

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



LAB REPORT on

DATABASE MANAGEMENT SYSTEMS

Submitted by

Dhanush H V
(1BM21CS052)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

October-2022 to Feb-2023

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "Database Management Systems (22CS3PCDBM)" carried out by DHANUSH HV (1BM21CS052), who is bonafide student of B. M. S. College of Engineering. It is in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (22CS3PCDBM) work prescribed for the said degree.

Pallavi G B

Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak

Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

| Sl. No. | Date | Experiment Title | Page No. |
|--------------------|-------------|------------------------------------|-----------------|
| 1 | 1/11/22 | Insurance Database | 3-8 |
| 2 | 8/11/22 | More Queries on Insurance Database | 8-9 |
| 3 | 15/11/22 | Bank Database | 10-16 |
| 4 | 29/11/22 | More Queries on Bank Database | 16-19 |
| 5 | 06/12/22 | Employee Database | 20-24 |
| 6 | 13/12/22 | More Queries on Employee Database | 25-30 |
| 7 | 20/12/22 | Supplier Database | 31-36 |
| 8 | 27/12/22 | Flight Database | 37-43 |
| 9 | 17/01/23 | NoSQL | 44-48 |
| 10 | 24/01/23 | NO SQL | 48-51 |

Program 1: INSURANCE DATABASE

PERSON (driver-id #: String, name: String, address: String)

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

ACCIDENT (report-number: int, date: date, location: String)

OWNS (driver-id #: String, Regno: String)

PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int)

Schema Diagram

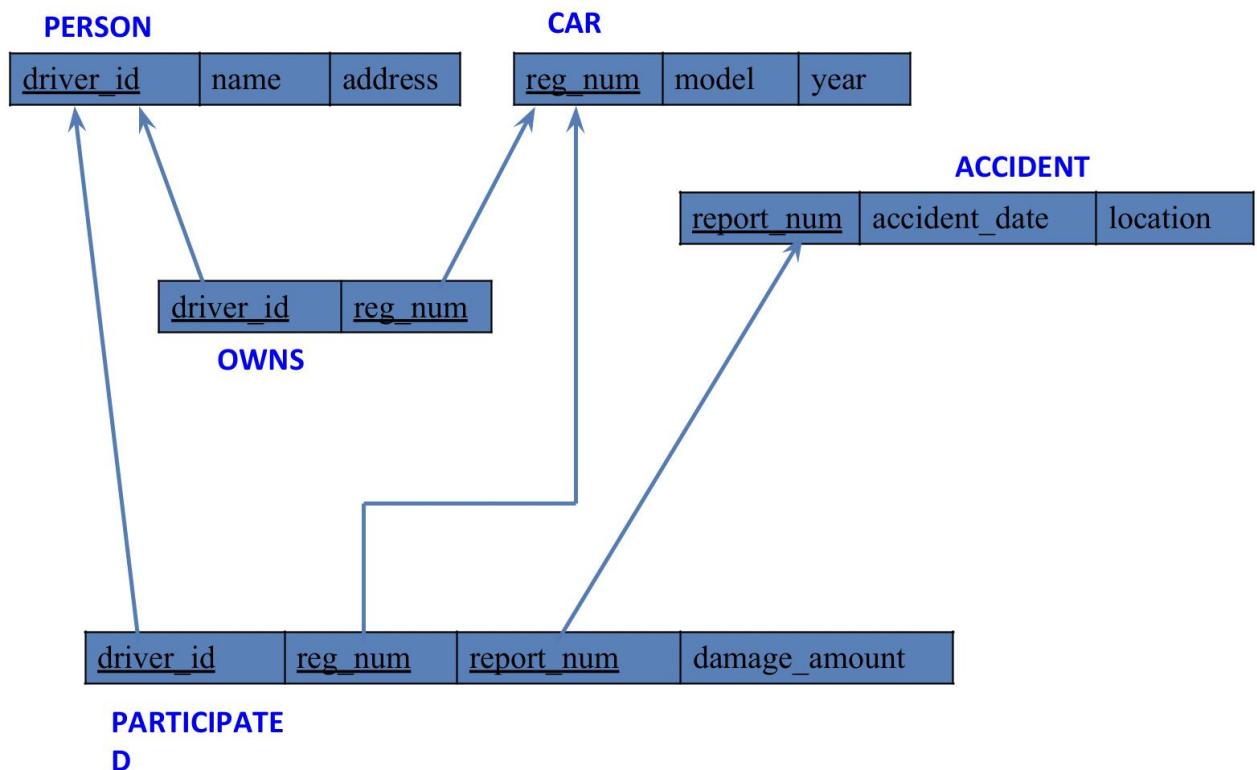


Table Creation:

```
create database 1bm21c052_insurance;
use 1bm21c052_insurance;
```

```
create table person (
driveri_d varchar(10),
name varchar(30),
address varchar(30), primary key(driver_id)
desc person);
```

```
create table car(
reg_num varchar(10),
model varchar(10), year int.
```

```

primary key(regnum)
create table accident
report_num int, accident_date date, location varchar(20), primary key(report_num);

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)
);
create table participated
driver_id varchar(10),
reg_num varchar(10),
reportnum int,
damageamount 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(reportnum) references accident(reportnum)

```

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|-----------|-------------|------|-----|---------|-------|
| ▶ | driver_id | varchar(10) | NO | PRI | NULL | |
| | name | varchar(20) | YES | | NULL | |
| | address | varchar(30) | YES | | NULL | |

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|---------|-------------|------|-----|---------|-------|
| ▶ | reg_num | varchar(10) | NO | PRI | NULL | |
| | model | varchar(10) | YES | | NULL | |
| | year | int | YES | | NULL | |

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|-----------|-------------|------|-----|---------|-------|
| ▶ | driver_id | varchar(10) | NO | PRI | NULL | |
| | reg_num | varchar(10) | NO | PRI | NULL | |

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|---------------|-------------|------|-----|---------|-------|
| ▶ | report_num | int | NO | PRI | NULL | |
| | accident_date | date | YES | | NULL | |
| | location | varchar(20) | YES | | NULL | |

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|---------------|-------------|------|-----|---------|-------|
| ▶ | driver_id | varchar(10) | NO | PRI | NULL | |
| | reg_num | varchar(10) | NO | PRI | NULL | |
| | report_num | int | NO | PRI | NULL | |
| | damage_amount | int | YES | | NULL | |

```
insert into accident values(11,2003-01-01,'Mysore road' );
insert into accident values(12,2004-02-02,'South end circle');
insert into accident values(13,2003-01-21,'Bull temple road' );
insert into accident values(14,2008-02-17,'Mysore road' );
insert into accident values(15,2004-03-05,'Kanakpura road' );
insert into person values('A01' "Richard", 'Srinivas nagar');
insert into person values(A02', Pradeep' Rajaji nagar');
```

Result Grid | Filter Rows: Edit:

| | report_num | accident_date | location |
|---|------------|---------------|-----------------|
| ▶ | 11 | 2003-01-01 | Mysore road |
| | 12 | 2004-02-02 | South end cirde |
| | 13 | 2003-01-21 | Bull temple end |
| | 14 | 2008-02-17 | Mysore road |
| | 15 | 2004-03-05 | Kanakapura road |
| * | HULL | HULL | HULL |

```
insert into person values('A01', 'Richard', 'Srinivas nagar');
insert into person values('A02', 'Pradeep', 'Rajaji nagar');
insert into person values(A03', 'Smith', 'Ashok nagar');
insert into person values('A04', 'Venu', 'N R Colony');
insert into person values('A05', 'Jhon', 'Hanumanth nagar');
```

47 * insert into car values('KA052250', 'Indica', 1990)

Result Grid | Filter Rows: Edit:

| | driver_id | name | address |
|---|-----------|---------|-----------------|
| ▶ | A01 | Richard | Srinivas nagar |
| | A02 | Pradeep | Rajaji nagar |
| | A03 | Smith | Ashok nagar |
| | A04 | Venu | N R. colony |
| | A05 | Jhon | Hanumanth nagar |
| * | HULL | HULL | HULL |

person 6 ×

Output

```
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);
```

```

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

```

| | driver_id | reg_num |
|---|-----------|----------|
| ▶ | A03 | KA031181 |
| | A05 | KA041702 |
| | A01 | KA052250 |
| | A02 | KA053408 |
| | A04 | KA095477 |

OWNS 9.x

```

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);

```

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

Queries:

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

SQL> 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 |

car 11 x

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

SQL> select count(reportnum)

from car c, participated p

where c.reg_num=p.reg_num and c.model='Lancer";

| count(report_num) |
|-------------------|
| 1 |

3. Find the total number of people who owned cars that were involved in accidents in 2008. SQL> select count(distinct driver_id)

from participated a, accident b

where a.reportnum=b.reportnum and b.accidentdate like '2008%"

| count(distinct driver_id) |
|---------------------------|
| 1 |

TO DO

List the entire participated relation in descending order of damage amount. SQL> select * from participated

order by damage_amount desc;

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

Find the average damage amount.

SQL> select avg(damage_amount) from participated;

| Result Grid | | Filter Rows: | Export: |
|---------------------------------|------------|--------------|---------|
| <code>avg(damage_amount)</code> | | | |
| ▶ | 18600.0000 | | |

Result 14 ×

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

SQL> delete from participated

where damage_amount < (select t.avg1 from (select avg (damage_amount) as avg1 from participated) t);

select *from participated;

| Result Grid | | | | Filter Rows: | Edit: |
|-------------|-----------|----------|------------|---------------|-------|
| | driver_id | reg_num | report_num | damage_amount | |
| ▶ | A02 | KA053408 | 12 | 50000 | |
| ▶ | A03 | KA095477 | 13 | 25000 | |
| * | NULL | NULL | NULL | NULL | |

participated 7 ×

Week 2

More Queries on Insurance Database

- LIST THE ENTIRE PARTICIPATED RELATION IN THE DESCENDING ORDER OF DAMAGE AMOUNT. SQL>select * from participated order by(damage_amount) desc;

| | driver_id | reg_num | report_num | damage_amount |
|---|-----------|---------|------------|---------------|
| ▶ | A02 | 031181 | 12 | 50000 |
| ▶ | A03 | 095477 | 13 | 25000 |
| | A01 | 052250 | 11 | 10000 |
| | A05 | 041702 | 15 | 5000 |
| | A04 | 053408 | 14 | 3000 |
| * | NULL | NULL | NULL | NULL |

- FIND THE AVERAGE DAMAGE AMOUNT SQL>select avg(damage_amount) from participated;

| Result Grid | |
|-------------|--------------------|
| | avg(damage_amount) |
| ▶ | 13600.0000 |

- DELETE THE TUPLE WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT SQL>delete from participated where damage_amount<(select p.amt from(select avg(damage_amount)as amt from participated) p); SQL>select * from participated;

| Result Grid | | | | |
|-------------|-----------|---------|------------|---------------|
| | driver_id | reg_num | report_num | damage_amount |
| ▶ | A02 | 031181 | 12 | 50000 |
| ▶ | A03 | 095477 | 13 | 25000 |
| ● | NULL | NULL | NULL | NULL |

- LIST THE NAME OF DRIVERS WHOSE DAMAGE IS GREATER THAN THE AVERAGE DAMAGE AMOUNT. SQL>select name from person,participated where person.driver_id=participated.driver_id and damage_amount>(select avg(damage_amount) from participated);

| Result Grid | |
|-------------|---------|
| | name |
| ▶ | Pradeep |
| | Smith |

- FIND MAXIMUM DAMAGE AMOUNT. SQL>select * from participated;

| Result Grid | |
|-------------|--------------------|
| | MAX(damage_amount) |
| ▶ | 25000 |

Week 3

Program 2: BANKING ENTERPRISE DATABASE

BRANCH (branch-name: String, branch-city: String, assets: real)

ACCOUNTS (accno: int, branch-name: String, balance: real)

DEPOSITOR (customer-name: String, customer-street: String, customer-city: String) LOAN (loan-number: int, branch-name: String, amount: real)

BORROWER (customer-name: String, loan-number: int)

