DBMS Project

Mobile Manufacture Company Database

Savvy Jain 21CSB0F15

ACKNOWLEDGEMENT

Inspiration and motivation play a key role in the success of any venture. Successful completion of any project requires proper guidance and help. We express our gratitude to the following people who have made this possible.

● We thank DBMS professors, **Dr. T Ramakrishnudu** and **Dr. RBV Subramanyam** for giving us the opportunity to develop deep into database concepts. Their lessons have been really helpful in learning the fundamental DBMS concepts like relations, E-R diagrams, Structured Query Language, functional dependencies, normalization, indexing, etc. in detail.

● We extend our thanks to the whole CSE Department for being so helpful and available whenever we require any kind of guidance. The DBMS Lab faculty has also trained and made us practice the Structured Query Language with the help of the periodic assignments on Google Classroom.

● We thank the NITW management, Dean - Academic and Director - NITW, for giving us this opportunity to discover this subject in the course, and enter new avenues in the field of database management. We also thank our parents, elders and well-wishers for being there with us and giving us all kinds of technical and moral support.

Regards,

TABLE OF CONTENT

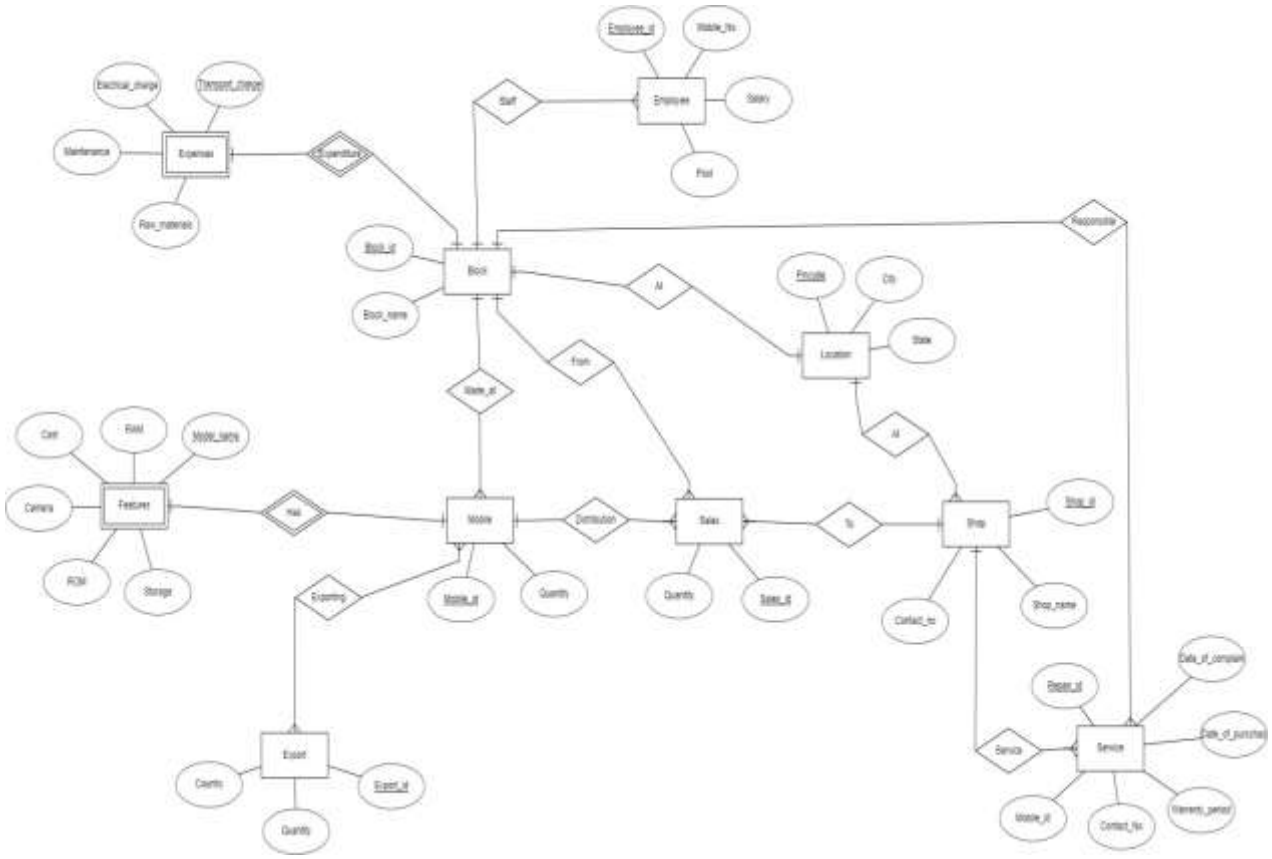
PROBLEM STATEMENT

In this project, we design a database about a manufacturing company that manufactures Samsung mobile phones which provides information about mobile features, suppliers, receivers, sales, and the budget of the company. From this database we can get information about employees, the most popular mobile phones, totalsales, profit, and unsold mobile phones.

