance money. The owner should hold some register as he needs to provide the consumers of the canteen via coupons.

## ER Model Diagram :



## Schema :

1. **LOG\_IN** (User\_id,Password)
2. **ADMIN** (Admin\_id,Admin\_name,Admin\_pass)
3. **CUSTOMER** (C\_id,F\_name,L\_name,Mobile\_NO,Email,Address)
4. **ORDER\_MUST** (O\_id,Food\_id,C\_id,O\_number,O\_date,O\_time)
5. **FOOD** (Food\_id,Food\_name,Production\_date,Expired\_date,F\_Quantity, Admin\_id,Price,Discription)
6. **Payment** (P\_id,C\_id,Amount,Payment\_mode)

## DDL Statement & Tables with Data :

### LOG\_IN

```
CREATE TABLE "LOG_IN"
(
    "USER_ID" VARCHAR2(50) NOT NULL ENABLE,
    "PASSWORD" VARCHAR2(50) NOT NULL ENABLE,
    CONSTRAINT "LOG_IN_PK" PRIMARY KEY ("USER_ID") ENABLE
)
```

LOG_IN		
Table	Data	Indexes
<a href="#">Query</a> <a href="#">Count Rows</a> <a href="#">Insert Row</a>		
EDIT	USER_ID	PASSWORD
	Sanjana_2456	cue@9i\$J
	Jinia_653	qu1Hik9q
	Marowa_8190	sO_jn@pKq
	Emon_7637	oe9\$jbnQ
	Sanaya_0483	vxUj2pQw
	Tanvir_0128	jghd&Gsd
	Disti_876	a_js#GReL
	Sazzad_8823	jkh@7huh
	Sahariar_872	nsiF#Hjk
	Tasfia_9811	bvG_g4Kr
row(s) 1 - 10 of 10		
<a href="#">Download</a>		

**ADMIN**

```
CREATE TABLE "ADMIN"
(
    "ADMIN_ID" VARCHAR2(50),
    "ADMIN_NAME" VARCHAR2(50),
    "ADMIN_PASS" VARCHAR2(50),
    CONSTRAINT "ADMIN_PK" PRIMARY KEY ("ADMIN_ID") ENABLE
)
```

**ADMIN**

[Table](#) [Data](#) [Indexes](#) [Model](#) [Constraints](#) [Grants](#) [Statistics](#) [UI Defaults](#) [Triggers](#) [Dependencies](#) [SQL](#)

[Query](#) [Count Rows](#) [Insert Row](#)

EDIT	ADMIN_ID	ADMIN_NAME	ADMIN_PASS
	A1006	Sumaiya Tabassum	Sumaiya_Tabassum
	A1012	Sanjida Alam	Sanjida_Alam
	A1010	Lamisa Mashiat	Lamisa_Mashiat
	A1019	Waziha Kabir	Waziha_Kabir
	A1021	Afrina Noor	Afrina_Noor
	A1002	Sanjida Afrin	Sanjida_Afrin
	A1009	Asif Chowdhury	Asif_Chowdhury
	A1015	Masud Haider	Masud_Haider
	A1017	Jihan Mirja	Jihan_Mirja
	A1025	Umme Habiba	Umme_Habiba

row(s) 1 - 10 of 10

[Download](#)

## CUSTOMER

```
CREATE TABLE "CUSTOMER"
(
    "C_ID" VARCHAR2(50) NOT NULL ENABLE,
    "F_NAME" VARCHAR2(50) NOT NULL ENABLE,
    "L_NAME" VARCHAR2(50) NOT NULL ENABLE,
    "MOBILE_NO" VARCHAR2(50) NOT NULL ENABLE,
    "EMAIL" VARCHAR2(50),
    "ADDRESS" VARCHAR2(50),
    CONSTRAINT "CUSTOMER_PK" PRIMARY KEY ("C_ID") ENABLE,
    CONSTRAINT "CUSTOMER_UK1" UNIQUE ("MOBILE_NO") ENABLE
)
```