Schema Diagram

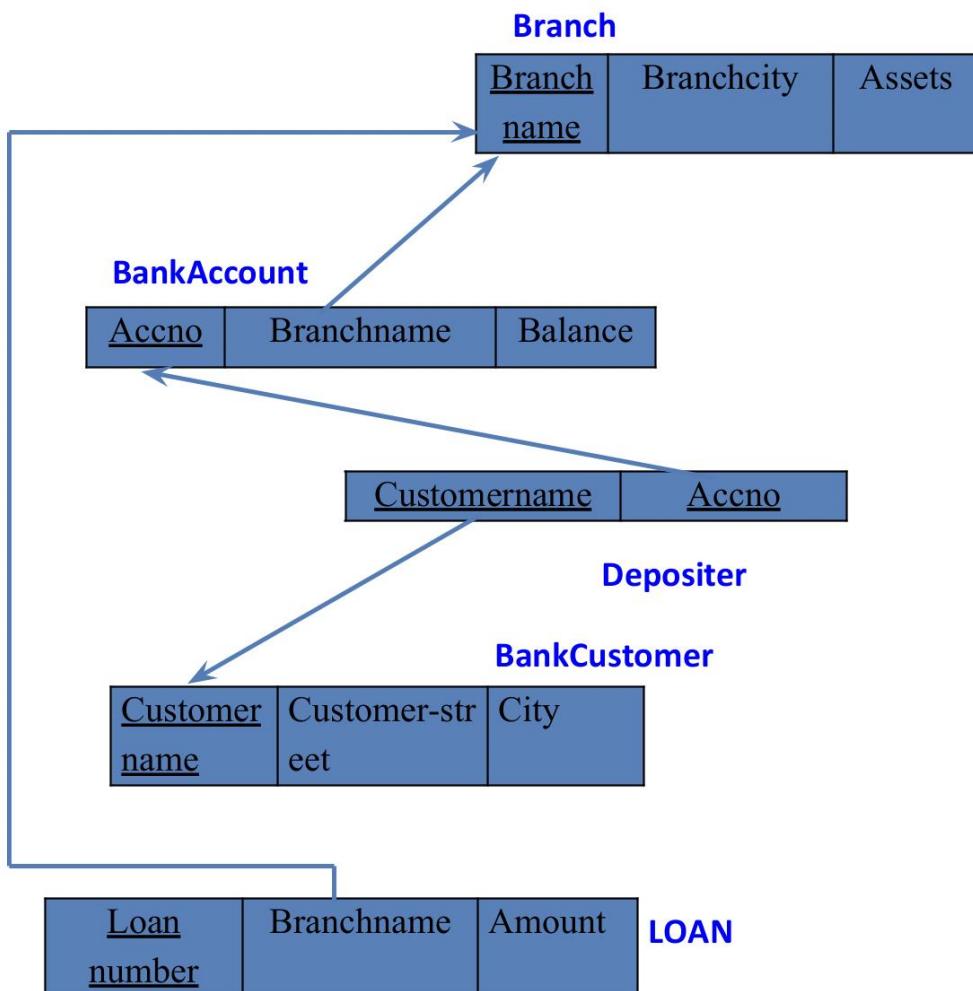


Table Creation:

```
create database 1bm21cs052_Bank1;
use 1bm21cs052_Bank1;
```

```
create table branch(
branch_name varchar(20),
branch_city varchar(10),
assets real,
PRIMARY KEY(branch_name)
);
create table bankCustomer(
customer_name varchar(20),
customer_street varchar(20),
customer_city varchar(15),
PRIMARY KEY(customer_name)
);
```

```
create table depositer(
customer_name varchar(20),
```

```

accno int,
FOREIGN KEY(customer_name) REFERENCES bankCustomer(customer_name)
ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY(accno) REFERENCES bankAccount(accno)
ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

create table loan(
loan_no int,
branch_name varchar(20),
amount real,
PRIMARY KEY(loan_no),
FOREIGN KEY(branch_name) REFERENCES branch(branch_name)
ON UPDATE CASCADE ON DELETE CASCADE
);

```

| | Field | Type | Null | Key | Default | Extra |
|---|------------|-------------|------|-----|---------|-------|
| ▶ | accno | int | NO | PRI | NULL | |
| | branchname | varchar(20) | YES | MUL | NULL | |
| | balance | double | YES | | NULL | |

| | Field | Type | Null | Key | Default | Extra |
|---|----------------|-------------|------|-----|---------|-------|
| ▶ | Customername | varchar(20) | NO | PRI | NULL | |
| | Customerstreet | varchar(20) | YES | | NULL | |
| | city | varchar(20) | YES | | NULL | |

| | Field | Type | Null | Key | Default | Extra |
|---|------------|-------------|------|-----|---------|-------|
| ▶ | branchname | varchar(20) | NO | PRI | NULL | |
| | branchcity | varchar(10) | YES | | NULL | |
| | assets | double | YES | | NULL | |

| | Field | Type | Null | Key | Default | Extra |
|---|--------------|-------------|------|-----|---------|-------|
| ▶ | Customername | varchar(20) | YES | MUL | NULL | |
| | accno | int | YES | MUL | NULL | |

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|------------|-------------|------|-----|---------|-------|
| ▶ | loannumber | int | NO | PRI | NULL | |
| | branchname | varchar(20) | YES | MUL | NULL | |
| | amount | double | YES | | NULL | |

(Inserting values in table)

```

insert into branch values("SBI_Chamrajpet","Banglore",50000);
insert into branch values("SBI_ResidencyRoad","Banglore",10000);
insert into branch values("SBI_ShivajiRoad","Bombay",20000);
insert into branch values("SBI_ParliamentRoad","Delhi",10000);
insert into branch values("SBI_Jantarmantar","Delhi",20000);

insert into bankaccount values(1,"SBI_Chamrajpet",2000);
insert into bankaccount values(2,"SBI_ResidencyRoad",5000);
insert into bankaccount values(3,"SBI_ShivajiRoad",6000);
insert into bankaccount values(4,"SBI_ParliamentRoad",9000);
insert into bankaccount values(5,"SBI_Jantarmantar",8000);
insert into bankaccount values(6,"SBI_ShivajiRoad",4000);
insert into bankaccount values(8,"SBI_ResidencyRoad",4000);
insert into bankaccount values(9,"SBI_ParliamentRoad",3000);
insert into bankaccount values(10,"SBI_ResidencyRoad",5000);
insert into bankaccount values(11,"SBI_Jantarmantar",2000);
insert into bankcustomer values("Avinash","Bull_Temple_Road","Banglore");
insert into bankcustomer values("Dinesh","Bannerghatta_Road","Banglore");
insert into bankcustomer values("Mohan","NationalCollege_Road","Banglore");
insert into bankcustomer values("Nikil","Akbar_Road","Delhi");
insert into bankcustomer values("Ravi","Prithviraj_Road","Delhi");

insert into depositer values("Avinash",1);
insert into depositer values("Dinesh",2);
insert into depositer values("Nikil",4);

```

```

insert into depositer values("Ravi",5);

insert into depositer values("Avinash",8);

insert into depositer values("Nikil",9);

insert into depositer values("Dinesh",10);

insert into depositer values("Nikil",11);

```

```

insert into loan values(1,"SBI_Chamrajpet",1000);

insert into loan values(2,"SBI_ResidencyRoad",2000);

insert into loan values(3,"SBI_ShivajiRoad",3000);

insert into loan values(4,"SBI_ParliamentRoad",4000);

insert into loan values(5,"SBI_Jantarmantar",5000);

```

Queries:

(Displaying Tables)

select * from branch;

| | branchname | branchcity | assets |
|---|--------------------|------------|--------|
| ▶ | SBI_Chamrajpet | Banglore | 50000 |
| | SBI_Jantarmantar | Delhi | 20000 |
| | SBI_ParliamentRoad | Delhi | 10000 |
| | SBI_ResidencyRoad | Banglore | 10000 |
| | SBI_ShivajiRoad | Bombay | 20000 |
| * | NULL | NULL | NULL |

select * from bankaccount;

| | accno | branchname | balance |
|---|-------|--------------------|---------|
| ▶ | 1 | SBI_Chamrajpet | 2000 |
| | 2 | SBI_ResidencyRoad | 5000 |
| | 3 | SBI_ShivajiRoad | 6000 |
| | 4 | SBI_ParliamentRoad | 9000 |
| | 5 | SBI_Jantarmantar | 8000 |
| | 6 | SBI_ShivajiRoad | 4000 |
| | 8 | SBI_ResidencyRoad | 4000 |
| | 9 | SBI_ParliamentRoad | 3000 |
| | 10 | SBI_ResidencyRoad | 5000 |
| | 11 | SBI_Jantarmantar | 2000 |
| * | NULL | NULL | NULL |

select * from bankcustomer;

| | customername | customerstreet | city |
|---|--------------|----------------------|----------|
| ▶ | Avinash | Bull_Temple_Road | Banglore |
| | Dinesh | Bannergatta_Road | Banglore |
| | Mohan | NationalCollege_Road | Banglore |
| | Nikil | Akbar_Road | Delhi |
| | Ravi | Prithviraj_Road | Delhi |
| * | HULL | NULL | NULL |

select * from depositer;

| | customername | accno |
|---|--------------|-------|
| ▶ | Avinash | 1 |
| | Avinash | 8 |
| | Dinesh | 2 |
| | Dinesh | 10 |
| | Nikil | 4 |
| | Nikil | 9 |
| | Nikil | 11 |
| | Ravi | 5 |
| * | HULL | NULL |

select * from loan;

| | loannumber | branchname | amount |
|---|------------|--------------------|--------|
| ▶ | 1 | SBI_Chamrajpet | 1000 |
| | 2 | SBI_ResidencyRoad | 2000 |
| | 3 | SBI_ShivajiRoad | 3000 |
| | 4 | SBI_ParliamentRoad | 4000 |
| | 5 | SBI_Jantarmantar | 5000 |

- Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.

select branchname,assets/100000 as assets_in_lakkhs from branch;

| | branchname | assets_in_lakkhs |
|---|--------------------|------------------|
| ▶ | SBI_Chamrajpet | 0.5000 |
| | SBI_Jantarmantar | 0.2000 |
| | SBI_ParliamentRoad | 0.1000 |
| | SBI_ResidencyRoad | 0.1000 |
| | SBI_ShivajiRoad | 0.2000 |

- Find all the customers who have at least two accounts at the same branch (ex. SBI_ResidencyRoad).**

select d.customername from bankaccount b, depositer d

where b.branchname="SBI_ResidencyRoad" and b.accno=d.accno group by d.customername having count(d.accno)>=2;

| customername | |
|--------------|--------|
| ▶ | Dinesh |

- Create view which gives each branch the sum of the amount of all the loans at the branch**

create view sum_of_loan

as select branchname,sum(balance) from bankaccount

group by branchname;

select * from sum_of_loan;

| | branchname | sum(balance) |
|---|--------------------|--------------|
| ▶ | SBI_Chamrajpet | 2000 |
| | SBI_Jantarmantar | 10000 |
| | SBI_ParliamentRoad | 12000 |
| | SBI_ResidencyRoad | 14000 |
| | SBI_ShivajiRoad | 10000 |

Week 4

More QUERIES ON BANK DATABASE

```
create table borrower(
customer_name varchar(20),
loan_no int,
primary key(customer_name,loan_no),
foreign key(loan_no) references loan(loan_no)
);
```

```
insert into borrower values('avinash',1);
insert into borrower values('dinesh',2);
insert into borrower values('mohan',3);
insert into borrower values('nikil',4);
insert into borrower values('ravi',5);
```

```
select customer_name from bankCustomer
where customer_name=ANY(select customer_name from bankCustomer where
customer_city='delhi');
```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas
Schemas
1bm21cs052_bank1
Tables bankaccount bankcustomer borrower branch depositor loan
Views
Stored Procedures
Functions
sys
Query 1 | bankaccount bankcustomer borrower branch depositor loan
99 )
100 *
101 • insert into borrower values('avinash',1);
102 • insert into borrower values('dinesh',2);
103 • insert into borrower values('mohan',3);
104 • insert into borrower values('nikhil',4);
105 • insert into borrower values('ravi',5);
106
107
108 • select customer_name from bankCustomer
109 where customer_name=ANY(select customer_name from bankCustomer where customer_city='delhi');
110
111 *
112 • select distinct b.customer_name from borrower b, depositor d
113 where b.customer_name NOT IN(
114 select d.customer_name from loan l, depositor d, borrower b
115 where l.loan_no=b.loan_no and d.customer_name=b.customer_name
116 )
117
118 • select distinct d.customer_name from depositor d
119 • where d.customer_name IN(
120 select d.customer_name from branch br, depositor d, bankAccount ba
121 where br.branch_city='Bangalore' and br.branch_name=ba.branch_name

```

Table: borrower

Columns:

- customer_name varchar(2)
- loan_no int PK

Result Grid | Filter Rows: | Edit | Export | Wrap Cell Content: |

| customer_name |
|---------------|
| ravi |
| mohan |
| nikhil |
| dinesh |
| avinash |

bankCustomer 12 x

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|---|--|-----------------------|
| 40 | 10:57:29 | select branch_name, concat(accno/100000,'lac'),asset_in_lakhs from branch LIMIT 0, 1000 | 6 rows(s) returned | 0.000 sec / 0.000 sec |
| 41 | 10:59:31 | select d.customer_name as CUSTOMER_NAME from bankAccount d where b.branch_name=ba.residencyRoad and b.acccno=accno group by branch_name | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 42 | 10:59:45 | create view sum_of_loan as select branch_name, sum(balance) from bankAccount group by branch_name | Error Code: 1050. Table 'sum_of_loan' already exists | 0.000 sec |
| 43 | 11:00:33 | select sum_of_loan from sum_of_loan LIMIT 0, 1000 | 5 rows(s) returned | 0.000 sec / 0.000 sec |
| 44 | 11:00:49 | create view sum_of_loan as select branch_name, sum(balance) from bankAccount group by branch_name | Error Code: 1050. Table 'sum_of_loan' already exists | 0.000 sec |
| 45 | 11:02:20 | select customer_name from bankCustomer where customer_name=ANY(select customer_name from bankCustomer where customer_city='delhi') | 2 rows(s) returned | 0.000 sec / 0.000 sec |

Activate Windows
Go to Settings to activate.

ENG IN 11:02 29-11-2022

