*Manav Rachna University*

*3rd Semester*

Database management System

**Submitted to: Submitted by:**

**Gaganjot Kaur Lakshaya bhardwaj**

**2K15CSUN01015**

**CSE – 3A**

SQL

The full form of SQL is Structured Query Language.

* ***SQL is divided into three categories :-***

1. DDL - Data Definition Languages
2. DML - Data Manipulation Languages
3. DCL/TCL - Data/Transaction Control Languages

***DDL***

This language is used to create, modify or delete structure of the table. The commands that comes under this languages are -> Create, Alter, Delete

***DML***

This language is used to enter/insert, modify or delete the values row of the table. The commands which comes under this languages are -> Insert, Select, Update and Delete

***TCL***

This language is used to manage the contents of the table.The commands that comes under this language are -> Rollback, Savepoint, Commit

***Data Types***

* Integer/Number
* Char/Varchar/Varchar2
* Float/Decimal
* Date

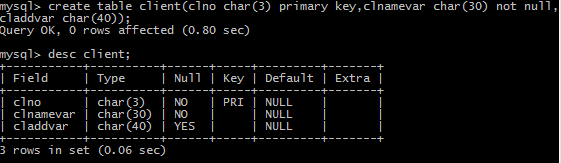
**Lab 0 - Creation of Tables**

**Lab Outcome** : *Students will be able to use DDL – create/alter table statement – to create relations along with the constraints specified.*

**Create the following relations along with the constraints specified:**

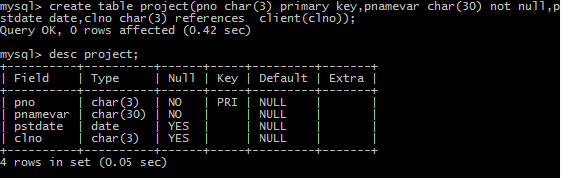
1. Client (clno char(3), clnamevarchar(30), claddvarchar(40))

* PRIMARY KEY- clno
* clnamecan not be left blank

**OUTPUT**:

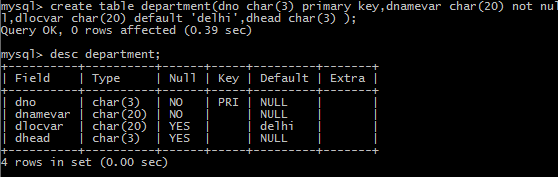
1. Project (pno char(3), pnamevarchar(30),pstdate date, clno char(3))

* PRIMARY KEY- pno
* FOREIGN KEY – clno refers to Client relation
* pnamecan not be left blank

**OUTPUT**:

1. Department (dno char(3), dnamevarchar(20), dlocvarchar(20), dhead char(3))

* PRIMARY KEY – dno
* FOREIGN KEY – dhead refers to eno of Employee relation
* dnamecan not be left blank
* default dloc is delhi

**OUTPUT**:

.

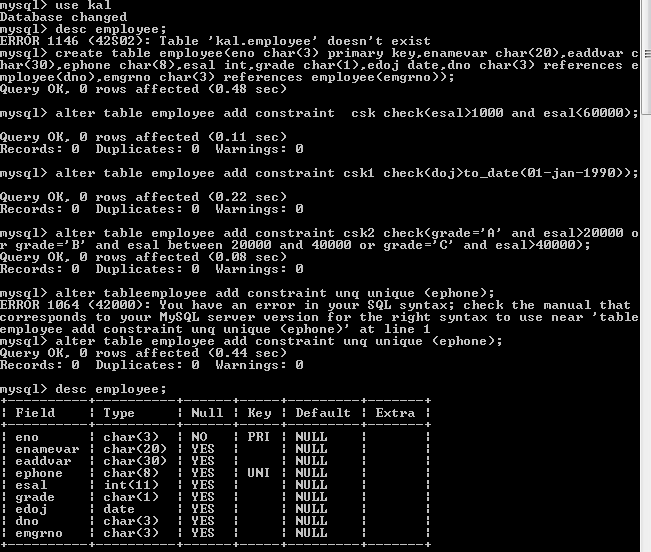
**Lab 1 - Creation of Tables**

**Lab Outcome**: Students will be able to use DDL – create/alter table statement – to create relations along with the constraints specified.

