VISVESVARAYATECHNOLOGICALUNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Database Management Systems (23CS3PCDBM)

Submitted by Chiraiya Sethiya(1BM23CS080)
in partial fulfillment for the award of the degree of
BACHELOROFENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019 Sep-2024 to Jan-2025
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 (23CS3PCDBM)" carried out by **Chiraiya Sethiya(1BM23CS080),** 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 (23CS3PCDBM) work prescribed for the said degree.

Sheetal V A Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor HOD Department of CSE, BMSCE

Index

Insurance Database

Question

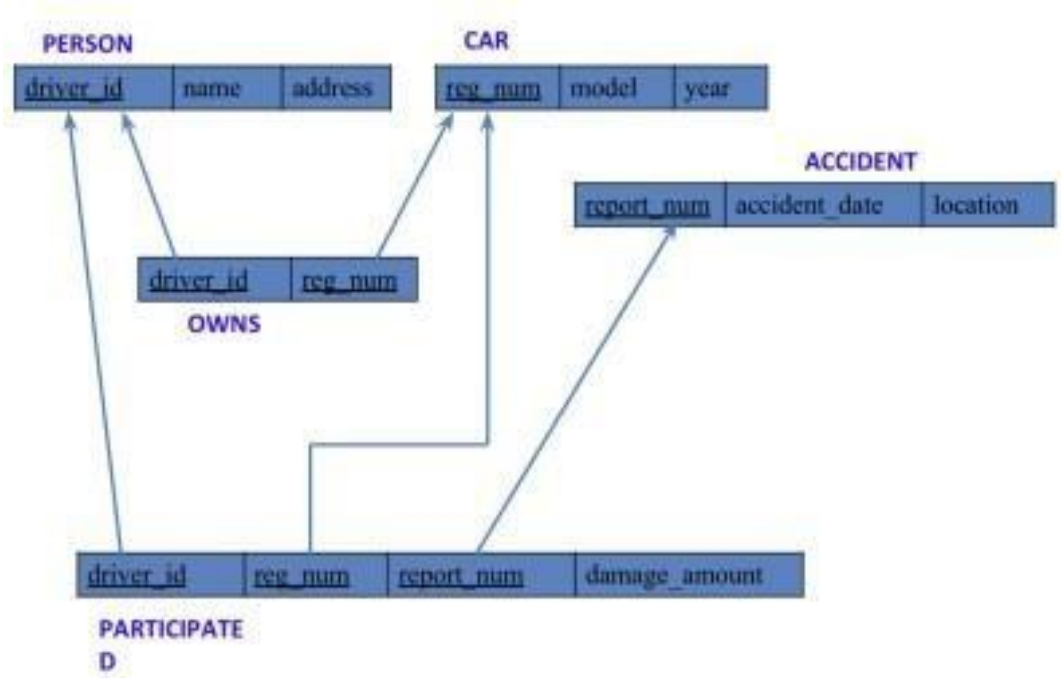
(Week 1)

- 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. -

Enter at least five tuples for each relation

- Display Accident date and location
- Update the damage amount to 25000 for the car with a specific reg_num (example 'K A053408') for which the accident report number was 12.
- Add a new accident to the database.
- To Do
- Display Accident date and location
- Display driver id who did accident with damage amount greater than or equal to Rs.25000

Schema Diagram



Create Database

```
create database insurance_1BM23CS080;

use insurance_1BM23CS080;
```

Create Table

```
create table person
( driver_id
varchar(10), name
varchar(20), address
varchar(30),
PRIMARY KEY(driver_id)
);

create table car
(
reg_num varchar(10),
model varchar(10),
year int,
PRIMARY KEY(reg_num)
);
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),
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)
);
```

Structure of the table

desc person;

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

desc car;

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

desc accident;

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

desc owns;

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

desc participated;

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	

Inserting Values to the table

insert into person values('A01','Richard','Srinivar 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','John','Hanumanth
Nagar'); select * from person;

driver_id	name	address
A01	Richard	Srinivar Nagar
A02	Pradeep	Rajaji Nagar
A03	Smith	Ashok Nagar
A04	Venu	N.R Colony
A05	John	Hanumanth Nagar
NULL	NULL	NULL

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','Honola',2008);
insert into car values('KA041702','Audi',2005); select
* from car;

	reg_num	model	year
▶	KA031181	Lancer	1957
	KA041702	Audi	2005
	KA052250	Indica	1990
	KA053408	Honola	2008
	KA095477	Toyota	1998
*	NULL	NULL	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');
select * from accident;

	report_num	accident_date	location
▶	11	2003-01-01	Mysore Road
	12	2004-02-02	South End Circle
	13	2003-01-21	Bull Temple Road
	14	2008-02-17	Mysore Road
	15	2004-03-05	Kanakpura Road
	16	2008-03-08	Dolmor
*	NULL	NULL	NULL

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

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

insert into participated values('A01','KA052250',11,10000);
insert into participated values('A02','KA031181',12,50000);
insert into participated values('A03','KA095477',13,25000);
insert into participated values('A04','KA053408',14,3000);

```
insert into participated values('A05','KA041702',15,5000);
select * from participated;
```

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

Queries:

Update the damage amount to 25000 for the car with a specific reg-num (example 'KA053408') for which the accident report number was 14.

```
UPDATE participated set damage_amount=25000
WHERE reg_num='KA053408' AND report_num=14;
select * from participated;
```

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

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

Add new accident to the database

```
INSERT into accident values(16,'2008-03-08','Dolmor');
select * FROM accident;
```

report_num	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South End Circle
13	2003-01-21	Bull Temple Road
14	2008-02-17	Mysore Road
15	2004-03-05	Kanakpura Road
16	2008-03-08	Dolmor
NULL	NULL	NULL

More Queries on Insurance Database:

List all the entire participated relation in descending order of damage_amount

```
select * FROM participated ORDER BY damage_amount desc;
```

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

Find average damage_amount

select avg(damage_amount) from participated;

	avg(damage_amount)
▶	23000.0000

Delete the tuple whose damage_amount is below average amount damage_amount

```
delete from participated
where  damage_amount
(
select avg_damage from(select
avg(damage_amount) as avg_damage from
participated) as avg_table
);
set sql_safe_updates=0;
```

List the name of drivers whose damage is greater than the avg 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
	Venu

