

INSURANCE DATABASE

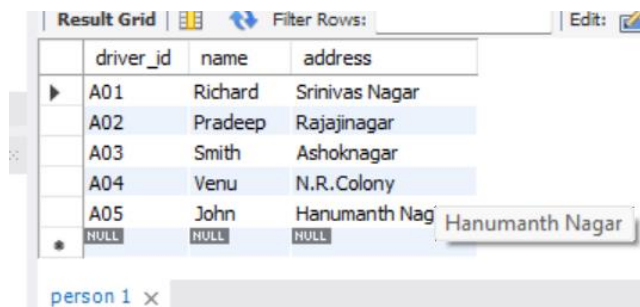
1. TABLE CREATION:

```
create database insurance;
use insurance;

create table person (
    driver_id varchar(10),
    name varchar(20),
    address varchar(30),
    primary key(driver_id)
);
insert into person values('A01','Richard','Srinivas Nagar');
insert into person values('A02','Pradeep','Rajajinagar');
insert into person values('A03','Smith','Ashoknagar');
insert into person values('A04','Venu','N.R.Colony');
insert into person values('A05','John','Hanumanth Nagar');

select * from person;
```

OUTPUT



	driver_id	name	address
▶	A01	Richard	Srinivas Nagar
	A02	Pradeep	Rajajinagar
	A03	Smith	Ashoknagar
	A04	Venu	N.R.Colony
	A05	John	Hanumanth Nagar
*	NULL	NULL	NULL

person 1 x

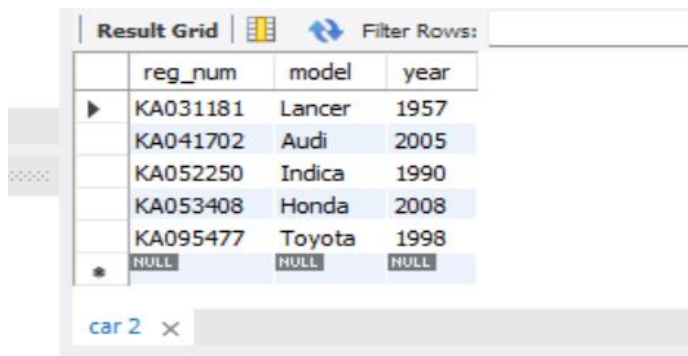
```
create table car(
    reg_num varchar(10),
    model varchar(10),
    year int,
    primary key(reg_num)
);
insert into car values('KA052250','Indica', 1990);
insert into car values('KA031181','Lancer', 1957);
```

```

insert into car values('KA095477','Toyota', 1998);
insert into car values('KA053408','Honda', 2008);
insert into car values('KA041702','Audi', 2005);
select*from car;

```

OUTPUT



	reg_num	model	year
▶	KA031181	Lancer	1957
	KA041702	Audi	2005
	KA052250	Indica	1990
	KA053408	Honda	2008
	KA095477	Toyota	1998
*	NULL	NULL	NULL

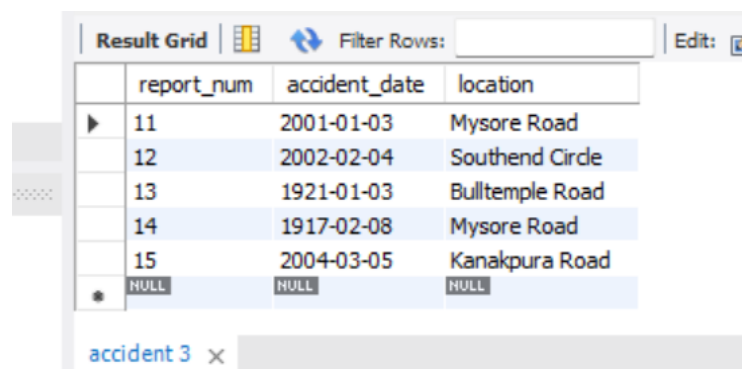
car 2 x

```

create table accident(report_num int,
accident_date date,
location varchar(20),
primary key(report_num)
);
insert into accident values(11,'2001-01-03','Mysore Road');
insert into accident values(12,'2002-02-04','Southend Circle');
insert into accident values(13,'1921-01-03','Bulltemple Road');
insert into accident values(14,'1917-02-08','Mysore Road');
insert into accident values(15,'2004-03-05','Kanakpura Road');
SELECT*FROM accident;

