

DBMS

MINI PROJECT

RATION CARD

MANAGEMENT SYSTEM

NAME : SANMAT SANJAYAKUMAR PAYAGOUDAR

SRN : PES1UG20CS385

SECTION : G

Description and Scope

Today, if a person goes to ration shop for buying anything he has to wait in queue till his chance comes, and ration shop person starts investigating about his details manually, which is very time consuming.

In present system there are many draw backs. For example

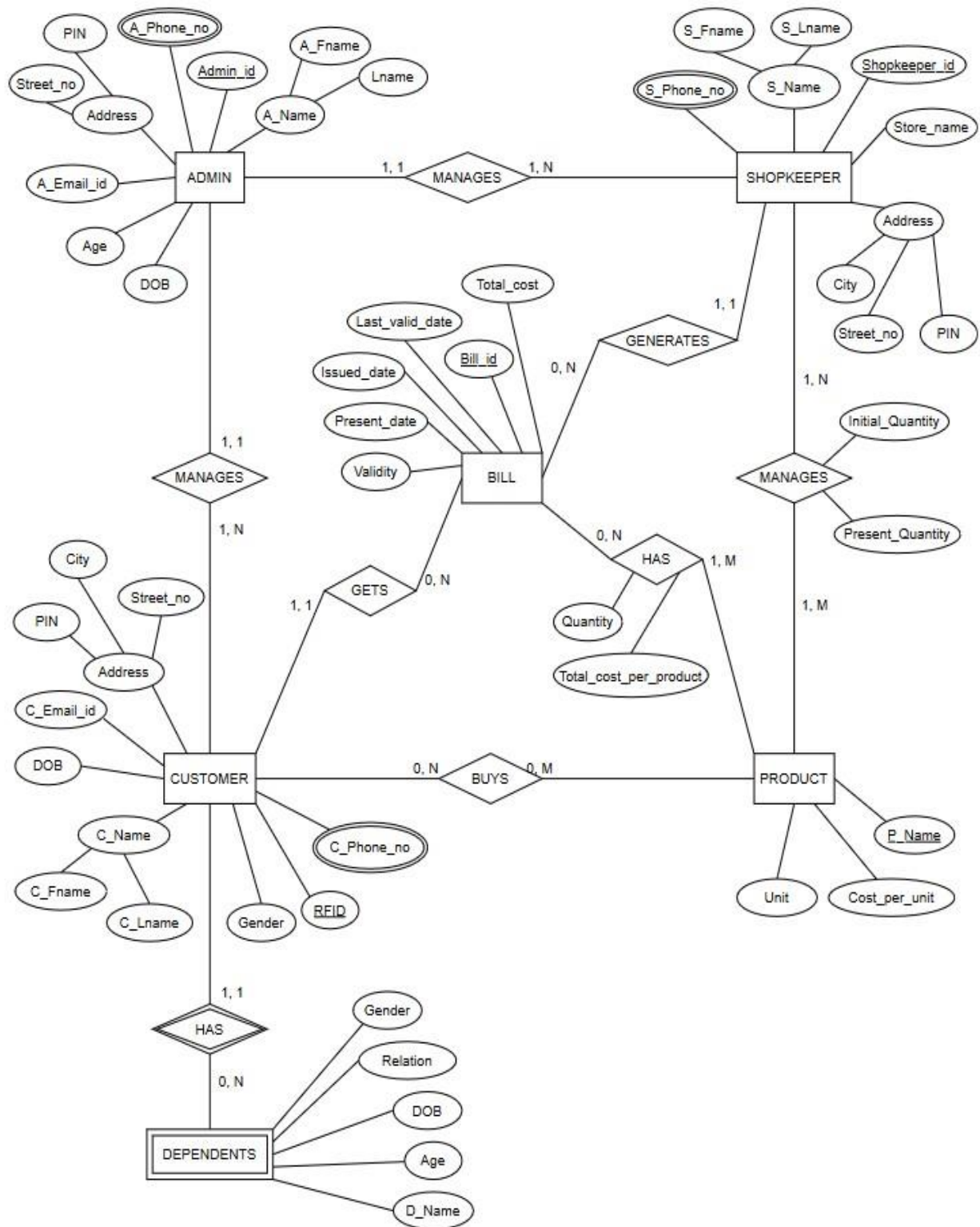
- Distributor undergoes illegal activities and fake transactions because of this reason end users are under loss and won't get their proper sock
- Government doesn't know how many users are there in particular city or district.
- All process done by manually using paper and this data can be lost in some accident like fire etc.
- Fast report generation is not possible.

ABOUT THIS PROJECT :

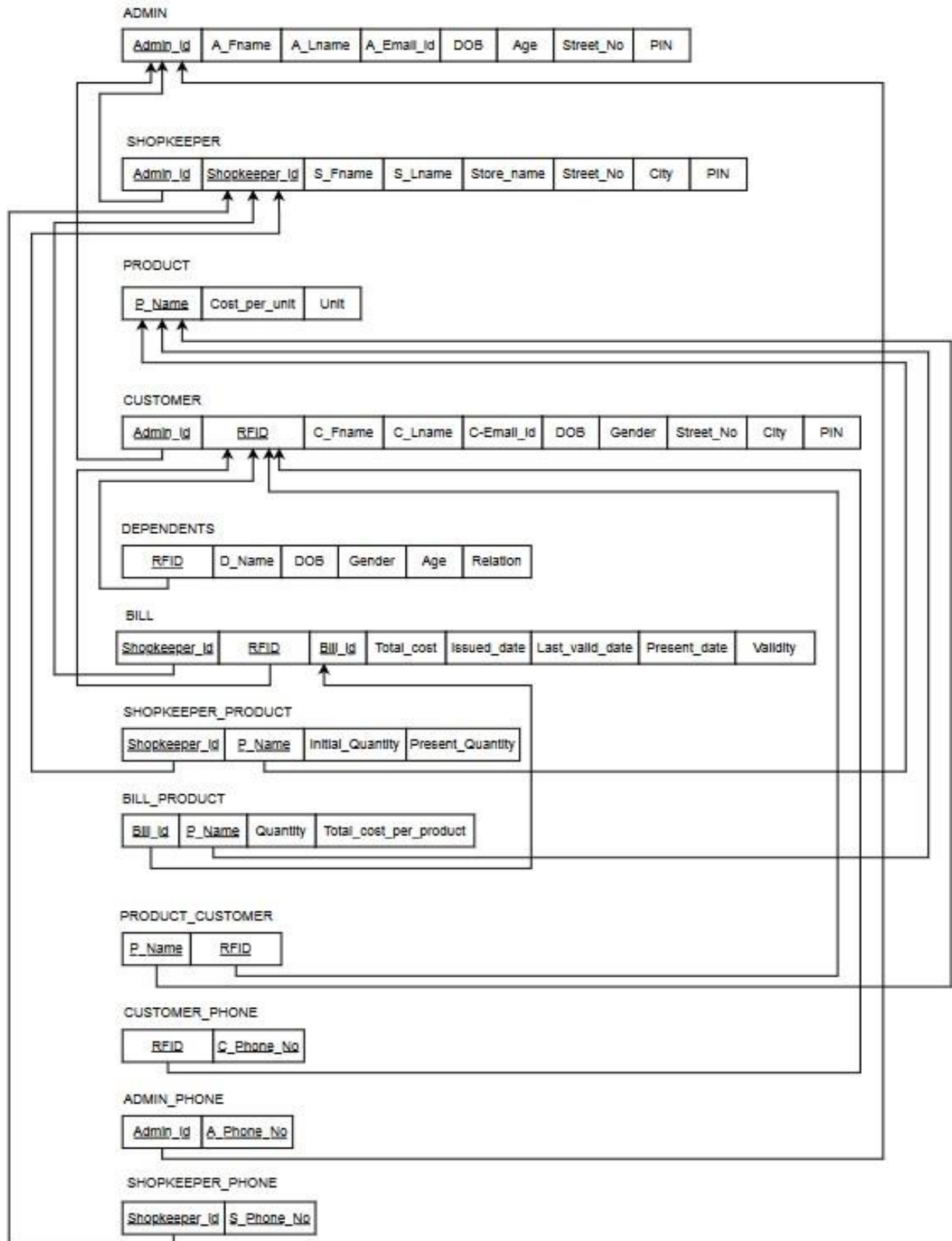
This "RATION CARS MANAGEMENT SYSTEM" is an efficient system to tackle the above problems. It has an interface for maintaining the database online. It provides services like rates of materials which is fixed by the state government and also provides the details of shopkeeper, customer and customer's past history about buying products and also the dependents of the customer.

