THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

USC310

DBMS Mini Project

Mobile Database System

Submitted to: Ms Radhika Bansal

Submitted by:

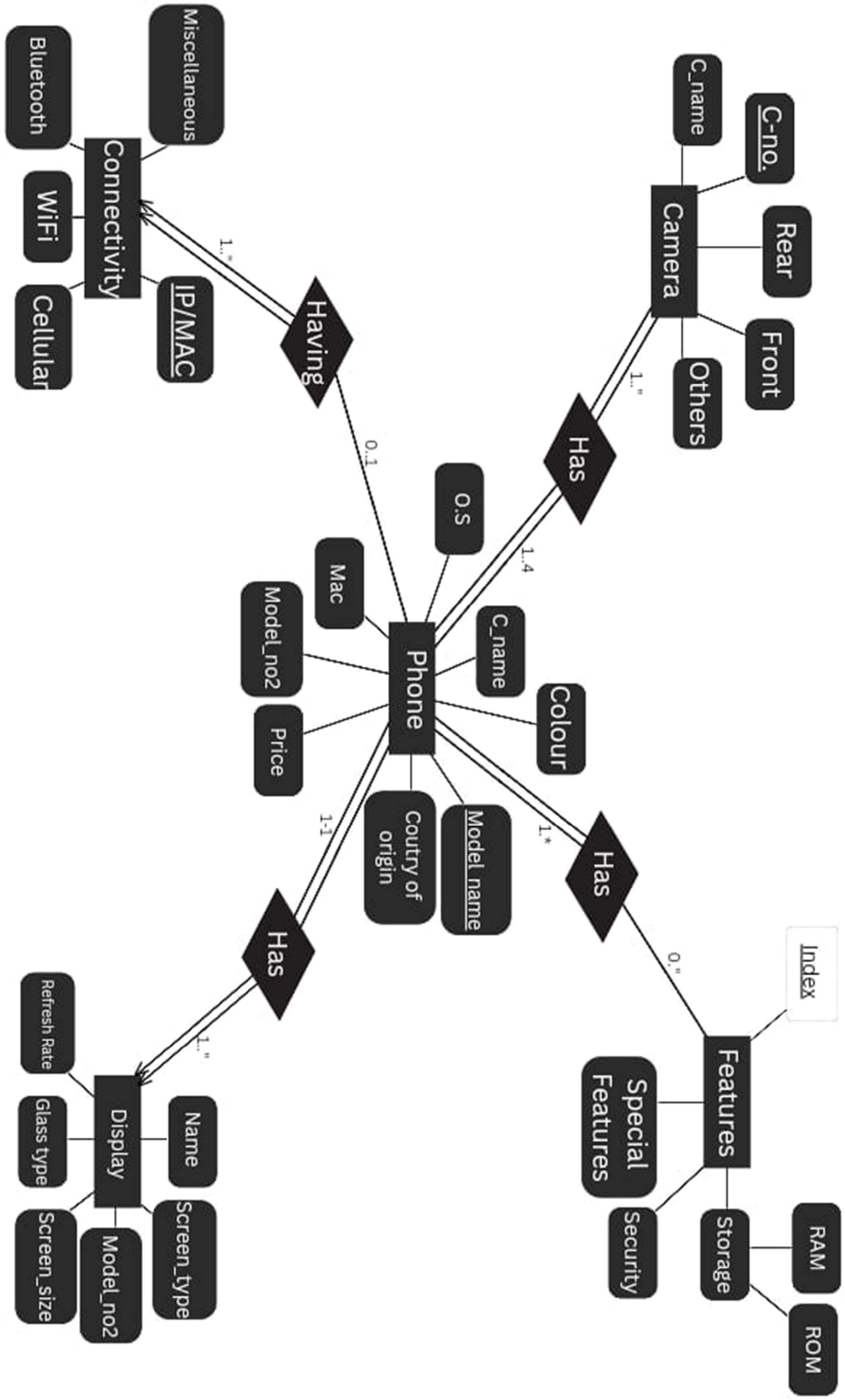
Aditi Sharma	**102103369**
Khushi Mittal	**102103357**
Susnigdha Kheda	**102103219**
Sneha Tuli	**102103349**

INDEX

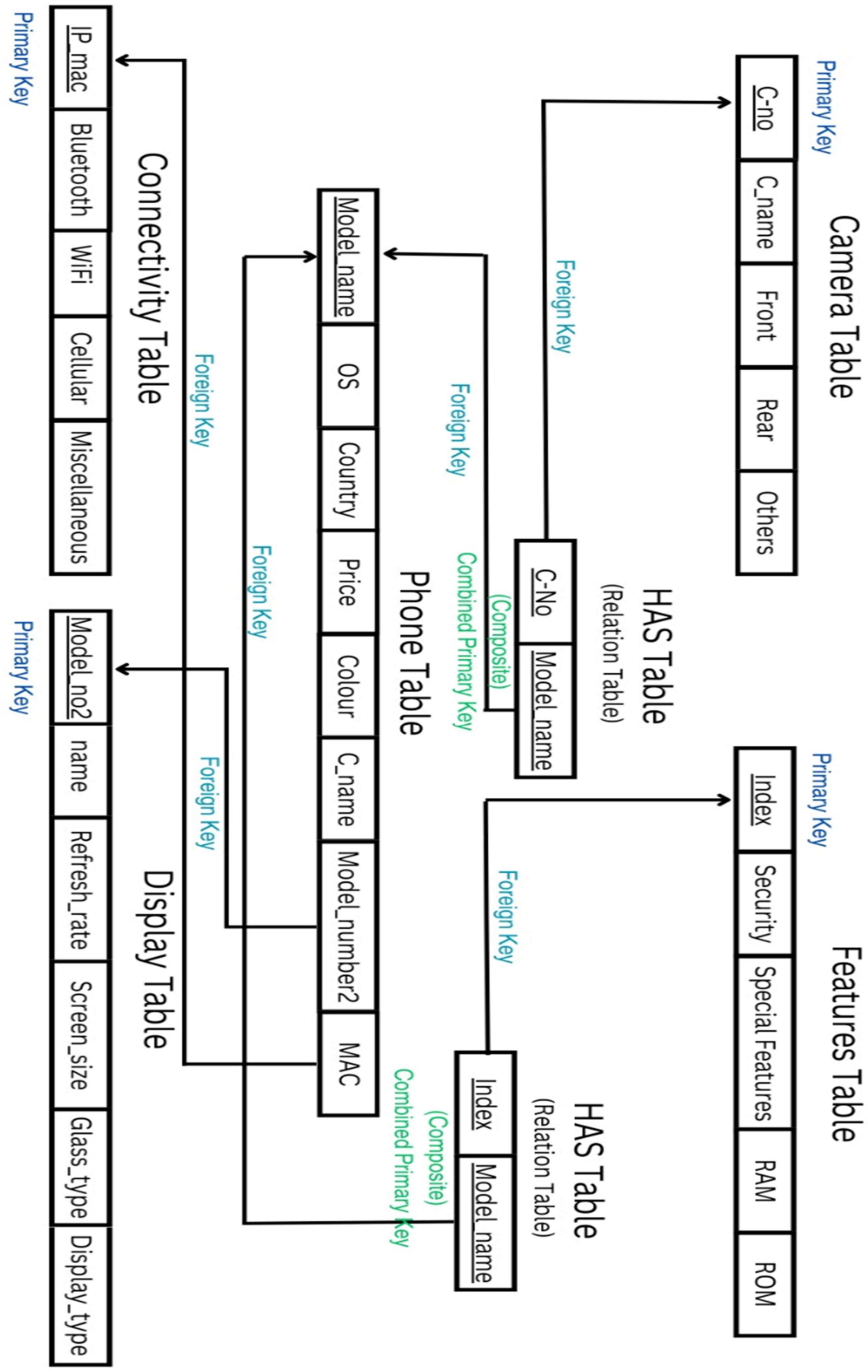
INTRODUCTION

We designed this project to help customers to compare and decide on the best mobile phone that meets their requirements. A phone contains many components. The features a user needs to consider depending on their requirements. There are some features say like charging speed and RAM which even layman users consider while buying a phone and some advanced and technical features like Wi-Fi and Bluetooth version numbers that only tech-savvy users consider. We have taken care of both users. The activity of comparing phones has always been manual/memory based where we write/ memorize the features of the phone to compare it to others. This is obviously tiresome and has room for errors. We have digitized it using our database. We hope that our project will help customers find the best phone for themselves per their needs and requirements.