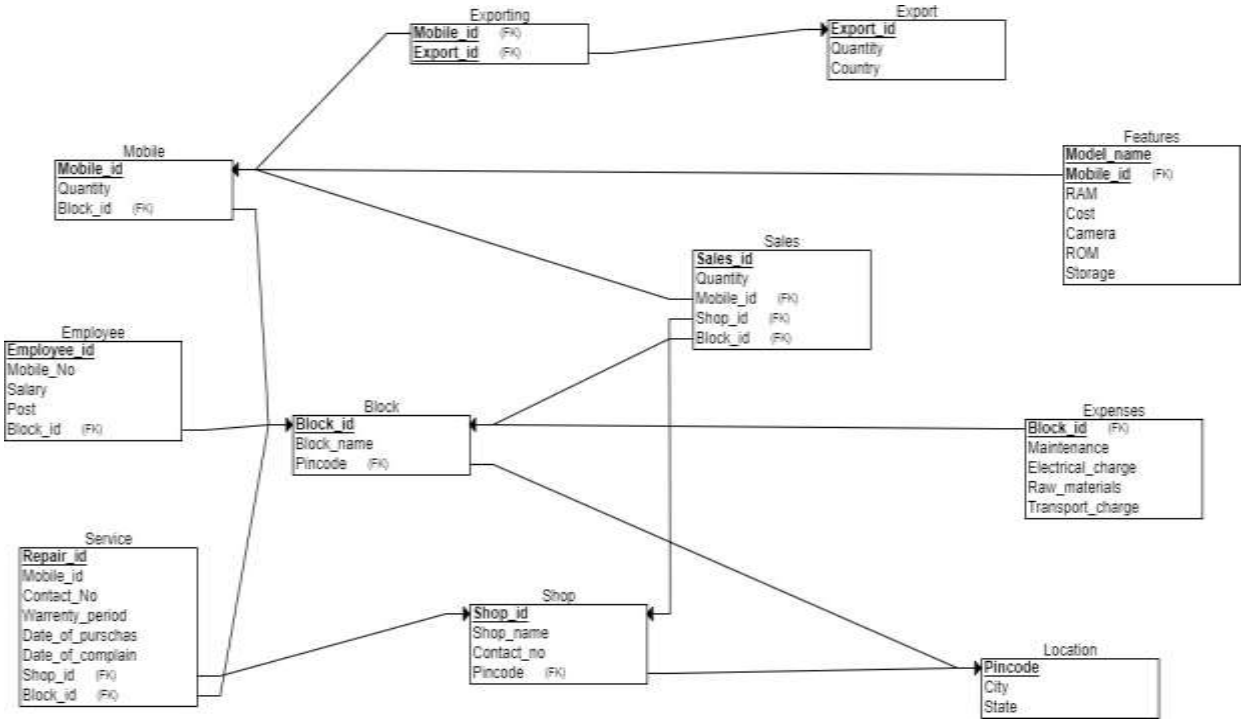
ASSUMPTION

A mobile is manufactured at particular block and block can manufacture any type of mobiles.

ER DIAGRAM


Employee_id
Mobile_No
Staff
Employee
Salary
Post
Electrical_charge
Transport_charge
Maintenance
Expenses
Expenditure
Raw_materials
Reconcile
Block_id
Block
Block_name
All
Pincode
City
State
Location
From
Make_at
Color
RAM
Mobile_name
Camera
Features
Has
ROM
Storage
Mobile
Distribution
Sales
To
Shop
Shop_id
Shop_name
Contact_no
Mobile_id
Quantity
Quantity
Sales_id
Exporting
Export
Country
Export_id
Quantity
Date_of_complaint
Repair_id
Service
Service
Date_of_purchase
Mobile_id
Contact_No
Warranty_period

REALTIONAL DIAGRAM



RELATIONS

1) Location:

The location relation is the collection of all the places where different shops and blocks are located. It contains the pincode, city and state of a particular place.

Pincode is the primary key here.

Pincode	City	State
110001	New Delhi	Delhi
400001	Mumbai	Maharashtra
560002	Bangalore	Karnataka
500018	Hyderabad	Telangana
600034	Chennai	Tamil Nadu
700043	Kolkata	West Bengal

2) Block:

The Block relation is the collection of data regarding the no manufacturing plants (i.e. Blocks), Block name, Block_id and pincode which describes where it is located. It has Block_id as the primary key here.

Block_id	Block_name	Pincode
B101	Left wing	400001

B103	Right wing	110001
B105	Central wing	560002
B109	Head Quarters	500018

3) Mobile:

The Mobile relation is the collection of data regarding the total stock of mobiles in all the manufacturing plants in india. This includes data like Mobile_id, Block_id in which the particular Mobile_id is manufactured and the Quantity it manufactured. Mobile_id is the primary key here.

Mobile id	Quantity	Block_id
SG19A23A	5000	B101
SG19A23B	1000	B101
SG20F12	3000	B103
SG20M32A	1500	B105
SG20M32B	9000	B105
SG21S22A	1000	B109
SG21S22B	500	B109
SG22M52A	2500	B105
SG22M52B	2000	B105

4) Shop:

The Shop relation is the collection of data related to the shops to which the mobiles are distributed. This includes data like Shop_id, Shop_name, contact_no and pincode (where it is located).Shop_id is the primary key here.

Shop_id	Shop_name	Contact_no	Pincode
101	Abc	123	500018
108	Def	456	400001
201	Ghi	789	110001
301	Jkl	101	600043
401	Mno	112	560002
501	Pqr	131	700043
601	Stu	415	110001
701	Vwx	161	500018
801	Yza	718	400001
901	Bcd	192	700043

5) Sales:

The Sales relation is the collection of data related to the Sales i.e. it maintains the data of how many mobiles are sent and to which shop it was sent. This includes data like Sales_id, Shop_id (to which shop it was distributed) , Mobile_id (which mobile type is being distributed) , how much quantity is being distributed and the Block_id (from which block the mobile_id is being manufactured). Sales_id is the primary key here.