```

select distinct b.customer_name from borrower b, depositor d
where b.customer_name NOT IN(
select d.customer_name from loan l, depositor d, borrower b
where l.loan_no=b.loan_no and d.customer_name=b.Customer_name
);

```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas
Schemas
1bm21cs052_bank1
Tables bankaccount bankcustomer borrower branch depositor loan
Views
Stored Procedures
Functions
sys
Query 1 | bankaccount bankcustomer borrower branch depositor loan
99 )
100 *
101 • insert into borrower values('avinash',1);
102 • insert into borrower values('dinesh',2);
103 • insert into borrower values('mohan',3);
104 • insert into borrower values('nikhil',4);
105 • insert into borrower values('ravi',5);
106
107
108 • select customer_name from bankCustomer
109 where customer_name=ANY(select customer_name from bankCustomer where customer_city='delhi');
110
111 *
112 • select distinct b.customer_name from borrower b, depositor d
113 where b.customer_name NOT IN(
114 select d.customer_name from loan l, depositor d, borrower b
115 where l.loan_no=b.loan_no and d.customer_name=b.customer_name
116 )
117
118 • select distinct d.customer_name from depositor d
119 • where d.customer_name IN(
120 select d.customer_name from branch br, depositor d, bankAccount ba
121 where br.branch_city='Bangalore' and br.branch_name=ba.branch_name

```

Table: borrower

Columns:

- customer_name varchar(2)
- loan_no int PK

Result Grid | Filter Rows: | Edit | Export | Wrap Cell Content: |

| customer_name |
|---------------|
| mohan |

bankCustomer 13 x

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|---|--|-----------------------|
| 41 | 10:59:31 | select d.customer_name as CUSTOMER_NAME from bankAccount d where b.branch_name=ba.residencyRoad and b.acccno=accno group by branch_name | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 42 | 10:59:45 | create view sum_of_loan as select branch_name, sum(balance) from bankAccount group by branch_name | Error Code: 1050. Table 'sum_of_loan' already exists | 0.000 sec |
| 43 | 11:00:33 | select sum_of_loan from sum_of_loan LIMIT 0, 1000 | 5 rows(s) returned | 0.000 sec / 0.000 sec |
| 44 | 11:00:49 | create view sum_of_loan as select branch_name, sum(balance) from bankAccount group by branch_name | Error Code: 1050. Table 'sum_of_loan' already exists | 0.000 sec |
| 45 | 11:02:20 | select customer_name from bankCustomer where customer_name=ANY(select customer_name from bankCustomer where customer_city='delhi') | 2 rows(s) returned | 0.000 sec / 0.000 sec |
| 46 | 11:02:38 | select distinct d.customer_name from depositor d, depositor d where b.customer_name from loan l, depositor d, customer_name from borrower | 1 row(s) returned | 0.000 sec / 0.000 sec |

Activate Windows
Go to Settings to activate.

ENG IN 11:02 29-11-2022

```

select distinct d.customer_name from depositor d
where d.customer_name IN(
select d.customer_name from branch br, depositor d, bankAccount ba
where br.branch_city='Bangalore' and br.branch_name=ba.branch_name
and ba.accno=d.accno and customer_name IN(
select customer_name from borrower)

```

);

```
MySQL Workbench - Local instance MySQL80 - Query1 - 1m021ca052_bank1 - 1000 rows - Limit to 1000 rows - Filter Rows - Wrap Cell Content -
```

```
188 • select customer_name from bankCustomer
189   where customer_name NOT IN(select customer_name from bankCustomer where customer_city='delhi')
190
191 • select distinct b.customer_name from borrower b, depositor d
192   where b.customer_name NOT IN(
193     select d.customer_name from depositor d, borrower b
194     where l.loan_no=b.loan_no and d.customer_name=b.customer_name
195   )
196
197 • select distinct d.customer_name from depositor d
198   where d.customer_name IN(
199     select d.customer_name from branch br,depositor d, bankAccount ba
200     where br.branch_city='bangalore' and br.branch_name=ba.branch_name
201     and ba.acco=d.acco and customer_name IN(
202       select customer_name from borrower)
203   )
204
205
206 • select b.branch_name from branch b
207   where b.assets> ALL (
208     select SUM(b.assets) from branch b
209     where b.branch_city='bangalore')
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
```

Result Grid | Filter Rows | Export | Wrap Cell Content |

Table: borrower

Columns:

- customer_name varchar(2)
- loan_no int(11) PK
- etc PK

Output

| customer_name |
|---------------|
| bangalore |
| dinesh |

depositor 14 x

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|--|-----------------------|
| 42 | 10:59:45 | create view sum_of_loan as select branch_name, sum(balance) from bankAccount group by branch_name | Error Code: 1050. Table 'sum_of_loan' already exists | 0.000 sec |
| 43 | 11:00:33 | select from sum_of_loan LIMIT 0, 1000 | 5 rows(s) returned | 0.000 sec / 0.000 sec |
| 44 | 11:00:49 | create view sum_of_loan as select branch_name, sum(balance) from bankAccount group by branch_name | Error Code: 1050. Table 'sum_of_loan' already exists | 0.000 sec |
| 45 | 11:02:20 | select customer_name from bankCustomer where customer_name NOT IN(select customer_name from bankCustomer where customer_city='delhi') | 2 rows(s) returned | 0.000 sec / 0.000 sec |
| 46 | 11:02:38 | select distinct d.customer_name from borrower b, depositor d where b.customer_name NOT IN(select d.customer_name from loan l,depositor d, borrower b where l.loan_no=b.loan_no and d.customer_name=b.customer_name) | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 47 | 11:03:02 | select distinct d.customer_name from depositor d where d.customer_name IN(select c.customer_name from branch b, depositor d, bankAccount ba where b.branch_name=d.branch_name and ba.acco=d.acco and c.customer_name=ba.branch_name) | 2 rows(s) returned | 0.000 sec / 0.000 sec |

Activate Windows
Go to Settings to activate!

ENGLISH IN 29-11-2022

```
select b.branch_name from branch b
where b.assets> ALL (
select SUM(b.assets) from branch b
where b.branch_City='bangalore' );
```

```
MySQL Workbench - Local instance MySQL80 - Query1 - 1m021ca052_bank1 - 1000 rows - Limit to 1000 rows - Filter Rows - Wrap Cell Content -
```

```
116
117 • select distinct d.customer_name from depositor d
118   where d.customer_name IN(
119     select d.customer_name from branch br,depositor d, bankAccount ba
120     where br.branch_city='bangalore' and br.branch_name=ba.branch_name
121     and ba.acco=d.acco and customer_name IN(
122       select customer_name from borrower)
123   )
124
125
126 • select b.branch_name from branch b
127   where b.assets> ALL (
128     select SUM(b.assets) from branch b
129     where b.branch_City='bangalore')
130
131 • delete ba.* from bankAccount ba, branch b where branch_city='Bombay' and ba.branch_name=b.branch_name;
132 • select * from bankAccount;
133
134 • UPDATE bankAccount set balance=(balance + (balance*0.05));
135
136
137
```

Result Grid | Filter Rows | Export | Wrap Cell Content |

Table: bankAccount

Columns:

- branch_name varchar(2)
- balance float
- branch_pk int(11) PK
- branch_name varchar(2) INT PK

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|--|-----------------------|
| 43 | 11:00:33 | select from sum_of_loan LIMIT 0, 1000 | 5 rows(s) returned | 0.000 sec / 0.000 sec |
| 44 | 11:00:49 | select from sum_of_loan LIMIT 0, 1000 | Error Code: 1050. Table 'sum_of_loan' already exists | 0.000 sec |
| 45 | 11:02:20 | select customer_name from bankCustomer where customer_name NOT IN(select customer_name from bankCustomer where customer_city='delhi') | 2 rows(s) returned | 0.000 sec / 0.000 sec |
| 46 | 11:02:38 | select distinct d.customer_name from borrower b, depositor d where b.customer_name NOT IN(select d.customer_name from loan l,depositor d, borrower b where l.loan_no=b.loan_no and d.customer_name=b.customer_name) | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 47 | 11:03:02 | select distinct d.customer_name from depositor d where d.customer_name IN(select c.customer_name from branch b, depositor d, bankAccount ba where b.branch_name=d.branch_name and ba.acco=d.acco and c.customer_name=ba.branch_name) | 2 rows(s) returned | 0.000 sec / 0.000 sec |
| 48 | 11:03:43 | select b.branch_name from branch b where b.assets> ALL (select SUM(b.assets) from branch b where b.branch_City='bangalore') LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |

Activate Windows
Go to Settings to activate!

ENGLISH IN 29-11-2022

```
delete ba.* from bankAccount ba, branch b where branch_city='Bombay' and
ba.branch_name=b.branch_name;
select * from bankAccount;
```

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

1bm21cs052_bank1

Tables bankaccount bankcustomer borrower branch depositor loan

Views

Stored Procedures

Functions

Schemas

Filter objects

Query 1

```

116
117
118 • select distinct d.customer_name from depositor d
119 where d.customer_name IN(
120 select d.customer_name from branch br.depositor d, bankAccount ba
121 where br.branch_city='Bangalore' and br.branch_name=ba.branch_name
122 and ba.acco_no=d.acco_no and customer_name IN(
123 select customer_name from borrower)
124 )
125
126 • select b.branch_name from branch b
127 where b.assets> ALL (
128 select sum(b.assets) from branch b
129 where b.branch_city='Bangalore')
130
131 • delete ba.* from bankAccount ba, branch b where branch_city='Bombay' and ba.branch_name=b.branch_name
132 • select * from bankAccount
133
134 • UPDATE bankAccount set balance=(balance + (balance*0.05));
135
136
137

```

Administration Schemas Information

Table: borrower

Columns:

- customer_name varchar(2)
- loan_no int PK

Result Grid | Filter Rows: | Edit | Export/Import | Wrap Cell Content: |

| acco | branch_name | balance |
|------|------------------|-----------|
| 1 | db_champet | 2111.25 |
| 2 | db_mahim Road | 3788.125 |
| 4 | db_parlamentRoad | 10418.625 |
| 5 | db_jantaManta | 9261 |
| 6 | db_residencyRoad | 4630.5 |
| 9 | db_parlamentRoad | 3472.875 |

bankAccount 17 >

Output

Action Output

```

# Time Action Message Duration / Fetch
45 11-03-20 select customer_name from bankCustomer where customer_name=ANY(select customer_name from bankCustomer where customer_name='dell') LIMIT 2 rows returned 0.000 sec / 0.000 sec
46 11-03-28 select distinct d.customer_name from borrower b, depositor d where b.customer_name NOT IN(select d.customer_name from loan.lender d, borrower b where b.customer_name=d.customer_name) LIMIT 1 rows returned 0.000 sec / 0.000 sec
47 11-03-02 select distinct d.customer_name from depositor d where d.customer_name IN(select d.customer_name from branch br.depositor d, bankAccount ba where br.branch_name=ba.branch_name) LIMIT 1 rows returned 0.000 sec / 0.000 sec
48 11-03-43 select b.branch_name from branch b where b.assets> ALL (select sum(b.assets) from branch b where b.branch_city='Bangalore') LIMIT 0, 1000 1 rows returned 0.000 sec / 0.000 sec
49 11-03-59 delete ba.* from bankAccount ba, branch b where branch_city='Bombay' and ba.branch_name=b.branch_name 0 rows affected 0.000 sec
50 11-03-59 select * from bankAccount 9 rows returned 0.000 sec / 0.000 sec
50 11-03-59 UPDATE bankAccount set balance=(balance + (balance*0.05)); 0.000 sec / 0.000 sec
51 11-07-13 delete from branch where branch_city='bangalore' 0 rows affected 0.000 sec / 0.000 sec
52 11-07-23 select * from branch LIMIT 0, 1000 6 rows returned 0.000 sec / 0.000 sec