It maintains all the details of distributors and his transaction, which will prevent the distributor from doing illegal activities. And government can easily access the whole data about customer and shopkeeper and any other data. Risk of losing data is less because of online data maintenance. And generation of report regarding any data is easy here.

ER – DIAGRAM



RELATIONAL SCHEMA



DDL Statements – Building the database

```
CREATE TABLE ADMIN(  
    Admin_id VARCHAR(10),  
    A_Fname VARCHAR(20),  
    A_Lname VARCHAR(20),  
    A_Email_id VARCHAR(50),  
    DOB DATE,  
    Age INT(3),  
    Street_no VARCHAR(40),  
    PIN INT(6),  
    PRIMARY KEY (Admin_id)  
);  
  
CREATE TABLE SHOPKEEPER(  
    Shopkeeper_id VARCHAR(10),  
    S_Fname VARCHAR(20),  
    S_Lname VARCHAR(20),  
    Store_name VARCHAR(20),  
    Street_no VARCHAR(40),  
    City VARCHAR(20),  
    PIN INT(6),  
    Admin_id VARCHAR(10),  
    PRIMARY KEY (Shopkeeper_id),  
    FOREIGN KEY (Admin_id) REFERENCES ADMIN (Admin_id)  
);  
  
CREATE TABLE PRODUCT(  
    P_Name VARCHAR(20),  
    Cost_per_unit float(5, 2),  
    Unit VARCHAR(20),  
    PRIMARY KEY (P_Name)  
);  
  
CREATE TABLE CUSTOMER(  
    RFID BIGINT(20),  
    C_Fname VARCHAR(20),  
    C_Lname VARCHAR(20),  
    C_Email_id VARCHAR(50),  
    DOB DATE,  
    Gender VARCHAR(10),  
    Street_no VARCHAR(40),  
    City VARCHAR(20),  
    PIN INT(6),  
    Admin_id VARCHAR(10),  
    PRIMARY KEY (RFID),  
    FOREIGN KEY (Admin_id) REFERENCES ADMIN (Admin_id)  
);
```

```
CREATE TABLE DEPENDENT(  
    RFID BIGINT(10),  
    D_Name VARCHAR(20),  
    DOB DATE,  
    Gender VARCHAR(10),  
    Age INT(3),  
    Relation VARCHAR(20),  
    FOREIGN KEY (RFID) REFERENCES CUSTOMER (RFID)  
    ON DELETE CASCADE  
);  
  
CREATE TABLE BILL(  
    Bill_id VARCHAR(10),  
    Total_cost float(5, 2),  
    Issued_date DATE,  
    Last_valid_date DATE,  
    Present_date DATE,  
    Validity VARCHAR(20),  
    Shopkeeper_id VARCHAR(10),  
    RFID BIGINT(10),  
    PRIMARY KEY (Bill_id),  
    FOREIGN KEY (Shopkeeper_id) REFERENCES SHOPKEEPER  
(Shopkeeper_id),  
    FOREIGN KEY (RFID) REFERENCES CUSTOMER (RFID)  
    ON DELETE CASCADE  
);  
  
CREATE TABLE SHOPKEEPER_PRODUCT(  
    Shopkeeper_id VARCHAR(10),  
    P_Name VARCHAR(20),  
    Initial_Quantity float(5, 2),  
    Present_Quantity float(5, 2),  
    PRIMARY KEY (Shopkeeper_id, P_Name),  
    FOREIGN KEY (Shopkeeper_id) REFERENCES SHOPKEEPER  
(Shopkeeper_id),  
    FOREIGN KEY (P_Name) REFERENCES PRODUCT (P_Name)  
    ON DELETE CASCADE  
);  
  
CREATE TABLE BILL_PRODUCT(  
    Bill_id VARCHAR(10),  
    P_Name VARCHAR(20),  
    Quantity float(5, 2),  
    Total_cost_per_product float(5, 2),  
    PRIMARY KEY (Bill_id, P_Name),  
    FOREIGN KEY (Bill_id) REFERENCES BILL (Bill_id),  
    FOREIGN KEY (P_Name) REFERENCES PRODUCT (P_Name)  
    ON DELETE CASCADE);
```

```

CREATE TABLE PRODUCT_CUSTOMER(
    P_Name VARCHAR(10),
    RFID BIGINT(20),
    PRIMARY KEY (P_Name, RFID),
    FOREIGN KEY (P_Name) REFERENCES PRODUCT (P_Name),
    FOREIGN KEY (RFID) REFERENCES CUSTOMER (RFID)
    ON DELETE CASCADE
);

CREATE TABLE CUSTOMER_PHONE(
    RFID BIGINT(20),
    Phone_no BIGINT(10),
    PRIMARY KEY (RFID, Phone_no),
    FOREIGN KEY (RFID) REFERENCES CUSTOMER (RFID)
    ON DELETE CASCADE
);

CREATE TABLE ADMIN_PHONE(
    Admin_id VARCHAR(10),
    Phone_no BIGINT(10),
    PRIMARY KEY (Admin_id, Phone_no),
    FOREIGN KEY (Admin_id) REFERENCES ADMIN (Admin_id)
    ON DELETE CASCADE
);

CREATE TABLE SHOPKEEPER_PHONE(
    Shopkeeper_id VARCHAR(10),
    Phone_no BIGINT(10),
    PRIMARY KEY (Shopkeeper_id, Phone_no),
    FOREIGN KEY (Shopkeeper_id) REFERENCES SHOPKEEPER
(Shopkeeper_id)
    ON DELETE CASCADE
);

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0109 seconds.)

```
CREATE TABLE ADMIN( Admin_id VARCHAR(10), A_Fname VARCHAR(20), A_Lname VARCHAR(20), A_Email_id VARCHAR(20), DOB DATE, Age INT(3), Street_No VARCHAR(20), PIN INT(6), PRIMARY KEY (Admin_id) );
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0087 seconds.)

```
CREATE TABLE SHOPKEEPER( Shopkeeper_id VARCHAR(10), S_Fname VARCHAR(20), S_Lname VARCHAR(20), Store_name VARCHAR(20), Street_No VARCHAR(20), City VARCHAR(20), PIN INT(6), Admin_id VARCHAR(10), PRIMARY KEY (Shopkeeper_id), FOREIGN KEY (Admin_id) REFERENCES ADMIN (Admin_id) );
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0105 seconds.)

```
CREATE TABLE PRODUCT( P_Name VARCHAR(20), Cost_per_unit INT(5), Unit VARCHAR(20), PRIMARY KEY (P_Name) );
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0096 seconds.)

```
CREATE TABLE CUSTOMER( RFID BIGINT(20), C_Fname VARCHAR(20), C_Lname VARCHAR(20), C_Email_id VARCHAR(20), DOB DATE, Gender VARCHAR(10), Street_No VARCHAR(20), City VARCHAR(20), PIN INT(6), Admin_id VARCHAR(10), PRIMARY KEY (RFID), FOREIGN KEY (Admin_id) REFERENCES ADMIN (Admin_id) );
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0077 seconds.)