ER-DIAGRAM


Connectivity
Miscellaneous
Bluetooth
WiFi
Cellular
IP/MAC
1.."
Having
0..1
Camera
C_name
C-no.
Rear
Front
Others
1.."
Has
1..4
Phone
O.S
Mac
Model_no2
Price
C_name
Colour
Coutry of origin
Model_name
1-1
Has
1.."
Display
Refresh Rate
Glass type
Screen_size
Model_no2
Screen_type
Name
1.*
Has
0.."
Features
Index
Special Features
Security
Storage
RAM
ROM

ER-TO-TABLES


Camera Table
Primary Key
C-no
C_name
Front
Rear
Others
Foreign Key
HAS Table
(Relation Table)
C-No
Model_name
(Composite)
Combined Primary Key
Foreign Key
Features Table
Primary Key
Index
Security
Special Features
RAM
ROM
Foreign Key
HAS Table
(Relation Table)
Index
Model_name
(Composite)
Combined Primary Key
Phone Table
Model_name
OS
Country
Price
Colour
C_name
Model_number2
MAC
Foreign Key
Foreign Key
Foreign Key
Connectivity Table
Primary Key
IP_mac
Bluetooth
WiFi
Cellular
Miscellaneous
Display Table
Primary Key
Model_no2
name
Refresh_rate
Screen_size
Glass_type
Display_type

NORMALISATION

Table no. 1-

DISPLAY_TABLE BEFORE NORMALIZATION— Display_table(name varchar(10),model_no2 number(10),display_type varchar(10), glass_type varchar(10),screen_size varchar(10),refresh_rate varchar(20)));

Here , the table is in 2nd normal form , but we observed a transitive FD , that is a non prime attribute display_type in determining a non – prime attribute glass_type so this table is not in 3rd normal form/BCNF ….

AFTER NORMALIZATION—

TABLE-1 Display_table (Name varchar(30), model_no2 number(2) primary key,display_type varchar(25) references display2 ,screen_size varchar(10),refresh_rate varchar(10))

TABLE-2 display2(Display_type VARCHAR(20) PRIMARY KEY, Glass_type VARCHAR(55))

Now the two tables are in BCNF…

Table no. 2-

connectivity(IP_MAC VARCHAR(12) PRIMARY KEY, Bluetooth VARCHAR(4), wifi VARCHAR(50), cellular VARCHAR(10), miscellaneous VARCHAR(20))

Here, the table is already normalized, There are no partial FDs, and no transitive fd,s Only candidate keys are determinant, So it is already in BCNF form…

Table no. 3-

features(ind INT PRIMARY KEY, sec VARCHAR(15), special_features VARCHAR(25), RAM VARCHAR(10), ROM VARCHAR(10))

This table is already BCNF normalized because only the candidate key (here-primary key) is the only determinant.

Table no. 4-

cameras(C_NO INT PRIMARY KEY, Front INT, REAR INT, OTHER VARCHAR(15), C_NAME VARCHAR(15))

This table is already BCNF normalized because only the candidate key (here-primary key) is the only determinant.

Table no. 5-

relation1(c_no number(3) REFERENCES cameras(c_no), model_name varchar(20) REFERENCES phone_table(model_name), primary key(c_no,model_name))

Since only two attributes are there in this table so it is BCNF normalised, because for this form at least 3 attributes are required.

Table no. 6-

relation2(idx int REFERENCES features(ind), model_name varchar(20) REFERENCES phone_table(model_name), primary key(idx,model_name))

Since only two attributes are there in this table so it is BCNF normalised because for this form at least 3 attributes are required.

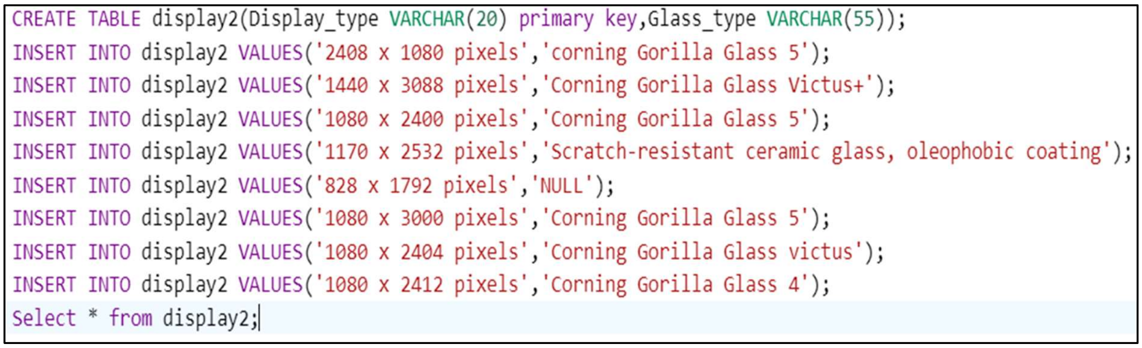
Table no. 7-

phone_table(model_name varchar(20) primary key, OS varchar(30), country varchar(10), price number(10), color char(10), c_name char(10), model_number2 number(3) REFERENCES display_table(model_no2), mac varchar(20) REFERENCES connectivity(ip_mac))

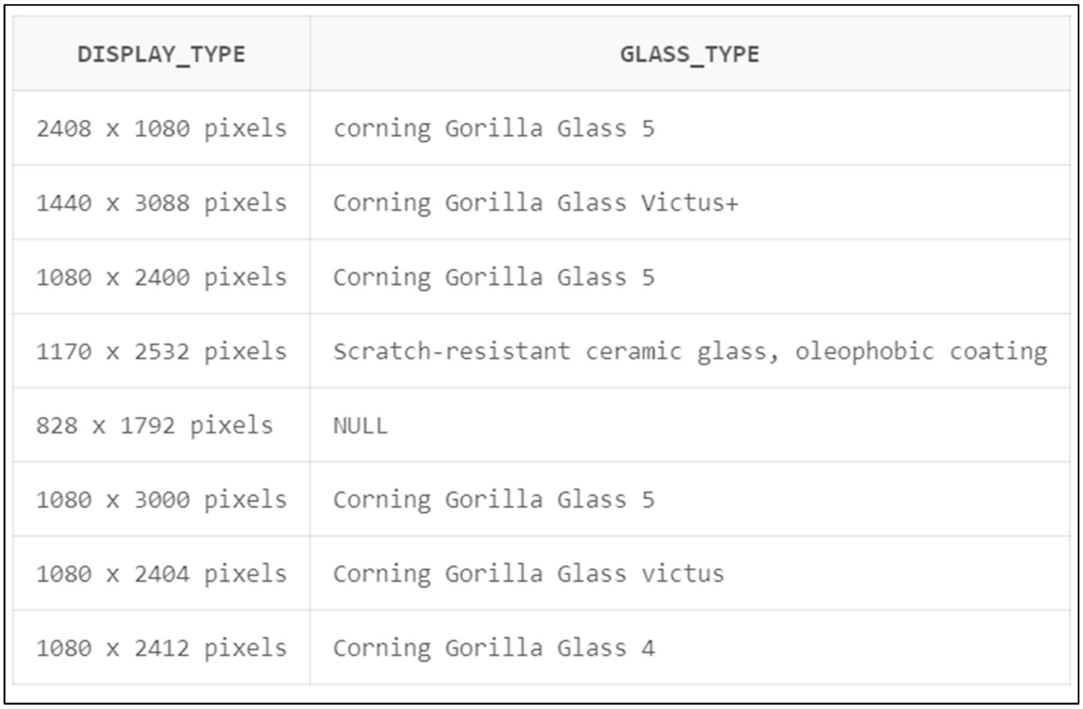
It is normalised till BCNF because only candidate keys are the determinants.

SQL/PLSQL

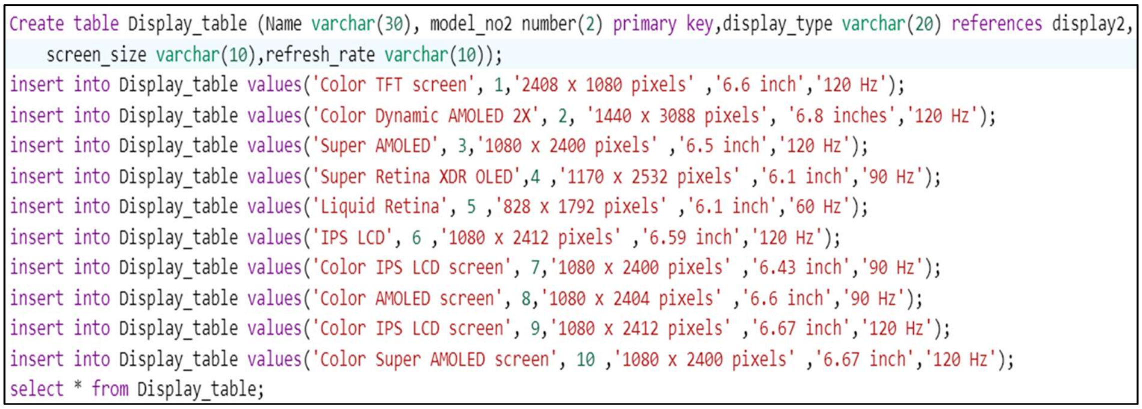
Creating tables and inserting values into tables:



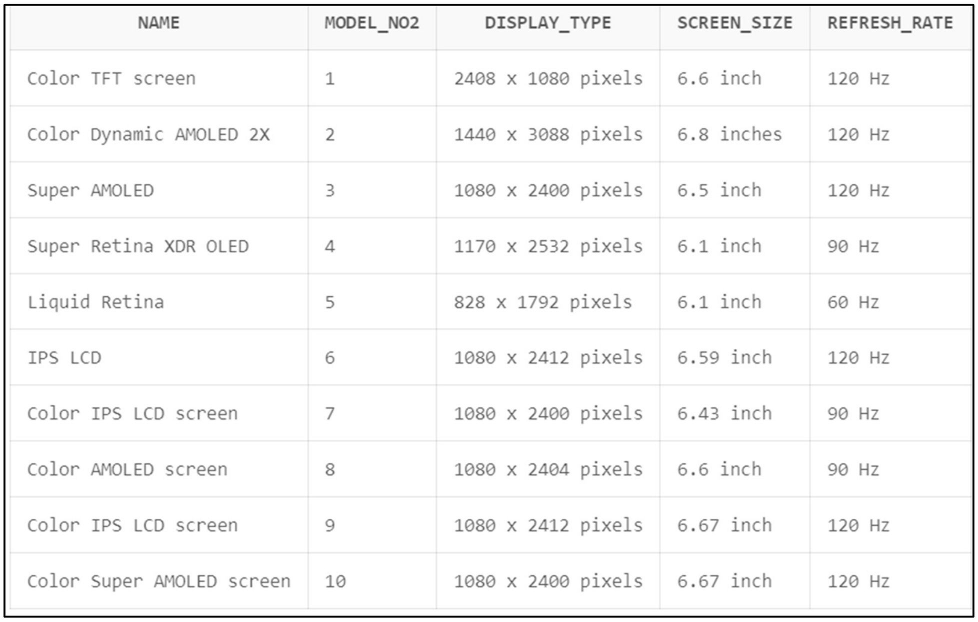
```
CREATE TABLE display2(Display_type VARCHAR(20) primary key,Glass_type VARCHAR(55));
INSERT INTO display2 VALUES('2408 x 1080 pixels','corning Gorilla Glass 5');
INSERT INTO display2 VALUES('1440 x 3088 pixels','Corning Gorilla Glass Victus+');
INSERT INTO display2 VALUES('1080 x 2400 pixels','Corning Gorilla Glass 5');
INSERT INTO display2 VALUES('1170 x 2532 pixels','Scratch-resistant ceramic glass, oleophobic coating');
INSERT INTO display2 VALUES('828 x 1792 pixels','NULL');
INSERT INTO display2 VALUES('1080 x 3000 pixels','Corning Gorilla Glass 5');
INSERT INTO display2 VALUES('1080 x 2404 pixels','Corning Gorilla Glass victus');
INSERT INTO display2 VALUES('1080 x 2412 pixels','Corning Gorilla Glass 4');
Select * from display2;
```



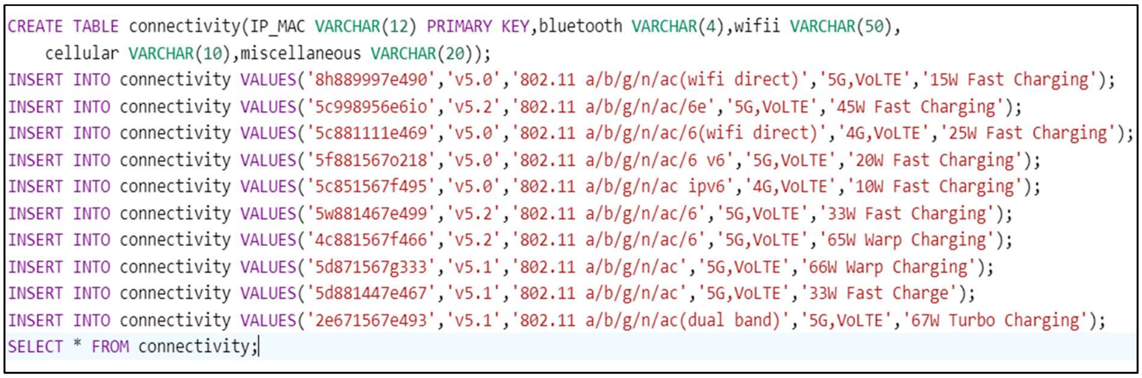
DISPLAY_TYPE	GLASS_TYPE
2408 x 1080 pixels	corning Gorilla Glass 5
1440 x 3088 pixels	Corning Gorilla Glass Victus+
1080 x 2400 pixels	Corning Gorilla Glass 5
1170 x 2532 pixels	Scratch-resistant ceramic glass, oleophobic coating
828 x 1792 pixels	NULL
1080 x 3000 pixels	Corning Gorilla Glass 5
1080 x 2404 pixels	Corning Gorilla Glass victus
1080 x 2412 pixels	Corning Gorilla Glass 4



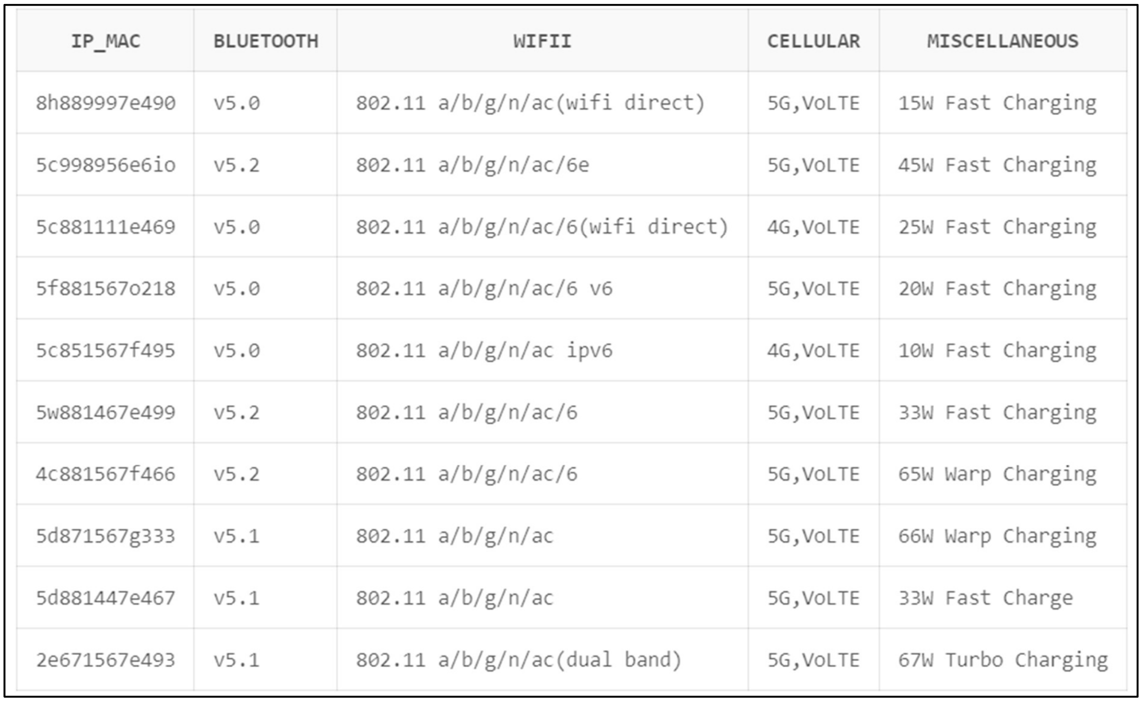
```
Create table Display_table (Name varchar(30), model_no2 number(2) primary key,display_type varchar(20) references display2,
    screen_size varchar(10),refresh_rate varchar(10));
insert into Display_table values('Color TFT screen', 1,'2408 x 1080 pixels' ,'6.6 inch','120 Hz');
insert into Display_table values('Color Dynamic AMOLED 2X', 2, '1440 x 3088 pixels', '6.8 inches','120 Hz');
insert into Display_table values('Super AMOLED', 3,'1080 x 2400 pixels' ,'6.5 inch','120 Hz');
insert into Display_table values('Super Retina XDR OLED',4 ,'1170 x 2532 pixels' ,'6.1 inch','90 Hz');
insert into Display_table values('Liquid Retina', 5 ,'828 x 1792 pixels' ,'6.1 inch','60 Hz');
insert into Display_table values('IPS LCD', 6 ,'1080 x 2412 pixels' ,'6.59 inch','120 Hz');
insert into Display_table values('Color IPS LCD screen', 7,'1080 x 2400 pixels' ,'6.43 inch','90 Hz');
insert into Display_table values('Color AMOLED screen', 8,'1080 x 2404 pixels' ,'6.6 inch','90 Hz');
insert into Display_table values('Color IPS LCD screen', 9,'1080 x 2412 pixels' ,'6.67 inch','120 Hz');
insert into Display_table values('Color Super AMOLED screen', 10 ,'1080 x 2400 pixels' ,'6.67 inch','120 Hz');
select * from Display_table;
```



