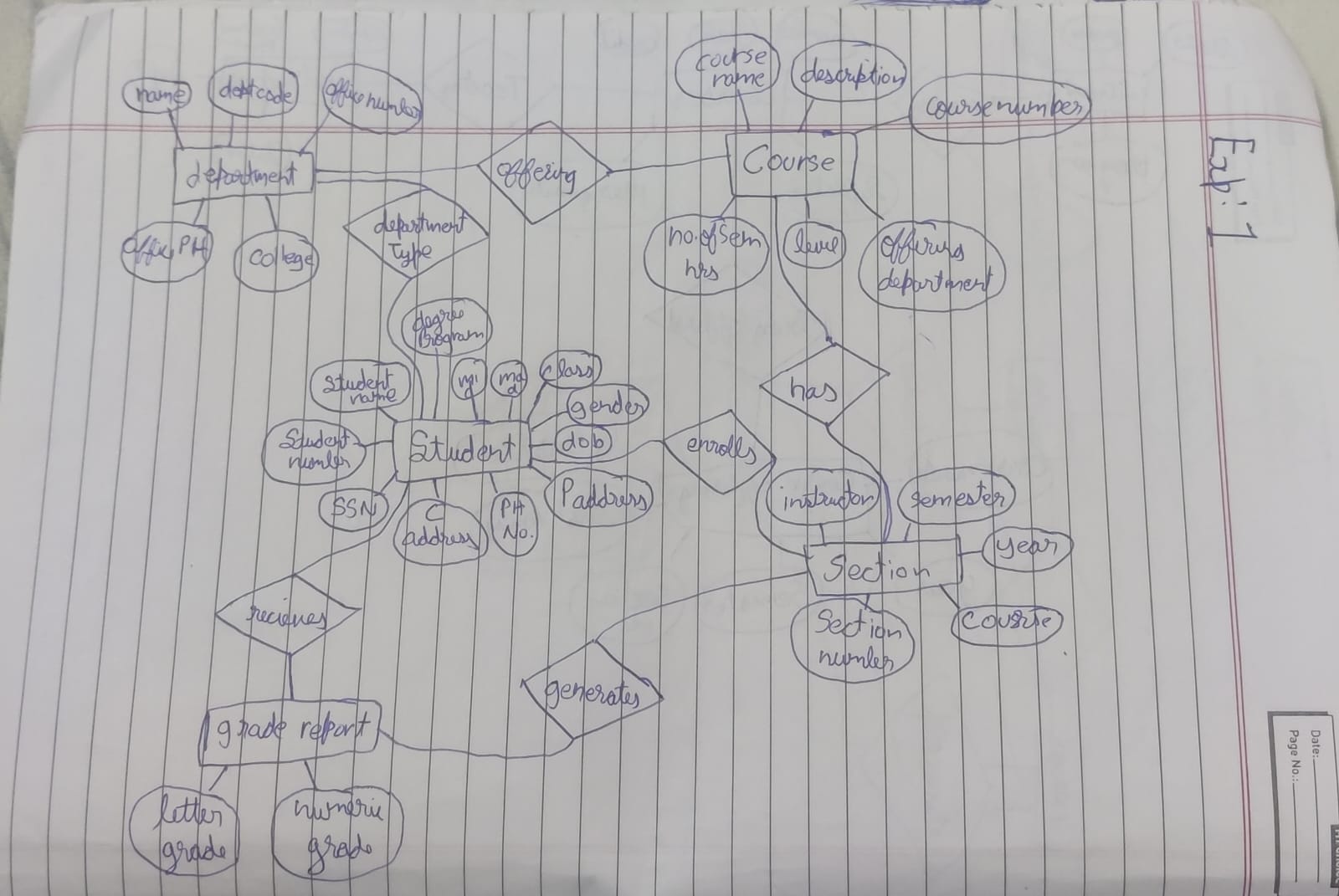
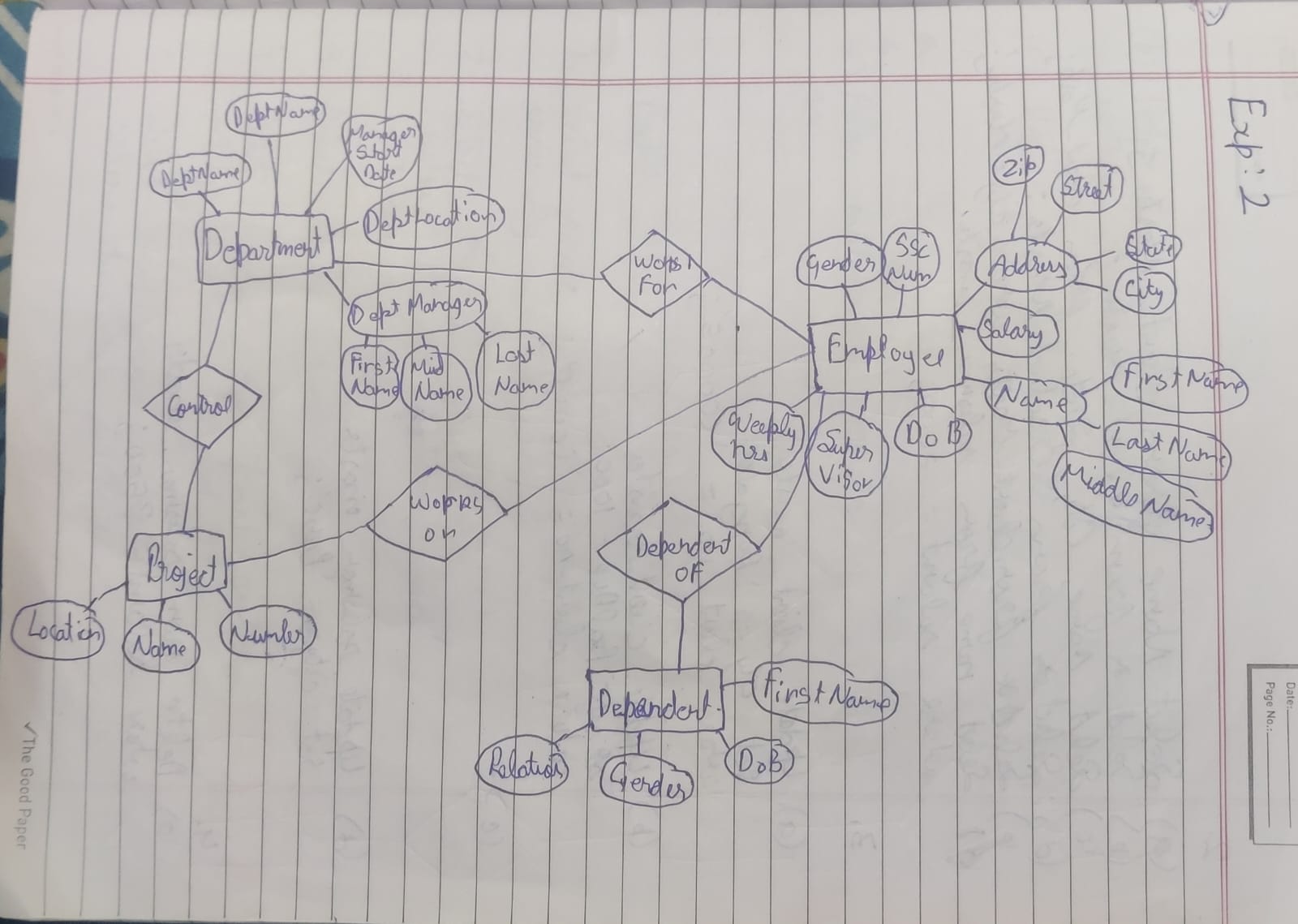
Experiment 1:



Experiment 2:



Experiment 3

Creating Tables

>>> create table employee ( Fname VARCHAR(15) NOT NULL,

-> Minit CHAR,

-> Lname VARCHAR(15) NOT NULL,

-> Ssn CHAR(9) NOT NULL,

-> Bdate DATE,

-> Address VARCHAR(30),

-> Sex CHAR,

-> Salary DECIMAL(10,2),

-> Super\_ssn CHAR(9),

-> Dno INT NOT NULL,

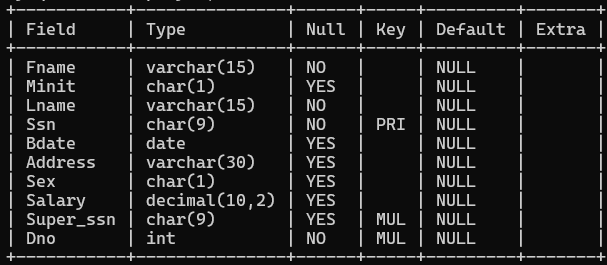
-> PRIMARY KEY (Ssn),

-> FOREIGN KEY (Super\_ssn) REFERENCES EMPLOYEE(Ssn),

-> FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)

-> );

>>> desc employee;



>>> create table DEPARTMENT (Dname VARCHAR(15) NOT NULL,

-> Dnumber INT NOT NULL,

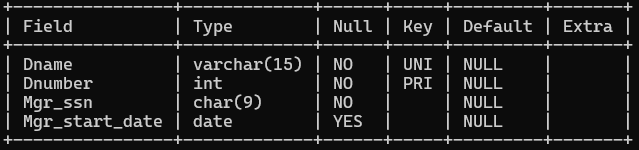
-> Mgr\_ssn CHAR(9) NOT NULL,

-> Mgr\_start\_date DATE,

-> PRIMARY KEY (Dnumber),

-> UNIQUE (Dname) );

>>> desc DEPARTMENT;



>>> create table DEPT\_LOCATIONS

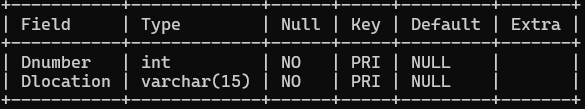
-> ( Dnumber INT NOT NULL,

-> Dlocation VARCHAR(15) NOT NULL,

-> PRIMARY KEY (Dnumber, Dlocation),

-> FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );

>>> desc DEPT\_LOCATIONS;



>>> create table PROJECT

-> ( Pname VARCHAR(15) NOT NULL,

-> Pnumber INT NOT NULL,

-> Plocation VARCHAR(15),

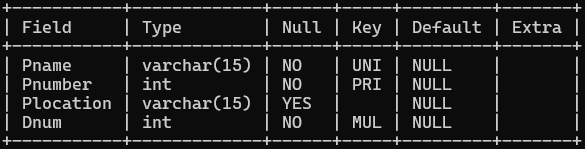
-> Dnum INT NOT NULL,

-> PRIMARY KEY (Pnumber),

-> UNIQUE (Pname),

-> FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );

>>> desc PROJECT;



>>> create table WORKS\_ON

-> ( Essn CHAR(9) NOT NULL,

-> Pno INT NOT NULL,

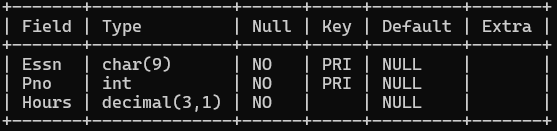
-> Hours DECIMAL(3,1) NOT NULL,

-> PRIMARY KEY (Essn, Pno),

-> FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),

-> FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );

>>> desc WORKS\_ON;