Find the maximum damage_amount

select max(damage_amount) from participated;

	max(damage_amount)
▶	50000

Display accident date and location

select accident_date,location from accident;

accident_date	location
2003-01-01	Mysore Road
2004-02-02	South End Circle
2003-01-21	Bull Temple Road
2008-02-17	Mysore Road
2004-03-05	Kanakpura Road
2008-03-08	Dolmor

Display driver_id who did accident with damage_amount>=25000

select driver_id from participated where damage_amount>=25000;

driver_id
A02
A03
A04

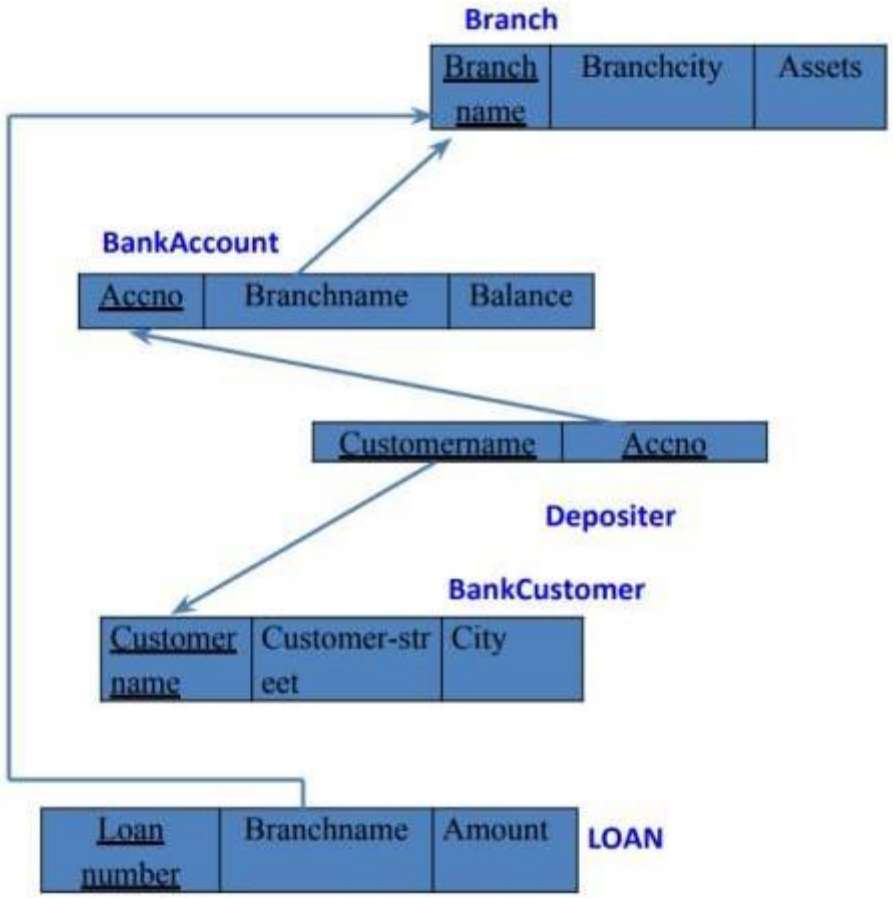
Bank Database

Question

(Week 3)

- Branch (branch-name: String, branch-city: String, assets: real)
- BankAccount(accno: int, branch-name: String, balance: real)
- BankCustomer (customer-name: String, customer-street: String, customer-city: String) - Depositer(customer-name: String, accno: int)
- LOAN (loan-number: int, branch-name: String, amount: real)
- Create the above tables by properly specifying the primary keys and the foreign keys. Enter at least five tuples for each relation.
- Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.
- Find all the customers who have at least two accounts at the same branch (ex. SBI_ResidencyRoad).
- Create a view which gives each branch the sum of the amount of all the loans at the branch.

Schema Diagram



Create Database

```
create database BankDatabase_080;

use BankDatabase_080;
```

Create Table

```
create table Branch
(
Branchname varchar(20),
Branchcity varchar(10),
Assets int,
PRIMARY KEY (Branchname)
);
create table BankAccount
(
Accno int,
```

```
Branchname varchar(20),
Balance int,
PRIMARY KEY (Accno,Branchname),
FOREIGN KEY(Branchname) references Branch(Branchname)
);

create table BankCustomer
(
Customername varchar(10),
Customerstreet varchar(20),
Customercity varchar(10),
PRIMARY KEY(Customername)
);

create table Depositor
(
Customername varchar(10),
Accno int,
PRIMARY KEY (Customername,Accno),
FOREIGN KEY(Customername) references BankCustomer(Customername),
FOREIGN KEY(Accno) references BankAccount(Accno)
);

create table Loan
(
Loannumber int,
Branchname varchar(20),
Amount int,
PRIMARY KEY (Loannumber,Branchname),
FOREIGN KEY(Branchname) references Branch(Branchname)
);
```

Structure of the table

desc Branch;

Field	Type	Null	Key	Default	Extra
Branchname	varchar(20)	NO	PRI	NULL	
Branchcity	varchar(10)	YES		NULL	
Assets	int	YES		NULL	

desc BankAccount;

Field	Type	Null	Key	Default	Extra
Accno	int	NO	PRI	NULL	
Branchname	varchar(20)	NO	PRI	NULL	
Balance	int	YES		NULL	

desc BankCustomer;

Field	Type	Null	Key	Default	Extra
Customername	varchar(10)	NO	PRI	NULL	
Customerstreet	varchar(20)	YES		NULL	
Customercity	varchar(10)	YES		NULL	

desc Depositor;

Field	Type	Null	Key	Default	Extra
Customername	varchar(10)	NO	PRI	NULL	
Accno	int	NO	PRI	NULL	

desc Loan;

Field	Type	Null	Key	Default	Extra
Loannumber	int	NO	PRI	NULL	
Branchname	varchar(20)	NO	PRI	NULL	
Amount	int	YES		NULL	

Inserting Values to the table

insert into Branch values('SBI_Chamrajpet','Bangalore',50000);
insert into Branch
values('SBI_ResidencyRoad','Bangalore',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); select * from
Branch;

Branchname	Branchcity	Assets
SBI_Chamrajpet	Bangalore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParliamentRoad	Delhi	10000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

