

DBMS - Mini Project

CAR Showroom Management System

Submitted By:KIRAN SS

SRN : PES1UG20CS212

V Semester Section D

Short Description and Scope of the Project

Car Showroom inventory management system helps to organize the data of available inventory and details of multiple branches of the showroom and cars with ease. It also enables to keep a detailed record of customer along with the car they have purchased

With the growing digitization, having a digital automated solution for data management is extremely important. It not only helps you manage your data without human interference but also connects with other modules of the management software to allow better organization of different tasks and functions of your business.

The database contains the following tables with attributes:

- **SHOWROOM**

- Name
- Showroom id
- State
- City
- Email id
- Phone number

- **CARS**

- Car id
- Make
- Model
- Year
- Price

- Showroom ID

- SPECS

- MAKEOFCAR

- MODELOFCAR

- YEAROFCAR

- TYPE

- DISP

- POWER

- FUELTYPE

- FUELCONS

- CUSTOMER

- CUSTID

- CUSTNAME

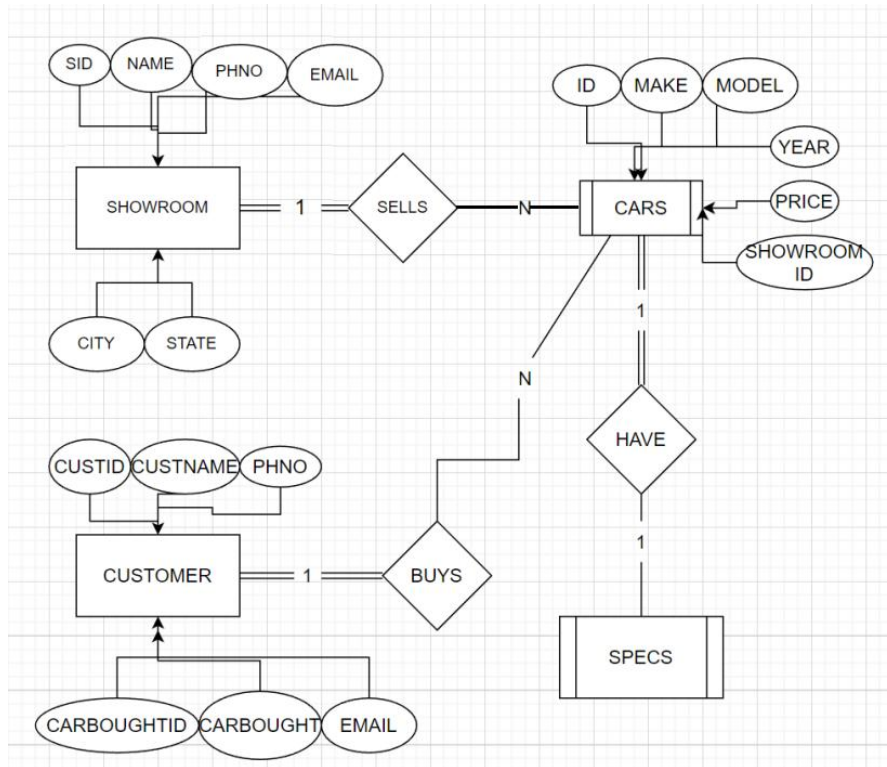
- EMAIL

- PHNO

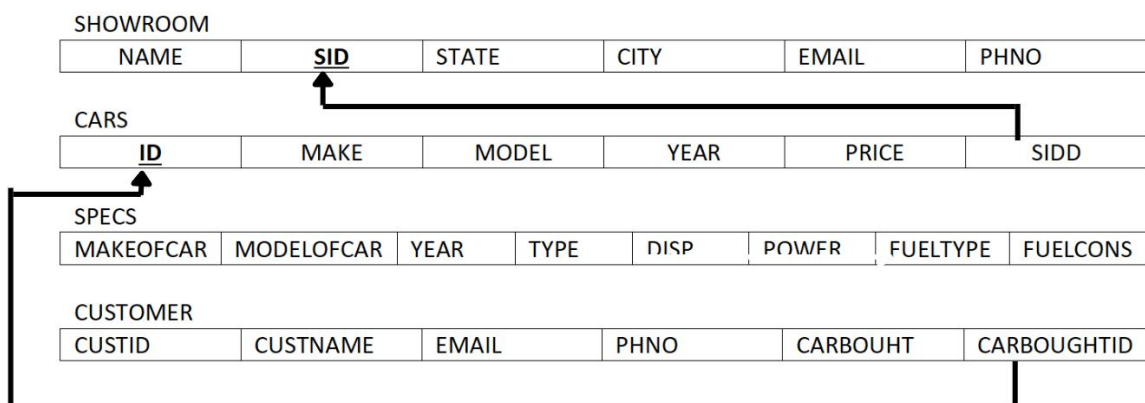
- CARBOUGHT

- CARBOUGHTID

ER Diagram



Relational Schema



DDL statements - Building the database

CREATING DATABASE:

1)

```
CREATE TABLE IF NOT EXISTS SHOWROOM (NAME  
varchar(30),SID varchar(13) PRIMARY KEY, STATE varchar(20),CITY  
varchar(20),EMAIL varchar(50),PHNO varchar(10))
```

2)

```
CREATE TABLE IF NOT EXISTS CARS (ID varchar(30) PRIMARY KEY,  
MAKE varchar(30), MODEL varchar(30),YEAR varchar(4),PRICE  
varchar(20),SIDD varchar(13),FOREIGN KEY(SIDD) REFERENCES  
SHOWROOM(SID))
```

3)

```
CREATE TABLE IF NOT EXISTS SPECS (MAKEOFCAR varchar(10),MODELOFCAR  
varchar(30),YEAROFCAR varchar(4),TYPE varchar(10),DISP  