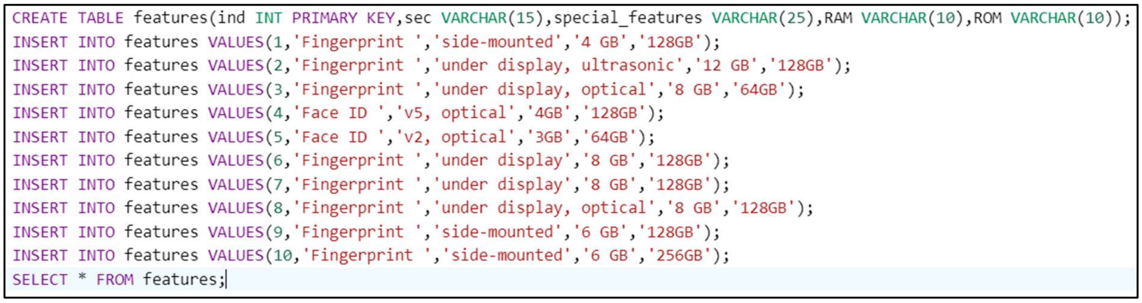
NAME	MODEL_NO2	DISPLAY_TYPE	SCREEN_SIZE	REFRESH_RATE
Color TFT screen	1	2408 x 1080 pixels	6.6 inch	120 Hz
Color Dynamic AMOLED 2X	2	1440 x 3088 pixels	6.8 inches	120 Hz
Super AMOLED	3	1080 x 2400 pixels	6.5 inch	120 Hz
Super Retina XDR OLED	4	1170 x 2532 pixels	6.1 inch	90 Hz
Liquid Retina	5	828 x 1792 pixels	6.1 inch	60 Hz
IPS LCD	6	1080 x 2412 pixels	6.59 inch	120 Hz
Color IPS LCD screen	7	1080 x 2400 pixels	6.43 inch	90 Hz
Color AMOLED screen	8	1080 x 2404 pixels	6.6 inch	90 Hz
Color IPS LCD screen	9	1080 x 2412 pixels	6.67 inch	120 Hz
Color Super AMOLED screen	10	1080 x 2400 pixels	6.67 inch	120 Hz



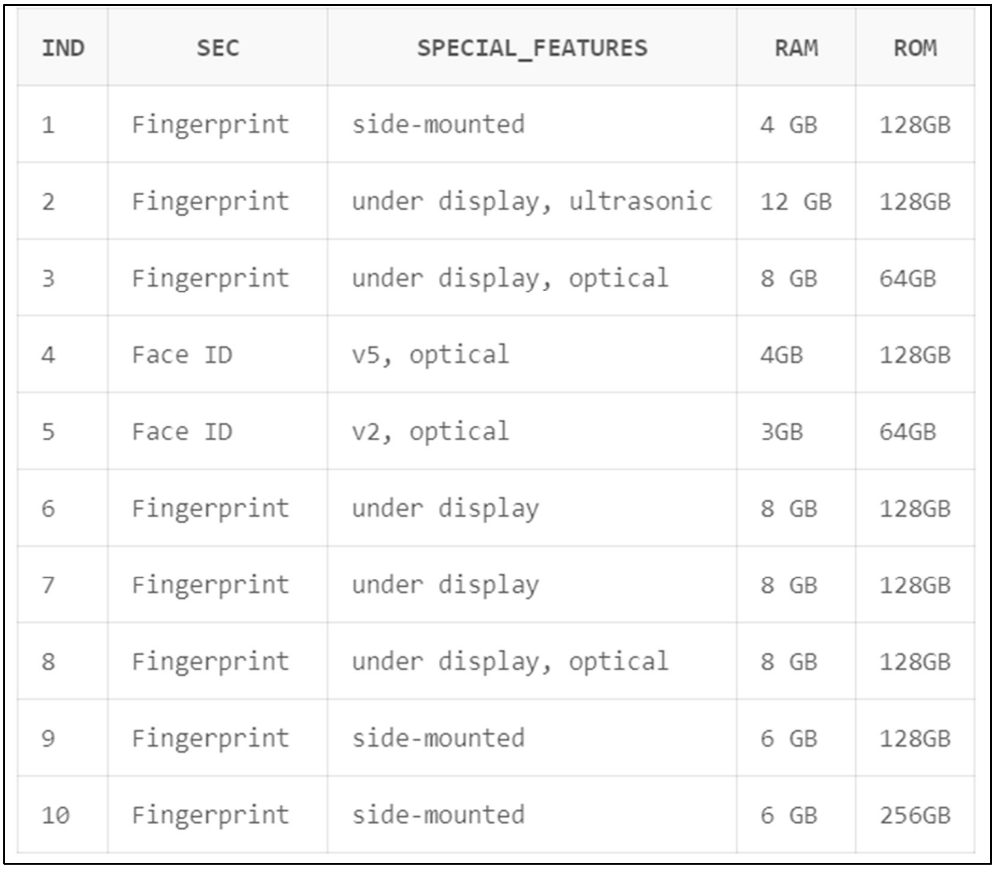
```
CREATE TABLE connectivity(IP_MAC VARCHAR(12) PRIMARY KEY,bluetooth VARCHAR(4),wifii VARCHAR(50),
    cellular VARCHAR(10),miscellaneous VARCHAR(20));
INSERT INTO connectivity VALUES('8h889997e490','v5.0','802.11 a/b/g/n/ac(wifi direct)','5G,VoLTE','15W Fast Charging');
INSERT INTO connectivity VALUES('5c998956e6io','v5.2','802.11 a/b/g/n/ac/6e','5G,VoLTE','45W Fast Charging');
INSERT INTO connectivity VALUES('5c881111e469','v5.0','802.11 a/b/g/n/ac/6(wifi direct)','4G,VoLTE','25W Fast Charging');
INSERT INTO connectivity VALUES('5f881567o218','v5.0','802.11 a/b/g/n/ac/6 v6','5G,VoLTE','20W Fast Charging');
INSERT INTO connectivity VALUES('5c851567f495','v5.0','802.11 a/b/g/n/ac ipv6','4G,VoLTE','10W Fast Charging');
INSERT INTO connectivity VALUES('5w881467e499','v5.2','802.11 a/b/g/n/ac/6','5G,VoLTE','33W Fast Charging');
INSERT INTO connectivity VALUES('4c881567f466','v5.2','802.11 a/b/g/n/ac/6','5G,VoLTE','65W Warp Charging');
INSERT INTO connectivity VALUES('5d871567g333','v5.1','802.11 a/b/g/n/ac','5G,VoLTE','66W Warp Charging');
INSERT INTO connectivity VALUES('5d881447e467','v5.1','802.11 a/b/g/n/ac','5G,VoLTE','33W Fast Charge');
INSERT INTO connectivity VALUES('2e671567e493','v5.1','802.11 a/b/g/n/ac(dual band)','5G,VoLTE','67W Turbo Charging');
SELECT * FROM connectivity;
```



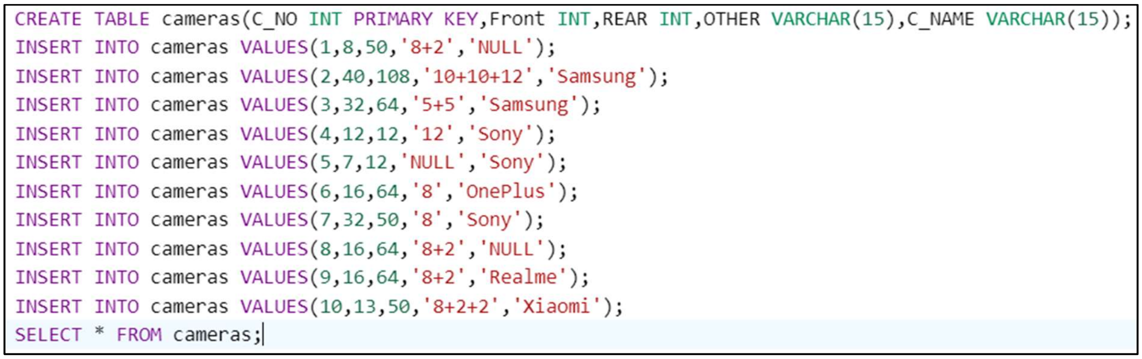
IP_MAC	BLUETOOTH	WIFII	CELLULAR	MISCELLANEOUS
8h889997e490	v5.0	802.11 a/b/g/n/ac(wifi direct)	5G,VoLTE	15W Fast Charging
5c998956e6io	v5.2	802.11 a/b/g/n/ac/6e	5G,VoLTE	45W Fast Charging
5c881111e469	v5.0	802.11 a/b/g/n/ac/6(wifi direct)	4G,VoLTE	25W Fast Charging
5f881567o218	v5.0	802.11 a/b/g/n/ac/6 v6	5G,VoLTE	20W Fast Charging
5c851567f495	v5.0	802.11 a/b/g/n/ac ipv6	4G,VoLTE	10W Fast Charging
5w881467e499	v5.2	802.11 a/b/g/n/ac/6	5G,VoLTE	33W Fast Charging
4c881567f466	v5.2	802.11 a/b/g/n/ac/6	5G,VoLTE	65W Warp Charging
5d871567g333	v5.1	802.11 a/b/g/n/ac	5G,VoLTE	66W Warp Charging
5d881447e467	v5.1	802.11 a/b/g/n/ac	5G,VoLTE	33W Fast Charge
2e671567e493	v5.1	802.11 a/b/g/n/ac(dual band)	5G,VoLTE	67W Turbo Charging



