VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

On

Database Management System (23CS3PCDBM)

Submitted by

MAKADIA RISHIT DILIPBHAI
(1BM23CS177)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

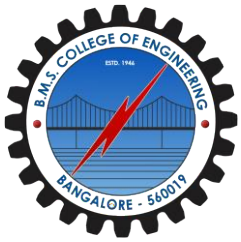
B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sep-2024 to Jan-2025

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 **MAKADIA RISHIT DILIPBHAI (1BM23CS177),** 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.

Saritha A.N. Assistant Professor Department of CSE, BMSCE	Dr. Joythi S NayakProfessor HOD Department of CSE, BMSCE

Index

Insurance Database

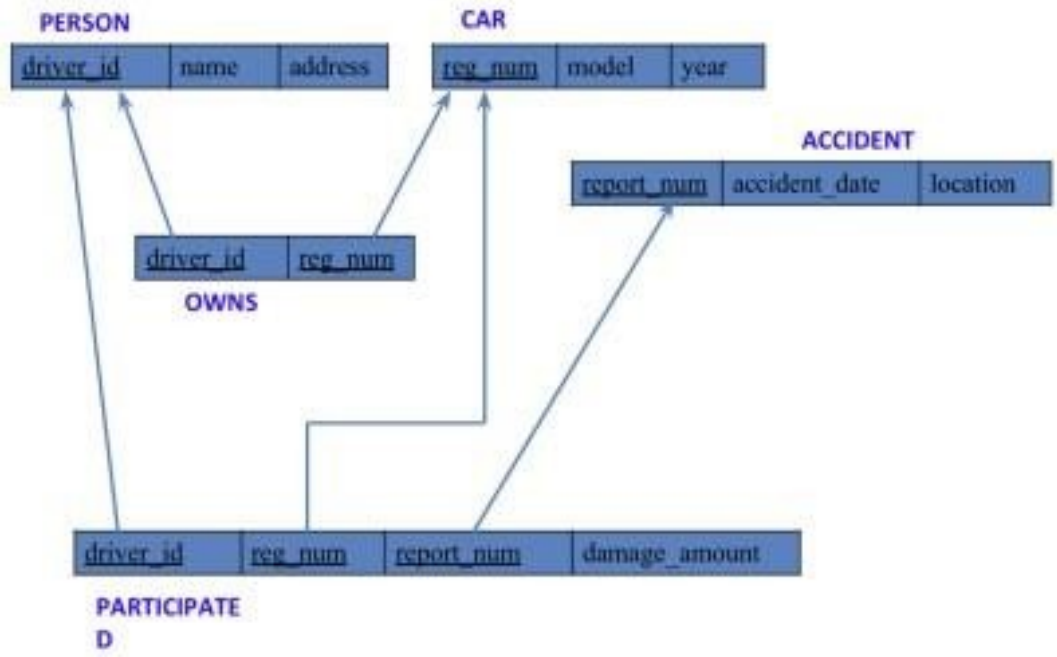
Question (Week 1)

- 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 'KA053408') 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

show databases;

create database IF NOT exists DBMS_3_1;

use DBMS_3_1;

Create Table

create table PERSON (driver_id varchar(3), name_ varchar(10), address varchar(25), PRIMARY KEY(driver_id));

create table CAR (reg_num varchar(10), model varchar(10), reg_year int(4), PRIMARY KEY(reg_num));

create table ACCIDENT (report_no int(2), acc_date varchar(15), location varchar(25), PRIMARY KEY(report_no));

create table OWNS(driver_id varchar(3), reg_num varchar(10), FOREIGN KEY(driver_id) REFERENCES

Field	Type	Null	Key	Default	Extra
driver_id	varchar(3)	NO	PRI	NULL	
name_	varchar(10)	YES		NULL	
address	varchar(25)	YES		NULL	

PERSON(driver_id),
FOREIGN KEY(reg_num) REFERENCES CAR(reg_num));

create table PARTICIPATED (driver_id varchar(3), reg_num varchar(10), report_no int(2), damage_amt int(7),
FOREIGN KEY(driver_id) REFERENCES PERSON(driver_id),
FOREIGN KEY(reg_num) REFERENCES CAR(reg_num),
FOREIGN KEY(report_no) REFERENCES ACCIDENT(report_no));

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(3)	YES	MUL	NULL	
reg_num	varchar(10)	YES	MUL	NULL	

desc PARTICIPATED;

Field	Type	Null	Key	Default	Extra
driver_id	varchar(3)	YES	MUL	NULL	
reg_num	varchar(10)	YES	MUL	NULL	
report_no	int	YES	MUL	NULL	
damage_amt	int	YES		NULL	

Inserting Values to the table

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

select * from person;

driver_id	name_	address
A04	Venu	N R Colony
A03	Smith	Ashok nagar
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A05	John	Hanumanth nagar
NULL	NULL	NULL

insert into CAR values ('KA052250','Indica', 1990),
('KA031181','Lancer', 1957), ('KA095477','Toyota', 1998),
('KA053408','Honda', 2008), ('KA041702','Audi', 2005);
select * from car;