varchar(20),POWER varchar(20),FUELTYPE varchar(20),FUELCONS  
varchar(20))
```

4)

```
CREATE TABLE IF NOT EXISTS CUSTOMER (CUSTID varchar(10),CUSTNAME  
varchar(30),EMAIL varchar(40),PHNO varchar(10),CARBOUGHT  
varchar(20),CARBOUGHTID varchar(20),FOREIGN KEY(CARBOUGHTID)  
REFERENCES CARS(ID))
```

Populating the Database

POPULATED the database using the streamlit frontend application

Join Queries

Showcase at least 4 join queries Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1) QUERY to display the PRICE AND DETAILS of each car in a showroom

```
SELECT CARS.PRICE,CARS.ID, MAKE,YEAR,TYPE,DISP,POWER,FUELTYPE,FUELCONS FR  
OM CARS RIGHT JOIN SPECS ON CARS.MODEL=SPECS.MODELOFCAR;
```

```
SELECT CARS.PRICE,CARS.ID, MAKE, YEAR, TYPE, DISP, POWER, FUELTYPE, FUELCONS FROM CARS RIGHT JOIN SPECS ON CARS.MODEL=SPECS.MODEL
```

☐ 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

PRICE	ID	MAKE	YEAR	TYPE	DISP	POWER	FUELTYPE	FUELCONS
23000000	C001	BMW	2021	GT	4000	600	PETROL	4
33000000	C002	BMW	2021	SEDAN	4000	650	PETROL	3
9500000	C003	VOLVO	2019	SUV	2500	320	DIESEL	5
32000000	C004	MERCEDES	2021	SUV	4000	560	PETROL	5
36000000	C005	PORSCHE	2022	SUERCAR	3800	650	PETROL	6
18000000	C006	PORSCHE	2019	SUV	3000	325	DIESEL	6
21000000	C007	TOYOTA	2019	SUV	3000	300	DIESEL	9
28000000	C008	BENTLEY	2018	SUV	4000	610	PETROL	4
8900000	C009	BENTLEY	2016	GT	4500	578	PETROL	4

2) QUERY TO DISPLAY NAME OF CUSTOMER WHO BOUGHT A CAR AND THE PRICE OF THE CAR

```
SELECT CUSTOMER.CUSTNAME,CUSTOMER.CUSTID, PRICE FROM CUSTOMER LEFT JOIN CARS ON CUSTOMER.CARBOUGHTID=cars.ID;
```

✓ Showing rows 0 - 2 (3 total, Query took 0.0001 seconds.)

```
SELECT CUSTOMER.CUSTNAME,CUSTOMER.CUSTID, PRICE FROM CUSTOMER LEFT JOIN CARS ON CUSTOMER.CARBOUGHTID=cars.ID;
```