Sale_id	Quantity	Mobile_id	Shop_id	Block_id
S201	1500	SG19A23A	201	B101
S203	1000	SG19A23A	401	B101
S205	500	SG19A23A	601	B101
S209	300	SG19A23B	101	B101
S213	300	SG19A23B	901	B101
S214	200	SG19A23B	301	B101
S215	1000	SG20F12	301	B103
S217	1000	SG20F12	201	B103
S219	1000	SG20F12	108	B103
S221	500	SG20M32A	108	B105
S223	500	SG20M32A	201	B105
S225	400	SG20M32A	701	B105
S227	1000	SG20M32B	601	B105
S229	500	SG20M32B	201	B105
S231	1000	SG20M32B	501	B105
S233	100	SG21S22A	501	B109
S235	200	SG21S22A	201	B109
S237	100	SG21S22B	401	B109
S239	200	SG21S22B	201	B109
S241	200	SG21S22B	901	B109
S243	100	SG22M52A	801	B105
S245	100	SG22M52A	201	B105
S247	500	SG22M52B	701	B105
S249	500	SG22M52B	101	B105
S251	500	SG22M52B	201	B105

6) Employee:

The Employee relation is the collection of data related to the Employee i.e. it maintains the data regarding the employees working in a particular block and their details. This includes data like Employee_id, Mobile_no, Salary, Post and Block_id. Employee_id is the primary key here.

Employee_id	Block_id	Mobile_No	Salary	Post
207141	B101	1111111111	100000	Manager
207143	B101	2222222222	80000	Asst. Manager
207358	B103	3333333333	50000	Clerk
207362	B103	4444444444	200000	Asst. Manager
207927	B109	5555555555	300000	Manager
207182	B101	6666666666	150000	Accountant
207518	B105	7777777777	250000	Asst. Manager
207905	B109	8888888888	100000	Accountant
207532	B105	9999999999	50000	Clerk
207372	B103	1010101010	80000	Accountant

7) Expenses:

The Expenses relation is the collection of data related to the Expenses like Electrical_charge, Transport_charge, Maintenance and Raw_materials for a particular Block_id. Block_id and Transport_charge combinedly act as primary key here.

Block_id	Maintenance	Electric_charge	Raw_materials	Transport_charge
B101	10000	5000	2000	1000
B103	15000	7000	1000	1500
B105	12000	8000	500	1300
B109	13000	4000	1300	1200

8) Export:

The Export relation is the collection of data related to the Export like Export_id, Mobile_id, Quantity and Country. Each Export_id represents the quantity of a particular Mobile_id being exported to a country.

Export_id	Quantity	Country
E501	1000	USA
E502	2000	Georgia
E503	1500	Canada
E504	4000	New York
E505	1000	Seoul
E506	2000	London
E507	2500	Paris
E508	1500	Italy

9)Exporting:

The Exporting relation is the collection of data related to the Exporting like Mobile_id and Export_id. This relation acts as a joint relationship between Mobile and Export. Mobile_id and Export_id combinedly act as primary key here.

Mobile_id	Export_id
SG19A23A	E503
SG20F12	E501
SG20M32B	E508
SG21S22B	E504
SG19A23B	E502
SG20M32A	E507
SG21S22A	E506
SG22M52A	E505
SG22M52B	E505
SG20F12	E503
SG21S22A	E502
SG22M52B	E503
SG20M32B	E504
SG19A23A	E505
SG21S22B	E507
SG19A23B	E505
SG22M52A	E503
SG20M32A	E508

10) Features:

The Features relation is the collection of data related to the Features of a Mobile_id like RAM, ROM, Cost, Camera, Storage and Model_name.

Mobile_id and Model_name combinedly acts as primary key here.

M_name	Mobile_id	RAM	Cost	Camera	ROM	Storage
AGalaxy	SG19A23A	4GB	12,000	32.0MP	32GB	32GB
AGalaxy	SG19A23B	8GB	15,000	40.0MP	64GB	64GB
FGalaxy	SG20F12	4GB	10,000	32.0MP	16GB	16GB
MGalaxy	SG20M32A	4GB	25,000	50.0MP	32GB	64GB
MGalaxy	SG20M32B	8GB	28,000	60.0MP	64GB	128GB
SGalaxy	SG21S22A	16GB	50,000	90.0MP	256GB	256GB
SGalaxy	SG21S22B	16GB	60,000	108.0MP	256GB	256GB
MGalaxy	SG22M52A	8GB	30,000	64.0MP	128GB	128GB
MGalaxy	SG22M52B	16GB	40,000	80.0MP	256GB	256GB

11) Service:

The Service relation is the collection of data related to the repairs received from the consumers which includes data like Repair_id, Date_of_complain, Date_of_purchase, Warrenty_period, Contact_no, Mobile_no, Shop_id and Block_id.

Repair_id	Mobile_id	Con_no	War_p	Dop	Doc	Shop_id	Block_id
1	SG20F12	123	2	2018	2022	301	B103
2	SG19A23B	456	1	2020	2022	101	B101
3	SG20F12	789	3	2019	2022	201	B103
4	SG19A23A	257	2	2021	2022	201	B101
5	SG20F12	243	1	2022	2022	108	B103

FUNCTIONAL DEPENDENCIES

1) Location:

PRIMARY KEY : Pincode

F.D: Pincode → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Location is in BCNF form.