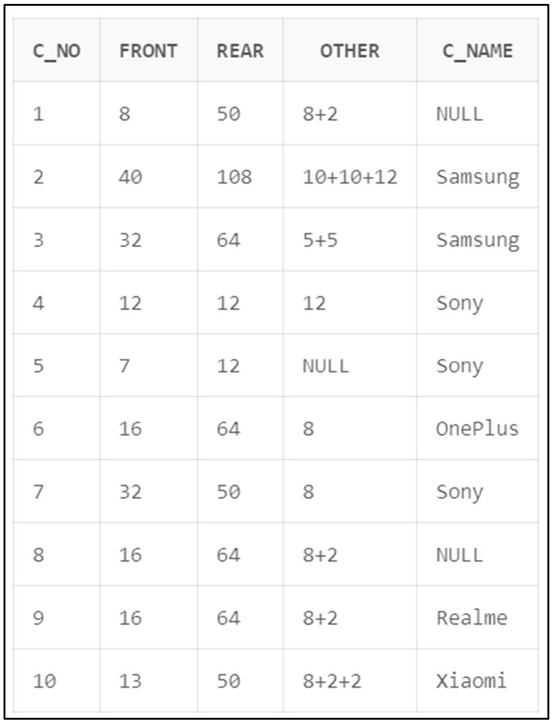
```
CREATE TABLE features(ind INT PRIMARY KEY,sec VARCHAR(15),special_features VARCHAR(25),RAM VARCHAR(10),ROM VARCHAR(10));
INSERT INTO features VALUES(1,'Fingerprint ','side-mounted','4 GB','128GB');
INSERT INTO features VALUES(2,'Fingerprint ','under display, ultrasonic','12 GB','128GB');
INSERT INTO features VALUES(3,'Fingerprint ','under display, optical','8 GB','64GB');
INSERT INTO features VALUES(4,'Face ID ','v5, optical','4GB','128GB');
INSERT INTO features VALUES(5,'Face ID ','v2, optical','3GB','64GB');
INSERT INTO features VALUES(6,'Fingerprint ','under display','8 GB','128GB');
INSERT INTO features VALUES(7,'Fingerprint ','under display','8 GB','128GB');
INSERT INTO features VALUES(8,'Fingerprint ','under display, optical','8 GB','128GB');
INSERT INTO features VALUES(9,'Fingerprint ','side-mounted','6 GB','128GB');
INSERT INTO features VALUES(10,'Fingerprint ','side-mounted','6 GB','256GB');
SELECT * FROM features;
```



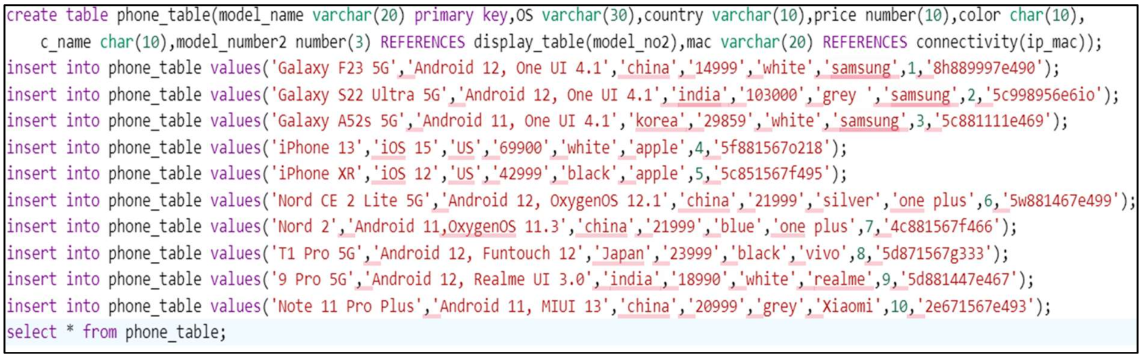
IND	SEC	SPECIAL_FEATURES	RAM	ROM
1	Fingerprint	side-mounted	4 GB	128GB
2	Fingerprint	under display, ultrasonic	12 GB	128GB
3	Fingerprint	under display, optical	8 GB	64GB
4	Face ID	v5, optical	4GB	128GB
5	Face ID	v2, optical	3GB	64GB
6	Fingerprint	under display	8 GB	128GB
7	Fingerprint	under display	8 GB	128GB
8	Fingerprint	under display, optical	8 GB	128GB
9	Fingerprint	side-mounted	6 GB	128GB
10	Fingerprint	side-mounted	6 GB	256GB



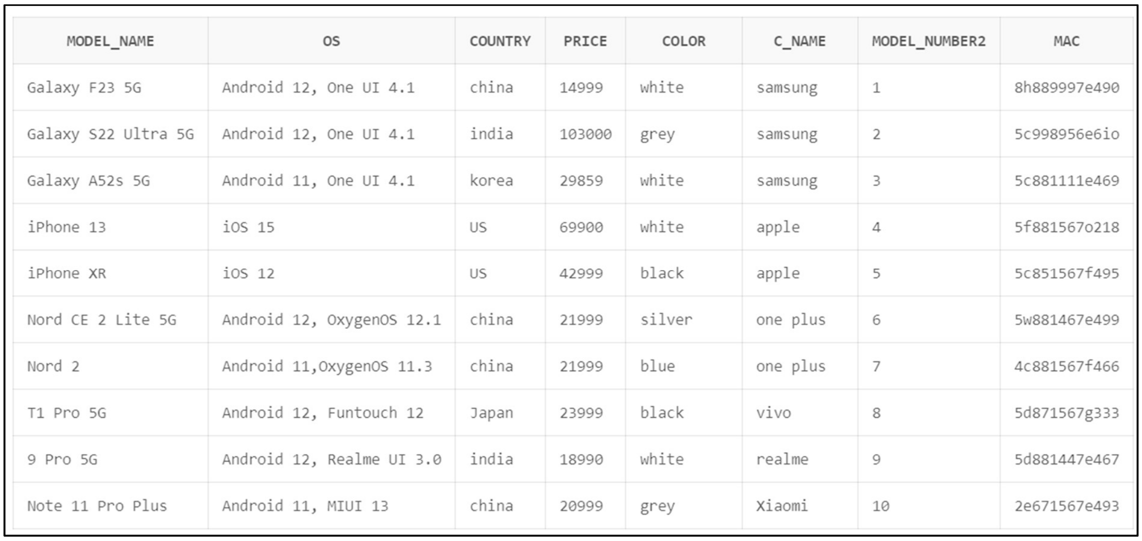
```
CREATE TABLE cameras(C_NO INT PRIMARY KEY,Front INT,REAR INT,OTHER VARCHAR(15),C_NAME VARCHAR(15));
INSERT INTO cameras VALUES(1,8,50,'8+2','NULL');
INSERT INTO cameras VALUES(2,40,108,'10+10+12','Samsung');
INSERT INTO cameras VALUES(3,32,64,'5+5','Samsung');
INSERT INTO cameras VALUES(4,12,12,'12','Sony');
INSERT INTO cameras VALUES(5,7,12,'NULL','Sony');
INSERT INTO cameras VALUES(6,16,64,'8','OnePlus');
INSERT INTO cameras VALUES(7,32,50,'8','Sony');
INSERT INTO cameras VALUES(8,16,64,'8+2','NULL');
INSERT INTO cameras VALUES(9,16,64,'8+2','Realme');
INSERT INTO cameras VALUES(10,13,50,'8+2+2','Xiaomi');
SELECT * FROM cameras;
```



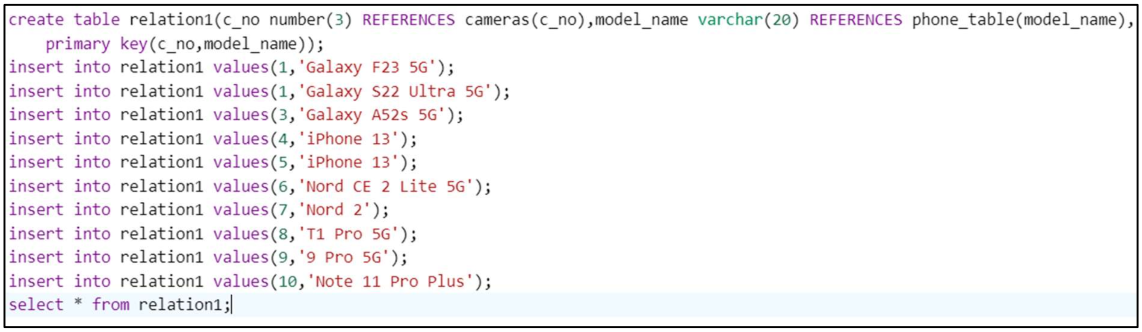
C_NO	FRONT	REAR	OTHER	C_NAME
1	8	50	8+2	NULL
2	40	108	10+10+12	Samsung
3	32	64	5+5	Samsung
4	12	12	12	Sony
5	7	12	NULL	Sony
6	16	64	8	OnePlus
7	32	50	8	Sony
8	16	64	8+2	NULL
9	16	64	8+2	Realme
10	13	50	8+2+2	Xiaomi