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);
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

insert into BankCustomer values('Avinash','BullTempleRoad ','Bangalore');
insert into BankCustomer values('Dinesh','BannergattaRoad','Bangalore');
insert into BankCustomer
values('Mohan','NationalCollegeRoad','Bangalore'); insert into
BankCustomer values('Nikil','AkbarRoad','Delhi'); insert into BankCustomer
values('Ravi','PrithvirajRoad','Delhi'); select * from BankCustomer;

	Customername	Customerstreet	Customercity
▶	Avinash	Bull Temple Road	Bangalore
	Dinesh	Bannergatta Road	Bangalore
	Mohan	NationalCollegeRoad	Bangalore
	Nikil	Akbar Road	Delhi
	Ravi	Prithviraj Road	Delhi
*	NULL	NULL	NULL

```
insert into Depositor value('Avinash',1);
insert into Depositor value('Dinesh',2);
insert into Depositor value('Nikil',4);
insert into Depositor value('Ravi',5);
insert into Depositor value('Avinash',8);
insert into Depositor value('Nikil',9);
insert into Depositor value('Dinesh',10);
insert into Depositor value('Ravi',11);
select * from Depositor;
```

	Customername	Accno
▶	Avinash	1
	Dinesh	2
	Nikil	4
	Ravi	5
	Avinash	8
	Nikil	9
	Dinesh	10
	Ravi	11
*	NULL	NULL

```
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);
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
*	NULL	NULL	NULL

Queries:

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 as Asset_in_lakhs from Branch;

Branchname	Asset_in_lakhs
SBI_Chamrajpet	50000
SBI_Jantarmantar	20000
SBI_ParliamentRoad	10000
SBI_ResidencyRoad	10000
SBI_ShivajiRoad	20000
NULL	NULL

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

select Customername , Branchname from Depositor D, BankAccount B
where D.Accno=B.Accno group by Customername,Branchname having count(B.Accno)>=2;

Customername	Branchname
Dinesh	SBI_ResidencyRoad
Nikil	SBI_ParliamentRoad
Ravi	SBI_Jantarmantar

Create a view which gives each branch the sum of the amount of all the Loans at the Branch.

Create view Branch_Loan_Sum as select Branchname, sum(Amount) as total_loan_amount from Loan group by Branchname;

Branchname	total_loan_amount
SBI_Chamrajpet	1000
SBI_Jantarmantar	5000
SBI_ParliamentRoad	4000
SBI_ResidencyRoad	2000
SBI_ShivajiRoad	3000

More Queries on Bank Database:

Find all the customers who have an account at all the branches located in a specific city (Ex. Delhi).

SELECT DISTINCT d.Customername FROM Depositor D JOIN BankAccount BA
ON D.Accno=BA.Accno JOIN Branch b on BA.Branchname=b.Branchname WHERE
b.Branchcity='Delhi' GROUP BY d.Customername
HAVING COUNT(DISTINCT BA.Branchname)=(SELECT COUNT(B2.Branchname) FROM Branch B2
WHERE B2.Branchcity='Delhi');

	Customername
▶	Nikil

Find all customers who have a loan at the bank but do not have an account.

SELECT DISTINCT B.Customername FROM Borrower B LEFT JOIN Depositor d
On B.Customername=d.Customername WHERE d.Customername IS NULL;

	Customername
▶	Mohan

Find all customers who have both an account and a loan at the Bangalore branch.

SELECT DISTINCT d.Customername FROM Depositor D JOIN BankAccount BA
ON D.Accno=BA.Accno JOIN Loan l on BA.Branchname=l.Branchname WHERE
BA.Branchname='Bangalore' AND l.Branchname='Bangalore';

	Customername

Find the names of all branches that have greater assets than all branches located in Bangalore.

SELECT B.Branchname FROM Branch b where b.Assets>(SELECT MAX(b.Assets) FROM
Branch B WHERE B.Branchcity='Bangalore');

	Branchname
▶	SBI_MantriMarg
*	NULL

Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).

DELETE FROM BankAccount WHERE Branchname IN (SELECT
Branchname FROM Branch WHERE Branchcity='Bombay'); select *
from BankAccount;

accno	Branchname	Balance
1	SBI_Chamrajpet Road	2000
2	SBI_Residency Road	5000
4	SBI_Parliament Road	9000
5	SBI_JantarMantar	8000
8	SBI_Residency Road	4000

Update the Balance of all accounts by 5%.

UPDATE BankAccount SET Balance=Balance*1.05;

select * from BankAccount;

Accno	Branchname	Balance
1	SBI_Chamrajpet	2431
2	SBI_ResidencyRoad	6078
4	SBI_ParliamentRoad	10940
5	SBI_Jantarmantar	9724
8	SBI_ResidencyRoad	4863
9	SBI_ParliamentRoad	3647
10	SBI_ResidencyRoad	6078
11	SBI_Jantarmantar	2431
12	SBI_MantriMarg	2315
NULL	NULL	NULL

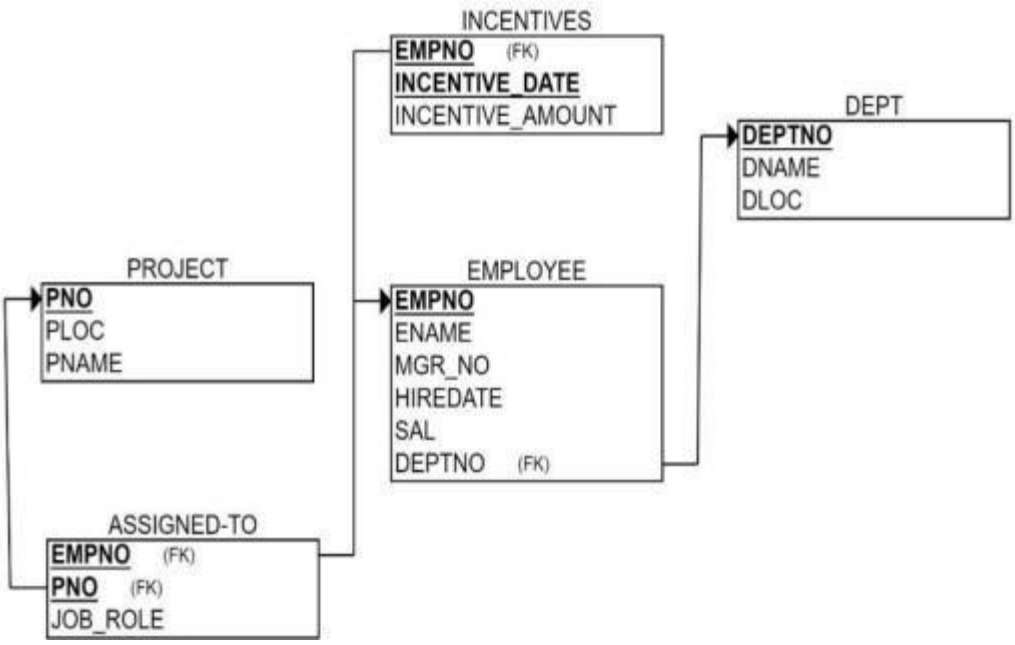
Employee Database

Question

(Week 5)

1. Using Scheme diagram, Create tables by properly specifying the primary keys and the foreign keys.
2. Enter greater than five tuples for each table.
3. Retrieve the employee numbers of all employees who work on project located in Bengaluru, Hyderabad, or Mysuru
4. Get Employee ID's of those employees who didn't receive incentives
5. 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.

Schema Diagram



Create Database