2) Block:

PRIMARY KEY : Block_id

F.D : Block_id → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Block is in BCNF form.

3) Mobile:

PRIMARY KEY : Mobile_id

F.D : Mobile_id → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Mobile is in BCNF form.

4) Shop:

PRIMARY KEY : Shop_id

F.D : Shop_id → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Shop is in BCNF form.

5) Sales:

PRIMARY KEY : Sales_id

F.D : Sales_id → R

- As there are no multi-valued variables it is in 1NF.

- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Sales is in BCNF form.

6) Employee:

PRIMARY KEY : Employee_id

F.D : Employee_id → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Employee is in BCNF form.

7) Expenses:

PRIMARY KEY : Block_id,Maintenance

F.D : Block_id,Maintenance → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Expenses is in BCNF form.

8) Export:

PRIMARY KEY : Export_id

F.D : Export_id → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Export is in BCNF form.

9)Exporting:

PRIMARY KEY : Mobile_id,Export_id

F.D : Mobile_id,Export_id → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Exporting is in BCNF form.

10) Features:

PRIMARY KEY : Mobile_name,Mobile_id

F.D : Mobile_name,Mobile_id → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Features is in BCNF form.

11) Service:

PRIMARY KEY : Repair_id

F.D : Repair_id → R

- As there are no multi-valued variables it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is in both 3NF and BCNF form.
- Therefore, the table Service is in BCNF form.

TABLES

LOCATION:

```
CREATE TABLE Location (
Pincode number primary key,
City varchar(20),
State varchar(20)
);
INSERT INTO Location VALUES(110001,'New Delhi','Delhi');
INSERT INTO Location VALUES(400001,'Mumbai','Maharashtra');
```

```
INSERT INTO Location VALUES(560002,'Bangalore','Karnataka');
INSERT INTO Location VALUES(500018,'Hyderabad','Telangana');
INSERT INTO Location VALUES(600043,'Chennai','Tamilnadu');
INSERT INTO Location VALUES(700043,'Kolkata','West Bengal');
select * from location;
```

```
Name     Null?    Type
-------  -------- ------------
PINCODE  NOT NULL NUMBER
CITY               VARCHAR2(20)
STATE              VARCHAR2(20)
```

	PINCODE	CITY	STATE
1	110001	New Delhi	Delhi
2	400001	Mumbai	Maharashtra
3	560002	Bangalore	Karnataka
4	500018	Hyderabad	Telangana
5	600043	Chennai	Tamilnadu
6	700043	Kolkata	West Bengal

BLOCK:

```
CREATE TABLE Block (
    Block_id varchar(20) primary key,
    Block_name varchar(20),
    Pincode number,
    FOREIGN KEY (Pincode)  REFERENCES Location(Pincode)
);
INSERT INTO Block VALUES('B101','Left Wing',400001);
INSERT INTO Block VALUES('B103','Right Wing',110001);
INSERT INTO Block VALUES('B105','Central Wing',560002);
```

```
INSERT INTO Block VALUES('B109','Head Quarters',500018);
```

```
Name          Null?    Type
----------- -------- ------------
BLOCK_ID      NOT NULL VARCHAR2(20)
BLOCK_NAME             VARCHAR2(20)
PINCODE                NUMBER
```

	REPAIR_ID	MOBILE_ID	CONTACT_NO	WARRENTY_PERIOD	DATE_OF_PURCHASE	DATE_OF_COMPLAIN	SHOP_ID	BLOCK_ID
1	1	SG20F12	123	2	2018	2022	301	B103
2	2	SG19A23B	456	1	2020	2022	101	B101
3	3	SG20F12	789	3	2019	2022	201	B103
4	4	SG19A23A	257	2	2021	2022	201	B101
5	5	SG20F12	243	1	2022	2022	108	B103

MOBILE:

```
CREATE TABLE Mobile (
    Mobile_id varchar(20) primary key,
    Quantity number,
    Block_id varchar(20),
    FOREIGN KEY (Block_id) REFERENCES Block(Block_id)
);
INSERT INTO Mobile VALUES('SG19A23A',5000,'B101');
INSERT INTO Mobile VALUES('SG19A23B',1000,'B101');
INSERT INTO Mobile VALUES('SG20F12',3000,'B103');
INSERT INTO Mobile VALUES('SG20M32A',1500,'B105');
INSERT INTO Mobile VALUES('SG20M32B',9000,'B105');
INSERT INTO Mobile VALUES('SG21S22A',1000,'B109');
INSERT INTO Mobile VALUES('SG21S22B',500,'B109');
INSERT INTO Mobile VALUES('SG22M52A',2500,'B105');
INSERT INTO Mobile VALUES('SG22M52B',2000,'B105');
```

```
Name        Null?    Type
----------- -------- ------------
MOBILE_ID   NOT NULL VARCHAR2(20)
QUANTITY             NUMBER
BLOCK_ID             VARCHAR2(20)
```

	MOBILE_ID	QUANTITY	BLOCK_ID
1	SG19A23A	5000	B101
2	SG19A23B	1000	B101
3	SG20F12	3000	B103
4	SG20M32A	1500	B105
5	SG20M32B	9000	B105
6	SG21S22A	1000	B109
7	SG21S22B	500	B109
8	SG22M52A	2500	B105
9	SG22M52B	2000	B105