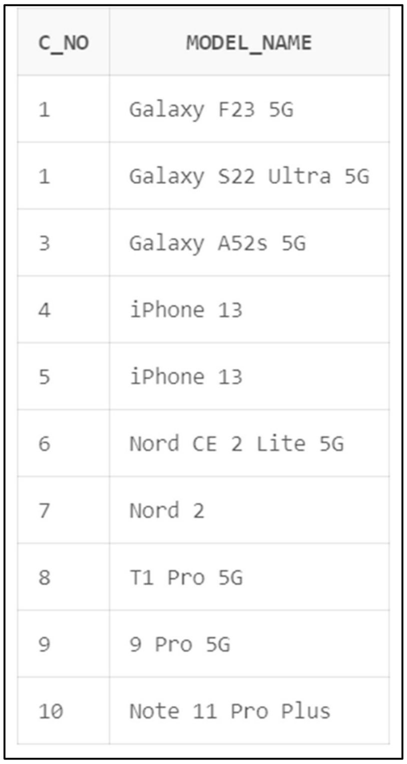
```
create table phone_table(model_name varchar(20) primary key,OS varchar(30),country varchar(10),price number(10),color char(10),
    c_name char(10),model_number2 number(3) REFERENCES display_table(model_no2),mac varchar(20) REFERENCES connectivity(ip_mac));
insert into phone_table values('Galaxy F23 5G','Android 12, One UI 4.1','china','14999','white','samsung',1,'8h889997e490');
insert into phone_table values('Galaxy S22 Ultra 5G','Android 12, One UI 4.1','india','103000','grey ','samsung',2,'5c998956e6io');
insert into phone_table values('Galaxy A52s 5G','Android 11, One UI 4.1','korea','29859','white','samsung',3,'5c881111e469');
insert into phone_table values('iPhone 13','iOS 15','US','69900','white','apple',4,'5f881567o218');
insert into phone_table values('iPhone XR','iOS 12','US','42999','black','apple',5,'5c851567f495');
insert into phone_table values('Nord CE 2 Lite 5G','Android 12, OxygenOS 12.1','china','21999','silver','one plus',6,'5w881467e499');
insert into phone_table values('Nord 2','Android 11,OxygenOS 11.3','china','21999','blue','one plus',7,'4c881567f466');
insert into phone_table values('T1 Pro 5G','Android 12, Funtouch 12','Japan','23999','black','vivo',8,'5d871567g333');
insert into phone_table values('9 Pro 5G','Android 12, Realme UI 3.0','india','18990','white','realme',9,'5d881447e467');
insert into phone_table values('Note 11 Pro Plus','Android 11, MIUI 13','china','20999','grey','Xiaomi',10,'2e671567e493');
select * from phone_table;
```



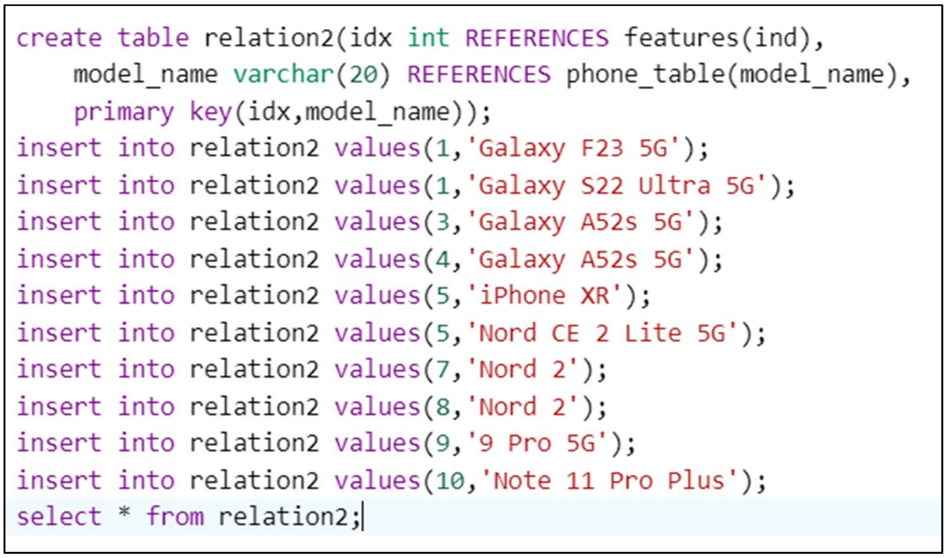
MODEL_NAME	OS	COUNTRY	PRICE	COLOR	C_NAME	MODEL_NUMBER2	MAC
Galaxy F23 5G	Android 12, One UI 4.1	china	14999	white	samsung	1	8h889997e490
Galaxy S22 Ultra 5G	Android 12, One UI 4.1	india	103000	grey	samsung	2	5c998956e6io
Galaxy A52s 5G	Android 11, One UI 4.1	korea	29859	white	samsung	3	5c881111e469
iPhone 13	iOS 15	US	69900	white	apple	4	5f881567o218
iPhone XR	iOS 12	US	42999	black	apple	5	5c851567f495
Nord CE 2 Lite 5G	Android 12, OxygenOS 12.1	china	21999	silver	one plus	6	5w881467e499
Nord 2	Android 11,OxygenOS 11.3	china	21999	blue	one plus	7	4c881567f466
T1 Pro 5G	Android 12, Funtouch 12	Japan	23999	black	vivo	8	5d871567g333
9 Pro 5G	Android 12, Realme UI 3.0	india	18990	white	realme	9	5d881447e467
Note 11 Pro Plus	Android 11, MIUI 13	china	20999	grey	Xiaomi	10	2e671567e493



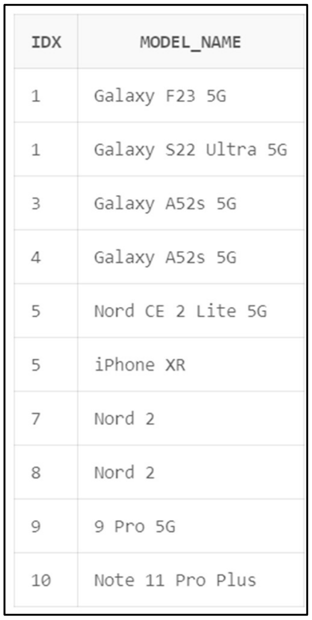
```
create table relation1(c_no number(3) REFERENCES cameras(c_no),model_name varchar(20) REFERENCES phone_table(model_name),
    primary key(c_no,model_name));
insert into relation1 values(1,'Galaxy F23 5G');
insert into relation1 values(1,'Galaxy S22 Ultra 5G');
insert into relation1 values(3,'Galaxy A52s 5G');
insert into relation1 values(4,'iPhone 13');
insert into relation1 values(5,'iPhone 13');
insert into relation1 values(6,'Nord CE 2 Lite 5G');
insert into relation1 values(7,'Nord 2');
insert into relation1 values(8,'T1 Pro 5G');
insert into relation1 values(9,'9 Pro 5G');
insert into relation1 values(10,'Note 11 Pro Plus');
select * from relation1;
```



C_NO	MODEL_NAME
1	Galaxy F23 5G
1	Galaxy S22 Ultra 5G
3	Galaxy A52s 5G
4	iPhone 13
5	iPhone 13
6	Nord CE 2 Lite 5G
7	Nord 2
8	T1 Pro 5G
9	9 Pro 5G
10	Note 11 Pro Plus



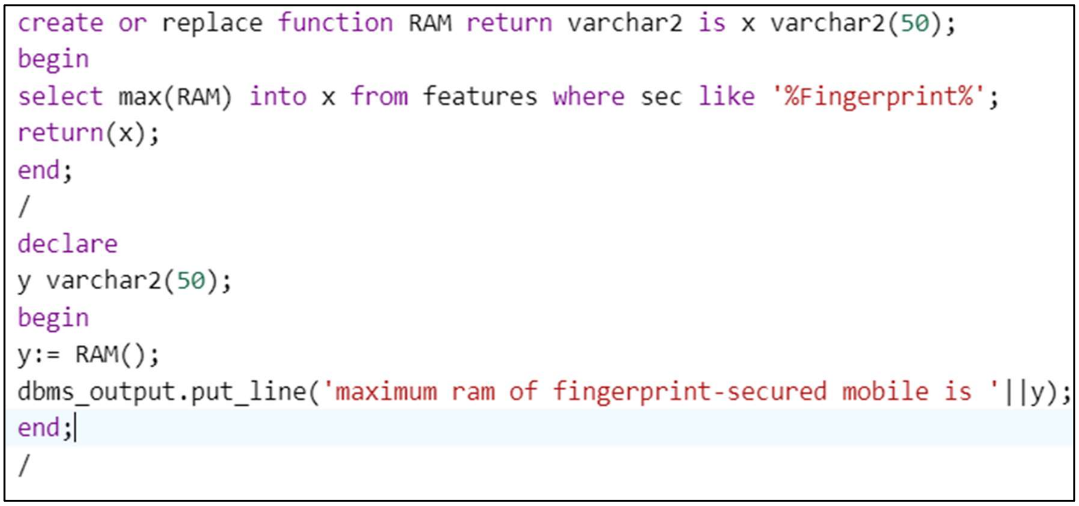
```
create table relation2(idx int REFERENCES features(ind),
    model_name varchar(20) REFERENCES phone_table(model_name),
    primary key(idx,model_name));
insert into relation2 values(1,'Galaxy F23 5G');
insert into relation2 values(1,'Galaxy S22 Ultra 5G');
insert into relation2 values(3,'Galaxy A52s 5G');
insert into relation2 values(4,'Galaxy A52s 5G');
insert into relation2 values(5,'iPhone XR');
insert into relation2 values(5,'Nord CE 2 Lite 5G');
insert into relation2 values(7,'Nord 2');
insert into relation2 values(8,'Nord 2');
insert into relation2 values(9,'9 Pro 5G');
insert into relation2 values(10,'Note 11 Pro Plus');
select * from relation2;
```