```
create database employee_Database_080;
use employee_Database_080;
```

Create Table

```
create table Dept(
Deptno int,
Dname varchar(50),
Dloc varchar(50),
primary key (Deptno)
);

create table Project(
Pno int,
Pname varchar(50),
Ploc varchar(50),
PRIMARY KEY(Pno)
);

create table Employee(
Empno int,
Ename varchar(50),
Mgrno int,
Hiredate date,
Sal int, Deptno int, primary key (Empno, Deptno),
foreign key(Deptno) REFERENCES
Dept(Deptno)
);

create table Incentive(
Empno int,
Incentivedate date, Incentiveamount int, primary
key(Incentivedate, Empno), foreign key (Empno)
REFERENCES Employee(Empno)
);

create table AssignedTo(
Empno int,
Pno int,
Jobrole varchar(50),
primary key(Empno, Pno),
foreign key(Empno)
references
Employee(Empno), foreign
key(Pno) references
Project(Pno)
);
```

Structure of the table

desc Dept;

Field	Type	Null	Key	Default	Extra
Deptno	int	NO	PRI	NULL	
Dname	varchar(50)	YES		NULL	
Dloc	varchar(50)	YES		NULL	

desc Project;

Field	Type	Null	Key	Default	Extra
Pno	int	NO	PRI	NULL	
Pname	varchar(50)	YES		NULL	
Ploc	varchar(50)	YES		NULL	

desc Employee;

Field	Type	Null	Key	Default	Extra
Empno	int	NO	PRI	NULL	
Ename	varchar(50)	YES		NULL	
Mgrno	int	YES		NULL	
Hiredate	date	YES		NULL	
Sal	int	YES		NULL	
Deptno	int	NO	PRI	NULL	

desc Incentive;

Field	Type	Null	Key	Default	Extra
Empno	int	NO	PRI	NULL	
Incentivedate	date	NO	PRI	NULL	
Incentiveamount	int	YES		NULL	

desc AssignedTo;

Field	Type	Null	Key	Default	Extra
Empno	int	NO	PRI	NULL	
Pno	int	NO	PRI	NULL	
Jobrole	varchar(50)	YES		NULL	

Inserting Values to the table

insert into Dept values(10, 'Computer Science', 'San Francisco');
insert into Dept values(20, 'Information Systems', 'New York');
insert into Dept values(30, 'Mechanical Engineering', 'Los Angeles'); insert into Dept values(40, 'Electrical Engineering', 'Boston'); insert into Dept values(50, 'Electronics', 'Chicago'); insert

into Dept values(60, 'Human Resources', 'Austin'); select * from Dept;

Deptno	Dname	Dloc
10	Computer Science	San Francisco
20	Information Systems	New York
30	Mechanical Engineering	Los Angeles
40	Electrical Engineering	Boston
50	Electronics	Chicago
60	Human Resources	Austin
NULL	NULL	NULL

insert into Project values (1, 'Market Research', 'Hyderabad');
insert into Project values (2, 'Software Update', 'Bengaluru');
insert into Project values (3, 'Product Launch', 'Mysuru');
insert into Project values (4, 'Website Redesign', 'Boston');
insert into Project values (5, 'Customer Support', 'Chicago');
insert into Project values (6, 'Employee Training', 'Austin');
select * from Project;

Pno	Pname	Ploc
1	Market Research	Hyderabad
2	Software Update	Bengaluru
3	Product Launch	Mysuru
4	Website Redesign	Boston
5	Customer Support	Chicago
6	Employee Training	Austin
NULL	NULL	NULL

insert into Employee values(1, 'Alice', 3, '2010-02-25', 72000, 10); insert into Employee values(2, 'Bob', 3, '2008-05-18', 56000, 20); insert into Employee values(3, 'Charlie', NULL, '2005-08-12', 90000, 10); insert into Employee values(4, 'David', 2, '2001-09-05', 65000, 20); insert into Employee values(5, 'Eve', 1, '2004-03-23', 71000, 30); insert into Employee values(6, 'Frank', 5, '2007-06-14', 51000, 30); insert into Employee values(7, 'Grace', 2, '2003-11-10', 78000, 40); select * from Employee;

Empno	Ename	Mgrno	Hiredate	Sal	Deptno
1	Alice	3	2010-02-25	72000	10
2	Bob	3	2008-05-18	56000	20
3	Charlie	NULL	2005-08-12	90000	10
4	David	2	2001-09-05	65000	20
5	Eve	1	2004-03-23	71000	30
6	Frank	5	2007-06-14	51000	30
7	Grace	2	2003-11-10	78000	40
NULL	NULL	NULL	NULL	NULL	NULL

```
insert into Incentive values(1, '2024-11-01', 5500);
insert into Incentive values(3, '2023-12-15', 9500);
insert into Incentive values(4, '2022-07-20', 3500);
insert into Incentive values(5, '2024-11-05', 4200);
insert into Incentive values(6, '2020-10-10', 4800);
insert into Incentive values(7, '2024-11-03', 8200);
select * from Incentive order by Empno asc;
```

Empno	Incentivedate	Incentiveamount
1	2024-11-01	5500
3	2023-12-15	9500
4	2022-07-20	3500
5	2024-11-05	4200
6	2020-10-10	4800
7	2024-11-03	8200
NULL	NULL	NULL