**CUSTOMER**

---

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
	<a href="#">Query</a>	<a href="#">Count Rows</a>	<a href="#">Insert Row</a>							
EDIT	C_ID	F_NAME	L_NAME	MOBILE_NO	EMAIL	ADDRESS				
	C105	kanita	Aziz	01765245222	kanita767@gmail.com	Chittagong				
	C101	Afsana	Aziz	01533623246	afsana01@gmail.com	Chittagong				
	C107	Santa	Aich	01772537487	santa21@gmail.com	Barishal				
	C115	Sadia	Chowdhury	01987673651	Sadia70@gmail.com	Khulna				
	C119	Jasmin	Akter	01776537625	Jasmin06@gmail.com	Dhaka				
	C103	Farhana	Ahmed	01636527838	farhana989@gmail.com	Comilla				
	C111	Sohana	Tasneem	01898787360	Sohana055@gmail.com	sylhet				
	C118	Fairoz	Ahmed	01976571235	Fairoz76@gmail.com	Rajshahi				
	C117	Ziana	Mahnaz	01554562465	Ziana443@gmail.com	Dhaka				
	C121	Rakib	Ahsan	01876236217	Rakib6623@gmail.com	Chittagong				
row(s) 1 - 10 of 10										
<a href="#">Download</a>										

**ORDER MUST**

```
CREATE TABLE "ORDER_MUST"
(
    "O_ID" VARCHAR2(50),
    "FOOD_ID" NUMBER(8,0),
    "O_NUMBER" NUMBER(8,0),
    "C_ID" VARCHAR2(50),
    "O_DATE" DATE,
    "O_TIME" VARCHAR2(50) NOT NULL ENABLE,
    CONSTRAINT "ORDER_MUST_PK" PRIMARY KEY ("O_ID") ENABLE,
    CONSTRAINT "ORDER_MUST_CK1" CHECK (O_number>0) ENABLE,
    CONSTRAINT "PAYMENT_FK2" FOREIGN KEY ("C_ID")
        REFERENCES "CUSTOMER" ("C_ID") ENABLE,
    CONSTRAINT "ORDER_MUST_FK" FOREIGN KEY ("FOOD_ID")
        REFERENCES "FOOD" ("FOOD_ID") ENABLE
)
```

**ORDER\_MUST**

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Query Count Rows Insert Row

EDIT	O_ID	FOOD_ID	O_NUMBER	C_ID	O_DATE	O_TIME
	O-7011	5009	2	C101	15-AUG-20	07:10 PM
	O-7017	5021	13	C111	25-NOV-21	09:25 PM
	O-7021	5037	1	C117	20-JUL-21	01:20 PM
	O-7033	5016	7	C115	30-MAY-20	10:55 AM
	O-7042	5031	16	C119	21-FEB-19	11:20 AM
	O-7005	5001	6	C103	01-JAN-20	11:45 PM
	O-7015	5025	15	C105	11-DEC-19	12:30 PM
	O-7028	5004	20	C118	10-APR-22	03:50 PM
	O-7008	5013	10	C107	22-MAY-21	10:00 AM
	O-7025	5044	8	C121	24-JUN-20	2:15 PM

row(s) 1 - 10 of 10

[Download](#)

**FOOD**

```
CREATE TABLE "FOOD"
```

```
(  "FOOD_ID" NUMBER(8,2),
  "FOOD_NAME" VARCHAR2(50) NOT NULL ENABLE,
  "PRODUCTION_DATE" DATE NOT NULL ENABLE,
  "EXPIRED_DATE" DATE NOT NULL ENABLE,
  "F_QUANTITY" NUMBER(8,0),
  "ADMIN_ID" VARCHAR2(50),
  "PRICE" NUMBER(8,2),
  "DISCRIPTION" VARCHAR2(50) NOT NULL ENABLE,
  CONSTRAINT "FOOD_PK" PRIMARY KEY ("FOOD_ID") ENABLE,
  CONSTRAINT "FOOD_CK1" CHECK (price>0) ENABLE,
  CONSTRAINT "FOOD_FK" FOREIGN KEY ("ADMIN_ID")
    REFERENCES "ADMIN" ("ADMIN_ID") ENABLE
)
```

**FOOD**

[Create ▾](#)