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

insert into ACCIDENT values (11, "01-JAN-03", 'Mysore Road'),
(12, "02-FEB-04", 'South end Circle'), (13, "21-JAN-03", 'Bull Temple Road'),
(14, "17-FEB-08", 'Mysore Road'), (15, "04-MAR-05", 'Kanakpura Road');

select * from accident;

report_no	acc_date	location
11	01-JAN-03	Mysore Road
12	02-FEB-04	South end Circle
13	21-JAN-03	Bull Temple Road
14	17-FEB-08	Mysore Road
15	04-MAR-05	Kanakpura Road
NULL	NULL	NULL

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

drop table owns; select * from owns;

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

insert into PARTICIPATED values ('A01', 'KA052250', 11, 10000),
('A02', 'KA031181', 12, 50000), ('A03', 'KA095477', 13, 25000),
('A04', 'KA053408', 14, 3000),('A05', 'KA041702', 15, 5000);
select * from participated;

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

Queries:

Display driver id who did accident with damage amount greater than or equal to Rs.25000

select driver_id, damage_amt from PARTICIPATED where damage_amt>=25000;

driver_id	damage_amt
A02	50000
A03	25000

Add new accident to the database

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

report_no	acc_date	location
11	01-JAN-03	Mysore Road
12	02-FEB-04	South end Circle
13	21-JAN-03	Bull Temple Road
14	17-FEB-08	Mysore Road
15	04-MAR-05	Kanakpura Road
16	2008-03-08	Dolmor

Display Accident date and location

select acc_date,location from ACCIDENT;

acc_date	location
01-JAN-03	Mysore Road
02-FEB-04	South end Circle
21-JAN-03	Bull Temple Road
17-FEB-08	Mysore Road
04-MAR-05	Kanakpura Road
2008-03-08	Dolmor

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_amt = 25000 WHERE reg_num = 'KA053408' AND report_no = 14;
select * from PARTICIPATED;

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

(Week 2)

More Queries on Insurance Database:

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

SELECT * FROM CAR ORDER BY reg_year ASC;

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

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

SELECT C.model, COUNT(*) AS num_accidents FROM PARTICIPATED P JOIN CAR C ON P.reg_num = C.reg_num WHERE C.model = 'Lancer';

model	num_a...
Lancer	1

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

SELECT COUNT(DISTINCT O.driver_id) AS num_people FROM OWNS O
JOIN PARTICIPATED P ON O.reg_num = P.reg_num JOIN ACCIDENT A ON P.report_no = A.report_no WHERE A.acc_date LIKE '%-03';

num_people
2

List all the entire participated relation in descending order of damage_amount

select * FROM participated ORDER BY damage_amt desc;

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

Find average damage_amount

select avg(damage_amt) from participated;

avg(damage_amt)
23000.0000

Delete the tuple whose damage_amount is below average amount damage_amount

DELETE FROM PARTICIPATED WHERE damage_amt < (SELECT AVG(damage_amt) FROM PARTICIPATED);

SELECT * FROM PARTICIPATED WHERE damage_amt < (SELECT AVG(damage_amt) FROM PARTICIPATED);

driver_id	reg_num	report_no	damage_amt
A01	KA052250	11	10000
A05	KA041702	15	5000

List the name of drivers whose damage is greater than the avgdamage_amount

SELECT P.name_ FROM PERSON P JOIN PARTICIPATED T ON P.driver_id = T.driver_id WHERE T.damage_amt > (SELECT AVG(damage_amt) FROM PARTICIPATED);

name_
Pradeep
Smith
Venu

Find the maximum damage_amount

select max(damage_amt) from participated;

max(damage_amt)
50000

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)

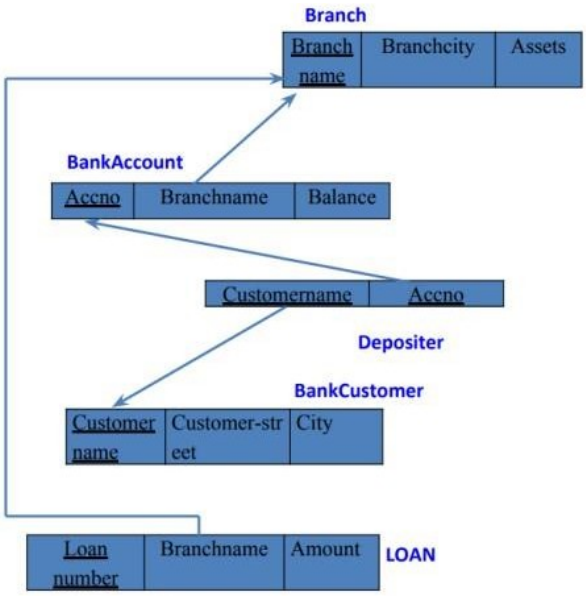
Create the above tables by properly specifying the primary keys and the foreign keys. Enter atleast five tuples for each relation.