```
CREATE TABLE DEPENDENTS( RFID BIGINT(10), D_Name VARCHAR(20), DOB DATE, Gender VARCHAR(10), Age INT(3), Relation VARCHAR(20), FOREIGN KEY (RFID) REFERENCES CUSTOMER (RFID) );
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0097 seconds.)

```
CREATE TABLE BILL( Bill_id VARCHAR(10), Total_cost INT(5), Issued_date DATE, Last_valid_date DATE, Present_date DATE, Validity VARCHAR(20), Shopkeeper_id VARCHAR(10), RFID BIGINT(10), PRIMARY KEY (Bill_id), FOREIGN KEY (Shopkeeper_id) REFERENCES SHOPKEEPER (Shopkeeper_id), FOREIGN KEY (RFID) REFERENCES CUSTOMER (RFID) );
```

[Edit inline] [Edit] [Create PHP code]

Populating the Database

Using INSERT command :

Query :

```
INSERT INTO SHOPKEEPER(Shopkeeper_id, S_Fname, S_Lname, Store_name,
Street_No, City, PIN, Admin_id) VALUES
('SHK_001', 'Manisha', 'Solanki', 'Lakshmi FPS', 'M G Road',
'Bengaluru', 575030, 'ADM_001'),
('SHK_002', 'Bharti', 'Devgan', 'Mahaveer FPS', 'NAL Wind Tunnel
Road', 'Mangaluru', 560059, 'ADM_003'),
('SHK_003', 'Roopa', 'Ram', 'Goudra FPS', 'Bunder
Road', 'Belagavi', 575023, 'ADM_002'),
('SHK_004', 'Pallavi', 'Ram', 'Mirje FPS', 'Lavelle Road',
'Kalaburgi', 560044, 'ADM_003'),
('SHK_005', 'Ravi', 'Naik', 'Padmavati FPS', 'Thiru V ka Salai',
'Udupi', 600045, 'ADM_001'),
('SHK_006', 'Virat', 'Murthy', 'Munnoli FPS', 'Bejai New
Road', 'KGF', 575011, 'ADM_002');
```

OUTPUT :

Run SQL query/queries on database ration_card_system1: ⓘ

```
1 INSERT INTO SHOPKEEPER(Shopkeeper_id, S_Fname, S_Lname, Store_name, Street_No, City, PIN, Admin_id) VALUES
2 ('SHK_001', 'Manisha', 'Solanki', 'Lakshmi FPS', 'M G Road', 'Bengaluru', 575030, 'ADM_001'),
3 ('SHK_002', 'Bharti', 'Devgan', 'Mahaveer FPS', 'NAL Wind Tunnel Road', 'Mangaluru', 560059, 'ADM_003'),
4 ('SHK_003', 'Roopa', 'Ram', 'Goudra FPS', 'Bunder Road', 'Belagavi', 575023, 'ADM_002'),
5 ('SHK_004', 'Pallavi', 'Ram', 'Mirje FPS', 'Lavelle Road', 'Kalaburgi', 560044, 'ADM_003'),
6 ('SHK_005', 'Ravi', 'Naik', 'Padmavati FPS', 'Thiru V ka Salai', 'Udupi', 600045, 'ADM_001'),
7 ('SHK_006', 'Virat', 'Murthy', 'Munnoli FPS', 'Bejai New Road', 'KGF', 575011, 'ADM_002');
8
```

Clear Format Get auto-saved query

☐ Bind parameters ⓘ

Delimiter ; ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks Go

Hide query box

✓ 6 rows inserted. (Query took 0.0026 seconds.)

```
INSERT INTO SHOPKEEPER(Shopkeeper_id, S_Fname, S_Lname, Store_name, Street_No, City, PIN, Admin_id) VALUES ('SHK_001', 'Manisha', 'Solanki', 'Lakshmi FPS', 'M G Road',
'Bengaluru', 575030, 'ADM_001'), ('SHK_002', 'Bharti', 'Devgan', 'Mahaveer FPS', 'NAL Wind Tunnel Road', 'Mangaluru', 560059, 'ADM_003'), ('SHK_003', 'Roopa', 'Ram', 'Goudra
FPS', 'Bunder Road', 'Belagavi', 575023, 'ADM_002'), ('SHK_004', 'Pallavi', 'Ram', 'Mirje FPS', 'Lavelle Road', 'Kalaburgi', 560044, 'ADM_003'), ('SHK_005', 'Ravi', 'Naik',
'Padmavati FPS', 'Thiru V ka Salai', 'Udupi', 600045, 'ADM_001'), ('SHK_006', 'Virat', 'Murthy', 'Munnoli FPS', 'Bejai New Road', 'KGF', 575011, 'ADM_002');
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Using IMPORT :

OUTPUT :

✔ Import has been successfully finished, 24 queries executed. (CUSTOMER.csv)

✔ 1 row inserted. (Query took 0.0019 seconds.)

```
INSERT INTO `customer` VALUES ('3453484810', 'Ajit', 'Ullal', 'ullalajit@gmail.com', '1971-01-01', 'M', '10 Janpath', 'Bengaluru', '560001', 'ADM_001');
```

[Edit inline] [Edit] [Create PHP code]

✔ 1 row inserted. (Query took 0.0014 seconds.)

```
INSERT INTO `customer` VALUES ('6815974590', 'Muhammed ', 'Ali', 'alimihammed@gmail.com', '2000-10-09', 'M', '10 Downing', 'Mangaluru', '560003', 'ADM_002');
```

[Edit inline] [Edit] [Create PHP code]

⚠ Warning: #1265 Data truncated for column 'C_Email_id' at row 1

✔ 1 row inserted. (Query took 0.0010 seconds.)

```
INSERT INTO `customer` VALUES ('5483807477', 'Sai Deepak', 'Reddy', 'reddysai@gmail.com', '1996-12-12', 'F', '14 Kailsh Marg', 'Belagavi', '600001', 'ADM_002');
```

[Edit inline] [Edit] [Create PHP code]

✔ 1 row inserted. (Query took 0.0011 seconds.)

```
INSERT INTO `customer` VALUES ('9187065308', 'Rudra', 'Agarwal', 'agarwalrudra@gmail.com', '1996-12-13', 'F', '5 Lohia Garden', 'Bengaluru', '600042', 'ADM_001');
```

[Edit inline] [Edit] [Create PHP code]

⚠ Warning: #1265 Data truncated for column 'C_Email_id' at row 1

Console

✔ 1 row inserted. (Query took 0.0017 seconds.)

```
INSERT INTO `customer` VALUES ('1700494946', 'Rahul', 'Khanna', 'khannarahul@gmail.com', '1997-12-14', 'M', '1 Pink Street', 'Belagavi', '600067', 'ADM_001');
```

[Edit inline] [Edit] [Create PHP code]

⚠ Warning: #1265 Data truncated for column 'C_Email_id' at row 1

✔ 1 row inserted. (Query took 0.0015 seconds.)

```
INSERT INTO `customer` VALUES ('4478160351', 'Nirmala', 'Seturaman', 'seturamannirmala@gmail.com', '1998-12-15', 'F', '3 Blue Avenue', 'Mangaluru', '560105', 'ADM_002');
```

[Edit inline] [Edit] [Create PHP code]

⚠ Warning: #1265 Data truncated for column 'C_Email_id' at row 1

✔ 1 row inserted. (Query took 0.0009 seconds.)

```
INSERT INTO `customer` VALUES ('6977990400', 'Smriti', 'Irani', 'iranismriti@gmail.com', '1999-12-16', 'F', '4 Banyan Avenie', 'Mangaluru', '575014', 'ADM_003');
```

[Edit inline] [Edit] [Create PHP code]

⚠ Warning: #1265 Data truncated for column 'C_Email_id' at row 1

JOIN Queries

1. Retrieve Customer name and son name iff customer has a son.

Query :

```
SELECT C_Fname, C_Lname, D_Name AS Son_Name FROM CUSTOMER JOIN  
DEPENDENTS ON CUSTOMER.RFID = DEPENDENTS.RFID WHERE  
DEPENDENTS.Relation = 'Son';
```

Output :

Run SQL query/queries on database **ration_card_system**:

1

SELECT C_Fname, C_Lname, D_Name AS Son_Name FROM CUSTOMER JOIN DEPENDENTS ON CUSTOMER.RFID = DEPENDENTS.RFID WHERE DEPENDENTS.Relation = 'Son';

Clear

Format

Get auto-saved query

☐ Bind parameters

Delimiter: ;

☐ Show this query here again

☐ Retain query box

☐ Rollback when finished

☒ Enable foreign key checks

Go

Hide query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✔ Showing rows 0 - 10 (11 total, Query took 0.0011 seconds.)

SELECT C_Fname, C_Lname, D_Name AS Son_Name FROM CUSTOMER JOIN DEPENDENTS ON CUSTOMER.RFID = DEPENDENTS.RFID WHERE DEPENDENTS.Relation = 'Son';

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

C_Fname	C_Lname	Son_Name
Rahul	Khanna	Moksh
Console	Udapa	Ajay
Ajit	Ullal	Arjun
Arun	Kumar	Zayn
Rashkit	Shetty	Ari
Rahul	Gandhi	Siddh
Muhammed	Ali	Aum
Ajit	Sethi	Laksh
Rishi	Sunak	Dev
Arjun	Allu	Moksh
Satish	Kotian	Akash

2. Retrieve Customer name and dependent name if customer has a dependent.

Query :