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
	<a href="#">Query</a>	<a href="#">Count Rows</a>	<a href="#">Insert Row</a>							
EDIT	FOOD_ID	FOOD_NAME	PRODUCTION_DATE	EXPIRED_DATE	F_QUANTITY	ADMIN_ID	PRICE	DISCRIPTION		
	5001	Fried Rice	01-JAN-19	15-JAN-19	200	A1006	1000	Available		
	5013	Chicken Biriyani	01-JAN-19	09-JAN-19	20	A1012	550	Available		
	5009	Pizza	30-JAN-19	06-FEB-19	30	A1009	1700	Available		
	5021	Cake	30-DEC-19	01-JAN-20	12	A1010	5000	Available		
	5025	Beef Tikka	22-SEP-20	03-OCT-20	200	A1021	150	Available		
	5037	Juice	03-MAR-21	11-MAR-22	532	A1017	800	Available		
	5044	Sea Food	20-FEB-19	28-AUG-20	15	A1019	2500	Available		
	5004	Chicken Fry	25-MAY-19	07-APR-19	70	A1002	1500	Available		
	5016	Burger	04-JUL-19	17-AUG-19	45	A1015	445	Available		
	5031	Sandwich	11-NOV-21	20-DEC-21	130	A1025	200	Available		
										row(s) 1 - 10 of 10
<a href="#">Download</a>										

## PAYMENT

```
CREATE TABLE "PAYMENT"
(
    "P_ID" VARCHAR2(50),
    "C_ID" VARCHAR2(50),
    "AMOUNT" NUMBER(8,2),
    "PAYMENT_MODE" VARCHAR2(50),
    CONSTRAINT "PAYMENT_PK" PRIMARY KEY ("P_ID") ENABLE,
    CONSTRAINT "PAYMENT_CK1" CHECK (Amount>0) ENABLE,
    CONSTRAINT "PAYMENT_FK" FOREIGN KEY ("C_ID")
        REFERENCES "CUSTOMER" ("C_ID") ENABLE
)
```

## PAYMENT

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Query Count Rows Insert Row

EDIT	P_ID	C_ID	AMOUNT	PAYMENT_MODE
	P3004	C107	4450	Bank
	P3001	C121	6000	Cash
	P3003	C103	9765	Credit card
	P3012	C117	1899	Credit Card
	P3002	C101	1126	Bank
	P3006	C105	5000	Cash
	P3015	C118	3200	Cash
	P3021	C119	6876	Cash
	P3009	C115	7985	Bank
	P3018	C111	8486	Bank

row(s) 1 - 10 of 10

[Download](#)

## SEARCHING DATA FROM INDIVIDUAL TABLE (At least 10 ways)

**Question-1 :** Find log in information which user\_id 'Marowa\_8190'.

**Answer :**

```
SELECT*
FROM LOG_IN
WHERE USER_ID = 'Marowa_8190';
```

USER_ID	PASSWORD
Marowa_8190	sO_jn@pKq

1 rows returned in 0.00 seconds

**Question-2 :** Find the admin name starting with 'S'.

**Answer :**

```
SELECT*
FROM ADMIN
WHERE ADMIN_NAME LIKE 'S%';
```

ADMIN_ID	ADMIN_NAME	ADMIN_PASS
A1006	Sumaiya Tabassum	Sumaiya_Tabassum
A1012	Sanjida Alam	Sanjida_Alam
A1002	Sanjida Afrin	Sanjida_Afrin

3 rows returned in 0.00 seconds

**Question-3 :** Find the details about 'kanita','Farhana','Ziana'.

**Answer :**

```
SELECT*
FROM customer
WHERE F_name IN ('Afsana','Ziana','Rakib');
```

C_ID	F_NAME	L_NAME	MOBILE_NO	EMAIL	ADDRESS
C101	Afsana	Aziz	01533623246	afsana01@gmail.com	Chittagong
C117	Ziana	Mahnaz	01554562465	Ziana443@gmail.com	Dhaka
C121	Rakib	Ahsan	01876236217	Rakib6623@gmail.com	Chittagong