SHOP:

```
CREATE TABLE Shop(
    Shop_id number primary key,
    Shop_name varchar(20),
    Contact_no number,
    Pincode number,
    FOREIGN KEY (Pincode) REFERENCES Location(Pincode)
);
INSERT INTO Shop VALUES(101,'Abc',123,500018);
INSERT INTO Shop VALUES(108,'Def',456,400001);
INSERT INTO Shop VALUES(201,'Ghi',789,110001);
INSERT INTO Shop VALUES(301,'Jkl',101,600043);
INSERT INTO Shop VALUES(401,'Mno',112,560002);
INSERT INTO Shop VALUES(501,'Pqr',131,700043);
INSERT INTO Shop VALUES(601,'Stu',415,110001);
INSERT INTO Shop VALUES(701,'Vwx',161,500018);
INSERT INTO Shop VALUES(801,'Yza',718,400001);
```

```
INSERT INTO Shop VALUES(901,'Bcd',192,700043);
```

```
Name        Null?    Type
----------- -------- ------------
SHOP_ID     NOT NULL NUMBER
SHOP_NAME            VARCHAR2(20)
CONTACT_NO           NUMBER
PINCODE              NUMBER
```

	SHOP_ID	SHOP_NAME	CONTACT_NO	PINCODE
1	101	Abc	123	500018
2	108	Def	456	400001
3	201	Ghi	789	110001
4	301	Jkl	101	600043
5	401	Mno	112	560002
6	501	Pqr	131	700043
7	601	Stu	415	110001
8	701	Vwx	161	500018
9	801	Yza	718	400001
10	901	Bcd	192	700043

SALES:

```
CREATE TABLE Sales(
    Sales_id varchar(20) primary key,
    Quantity number,
    Mobile_id varchar(20),
    Shop_id number,
    Block_id varchar(20),
    FOREIGN KEY (Mobile_id) REFERENCES Mobile(Mobile_id),
    FOREIGN KEY (Shop_id) REFERENCES Shop(Shop_id),
    FOREIGN KEY (Block_id) REFERENCES Block(Block_id)
);
INSERT INTO Sales VALUES('S201',1500,'SG19A23A',201,'B101');
```

```
INSERT INTO Sales VALUES('S203',1000,'SG19A23A',401,'B101');
INSERT INTO Sales VALUES('S205',500,'SG19A23A',601,'B101');
INSERT INTO Sales VALUES('S209',300,'SG19A23B',101,'B101');
INSERT INTO Sales VALUES('S213',300,'SG19A23B',901,'B101');
INSERT INTO Sales VALUES('S214',200,'SG19A23B',301,'B101');
INSERT INTO Sales VALUES('S215',1000,'SG20F12',301,'B103');
INSERT INTO Sales VALUES('S217',1000,'SG20F12',201,'B103');
INSERT INTO Sales VALUES('S219',1000,'SG20F12',108,'B103');
INSERT INTO Sales VALUES('S221',500,'SG20M32A',108,'B105');
INSERT INTO Sales VALUES('S223',500,'SG20M32A',201,'B105');
INSERT INTO Sales VALUES('S225',400,'SG20M32A',701,'B105');
INSERT INTO Sales VALUES('S227',1000,'SG20M32B',601,'B105');
INSERT INTO Sales VALUES('S229',500,'SG20M32B',201,'B105');
INSERT INTO Sales VALUES('S231',1000,'SG20M32B',501,'B105');
INSERT INTO Sales VALUES('S233',100,'SG21S22A',501,'B109');
INSERT INTO Sales VALUES('S235',200,'SG21S22A',201,'B109');
INSERT INTO Sales VALUES('S237',100,'SG21S22B',401,'B109');
INSERT INTO Sales VALUES('S239',200,'SG21S22B',201,'B109');
INSERT INTO Sales VALUES('S241',200,'SG21S22B',901,'B109');
INSERT INTO Sales VALUES('S243',100,'SG22M52A',801,'B105');
INSERT INTO Sales VALUES('S245',100,'SG22M52A',201,'B105');
INSERT INTO Sales VALUES('S247',500,'SG22M52B',701,'B105');
INSERT INTO Sales VALUES('S249',500,'SG22M52B',101,'B105');
INSERT INTO Sales VALUES('S251',500,'SG22M52B',201,'B105');
```

```
Name        Null?    Type
----------- -------- ------------
SALES_ID    NOT NULL VARCHAR2(20)
QUANTITY             NUMBER
MOBILE_ID            VARCHAR2(20)
SHOP_ID              NUMBER
BLOCK_ID             VARCHAR2(20)
```

	SALES_ID	QUANTITY	MOBILE_ID	SHOP_ID	BLOCK_ID
1	S201	1500	SG19A23A	201	B101
2	S203	1000	SG19A23A	401	B101
3	S205	500	SG19A23A	601	B101
4	S209	300	SG19A23B	101	B101
5	S213	300	SG19A23B	901	B101
6	S214	200	SG19A23B	301	B101
7	S215	1000	SG20F12	301	B103
8	S217	1000	SG20F12	201	B103
9	S219	1000	SG20F12	108	B103
10	S221	500	SG20M32A	108	B105
11	S223	500	SG20M32A	201	B105
12	S225	400	SG20M32A	701	B105
13	S227	1000	SG20M32B	601	B105
14	S229	500	SG20M32B	201	B105
15	S231	1000	SG20M32B	501	B105
16	S233	100	SG21S22A	501	B109
17	S235	200	SG21S22A	201	B109
18	S237	100	SG21S22B	401	B109
19	S239	200	SG21S22B	201	B109
20	S241	200	SG21S22B	901	B109
21	S243	100	SG22M52A	801	B105
22	S245	100	SG22M52A	201	B105
23	S247	500	SG22M52B	701	B105
24	S249	500	SG22M52B	101	B105
25	S251	500	SG22M52B	201	B105