```
insert into AssignedTo values(1, 1, 'Team Leader');
insert into AssignedTo values(2, 2, 'Support
Engineer'); insert into AssignedTo values(3, 3, 'Project
Lead'); insert into AssignedTo values(4, 2, 'Junior
Developer'); insert into AssignedTo values(5, 1,
'Senior Developer'); insert into AssignedTo values(6,
4, 'Intern'); insert into AssignedTo values(7, 5,
'Consultant'); select * from AssignedTo;
```

Empno	Pno	Jobrole
1	1	Team Leader
2	2	Support Engineer
3	3	Project Lead
4	2	Junior Developer
5	1	Senior Developer
6	4	Intern
7	5	Consultant
NULL	NULL	NULL

Queries :

Retrieve the employee numbers of all employees who work on project located in Bengaluru, Hyderabad, or Mysuru.

```
SELECT a.EMPNO FROM AssignedTo a
JOIN Project p ON a.Pno = p.Pno
WHERE p.Ploc IN ('Bengaluru', 'Hyderabad', 'Mysuru');
```

	EMPNO
▶	1
	5
	2
	4
	3

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

```
select Empno from Employee e where Empno Not in(select Empno from Incentive );
```

	Empno
▶	2

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.Ename, e.Empno, d.Deptno, a.Jobrole, d.Dloc, p.Ploc
FROM Employee e, Dept d, Project p, Assignedto a
WHERE e.Deptno = d.Deptno AND e.Empno = a.Empno AND a.Pno = p.Pno AND d.Dloc = p.Ploc;
```

	Ename	Empno	Deptno	Jobrole	Dloc	Ploc

More Queries on Employee Database:

List the name of the managers with the maximum employees.

```
select e.Mgrno as managerid from Employee e join Employee m on e.Mgrno = m.Empno group by
e.Mgrno having count(e.Empno) = (select max(employeecount) from (select count(Empno) as employeecount
from Employee where Mgrno is not null group by Mgrno) as managercounts);
```

	managerid
▶	3
	2

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

select m.Empno as managerid,m.Ename as managername,m.Sal as managersalary from Employee m where m.Sal >(select avg(e.Sal) from Employee e where e.Mgrno = m.Empno);

managerid	managername	managersalary
1	Alice	72000
3	Charlie	90000
5	Eve	71000

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

select distinct e1.Ename as second_manager from Employee e1 where e1.Empno in (select distinct e2.Mgrno from Employee e2 where e2.Mgrno is not null);

second_manager
Charlie
Bob
Alice
Eve

Find the employee details who got second maximum incentive in November 2024.

select Empno,Incentivedate,Incentiveamount from Incentive where Incentivedate between '2024-11-01' and '2024-11-05' order by Incentiveamount desc ;

Empno	Incentivedate	Incentiveamount
7	2024-11-03	8200
1	2024-11-01	5500
5	2024-11-05	4200
NULL	NULL	NULL

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

select e.Empno as employeeID, e.Ename as employeename, e.Deptno as departmentid from Employee e Join Employee m on e.Mgrno = m.Empno where e.Deptno = m.Deptno;

employeeID	employeename	departmentid
1	Alice	10
4	David	20
6	Frank	30

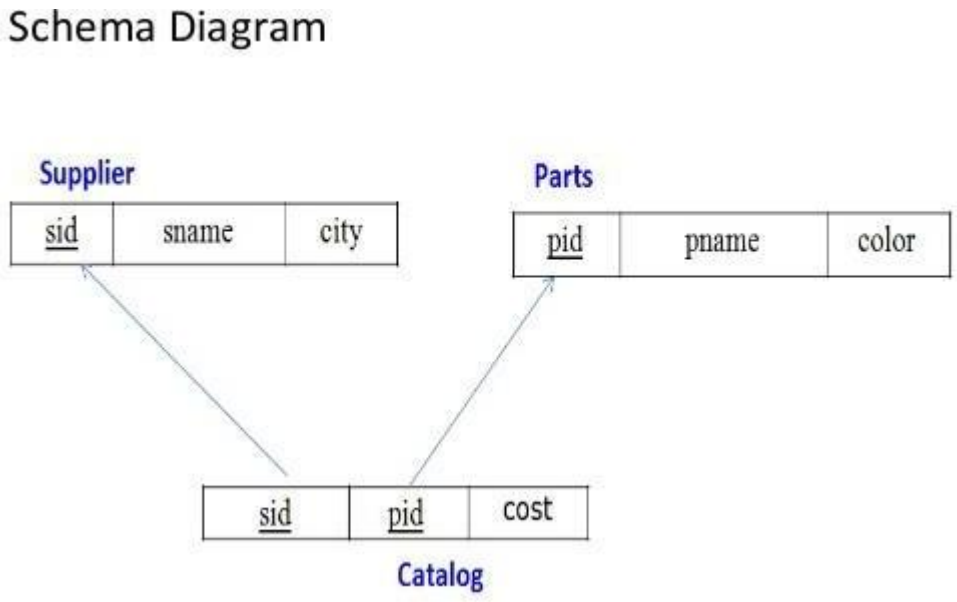
Supplier Database

Question

(Week 7)

1. Using Scheme diagram, Create tables by properly specifying the primary keys and the foreign keys.
2. Insert appropriate records in each table.
3. Find the pnames of parts for which there is some supplier.
4. Find the snames of suppliers who supply every part.
5. Find the snames of suppliers who supply every red part.
6. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
7. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).
8. For each part, find the sname of the supplier who charges the most for that part.