Display the branch name and assets from all branches in lakhs of rupees and rename the assetscolumn 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 IF NOT exists DBMS_3_2;

use DBMS_3_2;

Create Table

create table Branch(BranchName varchar(25), BranchCity varchar(10), Assets int(7),
PRIMARY KEY (BranchName));

create table BankAcc(AccNo int(2),BranchName varchar(25), Balance int(5),
PRIMARY KEY (AccNo),
FOREIGN KEY (BranchName) REFERENCES Branch(BranchName));

create table BankCustomer(CustomerName varchar(10), CustomerStreet varchar(25), City varchar(10),
PRIMARY KEY (CustomerName));

create table Depositor(CustomerName varchar(10),AccNo int(7),
FOREIGN KEY (AccNo) REFERENCES BankAcc(AccNo),
FOREIGN KEY (CustomerName) REFERENCES BankCustomer(CustomerName));

create table Loan(LoanNo int(3), BranchName varchar(25), Ammount int(7),
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),
('SBI_ResidencyRoad', 'Bangalore', 10000), ('SBI_ShivajiRoad', 'Bombay', 20000),
('SBI_ParliamentRoad', 'Delhi', 10000), ('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 BankAcc values (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);
select * from BankAcc;

AccNo	BranchName	Balance
1	SBI_Chamrajpet	2000.00
2	SBI_ResidencyRoad	5000.00
3	SBI_ShivajiRoad	6000.00
4	SBI_ParliamentRoad	9000.00
5	SBI_JantarMantar	8000.00
6	SBI_ShivajiRoad	4000.00
8	SBI_ResidencyRoad	4000.00
9	SBI_ParliamentRoad	3000.00
10	SBI_ResidencyRoad	5000.00
11	SBI_JantarMantar	2000.00
NULL	NULL	NULL

insert into BankCustomer values ('Avinash', 'Bull_Temple_Road', 'Bangalore'),
('Dinesh', 'Baneargatta_Road', 'Bangalore'), ('Mohan', 'NationalCollege_Road', 'Bangalore'),
('Nikhil', 'Prithvi_Road', 'Delhi'), ('Ravi', 'Abrar_Road', 'Delhi');
select * from BankCustomer;

CustomerName	CustomerStreet	City
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Baneargatta_Road	Bangalore
Mohan	NationalCollege_Road	Bangalore
Nikhil	Prithvi_Road	Delhi
Ravi	Abrar_Road	Delhi
NULL	NULL	NULL

insert into Depositor values ('Avinash', 1), ('Dinesh', 2),
('Nikhil', 4), ('Ravi', 5), ('Avinash', 8),
('Nikhil', 9), ('Dinesh', 10), ('Nikhil', 11);

select * from Depositor;

CustomerName	AccNo
Avinash	1
Dinesh	2
Nikhil	4
Ravi	5
Avinash	8
Nikhil	9
Dinesh	10
Nikhil	11

insert into Loan values (1, 'SBI_Chamrajpet', 10000),
(2, 'SBI_ResidencyRoad', 20000), (3, 'SBI_ShivajiRoad', 30000),
(4, 'SBI_ParliamentRoad', 40000), (5, 'SBI_JantarMantar', 50000);
select * from Loan;

LoanNo	BranchName	Ammount
1	SBI_Chamrajpet	10000
2	SBI_ResidencyRoad	20000
3	SBI_ShivajiRoad	30000
4	SBI_ParliamentRoad	40000
5	SBI_JantarMantar	50000

Queries:

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

select BranchName , Assets/100000.0 as Assets_in_Lakh from Branch;

BranchName	Assets_in_Lakh
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, b.BranchName, COUNT(d.AccNo) AS NumOfAccounts FROM Depositor d JOIN BankAcc b ON d.AccNo = b.AccNo GROUP BY d.CustomerName, b.BranchName HAVING COUNT(d.AccNo) >= 2;

CustomerName	BranchName	NumOfAccounts
Dinesh	SBI_ResidencyRoad	2
Nikhil	SBI_ParliamentRoad	2

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

CREATE VIEW BranchLoanSummary AS

SELECT BranchName, SUM(Ammount) AS TotalLoanAmount FROM Loan GROUP BY BranchName;

SELECT * FROM BranchLoanSummary;

BranchName	TotalLoanAmou...
SBI_Chamrajpet	10000
SBI_JantarMantar	50000
SBI_ParliamentRoad	40000
SBI_ResidencyRoad	20000
SBI_ShivajiRoad	30000

(Week 4)

More Queries on Bank Database:

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

```
SELECT C.CustomerName FROM BankCustomer C JOIN Depositor D ON C.CustomerName =
D.CustomerName
JOIN BankAcc A ON D.AccNo = A.AccNo JOIN Branch B ON A.BranchName = B.BranchName WHERE
B.BranchCity = 'Delhi'
GROUP BY C.CustomerName HAVING COUNT(DISTINCT B.BranchName) = (SELECT COUNT(*) FROM
                    Branch WHERE BranchCity = 'Delhi');
```

CustomerName
Nikhil

Find all customers who have a loan at the bank but do nothave an account.

```
SELECT C.CustomerName FROM BankCustomer C JOIN Loan L ON C.CustomerName = L.BranchName
WHERE C.CustomerName NOT IN (SELECT D.CustomerName FROM Depositor D);
```