```

OUTPUT



	report_num	accident_date	location
▶	11	2001-01-03	Mysore Road
	12	2002-02-04	Southend Circle
	13	1921-01-03	Bulltemple Road
	14	1917-02-08	Mysore Road
	15	2004-03-05	Kanakpura Road
*	NULL	NULL	NULL

accident 3 x

```

create table owns(
driver_id varchar(10),
reg_num varchar(10),
primary key(driver_id, reg_num),
foreign key(driver_id) references person(driver_id),
foreign key(reg_num) references car(reg_num)
);

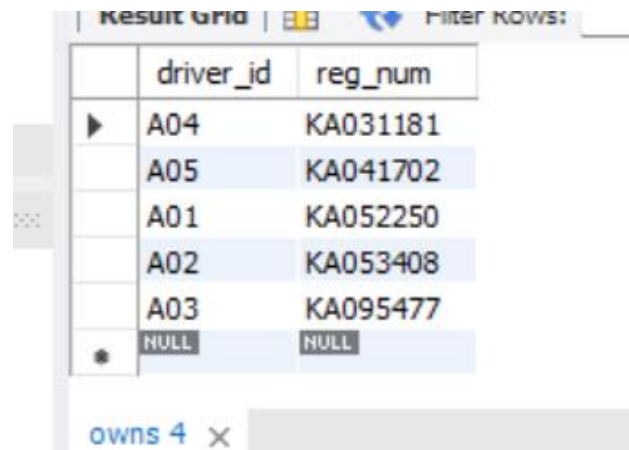
```

```

insert into owns values ('A01','KA052250');
insert into owns values ('A02','KA053408');
insert into owns values ('A04','KA031181');
insert into owns values ('A03','KA095477');
insert into owns values ('A05','KA041702');
select*from owns

```

OUTPUT



	driver_id	reg_num
▶	A04	KA031181
	A05	KA041702
⋮	A01	KA052250
	A02	KA053408
	A03	KA095477
•	NULL	NULL

owns 4 ×

```

create table participated(
driver_id varchar(10),
reg_num varchar(10),
report_num int,
damage_amount int,
primary key(driver_id, reg_num, report_num),
foreign key(driver_id) references person(driver_id),
foreign key(reg_num) references car(reg_num),

```

foreign key(report_num) references accident(report_num));

insert into participated values ('A01','KA052250',11,10000);

insert into participated values ('A02','KA053408',12,50000);

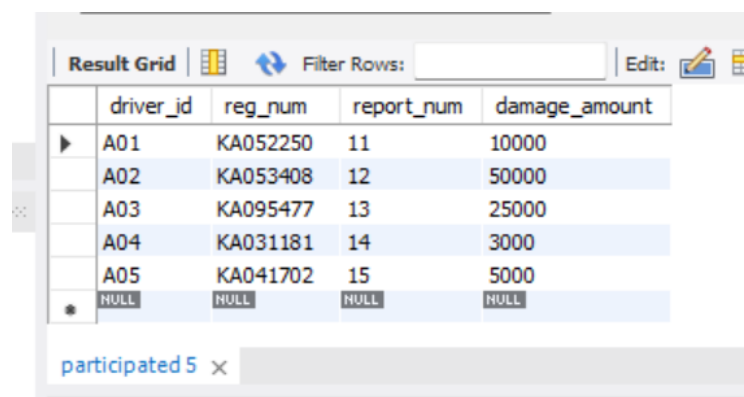
insert into participated values ('A03','KA095477',13,25000);

insert into participated values ('A04','KA031181',14,3000);

insert into participated values ('A05','KA041702',15,5000);

select*from participated;

OUTPUT



	driver_id	reg_num	report_num	damage_amount
▶	A01	KA052250	11	10000
	A02	KA053408	12	50000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000
*	NULL	NULL	NULL	NULL

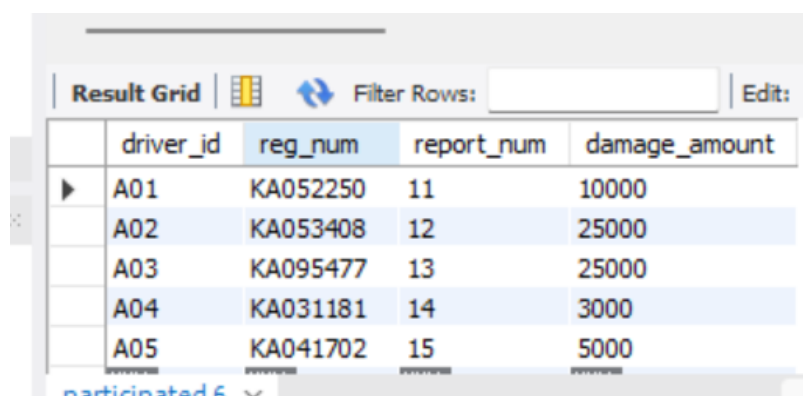
participated 5 ×

2. Update the damage amount to 25000 for the car with a specific re_num

update participated

set damage_amount=25000

where reg_num='KA053408' and report_num=12;