```

Activate Windows Go to Settings to activate Windows 0.000 sec / 0.000 sec

Object Info Session

24°C Partly sunny

1104 29-11-2022

UPDATE bankAccount set balance=(balance + (balance*0.05));

delete from branch where branch_city='bangalore';
 select * from branch;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

1bm21cs052_bank1

Tables bankaccount bankcustomer borrower branch depositor loan

Views

Stored Procedures

Functions

Schemas

Filter objects

Query 1

```

117
118 • select distinct d.customer_name from depositor d
119 where d.customer_name IN(
120 select d.customer_name from branch br.depositor d, bankAccount ba
121 where br.branch_city='Bangalore' and br.branch_name=ba.branch_name
122 and ba.acco_no=d.acco_no and customer_name IN(
123 select customer_name from borrower)
124 )
125
126 • select b.branch_name from branch b
127 where b.assets> ALL (
128 select sum(b.assets) from branch b
129 where b.branch_city='Bangalore')
130
131 • delete ba.* from bankAccount ba, branch b where branch_city='Bombay' and ba.branch_name=b.branch_name
132 • select * from bankAccount
133
134 • UPDATE bankAccount set balance=(balance + (balance*0.05));
135 • delete from branch where branch_city='bangalore'
136 • select * from branch
137

```

Administration Schemas Information

Table: borrower

Columns:

- customer_name varchar(2)
- loan_no int PK

Result Grid | Filter Rows: | Edit | Export/Import | Wrap Cell Content: |

| branch_name | branch_city | assets |
|------------------|-------------|--------|
| db_champet | bangalore | 50000 |
| db_jantaManta | dells | 20000 |
| db_mahim Road | dells | 20000 |
| db_residencyRoad | bangalore | 10000 |
| db_zeddenRoad | bangalore | 10000 |
| db_shivajiRoad | bombay | 20000 |

branch 17 >

Output

Action Output

```

# Time Action Message Duration / Fetch
48 11-03-43 select b.branch_name from branch b where b.assets> ALL (select sum(b.assets) from branch b where b.branch_city='Bangalore') LIMIT 0, 1000 1 rows returned 0.000 sec / 0.000 sec
49 11-03-59 delete ba.* from bankAccount ba, branch b where branch_city='Bombay' and ba.branch_name=b.branch_name 0 rows affected 0.000 sec
50 11-03-59 select * from bankAccount 9 rows returned 0.000 sec / 0.000 sec
50 11-03-59 UPDATE bankAccount set balance=(balance + (balance*0.05)); 0 rows affected Rows matched: 3 Changed: 3 Warnings: 0 0.000 sec / 0.000 sec
51 11-07-13 delete from branch where branch_city='bangalore' Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('1bm21cs052_bank1', 'borrower', CONSTRAINT: borrower_branch_name_fkey) 0.000 sec / 0.000 sec
52 11-07-23 select * from branch LIMIT 0, 1000 6 rows returned 0.000 sec / 0.000 sec

```

Activate Windows Go to Settings to activate Windows 0.000 sec / 0.000 sec

Object Info Session

24°C Partly sunny

1107 29-11-2022

Week 5

Program 3:EMPLOYEE DATABASE

Schema Diagram

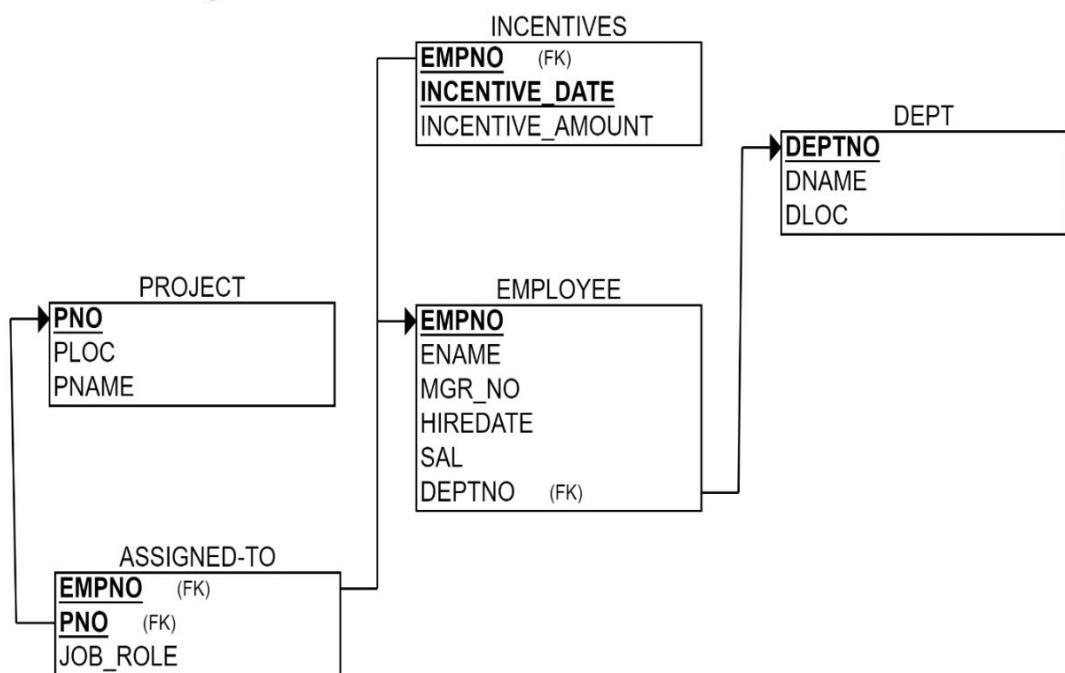


Table Creation:

```
create database employee;
use employee;
create table dept
( deptno int, dname varchar(50),
dloc varchar(50),
primary key(deptno));

create table employee
( empno int, ename varchar(50), mgrno int, hiredate date, sal int, deptno int,
primary key(empno), foreign key(deptno) references dept(deptno)
on update cascade on delete cascade);

create table incentive
( empno int, incentivedate date, incentiveamount int,
primary key(incentivedate), foreign key(empno) references employee(empno)
on update cascade on delete cascade);

create table project
(pno int, ploc varchar(50), pname varchar(50),
primary key(pno));

create table assignedto
( empno int,pno int, jobrole varchar(50),
foreign key(empno) references employee(empno),
foreign key(pno) references project(pno)
on update cascade on delete cascade);
```

| | Field | Type | Null | Key | Default | Extra |
|---|----------|----------|------|-----|---------|-------|
| ▶ | empno | int | NO | PRI | NULL | |
| | pno | int | NO | PRI | NULL | |
| | job_role | char(25) | YES | | NULL | |

| | Field | Type | Null | Key | Default | Extra |
|---|--------|----------|------|-----|---------|-------|
| ▶ | deptno | int | NO | PRI | NULL | |
| | dname | char(25) | YES | | NULL | |
| | dloc | char(25) | YES | | NULL | |

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|----------|-------------|------|-----|---------|-------|
| ▶ | empno | int | NO | PRI | NULL | |
| | ename | char(25) | YES | | NULL | |
| | mgrno | int | YES | | NULL | |
| | hiredate | varchar(10) | YES | | NULL | |
| | sal | int | YES | | NULL | |
| | deptno | int | NO | PRI | NULL | |

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|------------------|-------------|------|-----|---------|-------|
| ▶ | empno | int | NO | PRI | NULL | |
| | incentive_date | varchar(10) | YES | | NULL | |
| | incentive_amount | int | YES | | NULL | |

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|-------|----------|------|-----|---------|-------|
| ▶ | pno | int | NO | PRI | NULL | |
| | ploc | char(25) | YES | | NULL | |
| | pname | char(25) | YES | | NULL | |

Values:

```
insert into dept values(1,'CSE','Siddapur');
```

```
insert into dept values(2,'ISE','Sirsi');
```

```
insert into dept values(3,'ECE','Pune');
```

```
insert into dept values(4,'EEE','Hubli');
```

```
insert into dept values(5,'IM','Mumbai');
```

```
insert into dept values(6,'BT','Goa');
```

```
insert into emp values(1,'Akash',7,'2020-01-12',20000,4);
insert into emp values(2,'Bhuvan',8,'2020-03-14',40000,1);
insert into emp values(3,'Chris',9,'2021-02-25',35000,2);
insert into emp values(4,'Kiran',10,'2021-11-30',60000,6);
insert into emp values(5,'Pavan',11,'2022-3-10',52000,3);
insert into emp values(6,'Santosh',12,'2022-5-24',48000,5);
select *from emp;
```

```
insert into incentives values(1,'2021-01-12',00000);
insert into incentives values(2,'2021-12-12',8000);
insert into incentives values(3,'2022-06-09',6500);
insert into incentives values(4,'2022-11-29',0000);
insert into incentives values(5,'2023-05-30',9800);
insert into incentives values(6,'2023-08-17',7000);
select *from incentives;
```

```
insert into project values(1,'pro1','Bengaluru');
insert into project values(2,'pro2','Pune');
insert into project values(3,'pro3','Hyderabad');
insert into project values(4,'pro4','Kerala');
insert into project values(5,'pro5','Mysuru');
insert into project values(6,'pro6','Sirsi');
select *from project;
```

```
insert into assigned_to values(1,1,'Developer');
insert into assigned_to values(2,2,'Assistant');
insert into assigned_to values(3,3,'Co-assistant');
insert into assigned_to values(4,4,'Organiser');
insert into assigned_to values(5,5,'Manager');
insert into assigned_to values(6,6,'Supervisor');
select *from assigned_to;
```

Queries:

- Retrieve the employee numbers of all employees who work on project located in Bangalore, Hyderabad, or Mysore

```
select emp_no from assigned_to a where p_no in (select p_no from project where p_loc in ('Bengaluru','Hyderabad','Mysuru'));
```

| Result Grid | | Filter |
|-------------|-------|--------|
| | empno | |
| ▶ | 1 | |
| | 2 | |
| | 3 | |
| | 4 | |
| | 5 | |
| | 6 | |

- Get Employee ID's of those employees who didn't receive incentives

```
select emp_no,e_name from emp where emp_no in (select emp_no from incentives where incentive_amount=0);
```

| Result Grid | | Filter Rows: |
|-------------|-------|--------------|
| | empno | |
| ▶ | 2 | |
| | 4 | |
| | 6 | |
| * | NULL | |

- Write a SQL query to find the employees name, number, dept, job_role, department location and project location who are working for a project location same as his/her department location.

```
select e.e_name, e.emp_no,d.d_name,a.job_role,d.d_loc,p.p_loc from emp e,dept d,assigned_to a,project p where d.d_loc=p.p_loc and e.emp_no=a.emp_no and a.p_no=p.p_no;
```

| | ename | empno | dname | jobrole | dloc | ploc |
|---|---------|-------|------------|----------------|-----------|-----------|
| ▶ | Avinash | 1 | Admin | Administration | Banglore | Banglore |
| | Chandan | 4 | Marketing | Advertising | Mysore | Mysore |
| | Aravind | 5 | Shipping | Transporting | Hyderabad | Hyderabad |
| | Amal | 6 | Purchasing | Purchasing | Mysore | Mysore |

- Find the employee name, dept name and job_role of an employee who received max incentive in year 2005

```
select e.e_name,max(i.incentive_amount),d.d_name,a.job_role from emp e,dept d,assigned_to a,incentives i where e.emp_no in(select emp_no from incentives where incentive_amount= (select max(incentive_amount) from incentives where incentive_date between '2022-01-01' and '2022-12-31')) and d.dept_no=e.dept_no and a.emp_no=e.emp_no;
```

| Result Grid | | | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|---------|------------|----------------|---------------|---------|--------------------|
| | ename | dname | jobrole | max_incentive | | |
| ▶ | Avinash | Purchasing | Administration | 55000 | | |