3 rows returned in 0.02 seconds

**Question-4 :** Find the Order information which order date between 01-JAN-19 AND 30-DEC-20.

**Answer :**

```
SELECT *
FROM ORDER_MUST
WHERE O_DATE BETWEEN '01-JAN-19' AND '30-DEC-20';
```

O_ID	FOOD_ID	O_NUMBER	C_ID	O_DATE	O_TIME
O-7011	5009	2	C101	15-AUG-20	07:10 PM
O-7033	5016	7	C115	30-MAY-20	10:55 AM
O-7042	5031	16	C119	21-FEB-19	11:20 AM
O-7005	5001	6	C103	01-JAN-20	11:45 PM

O-7015	5025	15	C105	11-DEC-19	12:30 PM
O-7025	5044	8	C121	24-JUN-20	2:15 PM

6 rows returned in 0.00 seconds

**Question-5 : Find all details from food table ,sorted by the price column.****Answer :**

```
SELECT FOOD_ID,FOOD_NAME,PRODUCTION_DATE,EXPIRED_DATE,F_QUANTITY,PRICE
FROM FOOD
ORDER BY PRICE;
```

FOOD_ID	FOOD_NAME	PRODUCTION_DATE	EXPIRED_DATE	F_QUANTITY	PRICE
5025	Beef Tikka	22-SEP-20	03-OCT-20	200	150
5031	Sandwich	11-NOV-21	20-DEC-21	130	200
5016	Burger	04-JUL-19	17-AUG-19	45	445
5013	Chicken Biryani	01-JAN-19	09-JAN-19	20	550
5037	Juice	03-MAR-21	11-MAR-22	532	800
5001	Fried Rice	01-JAN-19	15-JAN-19	200	1000
5004	Chicken Fry	25-MAY-19	07-APR-19	70	1500
5009	Pizza	30-JAN-19	06-FEB-19	30	1700
5044	Sea Food	20-FEB-19	28-AUG-20	15	2500
5021	Cake	30-DEC-19	01-JAN-20	12	5000

10 rows returned in 0.02 seconds

**Question-6 : Find the maximum order number.****Answer :**

```
SELECT MAX(o_number)
FROM Order_must;
```

MAX(O_NUMBER)
20

1 rows returned in 0.00 seconds

**Question-7 : Find the details payment table which amount greater than equal 5000.****Answer :**

```
SELECT*
FROM PAYMENT
WHERE AMOUNT>=5000;
```

P_ID	C_ID	AMOUNT	PAYMENT_MODE
P3001	C121	6000	Cash
P3003	C103	9765	Credit card
P3006	C105	5000	Cash
P3021	C119	6876	Cash
P3009	C115	7985	Bank
P3018	C111	8486	Bank

6 rows returned in 0.00 seconds

**Question -8 :** Find the Customers which Last name 'Aziz' OR Address 'Dhaka'.

**Answer :**

```
SELECT C_ID,F_NAME,L_NAME,ADDRESS
FROM CUSTOMER
WHERE L_NAME ='Aziz' OR Address='Dhaka';
```

C_ID	F_NAME	L_NAME	ADDRESS
C105	kanita	Aziz	Chittagong
C101	Afsana	Aziz	Chittagong
C119	Jasmin	Akter	Dhaka
C117	Ziana	Mahnaz	Dhaka

4 rows returned in 0.01 seconds

**Question-9 :** Find the summation of all food price.

**Answer :**

```
SELECT SUM(price)
FROM food;
```

SUM(PRICE)
13845

1 rows returned in 0.00 seconds

**Question-10 :** Select distinct values address from Customer table.

**Answer :**

```
SELECT DISTINCT Address FROM Customer;
```

ADDRESS
sylhet
Barishal
Rajshahi
Comilla
Khulna
Dhaka
Chittagong

7 rows returned in 0.08 seconds

## SEARCHING DATA FROM MULTIPLE TABLE (At least 5 ways)

**Question-1 :** Show the details whose mobile no is “01776537625” show column details cid , name , oid , food\_id, o\_date , o\_time.