```
SELECT C_Fname, C_Lname, D_Name FROM CUSTOMER LEFT JOIN  
DEPENDENTS ON CUSTOMER.RFID = DEPENDENTS.RFID;
```

Output :

Run SQL query/queries on table `ration_card_system.customer`:

1 `SELECT C_Fname, C_Lname, D_Name FROM CUSTOMER LEFT JOIN DEPENDENTS ON CUSTOMER.RFID = DEPENDENTS.RFID;`

[SELECT *](#) [SELECT](#) [INSERT](#) [UPDATE](#) [DELETE](#) [Clear](#) [Format](#) [Get auto-saved query](#)

☐ Bind parameters

Delimiter: `;` ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

[Hide query box](#)

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 24 (44 total, Query took 0.0005 seconds.)

`SELECT C_Fname, C_Lname, D_Name FROM CUSTOMER LEFT JOIN DEPENDENTS ON CUSTOMER.RFID = DEPENDENTS.RFID;`

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

1 > >> | ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: Not set

[Extra options](#)

C_Fname	C_Lname	D_Name
Anshu	Bhardwaj	NULL
Console	Khanna	NULL
Samanta	Prabhu	Divya
Samanta	Prabhu	Sajan
Srihari	Udupa	Amara
Srihari	Udupa	Ajay
Ajit	Ullal	Aashvi
Ajit	Ullal	Arjun
Arun	Kumar	NULL
Nirmala	Seturaman	Shyla
Nirmala	Seturaman	Rishi
Shradha	Nayar	Aniya
Shradha	Nayar	Akhil
Sai Deepak	Reddy	NULL
Margaret	Alva	Aanya
Margaret	Alva	Sai
Rashkit	Shetty	Kimaya
Rashkit	Shetty	Ari
Rahul	Gandhi	Lavanya
Rahul	Gandhi	Siddh
Muhammed	Ali	Lakshmi
Muhammed	Ali	Aum
Smriti	Irani	Sita
Smriti	Irani	Veer
Subbu	Saravana	Zoya

3. Retrieve Dependent and Customer name if customer has a dependent.

Query :

```
SELECT D_Name, C_Fname, C_Lname FROM DEPENDENTS RIGHT JOIN CUSTOMER ON CUSTOMER.RFID = DEPENDENTS.RFID;
```

Output :

Run SQL query/queries on table **ration_card_system.shopkeeper_phone**:

1 `SELECT D_Name, C_Fname, C_Lname FROM DEPENDENTS RIGHT JOIN CUSTOMER ON CUSTOMER.RFID = DEPENDENTS.RFID;`

[SELECT *](#) [SELECT](#) [INSERT](#) [UPDATE](#) [DELETE](#) [Clear](#) [Format](#) [Get auto-saved query](#)

☐ Bind parameters

Delimiter: `;` ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key

[Hide query box](#)

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 24 (44 total, Query took 0.0002 seconds.)

`SELECT D_Name, C_Fname, C_Lname FROM DEPENDENTS RIGHT JOIN CUSTOMER ON CUSTOMER.RFID = DEPENDENTS.RFID;`

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

1 > >> | ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

[Extra options](#)

D_Name	C_Fname	C_Lname
NULL	Anshu	Bhardwaj
Console	Rahul	Khanna
Divya	Samanta	Prabhu
Sajan	Samanta	Prabhu
Amara	Srihari	Udupa
Ajay	Srihari	Udupa
Aashvi	Ajit	Ullal
Arjun	Ajit	Ullal
NULL	Arun	Kumar
Shyla	Nirmala	Seturaman
Rishi	Nirmala	Seturaman
Aniya	Shradha	Nayar
Akhil	Shradha	Nayar
NULL	Sai Deepak	Reddy
Aanya	Margaret	Alva
Sai	Margaret	Alva
Kimaya	Rashkit	Shetty
Ari	Rashkit	Shetty
Lavanya	Rahul	Gandhi
Siddh	Rahul	Gandhi
Lakshmi	Muhammed	Ali
Aum	Muhammed	Ali
Sita	Smriti	Irani
Veer	Smriti	Irani
Zoya	Subbu	Saravana

4. Retrieve customer name and product bought date for those customers who bought the products between 28th February 2021 and 3rd October 2021.

Query :

```
SELECT C_Fname, C_Lname, Present_date AS Bought_date FROM CUSTOMER JOIN BILL ON CUSTOMER.RFID = BILL.RFID WHERE BILL.Present_date BETWEEN '2021-02-28' AND '2021-10-03';
```

Output :

Run SQL query/queries on table `ration_card_system.bill`:

```
1 SELECT C_Fname, C_Lname, Present_date AS Bought_date FROM CUSTOMER JOIN BILL ON CUSTOMER.RFID = BILL.RFID WHERE BILL.Present_date BETWEEN '2021-02-28' AND '2021-10-03';
```

☐ Bind parameters

Delimiter: `;` ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks **Go**

Hide query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 24 (31 total, Query took 0.0010 seconds.)

```
SELECT C_Fname, C_Lname, Present_date AS Bought_date FROM CUSTOMER JOIN BILL ON CUSTOMER.RFID = BILL.RFID WHERE BILL.Present_date BETWEEN '2021-02-28' AND '2021-10-03';
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

1 > >> ☐ Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

C_Fname	C_Lname	Bought_date
	Bhardwaj	2021-09-28
Rahul	Khanna	2021-07-30
Samanta	Prabhu	2021-05-02
Samanta	Prabhu	2021-08-23
Srihari	Udupa	2021-03-03
Srihari	Udupa	2021-02-28
Ajit	Ullal	2021-06-07
Arun	Kumar	2021-05-17
Nirmala	Seturaman	2021-05-11
Nirmala	Seturaman	2021-07-23
Shradha	Nayar	2021-06-17
Margaret	Alva	2021-08-28
Rashkit	Shetty	2021-09-22
Rahul	Gandhi	2021-03-17
Rahul	Gandhi	2021-05-14
Muhammed	Ali	2021-07-14
Muhammed	Ali	2021-05-28
Smriti	Irani	2021-07-09
Smriti	Irani	2021-08-16
Subbu	Saravana	2021-06-24
Ajit	Sethi	2021-05-09
Sheela	Dixit	2021-03-19
Rishi	Sunak	2021-09-09
Rishi	Sunak	2021-03-13
Arjun	Allu	2021-06-09

Aggregate Functions

1. Retrieve the total amount spent by each customer.

Query:

```
SELECT SUM(Total_cost) FROM bill GROUP BY RFID;
```

Output:

Run SQL query/queries on database **ration_card_system**:

```
1 SELECT RFID,SUM(Total_cost) FROM bill GROUP BY RFID;
```

[Clear](#) [Format](#) [Get auto-saved query](#)

☐ Bind parameters

Delimiter: Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks [Go](#)

[Hide query box](#)

Showing rows 0 - 23 (24 total, Query took 0.0018 seconds.)

```
SELECT RFID,SUM(Total_cost) FROM bill GROUP BY RFID;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: Filter rows:

[Extra options](#)

		RFID	SUM(Total_cost)
<input type="checkbox"/>	Edit	1637382417	315
<input type="checkbox"/>	Edit	1700494946	215
<input checked="" type="checkbox"/>	Console	2192227997	160
<input type="checkbox"/>	Edit	2624574553	40
<input type="checkbox"/>	Edit	3453484810	310
<input type="checkbox"/>	Edit	3740546453	170
<input type="checkbox"/>	Edit	4478160351	135
<input type="checkbox"/>	Edit	4693726418	100
<input type="checkbox"/>	Edit	5483807477	225
<input type="checkbox"/>	Edit	6107074114	45
<input type="checkbox"/>	Edit	6283738531	340
<input type="checkbox"/>	Edit	6597759624	65
<input type="checkbox"/>	Edit	6815974590	240
<input type="checkbox"/>	Edit	6977990400	105
<input type="checkbox"/>	Edit	7243413566	155
<input type="checkbox"/>	Edit	7575833003	65
<input type="checkbox"/>	Edit	8565854116	70
<input type="checkbox"/>	Edit	8646892910	65
<input type="checkbox"/>	Edit	8727666988	140
<input type="checkbox"/>	Edit	8808967440	275
<input type="checkbox"/>	Edit	8945871085	55
<input type="checkbox"/>	Edit	8979106724	80
<input type="checkbox"/>	Edit	9187065308	185
<input type="checkbox"/>	Edit	9579531991	105

2. Retrieve the bill id and total cost for bill which have the maximum value for total cost.

Query :

```
SELECT Bill_id, Total_cost from bill where bill.Total_cost IN (SELECT MAX(Total_cost) FROM BILL);
```

Output:

Run SQL query/queries on table `ration_card_system.bill`:

```
1 SELECT Bill_id, Total_cost from bill where bill.Total_cost IN (SELECT MAX(Total_cost) FROM BILL);
```

[SELECT *](#) [SELECT](#) [INSERT](#) [UPDATE](#) [DELETE](#) [Clear](#) [Format](#) [Get auto-saved query](#)

☐ Bind parameters

Delimiter: ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks [Go](#)

[Hide query box](#)

✓ Showing rows 0 - 5 (6 total, Query took 0.0014 seconds.)

