

2. More Queries on Insurance Database

PROGRAM 2. More Queries on Insurance Database

PERSON (driver_id: String, name: String, address: String)

CAR (reg_num: String, model: String, year: int)

ACCIDENT (report_num: int, accident_date: date, location: String)

OWNS (driver_id: String, reg_num: String)

PARTICIPATED (driver_id: String, reg_num: String, report_num: int, damage_amount: int)

Create the above tables by properly specifying the primary keys and the foreign keys as done in "Program 1" week's lab and Enter at least five tuples for each relation.

- Display the entire CAR relation in the ascending order of manufacturing year.
- Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved.
- Find the total number of people who owned cars that involved in accidents in 2008.
- List the Entire Participated Relation in the Descending Order of Damage Amount. Find the Average Damage Amount.
- Delete the Tuple Whose Damage Amount is below the Average Damage Amount.
- List the Name of Drivers Whose Damage is Greater than The Average Damage Amount.
- Find Maximum Damage Amount.

Creating database and table:

Database insurance_141 and tables as per schema were created in the previous lab and it is as shown in the previous experiment.

Queries :

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

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

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

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

```
select count(report_num)
from car c, participated p
where c.reg_num=p.reg_num and c.model='Lancer';
```

count(report_num)
1

- iii. Find the total number of people who owned cars that were involved in 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 '%2008%';
```

CNT
1

- iv. List the entire participated relation in the descending order of damage amount.

```
select * from participated order by damage_amount desc;
```

Find the average damage amount

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

AVG(damage_amount)
13600.0000

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

v. Delete the tuple whose damage amount is below the average damage amount .

```
delete from participated where damage_amount < (select
p.damage_amount from(select AVG(damage_amount) as
damage_amount FROM participated p);
select * from participated;
```

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

vi. List the name of drivers whose damage is greater than the average damage amount.

```
select name from person p, participated part where p.driver_id=part.driver_id and
damage_amount>(select AVG(damage_amount) FROM participated);
```

name

vii. Find maximum damage amount.

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

MAX(damage_amount)
25000