Schema Diagram



Create Database

```
create database supplier_database_080;

use supplier_database_080;
```

Create Table

```
create table Supplier
(
```

```
SID int,
Sname varchar(20),
City varchar(20),
PRIMARY KEY(SID)
);

create table Parts
(
PID int,
Pname varchar(20),
Color varchar(20),
PRIMARY KEY(PID)
);

create table Catalog
(
SID  int,
PID  int,
Cost int,
PRIMARY KEY(SID,PID),
FOREIGN KEY(SID) references Supplier(SID),
FOREIGN KEY(PID) references Parts(PID)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

Structure of the table

desc Supplier;

	Field	Type	Null	Key	Default	Extra
▶	SID	int	NO	PRI	NULL	
	Sname	varchar(20)	YES		NULL	
	City	varchar(20)	YES		NULL	

desc Parts;

	Field	Type	Null	Key	Default	Extra
▶	PID	int	NO	PRI	NULL	
	Pname	varchar(20)	YES		NULL	
	Color	varchar(20)	YES		NULL	

desc Catalog;

	Field	Type	Null	Key	Default	Extra
▶	SID	int	NO	PRI	NULL	
	PID	int	NO	PRI	NULL	
	Cost	int	YES		NULL	

Inserting Values to the table

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;

	SID	Sname	City
▶	10001	Acme Widget	Bangalore
	10002	Johns	Kolkata
	10003	Vimal	Mumbai
	10004	Reliance	Delhi
*	NULL	NULL	NULL

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(20005,'Charger','Black'); select * from Parts;

	PID	Pname	Color
▶	20001	Book	Red
	20002	Pen	Red
	20003	Pencil	Green
	20004	Mobile	Green
	20005	Charger	Black
*	NULL	NULL	NULL

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(20005,'Charger','Black'); select * from Parts;

	PID	Pname	Color
▶	20001	Book	Red
	20002	Pen	Red
	20003	Pencil	Green
	20004	Mobile	Green
	20005	Charger	Black
*	NULL	NULL	NULL

```
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;
```

	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
*	NULL	NULL	NULL

Queries :

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

select distinct 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 Supplier where
SID NOT IN(select s.SID from Supplier s , Parts p
where p.PID NOT IN(select c.PID from Catalog c where c.SID=s.SID));

	Sname
▶	Acme Widget

Find the snames of suppliers who supply every red part.

select Sname from Supplier where
SID NOT IN(select s.SID from Supplier s , Parts p
where p.Color='Red' and p.PID NOT IN(select c.PID from Catalog c where c.SID=s.SID));

	Sname
▶	Acme Widget
	Johns

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

```
SELECT p.Pname FROM Parts p
JOIN Catalog c ON p.PID = c.PID
JOIN Supplier s ON c.SID = s.SID
WHERE s.Sname = 'Acme Widget'
AND NOT EXISTS (
   SELECT 1 FROM Catalog c1
   JOIN Supplier s1 ON c1.SID = s1.SID
   WHERE c1.PID = p.PID
   AND s1.Sname != 'Acme Widget'
);
```

	Pname
▶	Mobile
	Charger

Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).

```
select distinct c.SID from Catalog c join
(select PID,avg(Cost) as Avg_Cost from Catalog group by PID)
avg_cost_table on c.PID=avg_Cost_table.PID where
c.Cost>avg_Cost_table.Avg_Cost;
```

	SID
▶	10002
	10004

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

```
select p.PID,s.Sname from Supplier s join Catalog c on s.SID=c.SID join
Parts p on c.PID=p.PID where c.Cost=(select max(c2.Cost) from Catalog
c2 where c2.PID=p.PID);
```

	PID	Sname
▶	20001	Acme Widget
	20001	Johns
	20002	Johns
	20003	Reliance
	20004	Acme Widget
	20005	Acme Widget

No SQL Student Database

Question

(Week 8)

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.

Queries:

1. Create a database “Student” with the following attributes Rollno,Age, ContactNo, Email-Id. db.createCollection("Student");

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

Atlas atlas-mozg5o-shard-0 [primary] test> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-mozg5o-shard-0 [primary] test> show dbs
Student   72.00 KiB
test       8.00 KiB
admin    328.00 KiB
local     88.62 GiB
Atlas atlas-mozg5o-shard-0 [primary] test>
```

2. Insert appropriate values

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"});

```
Atlas atlas-okge9d-shard-0 [primary] test> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6746b7a60ffbfb92d32f8e1a") }
}
Atlas atlas-okge9d-shard-0 [primary] test> db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});

{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6746b7fb0ffbfb92d32f8e1b") }
}
Atlas atlas-okge9d-shard-0 [primary] test> db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});

{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6746b8060ffbfb92d32f8e1c") }
}
Atlas atlas-okge9d-shard-0 [primary] test> db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6746b8110ffbfb92d32f8e1d") }
}
Atlas atlas-okge9d-shard-0 [primary] test> db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6746b8180ffbfb92d32f8e1e") }
}
```

3.)Write query to update Email-Id of a student with rollno 10.

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

```
Atlas atlas-okge9d-shard-0 [primary] test> db.Student.update({RollNo:10},{$set:{email:"Abhinav@gmail.com"}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

4. Replace the student name from “ABC” to “FEM” of rollno 11.

db.Student.insert({RollNo:11,Age:22,Name:"ABC",Cont:2276,email:"rea.de9@gmail.com"});
db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})

```
Atlas atlas-okge9d-shard-0 [primary] test> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
  {
    _id: ObjectId("63bfd4de56eba0e23c3a5c78"
    RollNo: 11,
    Age: 22,
    Name: 'ABC',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
```