EMPLOYEE:

```
CREATE TABLE Employee(
   Employee_id number primary key,
   Block_id varchar(20),
   Mobile_No number,
   Salary number,
   Post varchar(20),
   FOREIGN KEY (Block_id) REFERENCES Block(Block_id)
);


INSERT INTO Employee values(207141,'B101',1111111111,100000,'Manager');
```

```
INSERT INTO Employee values(207143,'B101',2222222222,80000,'Asst. Manager');
INSERT INTO Employee values(207358,'B103',3333333333,50000,'Clerk');
INSERT INTO Employee values(207362,'B103',4444444444,200000,'Asst. Manager' );
INSERT INTO Employee values(207927,'B109',5555555555,300000,'Manager');
INSERT INTO Employee values(207182,'B101',6666666666,150000,'Accountant');
INSERT INTO Employee values(207518,'B105',7777777777,250000,'Asst. Manager');
INSERT INTO Employee values(207905,'B109',8888888888,100000,'Accountant');
INSERT INTO Employee values(207532,'B105',9999999999,50000,'Clerk');
INSERT INTO Employee values(207372,'B103',1010101010,80000,'Accountant');
```

```
Name         Null?    Type
------------ -------- ------------
EMPLOYEE_ID  NOT NULL NUMBER
BLOCK_ID               VARCHAR2(20)
MOBILE_NO              NUMBER
SALARY                 NUMBER
POST                   VARCHAR2(20)
```

	EMPLOYEE_ID	BLOCK_ID	MOBILE_NO	SALARY	POST
1	207141	B101	1111111111	100000	Manager
2	207143	B101	2222222222	80000	Asst. Manager
3	207358	B103	3333333333	50000	Clerk
4	207362	B103	4444444444	200000	Asst. Manager
5	207927	B109	5555555555	300000	Manager
6	207182	B101	6666666666	150000	Accountant
7	207518	B105	7777777777	250000	Asst. Manager
8	207905	B109	8888888888	100000	Accountant
9	207532	B105	9999999999	50000	Clerk
10	207372	B103	1010101010	80000	Accountant

EXPENSES:

```
CREATE TABLE Expenses(
   Block_id varchar(20),
   Maintenance number,
```

```
    Electric_charge number,
    Raw_material number,
    Trasport_charge number,
    PRIMARY KEY(Block_id,Maintenance),
    FOREIGN KEY(Block_id) REFERENCES Block(Block_id)
);

INSERT INTO Expenses values('B101',10000,5000,2000,1000);
INSERT INTO Expenses values('B103',15000,7000,1000,1500);
INSERT INTO Expenses values('B105',12000,8000,500,1300);
INSERT INTO Expenses values('B109',13000,4000,1300,1200);
```

```
Name             Null?    Type
---------------- -------- ------------
BLOCK_ID         NOT NULL VARCHAR2(20)
MAINTENANCE      NOT NULL NUMBER
ELECTRIC_CHARGE           NUMBER
RAW_MATERIAL              NUMBER
TRASPORT_CHARGE           NUMBER
```

	BLOCK_ID	MAINTENANCE	ELECTRIC_CHARGE	RAW_MATERIAL	TRASPORT_CHARGE
1	B101	10000	5000	2000	1000
2	B103	15000	7000	1000	1500
3	B105	12000	8000	500	1300
4	B109	13000	4000	1300	1200

EXPORT:

```
CREATE TABLE Export(
Export_id varchar(20) primary key,
Quantity number,
Country varchar(20));
```

```
INSERT INTO Export values('E501',1000,'USA');
INSERT INTO Export values('E502',2000,'Georgia');
INSERT INTO Export values('E503',1500,'Canada');
INSERT INTO Export values('E504',4000,'New York');
INSERT INTO Export values('E505',1000,'Seoul');
INSERT INTO Export values('E506',2000,'London');
INSERT INTO Export values('E507',2500,'Paris');
INSERT INTO Export values('E508',1500,'Italy');
```

```
Name       Null?    Type
---------- -------- ------------
EXPORT_ID  NOT NULL VARCHAR2(20)
QUANTITY            NUMBER
COUNTRY             VARCHAR2(20)
```

	EXPORT_ID	QUANTITY	COUNTRY
1	E501	1000	USA
2	E502	2000	Georgia
3	E503	1500	Canada
4	E504	4000	New York
5	E505	1000	Seoul
6	E506	2000	London
7	E507	2500	Paris
8	E508	1500	Italy