1. Employee (eno char(3), enamevarchar(20), eaddvarchar(30), ephone char(8), esalint, grade char(1), edoj date, dno char(3), emgrno char(3))

* PRIMARY KEY – eno
* FOREIGN KEY – dno refers to Department relation
* FOREIGN KEY – emgrno refers to eno of Employee relation
* salary can hold values between 1000 and 60000
* grade can be A – if salary less than 20000, B – if salary is between 20000 and 40000, C – if salary is greater than 40000
* doj should be more than 01-jan-1990
* ephone can have unique values only

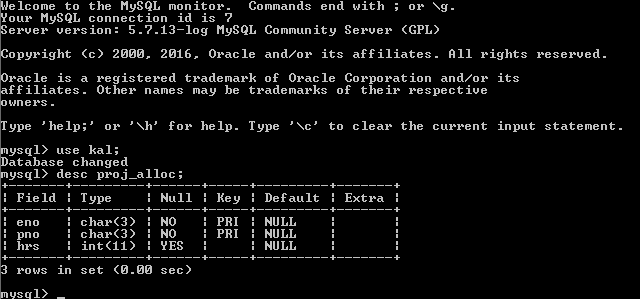
**OUTPUT**:



1. Proj\_alloc (eno char(3), pno char(3), hrsint)

* PRIMARY KEY – eno, pno
* FOREIGN KEY – eno refers to Employee relation
* FOREIGN KEY – pno refers to Project relation
* hrs can have values more than 2

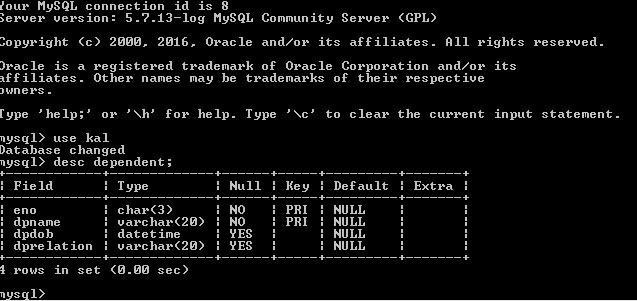
**OUTPUT**:



1. Dependent (eno char(3), dpnamevarchar(20), dpdobdatetime, dprelationvarchar(20))

* PRIMARY KEY – eno, dpname
* FOREIGN KEY – eno refers to Employee relation
* Dprelation can have only one of the values {son,daughter,father,mother}

**OUTPUT**:



**Lab 3 – Insertion & Updation of records in Tables**

**Lab Outcome:** Students will be able to use DML – insert/update table statement – to insert and update records in existing relations.

1. **Insert the records in the following relations:**
2. Client (clno char(3), clname varchar(30), cladd varchar(40))

CLNO CLNAME CLADD

-------- ------------- -----------

c01 xyz Delhi

c02 abc Fbd

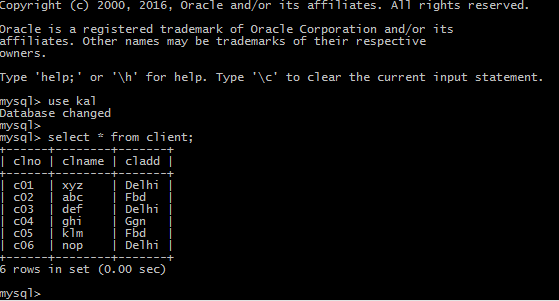
c03 def Delhi

c04 ghi Ggn

c05 klm Fbd

c06 nop Delhi

OUTPUT:



1. Project (pno char(3), pname varchar(30), pstdate date, clno char(3))

PNO PNAME PSTDATE CLNO

--- ------------ -------------- -------

p01 ab 01-JAN-06 c01

p02 bc 01-FEB-06 c02

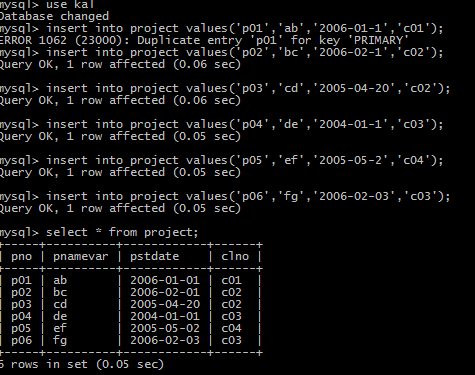
p03 cd 20-APR-05 c02

p04 de 01-JAN-04 c03

p05 ef 02-MAY-05 c04

p06 fg 03-FEB-06 c03

OUTPUT:



1. Department (dno char(3), dname varchar(20), dloc varchar(20), dhead char(3))

DNO DNAME DLOC DHEAD

--- --------------- -------------------- --------

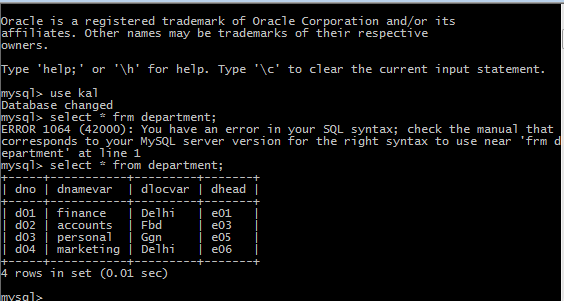
d01 finance Delhi e01

d02 accounts Fbd e03

d03 personal Ggn e05

d04 marketing Delhi e06

OUTPUT:



1. Employee (eno char(3), ename varchar(20), eadd varchar(30), ephone char(8), esal int, grade char(1), edoj date, dno char(3), emgrno char(3))

ENO ENAME EADD EPHONE ESAL GRADE EDOJ DNO EMGRNO

--- ---------- -------- ------- -------- ---------- ------ ------ -------

e01 a Delhi 123453 60000 C 01-JAN-06 d01

e02 b Ggn 231456 20000 B 02-JAN-06 d01 e01

e03 c Fbd 341566 15000 A 08-FEB-05 d02 e02

e04 d Fbd 789012 5000 A 03-JUN-04 d03 e03

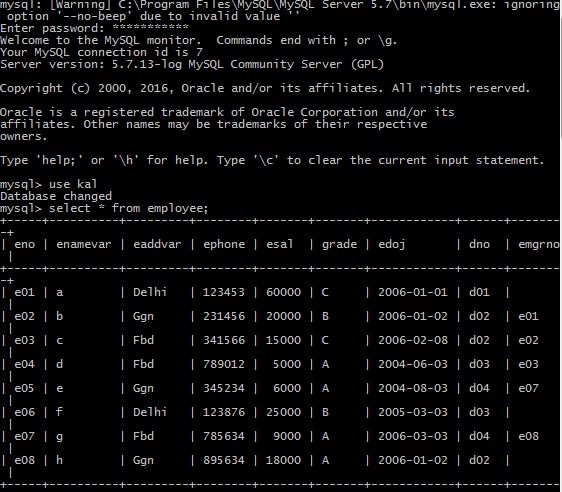
e05 e Ggn 345234 6000 A 03-AUG-04 d04 e07

e06 f Delhi 123876 25000 B 03-MAR-05 d03

e07 g Fbd 785634 9000 A 08-MAR-06 d04 e08

e08 h Ggn 895634 18000 A 02-JAN-06 d02

OUTPUT:



1. Proj\_alloc (eno char(3), pno char(3), hrs int)

ENO PNO HRS

--- --- ----------

e01 p01 4

e02 p02 6

e02 p03 3

e03 p03 2

e04 p02 8

e06 p01 9

e05 p02 10

e05 p03 20

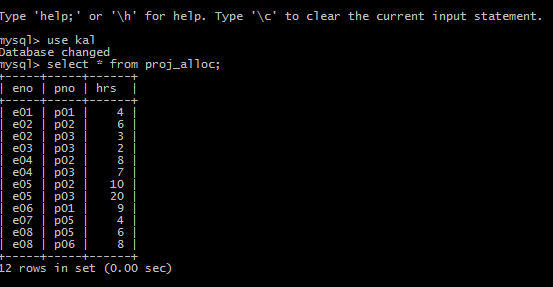
e07 p05 4

e08 p05 6

e08 p06 8

e04 p03 7

OUTPUT:



1. Dependent (eno char(3), dpname varchar(20), dpdob date, dprelation varchar(20))

ENO DPNAME DPDOB DPRELATION

--- -------------- --------- --------------------

e02 ab 01-MAR-65 father

e02 bc 02-AUG-80 brother

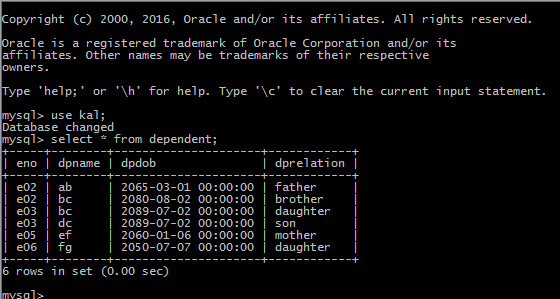
e03 dc 02-JUL-89 son

e03 bc 02-JUL-89 daughter

e05 ef 06-JAN-60 mother

e06 fg 07-JUL-50 daughter

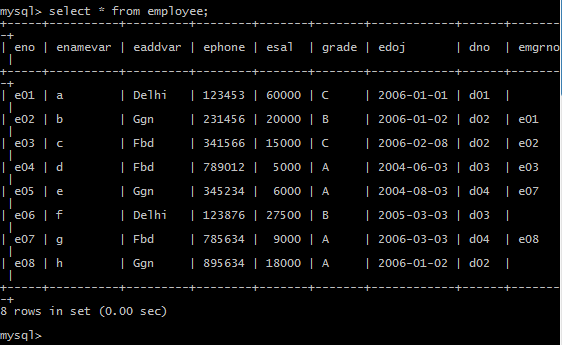
OUTPUT:



**UPDATATION**

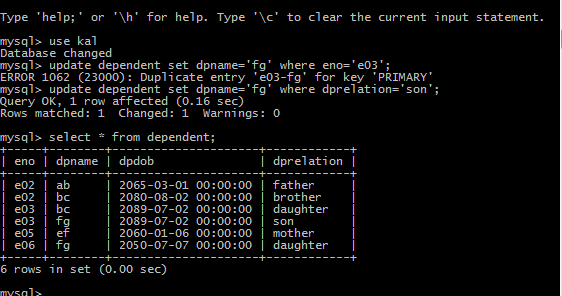
1. **Update the salary of employee e06 by increasing it by 10%.**

OUTPUT:



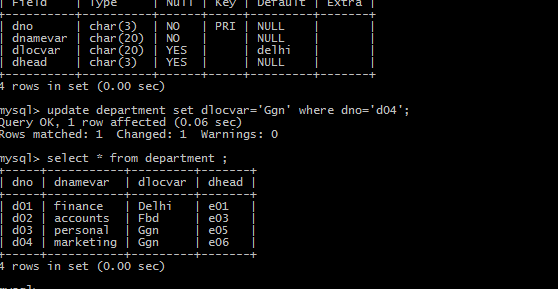
1. **Update the dependent relationship of e03 employee with dependent named fg.**

OUTPUT:

****

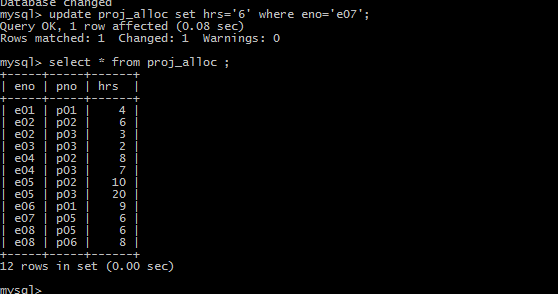
1. **Update the location of department no d04 to Ggn.**

OUTPUT:



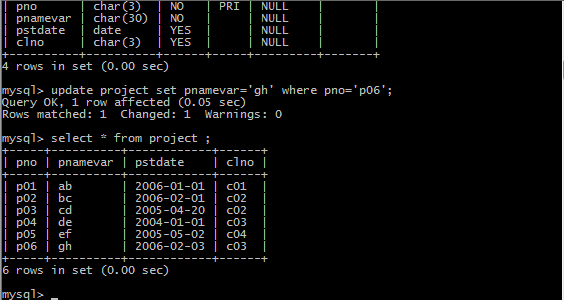
1. **Update the hours allocated for employee e07 for project p05 to 6 hrs.**

OUTPUT:



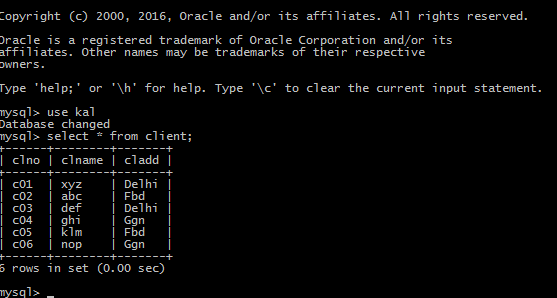
1. **Update the name of project no p06 to gh.**

OUTPUT:



1. **Update the client address of client no c06 to Ggn.**

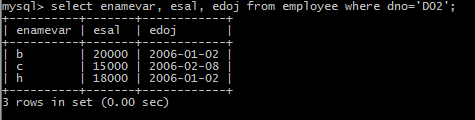
OUTPUT:



**Lab 4 – Retreival of records from single table**

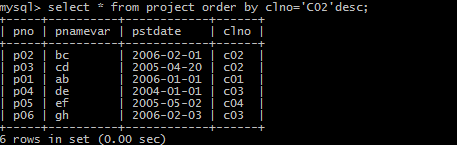
1. Retrieve employee name, employee salary and date of joining for the employees working for department no. ‘D02’.

OUTPUT:

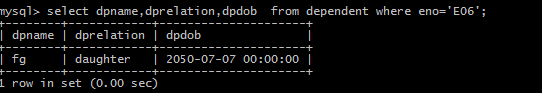


2.Find the complete details of the projects ordered by client no ‘C02’.

OUTPUT:

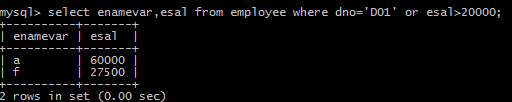


3.Find the dependent name, dependent relation and dependent date of birth for employee no ‘E06’.

OUTPUT:

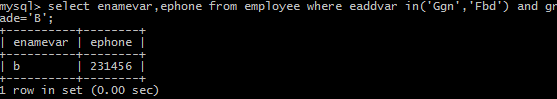
4.Retrieve the employee name and employee salary for the employees whose department is D01 or have salary more than 20000.

OUTPUT:



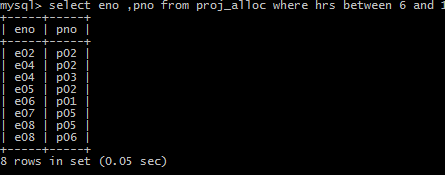
5.Retrieve the employee name and phone number for the employees whose address contains Ggn or Fbd and have grade B.

OUTPUT:



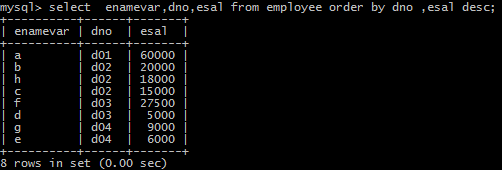
6.Find the employee numbers along with the project numbers for employees who working on a project between 6 to 10 hrs.

OUTPUT:



1. Find the employee name, department number and salary they draw in increasing order of department number and decreasing order of salary.

OUTPUT:



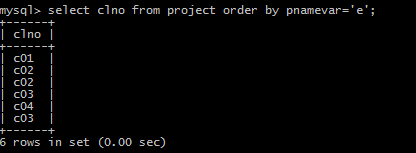
1. Find the employee names and their address for the employees who do not have any manager.