>>> create table DEPENDENT

-> ( Essn CHAR(9) NOT NULL,

-> Dependent\_name VARCHAR(15) NOT NULL,

-> Sex CHAR,

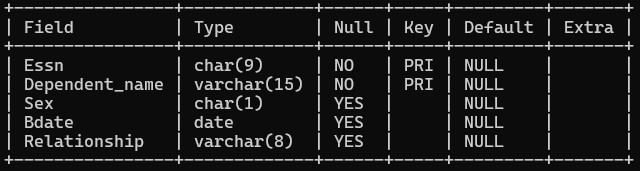
-> Bdate DATE,

-> Relationship VARCHAR(8),

-> PRIMARY KEY (Essn, Dependent\_name),

-> FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );

>>> desc DEPENDENT;

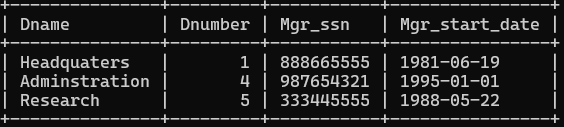


Inserting values into DEPARTMENT

>>> insert into department values ("Research",5,"333445555","1988-05-22");

>>> insert into department values ("Adminstration",4,"987654321","1995-01-01"),("Headquaters",1,"888665555","1981-06-19");

>>> select \* from DEPARTMENT;



Inserting values into Employee

>>> Insert into Employee (Fname, Lname , SSn , Bdate ,Address ,Sex ,Salary, Super\_ssn , Dno )

-> values ("James" ,"Borg" ,"888665555" ,"1937-11-10" ,"450 stone, Houston TX" ,"M" ,"55000" ,NULL , 1);

>>> Insert into Employee (Fname, Lname , SSn , Bdate ,Address ,Sex ,Salary, Super\_ssn , Dno )

-> values ("Jennifer" ,"Wallace","987654321","1941-06-20","291 Berry, Bellaire TX","F","43000","888665555",4);

>>> Insert into Employee (Fname, Lname , SSn , Bdate ,Address ,Sex ,Salary, Super\_ssn , Dno )

-> values ("Alicia" ,"Zelaya","999887777","1968-01-19","3321 castle, Spring TX","F","25000","987654321",4);

>>> Insert into Employee (Fname, Lname , SSn , Bdate ,Address ,Sex ,Salary, Super\_ssn , Dno )

-> values ("Ahmad" ,"Jabbar" ,"987987987" ,"1969-03-29" ,"980 Dallas, Houstan TX","M" ,"25000" ,987654321 , 4);

>>> Insert into Employee (Fname, Lname , SSn , Bdate ,Address ,Sex ,Salary, Super\_ssn , Dno )

-> values ("Franklin" ,"Wong" ,"333445555" ,"1965-12-08" ,"638 Voss, Houstan TX","M","40000","888665555",5);

>>> Insert into Employee (Fname, Lname , SSn , Bdate ,Address ,Sex ,Salary, Super\_ssn , Dno )

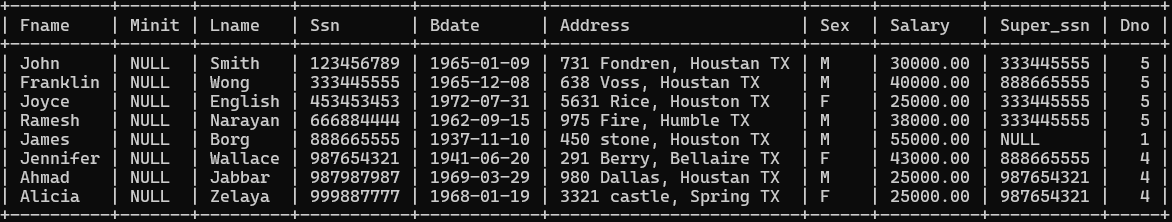
-> values ("Ramesh" ,"Narayan" ,"666884444" ,"1962-09-15" ,"975 Fire, Humble TX","M" ,"38000" ,333445555 , 5);

>>> Insert into Employee (Fname, Lname , SSn , Bdate ,Address ,Sex ,Salary, Super\_ssn , Dno )

-> values ("John" ,"Smith" ,"123456789" ,"1965-01-09" ,"731 Fondren, Houstan TX","M","30000",333445555,5);

>>> Insert into Employee (Fname, Lname , SSn , Bdate ,Address ,Sex ,Salary, Super\_ssn , Dno )

-> values ("Joyce" ,"English" ,"453453453", "1972-07-31" ,"5631 Rice, Houston TX","F","25000",333445555,5);

>>> select \* from Employee;

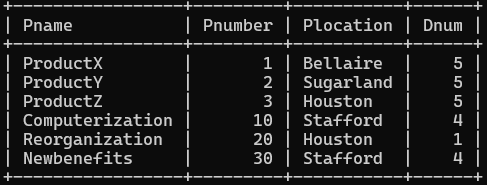
Inserting values into Project

>>> insert into project values ("ProductX",1,"Bellaire" ,5),("ProductY",2,"Sugarland",5),

-> ("ProductZ",3,"Houston",5),("Computerization",10,"Stafford",4),

-> ("Reorganization" ,20,"Houston",1),("Newbenefits",30, "Stafford" ,4);

>>> select \* from project;



Inserting values into WORKS\_ON

>>> insert into works\_on values ("123456789",1,32.5),("123456789",2,7.5),("666884444",3,40.0),

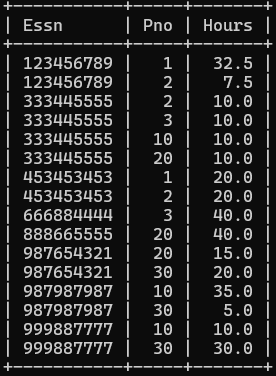
-> ("453453453",1,20.0),("453453453",2,20.0),("333445555",2,10.0),("333445555",3,10.0),

-> ("333445555",10,10.0),("333445555",20,10.0),("999887777",30,30.0),("999887777",10,10.0),

-> ("987987987",10,35.0),("987987987",30,5.0),("987654321",30,20.0),("987654321",20,15.0),

-> ("888665555",20,40.0);

>>> select \* from WORKS\_ON;



Inserting values into DEPENDENT

>>> insert into dependent values ("333445555","Alice","F","1986-04-04","Daughter"),

-> ("333445555","Theodore","M","1983-10-25","Son"),

-> ("333445555","Joy","F","1958-05-03","Spouse"),

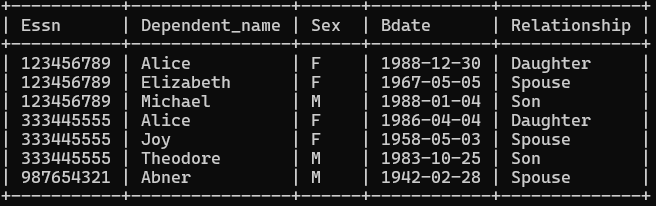
-> ("987654321","Abner","M","1942-02-28","Spouse"),

-> ("123456789","Michael","M","1988-01-04","Son"),

-> ("123456789","Alice","F","1988-12-30","Daughter"),

-> ("123456789","Elizabeth","F","1967-05-05","Spouse");

>>> select \* from DEPENDENT;

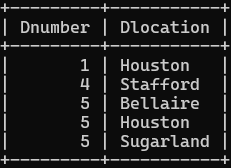


Inserting values into DEPT\_LOCATIONS

>>> insert into dept\_locations values (1,"Houston"),(4,"Stafford"),(5,"Bellaire"),(5,"Houston"),

-> (5,"Sugarland");

>>> select \* from DEPT\_LOCATIONS;



Exp 4:

CREATING TABLES

>>>create table CLIENT\_MASTER ( CLIENTNO varchar(6) primary key check( CLIENTNO LIKE 'C%'),

-> NAME varchar(20) NOT NULL,

-> ADDRESS1 varchar(30),

-> ADDRESS2 varchar(30),

-> CITY varchar(15),

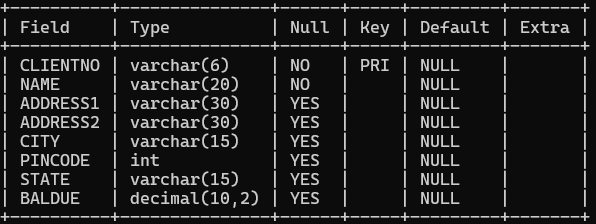
-> PINCODE integer(8),

-> STATE varchar(15),

-> BALDUE decimal(10,2)

-> );

>>> desc CLIENT\_MASTER;



>>>create table PRODUCT\_MASTER ( PRODUCTNO varchar(6) primary key check( PRODUCTNO LIKE 'P%'),

-> DESCRIPTION varchar(15) NOT NULL,

-> PROFITPERCENT decimal(4,2) NOT NULL,

-> UNIT\_MEASURE varchar(10) NOT NULL,

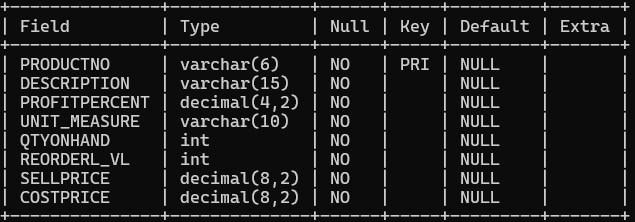
-> QTYONHAND integer(8) NOT NULL,

-> REORDERL\_VL integer(8) NOT NULL,

-> SELLPRICE decimal(8,2) NOT NULL,

-> COSTPRICE decimal(8,2) NOT NULL);

>>> desc PRODUCT\_MASTER;



>>>create table SALESMAN\_MASTER ( SALESMANNO varchar(6) primary key check( SALESMANNO LIKE 'S%'),

-> SALESMANNAME varchar(20) NOT NULL,

-> ADDRESS1 varchar(30) NOT NULL,

-> ADDRESS2 varchar(30) NOT NULL,

-> CITY varchar(20),

-> PINCODE integer(8),

-> STATE varchar(20),

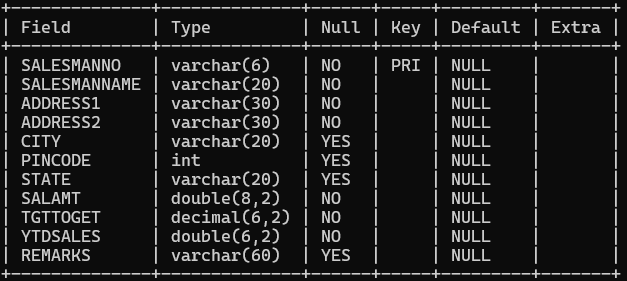
-> SALAMT real(8,2) NOT NULL CHECK( SALAMT !=0),

-> TGTTOGET decimal(6,2) NOT NULL CHECK( TGTTOGET !=0),

-> YTDSALES double(6,2) NOT NULL,

-> REMARKS varchar(60));

>>> desc SALESMAN\_MASTER;



Inserting values into CLIENT\_MASTER

> insert into CLIENT\_MASTER ( CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE)

-> VALUES( 'C00001', 'Ivan bayross', 'Mumbai', 400054, 'Maharashtra', 15000);

> insert into CLIENT\_MASTER ( CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE)

-> VALUES( 'C00002', 'Mamta muzumdar', 'Madras', 780001, 'Tamil nadu', 0);

> insert into CLIENT\_MASTER ( CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE)

-> VALUES( 'C00003', 'Chhaya bankar', 'Mumbai', 400057, 'Maharashtra', 5000);

> insert into CLIENT\_MASTER ( CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE)