More Queries on Employee Database

week 6

TO DO

1.Using Scheme diagram, Create tables by properly specifying the primary keys and the foreign keys.

```
create database 1bm21cs052_employee2;
```

```
use 1bm21cs052_employee2;
```

```
create table dept(  
deptno int,  
dname varchar(20),  
dloc varchar(20),  
primary key(deptno)  
);  
create table employee(  
empno int,  
ename varchar(20),  
mgr_no int,  
hiredate date,  
sal double,  
deptno int,  
primary key(empno),  
foreign key (deptno) references dept(deptno)  
on delete cascade  
on update cascade  
);  
create table incentives(  
empno int,  
incentive_date date,  
incentive_amount float,  
primary key(empno,incentive_date),  
foreign key (empno) references employee(empno)  
on delete cascade  
on update cascade  
);  
create table project(  
pno int,  
ploc varchar(20),  
pname varchar(20),  
primary key(pno)  
);  
create table assigned_to(  
empno int,  
pno int,  
job_role varchar(20),
```

```
primary key(empno,pno),
foreign key (empno) references employee(empno),
foreign key (pno) references project(pno)
on delete cascade
on update cascade
);
```

2. Enter greater than five tuples for each table.

```
insert into dept values(10,'cse','bangalore');
insert into dept values(20,'ise','bangalore');
insert into dept values(30,'aiml','hyderabad');
insert into dept values(40,'ece','mysore');
insert into dept values(50,'eee','delhi');
insert into dept values(60,'iem','chennai');
```

```
insert into employee values(11,'Rajesh',21,'2000-04-03',80000,10);
insert into employee values(12,'Ajay',11,'2003-04-06',70000,20);
insert into employee values(13,'Divya',11,'2006-03-07',60000,30);
insert into employee values(14,'Chandan',12,'2007-09-03',50000,40);
insert into employee values(15,'Bhavesh',13,'2009-11-13',40000,50);
insert into employee values(16,'Tarun',14,'2012-02-10',30000,60);
insert into employee values(17,'Brinda',11,'2009-05-12',50000,10);
insert into employee values(18,'Anil',15,'2015-01-01',30000,20);
insert into employee values(19,'Puja',15,'2020-10-21',60000,30);
insert into employee values(20,'Ram',16,'2021-09-17',45000,40);
insert into employee values(21,'Priya',22,'2002-03-13',85000,10);
```

```
insert into incentives values(11,'2012-09-08',40000);
insert into incentives values(12,'2015-07-10',33000);
insert into incentives values(13,'2019-01-21',7000);
insert into incentives values(14,'2019-01-05',8000);
insert into incentives values(15,'2019-01-13',5000);
insert into incentives values(17,'2021-03-17',6000);
insert into incentives values(18,'2021-04-16',8000);
insert into incentives values(19,'2021-08-11',9000);
```

```
insert into project values(121,'bangalore','proj1');
insert into project values(122,'bangalore','proj2');
insert into project values(123,'mysore','proj3');
insert into project values(124,'hyderabad','proj4');
insert into project values(125,'delhi','proj5');
insert into project values(126,'mumbai','proj6');
insert into project values(127,'calicut','proj7');
insert into project values(128,'calicut','proj8');
```

```
insert into assigned_to values(11,121,'manager');
```

```

insert into assigned_to values(12,122,'team_lead');
insert into assigned_to values(13,123,'analyst');
insert into assigned_to values(14,124,'team_lead');
insert into assigned_to values(15,125,'manager');
insert into assigned_to values(16,126,'programmer');
insert into assigned_to values(17,127,'team_lead');
insert into assigned_to values(19,128,'team_lead');

```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the schema browser with the 'employee' table selected. The main area shows a results grid with 21 rows of employee data. The output pane below shows the executed SQL statement and its execution details.

| empno | ename | mgr_no | hiredate | sal | deptno |
|-------|---------|--------|------------|-------|--------|
| 11 | Rajesh | 21 | 2000-04-03 | 8000 | 10 |
| 12 | Ajay | 11 | 2003-04-06 | 7000 | 20 |
| 13 | Dixy | 11 | 2006-03-07 | 6000 | 30 |
| 14 | Chandan | 12 | 2007-09-03 | 50000 | 40 |
| 15 | Thavesh | 13 | 2009-11-13 | 40000 | 50 |
| 16 | Tarun | 14 | 2012-02-10 | 30000 | 60 |
| 17 | Binda | 11 | 2009-05-12 | 50000 | 10 |
| 18 | Shreya | 15 | 2013-05-01 | 50000 | 10 |
| 19 | Puja | 15 | 2020-10-21 | 60000 | 30 |
| 20 | Ram | 16 | 2021-09-17 | 45000 | 40 |
| 21 | Priva | 22 | 2002-02-13 | 85000 | 10 |

3. List the name of the managers with the maximum employees

```

select emp.ename
from employee emp
where emp.empno=(  

select mgr_no
from employee e
group by mgr_no
having count(empno) >= all(
select (count(empno))
from employee
group by mgr_no ));

```

The screenshot shows the Oracle SQL Developer interface with the code editor open. The code implements a self-join on the employee table to find managers with the maximum number of employees.

```

92 • insert into assigned_to values(19,128,'team_lead');
93
94 • select emp.ename
95   from employee emp
96   where emp.empno=(  

97     select mgr_no
98       from employee e
99       group by mgr_no
100      having count(empno) >= all(
101        select (count(empno))
102          from employee
103          group by mgr_no));

```

4. Display those managers name whose salary is more than average salary of his employee.

```

select emp.ename
from employee emp
where emp.sal > any (
select avg(e.sal)

```

```
from employee e
where emp.empno=e.mgr_no
);
```

The screenshot shows the IBM DB2 Command Line Processor interface. The left pane displays the database schema with tables like employee, assigned_to, dept, emp, incentives, project, and views. The right pane shows the command window with the following SQL code:

```
101  naving count(empno) >= all(
102   select (count(empno))
103   from employee
104   group by mgr_no );
105  select emp.ename
```

The result grid shows the names of managers who have at least two employees:

| ename |
|---------|
| Rajesh |
| Ajay |
| Divya |
| Chandan |
| Priya |

The output pane below shows the history of the session with log entries:

| # | Action | Time | Message | Duration / Fetch |
|----|---|----------|--------------------|-----------------------|
| 50 | SELECT * FROM ibm21cs052_employee2.dept | 18:44:11 | 0 rows returned | 0.000 sec / 0.000 sec |
| 51 | SELECT * FROM ibm21cs052_employee2.employee | 18:44:13 | 11 row(s) returned | 0.000 sec / 0.000 sec |
| 52 | SELECT * FROM ibm21cs052_employee2.incentives | 18:44:15 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 53 | SELECT * FROM ibm21cs052_employee2.project | 18:44:17 | 0 row(s) returned | 0.000 sec / 0.000 sec |

5. Find the name of the second top level managers of each department

```
select emp.ename
from employee emp
where emp.ename = any(
select e2.ename
from employee e, employee e2
where e2.empno=e.mgr_no and e2.deptno = e.deptno and e.ename = any(
select e1.ename
from employee e1, employee e0
where e1.empno=e0.mgr_no and e1.deptno = e0.deptno
group by e1.mgr_no
having count(e1.empno)>1
));
```

The screenshot shows the IBM DB2 Command Line Processor interface. The right pane shows the command window with the same complex SQL query as the previous screenshot. The result grid shows the name of the second top level manager for each department:

| ename |
|-------|
| Arun |

6. Find the employee details who got second maximum incentive in January 2019.

```
select i.empno, i.incentive_date, max(i.incentive_amount)second_max
```

```

from incentives i
where i.incentive_date between '2019-01-01' and '2019-01-31' and i.incentive_amount not
in(
select max(incentive_amount)
from incentives
where incentive_date between '2019-01-01' and '2019-01-31');

```

The screenshot shows the IBM DB2 SQL Workbench interface. The left pane displays the database schema for 'ibm21cs052_employee2', specifically the 'Tables' section which includes 'assigned_to', 'dept', 'employee', 'incentives', and 'project'. The right pane shows the SQL editor with the following query:

```

113 • select emp.ename
  from employee emp
114   where emp.ename = any(
115     select e2.ename
      from employee e1, employee e2
116       where e1.empno=e2.mgr_no and e1.deptno = e2.deptno and e1.ename = any(
117         select e1.ename
          from employee e1, employee e0
118           where e1.empno=e0.mgr_no and e1.deptno = e0.deptno
119             group by e1.mgr_no
120               having count(e1.empno)>1
121             );
122
123
124
125
126 • select i.empno, i.incentive_date, max(i.incentive_amount)second_max

```

The results grid shows one row:

| empno | incentive_date | second_max |
|-------|----------------|------------|
| 13 | 2019-01-21 | 7000 |

The status bar at the bottom indicates the query was executed in 0.000 sec / 0.000 sec.

7. Display those employees who are working in the same department where his manager is working.

```

select e.ename, e.deptno
from employee e, employee e2
where e2.empno=e.mgr_no and e2.deptno = e.deptno;

```

The screenshot shows the IBM DB2 SQL Workbench interface. The left pane displays the database schema for 'ibm21cs052_employee2', specifically the 'Tables' section which includes 'assigned_to', 'dept', 'employee', 'incentives', and 'project'. The right pane shows the SQL editor with the following query:

```

119 • select el.ename
  from employee el, employee e0
120   where el.empno=e0.mgr_no and el.deptno = e0.deptno
121     group by el.mgr_no
122       having count(el.empno)>1
123
124
125
126 • select i.empno, i.incentive_date, max(i.incentive_amount)second_max
  from incentives i
127   where i.incentive_date between '2019-01-01' and '2019-01-31' and i.incentive_amount not in(
128     select max(incentive_amount)
      from incentives
129       where incentive_date between '2019-01-01' and '2019-01-31');
130
131
132
133 • select e.ename, e.deptno

```

The results grid shows two rows:

| ename | deptno |
|--------|--------|
| Rajesh | 10 |
| Brida | 10 |

Spot query-Find the employee details who got third maximum incentive in January 2019

```

select i.empno, i.incentive_amount
from incentives i
where 3 = (
select count(*)
from incentives j
where incentive_date between '2019-01-01' and '2019-01-31' and i.incentive_amount <=

```

j.incentive_amount)
and incentive_date between '2019-01-01' and '2019-01-31';

The screenshot shows the Oracle SQL Developer interface with a query editor window open. The code in the editor is as follows:

```
126 • select i.empno, i.incentive_date, max(i.incentive_amount)second_max
  from incentives i
127
128 • where i.incentive_date between '2019-01-01' and '2019-01-31' and i.incentive_amount not in(
129   select max(incentive_amount)
130   from incentives
131   where incentive_date between '2019-01-01' and '2019-01-31');
132
133 • select e.ename, e.deptno
  from employee e, employee e2
134
135 where e2.empno=e.mgr_no and e2.deptno = e.deptno;
136
137
138 • select i.empno, i.incentive_amount
  . . .
<--
```

Below the code, there is a Result Grid with the following data:

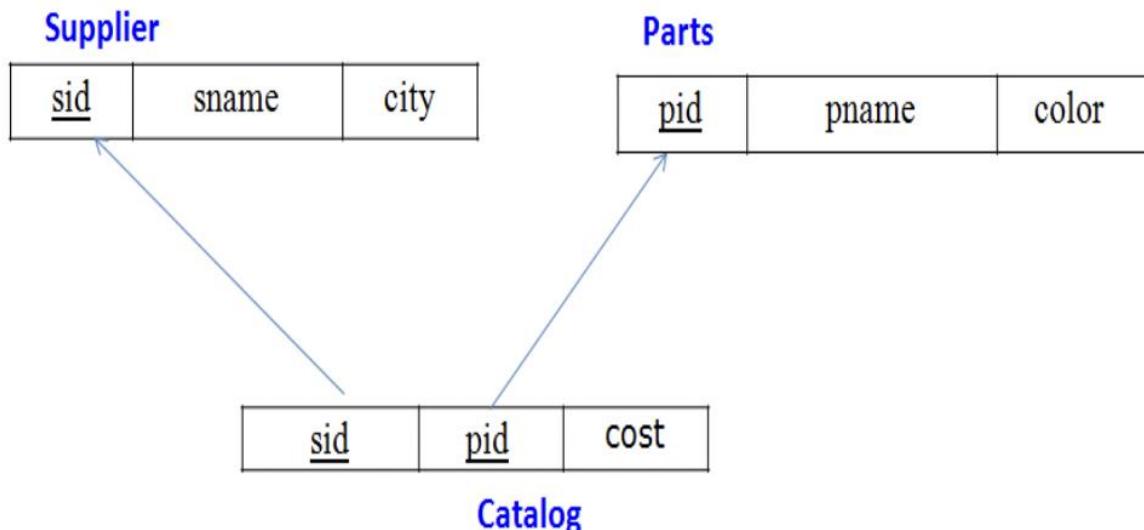
| empno | incentive_amount |
|-------|------------------|
| 15 | 5000 |

At the bottom of the interface, there are several tabs: Administration, Schemas, Information, and another Administration tab. The Schemas tab is currently selected.