```
SELECT Bill_id, Total_cost from bill where bill.Total_cost IN (SELECT MAX(Total_cost) FROM BILL);
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	Bill_id	Total_cost
<input type="checkbox"/> Edit Copy Delete	B_002	170
<input type="checkbox"/> Edit Copy Delete	B_005	170
<input checked="" type="checkbox"/> Console Edit Copy Delete	B_009	170
<input type="checkbox"/> Edit Copy Delete	B_025	170
<input type="checkbox"/> Edit Copy Delete	B_026	170
<input type="checkbox"/> Edit Copy Delete	B_050	170

3. Retrieve the bill id and total cost for bill which have the minimum value for total cost.

Query :

```
SELECT Bill_id, Total_cost from bill where bill.Total_cost IN (SELECT MIN(Total_cost) FROM BILL);
```

Output:

Run SQL query/queries on table `ration_card_system.bill`:

```
1 SELECT Bill_id, Total_cost from bill where bill.Total_cost IN (SELECT MIN(Total_cost) FROM BILL);
```

SELECT* SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query

☐ Bind parameters

Delimiter: ; ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks Go

Hide query box

✓ Showing rows 0 - 2 (3 total, Query took 0.0014 seconds.)

```
SELECT Bill_id, Total_cost from bill where bill.Total_cost IN (SELECT MIN(Total_cost) FROM BILL);
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

		Bill_id	Total_cost
<input type="checkbox"/>	Edit Copy Delete	B_013	10
<input type="checkbox"/>	Edit Copy Delete	B_036	10
<input checked="" type="checkbox"/>	Console Edit Copy Delete	B_043	10

4. Retrieve the number of bills which have the total cost equal to maximum value for total cost.

Query :

```
SELECT COUNT(Total_cost) from bill WHERE Total_cost IN (SELECT MAX(Total_cost) FROM BILL);
```

Output :

Run SQL query/queries on table `ration_card_system.bill`: ⓘ

```
1 SELECT COUNT(Total_cost) from bill WHERE Total_cost IN (SELECT MAX(Total_cost) FROM BILL);
```

☐ Bind parameters ⓘ

Delimiter ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

Your SQL query has been executed successfully.

```
SELECT COUNT(Total_cost) from bill WHERE Total_cost IN (SELECT MAX(Total_cost) FROM BILL);
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

COUNT(Total_cost)
6

5. Retrieve the RFID of the customer and average cost spent by them.

Query :

```
SELECT RFID, AVG(Total_cost) from bill GROUP BY RFID;
```

Output:

Run SQL query/queries on table `ration_card_system.bill`:

```
1 SELECT RFID, AVG(Total_cost) from bill GROUP BY RFID;
```

☐ Bind parameters

Delimiter: ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

✓ Showing rows 0 - 23 (24 total, Query took 0.0051 seconds.)

```
SELECT RFID, AVG(Total_cost) from bill GROUP BY RFID;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: Filter rows:

	RFID	AVG(Total_cost)
<input type="checkbox"/> Edit Copy Delete	1637382417	105.0000
<input type="checkbox"/> Edit Copy Delete	1700494946	107.5000
<input checked="" type="checkbox"/> Console Edit Copy Delete	2192227997	80.0000
<input type="checkbox"/> Edit Copy Delete	2624574553	20.0000
<input type="checkbox"/> Edit Copy Delete	3453484810	155.0000
<input type="checkbox"/> Edit Copy Delete	3740546453	85.0000
<input type="checkbox"/> Edit Copy Delete	4478160351	67.5000
<input type="checkbox"/> Edit Copy Delete	4693726418	50.0000
<input type="checkbox"/> Edit Copy Delete	5483807477	112.5000
<input type="checkbox"/> Edit Copy Delete	6107074114	22.5000
<input type="checkbox"/> Edit Copy Delete	6283738531	170.0000
<input type="checkbox"/> Edit Copy Delete	6597759624	32.5000
<input type="checkbox"/> Edit Copy Delete	6815974590	120.0000
<input type="checkbox"/> Edit Copy Delete	6977990400	52.5000
<input type="checkbox"/> Edit Copy Delete	7243413566	77.5000
<input type="checkbox"/> Edit Copy Delete	7575833003	32.5000
<input type="checkbox"/> Edit Copy Delete	8565854116	35.0000
<input type="checkbox"/> Edit Copy Delete	8646892910	32.5000
<input type="checkbox"/> Edit Copy Delete	8727666988	70.0000
<input type="checkbox"/> Edit Copy Delete	8808967440	137.5000
<input type="checkbox"/> Edit Copy Delete	8945871085	27.5000
<input type="checkbox"/> Edit Copy Delete	8979106724	40.0000
<input type="checkbox"/> Edit Copy Delete	9187065308	92.5000
<input type="checkbox"/> Edit Copy Delete	9579531991	52.5000

SET Operations

1. Retrieve the phone numbers that are either of admin or shopkeeper table.

Query :

```
SELECT ADMIN_PHONE.Phone_no FROM ADMIN_PHONE
UNION
SELECT SHOPKEEPER_PHONE.Phone_no FROM SHOPKEEPER_PHONE;
```

Output :

Run SQL query/queries on database **ration_card_system**:

```
1 SELECT ADMIN_PHONE.Phone_no FROM ADMIN_PHONE
2 UNION
3 SELECT SHOPKEEPER_PHONE.Phone_no FROM SHOPKEEPER_PHONE;
```

ClearFormatGet auto-saved query

☐ Bind parameters

Delimiter ;

☐ Show this query here again☐ Retain query box☐ Rollback when finished☒ Enable foreign key checks

Go

Hide query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

```
SELECT ADMIN_PHONE.Phone_no FROM ADMIN_PHONE UNION SELECT SHOPKEEPER_PHONE.Phone_no FROM SHOPKEEPER_PHONE;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

Phone_no

Console

7333012345

9900456456

6347012345

7346123456

Query :

Output :

[illegible]

3. Retrieve the first name which are present in both customer and shopkeeper table.

Query :

```
SELECT C.C_Fname FROM CUSTOMER C
INTERSECT
SELECT S.S_Fname FROM SHOPKEEPER S;
```

Output :

The screenshot shows a web-based SQL query execution tool. At the top, a header bar says "Run SQL query/queries on table `ration_card_system_385.customer`:". Below this, the query is entered in a text area: `1 SELECT C.C_Fname FROM CUSTOMER C`, `2 INTERSECT`, and `3 SELECT S.S_Fname FROM SHOPKEEPER S;`. The query is highlighted with a red box. Below the query area, there are buttons for "SELECT *", "SELECT", "INSERT", "UPDATE", "DELETE", "Clear", "Format", and "Get auto-saved query". There is also a checkbox for "Bind parameters". Below these buttons, there is a "Delimiter" dropdown set to ";" and several checkboxes: "Show this query here again", "Retain query box", "Rollback when finished", and "Execute". A "Hide query box" button is also present. Below the query area, a message states: "Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available". A green banner indicates: "Showing rows 0 - 0 (1 total, Query took 0.0010 seconds.)". Below this, the query is repeated: `SELECT C.C_Fname FROM CUSTOMER C INTERSECT SELECT S.S_Fname FROM SHOPKEEPER S;`. There are links for "Profiling", "Edit inline", "Edit", "Explain SQL", "Create PHP code", and "Refresh". Below the query, there is a "Show all" checkbox, a "Number of rows" dropdown set to "25", and a "Filter rows" input field with the placeholder "Search this table". At the bottom, there is an "Extra options" button and a "Console" button. The name "Virat" is visible in the bottom left corner.