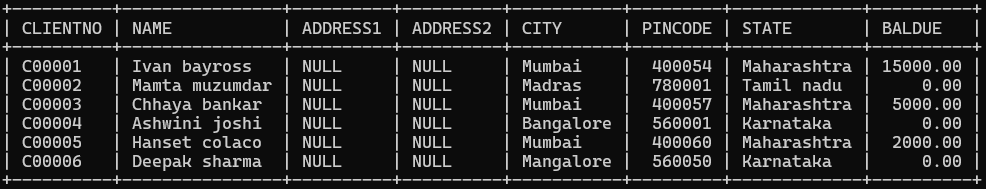
-> VALUES( 'C00004', 'Ashwini joshi', 'Bangalore', 560001, 'Karnataka', 0);

> insert into CLIENT\_MASTER ( CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE)

-> VALUES( 'C00005', 'Hanset colaco', 'Mumbai', 400060, 'Maharashtra', 2000);

> insert into CLIENT\_MASTER ( CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE)

-> VALUES( 'C00006', 'Deepak sharma', 'Mangalore', 560050, 'Karnataka', 0);

>>> select \* from CLIENT\_MASTER;

Inserting values into PRODUCT\_MASTER

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

-> VALUES ( 'P00001', 'T-Shirt', 5, 'Piece', 200, 50, 350, 250);

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

-> VALUES ( 'P0345', 'Shirts', 6, 'Piece', 150, 50, 500, 350);

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

-> VALUES ( 'P06734', 'Cotton jeans', 5, 'Piece', 100, 20, 600, 450);

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

-> VALUES ( 'P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500);

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

-> VALUES ( 'P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550);

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

-> VALUES ( 'P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450);

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

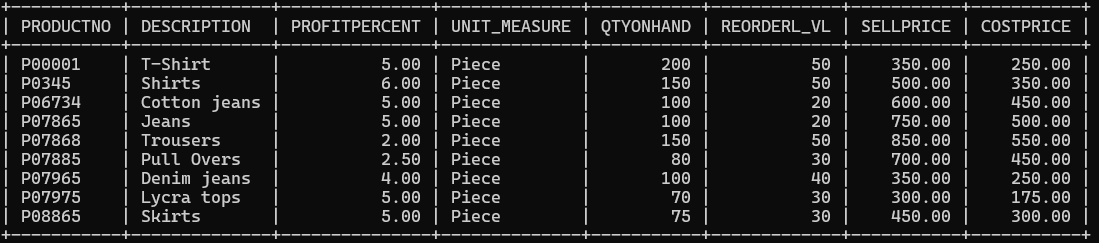
-> VALUES ( 'P07965', 'Denim jeans', 4, 'Piece', 100, 40, 350, 250);

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

-> VALUES ( 'P07975', 'Lycra tops', 5, 'Piece', 70, 30, 300, 175);

> insert into PRODUCT\_MASTER ( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT\_MEASURE, QTYONHAND, REORDERL\_VL, SELLPRICE, COSTPRICE)

-> VALUES ( 'P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);

>>> select \* from PRODUCT\_MASTER;

Inserting values into Salesman\_master

mysql> insert into Salesman\_master (Salesmanno,salesmanname,address1,address2,city,pincode,state,salamt,tgttoget,ytdsales) values

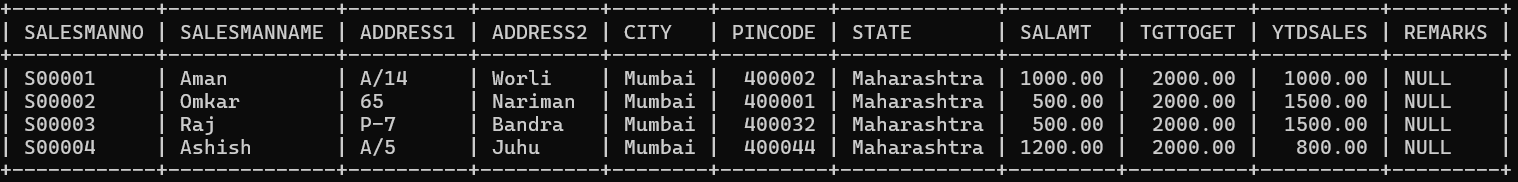
-> ("S00001","Aman","A/14","Worli","Mumbai",400002,"Maharashtra",1000,2000,1000),

-> ("S00002","Omkar","65","Nariman","Mumbai",400001,"Maharashtra",500,2000,1500),

-> ("S00003","Raj","P-7","Bandra","Mumbai",400032,"Maharashtra",500, 2000 ,1500 ),

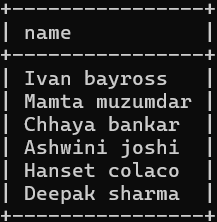
-> ("S00004","Ashish","A/5","Juhu","Mumbai",400044,"Maharashtra",1200,2000,800 ) ;

>>> select \* from salesman\_master;

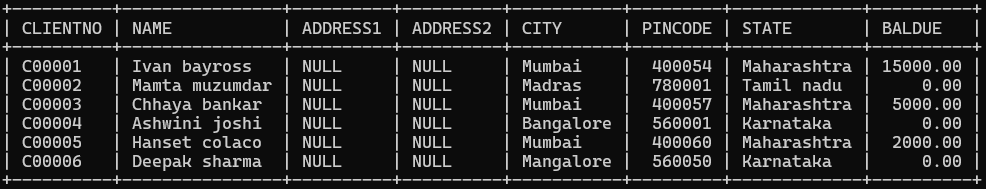


**Exercise on retrieving records from a table.**   
**a.    Find out the names of all the clients.**

Ans: mysql> select name from CLIENT\_MASTER;

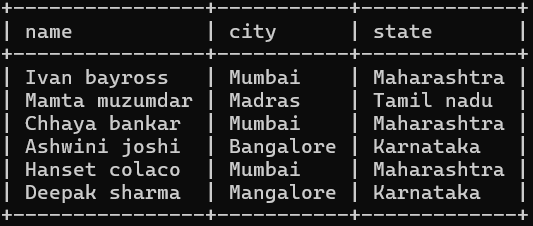


**b.    Retrieve the entire contents of the Client\_Master table.**

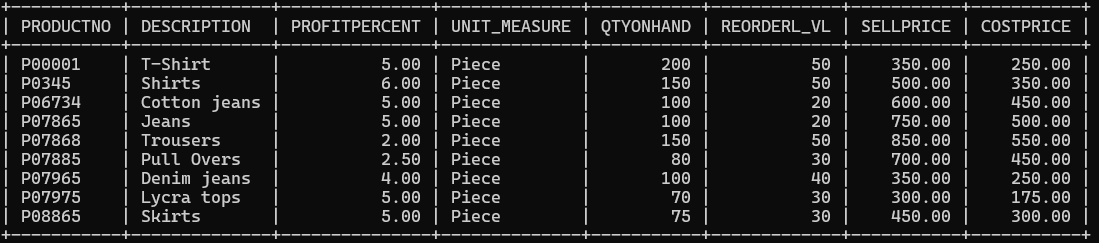
Ans: select \* from CLIENT\_MASTER;

**c.    Retrieve the list of names, city and the state of all the clients.**

Ans: select name,city,state from CLIENT\_MASTER;

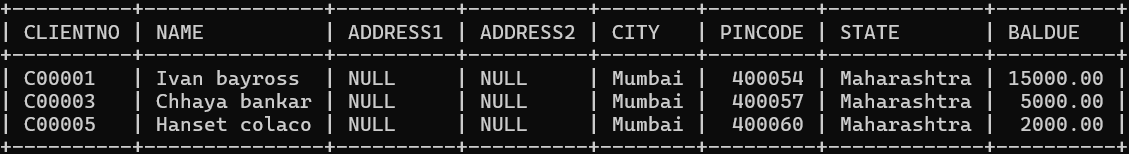


**d.    List the various products available from the Product\_Master table.**

Ans: select \* from PRODUCT\_MASTER;

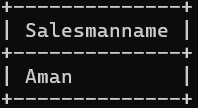
**e.    List all the clients who are located in Mumbai.**

Ans: select \* from CLIENT\_MASTER where city='Mumbai';



**f.     Find the names of salesman who have a salary equal to Rs.3000.**

Ans: mysql> select Salesmanname from SALESMAN\_MASTER where salamt=1000;



**4.Exercise on deleting records in a table**   
**a.     Delete all salesman from the Salesman\_Master whose salaries are equal to Rs.3500.**

Ans: update CLIENT\_MASTER set city='Bangalore'where ClientNo='C00005';

**b.     Delete all products from Product\_Master where the quantity on hand is equal to 100.**

Ans: update CLIENT\_MASTER set BalDue='1000'where ClientNo='C00001';

**c.     Delete from Client\_Master where the column state holds the value ‘Tamil Nadu’.**

Ans: update PRODUCT\_MASTER set CostPrice='950'where Description='Trousers';

**5.Exercise on altering the table structure   
a.     Add a column called ‘Telephone’ of data type integer to the Client\_Master table.**

Ans:  **b.    Change the size off SellPrice column in Product \_Master to 10, 2.**

1. **Exercise on deleting the table structure along with the data   
   a.    Destroy the table Client\_Master along with its data.**

Experiment 5

Creating Table

mysql> CREATE TABLE Supplier (

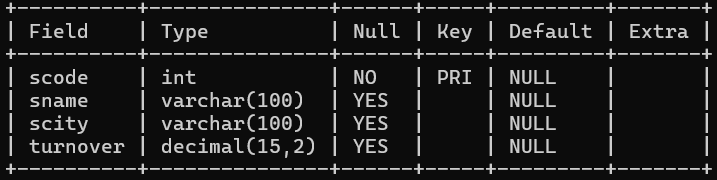
-> scode INT PRIMARY KEY,

-> sname VARCHAR(100),

-> scity VARCHAR(100),

-> turnover DECIMAL(15, 2)

-> );



mysql> CREATE TABLE Part (

-> pcode INT PRIMARY KEY,

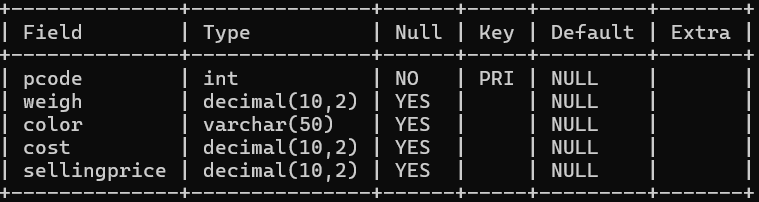
-> weigh DECIMAL(10, 2),

-> color VARCHAR(50),

-> cost DECIMAL(10, 2),

-> sellingprice DECIMAL(10, 2)

-> );



mysql> CREATE TABLE Supplier\_Part (

-> scode INT,

-> pcode INT,

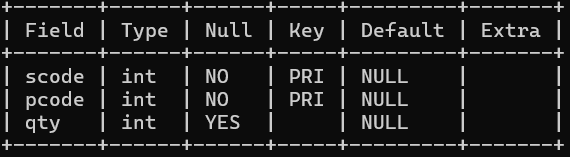
-> qty INT,

-> PRIMARY KEY (scode, pcode), -- Composite primary key combining scode and pcode

-> FOREIGN KEY (scode) REFERENCES Supplier(scode), -- Foreign key constraint on scode

-> FOREIGN KEY (pcode) REFERENCES Part(pcode) -- Foreign key constraint on pcode

-> );