WEEK 7

Program 5: SUPPLIER

Schema Diagram



Creating tables

```
create table supplier
(sid int, sname varchar(50), city varchar(50),
primary key(sid));
```

```
create table parts
(pid int, pname varchar(50), colour varchar(50),
primary key(pid));
```

```
create table catalog
(sid int, pid int, cost int,
foreign key(sid) references supplier(sid),
foreign key(pid) references parts(pid));
```

| | Field | Type | Null | Key | Default | Extra |
|---|-------|----------|------|-----|---------|-------|
| ▶ | sid | int | NO | PRI | NULL | |
| | sname | char(25) | YES | | NULL | |
| | city | char(25) | YES | | NULL | |

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|-------|----------|------|-----|---------|-------|
| ▶ | pid | int | NO | PRI | NULL | |
| | pname | char(15) | YES | | NULL | |
| | color | char(15) | YES | | NULL | |

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|-------|------|------|-----|---------|-------|
| ▶ | sid | int | NO | PRI | NULL | |
| | pid | int | NO | PRI | NULL | |
| | cost | int | YES | | NULL | |

Values:

```
insert into supplier values(10001,'acme widget','bangalore'),
insert into supplier values (10002,'johns','kolkata'),
insert into supplier values (10003,'vimal','mumbai'),
insert into supplier values (10004,'reliance','delhi');
select * from supplier;
```

```
insert into parts values(20001,'book','red'),
insert into parts values (20002,'pen','red'),
insert into parts values (20003,'pencil','green'),
insert into parts values (20004,'mobile','green'),
insert into parts values (200 05,'charger','black');
select * from parts;
```

```
insert into catalog values(10001,20001,10),
insert into catalog values (10001,20002,10),
insert into catalog values (10001,20003,30),
insert into catalog values (10001,20004,10),
insert into catalog values (10001,20005, 10),
insert into catalog values (10002,20001,10),
insert into catalog values (10002,20002,20);
insert into catalog values(10003,20003,30),
insert into catalog values (10004,20003,40);
select * from catalog;
```

Queries:

```
select * from supplier;
select * from parts;
```

| | sid | sname | city |
|---|-------|-------------|-----------|
| ▶ | 10001 | acme widget | bangalore |
| | 10002 | johns | kolkata |
| | 10003 | vimal | mumbai |
| | 10004 | reliance | delhi |
| * | NULL | NULL | NULL |

| | pid | pname | colour |
|---|-------|---------|--------|
| ▶ | 20001 | book | red |
| | 20002 | pen | red |
| | 20003 | pencil | green |
| | 20004 | mobile | green |
| | 20005 | charger | black |
| * | NULL | NULL | NULL |

| | sid | pid | cost |
|---|-------|-------|------|
| ▶ | 10001 | 20001 | 10 |
| | 10001 | 20002 | 10 |
| | 10001 | 20003 | 30 |
| | 10001 | 20004 | 10 |
| | 10001 | 20005 | 10 |
| | 10002 | 20001 | 10 |
| | 10002 | 20002 | 20 |
| | 10003 | 20003 | 30 |
| | 10004 | 20003 | 40 |

supplier 7 parts 8 catalog 9 X

select * from catalog;

- Find the pnames of parts for which there is some supplier.

select pname from parts where pid in (select pid from catalog)

| | pname |
|---|---------|
| ▶ | book |
| | pen |
| | pencil |
| | mobile |
| | charger |

- Find the snames of suppliers who supply every part.

select sname from

```
(select c.sname,count(distinct a.pid) as cnt from catalog a
```

```
left join parts b on a.pid=b.pid
```

```
left join supplier c on c.sid=a.sid group by 1) a
```

```
where cnt=(select count(distinct a.pid) from catalog a
```

```
left join parts b on a.pid=b.pid);
```

| | sname |
|---|-------------|
| ▶ | acme widget |

- Find the snames of suppliers who supply every red part.

select sname from supplier

```
where sid in( select sid from catalog where pid in( select pid from parts where
colour='red'));
```

| | sname |
|---|-------------|
| ▶ | acme widget |
| | johns |

- Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

select pname from parts

where pid in(select pid

from catalog where sid in(select sid from supplier

where sname='acme widget')) and pid not in(select pid

from catalog where sid in(select sid

from supplier where sname!='acme widget');

| | pname |
|---|---------|
| ▶ | mobile |
| | charger |

- Find the sids of suppliers who charge more for some part than the average cost of that part.

select c.sid from catalog c

where c.cost >(select avg(cc.cost) from catalog cc where c.pid=cc.pid group by cc.pid);

Result Grid

The screenshot shows a 'Result Grid' window from MySQL Workbench. The grid has a single column labeled 'sid'. It contains three rows: an empty header row, a row with value '10002', and a row with value '10004'. A large blue arrow icon is positioned to the left of the grid.

| | sid |
|---|-------|
| ▶ | 10002 |
| | 10004 |

- For each part, find the sname of the supplier who charges the most for that part.

```
select sname  
from supplier  
where sid in( select sid from catalog  
where cost in( select max(cost)  
from catalog  
group by pid));
```

| | sname |
|---|-------------|
| ▶ | acme widget |
| | johns |
| | reliance |

WEEK 8

Program 6 week 8:FLIGHT

FLIGHTS(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)

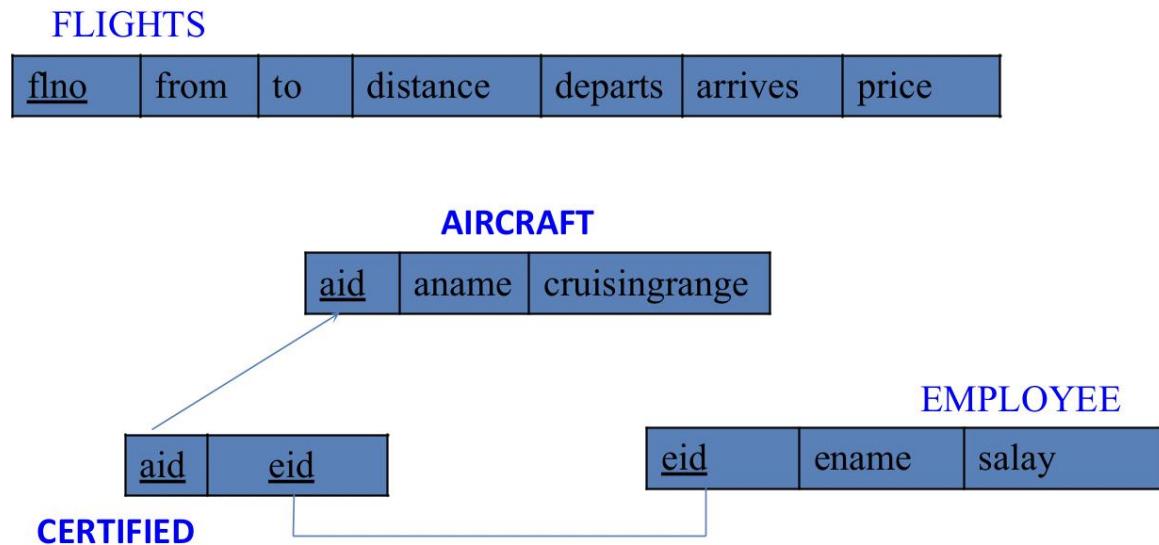
AIRCRAFT(aid: integer, fname: string, cruising_range: integer)

CERTIFIED(eid: integer, aid: integer)

EMPLOYEES(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well;
Every pilot is certified for some aircraft, and only pilots are certified to fly.

SCHEMADIAGRAM



DBMS WEEK 8 create database Airline;
use Airline;

create table flights(
flno int, ffrom varchar(50), tto varchar(50),
distance int, departs time,
arrives time, price int, primary key(flno));

```
create table aircraft(
aid int, aname varchar(50), cruisingrange int,
primary key(aid));
```

```
create table employee(
eid int, ename varchar(50), salary int,
primary key(eid));
```

```
create table certified( eid int, aid int,
foreign key(aid) references aircraft(aid) on update cascade on delete
cascade,
foreign key(eid) references employee(eid) on update cascade on delete
cascade);
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|---------------|----------|------|-----|---------|-------|
| ▶ | aid | int | NO | PRI | HULL | |
| | aname | char(25) | YES | | HULL | |
| | cruisingrange | int | YES | | HULL | |

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|-----------|----------|------|-----|---------|-------|
| ▶ | fno | int | NO | PRI | HULL | |
| | departure | char(25) | YES | | HULL | |
| | arrival | char(25) | YES | | HULL | |
| | distance | int | YES | | HULL | |
| | departs | time | YES | | HULL | |
| | arrives | time | YES | | HULL | |
| | price | int | YES | | HULL | |

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|--------|----------|------|-----|---------|-------|
| ▶ | eid | int | NO | PRI | HULL | |
| | ename | char(25) | YES | | HULL | |
| | salary | int | YES | | HULL | |

Result Grid | Filter Rows: Export: Wrap Cell Content:

| | Field | Type | Null | Key | Default | Extra |
|---|-------|------|------|-----|---------|-------|
| ▶ | eid | int | NO | PRI | HULL | |
| | aid | int | NO | PRI | HULL | |

```
insert into employee values (101,'Avinash',50000), (102,'Lokesh',60000),  
(103,'Rakesh',70000), (104,'Santhosh',82000), (105,'Tilak',5000);
```

```
select * from employee;
```

| | eid | ename | salary |
|---|------|----------|--------|
| ▶ | 101 | Avinash | 50000 |
| | 102 | Lokesh | 60000 |
| | 103 | Rakesh | 70000 |
| | 104 | Santhosh | 82000 |
| | 105 | Tilak | 5000 |
| * | NULL | NULL | NULL |

```
insert into aircraft values (1,'Airbus',2000), (2,'Boeing',700),  
(3,'JetAirways',550), (4,'Indigo',5000), (5,'Boeing',4500), (6,'Airbus',2200);
```

```
select * from aircraft;
```

| | aid | aname | cruising_range |
|---|------|------------|----------------|
| ▶ | 1 | Airbus | 2000 |
| | 2 | Boeing | 700 |
| | 3 | Jetairways | 550 |
| | 4 | Indigo | 5000 |
| | 5 | Boeing | 4500 |
| | 6 | Airbus | 2200 |
| * | NULL | NULL | NULL |

```
insert into certified values (101,2),(101,4),(101,5), (101,6),(102,1),(102,3),  
(102,5),(103,2),(103,3), (103,5),(103,6),(104,6), (104,1),(104,3),(105,3);
```

Result Grid | Filter Rows: _____

| | eid | aid |
|---|-----|-----|
| ▶ | 101 | 2 |
| | 101 | 4 |
| | 101 | 5 |
| | 101 | 6 |
| | 102 | 1 |
| | 102 | 3 |
| | 102 | 5 |
| | 103 | 2 |
| | 103 | 3 |
| | 103 | 5 |
| | 103 | 6 |
| | 104 | 6 |
| | 104 | 1 |
| | 104 | 3 |
| | 105 | 3 |