4. Retrieve the city names which are present in both customer and shopkeeper table.

Query :

```
SELECT S.City FROM SHOPKEEPER S
INTERSECT
SELECT C.City FROM CUSTOMER C;
```

Output :

Run SQL query/queries on table `ration_card_system.admin`:

1 `SELECT S.City FROM SHOPKEEPER S`

2 `INTERSECT`

3 `SELECT C.City FROM CUSTOMER C;`

SELECT *SELECTINSERTUPDATEDELETEClearFormatGet auto-saved query

☐ Bind parameters

Delimiter ;☐ Show this query here again☐ Retain query box☐ Rollback when finished☒ Enable foreign key checksGo

Hide query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✔ Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

`SELECT S.City FROM SHOPKEEPER S INTERSECT SELECT C.City FROM CUSTOMER C;`

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

City

Console

Mangaluru

Belagavi

Functions and Procedures

1. Create a function to check the eligibility of a particular are for ration card.

Query :

```
DELIMITER $$
create function eligible(Age int)
  RETURNS varchar(20)
  DETERMINISTIC
  BEGIN
    IF Age > 10 THEN
      RETURN ("yes");
    ELSE
      RETURN ("No");
    END IF;
  end$$
DELIMITER ;
```

Output :

```
MariaDB [ration_card_system]> select eligible(11);
+-----+
| eligible(11) |
+-----+
| yes          |
+-----+
1 row in set (0.001 sec)

MariaDB [ration_card_system]> select eligible(9);
+-----+
| eligible(9) |
+-----+
| No          |
+-----+
1 row in set (0.000 sec)
```

2. Create a procedure to count valid bills.

Query :

```
DELIMITER $$
CREATE PROCEDURE count_valid_bills(OUT p1 INT)
BEGIN
  select COUNT(Bill_id) INTO p1 from BILL WHERE Validity = 'Valid';
END $$
DELIMITER ;
```

Output :

Run SQL query/queries on database **ration_card_system**:

```
1 CALL count_valid_bills(@p);
2 SELECT @p;
```

[Clear](#) [Format](#) [Get auto-saved query](#)

☐ Bind parameters

Delimiter: ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks [Go](#)

[Hide query box](#)

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0028 seconds.)

```
CALL count_valid_bills(@p);
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT @p;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

Console | w all | Number of rows: 25 | Filter rows:

Extra options

@p

40

☐ Show all | Number of rows: 25 | Filter rows:

3. Create a procedure to retrieve given number of customers (Number of customer to be retrieved should be given as input).

Query :

```
DELIMITER $$
CREATE PROCEDURE get_customer_limit(IN var1 INT)
BEGIN
    select * from CUSTOMER limit var1;
    select count(RFID) as total_customer from CUSTOMER;
END $$
DELIMITER ;
```

Output :

```
mysql> CALL get_customer_limit(10);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| RFID      | C_Fname | C_Lname | C_Email_id          | DOB      | Gender | Street_no      | City      | PIN      | Admin_id |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1637382417 | Anshu   | Bhardwaj | bhardwajanshu@gmail.com | 1978-04-14 | F      | KSR Road       | Hubballi  | 575001   | ADM_003 |
| 1700494946 | Rahul   | Khanna   | khannarahul@gmail.com  | 1997-12-14 | M      | 1 Pink Street  | Belagavi  | 600067   | ADM_001 |
| 2192227997 | Samanta | Prabhu   | prabhusamanta@gmail.com | 1968-04-14 | F      | 8 Clone Colony | Belagavi  | 600042   | ADM_001 |
| 2624574553 | Srihari | Udupa    | udupasrihari@gmail.com | 1980-04-14 | M      | Suranjan Das Road | Hubballi  | 560015   | ADM_001 |
| 3453484810 | Ajit    | Ullal    | ullalajit@gmail.com    | 1971-01-01 | M      | 10 Janpath     | Bengaluru | 560001   | ADM_001 |
| 3740546453 | Arun    | Kumar    | kumararun@gmail.com    | 1977-04-14 | M      | Mint Street    | Bengaluru | 600046   | ADM_003 |
| 4478160351 | Nirmala | Seturaman | seturamannirmala@gmail.com | 1998-12-15 | F      | 3 Blue Avenue  | Mangaluru | 560105   | ADM_002 |
| 4693726418 | Shradha | Nayar    | nayarshradha@gmail.com  | 1982-04-08 | F      | M G Road Bangalore | Belagavi  | 560006   | ADM_003 |
| 5483807477 | Sai Deepak | Reddy   | reddysai@gmail.com      | 1996-12-12 | F      | 14 Kailsh Marg | Belagavi  | 600001   | ADM_002 |
| 6107074114 | Margaret | Alva     | alvamargaret@gmail.com  | 1976-04-14 | F      | New BEL Road   | Hubballi  | 560008   | ADM_003 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

+-----+
| total_customer |
+-----+
| 24 |
+-----+
1 row in set (0.03 sec)

Query OK, 0 rows affected (0.04 sec)
```

4. Create a procedure to retrieve all the customers managed by given admin (Admin id should be given as input).

Query :

```
DELIMITER $$
CREATE PROCEDURE get_customer_admin(in Adm varchar(10))
BEGIN
    select C_Fname, C_Lname, RFID
    from CUSTOMER WHERE CUSTOMER.Admin_id = Adm;
end $$
DELIMITER ;
```

Output :

```
mysql> CALL get_customer_admin('ADM_001');
+-----+-----+-----+
| C_Fname | C_Lname | RFID      |
+-----+-----+-----+
| Rahul   | Khanna  | 1700494946 |
| Samanta | Prabhu  | 2192227997 |
| Srihari | Udupa   | 2624574553 |
| Ajit    | Ullal   | 3453484810 |
| Sheela  | Dixit   | 8565854116 |
| Suma    | Sampat  | 8808967440 |
| Rudra   | Agarwal | 9187065308 |
| Vedavalli | Srinath | 9579531991 |
+-----+-----+-----+
8 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

Triggers and Cursors

1. Create a trigger to check and update the validity of newly inserting bills.

Query :

```
DELIMITER $$
CREATE TRIGGER buying_date
BEFORE INSERT
ON BILL
FOR EACH ROW
BEGIN
    IF new.Present_date > new.Last_valid_date THEN SET new.Validity
= 'Invalid';
    END IF;
END$$
DELIMITER ;
INSERT INTO BILL VALUES('B_049', 160, '2022-11-114', '2022-11-17',
'2022-11-18', 'Valid', 'SHK 003', 3453484810);
```

Output :

B_047	75	2021-01-28	2021-01-31	2021-01-28	Valid	SHK_004	4693726418
B_048	25	2021-06-14	2021-06-17	2021-06-17	Valid	SHK_004	4693726418
B_049	160	2022-11-14	2022-11-17	2022-11-18	Invalid	SHK_003	3453484810

49 rows in set (0.002 sec)

2. Create the trigger to show error when age of the newly inserting dependent is less than 10.

Query :

```
DELIMITER $$
CREATE TRIGGER dependent_age
BEFORE INSERT
ON DEPENDENT
FOR EACH ROW
BEGIN
    DECLARE error_msg VARCHAR(300);
    SET error_msg = ("Age of the dependent should be 10 or more");
    IF new.Age < 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = error_msg;
    END IF;
END $$
DELIMITER ;
INSERT INTO DEPENDENT VALUES(3453484810, 'Aniket', '2000-11-3', 'M', '9', 'Son');
```

Output :

```
MariaDB [ration_card_system]> INSERT INTO DEPENDENTS VALUES(3453484810, 'Aniket', '2000-11-3', 'M', '9', 'Son');
ERROR 1644 (45000): Age of the dependent should be 10 or more
MariaDB [ration_card_system]>
```