☐ 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

CUSTNAME	CUSTID	PRICE
KIRAN SS	CUST001	23000000
RAHUL M	CUST002	33000000
CHIRAYU	CUST003	9500000

3) QUERY TO DISPLAY CAR ACCORDING TO THE SHOWROOM IN WHICH THEY ARE AVAILABLE

```
SELECT CARS.ID,CARS.MAKE,CARS.MODEL,CARS.PRICE,NAME,SID,STATE,CITY,PHNO FROM CARS LEFT JOIN SHOWROOM ON CARS.SIDD=SHOWROOM.SID ORDER BY showroom.SID;
```

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

☐ Show all | Number of rows: Filter rows: Sort by key:

Extra options

ID	MAKE	MODEL	PRICE	NAME	SID	STATE	CITY	PHNO
C001	BMW	M8	23000000	AUTOBEST	CS001	KA	BLR	9999007777
C006	PORSCHE	MACAN GTS	18000000	AUTOBEST	CS001	KA	BLR	9999007777
C002	BMW	M5	33000000	AUTOBEST	CS001	KA	BLR	9999007777
C003	VOLVO	XC90	9500000	AUTOBAHN	CS002	KA	MYS	9999005555
C004	MERCEDES	G WAGON	32000000	AUTOBAHN	CS002	KA	MYS	9999005555
C005	PORSCHE	911 TURBO S	36000000	AUTOBAHN	CS002	KA	MYS	9999005555
C008	BENTLEY	BENTAYGA	28000000	AUTOMARC	CS003	KA	MNGLR	9999006666
C009	BENTLEY	CONTINENTAL GT	8900000	AUTOMARC	CS003	KA	MNGLR	9999006666
C007	TOYOTA	LANDCRUISER LC300	21000000	AUTOMARC	CS003	KA	MNGLR	9999006666

4) QUESRY TO DISPLAY CARS BOUGHT BY CUSTOMER

```
SELECT CUSTOMER.CUSTNAME,CUSTOMER.CUSTID,CUSTOMER.PHNO,CUSTOMER.EMAIL,MAKE,MODEL,YEAR,PRICE FROM CUSTOMER RIGHT JOIN CARS ON CUSTOMER.CARBOUGHTID=CARS.ID ORDER BY CUSTOMER.CUSTID;
```

CUSTNAME	CUSTID ▲ 1	PHNO	EMAIL	MAKE	MODEL	YEAR	PRICE
NULL	NULL	NULL	NULL	PORSCHE	MACAN GTS	2019	18000000
NULL	NULL	NULL	NULL	BENTLEY	BENTAYGA	2018	28000000
KIRAN SS	CUST001	988019345	kiran@gmail.com	BMW	M8	2021	23000000
KIRAN SS	CUST001	988019345	kiran@gmail.com	PORSCHE	911 TURBO S	2022	36000000
RAHUL M	CUST002	991239345	rahul@gmail.com	BMW	M5	2021	33000000
RAHUL M	CUST002	991239345	rahul@gmail.com	BENTLEY	CONTINENTAL GT	2016	8900000
CHIRAYU	CUST003	991231232	chirayu@gmail.com	MERCEDES	G WAGON	2021	32000000
CHIRAYU	CUST003	991231232	chirayu@gmail.com	VOLVO	XC90	2019	9500000
BHAVANTH	CUST004	8989989800	bhavanth@gmail.com	TOYOTA	LANDCRUISER LC300	2019	21000000

Aggregate Functions

Showcase at least 4 Aggregate function queries Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1) QUERY TO DISPLAY NUMBER OF CARS BOUGHT BY EACH CUSTOMER

```
SELECT customer.CUSTNAME, count(*) as COUNT FROM CUSTOMER GROUP BY customer.CUSTNAME;
```

CUSTNAME	COUNT
BHAVANTH	1
CHIRAYU	2
KIRAN SS	2
RAHUL M	2

2) QUERY TO DISPLAY THE AVERAGE PRICE OF CAR PER TYPE OF CAR

ALTER TABLE CARS MODIFY PRICE int;

```
SELECT specs.TYPE,AVG(CARS.PRICE) FROM CARS RIGHT JOIN SPECS ON CARS.MODEL=SPECS.MODELOFCAR GROUP BY SPECS.TYPE;
```

✓ Showing rows 0 - 3 (4 total, Query took 0.0005 seconds.)