Inserting Values into the table

mysql> INSERT INTO Supplier (scode, sname, scity, turnover) VALUES

-> (101, 'Reliance Industries', 'Mumbai', 150000.00),

-> (102, 'Tata Steel', 'Jamshedpur', 75000.00),

-> (103, 'Infosys', 'Bengaluru', 90000.00),

-> (104, 'Mahindra & Mahindra', 'Pune', 65000.00),

-> (105, 'Bharat Petroleum', 'New Delhi', 110000.00);



mysql> INSERT INTO Part (pcode, weigh, color, cost, sellingprice) VALUES

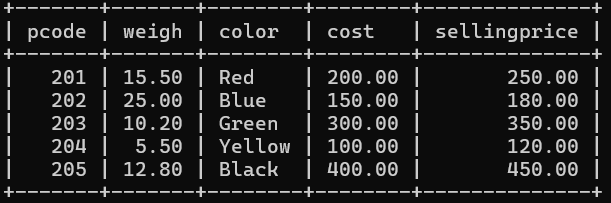
-> (201, 15.5, 'Red', 200.00, 250.00),

-> (202, 25.0, 'Blue', 150.00, 180.00),

-> (203, 10.2, 'Green', 300.00, 350.00),

-> (204, 5.5, 'Yellow', 100.00, 120.00),

-> (205, 12.8, 'Black', 400.00, 450.00);



mysql> INSERT INTO Supplier\_Part (scode, pcode, qty) VALUES

-> (101, 201, 500),

-> (102, 202, 300),

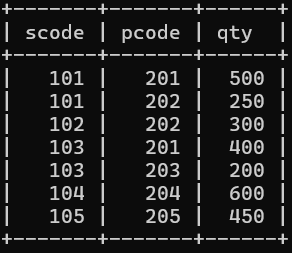
-> (103, 203, 200),

-> (104, 204, 600),

-> (105, 205, 450),

-> (101, 202, 250),

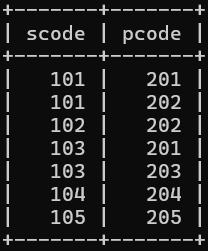
-> (103, 201, 400);



Questions:

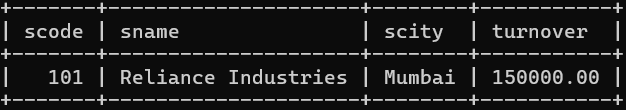
**1. Get the supplier number and part number in ascending order of supplier number.**

Ans: mysql> select scode, pcode from supplier\_part order by scode asc;



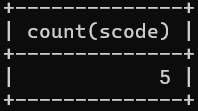
**2. Get the details of supplier who operate from Mumbai with turnover 150000.**

Ans: mysql> select \* from supplier where scity = 'Mumbai' and turnover = 150000;



**3. Get the total number of supplier.**

Ans: mysql> select count(scode) from supplier;



**4. Get the part number weighing between 25 and 35.**

Ans: mysql> select pcode from part where weigh between 25 and 35;



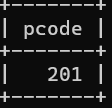
**5. Get the supplier number whose turnover is null.**

Ans: mysql> select scode from supplier where turnover is NULL;



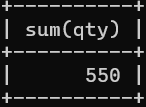
**6. Get the part number that cost 20, 200 or 40 rupees.**

Ans: mysql> select pcode from part where cost in(20, 200, 40);



**7. Get the total quantity of Part 202 that is supplied.**

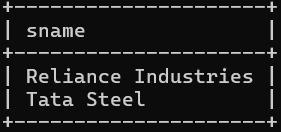
Ans: mysql> select sum(qty) from supplier\_part where pcode=202;



**8. Get the name of supplier who supply part 2.**

Ans: mysql> select sname from supplier where scode in

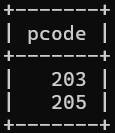
-> (select scode from supplier\_part where pcode=202);



**9. Get the part number whose cost is greater than the average cost.**

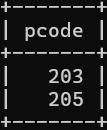
Ans: mysql> select pcode from part where cost>

-> (select avg(cost) from part);



**10. Get the supplier number and turnover in descending order of turnover.**

Ans: mysql> select scode, turnover from supplier order by turnover desc;



**EXPERIMENT-6**

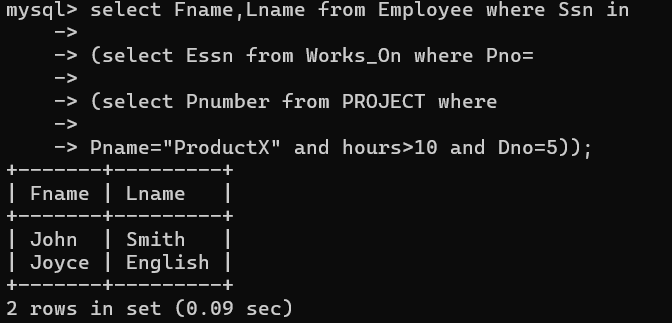
**Title:  Use of Inbuilt functions and relational algebra operation**

**Objective:** To understand the use of inbuilt function and relational algebra with sql query.

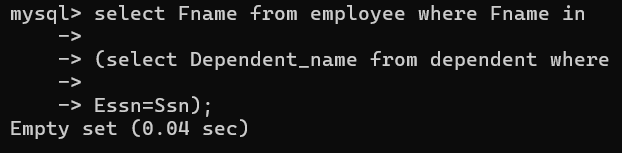
     Write and execute the following queries using the Relational Algebra on the COMPANY

          database schema.

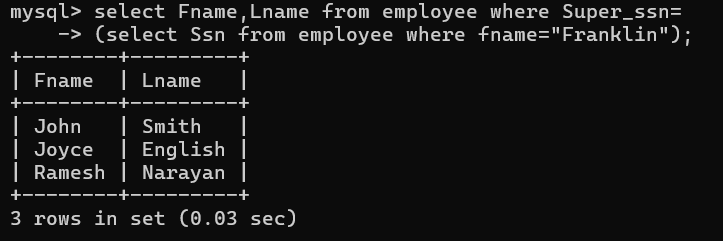
1. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ‘ProductX’ project.



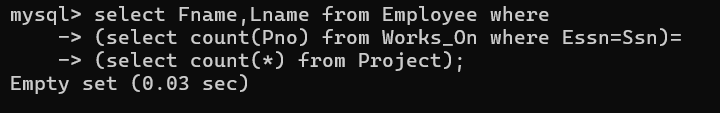
1. List the names of all employees who have a dependent with the same first name as themselves.



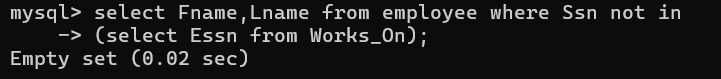
1. Find the names of employees who are directly supervised by ‘Franklin Wong’.



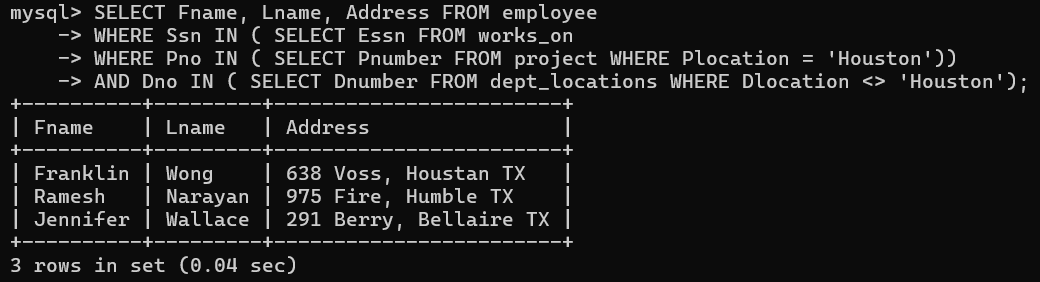
1. Retrieve the names of employees who work on every project.



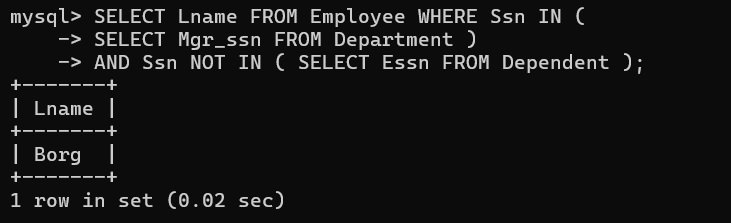
1. Retrieve the names of employees who do not work on any project.



1. Retrieve the names and addresses of all employees who work on at least one project located in Houston but whose department has no location in Houston.



1. Retrieve the last names of all department managers who have no dependents.



**Experiment 7:**

**Question 1 :-**

**Create the following two tables (EMP and DEPT)**

**Program:-**

**CREATE DATABASE LAB7;**

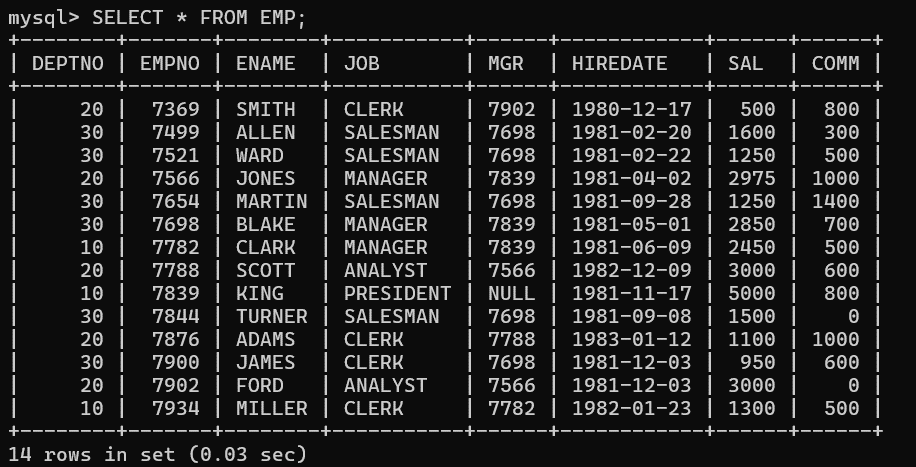
**USE LAB7;**

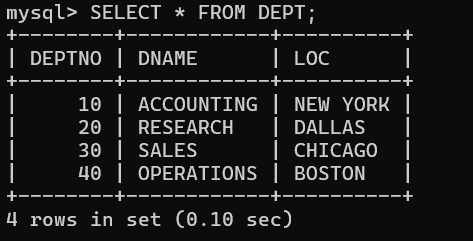
**CREATE TABLE EMP (DEPTNO INT, EMPNO INT PRIMARY KEY, ENAME VARCHAR(20), JOB VARCHAR(20), MGR INT, HIREDATE DATE, SAL INT, COMM INT);**