3. Use cursor to add valid bills to new table.

Query :

```
CREATE TABLE VALID_BILLS(  
    Bill_id VARCHAR(10),  
    Total_cost INT(5),  
    Issued_date DATE,  
    Last_valid_date DATE,  
    Present_date DATE,  
    Validity VARCHAR(20),  
    Shopkeeper_id VARCHAR(10),  
    RFID BIGINT(10)  
);  
  
DELIMITER $$  
CREATE PROCEDURE get_valid_bills()  
BEGIN  
    DECLARE done int default 0;  
    DECLARE b_total_cost INTEGER;  
    DECLARE b_RFID bigint;  
    DECLARE b_bill_id, b_validity, b_shopkeeper_id varchar(20);  
    DECLARE b_issued_date, b_last_valid_date, b_present_date date;  
    DECLARE cur cursor for select * from BILL WHERE BILL.VALIDITY =  
'Valid';  
    DECLARE continue handler for not found set done = 1;  
  
    OPEN cur;  
  
    label: LOOP  
        fetch cur into b_bill_id, b_total_cost, b_issued_date,  
b_last_valid_date, b_present_date, b_validity, b_shopkeeper_id,  
b_RFID;  
        if done = 1 then leave label;  
        end if;  
        INSERT INTO VALID_BILLS VALUES(b_bill_id, b_total_cost,  
b_issued_date, b_last_valid_date, b_present_date, b_validity,  
b_shopkeeper_id, b_RFID);  
    end loop;  
  
    close cur;  
end$$  
DELIMITER ;
```

Output :

```
MariaDB [ration_card_system]> CALL get_valid_bills;  
Query OK, 41 rows affected (0.047 sec)
```

```
MariaDB [ration_card_system]> select * from valid_bills;
```

Bill_id	Total_cost	Issued_date	Last_valid_date	Present_date	Validity	Shopkeeper_id	RFID
B_001	140	2021-06-07	2021-06-09	2021-06-07	Valid	SHK_001	3453484810
B_002	170	2021-02-14	2021-02-17	2021-02-16	Valid	SHK_001	3453484810
B_003	105	2021-07-13	2021-07-16	2021-07-14	Valid	SHK_003	6815974590
B_005	170	2021-10-10	2021-10-13	2021-10-10	Valid	SHK_004	5483807477
B_006	55	2021-12-22	2021-12-25	2021-12-24	Valid	SHK_004	5483807477
B_007	70	2021-11-09	2021-11-12	2021-11-10	Valid	SHK_001	9187065308
B_008	115	2021-07-29	2021-08-01	2021-07-29	Valid	SHK_001	9187065308
B_009	170	2021-01-06	2021-01-09	2021-01-09	Valid	SHK_005	1700494946
B_011	100	2021-05-08	2021-05-11	2021-05-11	Valid	SHK_006	4478160351
B_012	35	2021-07-23	2021-07-26	2021-07-23	Valid	SHK_006	4478160351
B_013	10	2021-07-08	2021-07-11	2021-07-09	Valid	SHK_001	6977990400
B_014	95	2021-08-16	2021-08-19	2021-08-16	Valid	SHK_001	6977990400
B_016	35	2021-05-06	2021-05-09	2021-05-09	Valid	SHK_005	7575833003
B_017	65	2021-11-29	2021-12-02	2021-11-30	Valid	SHK_002	8727666988
B_018	75	2021-06-09	2021-06-12	2021-06-09	Valid	SHK_002	8727666988
B_019	30	2021-05-02	2021-05-05	2021-05-02	Valid	SHK_006	2192227997
B_021	40	2021-01-03	2021-01-06	2021-01-03	Valid	SHK_003	8979106724
B_022	40	2021-06-25	2021-06-28	2021-06-28	Valid	SHK_003	8979106724
B_023	30	2021-03-15	2021-03-18	2021-03-17	Valid	SHK_004	6597759624
B_024	35	2021-05-14	2021-05-17	2021-05-14	Valid	SHK_004	6597759624
B_025	170	2021-02-16	2021-02-19	2021-02-19	Valid	SHK_001	6283738531
B_026	170	2021-09-21	2021-09-24	2021-09-22	Valid	SHK_001	6283738531
B_027	25	2021-09-08	2021-09-11	2021-09-09	Valid	SHK_005	8646892910
B_028	40	2021-03-11	2021-03-14	2021-03-13	Valid	SHK_005	8646892910
B_030	75	2021-02-01	2021-02-04	2021-02-01	Valid	SHK_002	9579531991
B_031	55	2021-12-12	2021-12-15	2021-12-12	Valid	SHK_003	7243413566
B_032	100	2021-06-24	2021-06-27	2021-06-24	Valid	SHK_003	7243413566
B_033	40	2021-03-19	2021-03-22	2021-03-19	Valid	SHK_006	8565854116
B_034	30	2021-10-10	2021-10-13	2021-10-11	Valid	SHK_006	8565854116
B_035	35	2021-08-28	2021-08-31	2021-08-28	Valid	SHK_001	6107074114
B_037	30	2021-01-29	2021-02-01	2021-01-29	Valid	SHK_004	3740546453
B_038	140	2021-05-17	2021-05-20	2021-05-17	Valid	SHK_004	3740546453
B_039	70	2021-09-25	2021-09-28	2021-09-28	Valid	SHK_002	1637382417
B_040	75	2021-12-10	2021-12-13	2021-12-10	Valid	SHK_002	1637382417
B_041	115	2021-06-12	2021-06-15	2021-06-14	Valid	SHK_006	8808967440
B_042	160	2021-07-12	2021-07-15	2021-07-12	Valid	SHK_006	8808967440
B_043	10	2021-03-02	2021-03-05	2021-03-03	Valid	SHK_002	2624574553
B_045	25	2021-02-07	2021-02-10	2021-02-09	Valid	SHK_005	8945871085
B_046	30	2021-09-20	2021-09-23	2021-09-23	Valid	SHK_005	8945871085
B_047	75	2021-01-28	2021-01-31	2021-01-28	Valid	SHK_004	4693726418
B_048	25	2021-06-14	2021-06-17	2021-06-17	Valid	SHK_004	4693726418


```
41 rows in set (0.001 sec)
```

4. Use cursor to add name and email id of customers to new table.

Query :

```
CREATE TABLE CUSTOMER_EMAIL(  
    Fname varchar(10),  
    Lname varchar(10),  
    Email_id VARCHAR(50)  
);  
  
DELIMITER $$  
CREATE PROCEDURE create_email_list ()  
BEGIN  
    DECLARE done INTEGER DEFAULT 0;  
    DECLARE Fname varchar(10) ;  
    DECLARE Lname varchar(10) ;  
    DECLARE emailAddress varchar(100) ;  
    DECLARE curEmail CURSOR FOR SELECT C_Fname, C_Lname, C_Email_id  
FROM CUSTOMER;  
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;  
  
    OPEN curEmail;  
  
    LABEL: LOOP  
        FETCH curEmail INTO Fname, Lname, emailAddress;  
        IF done = 1 THEN  
            LEAVE LABEL;  
        END IF;  
        INSERT INTO CUSTOMER_EMAIL VALUES(Fname, Lname, emailAddress);  
    END LOOP;  
  
    CLOSE curEmail;  
END$$  
  
DELIMITER ;
```

Output :