Answer :

```
SELECT Customer.C_id,F_name || '_' || L_name as C_id ,O_id,Food_id,O_date,O_time
FROM Customer JOIN Order_must
ON Customer.C_id = Order_must.C_id
WHERE Mobile_No = '01776537625';
```

C_ID	C_ID	O_ID	FOOD_ID	O_DATE	O_TIME
C119	Jasmin_Akter	O-7042	5031	21-FEB-19	11:20 AM

1 rows returned in 0.00 seconds

**Question-2 :** Enter o\_id and show food\_name , food\_id , o\_date , f\_quantity , price.

Answer :

```
Select FOOD_NAME, Food.FOOD_ID, O_DATE, F_QUANTITY, PRICE
From ORDER_MUST join FOOD
On ORDER_MUST.FOOD_ID = FOOD.FOOD_ID
Where O_ID = :Enter_O_ID;
Enter : O-7021
```

FOOD_NAME	FOOD_ID	O_DATE	F_QUANTITY	PRICE
Juice	5037	20-JUL-21	532	800

1 rows returned in 0.00 seconds

**Question-3 :** Show c\_id , name , food\_id , o\_id,o\_date , o\_time.

Answer :

```
select om.c_id , f_name || '' || l_name As name , om.food_id , o_id , food_name , o_date,
o_time
from CUSTOMER c join ORDER_MUST om
ON c.c_id = om.c_id
JOIN FOOD F
ON om.food_id = F.food_id;
```

C_ID	NAME	FOOD_ID	O_ID	FOOD_NAME	O_DATE	O_TIME
C103	Farhana Ahmed	5001	O-7005	Fried Rice	01-JAN-20	11:45 PM
C107	Santa Aich	5013	O-7008	Chicken Biriyani	22-MAY-21	10:00 AM
C101	Afsana Aziz	5009	O-7011	Pizza	15-AUG-20	07:10 PM
C111	Sohana Tasneem	5021	O-7017	Cake	25-NOV-21	09:25 PM
C105	kanita Aziz	5025	O-7015	Beef Tikka	11-DEC-19	12:30 PM
C117	Ziana Mahnaz	5037	O-7021	Juice	20-JUL-21	01:20 PM
C121	Rakib Ahsan	5044	O-7025	Sea Food	24-JUN-20	2:15 PM

C118	Fairoz Ahmed	5004	O-7028	Chicken Fry	10-APR-22	03:50 PM
C115	Sadia Chowdhury	5016	O-7033	Burger	30-MAY-20	10:55 AM
C119	Jasmin Akter	5031	O-7042	Sandwich	21-FEB-19	11:20 AM

10 rows returned in 0.00 seconds

**Question-4 : Shown details who payment mode is 'bank'.**

**Answer :**

```
select*
from customer join payment
on customer.c_id = payment.c_id
where payment_mode = 'Bank';
```

C_I_D	F_NAM_E	L_NAM_E	MOBILE_N_O	EMAIL	ADDRES_S	P_I_D	C_I_D	AMOUN_T	PAYMENT_MO_DE
C107	Santa	Aich	01772537487	santa21@gmail.com	Barishal	P3004	C107	4450	Bank
C101	Afsana	Aziz	01533623246	afsana01@gmail.co m	Chittagong	P3002	C101	1126	Bank
C115	Sadia	Chowdhuy	01987673651	Sadia70@gmail.com	Khulna	P3009	C115	7985	Bank
C111	Sohana	Tasneem	01898787360	Sohana055@gmail.c om	sylhet	P3018	C111	8486	Bank

4 rows returned in 0.00 seconds

**Question-5 : Show down the details of admin \_id 'A1006'.**

**Answer :**

```
select*
from admin join food
on admin.ADMIN_ID = food.ADMIN_ID
where food.admin_id = 'A1006';
```

ADMIN_ID	ADMIN_NAME	ADMIN_PASS	FOOD_ID	FOOD_NAME	PRODUCTIO_N_DATE	EXPIRED_DATE	F_QUANTITY	ADMIN_ID	PRI_CE	DIS_CRI PTI ON
A1006	Sumaiya Tabassum	Sumaiya_T abassum	5001	Fried Rice	01-JAN-19	15-JAN-19	200	A1006	1000	Available