```
{
  _id: ObjectId("63bfd4de56eba0e23c3a5c78"),
  RollNo: 11,
  Age: 22,
  Name: 'FEM',
  Cont: 2276,
  email: 'rea.de9@gmail.com'
}
]
```

1	_Id	RollNo	Age	Cont	email	Name
2	6746b6c4f73fea43f1	1	21	9876	antara.de9@gmail.com	
3	6746b6cbf73fea43f1	2	22	9976	anushka.de9@gmail.com	
4	6746b6d2f73fea43f1	3	21	5576	anubhav.de9@gmail.com	
5	6746b6d8f73fea43f1	4	20	4476	pani.de9@gmail.com	
6	6746b6def73fea43f1	10	23	2276	Abhinav@gmail.com	
7	6746b710f73fea43f1	11	22	2276	rea.de9@gmail.com	FEM

NoSQL Customer Database

Question

(Week 9)

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

Cust_id, Acc_Bal, Acc_Type

2. Insert at least 5 values into the table
3. Write a query to display those records whose total account balance is greater than 1200 of account type ‘Z’ for each customer_id.
4. Determine Minimum and Maximum account balance for each customer_id. 5. Export the created collection into local file system
6. Drop the table.
7. Import a given csv dataset from local file system into mongodb collection.

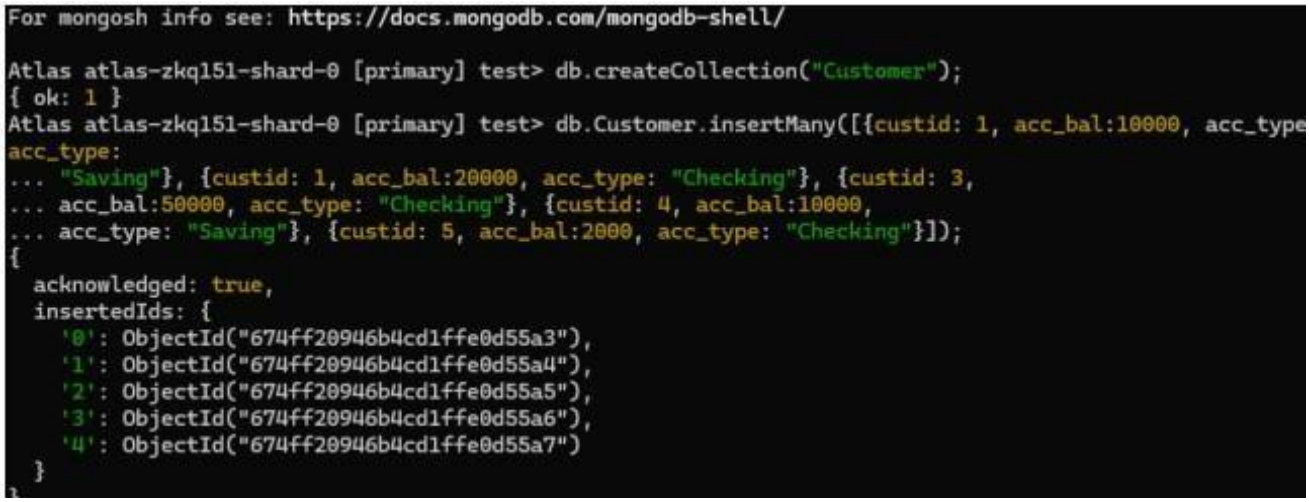
QUERIES

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

Cust_id, Acc_Bal, Acc_Type.

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

```
db.Customer.insertMany([{custid: 1, acc_bal:10000, acc_type:
"Saving"}, {custid: 1, acc_bal:20000, acc_type: "Checking"}, {custid: 3,
acc_bal:50000, acc_type: "Checking"}, {custid: 4, acc_bal:10000,
acc_type: "Saving"}, {custid: 5, acc_bal:2000, acc_type: "Checking"}]);
```



2. Write a query to display those records whose total account balance is greater than 12000 of account type ‘Z’ for each customer_id.

```
db.Customer.find({acc_bal: {$gt: 12000}, acc_type:"Checking"});
```

```
Atlas atlas-zkq151-shard-0 [primary] test> db.Customer.find({acc_bal: {$gt: 12000}, acc_type:"Checking
"});
[
  {
    _id: ObjectId("674ff20946b4cd1ffe0d55a4"),
    custid: 1,
    acc_bal: 20000,
    acc_type: 'Checking'
  },
  {
    _id: ObjectId("674ff20946b4cd1ffe0d55a5"),
    custid: 3,
    acc_bal: 50000,
    acc_type: 'Checking'
  }
]
```

3. Determine Minimum and Maximum account balance for each customer_id.

db.Customer.aggregate([{$group:{_id:"$custid", minBal:{$min:"$acc_bal"}, maxBal: {$max:"$acc_bal"}}}]);

```
Atlas atlas-zkq151-shard-0 [primary] test> db.Customer.aggregate([{$group:{_id:"$custid", minBal:{$min
$min:"$acc_bal"}, maxBal:
... {$max:"$acc_bal"}}}]);
[
  { _id: 5, minBal: 2000, maxBal: 2000 },
  { _id: 3, minBal: 50000, maxBal: 50000 },
  { _id: 4, minBal: 10000, maxBal: 10000 },
  { _id: 1, minBal: 10000, maxBal: 20000 }
]
```

4. Export the created collection into local file system

5. Drop the table

db.Customer.drop();

```
test> db.Customer.drop();
true
```

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

	_id	custid	acc_bal	acc_type
2	674ff20946b4cd1ffe	1	10000	Saving
3	674ff20946b4cd1ffe	1	20000	Checking
4	674ff20946b4cd1ffe	3	50000	Checking
5	674ff20946b4cd1ffe	4	10000	Saving
6	674ff20946b4cd1ffe	5	2000	Checking

NoSQL Restaurant Database

Question

(Week 10)

1. Write a MongoDB query to display all the documents in the collection restaurants.
2. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.
3. Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which achieved a score which is not more than 10.
4. Write a MongoDB query to find the average score for each restaurant.
5. Write a MongoDB query to find the name and address of the restaurants that have a zipcode that starts with '10'.

QUERIES

1. In MongoDB create a collection for "Restaurant" and insert atleast five records

db.createCollection("restaurants");

{ name: "Meghna Foods", town: "Jayanagar", cuisine: "Indian", score: 8, address: { zipcode: "10001", street: "Jayanagar"} },{ name: "Empire", town: "MG Road", cuisine: "Indian", score: 7, address: { zipcode: "10100", street: "MG Road" } },{ name: "Chinese WOK", town: "Indiranagar", cuisine: "Chinese", score: 12, address: { zipcode: "20000", street: "Indiranagar" } },{ name: "Kyotos", town: "Majestic", cuisine: "Japanese", score: 9, address: { zipcode: "10300", street: "Majestic" } },{ name: "WOW Momos", town: "Malleshwaram", cuisine:
"Indian", score: 5, address: { zipcode: "10400", street: "Malleshwaram" }}])

```
Atlas atlas-zkq151-shard-0 [primary] test> db.createCollection("restaurants");
{ ok: 1 }
```