CustomerName

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

```
SELECT DISTINCT C.CustomerName FROM BankCustomer C JOIN Depositor D ON C.CustomerName =
D.CustomerName JOIN BankAcc A ON D.AccNo = A.AccNo
JOIN Branch B1 ON A.BranchName = B1.BranchName JOIN Loan L ON L.BranchName = B1.BranchName
WHERE B1.BranchCity = 'Bangalore';
```

CustomerName
Avinash
Dinesh

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

SELECT B1.BranchName FROM Branch B1
WHERE B1.Assets > ALL (SELECT B2.Assets FROM Branch B2 WHERE B2.BranchCity = 'Bangalore');

BranchName
NULL

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

SELECT * FROM BankAcc WHERE BranchName IN (SELECT BranchName FROM Branch WHERE BranchCity = 'Bombay');

AccNo	BranchName	Balance
3	SBI_ShivajiRoad	6000.00
6	SBI_ShivajiRoad	4000.00
NULL	NULL	NULL

Update the Balance of all accounts by 5%.

UPDATE BankAcc SET Balance = Balance * 1.05;
SELECT * FROM BankAcc;

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)

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

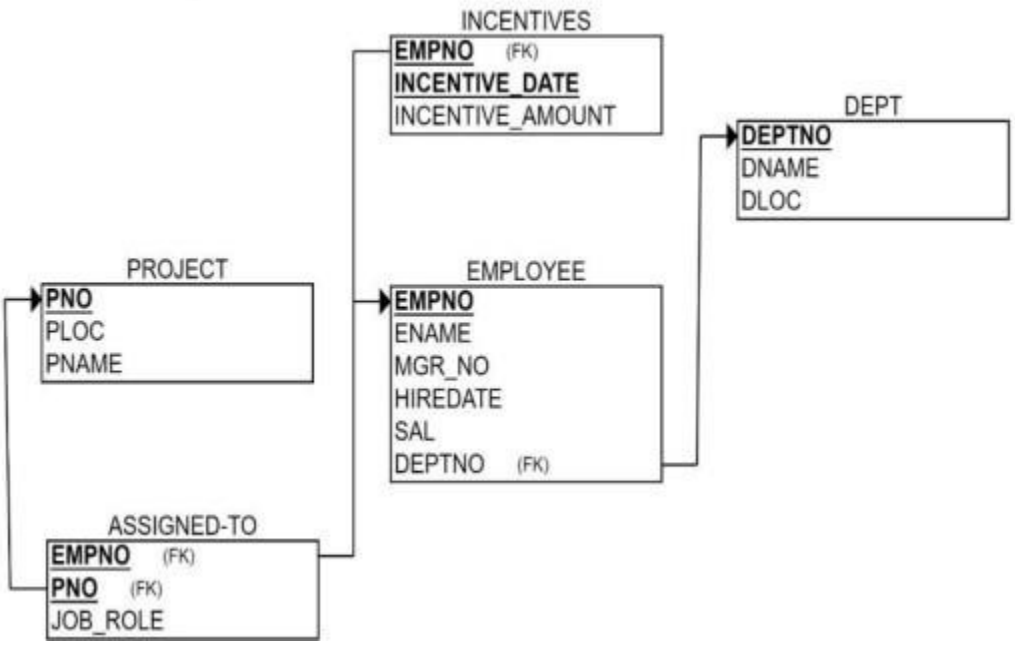
Enter greater than five tuples for each table.

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

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

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

Schema Diagram



Create Database

```
create database DBMS_3_3;
use DBMS_3_3;
```

Create Table

```
create table DEPT(DeptNo int(5) , Dname varchar(25), Dloc varchar(25), PRIMARY KEY(DeptNo));


create table PROJECT(Ploc varchar(25), Pname varchar(25), Pno int(5),
PRIMARY KEY(Pno));


create table EMPLOYEE(EmpNo int(5), Ename varchar(25), MGR_No int(10),
HireDate varchar(10), Salary int(10), DeptNo int(5),
PRIMARY KEY(EmpNo),
FOREIGN KEY(DeptNo) REFERENCES DEPT(DeptNo));


create table INCENTIVES(EmpNo int(5), IncDate varchar(10), IncAmount int(10), PRIMARY
KEY(EmpNo, IncDate),
FOREIGN KEY(EmpNo) REFERENCES EMPLOYEE(EmpNo));


create table AssignedTo(Pno int(5),EmpNo int(5), JobRole varchar(25),
FOREIGN KEY(Pno) REFERENCES PROJECT(Pno),
FOREIGN KEY(EmpNo) REFERENCES EMPLOYEE(EmpNo));
```

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 DEPT values (1, "Manager", "DEL"),
(2, "Director", "BLR"), (3, "Executive", "DEL"),
(4, "Director", "DEL"),(5, "Manager", "BLR");
select * from Dept;

DeptNo	Dname	Dloc
1	Manager	DEL
2	Director	BLR
3	Executive	DEL
4	Director	DEL
5	Manager	BLR
NULL	NULL	NULL

insert PROJECT values ("BLR", "A to B", 3),

("DEL", "D to E", 1),("HYD", "R to O", 2),

("DEL", "G to H", 5),("BLR", "P to Q", 4);

select * from Project;