**INSERT INTO EMP (DEPTNO, EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM) VALUES (20, 7369, 'SMITH', 'CLERK', 7902, '1980-12-17', 500, 800), (30, 7499, 'ALLEN', 'SALESMAN', 7698, '1981-02-20', 1600, 300), (30, 7521, 'WARD', 'SALESMAN', 7698, '1981-02-22', 1250, 500), (20, 7566, 'JONES', 'MANAGER', 7839, '1981-04-02', 2975, 1000), (30, 7654, 'MARTIN', 'SALESMAN', 7698, '1981-09-28', 1250, 1400), (30, 7698, 'BLAKE', 'MANAGER', 7839, '1981-05-01', 2850, 700), (10, 7782, 'CLARK', 'MANAGER', 7839, '1981-06-09', 2450, 500), (20, 7788, 'SCOTT', 'ANALYST', 7566, '1982-12-09', 3000, 600), (10, 7839, 'KING', 'PRESIDENT', NULL, '1981-11-17', 5000, 800), (30, 7844, 'TURNER', 'SALESMAN', 7698, '1981-09-08', 1500, 0), (20, 7876, 'ADAMS', 'CLERK', 7788, '1983-01-12', 1100, 1000), (30, 7900, 'JAMES', 'CLERK', 7698, '1981-12-03', 950, 600), (20, 7902, 'FORD', 'ANALYST', 7566, '1981-12-03', 3000, 0), (10, 7934,'MILLER','CLERK',7782 ,'1982-01-23' ,1300 ,500);**

**CREATE TABLE DEPT(DEPTNO INT PRIMARY KEY, DNAME VARCHAR(20), LOC VARCHAR(20));**

**INSERT INTO DEPT VALUES(10 ,'ACCOUNTING' ,'NEW YORK'), (20 ,'RESEARCH' ,'DALLAS') , (30,'SALES','CHICAGO'),(40,'OPERATIONS','BOSTON');**

**OUTPUT:-**

****

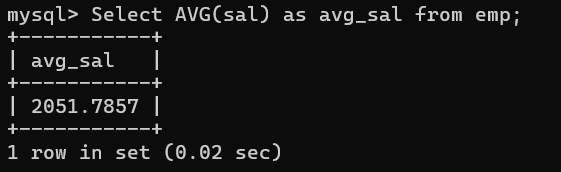
**Question 2 :-**

**Write the Queries for the following using In-built functions.**

1. **Retrieve average salary of all employees.**

**Program:-**

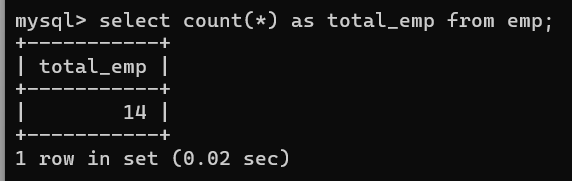
**Select AVG(sal) as avg\_sal from emp;**

**OUTPUT:-**

1. **Retrieve the number of employees.**

**Program:-**

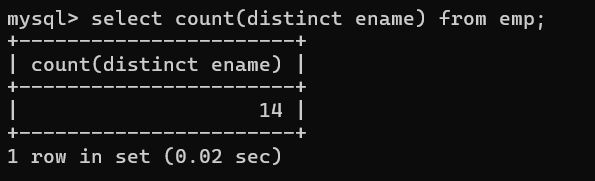
**select count(\*) as total\_emp from emp;**

**OUTPUT:-**

1. **Retrieve distinct number of employee.**

**Program:-**

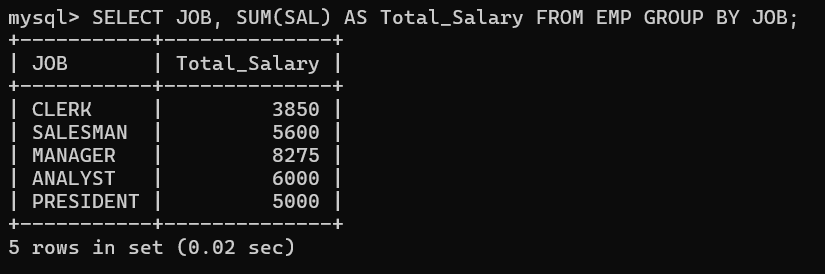
**select count(distinct ename) from emp;**

**OUTPUT:-**

1. **Retrieve total salary of employee group by job.**

**Program:-**

**SELECT JOB, SUM(SAL) AS Total\_Salary FROM EMP GROUP BY JOB;**

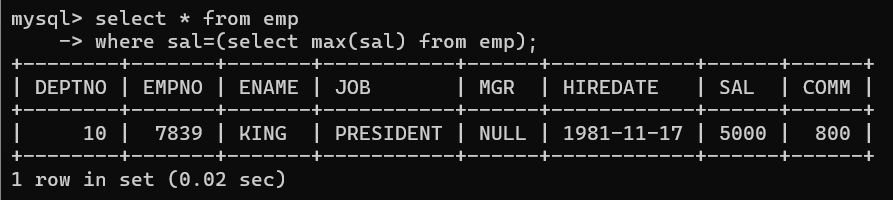
**OUTPUT:-**

1. **Display the employee information with maximum salary.**

**Program:-**

**select \* from emp**

**where sal=(select max(sal) from emp);**

**OUTPUT:-**

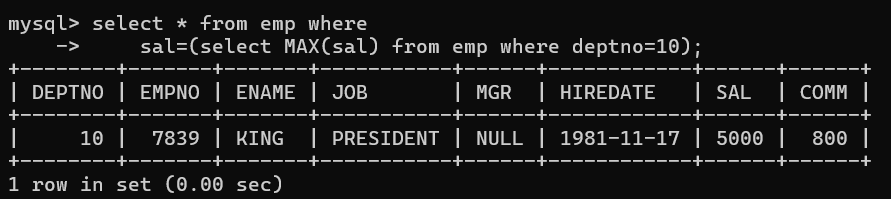
1. **Find the highest paid employee in department 10.**

**Program:-**

**select \* from emp where**

**sal=(select MAX(sal) from emp where deptno=10);**

**OUTPUT:-**

****

1. **List the emps whose sal is equal to the average of max and minimum.**

**Program:-**

**select \* from emp where sal = (select avg(sal) from (select min(sal) from emp UNION select max(sal) from emp) as AVGSAL);**

**OUTPUT:-**

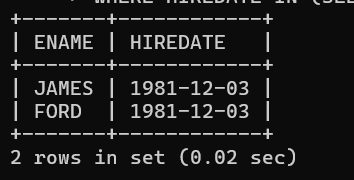
1. **List the emps who joined in the company on the same date.**

**Program:-**

**ENAME, HIREDATE**

**FROM EMP**

**WHERE HIREDATE IN (SELECT HIREDATE FROM EMP GROUP BY HIREDATE HAVING COUNT(\*) > 1);**

**OUTPUT:-**

1. **Display the employee names in upper and lower case.**

**Program:-**

**select UPPER(ename),**

**LOWER(ename)**

**from emp;**

**OUTPUT:-**

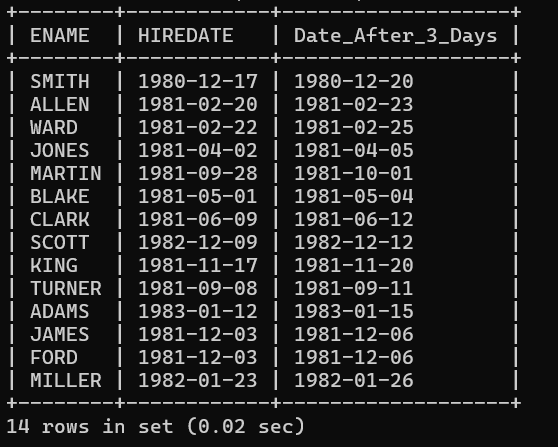
****

1. **find the date of 3 days later from hiredate.**

**Program:-**

**SELECT ENAME, HIREDATE, HIREDATE + INTERVAL '3' DAY AS Date\_After\_3\_Days FROM EMP;**

**OUTPUT:-**

****

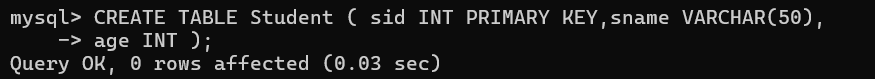
**EXPERIMENT-8**

**Title: Use of different SQL clauses and join**

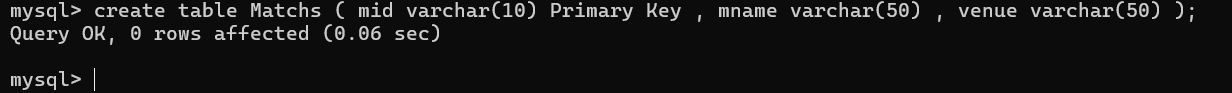
**Objective:** To understand the use of group by and having clause and execute the SQL commands using JOIN

1.Consider the following schema:

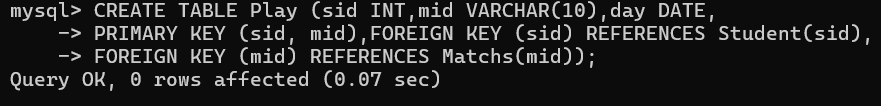
**Student (sid, sname, age)**



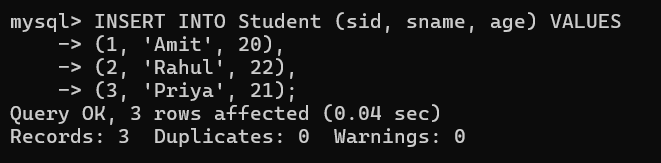
**Match (mid, mname, venue)**

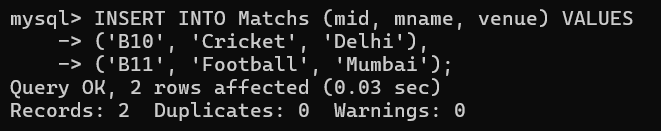


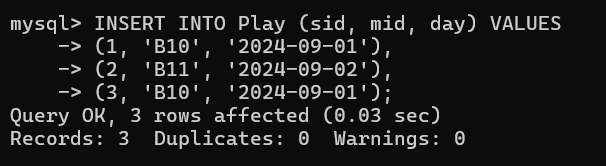
**Play (sid, mid, day(date))**



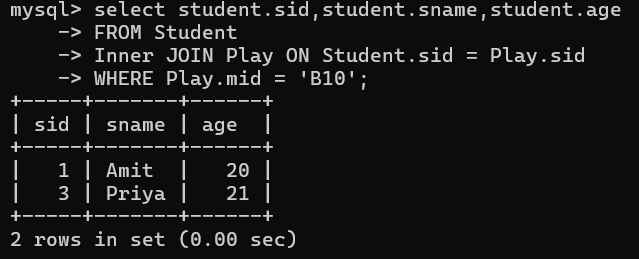
1. Populate all the tables.



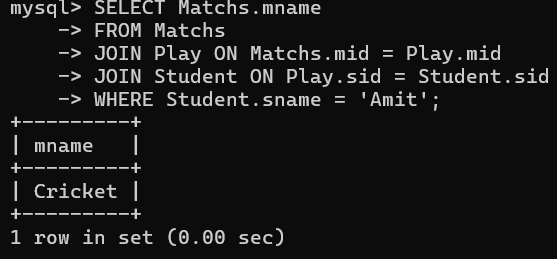




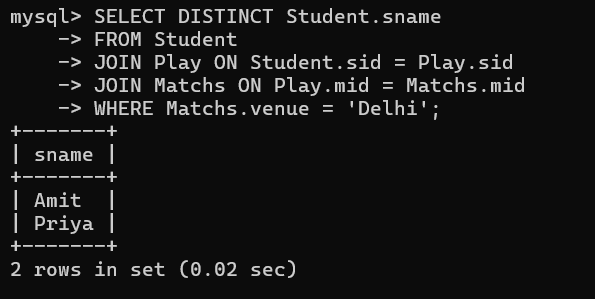
1. Find all information of students who have played match number B10.