```
Atlas atlas-zkq151-shard-0 [primary] test> db.restaurants.insertMany([
...   {name: "Meghna Foods",town: "Jayanagar",cuisine: "Indian",score: 8,address: {zipcode: "10001",street: "Jayanagar"}},
...   {name: "Empire",town: "MG Road",cuisine: "Indian",score: 7,address: {zipcode: "10100",street: "MG Road"}},
...   {name: "Chinese WOK",town: "Indiranagar",cuisine: "Chinese",score: 8,address: {zipcode: "20000",street: "Indiranagar"}},
...   {name: "Kyotos",town: "Majestic",cuisine: "Japanese",score: 9,address: {zipcode: "10300",street:
 "Majestic"}},
...   {name: "WOW Momos",town: "Malleshwaram",cuisine: "Indian",score: 5,address: {zipcode: "10400",street: "Malleshwaram"}}
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("674ff54346b4cd1ffe0d55a8"),
    '1': ObjectId("674ff54346b4cd1ffe0d55a9"),
    '2': ObjectId("674ff54346b4cd1ffe0d55aa"),
    '3': ObjectId("674ff54346b4cd1ffe0d55ab"),
    '4': ObjectId("674ff54346b4cd1ffe0d55ac")
  }
}
```

2. Write a MongoDB query to display all the documents in the collection restaurants.

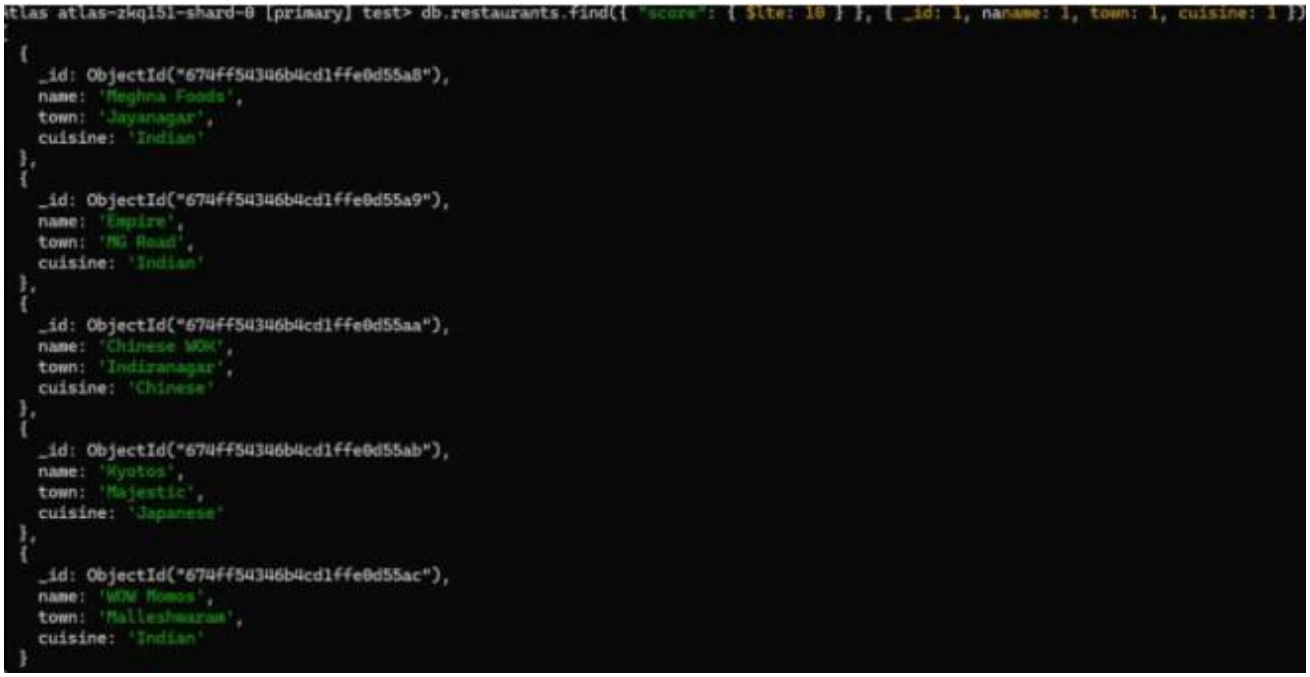
db.restaurants.find({})

3. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

db.restaurants.find({}).sort({ name: -1 })

4. Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which achieved a score which is not more than 10.

db.restaurants.find({ "score": { $lte: 10 } }, { _id: 1, name: 1, town: 1, cuisine: 1 })



5. Write a MongoDB query to find the average score for each restaurant.

db.restaurants.aggregate([{ $group: { _id: "$name", average_score: { $avg: "$score" } } }])

```
Atlas atlas-zkql5l-shard-0 [primary] test> db.restaurants.aggregate([ { $group: { _id: "$name", average_score: { $avg: "$score" } } }
... ])
[
  { _id: 'Chinese WOK', average_score: 8 },
  { _id: 'Kyotos', average_score: 9 },
  { _id: 'Meghna Foods', average_score: 8 },
  { _id: 'WOW Momos', average_score: 5 },
  { _id: 'Empire', average_score: 7 }
]
```

6. Write a MongoDB query to find the name and address of the restaurants that have a zipcode that starts with '10'.

db.restaurants.find({ "address.zipcode": /^10/ }, { name: 1, "address.street": 1, _id: 0 })

```
Atlas atlas-zkql5l-shard-0 [primary] test> db.restaurants.find({ "address.zipcode": /^10/ }, { name: 1, "address.street": 1, _id: 0 })
[
  { name: 'Meghna Foods', address: { street: 'Jayanagar' } },
  { name: 'Empire', address: { street: 'MG Road' } },
  { name: 'Kyotos', address: { street: 'Majestic' } },
  { name: 'WOW Momos', address: { street: 'Malleshwaram' } }
]
```

	_id	name	town	cuisine	score	address.zipcode	address.street
2	674ff54346b4cd1ffe	Meghna Foods	Jayanagar	Indian	8	10001	Jayanagar
3	674ff54346b4cd1ffe	Empire	MG Road	Indian	7	10100	MG Road
4	674ff54346b4cd1ffe	Chinese WOK	Indiranagar	Chinese	8	20000	Indiranagar
5	674ff54346b4cd1ffe	Kyotos	Majestic	Japanese	9	10300	Majestic
6	674ff54346b4cd1ffe	WOW Momos	Malleshwaram	Indian	5	10400	Malleshwaram

4