Ploc	Pname	Pno
DEL	D to E	1
HYD	R to O	2
BLR	A to B	3
BLR	P to Q	4
DEL	G to H	5
NULL	NULL	NULL

insert EMPLOYEE values (121, "Ria", 122, "21/10/2018", 30000, 4),

(122, "Kia", 121, "21/11/2020", 20000, 4),(123, "Jiya", 121, "1/10/2008", 70000, 4),

(124, "Lia", 122, "21/1/2019", 35000, 4),(125, "Piya", 121, "27/5/2017", 40000, 4);
select * from Employee;

EmpNo	Ename	MGR_No	HireDate	Salary	DeptNo
121	Ria	51	21/10/2018	30000	4
122	Kia	57	21/11/2020	20000	4
123	Jiya	77	1/10/2008	70000	4
124	Lia	63	21/1/2019	35000	4
125	Piya	71	27/5/2017	40000	4
NULL	NULL	NULL	NULL	NULL	NULL

insert INCENTIVES values (123, "12/3/2021", 20000),
(122, "21/3/2022", NULL),(124, "18/7/2023", 10000),
(125, "12/12/2022", 17000),(121, "11/5/2023", NULL);
select * from Incentives;

EmpNo	IncDate	IncAmount
121	11/5/2023	NULL
122	21/3/2022	NULL
123	12/3/2021	20000
124	18/7/2023	10000
125	12/12/2022	17000
NULL	NULL	NULL

insert AssignedTo values (3, 123, "Manager"),

(2, 122, "Director"),(5, 121, "Manager"),

(1, 124, "Executive"),(4, 125, "Director");

select * from AssignedTo;

Pno	EmpNo	JobRole
3	123	Manager
2	122	Director
5	121	Manager
1	124	Executive
4	125	Director

Queries :

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

select distinct a.EmpNo, p.Ploc from AssignedTo a
join PROJECT p on a.Pno = p.Pno where p.Ploc in ('BLR', 'HYD', 'MYS');

EmpNo	Ploc
122	HYD
123	BLR
125	BLR

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

SELECT e.EmpNo, i.IncAmount, e.Ename FROM EMPLOYEE e
LEFT JOIN INCENTIVES i ON e.EmpNo = i.EmpNo WHERE i.IncAmount IS NULL;

EmpNo	IncAmount	Ename
121	NULL	Ria
122	NULL	Kia

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

SELECT E.Ename, E.EmpNo, D.Dname AS Dept, A.JobRole, D.Dloc AS DeptLocation, P.Ploc AS ProjectLocation FROM EMPLOYEE E

JOIN DEPT D ON E.DeptNo = D.DeptNo JOIN AssignedTo A ON E.EmpNo = A.EmpNo JOIN PROJECT P ON A.Pno = P.Pno

WHERE D.Dloc = P.Ploc;

Ename	EmpNo	Dept	JobRole	DeptLocati...	ProjectLocati...
Ria	121	Director	Manager	DEL	DEL
Lia	124	Director	Executive	DEL	DEL

(Week 6)

More Queries on Employee Database:

List the name of the managers with the maximum employees.

SELECT Ename FROM EMPLOYEE WHERE EmpNo = (SELECT MGR_No FROM EMPLOYEE GROUP BY MGR_No ORDER BY COUNT(*) DESC LIMIT 1);

Ename

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

SELECT E1.Ename AS ManagerName FROM EMPLOYEE E1 WHERE E1.Salary > (SELECT AVG(E2.Salary) FROM EMPLOYEE E2 WHERE E2.MGR_No = E1.EmpNo);

ManagerName

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

SELECT D.Dname, RankedManagers.Ename FROM (SELECT D.DeptNo, E.Ename, RANK() OVER (PARTITION BY D.DeptNo ORDER BY E.Salary DESC) AS Ranks
FROM EMPLOYEE E JOIN DEPT D ON E.DeptNo = D.DeptNo WHERE E.MGR_No IS NOT NULL)
AS RankedManagers JOIN DEPT D ON RankedManagers.DeptNo = D.DeptNo WHERE RankedManagers.Ranks = 2;

	Dname	Ename
	Director	Piya

Find the employee details who got second maximum incentive in November2024.

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

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

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

SELECT E1.Ename AS EmployeeName FROM EMPLOYEE E1 JOIN EMPLOYEE E2 ON E1.MGR_No = E2.EmpNo WHERE E1.DeptNo = E2.DeptNo;

EmployeeName

Supplier Database

Question(Week 7)

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

Insert appropriate records in each table.

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

Find the snames of suppliers who supply every part.