EXPORTING:

```
CREATE TABLE Exporting(
Mobile_id varchar(20),
Export_id varchar(20),
PRIMARY KEY(Mobile_id,Export_id),
FOREIGN KEY(Mobile_id) REFERENCES Mobile(Mobile_id),
FOREIGN KEY(Export_id) REFERENCES Export(Export_id)
);
INSERT INTO Exporting values('SG19A23A','E503');
```

```
INSERT INTO Exporting values('SG20F12','E501');
INSERT INTO Exporting values('SG20M32B','E508');
INSERT INTO Exporting values('SG21S22B','E504');
INSERT INTO Exporting values('SG19A23B','E502');
INSERT INTO Exporting values('SG20M32A','E507');
INSERT INTO Exporting values('SG21S22A','E506');
INSERT INTO Exporting values('SG22M52A','E505');
INSERT INTO Exporting values('SG22M52B','E505');
INSERT INTO Exporting values('SG20F12','E503');
INSERT INTO Exporting values('SG21S22A','E502');
INSERT INTO Exporting values('SG22M52B','E503');
INSERT INTO Exporting values('SG20M32B','E504');
INSERT INTO Exporting values('SG19A23A','E505');
INSERT INTO Exporting values('SG21S22B','E507');
INSERT INTO Exporting values('SG19A23B','E505');
INSERT INTO Exporting values('SG22M52A','E503');
INSERT INTO Exporting values('SG20M32A','E508');
```

```
Name       Null?    Type
---------- -------- ------------
MOBILE_ID  NOT NULL VARCHAR2(20)
EXPORT_ID  NOT NULL VARCHAR2(20)
```

	MOBILE_ID	EXPORT_ID
1	SG19A23A	E503
2	SG19A23A	E505
3	SG19A23B	E502
4	SG19A23B	E505
5	SG20F12	E501
6	SG20F12	E503
7	SG20M32A	E507
8	SG20M32A	E508
9	SG20M32B	E504
10	SG20M32B	E508
11	SG21S22A	E502
12	SG21S22A	E506
13	SG21S22B	E504
14	SG21S22B	E507
15	SG22M52A	E503
16	SG22M52A	E505
17	SG22M52B	E503
18	SG22M52B	E505

FEATURES:

```
CREATE TABLE Features(
Mobile_name varchar(20),
Mobile_id varchar(20),
RAM varchar(20),
Cost number,
Camera varchar(20),
ROM varchar(20),
Storage varchar(20),
PRIMARY KEY(Mobile_name,Mobile_id),
FOREIGN KEY(Mobile_id) REFERENCES Mobile(Mobile_id)
);
INSERT INTO Features values('AGalaxy','SG19A23A','4GB',12000,'13.0MP','32GB','32GB');
INSERT INTO Features values('AGalaxy','SG19A23B','8GB',15000,'40.0MP','64GB','64GB');
INSERT INTO Features values('FGalaxy','SG20F12','4GB',10000,'32.0MP','16GB','16GB');
INSERT INTO Features values('MGalaxy','SG20M32A','4GB',25000,'50.0MP','32GB','64GB');
```

```
INSERT INTO Features values('MGalaxy','SG20M32B','8GB',28000,'60.0MP','64GB','128GB');
INSERT INTO Features values('SGalaxy','SG21S22A','16GB',50000,'90.0MP','256GB','256GB');
INSERT INTO Features values('SGalaxy','SG21S22B','16GB',60000,'108.0MP','256GB','256GB');
INSERT INTO Features values('MGalaxy','SG22M52A','8GB',30000,'64.0MP','128GB','128GB');
INSERT INTO Features values('MGalaxy','SG22M52B','16GB',40000,'80.0MP','256GB','256GB');
```

```
Name          Null?    Type
------------ -------- ------------
MOBILE_NAME  NOT NULL VARCHAR2(20)
MOBILE_ID    NOT NULL VARCHAR2(20)
RAM                   VARCHAR2(20)
COST                  NUMBER
CAMERA                VARCHAR2(20)
ROM                   VARCHAR2(20)
STORAGE               VARCHAR2(20)
```

	MOBILE_NAME	MOBILE_ID	RAM	COST	CAMERA	ROM	STORAGE
1	AGalaxy	SG19A23A	4GB	12000	13.0MP	32GB	32GB
2	AGalaxy	SG19A23B	8GB	15000	40.0MP	64GB	64GB
3	FGalaxy	SG20F12	4GB	10000	32.0MP	16GB	16GB
4	MGalaxy	SG20M32A	4GB	25000	50.0MP	32GB	64GB
5	MGalaxy	SG20M32B	8GB	28000	60.0MP	64GB	128GB
6	SGalaxy	SG21S22A	16GB	50000	90.0MP	256GB	256GB
7	SGalaxy	SG21S22B	16GB	60000	108.0MP	256GB	256GB
8	MGalaxy	SG22M52A	8GB	30000	64.0MP	128GB	128GB
9	MGalaxy	SG22M52B	16GB	40000	80.0MP	256GB	256GB

SERIVICE:

```
CREATE TABLE Service (
    Repair_id varchar(20),
    Mobile_id varchar(20),
    Contact_No number,
    Warrenty_period number,
    Date_of_Purchase number,
    Date_of_complain number,
```

```
    Shop_id number,
    Block_id varchar(20),
    PRIMARY KEY(Repair_id),
    FOREIGN KEY(Block_id) REFERENCES Block(Block_id),
    FOREIGN KEY(Shop_id) REFERENCES Shop(Shop_id)
);
INSERT INTO Service values(1,'SG20F12',123,2,2018,2022,301,'B103');
INSERT INTO Service values(2,'SG19A23B',456,1,2020,2022,101,'B101');
INSERT INTO Service values(3,'SG20F12',789,3,2019,2022,201,'B103');
INSERT INTO Service values(4,'SG19A23A',257,2,2021,2022,201,'B101');
INSERT INTO Service values(5,'SG20F12',243,1,2022,2022,108,'B103');
```

```
Name                 Null?    Type
-------------------- -------- ------------
REPAIR_ID            NOT NULL VARCHAR2(20)
MOBILE_ID                     VARCHAR2(20)
CONTACT_NO                    NUMBER
WARRENTY_PERIOD               NUMBER
DATE_OF_PURCHASE              NUMBER
DATE_OF_COMPLAIN              NUMBER
SHOP_ID                       NUMBER
BLOCK_ID                      VARCHAR2(20)
```