```
insert into flights values (1,'Banglore','New Delhi',500,'6:00','9:00',5000),
(2,'Banglore','Chennai',300,'7:00','8:30',3000), (3,'Trivandrum','New
Delhi',800,'8:00','11:30',6000),
(4,'Banglore','Frankfurt',10000,'6:00','23:30',50000), (5,'Kolkata','New
Delhi',2400,'11:00','3:30',9000),
(6,'Banglore','Frankfurt',8000,'9:00','23:00',40000);
select * from flights;
```

Result Grid | Filter Rows: _____ | Edit: Export/Import: | W

| | fno | from_place | to_place | distance | departs | arrives | price |
|---|------|------------|-----------|----------|----------|----------|-------|
| ▶ | 1 | Bengaluru | New Delhi | 500 | 06:00:00 | 09:00:00 | 5000 |
| | 2 | Bengaluru | Chennai | 300 | 07:00:00 | 08:30:00 | 3000 |
| | 3 | Trivandrum | New Delhi | 800 | 08:00:00 | 11:30:00 | 6000 |
| | 4 | Bengaluru | Frankfurt | 1000 | 06:00:00 | 23:30:00 | 50000 |
| | 5 | Kolkata | New Delhi | 2400 | 11:00:00 | 03:30:00 | 9000 |
| * | HULL | HULL | HULL | HULL | HULL | HULL | HULL |

flights 16 ×

Queries:

- i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

select a.aname from aircraft a,employee s, certified c where s.salary >80000 and a.aid=c.aid and c.eid=s.eid;

| | aname |
|---|------------|
| ▶ | Airbus |
| | JetAirways |
| | Airbus |

- ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.

SELECT c.eid,MAX(cruisingrange) FROM certified c,aircraft a WHERE c.aid=a.aid

| Result Grid | | Filter Rows: | Export: | Wrap Cell Content |
|-------------|-----|----------------------|---------|-------------------|
| | eid | max(a.cruisingrange) | | |
| ▶ | 102 | 4500 | | |
| | 104 | 2200 | | |
| | 101 | 5000 | | |
| | 103 | 4500 | | |

Result 17 x

- iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.

select e.ename from employee e where e.salary<(select min(f.price) from flights f where f.ffrom='Banglore' and f.tto='Frankfurt');

| | ename |
|---|-------|
| ▶ | Tilak |

- iii. For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the Average salary of all pilots certified for this aircraft. select a.aname,avg(e.salary) from aircraft a,employee e,certified c where a.aid=c.aid and e.eid=c.eid and a.cruisingrange>1000 group by c.aid;

The screenshot shows the Oracle SQL Developer interface. The code editor at the top contains the following SQL query:

```

SELECT a.aid,a.aname,AVG(e.salary)
FROM aircraft a,certified c,employee e
WHERE a.aid=c.aid
AND c.eid=e.eid
AND a.cruisingrange>1000
GROUP BY a.aid,a.aname;

```

Below the code editor is a result grid showing the output of the query:

| aid | aname | AVG(e.salary) |
|-----|--------|---------------|
| 1 | Airbus | 71000.0000 |
| 4 | Indigo | 50000.0000 |
| 5 | Boeing | 60000.0000 |
| 6 | Airbus | 67333.3333 |

At the bottom of the interface, there is a log titled "Result 23" showing the execution history:

| # | Time | Action | Message |
|----|----------|--|-------------------|
| 32 | 10:17:19 | SELECT c.eid,MAX(cruisingrange) FROM certified c,aircraft a WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>=3 LIMIT 0, 10000 | 4 row(s) returned |
| 33 | 10:17:30 | SELECT c.eid,MAX(cruisingrange) FROM certified c,aircraft a WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>=3 LIMIT 0, 10000 | 4 row(s) returned |
| 34 | 10:18:10 | SELECT c.eid,MAX(cruisingrange) FROM certified c,aircraft a WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>=3 LIMIT 0, 10000 | 4 row(s) returned |
| 35 | 10:18:48 | SELECT c.eid,MAX(cruisingrange) FROM certified c,aircraft a WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>=3 LIMIT 0, 10000 | 4 row(s) returned |

- iv. Find the names of pilots certified for some Boeing aircraft. select e.ename from employee e where e.eid in (select c.eid from certified c where c.aid in (select a.aid from aircraft a where a.aname='Boeing'));

```

10m21c10_0_supplier
10m21c10_03
10m21c10_3_emp_database
10m21c10_4_bank
10m21c10_6_employee
21cs103
employee
employee_emp_database
employees
managelemployees_emp_database
sys

88
89 -- 4th lv. For all aircraft with cruising range over 1000 Kms, find the name of the
90 -- aircraft and the average salary of all pilots certified for this aircraft.
91
92 *   SELECT a.aid,a.aname,AVG(e.salary)
93     FROM aircraft a,certified c,employee e
94       WHERE a.aid=c.aid
95       AND c.aid=e.aid
96       AND a.cruisingrange>1000
97     GROUP BY a.aid,a.aname;
98
99 -- v. Find the names of pilots certified for some Boeing aircraft
100
101 *   SELECT distinct e.ename
...

```

Administration Schemas Information No object selected

Result Grid | Filter Rows: Export: Wrap Cell Contents:

| ename |
|--------|
| Anvash |
| Rakesh |
| Lokesh |

Action Output

| Time | Action | Message |
|------|---|---|
| 37 | 10:25:11 SELECT a.aid,a.aname,AVG(e.salary) FROM aircraft a,certified c,employee e WHERE a.aid=c.aid AND c.aid=e.aid AND a.cruisingrange>1000 GROUP BY a.aid,a.aname; | 4 rows(s) returned |
| 38 | 10:28:33 select distinct e.ename from employee e WHERE e.salary=(SELECT MIN(f.price) FROM flights f WHERE f.ffrom='Banglore' AND f.tto='Frankfort') LIMIT 0,10 | 1 row(s) returned |
| 39 | 10:28:44 SELECT a.aid,a.aname,AVG(e.salary) FROM aircraft a,certified c,employee e WHERE a.aid=c.aid AND c.aid=e.aid AND a.cruisingrange>1000 GROUP BY a.aid,a.aname; | 4 rows(s) returned |
| 40 | 10:33:20 SELECT distinct e.ename FROM employees e,aircraft a,certified c WHERE e.aid=c.aid AND c.aid=e.aid AND a.aname='Boeing' LIMIT 0,10000 | Error Code: 1146. Table 'airline.employees' doesn't exist |
| 41 | 10:33:35 SELECT a.aid,a.aname,AVG(e.salary) FROM aircraft a,certified c,employee e WHERE a.aid=c.aid AND c.aid=e.aid AND a.cruisingrange>1000 GROUP BY a.aid,a.aname; | 4 rows(s) returned |
| 42 | 10:34:09 SELECT distinct e.ename FROM employee e,aircraft a,certified c WHERE e.aid=c.aid AND c.aid=e.aid AND a.aname='Boeing' LIMIT 0,10000 | 3 rows(s) returned |

Object Info Session 100 feet Road Construction

vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
 select a.aid from aircraft a where a.cruisingrange>(select distance from flights where ffrom='Banglore' and tto='New Delhi');

Result Grid | Filter Rows: Export:

| | aname | avg(e.salary) |
|---|--------|---------------|
| ▶ | Airbus | 71000.0000 |
| | Indigo | 50000.0000 |
| | Boeing | 60000.0000 |
| | Airbus | 67333.3333 |

Program 7 week 9: NOSQL

Perform the following DB operations using MongoDB.

1. Create a database “Student” with the following attributes Rollno, Age, ContactNo, Email-Id.
2. Insert appropriate values
3. Write query to update Email-Id of a student with rollno 10.
4. Replace the student name from “ABC” to “FEM” of rollno 11.
5. Export the created table into local file system
6. Drop the table

7. Import a given csv dataset from local file system into mongodb collection.

Create database

```
db.createCollection("Student");
```

```
db.Student.insert({RollNo:1, Age:21, Cont:9876, email:"antara.de9@gmail.com"});  
db.Student.insert({RollNo:2, Age:22, Cont:9976, email:"anushka.de9@gmail.com"});  
db.Student.insert({RollNo:3, Age:21, Cont:5576, email:"anubhav.de9@gmail.com"});  
db.Student.insert({RollNo:4, Age:20, Cont:4476, email:"pani.de9@gmail.com"});  
db.Student.insert({RollNo:10, Age:23, Cont:2276, email:"rekha.de9@gmail.com"});
```

```
db.Student.find()
```

```
db.Student.update({RollNo:10}, {$set: { email:"Abhinav@gmail.com" }})
```

Insert appropriate data: db.Student.insert({RollNo:11, Age:22, Name: "ABC", Cont:2276, email:"rea.de9@gmail.com"});

Replace Query: db.Student.update({RollNo:11, Name:"ABC"}, {\$set: {Name:"FEM"}})

The screenshot shows the MongoDB Atlas interface with the following details:

- Database:** lab_9
- Collection:** Student
- Storage Size:** 36KB
- Logical Data Size:** 6MB
- Total Documents:** 6
- Indexes Total Size:** 54KB

The interface shows three documents in the list:

- Document 1:
_id: ObjectId('63c629dc8ea8e4716c8aa114')
RollNo: 4
Age: 21
Cont: 4476
email: "pani.de9@gmail.com"
- Document 2:
_id: ObjectId('63c629f15ea8e4716c8aa115')
RollNo: 10
Age: 23
Cont: 2276
email: "abhinav@gmail.com"
- Document 3:
_id: ObjectId('63c62abb5ea8e4716c8aa116')
RollNo: 11
Age: 22
Name: "FEM"
Cont: 2276
email: "rea.de9@gmail.com"

Screenshot of the MongoDB Cloud Atlas interface showing the results of a query on the 'Student' collection in the 'lab_9' database.

Find Results:

```

_id: ObjectId('63c829dc5ea8e4716c8aa11')
RollNo: 1
Age: 21
Cont: 9876
email: "antara.de@gmail.com"

_id: ObjectId('63c829dc5ea8e4716c8aa112')
RollNo: 2
Age: 22
Cont: 9976
email: "anuska.de@gmail.com"

_id: ObjectId('63c829dc5ea8e4716c8aa113')
RollNo: 3
Age: 23
Cont: 5576
email: "anubhav.de@gmail.com"

```

System Status: All Good
©2023 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Activate Windows: Go to Settings to activate Windows.

Command Prompt:

```

SyntaxError: Unexpected token, expected ',' (1:30)
1 db.student.update({RollNo:11},Name:"ABC"),{$set
2 t:{Name:"FEM"}}
3

Atlas:atlas-2z78d4-shard-0 [primary] lab_9> db.Student.update({RollNo:11},Name:"ABC"),{$set
... db.Student.update({RollNo:11},Name:"ABC"),{$set t:{Name:"FEM"}}
Uncatched:
SyntaxError: Unexpected token, expected ',' (1:30)
1 db.student.update({RollNo:11},Name:"ABC"),{$set
2 t:{Name:"FEM"}}
3

Atlas:atlas-2z78d4-shard-0 [primary] lab_9> db.Student.update({RollNo:Name:"ABC"},{$set:{Name:"FEM"}})
Uncatched:
SyntaxError: Unexpected token, expected ',' (1:30)
1 db.student.update({RollNo:Name:"ABC"},{$set:{Name:"FEM"}})
2

Atlas:atlas-2z78d4-shard-0 [primary] lab_9> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas:atlas-2z78d4-shard-0 [primary] lab_9>

```

Activate Windows: Go to Settings to activate Windows.