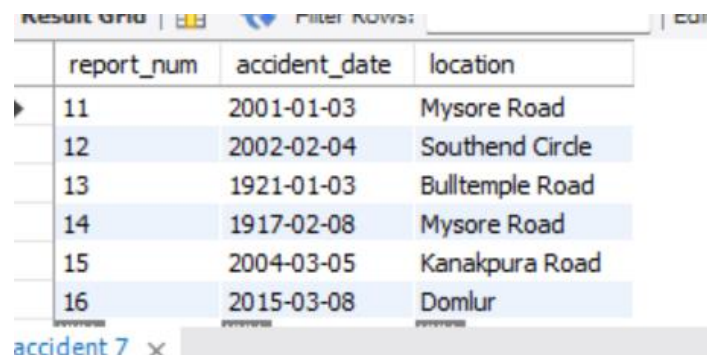


	driver_id	reg_num	report_num	damage_amount
▶	A01	KA052250	11	10000
	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000

participated 6 ▼

3. Add a new accident to the database.

```
insert into accident values(16,'15-03-08','Domlur');  
select*from accident;
```



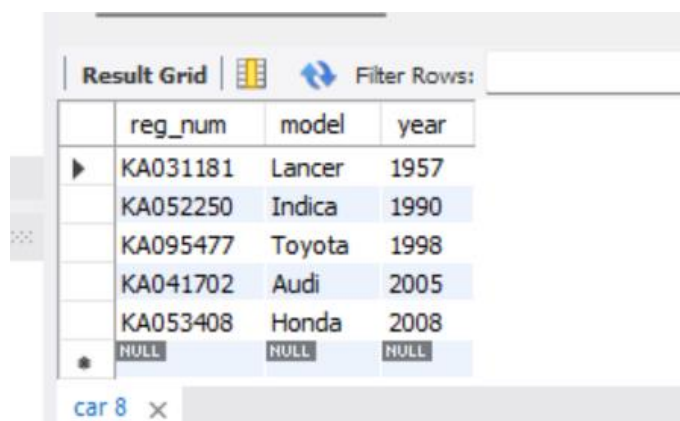
The screenshot shows a 'Result Grid' window with a table containing accident records. The table has three columns: 'report_num', 'accident_date', and 'location'. The data is as follows:

report_num	accident_date	location
11	2001-01-03	Mysore Road
12	2002-02-04	Southend Circle
13	1921-01-03	Bulltemple Road
14	1917-02-08	Mysore Road
15	2004-03-05	Kanakpura Road
16	2015-03-08	Domlur

Below the table, there is a tab labeled 'accident 7' with a close button 'x'.

4. Display the entire CAR relation in the ascending order of manufacturing year.

```
select * from car  
order by year asc;
```



The screenshot shows a 'Result Grid' window with a table containing car records. The table has three columns: 'reg_num', 'model', and 'year'. The data is as follows:

reg_num	model	year
KA031181	Lancer	1957
KA052250	Indica	1990
KA095477	Toyota	1998
KA041702	Audi	2005
KA053408	Honda	2008
NULL	NULL	NULL

Below the table, there is a tab labeled 'car 8' with a close button 'x'.

5. Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved

```
select count(report_num) AS CNT  
from car c,participated p  
where c.reg_num=p.reg_num and model="Indica";
```

Result Grid		Filter Rows:
	CNT	
▶	1	

6. Find the total number of people who owned cars that involved in accidents in 2008.

```
SELECT COUNT(DISTINCT A.driver_id) AS CNT
FROM
    participated A,
    accident B
WHERE
    A.report_num = B.report_num
    AND B.accident_date LIKE '%08';
```

Result Grid		Filter Rows:
	CNT	
▶	1	

7. Find the number of accidents in which cars belonging to a specific model (ex: 'Lancer') were involved accidents in 2008.

```
select count(distinct driver_id) CNT
from participated a, accident b
where a.report_num= b.report_num and b.accident_date like '%08';
```