 Showing rows 0 - 24 (25 total, Query took 0.0004 seconds.)

```
SELECT * FROM `customer_email`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

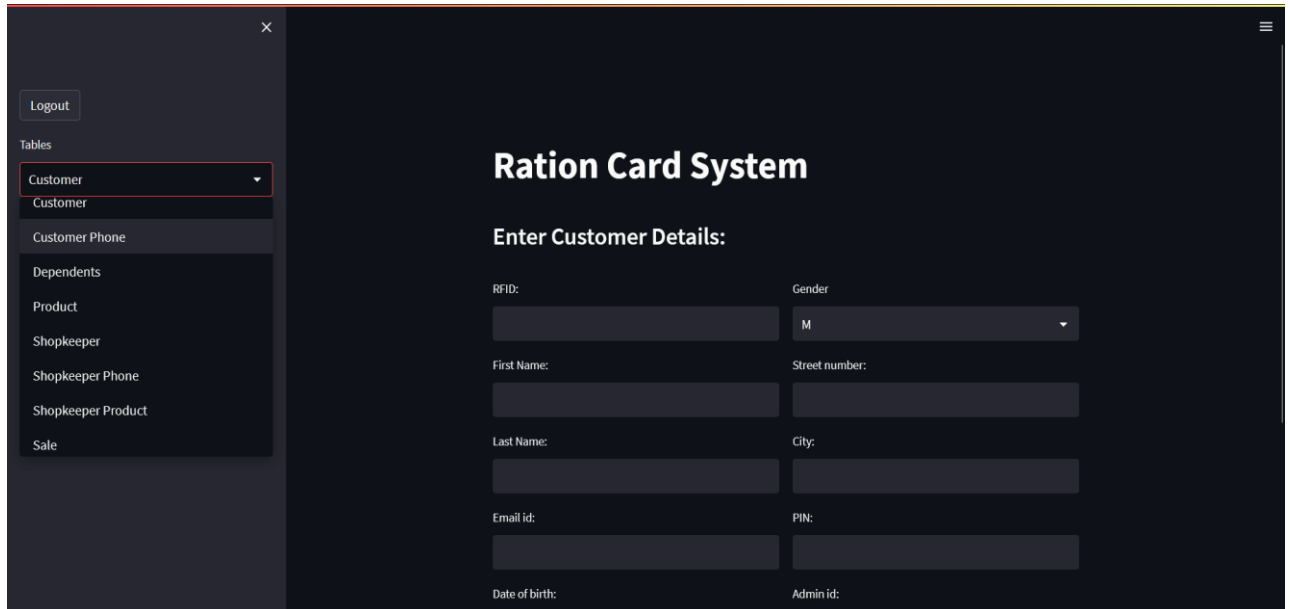
☐ Show all | Number of rows: Filter rows:

Extra options

Fname	Lname	Email_id
Virat	Kohli	kohlivirat@gmail.com
Anshu	Bhardwaj	bhardwajanshu@gmail.com
Rahul	Khanna	khannarahul@gmail.com
Samanta	Prabhu	prabhusamanta@gmail.com
Srihari	Udupa	udupasrihari@gmail.com
Ajit	Ullal	ullalajit@gmail.com
Arun	Kumar	kumararun@gmail.com
Nirmala	Seturaman	seturamannirmala@gmail.com
Shradha	Nayar	nayarshradha@gmail.com
Sai Deepak	Reddy	reddysai@gmail.com
Margaret	Alva	alvamargaret@gmail.com
Rashkit	Shetty	shettyrashkit@gmail.com
 Console	Gandhi	gandhirahul@gmail.com
Muhammed	Ali	alimihammed@gmail.com
Smriti	Irani	iranismriti@gmail.com
Subbu	Saravana	saravanasubbu@gmail.com
Ajit	Sethi	sethiajit@gmail.com
Sheela	Dixit	dixitsheela@gmail.com
Rishi	Sunak	sunakrishi@gmail.com
Arjun	Allu	alluarjun@gmail.com
Suma	Sampat	sampatsuma@gmail.com
Satish	Kotian	kotiansatish@gmail.com
Kiran	Bedi	bedikiran@gmail.com
Rudra	Agarwal	agarwalrudra@gmail.com
Vedavalli	Srinath	srinathvedavalli@gmail.com

FRONT END

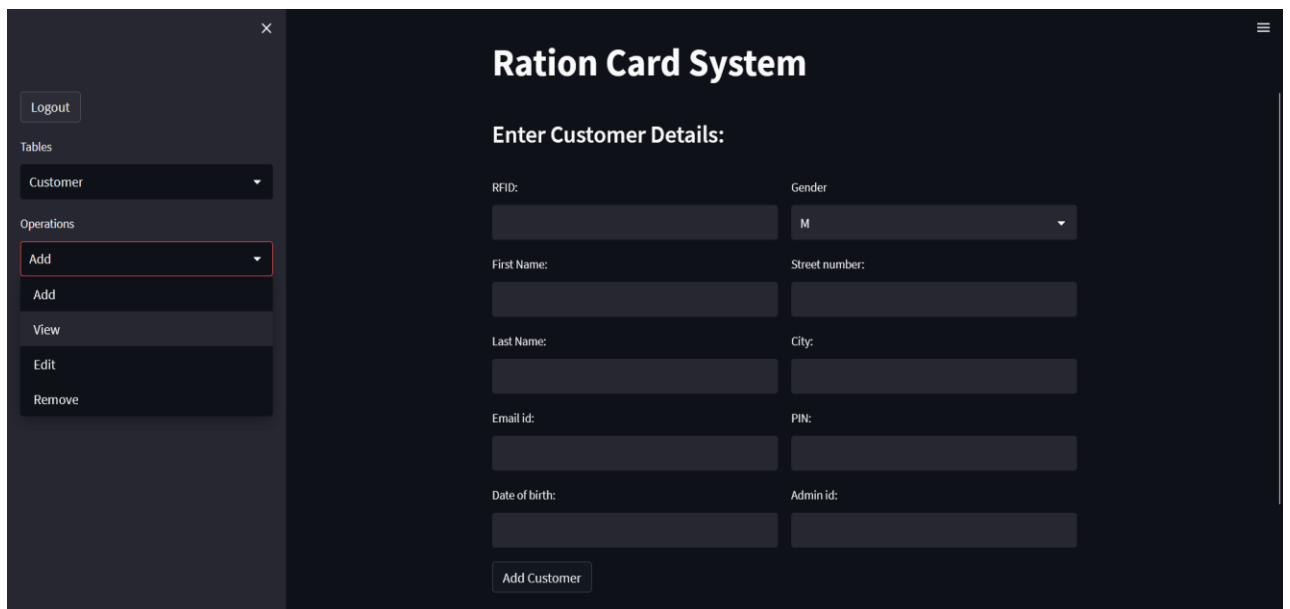
1. Tables that admin have the access.



The screenshot shows the 'Ration Card System' admin interface. On the left sidebar, under the 'Tables' section, the 'Customer' table is selected. The main content area displays the 'Enter Customer Details:' form with the following fields:

- RFID:
- Gender:
- First Name:
- Street number:
- Last Name:
- City:
- Email id:
- PIN:
- Date of birth:
- Admin id:

2. Operations that admin can perform on these tables.



The screenshot shows the 'Ration Card System' admin interface. On the left sidebar, under the 'Operations' section, the 'Add' operation is selected. The main content area displays the 'Enter Customer Details:' form with the following fields:

- RFID:
- Gender:
- First Name:
- Street number:
- Last Name:
- City:
- Email id:
- PIN:
- Date of birth:
- Admin id:

At the bottom of the form, there is an 'Add Customer' button.

3. Tables that shopkeeper have the access.

The screenshot shows a web application titled "Ration Card System". On the left, there is a sidebar menu with a "Logout" button at the top. Below it, a section labeled "Tables" contains a dropdown menu where "Bill" is selected. Other options in the "Tables" list include "Bill", "Bill Product", "Product Customer", "Shopkeeper Product", "Sell", "Sale", and "Query". The main content area on the right is titled "View all Product" and contains a form with the following fields: "Bill id:", "Present Date:", "Total cost:", "Validity:", "Issued date:", "Shopkeeper id:", "Last Valid Date:", and "RFID:". Each field has a corresponding input box. At the bottom of the form is an "Add Bill" button.

4. Operations that shopkeeper can perform on these tables.

The screenshot shows the same "Ration Card System" interface. In the sidebar menu, the "Operations" section is now expanded, and the "Add" option is selected. The "Add" option is highlighted with a red border. The main content area remains the same, showing the "View all Product" form with fields for "Bill id:", "Present Date:", "Total cost:", "Validity:", "Issued date:", "Shopkeeper id:", "Last Valid Date:", and "RFID:", and an "Add Bill" button at the bottom.

5. Selling products.

×

Logout

Tables

Sell

Ration Card System

Bill id:RFID:

Shopkeeper id:

Issued date:

2022/11/27

Last Valid Date:

2022/11/27

Present Date:

2022/11/27

Validity:

ProductsProductsProductsProductsProductsProducts

×

Logout

Tables

Sell

Products	Products	Products	Products	Products	Products
Daal	Daal	Daal	Daal	Daal	Daal
Product name :	Product name :	Product name :	Product name :	Product name :	Product name :
Daal	Daal	Daal	Daal	Daal	Daal
Quantity:	Quantity:	Quantity:	Quantity:	Quantity:	Quantity:
1 - +	1 - +	1 - +	1 - +	1 - +	1 - +
Price :	Price :	Price :	Price :	Price :	Price :
0 - +	0 - +	0 - +	0 - +	0 - +	0 - +
Total cost per product:	Total cost per product:	Total cost per product:	Total cost per product:	Total cost per product:	Total cost per product:
0 - +	0 - +	0 - +	0 - +	0 - +	0 - +
Total cost :					
0					
Add Product					

6. Pie Chart for Sales between given Dates.