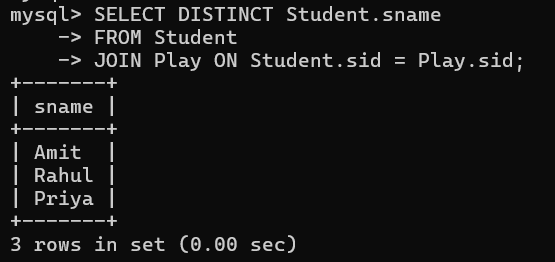
1. Find the name of matches played by Amit.



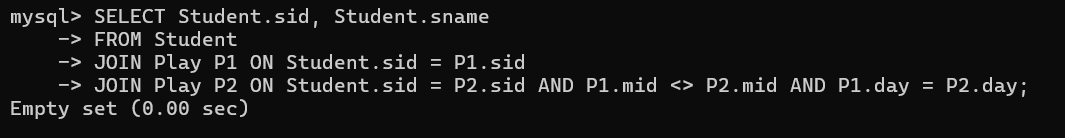
1. Find the names of students who have played a match in Delhi.



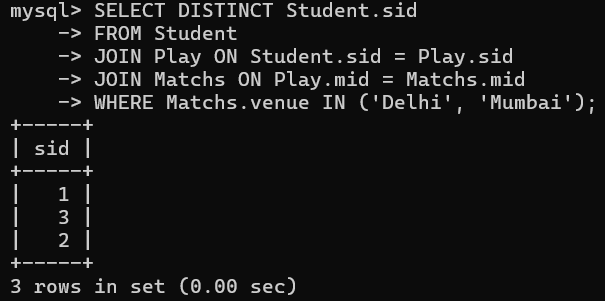
1. Find the names of students who have played at least one match.



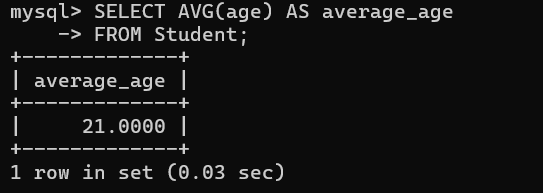
1. Find the ids and names of students who have played two different matches on the same day.



1. Find the ids of students who have played a match in Delhi or Mumbai.



1. Find the average age of students.



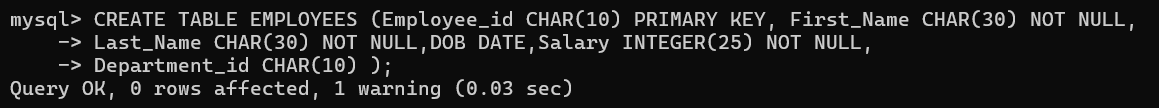
**EXPERIMENT-9**

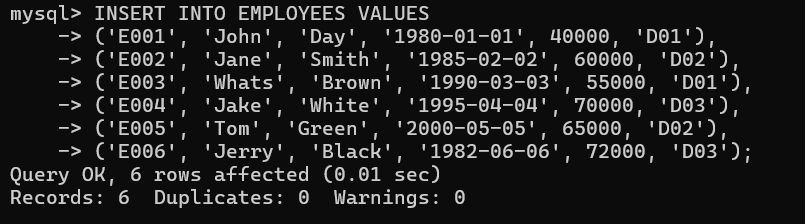
**Title:  To understand the concepts of Views.**

**Objective:** Students will be able to implement the concept of views.

**1. Create table of table name: EMPLOYEES and add 6 rows**

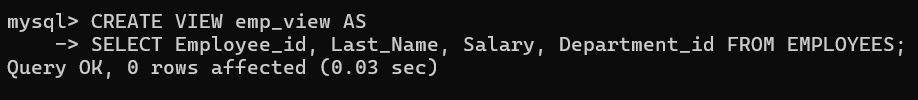
|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Width | Attributes |
| Employee\_id | Character | 10 | PK |
| First\_Name | Character | 30 | NN |
| Last\_Name | Character | 30 | NN |
| DOB | Date |  |  |
| Salary | Number | 25 | NN |
| Department\_id | Character | 10 |  |



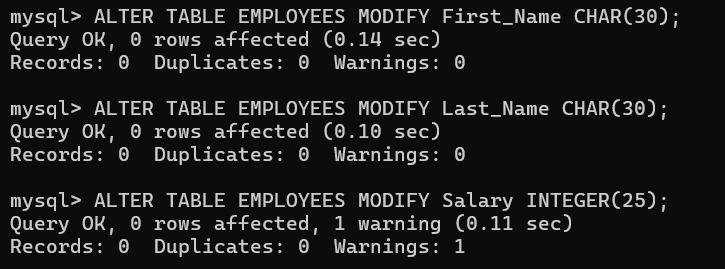


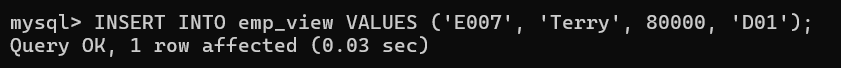
**2. Execute the following view related queries:**

1. Create View of name emp\_view and the column would be Employee\_id, Last\_Name, salary and department\_id only.

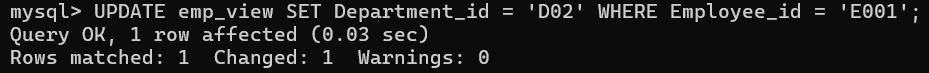


1. Insert values into view(remove the NOT NULL constraint and then insert values):





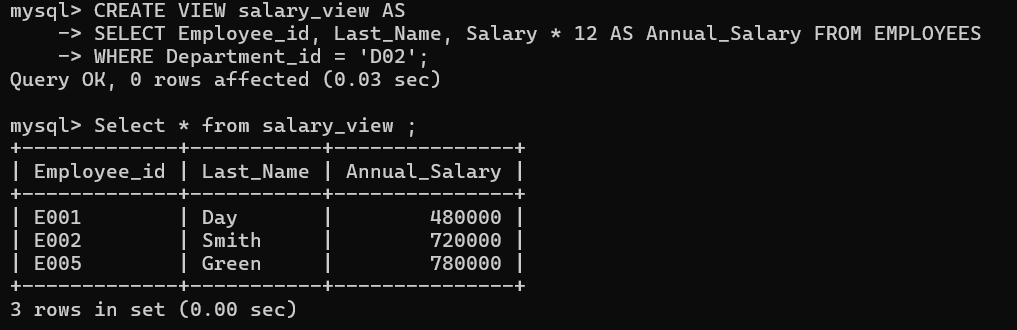
1. Modify, delete and drop operations are performed on view.







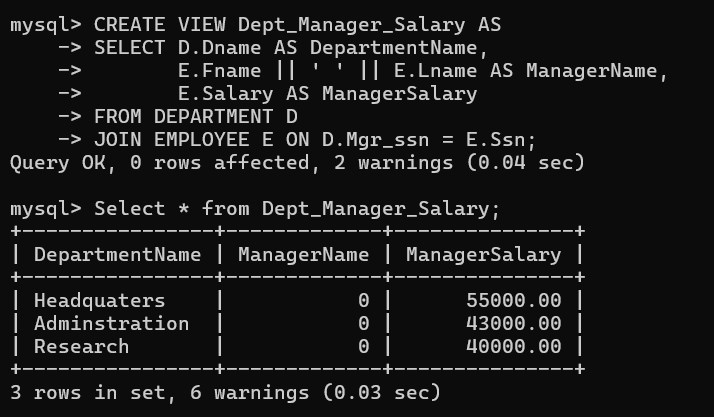
1. Creates a view  named salary\_view. The view shows the employees in department 20 and their annual salary.



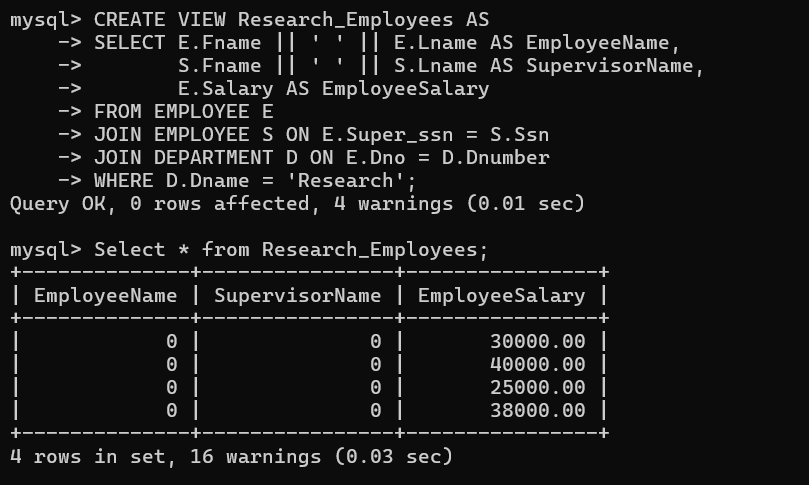
**EXPERIMENT-10**

**Title: Create the following views in SQL on the COMPANY database schema presented in Experiment 2.**

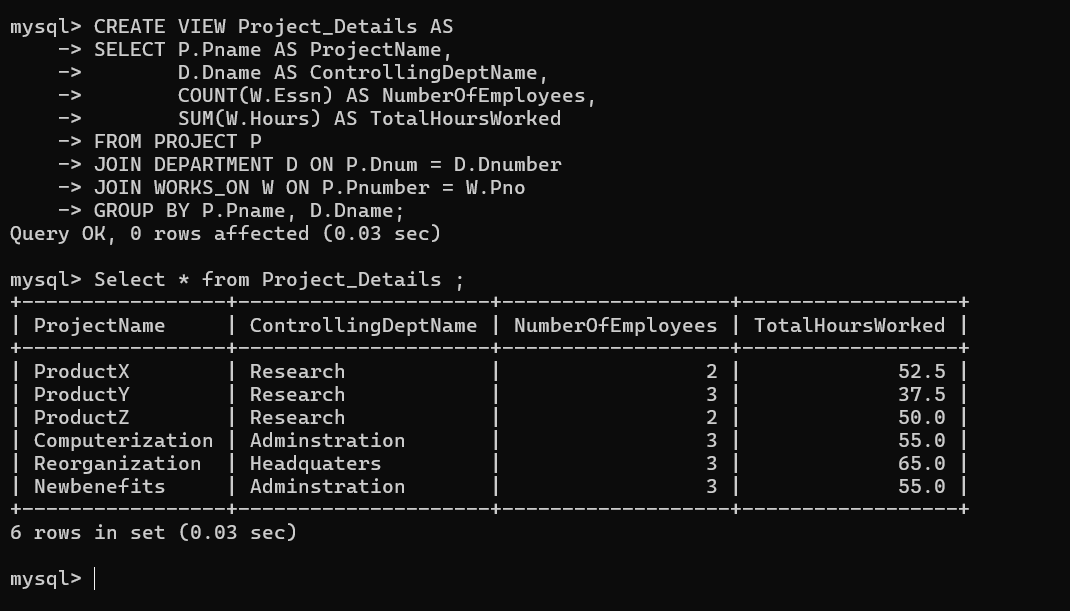
1. A view that has the department name, manager name, and manager salary for every department.