```
3 of 4 Page view Read aloud Add text Draw Highlight Erase
Command Prompt
Cont: 9876,
email: "antara.dev@gmail.com"
{
  "_id": ObjectId("63c629dc5ea8e4716c8aa112"),
  RollNo: 8,
  Age: 22,
  Cont: 9976,
  email: "anushka.dev@gmail.com"
},
{
  "_id": ObjectId("63c629dc5ea8e4716c8aa113"),
  RollNo: 5,
  Age: 21,
  Cont: 5576,
  email: "anubhav.dev@gmail.com"
},
{
  "_id": ObjectId("63c629dc5ea8e4716c8aa114"),
  RollNo: 4,
  Age: 20,
  Cont: 4476,
  email: "pani.dev@gmail.com"
},
{
  "_id": ObjectId("63c629f15ea8e4716c8aa115"),
  RollNo: 10,
  Age: 23,
  Cont: 2276,
  email: "Abinav@gmail.com"
},
{
  "_id": ObjectId("63cb2ab65ea8e4716c8aa116"),
  RollNo: 11,
  Age: 22,
  Name: "Rac",
  Cont: 2276,
  email: "rea.dev@gmail.com"
}
]
Atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.update({RollNo:11,Name:"ABC"},{$set:{t:(Name:"FEM")}}
{
  "_id": ObjectId("63cb2ab65ea8e4716c8aa116")
}
caught:
SyntaxError: Unexpected token, expected "," (2:0)
at /db.Student.update({RollNo:11,Name:"ABC"},{$set:{t:(Name:"FEM"))})
at ... db.Student.update({RollNo:11,Name:"ABC"},{$set:t:(Name:"FEM")})
caught:
SyntaxError: Unexpected token, expected "," (2:0)
at /db.Student.update({RollNo:11,Name:"ABC"},{$set:{t:(Name:"FEM")}})
at ... db.Student.update({RollNo:11,Name:"ABC"},{$set:t:(Name:"FEM")})
caught:
SyntaxError: Unexpected token, expected "," (1:0)
Atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.update({RollNo:Name:"ABC"},{$set:(Name:"FEM")})
caught:
SyntaxError: Unexpected token, expected "," (1:0)

Activate Windows
Go to Settings to activate Windows.
ENG IN 17-01-2023
```

```
2 of 4 Page view Read aloud Add text Draw Highlight Erase
[{"id": "2", "Cont": "9976", "email": "anushka.dev@gmail.com"}, {"_id: ObjectId("63c629dc5ea8e4716c8aa113"), "RollNo": 3, "Age": 22, "Cont": 5576, "email": "anubhav.dev@gmail.com"}, {"_id: ObjectId("63c629dc5ea8e4716c8aa114"), "RollNo": 4, "Age": 22, "Cont": 4476, "email": "pani.dev@gmail.com"}, {"_id: ObjectId("63c629f15ea8e4716c8aa115"), "RollNo": 10, "Age": 22, "Cont": 5576, "email": "reduha.dev@gmail.com"}]
atlas atlas-278d4-shard-0 [primary] lab_9> db.Student.update({RollNo:10},{$set:{... email:"Anubhav@gmail.com"})}
DeprecationWarning: collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.

acknowledged: true,
insertedId: null,
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0

atlas atlas-278d4-shard-0 [primary] lab_9> db.Student.insert({RollNo:11,Age:22,Name:"ABC",Cont:2226,email:"rea.dev@gmail.com"})
acknowledged: true,
insertedIds: { _id: ObjectId("63c62abb5ea8e4716c8aa116") }

atlas atlas-278d4-shard-0 [primary] lab_9> db.Student.find()
{
    "_id": ObjectId("63c629dc5ea8e4716c8aa111"),
    "RollNo": 1,
    "Age": 21,
    "Cont": 9876,
    "email": "antara.dev@gmail.com",
    "_id: ObjectId("63c629dc5ea8e4716c8aa112"),
    "RollNo": 2,
    "Age": 22,
    "Cont": 9976,
    "email": "anushka.dev@gmail.com",
    "_id: ObjectId("63c629dc5ea8e4716c8aa113"),
    "RollNo": 3,
    "Age": 22,
    "Cont": 5576,
```

```

Microsoft Windows [Version 10.0_22000.795]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dhanush\mongosh "mongodb://cluster0.na6tbyc.mongodb.net/lab_9" --apiVersion 1 --username dhanush
Enter password: *****
Current Mongosh Log ID: 63c626ad86633c814e34857
Connecting to: mongodb://cluster0.na6tbyc.mongodb.net/lab_9?appName=mongosh+1.6.2
Mongo Node(s): 5.0.24 (API Version 1)
Using Mongosh: 1.6.2

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

atlas atlas-2z78d4-shard-0 [primary] lab_9> db.createCollection("Student");
{
  "ok": 1
}
atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.insert({RollNo:1, Age:21, Cont:9876, email:"antara.de@gmail.com"});
{
  "acknowledged": true,
  "insertedId": {
    "$oid": "63c629dc5ea8e4716c8aa111"
  }
}
atlas atlas-2z78d4-shard-0 [primary] lab_9>
atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.insert({RollNo:2, Age:22, Cont:9976, email:"anushka.de@gmail.com"});
{
  "acknowledged": true,
  "insertedId": {
    "$oid": "63c629dc5ea8e4716c8aa112"
  }
}
atlas atlas-2z78d4-shard-0 [primary] lab_9>
atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.insert({RollNo:3, Age:21, Cont:5576, email:"anubhav.de@gmail.com"});
{
  "acknowledged": true,
  "insertedId": {
    "$oid": "63c629dc5ea8e4716c8aa113"
  }
}
atlas atlas-2z78d4-shard-0 [primary] lab_9>
atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.insert({RollNo:4, Age:20, Cont:4476, email:"puni.de@gmail.com"});
{
  "acknowledged": true,
  "insertedId": {
    "$oid": "63c629dc5ea8e4716c8aa114"
  }
}
atlas atlas-2z78d4-shard-0 [primary] lab_9>
atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.insert({RollNo:10, Age:23, Cont:2276, email:"rekha.de@gmail.com"});
{
  "acknowledged": true,
  "insertedId": {
    "$oid": "63c629f15ea8e4716c8aa115"
  }
}
atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.find()
[
  {
    "_id": ObjectId("63c629dc5ea8e4716c8aa111"),
    "RollNo": 1,
    "Age": 21,
    "Cont": 9876,
    "email": "antara.de@gmail.com"
  },
  {
    "_id": ObjectId("63c629dc5ea8e4716c8aa112"),
    "RollNo": 2,
    "Age": 22,
    "Cont": 9976,
    "email": "anushka.de@gmail.com"
  },
  {
    "_id": ObjectId("63c629dc5ea8e4716c8aa113"),
    "RollNo": 3,
    "Age": 21,
    "Cont": 5576,
    "email": "anubhav.de@gmail.com"
  },
  {
    "_id": ObjectId("63c629dc5ea8e4716c8aa114"),
    "RollNo": 4,
    "Age": 20,
    "Cont": 4476,
    "email": "puni.de@gmail.com"
  },
  {
    "_id": ObjectId("63c629f15ea8e4716c8aa115"),
    "RollNo": 10,
    "Age": 23,
    "Cont": 2276,
    "email": "rekha.de@gmail.com"
  }
]

Activate Windows
Go to Settings to activate Windows.

ENG IN 17-01-2023 10:31

```

Week 10 :NOSQL

1. Create a collection by name Customers with the following attributes.

Cust_id, Acc_Bal, Acc_Type

```
db.createCollection("Customer");
```

2.DROP COLLECTION

```
db.Customer.drop();
```

3. Insert at least 5 values into the table

```
db.Student.insert(  
{_id:1,  
StudName:"MichelleJacintha",  
Grade:"VII",  
Hobbies:"InternetSurfing"});
```

BULK INSERT//

```
var mystudent =  
[ {_id:4, StudName:"saurav", Grade:"V", Hobbies:"Dance"}, {_id:5, StudName:"kumar",  
Grade:"VI", Hobbies:"Singing"}]  
db.Student.insert(mystudent)
```

4.UPDATE

```
db.student.update({StudName:"Aryan David"},{$set:{StudName:"Saurav"}});
```

5. View the entire table

```
db.Student.find()  
db.Student.find({StudName:"Aryan David"});  
db.Students.find({}, {StudName:1,Grade:1,_id:0});  
db.Students.find({Grade:{$eq:'VII'}})
```

6.COUNT

```
db.Student.count();
```

7.SORT

```
db.Students.find().sort({StudName:1})
```

8.REMOVE

```
db.Student.remove({"StudName":"saurav"})  
db.student.find({"StudName":"saurav"});
```

9.DISPLAY

```
db.Student.find().limit(2)
```

```

cd mongosh mongoDB>run//<credentials>@cluster0.na0tbyc.mongodb.net/myFirstDatabase
C:\Users\Admin\mongosh "mongodb://cluster0.na0tbyc.mongodb.net/myFirstDatabase" --apiVersion 1 --username dhanush
Enter password: *****
Current Mongosh Log ID: 63c6f7922d5ab949480e886
Connecting to: mongodb://<credentials>@cluster0.na0tbyc.mongodb.net/myFirstDatabase?appName=mongosh+1.6.2
Using MongoDB: 5.0.14 (API Version 1)
Using Mongosh: 1.6.2
For mongosh info see: https://docs.mongodb.com/mongodb-shell/
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.createCollection("Student");
Uncought:
SyntaxError: Untermited string constant. (1:20)
> 1 | db.createCollection("Student");
^
2 |
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.createCollection("Student");
MongoServerError: Collection already exists. NS: myFirstDatabase.Student
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.drop();
true
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.insert(
... {_id:,
...
... StudName:"MichelleJacintha",
...
... Grade:"VII",
...
... Hobbies:"InternetSurfing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
( acknowledged: true, insertedIds: [ '_id' ] )
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> var mystudent =
... [ { _id:4, StudName:"saurav", Grade:"V", Hobbies:"Dance"}, 
... { _id:5, StudName:"kumar", Grade:"VI", Hobbies:"Singing"} ]
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase>
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> var mystudent =
... [ { _id:4, StudName:"saurav", Grade:"V", Hobbies:"Dance"}, 
... { _id:5, StudName:"kumar", Grade:"VI", Hobbies:"Singing"} ]
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase>
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.insert(mystudent)
( acknowledged: true, insertedIds: [ '_id' ] )
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.student.update({$set:{StudName:"Aryan"
Uncought:
SyntaxError: Untermited string constant. (1:28)
> 1 | db.student.update({StudName:"Aryan";
^
2 |
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> David",{$set:{StudName:"Saurav"}});
Uncought:
SyntaxError: Missing semicolon. (1:5)
> 1 | David",{$set:{StudName:"Saurav"}});
^
2 |
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.student.update({StudName:"Aryan"
Uncought:
SyntaxError: Untermited string constant. (1:28)
> 1 | db.student.update({StudName:"Aryan
^
2 |
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.update({StudName:"Aryan David",{$set:{StudName:"Saurav"}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
( acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
)
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.update({_id:3, StudName:"Aryan David", Grade:
... "VII"},{$set:{Hobbies:"Skating"}},(upsert:true));
( acknowledged: true,
  insertedId: 3,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
)
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find()
{
  {
    _id: 3,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  {
    _id: 4, StudName: "saurav", Grade: 'V', Hobbies: 'Dance' },
  { _id: 5, StudName: "kumar", Grade: 'VI', Hobbies: 'Singing' },
  { _id: 3, Grade: "VII", StudName: "Aryan David", Hobbies: "Skating" }
}
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({studname:"Aryan David"});
{
  {
    _id: 3, Grade: "VII", StudName: "Aryan David", Hobbies: "Skating"
  }
}
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Students.find({}, {StudName:1,Grade:1,_id:0});
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({}, {StudName:1,Grade:1,_id:0});
[ { StudName: 'MichelleJacintha', Grade: 'VII' },
  { StudName: 'saurav', Grade: 'V' },
  { StudName: 'kumar', Grade: 'VI' },
  { Grade: 'VII', StudName: "Aryan David" } ]
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({Grade:{$eq:"VII"}})
{
  {
    _id: 3,
    StudName: 'MichelleJacintha',
  }
}

```

```
[mongosh mongodbsrv://<credentials>@cluster0.rrffbc.mongodb.net/myFirstDatabase]
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({Grade:{$eq:'VII'}})
[{"_id": 1, "StudentName": "MichelleJacintha", "Grade": "VII", "Hobbies": "Internetsurfing"}, {"_id: 3, Grade: "VII", StudentName: "Aryan David", Hobbies: "Skating"}]
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find ({Hobbies :{ $in: ['Chess','Skating']}})
[{"_id: 3, Grade: "VII", StudentName: "Aryan David", Hobbies: "Skating"}]
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({studentName:/^M/});
[{"_id": 1, "StudentName": "MichelleJacintha", "Grade": "VII", "Hobbies": "Internetsurfing"}]
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({studentName:/^A/});
[{"_id": 1, "StudentName": "MichelleJacintha", "Grade": "VII", "Hobbies": "Internetsurfing"}, {"_id: 3, Grade: "VII", StudentName: "Aryan David", Hobbies: "Skating"}]
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Students.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Students.find().sort({StudentName:1})
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find().sort({StudentName:1})
[{"_id: 3, Grade: "VII", StudentName: "Aryan David", Hobbies: "Skating"}, {"_id: 1, "StudentName": "MichelleJacintha", "Grade": "VII", "Hobbies": "Internetsurfing"}, {"_id: 5, StudentName: "kumar", Grade: "VI", Hobbies: "Singing"}, {"_id: 4, StudentName: "saurav", Grade: "V", Hobbies: "Dance"}]
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find().limit(2)
[{"_id": 1, "StudentName": "MichelleJacintha", "Grade": "VII", "Hobbies": "Internetsurfing"}, {"_id: 4, StudentName: "saurav", Grade: "V", Hobbies: "Dance"}]
atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase>
```

ENG IN 24-01-2023 10:49