1 rows returned in 0.00 seconds

## SUB-QUERIES

**Question-1 : Find details whose customer id='c115'.**

**Answer ;**

```
SELECT*
FROM CUSTOMER
WHERE C_ID =(SELECT C_ID
FROM CUSTOMER
WHERE C_ID='C115');
```

C_ID	F_NAME	L_NAME	MOBILE_NO	EMAIL	ADDRESS
C115	Sadia	Chowdhury	01987673651	Sadia70@gmail.com	Khulna

1 rows returned in 0.00 seconds

**Question-2 :** Find the details of the may month from order\_must table.

**Answer :**

```
select *
from ORDER_MUST
where O_DATE IN ( select O_DATE
                  from ORDER_MUST
                  where O_DATE like '%MAY%');
```

O_ID	FOOD_ID	O_NUMBER	C_ID	O_DATE	O_TIME
O-7033	5016	7	C115	30-MAY-20	10:55 AM
O-7008	5013	10	C107	22-MAY-21	10:00 AM

2 rows returned in 0.00 seconds

**Question-3 :** Find max amount details.

**Answer :**

```
select *
from PAYMENT
where amount IN ( select max(amount)
                  from payment);
```

P_ID	C_ID	AMOUNT	PAYMENT_MODE
P3003	C103	9765	Credit card

1 rows returned in 0.00 seconds

**Question-4 :** find details whose food is less than average price.

**Answer :**

```
select*
from ADMIN join food
ON admin.ADMIN_ID = food.ADMIN_ID
where price < ( select avg(price)
                 from food);
```

ADMIN_ID	ADMIN_NAME	ADMIN_PASS	FOOD_ID	FOOD_NAME	PRODUCTION_DATE	EXPIRED_DATE	F_QUANTITY	ADMIN_ID	PRICE	DISCRIPTION
A1006	Sumaiya Tabassum	Sumaiya_Tabassum	5001	Fried Rice	01-JAN-19	15-JAN-19	200	A1006	1000	Available
A1012	Sanjida Alam	Sanjida_Alam	5013	Chicken Biriyani	01-JAN-19	09-JAN-19	20	A1012	550	Available
A1021	Afrina Noor	Afrina_Noor	5025	Beef Tikka	22-SEP-20	03-OCT-20	200	A1021	150	Available
A1017	Jihan Mirja	Jihan_Mirja	5037	Juice	03-MAR-21	11-MAR-22	532	A1017	800	Available
A1015	Masud Haider	Masud_Haider	5016	Burger	04-JUL-19	17-AUG-19	45	A1015	445	Available
A1025	Umme Habiba	Umme_Habiba	5031	Sandwich	11-NOV-21	20-DEC-21	130	A1025	200	Available

6 rows returned in 0.00 seconds

**Question-5 :** Find the food\_id,food\_name,price where food price is greater than 200

**Answer :**

```
SELECT FOOD_ID,FOOD_NAME,PRICE
FROM FOOD
WHERE PRICE > (SELECT PRICE
                 FROM FOOD
                WHERE PRICE=200);
```

FOOD_ID	FOOD_NAME	PRICE
5001	Fried Rice	1000
5013	Chicken Biryani	550
5009	Pizza	1700
5021	Cake	5000
5037	Juice	800
5044	Sea Food	2500
5004	Chicken Fry	1500
5016	Burger	445

8 rows returned in 0.00 seconds

## QUERIES WITH PL/SQL

**Question-1 :** Show the customer details whose c\_id = 'C105'

**Answer :**

Declare

```
cid customer.c_id%type;
fname customer.f_name%type;
lname customer.l_name%type;
mobil customer.MOBILE_NO%type;
em customer.email%type;
add customer.address%type;
```

Begin

```
SELECT c_id , f_name , l_name , mobile_no , email , address into cid , fname , lname ,
mobil , em , add
FROM customer
WHERE c_id = 'C105';
```

```
dbms_output.put_line('Customer c_id : ' || cid );
dbms_output.put_line('Customer Name : ' || fname || ' ' || lname);
dbms_output.put_line('Customer mobile : ' || mobil);
dbms_output.put_line('Customer email : ' || em);
dbms_output.put_line('Customer address : ' || add);
```

END;