SELECT specs.TYPE,AVG(CARS.PRICE) FROM CARS RIGHT JOIN SPECS ON CARS.MODEL=SPECS.MODELOFCAR GROUP BY SPECS.TYPE;

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

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

Extra options

TYPE	AVG(CARS.PRICE)
GT	15950000.0000
SEDAN	33000000.0000
SUERCAR	36000000.0000
SUV	21700000.0000

3) QUERY TO DISPLAY THE MAX &MIN VALUE OF A CAR ACCORDING TO TYPE

```
SELECT specs.TYPE,AVG(CARS.PRICE),MAX(CARS.PRICE),MIN(CARS.PRICE) FROM CARS RIGHT JOIN SPECS ON CARS.MODEL=SPECS.MODELOFCAR GROUP BY SPECS.TYPE;
```


TYPE	AVG(CARS.PRICE)	MAX(CARS.PRICE)	MIN(CARS.PRICE)
GT	15950000.0000	23000000	8900000
SEDAN	33000000.0000	33000000	33000000
SUERCAR	36000000.0000	36000000	36000000
SUV	21700000.0000	32000000	9500000

4) QUERY TO FIND THE NUMBER OF CARS EACH SHOWROOM HAS

```
select showroom.NAME, count(*) as Count from showroom NATURAL join CARS WHERE showroom.SID=cars.SIDD group by showroom.SID;
```

Extra options

NAME	Count
AUTOBEST	3
AUTOBAHN	3
AUTOMARC	3

Set Operations

Showcase at least 4 Set Operations queries Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1) QUERY TO FIND LIST OF CUSTOMERS WHO HAVE EITHER AN SUV OR A SUPERCAR

```
select customer.CUSTNAME from (CUSTOMER NATURAL join CARS) NATURAL join specs where specs.TYPE='SUV' UNION select customer.CUSTNAME from (CUSTOMER NATURAL join CARS) NATURAL join specs where specs.TYPE='SUPERCAR';
```

CUSTNAME
KIRAN SS
RAHUL M
CHIRAYU
BHAVANTH
HITLER
SELMON BOI
JAMES BOND

2) QUERY TO FIND LIST OF CUSTOMERS WHO HAVE A BMW AND A FERRARI

```
SELECT CUSTNAME,CARBOUGHT FROM customer FULL JOIN cars WHERE CARBOUGHT LIKE '%BMW%' GROUP BY CARBOUGHT UNION SELECT CUSTNAME,CARBOUGHT FROM customer FULL JOIN cars WHERE CARBOUGHT LIKE '%FERRARI%' GROUP BY CARBOUGHT;
```

CUSTNAME	CARBOUGHT
RAHUL M	BMW M5
KIRAN SS	BMW M8
KIRAN SS	FERRARI 812 SUPERFAS
RAHUL M	FERRARI F8 TRIBUTO

3) QUERY TO FIND ALL SUVS WITH DISP>3000cc

```
SELECT MODELOFCAR,MAKEOFCAR,TYPE FROM SPECS WHERE TYPE='SUV' INTERSECT SELECT MODELOFCAR,MAKEOFCAR,TYPE FROM SPECS WHERE DISP>'3000';
```

MODELOFCAR	MAKEOFCAR	TYPE
G WAGON	MERCEDES	SUV
BENTAYGA	BENTLEY	SUV
MODEL X	TESLA	SUV

Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results

1) FUNCTION TO CALCULATE SERVICE COST ACCORDING TO THE CAR

DELIMITER \$\$

CREATE FUNCTION getservicecost(POWER INT) RETURNS int

BEGIN

DECLARE COST INT;

IF POWER>=600 THEN

SET COST=((10000-2000)*3+4000);

ELSEIF DISP<600 THEN

POWER COST=((10000-4000)*3+2000);

END IF;

RETURN COST;

END

\$\$

DELIMITER

ALTER TABLE SPECS MODIFY POWER INT

SELECT MODELOFCAR,POWER,getservicecost(POWER) AS SERVICECOST FROM specs;

MODELOFCAR	POWER	SERVICECOST
M5	650	28000
XC90	320	20000
G WAGON	560	20000
911 TURBO S	650	28000
MACAN GTS	325	20000
LANDCRUISER LC300	300	20000
BENTAYGA	610	28000
CONTINENTAL GT	578	20000
HURACAN 580-2	580	20000
HURACAN 610-4	610	28000
812 SUPERFAST	820	28000
F8 TRIBUTO	760	28000
MODEL X	1000	28000
MODEL S	1000	28000
ONE-77	770	28000
A45 AMG	450	20000
S JCW	330	20000
GT	750	28000

Triggers and Cursors

Create a Trigger and a Cursor. State the objective. Run and display the results.