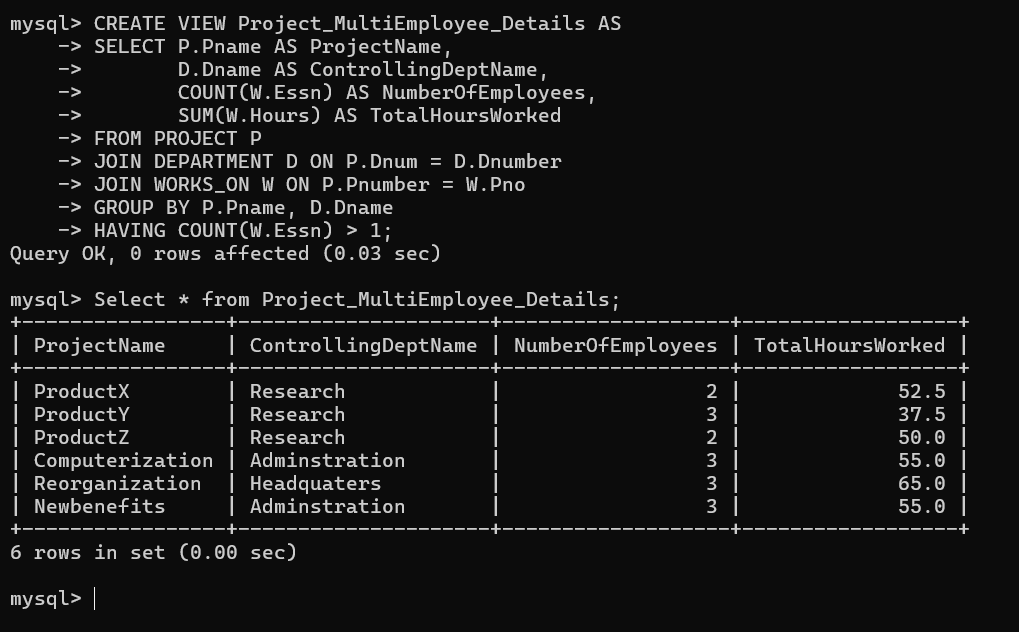
1. A view that has the employee name, supervisor name, and employee salary for each employee who works in the ‘Research’ department.



1. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.



1. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.



**EXPERIMENT-18**

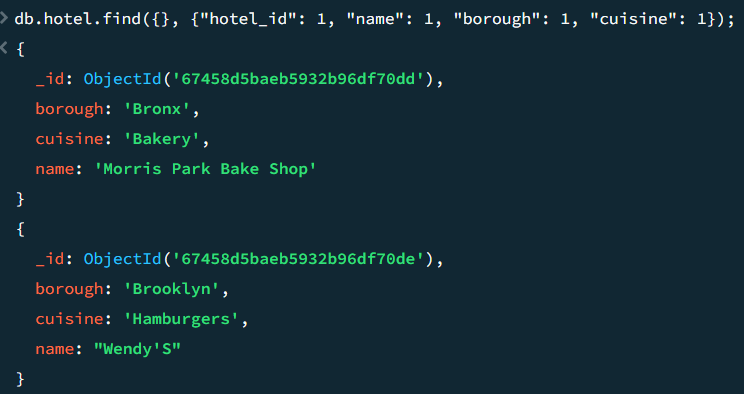
**Title:  To understand the concepts of NoSQL Database**

**Objective:** Students will be able to implement the concept of NoSQL Database MongoDB.

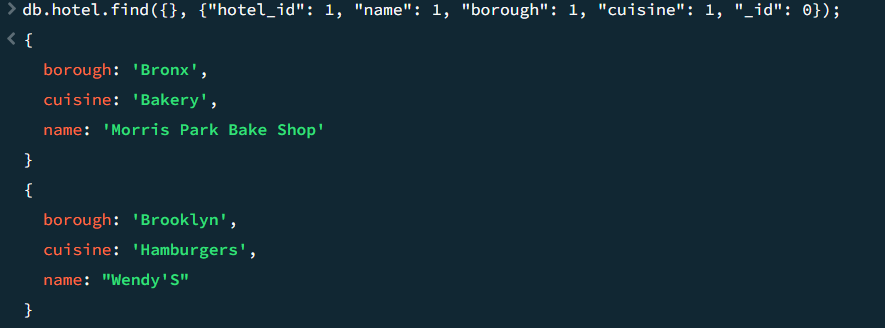
1. Write a MongoDB query to display all the documents in the collection hotel.



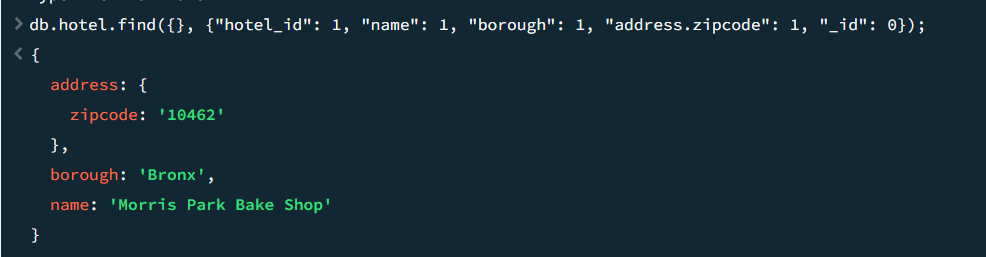
1. Write a MongoDB query to display the fields hotel\_id, name, Borough and cuisine for all the documents in the collection hotel.



1. Write a MongoDB query to display the fields hotel\_id, name, Borough and cuisine, but exclude the field \_id for all the documents in the collection hotel.



1. Write a MongoDB query to display the fields hotel\_id, name, Borough and zip code, but exclude the field \_id for all the documents in the collection hotel. hotel



1. Write a MongoDB query to display all the hotel which is in the Borough Bronx.

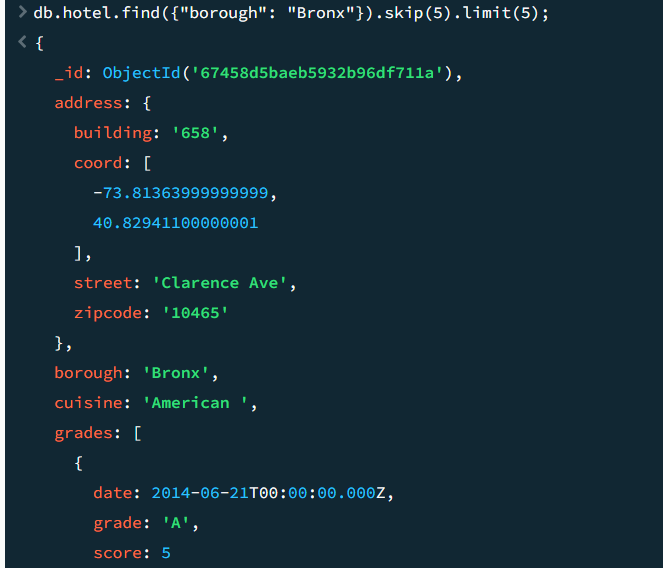


**EXPERIMENT-19**

**Title:  To understand the concepts of NoSQL Database**

**Objective:** Students will be able to implement the concept of NoSQL Database MongoDB.

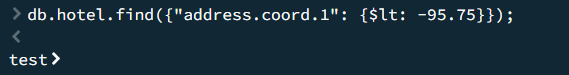
1. Write a MongoDB query to display the next 5 hotels after skipping first 5 which are in the Borough Bronx.



1. Write a MongoDB query to find the hotels that achieved a score, more than 80 but less than 100.



1. Write a MongoDB query to find the hotels which locate in latitude value less than -95.75



1. Write a MongoDB query to find the hotels that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

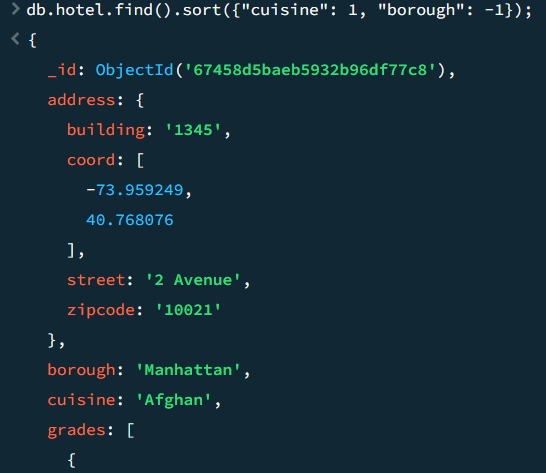


**EXPERIMENT-20**

**Title:  To understand the concepts of NoSQL Database**

**Objective:** Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to arrange the name of the cuisine in ascending order and for that same cuisine Borough should be in descending order.



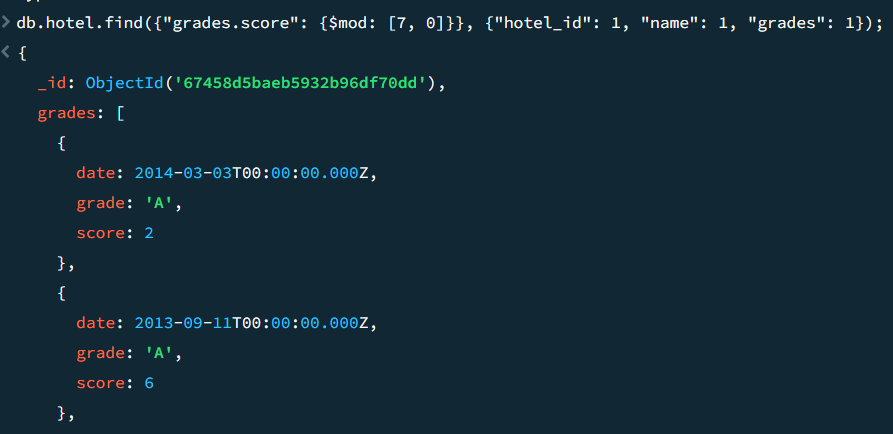
1. Write a MongoDB query to know whether all the addresses contains the street or not.



1. Write a MongoDB query which will select all documents in the hotels collection where the coord field value is Double.



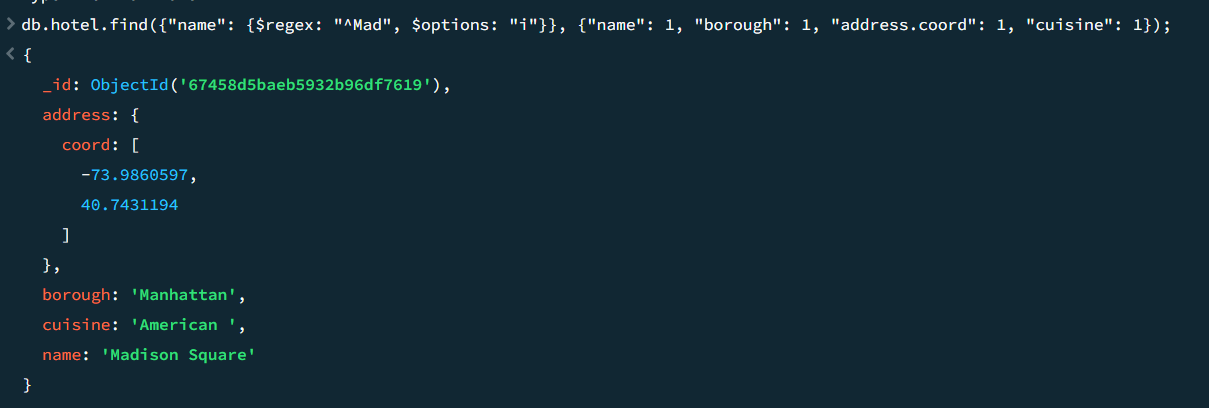
1. Write a MongoDB query which will select the hotel Id, name and grades for those hotels which returns 0 as a remainder after dividing the score by 7.