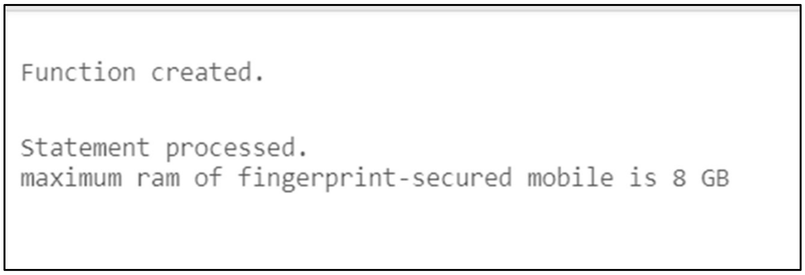
IDX	MODEL_NAME
1	Galaxy F23 5G
1	Galaxy S22 Ultra 5G
3	Galaxy A52s 5G
4	Galaxy A52s 5G
5	Nord CE 2 Lite 5G
5	iPhone XR
7	Nord 2
8	Nord 2
9	9 Pro 5G
10	Note 11 Pro Plus

Functions and procedures:

--Function to get a phone with fingerprint security with max RAM

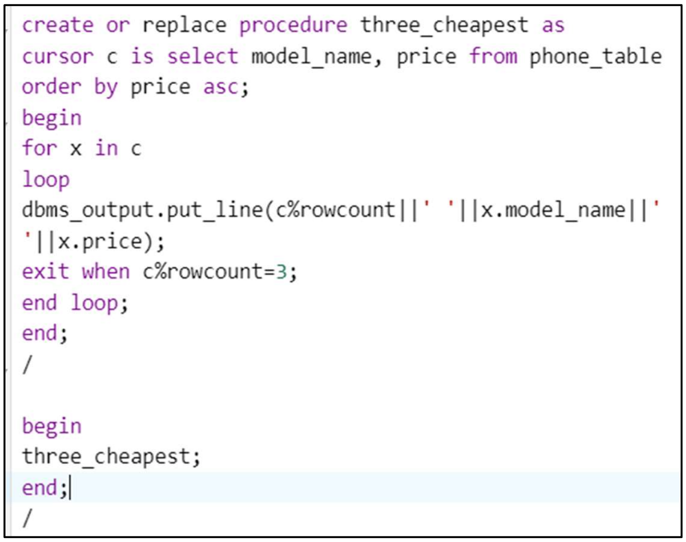


```
create or replace function RAM return varchar2 is x varchar2(50);
begin
select max(RAM) into x from features where sec like '%Fingerprint%';
return(x);
end;
/
declare
y varchar2(50);
begin
y:= RAM();
dbms_output.put_line('maximum ram of fingerprint-secured mobile is '||y);
end;
/
```

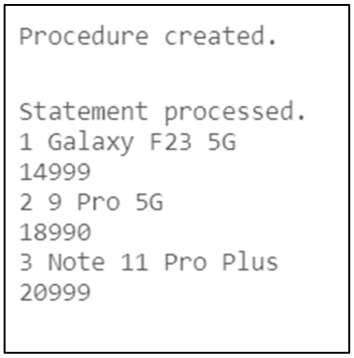


```
Function created.


Statement processed.
maximum ram of fingerprint-secured mobile is 8 GB
```

-- Procedure to find 3 cheapest phones (using cursors)

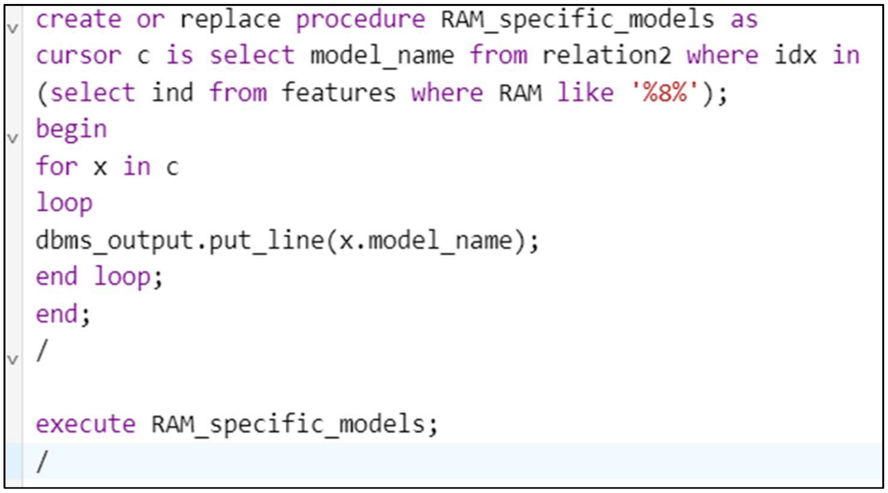


```
create or replace procedure three_cheapest as
cursor c is select model_name, price from phone_table
order by price asc;
begin
for x in c
loop
dbms_output.put_line(c%rowcount||' '||x.model_name||'
'||x.price);
exit when c%rowcount=3;
end loop;
end;
/

begin
three_cheapest;
end;
/
```

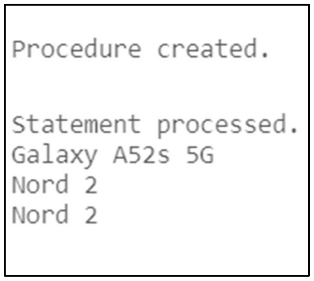


```
Procedure created.

Statement processed.
1 Galaxy F23 5G
14999
2 9 Pro 5G
18990
3 Note 11 Pro Plus
20999
```

-- Creating a procedure to get models with desired RAM(using cursors)

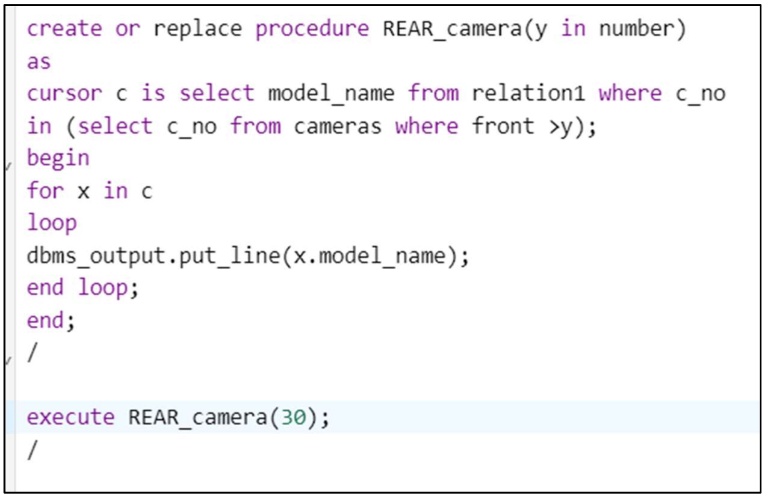


```
create or replace procedure RAM_specific_models as
cursor c is select model_name from relation2 where idx in
(select ind from features where RAM like '%8%');
begin
for x in c
loop
dbms_output.put_line(x.model_name);
end loop;
end;
/

execute RAM_specific_models;
/
```

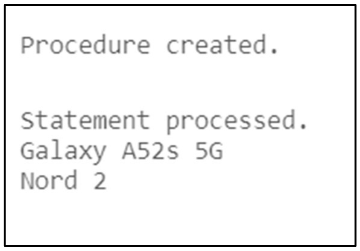


```
Procedure created.

Statement processed.
Galaxy A52s 5G
Nord 2
Nord 2
```

-- Procedure for displaying the data with specific rear camera details (using cursors)

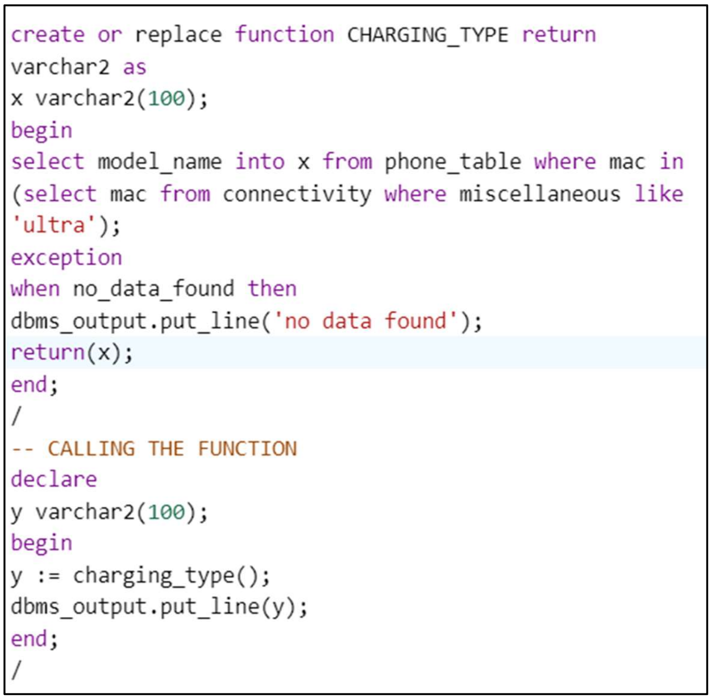


```
create or replace procedure REAR_camera(y in number)
as
cursor c is select model_name from relation1 where c_no
in (select c_no from cameras where front >y);
begin
for x in c
loop
dbms_output.put_line(x.model_name);
end loop;
end;
/

execute REAR_camera(30);
/
```