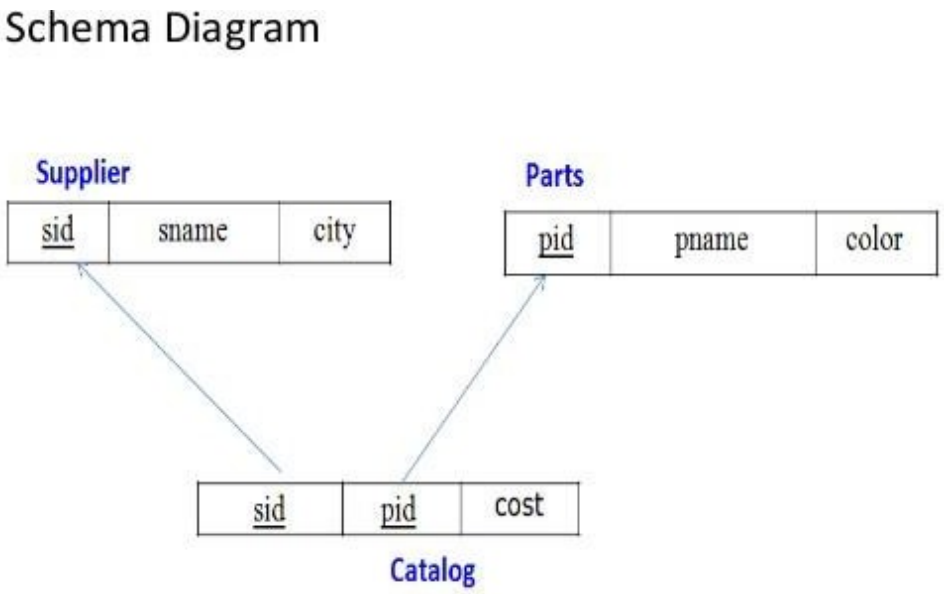
Find the snames of suppliers who supply every red part.

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

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

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

Schema Diagram



Create Database

create database DBMS_3_4;

use DBMS_3_4;

Create Table

create table Suppliers (Sid int(10), Sname varchar(25), City varchar(15),
PRIMARY KEY (Sid));

create table Parts (Pid int(10), Pname varchar(25), Color varchar(15),
primary key (Pid));

create table Catalog (Sid int(10), Pid int(10), Cost int(10),
foreign key (Sid) references Suppliers(Sid),
foreign key (Pid) references Parts(Pid));

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 Suppliers value (10001, "Acme Widget", "Bangalore"),
(10002, "Johns", "Kolkata"),(10003, "Vimal", "Mumbai"),
(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 Parts value (20001, "Book", "Red"),

(20002, "Pen", "Red"),(20003, "Pencil", "Green"),

(20004, "Mobile", "Green"),(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 Catalog values (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);
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

Queries :

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

select distinct p.pname from Parts p, Catalog c where p.Pid = c.Pid;

pname
Book
Pen
Pencil
Mobile
Charger

Find the snames of suppliers who supply every part.

select s.sname from Suppliers s JOIN Catalog c ON s.sid = c.sid GROUP BY s.sname
HAVING COUNT(DISTINCT c.pid) = (SELECT COUNT(*) FROM Parts);

	Sname
▶	Acme Widget

Find the snames of suppliers who supply every red part.

select s.sname from Suppliers s JOIN Catalog c ON s.sid = c.sid JOIN Parts p ON c.pid = p.pid
WHERE p.color = 'Red' GROUP BY s.sid, s.sname HAVING COUNT(DISTINCT p.pid) = (select
COUNT(*) from Parts WHERE color = 'Red');

	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 Suppliers S ON C.Sid =
S.Sid
WHERE S.Sname = 'Acme Widget' GROUP BY P.Pname HAVING COUNT(DISTINCT C.Sid) = 1;

	Pname
▶	Mobile
	Charger

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

select distinct c.sid from Catalog c where c.cost > (select AVG(c1.cost) from Catalog c1 where c1.pid = c.pid);

	SID
▶	10002
	10004

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

select p.pname, s.sname from Parts p JOIN Catalog c ON p.pid = c.pid JOIN Suppliers s ON s.sid = c.sid
WHERE c.Cost = (select MAX(c1.Cost) from Catalog c1 WHERE c1.pid = p.pid);

pname	sname
Book	Acme Widget
Mobile	Acme Widget
Charger	Acme Widget
Book	Johns
Pen	Johns
Pencil	Reliance

NoSQL Student Database

Question(Week 8)

Perform the following DB operations using MongoDB:-

Create a database “Student” with the following attributes Rollno,Age, ContactNo, Email-Id.

Insert appropriate values

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

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

Queries:

Create a database “Student” with the following attributes Rollno,Age, ContactNo, Email-Id.

db.createCollection("Student");

```
> use test
< already on db test
> use NOSQL1
< switched to db NOSQL1
> db.createCollection("Student");
< { ok: 1 }
```

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"});

```
> 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"});
< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('674aad9e84d77b23c9bc4bc4')
    }
  }
```

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

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

```
> 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
  }
```

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"}})

```
> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

```
{
  _id: ObjectId('674aadbe84d77b23c9bc4bc5'),
  RollNo: 11,
  Age: 22,
  Name: 'FEM',
  Cont: 2276,
  email: 'rea.de9@gmail.com'
}
```

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

_id__$oid	RollNo	Age	Name	Cont	email
674aad9d84d77b23c9bc4bc0	1	21		9876	antara.de9@gmail.com
674aad9d84d77b23c9bc4bc1	2	22		9976	anushka.de9@gmail.com
674aad9d84d77b23c9bc4bc2	3	21		5576	anubhav.de9@gmail.com
674aad9d84d77b23c9bc4bc3	4	20		4476	pani.de9@gmail.com
674aad9e84d77b23c9bc4bc4	10	23		2276	Abhinav@gmail.com
674aadbe84d77b23c9bc4bc5	11	22	FEM	2276	rea.de9@gmail.com

NoSQL Customer Database

Question(Week 9)

Create a collection by name Customers with the following attributes.

Cust_id, Acc_Bal, Acc_Type

Insert at least 5 values into the table

Write a query to display those records whose total account balance is greater than 1200 ofaccount type ‘Z’ for each customer_id.

Determine Minimum and Maximum account balance for each customer_id. 5. Export thecreated collection into local file system

Drop the table.

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

QUERIES

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"}]);