**Output :**

```
Results Explain Describe Saved SQL History

Customer c_id      : C105
Customer Name     : kanita Aziz
Customer mobile   : 01765245222
Customer email    : kanita767@gmail.com
Customer address  : Chittagong

Statement processed.

0.01 seconds
```

**Question-2 :** show the food\_name , price , f\_quantity , o\_date , o\_time

**Answer:**

```
create or replace view "ORDER_FOOD" as
select FOOD_NAME, PRICE, F_QUANTITY, O_DATE, O_TIME
from ORDER_MUST JOIN FOOD
on ORDER_MUST.FOOD_ID = FOOD.FOOD_ID
```

**Output :**

The screenshot shows the Oracle Database SQL Developer interface. On the left, there's a sidebar with a search bar and a dropdown menu set to 'Views'. Below it, a list shows 'ORDER\_FOOD' highlighted. The main workspace has a title bar 'ORDER\_FOOD' and a menu bar with 'View', 'Code', 'Data', 'Grants', 'UI Defaults', 'Dependencies', and 'SQL'. A toolbar below the menu includes 'Compile' and 'Drop' buttons. The bottom half of the screen displays a table titled 'ORDER\_FOOD' with five columns: 'Column Name', 'Data Type', and 'Nullable'. The data rows are: FOOD\_NAME (VARCHAR2(50), No), PRICE (NUMBER(8,2), Yes), F\_QUANTITY (NUMBER(8,0), Yes), O\_DATE (DATE, Yes), and O\_TIME (VARCHAR2(50), No).

Column Name	Data Type	Nullable
FOOD_NAME	VARCHAR2(50)	No
PRICE	NUMBER(8,2)	Yes
F_QUANTITY	NUMBER(8,0)	Yes
O_DATE	DATE	Yes
O_TIME	VARCHAR2(50)	No

Declare

```
fname ORDER_FOOD.FOOD_NAME%type;
pri ORDER_FOOD.PRICE%type;
fqu ORDER_FOOD.F_QUANTITY%type;
odate ORDER_FOOD.O_DATE%type;
Otime ORDER_FOOD.O_TIME%type;
```

Begin

```
select food_name, price, F_QUANTITY, O_DATE, O_TIME into fname, pri, fqu, odate,
OTIME
```

```
from order_food
where FOOD_NAME = 'Pizza';
```

```
dbms_output.put_line('Food name : ' || fname);
dbms_output.put_line('Price : ' || pri);
dbms_output.put_line('Food Quantity : ' || fqu);
dbms_output.put_line('Order DATE : ' || odate);
dbms_output.put_line('O_TIME : ' || otime);
```

END;

Output :

Results Explain Describe Saved SQL History

```
Food name      : Pizza
Price          : 1700
Food Quantity  : 30
Order DATE    : 15-AUG-20
O_TIME         : 07:10 PM
```

```
Statement processed.
```

```
0.00 seconds
```

**Question-3 :** Show details all the admin in database using plsql.

Answer :

Declare

```
CURSOR admin_cur is
SELECT ADMIN_ID, ADMIN_NAME ,ADMIN_PASS
FROM admin;
project_rec admin_cur%rowtype;
```

**Begin**

```
OPEN admin_cur;
LOOP
FETCH admin_cur into project_rec;
EXIT WHEN admin_cur%notfound;
DBMS_OUTPUT.put_line('Admain id    :' || project_rec.admin_id);
DBMS_OUTPUT.put_line('Admain name   :' || project_rec.admin_name);
DBMS_OUTPUT.put_line('Admin password : ' || project_rec.admin_pass);
DBMS_OUTPUT.put_line(' ');
END LOOP;
```

END;