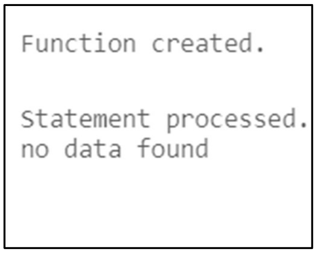


```
Procedure created.

Statement processed.
Galaxy A52s 5G
Nord 2
```

-- Creating function for getting phone of specific charging type

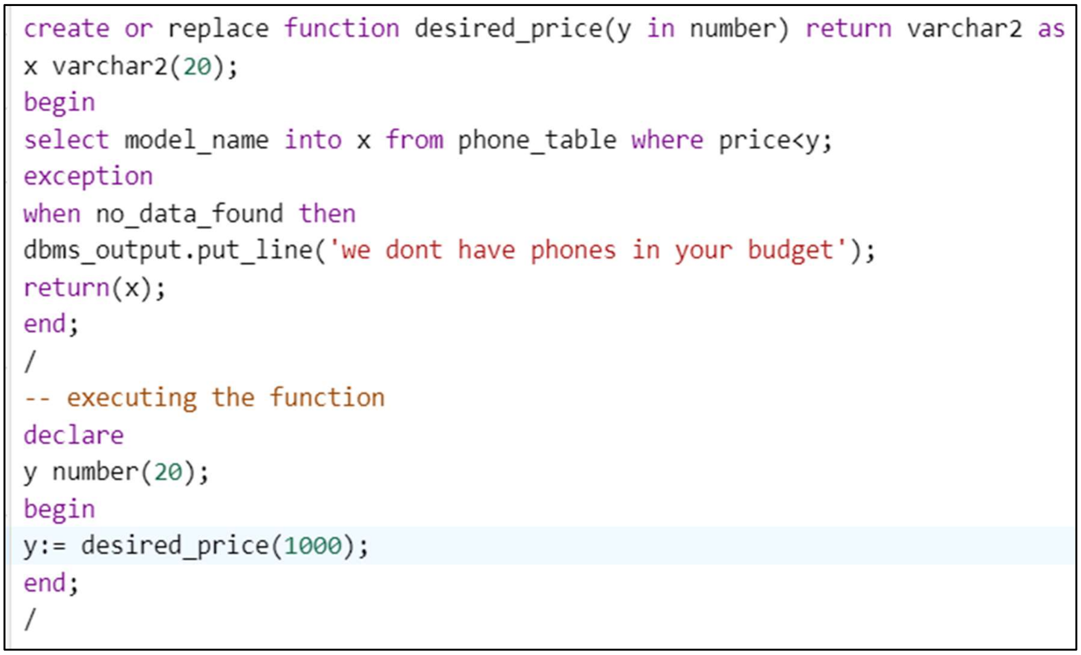


```
create or replace function CHARGING_TYPE return
varchar2 as
x varchar2(100);
begin
select model_name into x from phone_table where mac in
(select mac from connectivity where miscellaneous like
'ultra');
exception
when no_data_found then
dbms_output.put_line('no data found');
return(x);
end;
/
-- CALLING THE FUNCTION
declare
y varchar2(100);
begin
y := charging_type();
dbms_output.put_line(y);
end;
/
```

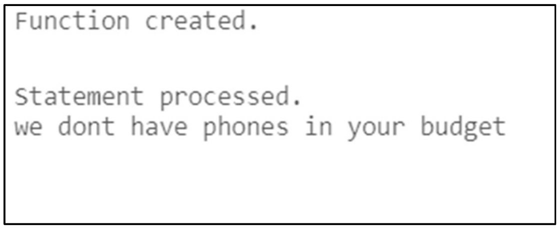


```
Function created.

Statement processed.
no data found
```

--Creating function which displays phones in our budget



```
create or replace function desired_price(y in number) return varchar2 as
x varchar2(20);
begin
select model_name into x from phone_table where price<y;
exception
when no_data_found then
dbms_output.put_line('we dont have phones in your budget');
return(x);
end;
/
-- executing the function
declare
y number(20);
begin
y:= desired_price(1000);
end;
/
```

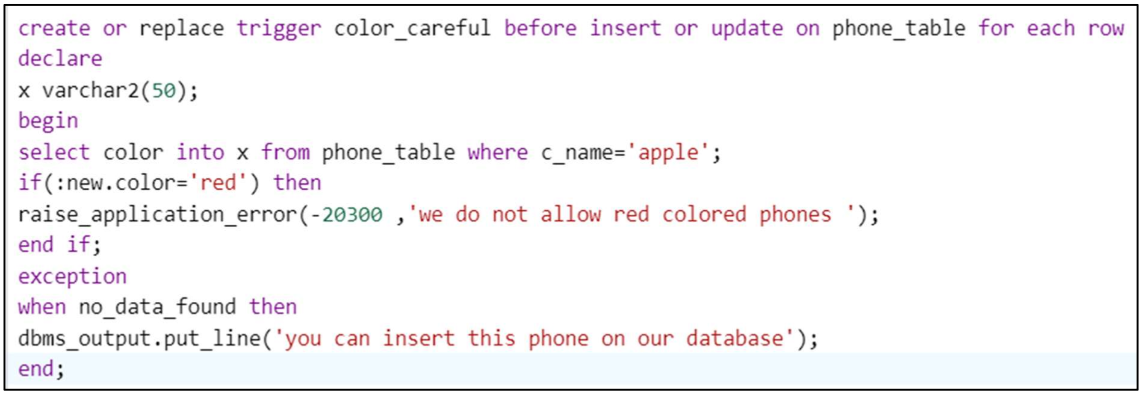


```
Function created.

Statement processed.
we dont have phones in your budget
```

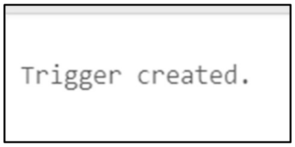
TRIGGERS

TRIGGER 1-

--Creating trigger which will not allow red-colored phones in our database because its sale is low



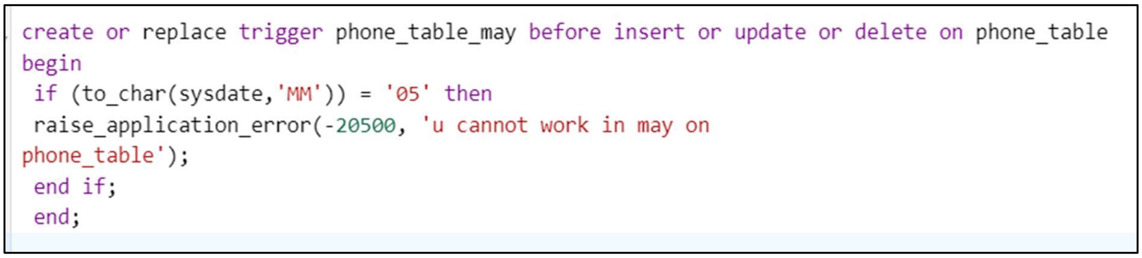
```
create or replace trigger color_careful before insert or update on phone_table for each row
declare
x varchar2(50);
begin
select color into x from phone_table where c_name='apple';
if(:new.color='red') then
raise_application_error(-20300 ,'we do not allow red colored phones ');
end if;
exception
when no_data_found then
dbms_output.put_line('you can insert this phone on our database');
end;
```



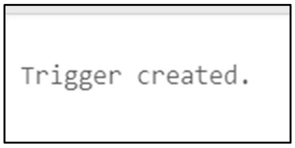
```
Trigger created.
```

TRIGGER 2-

--Creating trigger to ensure no one changes our table in the month of may



```
create or replace trigger phone_table_may before insert or update or delete on phone_table
begin
 if (to_char(sysdate,'MM')) = '05' then
 raise_application_error(-20500, 'u cannot work in may on
phone_table');
 end if;
 end;
```



```
Trigger created.
```

CONCLUSION

Our DBMS project is mainly targeted to help the user buy the desired featured phone. Nowadays phone is a necessity for everyone around and nevertheless, it's still difficult to choose a phone with desired requirements among the numerous options which may include basic characteristics like colour, camera to technical characteristics. Hence, this is an attempt to make it easy for the users while making a choice.

We used normalisation, PlSql with cursors and triggers included.

We aimed at all the characteristics possible and included the detailed ER diagram (Entity Relation) to represent the basis.

The main tables include connectivity, camera, display, features and phone along with the attributes that come under the particular category.