```
> use NOSQL2;
< switched to db NOSQL2
> db.createCollection("Customers");
< { ok: 1 }
> db.Customers.insertMany([{ Cust_id: "C001", Acc_Bal: 2500, Acc_Type: "Savings" },
    { Cust_id: "C002", Acc_Bal: 12000, Acc_Type: "Current" },
    { Cust_id: "C003", Acc_Bal: 8000, Acc_Type: "Savings" },
    { Cust_id: "C004", Acc_Bal: 500, Acc_Type: "Current" },
    { Cust_id: "C005", Acc_Bal: 3000, Acc_Type: "Savings" } ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('674d9d56f63dc7b385ea56ff'),
      '1': ObjectId('674d9d56f63dc7b385ea5700'),
      '2': ObjectId('674d9d56f63dc7b385ea5701'),
      '3': ObjectId('674d9d56f63dc7b385ea5702'),
      '4': ObjectId('674d9d56f63dc7b385ea5703')
    }
  }
```

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

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

```
> db.Customers.find({ Acc_Bal: { $gt: 1200 }, Acc_Type: "Savings" },
  { Cust_id: 1, Acc_Bal: 1, _id: 0 });
< {
    Cust_id: 'C001',
    Acc_Bal: 2500
  }
  {
    Cust_id: 'C003',
    Acc_Bal: 8000
  }
  {
    Cust_id: 'C005',
    Acc_Bal: 3000
  }
```

Determine Minimum and Maximum account balance for each customer_id.

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

```
> db.Customers.aggregate([{ $group: { _id: "$Cust_id", Min_Acc_Bal: { $min: "$Acc_Bal" },
  Max_Acc_Bal: { $max: "$Acc_Bal" } }},
  { $project: {_id: 0, Cust_id: "$_id", Min_Acc_Bal: 1, Max_Acc_Bal: 1 } } ]);
< {
```

```
< {
    Min_Acc_Bal: 2500,
    Max_Acc_Bal: 2500,
    Cust_id: 'C001'
  }
  {
    Min_Acc_Bal: 500,
    Max_Acc_Bal: 500,
    Cust_id: 'C004'
  }
```

Export the created collection into local file system

Drop the table

db.Customer.drop();

```
> db.Customers.drop();
< true
```

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

_id__$oid	Cust_id	Acc_Bal	Acc_Type
674d9d56f63dc7b385ea56ff	C001	2500	Savings
674d9d56f63dc7b385ea5700	C002	12000	Current
674d9d56f63dc7b385ea5701	C003	8000	Savings
674d9d56f63dc7b385ea5702	C004	500	Current
674d9d56f63dc7b385ea5703	C005	3000	Savings

NoSQL Restaurant Database

Question (Week 10)

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

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

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

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

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

QUERIES

In MongoDB create a collection for “Restaurant” and insert atleast five records

db.createCollection("restaurants");

```
> db.Restaurant.drop();
< true
> db.createCollection("Restaurant");
< { ok: 1 }
```

db.Restaurant.insertOne({ "address": { "building": "12A", "coord": [77.5946, 12.9716], "street": "Church Street", "zipcode": "560001" }, "borough": "Bangalore", "cuisine": "North Indian", "grades": [{ "date": { "$date": 1630454400000 }, "grade": "A", "score": 3 }, { "date": { "$date": 1599004800000 }, "grade": "A", "score": 5 }], "name": "Royal Punjabi Dhaba", "restaurant_id": "500001" });
db.Restaurant.insertOne({ "address": { "building": "45", "coord": [72.8777, 19.0760], "street": "Marine Drive", "zipcode": "400002" }, "borough": "Mumbai", "cuisine": "Maharashtrian", "grades": [{ "date": { "$date": 1630454400000 }, "grade": "A", "score": 4 }, { "date": { "$date": 1601510400000 }, "grade": "B", "score": 7 }], "name": "Konkan Coastal Kitchen", "restaurant_id": "500002" });

```
< {
    acknowledged: true,
    insertedId: ObjectId('674c9a1645c47af2ccfae472')
  }
```

```
> db.Restaurant.insertOne({
    "address": {
      "building": "12A",
      "coord": [77.5946, 12.9716],
      "street": "Church Street",
      "zipcode": "560001"
    },
    "borough": "Bangalore",
    "cuisine": "North Indian",
    "grades": [
      { "date": ISODate("2021-09-02T00:00:00Z"), "grade": "A", "score": 3 },
      { "date": ISODate("2020-09-04T00:00:00Z"), "grade": "A", "score": 5 }
    ],
    "name": "Royal Punjabi Dhaba",
    "restaurant_id": "500001"
  });

  db.Restaurant.insertOne({
    "address": {
      "building": "45",
      "coord": [72.8777, 19.0760],
      "street": "Marine Drive",
      "zipcode": "400002"
    },
    "borough": "Mumbai",
```

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

db.restaurants.find({})