**Output :**

Results Explain Describe Saved SQL History

Admain id	:	A1006
Admain name	:	Sumaiya Tabassum
Admin password	:	Sumaiya_Tabassum
Admain id	:	A1012
Admain name	:	Sanjida Alam
Admin password	:	Sanjida_Alam
Admain id	:	A1010
Admain name	:	Lamisa Mashiat
Admin password	:	Lamisa_Mashiat
Admain id	:	A1019
Admain name	:	Waziha Kabir
Admin password	:	Waziha_Kabir
Admain id	:	A1021
Admain name	:	Afrina Noor
Admin password	:	Afrina_Noor
Admain id	:	A1002
Admain name	:	Sanjida Afrin
Admin password	:	Sanjida_Afrin
Admain id	:	A1009
Admain name	:	Asif Chowdhury
Admin password	:	Asif_Chowdhury
Admain id	:	A1015
Admain name	:	Masud Haider
Admin password	:	Masud_Haider
Admain id	:	A1017
Admain name	:	Jihan Mirja
Admin password	:	Jihan_Mirja
Admain id	:	A1025
Admain name	:	Umme Habiba
Admin password	:	Umme_Habiba

Statement processed.

**Question-4 :** Show the details whose payment is less than 4000.

**Answer :**

Declare

```
amo number(15);
CURSOR payment_cur is
SELECT P_ID, C_ID, AMOUNT, PAYMENT_MODE
FROM PAYMENT;
payment_rec payment_cur%rowtype;
```

Begin

```
OPEN payment_cur;
LOOP
FETCH payment_cur into payment_rec;
EXIT WHEN payment_cur%notfound;
amo := payment_rec.amount;

if( amo < 4000) THEN
DBMS_OUTPUT.put_line('Payment Id      :' || payment_rec.p_id );
DBMS_OUTPUT.put_line('Customer Id    :' || payment_rec.c_id);
DBMS_OUTPUT.put_line('Amount          :' || payment_rec.amount);
DBMS_OUTPUT.put_line('Payment method' : ' || payment_rec.PAYMENT_MODE);
DBMS_OUTPUT.put_line(' ');
ELSE
dbms_output.put_line(' ');
END IF;
END LOOP;
```

END;

**Output :**

Results Explain Describe Saved SQL History

```
Payment Id      : P3012
Customer Id    : C117
Amount          : 1899
Payment method  : Credit Card

Payment Id      : P3002
Customer Id    : C101
Amount          : 1126
Payment method  : Bank

Payment Id      : P3015
Customer Id    : C118
Amount          : 3200
Payment method  : Cash

Statement processed.

0.01 seconds
```

**Question-5 :** Show the details which food is expired from today.

**Answer :**

Declare

```
CURSOR food_cur is
SELECT FOOD_ID, FOOD_NAME, EXPIRED_DATE
FROM food;
food_rec food_cur%rowtype;
```

Begin

```
OPEN food_cur;
```

```
LOOP
```

```
FETCH food_cur into food_rec;
```

```
EXIT WHEN food_cur%notfound;
```

```
IF( sysdate > food_rec.EXPIRED_DATE) then
```

```
DBMS_OUTPUT.put_line('Food ID : ' || food_rec.food_id);
```

```
DBMS_OUTPUT.put_line('Food Name : ' || food_rec.food_name);
```

```
DBMS_OUTPUT.put_line('Food Expire : ' || food_rec.EXPIRED_DATE );
```

```
DBMS_OUTPUT.put_line('');
```

```
END IF;
```

```
END LOOP;
```

```
END;
```

**Output :**

Results Explain Describe Saved SQL History

```
Food ID      : 5001
Food Name    : Fried Rice
Food Expire  : 15-JAN-19
```

```
Food ID      : 5013
Food Name    : Chicken Biriyani
Food Expire  : 09-JAN-19
```

```
Food ID      : 5009
Food Name    : Pizza
Food Expire  : 06-FEB-19
```

```
Food ID      : 5021
Food Name    : Cake
Food Expire  : 01-JAN-20
```

```
Food ID      : 5025
Food Name    : Beef Tikka
Food Expire  : 03-OCT-20
```

Food ID : 5037  
Food Name : Juice  
Food Expire : 11-MAR-22

Food ID : 5044  
Food Name : Sea Food  
Food Expire : 28-AUG-20

Food ID : 5004  
Food Name : Chicken Fry  
Food Expire : 07-APR-19

Food ID : 5016  
Food Name : Burger  
Food Expire : 17-AUG-19

Food ID : 5031  
Food Name : Sandwich  
Food Expire : 20-DEC-21

Statement processed.

0.00 seconds