1. Write a MongoDB query to find the hotel name, Borough, longitude and attitude and cuisine for those hotels which contains 'mon' as three letters somewhere in its name.



1. Write a MongoDB query to find the hotel name, Borough, longitude and latitude and cuisine for those hotels which contain 'Mad' as first three letters of its name.

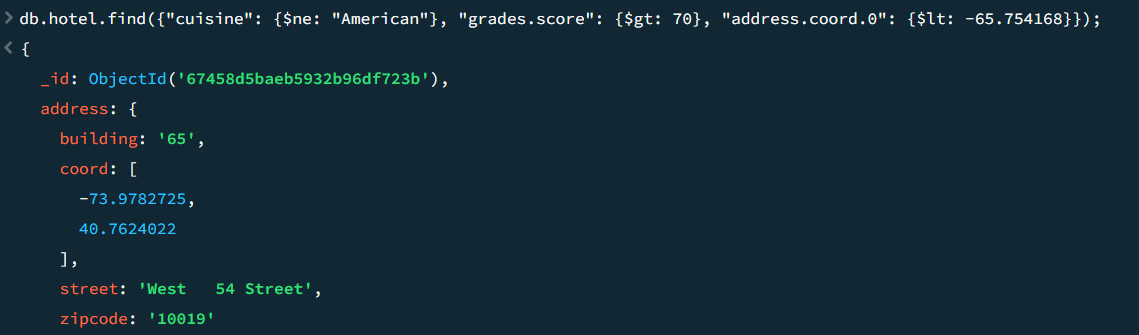


**EXPERIMENT-21**

**Title:  To understand the concepts of NoSQL Database**

**Objective:** Students will be able to implement the concept of NoSQL Database MongoDB.

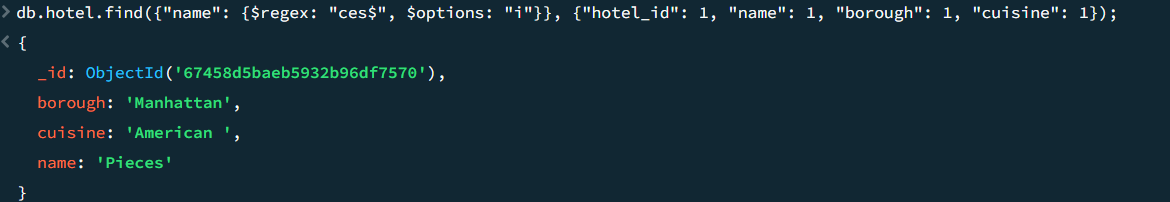
1. Write a MongoDB query to find the hotels which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.



2. Write a MongoDB query to find the hotels which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the Borough Brooklyn. The document must be displayed according to the cuisine in descending order.



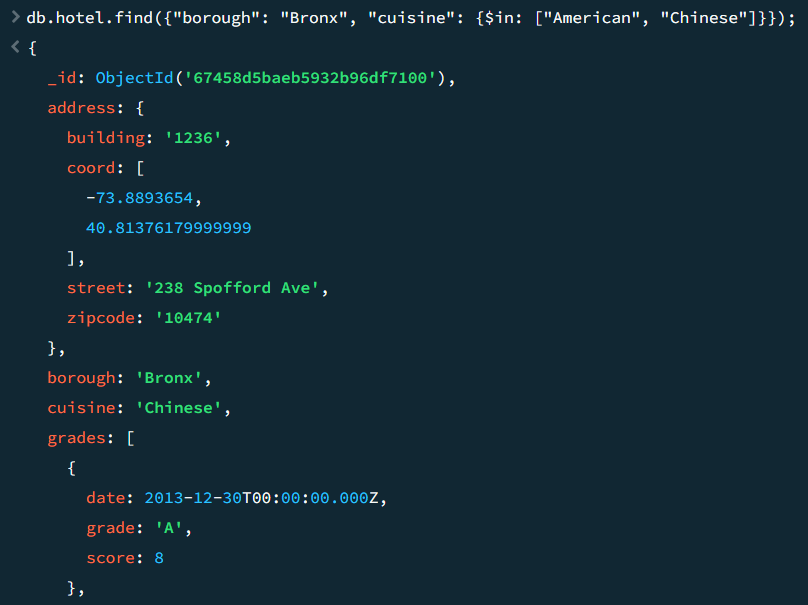
3. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which contain 'ces' as last three letters for its name.



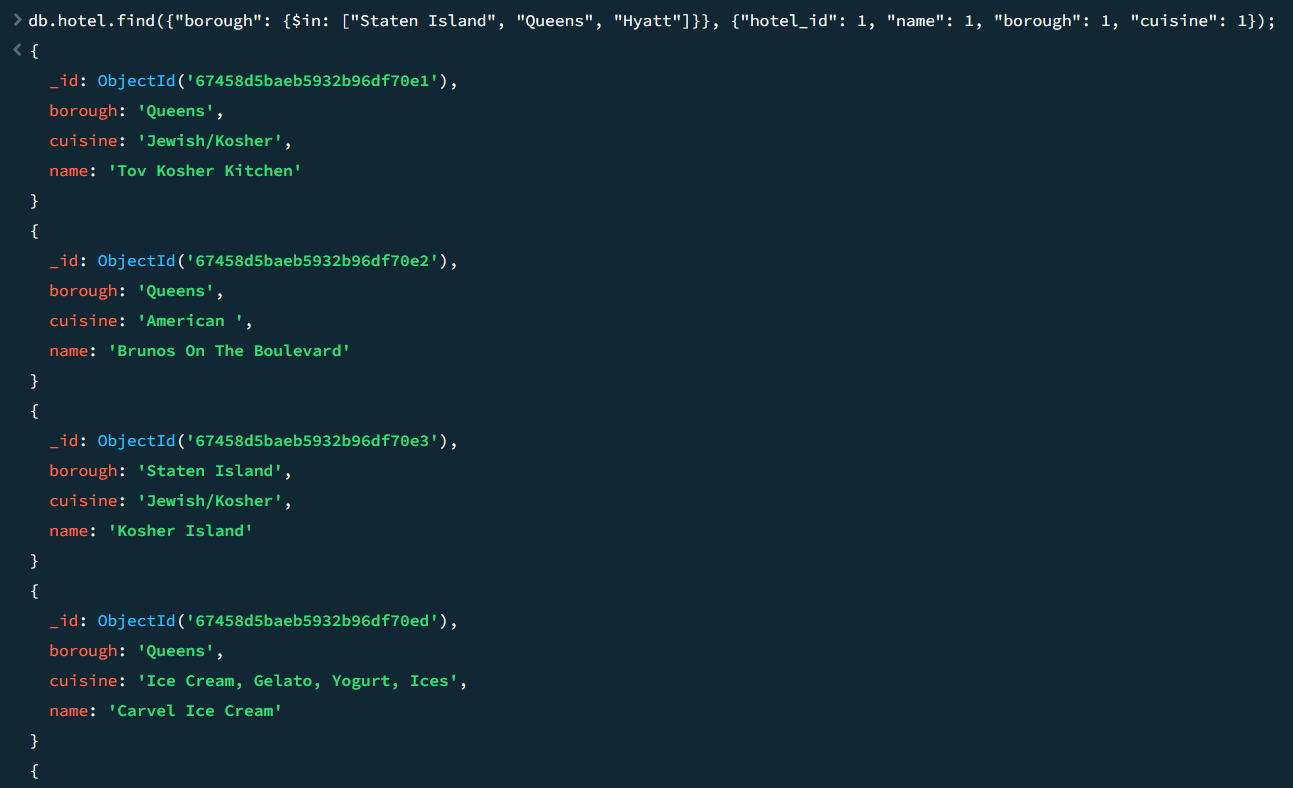
 4. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which contain 'Reg' as three letters somewhere in its name.



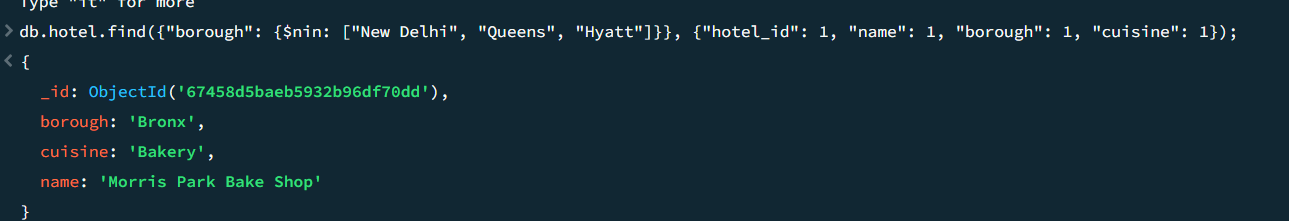
5. Write a MongoDB query to find the hotels which belong to the Borough Bronx and prepared either American or Chinese dish.



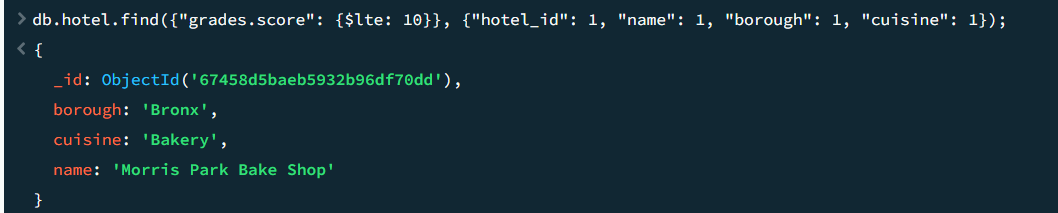
6. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which belong to the Borough Staten Island or Queens or Hyatt.



7. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which are not belonging to the Borough New Delhi or Queens or Hyatt.

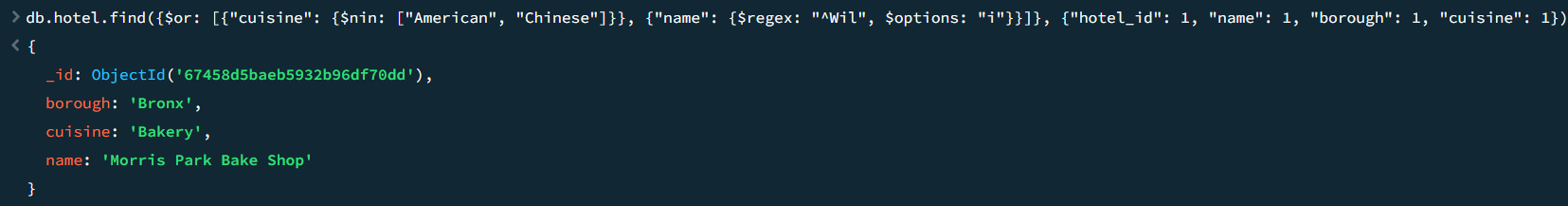


8. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which achieved a score which is not more than 10.



 9. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which prepared dish except 'American' and 'Chinees' or hotel's name begins with letter 'Wil'.

db.hotel.find({$or: [{"cuisine": {$nin: ["American", "Chinese"]}}, {"name": {$regex: "^Wil", $options: "i"}}]}, {"hotel\_id": 1, "name": 1, "borough": 1, "cuisine": 1});



 10. Write a MongoDB query to find the hotel Id, name, and grades for those hotels which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates.