```
> db.Restaurant.find();
< {
    _id: ObjectId('674c9a1645c47af2ccfae469'),
    address: {
      building: '12A',
      coord: [
        77.5946,
        12.9716
      ],
      street: 'Church Street',
      zipcode: '560001'
    },
    borough: 'Bangalore',
    cuisine: 'North Indian',
    grades: [
```

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

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

```
>_MONGOSH
> db.Restaurant.find().sort({ "name": -1 });
< {
    _id: ObjectId('674c9a1645c47af2ccfae46b'),
```

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 })

```
> db.Restaurant.find({ "grades.score": { $lte: 10 } },
  { "restaurant_id": 1, "name": 1, "borough": 1, "cuisine": 1, "_id": 0 });
< {
    borough: 'Bangalore',
    cuisine: 'North Indian',
    name: 'Royal Punjabi Dhaba',
    restaurant_id: '500001'
  }
  {
    borough: 'Mumbai',
    cuisine: 'Maharashtrian',
    name: 'Konkan Coastal Kitchen',
    restaurant_id: '500002'
  }
  {
    borough: 'Delhi',
    cuisine: 'Mughlai',
    name: 'Zam Zam Biryani',
    restaurant_id: '500003'
  }
```

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

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

```
> db.Restaurant.aggregate([{$unwind: "$grades" },
   {$group: {_id: "$restaurant_id", average_score: { $avg: "$grades.score" }  } } ] );
< {
    _id: '500007',
    average_score: 4.5
  }
  {
    _id: '500005',
    average_score: 6.5
  }
  {
    _id: '500009',
    average_score: 2
  }
  {
    _id: '500001',
    average_score: 4
  }
```

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

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

```
> db.Restaurant.find({ "address.zipcode": /^50/ }, { "name": 1, "address": 1 });
< {
    _id: ObjectId('674c9a1645c47af2ccfae46d'),
    address: {
      building: '9',
      coord: [
        78.4867,
        17.385
      ],
      street: 'Banjara Hills',
      zipcode: '500034'
    },
    name: "Nawab's Biryani House"
  }
```

_id__$oid	addre	address__	address__	address__street	address_	borough	cuisine	grades__date__$date	grade	gra	name	restaurant_id
674c9a1645c47af2ccfae469	12A	77.5946	12.9716	Church Street	560001	Bangalore	North Indian	2021-09-02T00:00:00Z	A	3	Royal Punjabi Dhaba	500001
								2020-09-04T00:00:00Z	A	5		
674c9a1645c47af2ccfae46a	45	72.8777	19.076	Marine Drive	400002	Mumbai	Maharashtrian	2021-09-02T00:00:00Z	A	4	Konkan Coastal Kitchen	500002
								2020-10-01T00:00:00Z	B	7		
674c9a1645c47af2ccfae46b	78	77.1025	28.7041	Chandni Chowk	110006	Delhi	Mughlai	2021-06-01T00:00:00Z	A	2	Zam Zam Biryani	500003
								2020-07-28T00:00:00Z	A	4		
674c9a1645c47af2ccfae46c	17	88.3639	22.5726	Park Street	700016	Kolkata	Bengali	2021-05-01T00:00:00Z	A	3	Bong Flavors	500004
								2020-05-09T00:00:00Z	B	6		
674c9a1645c47af2ccfae46d	9	78.4867	17.385	Banjara Hills	500034	Hyderabad	Hyderabadi	2021-11-02T00:00:00Z	A	5	Nawab's Biryani House	500005
								2020-12-31T00:00:00Z	B	8		
674c9a1645c47af2ccfae46e	34	76.7794	30.7333	Sector 17	160017	Chandigarh	Punjabi	2021-09-02T00:00:00Z	A	4	Amritsari Tadka	500006
								2020-09-04T00:00:00Z	A	5		
674c9a1645c47af2ccfae46f	56	73.8567	18.5204	Fergusson College Road	411004	Pune	South Indian	2021-09-02T00:00:00Z	A	3	Sree Saravana Bhavan	500007
								2020-09-04T00:00:00Z	B	6		
674c9a1645c47af2ccfae470	104	76.9366	8.5241	MG Road	695001	Thiruvananthapuram	Kerala	2021-12-03T00:00:00Z	A	2	Malabar Magic	500008
								2020-10-01T00:00:00Z	A	4		
674c9a1645c47af2ccfae471	88	72.5714	23.0225	C.G. Road	380009	Ahmedabad	Gujarati	2021-07-01T00:00:00Z	A	1	Kathiawadi Junction	500009
								2020-08-07T00:00:00Z	A	3		
674c9a1645c47af2ccfae472	21	74.8792	31.634	Golden Temple Road	143001	Amritsar	Punjabi	2021-09-02T00:00:00Z	A	2	Golden Tandoori	500010
								2020-09-04T00:00:00Z	B	5		