Result Grid		Filter Rows:
	CNT	
▶	1	

Result 11 x

8. LIST THE ENTIRE PARTICIPATED RELATION IN THE DESCENDING ORDER OF DAMAGE AMOUNT.

```
SELECT * FROM participated
ORDER BY damage_amount DESC;
```

Result Grid

Filter Rows:

Edit:

	driver_id	reg_num	report_num	damage_amount
▶	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A01	KA052250	11	10000
	A05	KA041702	15	5000
	A04	KA031181	14	3000
*	NULL	NULL	NULL	NULL

participated 12

×

Apply

Revert

9. FIND THE AVERAGE DAMAGE AMOUNT

```
SELECT AVG(damage_amount)
FROM participated;
```

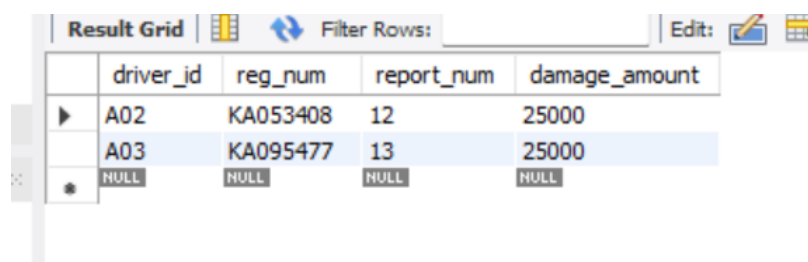
Result Grid		Filter Rows:
	AVG(damage_amount)	
▶	13600.0000	

10.DELETE THE TUPLE FROM PARTICIPATED RELATION WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT

```
SET SQL_SAFE_UPDATES = 0;
```

```
DELETE FROM PARTICIPATED
WHERE DAMAGE_AMOUNT < (
    SELECT avg_damage FROM (
        SELECT AVG(DAMAGE_AMOUNT) AS avg_damage FROM
PARTICIPATED
    ) AS temp
);
```

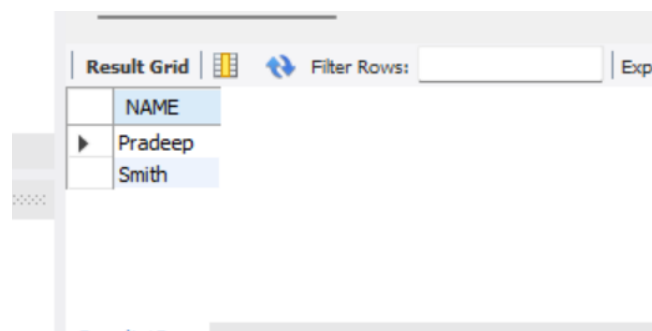
```
SELECT * FROM PARTICIPATED;
```



	driver_id	reg_num	report_num	damage_amount
▶	A02	KA053408	12	25000
	A03	KA095477	13	25000
✱	NULL	NULL	NULL	NULL

11. LIST THE NAME OF DRIVERS WHOSE DAMAGE IS GREATER THAN THE AVERAGE DAMAGE AMOUNT.

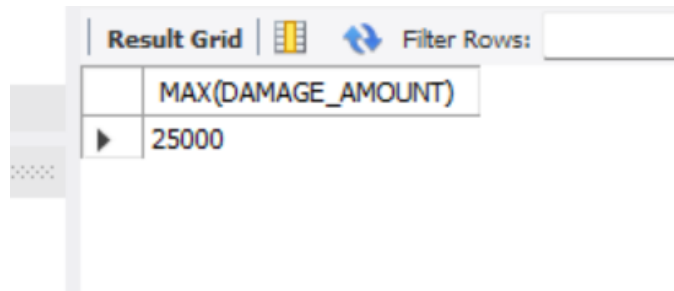
```
SELECT NAME FROM person A, participated B
WHERE A.driver_id = B.driver_id
AND damage_amount > (SELECT AVG(damage_amount) FROM participated);
```



	NAME
▶	Pradeep
	Smith

12.FIND MAXIMUM DAMAGE AMOUNT.

```
select max(damage_amount) from participated;
```



The screenshot shows a database query result grid. At the top, there is a header bar with the text "Result Grid" and a "Filter Rows:" input field. Below the header, the query result is displayed in a table with one column and one row. The column header is "MAX(DAMAGE_AMOUNT)" and the value in the row is "25000".

MAX(DAMAGE_AMOUNT)
25000