1) TRIGGER TO MAKE SURE THAT ANY NEW CAR INSERTED SHOULD HAVE A PRICE OF MORE THAN 100,000

```
DELIMITER $$
CREATE TRIGGER check_price
AFTER INSERT
ON cars FOR EACH ROW
BEGIN
    DECLARE error_msg VARCHAR(255);
    declare val int;
    SET error_msg = ('The cost cannot be less than 100000. ');
    SET val=(select PRICE from CARS where ID=new.ID);
    IF val<100000 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = error_msg;
        DELETE FROM CARS WHERE ID=new.ID;
    END IF;
END $$
DELIMITER ;
```

Error

SQL query: [Copy](#)

```
INSERT INTO CARS VALUES('C020','TATA','INDICA','2007','80000','CS001');
```

MySQL said: 

#1644 - The cost cannot be less than 100000.

✓ 1 row inserted. (Query took 0.0002 seconds.)

```
INSERT INTO CARS VALUES('C020','HONDA','CITY','2017','800000','CS001');
```

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

- 1) Procedure that backups the contents of the cars table to the backup table using cursors:

```
DELIMITER //
CREATE PROCEDURE carbackup()
BEGIN
DECLARE done INT DEFAULT 0;
DECLARE SID varchar(25);
DECLARE Fname, Lname, Gender, Street, City VARCHAR(255);
DECLARE ID varchar(30);
DECLARE MAKE varchar(30);
DECLARE MODEL varchar(30);
DECLARE YEAR varchar(30);
DECLARE PRICE int;
DECLARE SIDD varchar(13);
DECLARE cur CURSOR FOR SELECT * FROM cars;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
OPEN cur;
label: LOOP
FETCH cur INTO ID, MAKE, MODEL, YEAR, PRICE, SIDD
INSERT INTO backupcars VALUES(ID, MAKE, MODEL, YEAR, PRICE, SIDD);
IF done = 1 THEN LEAVE label;
END IF;
END LOOP;
CLOSE cur;
END//
DELIMITER ;
```

SELECT * FROM `backupcars`

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
















































☐ Show all

Number of rows: 25

Filter rows:

Sort by key: None

Extra options

<div><div><div><div></div><div></div><div></div></div></div><div></div></div>						ID	MAKE	MODEL	YEAR	PRICE	SIDD
<input type="checkbox"/>	 Edit	 Copy	 Delete	C001	BMW	M8	2021	23000000	CS001		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C002	BMW	M5	2021	33000000	CS001		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C003	VOLVO	XC90	2019	9500000	CS002		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C004	MERCEDES	G WAGON	2021	32000000	CS002		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C005	PORSCHE	911 TURBO S	2022	36000000	CS002		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C006	PORSCHE	MACAN GTS	2019	18000000	CS001		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C007	TOYOTA	LANDCRUISER LC300	2019	21000000	CS003		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C008	BENTLEY	BENTAYGA	2018	28000000	CS003		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C009	BENTLEY	CONTINENTAL GT	2016	8900000	CS003		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C010	LAMBORGHINI	HURACAN 580-2	2017	25000000	CS001		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C011	LAMBORGHINI	HURACAN 610-4	2019	27500000	CS001		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C012	FERRARI	812 SUPERFAST	2020	47500000	CS003		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C013	FERRARI	F8 TRIBUTO	2022	37500000	CS002		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C014	TESLA	MODEL S	2020	17500000	CS003		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C015	TESLA	MODEL X	2021	18500000	CS003		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C016	ASTON MARTIN	ONE-77	2015	20000000	CS001		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C017	MERCEDES	A45 AMG	2021	12000000	CS001		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C018	MINI COOPER	S JCW	2017	2900000	CS003		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C019	MC LAREN	GT	2020	32900000	CS001		
<input type="checkbox"/>	 Edit	 Copy	 Delete	C020	HONDA	CITY	2017	800000	CS001		