	REPAIR_ID	MOBILE_ID	CONTACT_NO	WARRENTY_PERIOD	DATE_OF_PURCHASE	DATE_OF_COMPLAIN	SHOP_ID	BLOCK_ID
1	1	SG20F12	123	2	2018	2022	301	B103
2	2	SG19A23B	456	1	2020	2022	101	B101
3	3	SG20F12	789	3	2019	2022	201	B103
4	4	SG19A23A	257	2	2021	2022	201	B101
5	5	SG20F12	243	1	2022	2022	108	B103

GENERAL QUERIES

Q1) Finding mobile phones within a certain price range and with certain features.

Example:

Find mobiles that cost between 10,00 and 15000 and come with 4GB of RAM.

```
select mobile_id,cost,ram

from Mobile natural join Features

where (cost between 10000 and 15000) and RAM ='4GB' order by cost ;
```

Output:

	MOBILE_ID	COST	RAM
1	SG20F12	10000	4GB
2	SG19A23A	12000	4GB

Q2) Finding mobile phones with certain features at a minimum price.

Example:

Find mobiles with 8GB RAM at minimum cost.

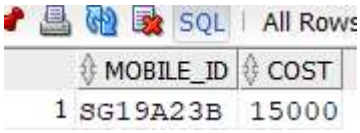
```
select mobile_id,cost

from features

where RAM='8GB' and cost=

      (select min(cost) from (select mobile_id,cost from features where RAM='8GB'));
```

Output:



	MOBILE_ID	COST
1	SG19A23B	15000

Q3) Finding the most and least purchased mobile devices.

Most purchased:

```
select mobile_id,quantity from
  (select mobile_id,sum(quantity) as quantity from sales group by mobile_id)
    where quantity=(select max(quantity) from
   (select distinct mobile_id as mobile_id,sum(quantity) as quantity from sales
group by mobile_id));
```

Output:

	MOBILE_ID	QUANTITY
1	SG20F12	3000
2	SG19A23A	3000

Least purchased:

```
select mobile_id,quantity from
   (select mobile_id,sum(quantity) as quantity from sales group by mobile_id)
   where quantity=(select min(quantity) from
   (select distinct mobile_id as mobile_id,sum(quantity) as quantity from sales
group by mobile_id));
```

Output:

	MOBILE_ID	QUANTITY
1	SG22M52A	200

Q4) Finding id of the block from where manufactured mobiles get repair most frequently.

```
select block_id,count from
 (select block_id,count(*) as count from Service group by block_id) where count=
 (select max(count) from
 (select block_id,count(*) as count from Service group by block_id));
```

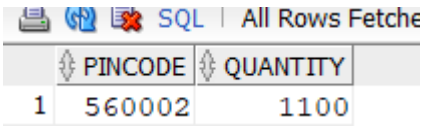
Output:

	BLOCK_ID	COUNT
1	B103	3

Q5) Finding out the location where the majority of manufactured mobile phones are sold.

```
select pincode,quantity from
 (select pincode,sum(quantity) as quantity from Sales natural join Shop group by pincode)
 where quantity=(select min(quantity) from
 (select sum(Quantity) as quantity from Sales natural join Shop group by pincode));
```

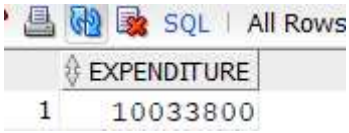
Output:



Q6) Finding the expenditure of the company.

```
select
sum(Electric_charge+Trasport_charge+Maintenance*200+Raw_material)
 as Expenditure from Expenses;
```

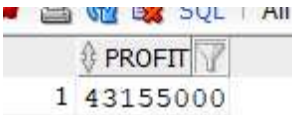
Output:



Q7) Finding the profit of the company.

```
select sum(quantity)-sum(Electric_charge) as profit from
 (select mobile_id,quantity*cost as quantity from sales natural join Features),
 (select
sum(ELectric_charge+Trasport_charge+Maintenance*200+Raw_material) as
Electric_charge from Expenses);
```

Output:



Q8) Finding out the employee details of the block having less sales.

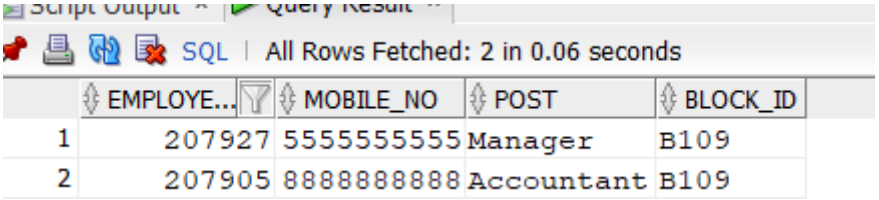
```
select distinct Employee_id,Mobile_No,Post,Block_id from Employee natural
join Block natural join Sales where

 Block_id=(select block_id from (select Block_id,sum(Quantity) as quantity
from

 Sales group by Block_id) where quantity= (select min(Quantity) from

 (select Block_id,sum(Quantity)as quantity from Sales group by Block_id)));
```

Output:



	EMPLOYE...	MOBILE_NO	POST	BLOCK_ID
1	207927	5555555555	Manager	B109
2	207905	8888888888	Accountant	B109