OUTPUT:



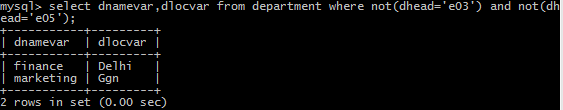
9.Find the client number who have ordered projects which have ‘e’ in their name.

OUTPUT:



10.Find the department names and their corresponding location for the departments having head other than e03 and e05.

OUTPUT:

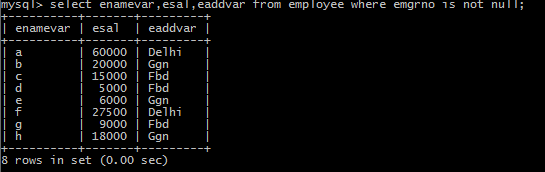


**Lab 5 – Retreival of records from single table and Relational algebraic operations using SQL**

**Lab Outcome:** Students will be able to perform relational algebraic operations - set operations, restriction, projection, and join – using SQL statement to retrieve rows.

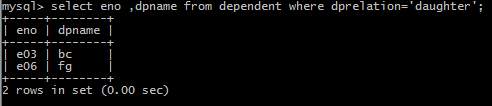
1. Find the employees name, their salary and address for employees having a manager.

OUTPUT:



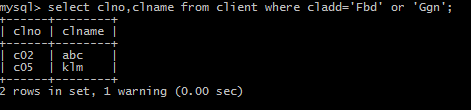
2.Find the employee number along with the dependent name who are daughter.

OUTPUT:



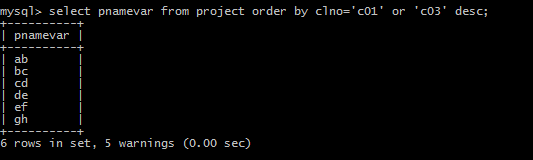
3.Find the client number and client names who are situated either in Fbd or Ggn.

OUTPUT:



4.Find the project names who have been ordered by either client no c01 or c03.

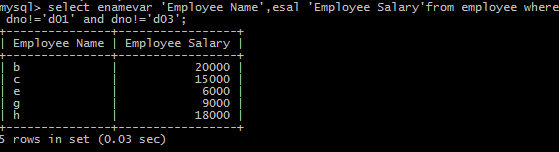
OUTPUT:



5.Display the employee name and salary under the heading

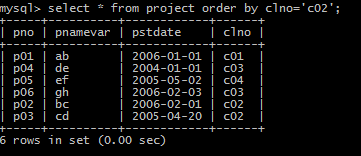
Employee Name Employee Salary

for the empoyees who donot belong to department d01 and d03.

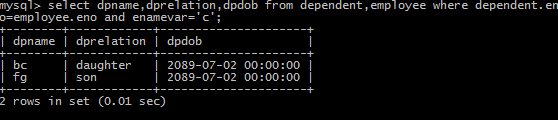
OUTPUT:

6. Find the complete details of the projects ordered by client no ‘C02’

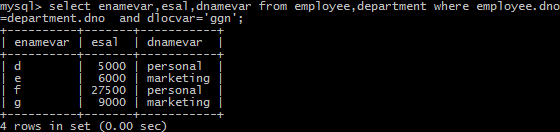
OUTPUT:



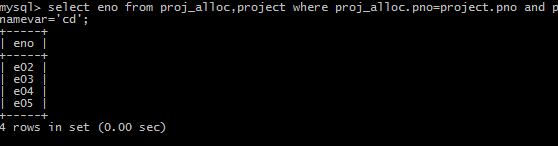
7. Find the dependent name, dependent relation and date of birth of dependents of employee named ‘c’.

OUTPUT:

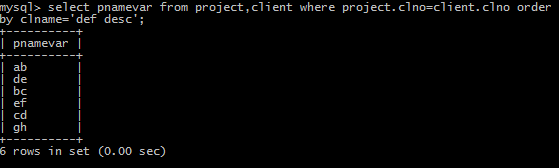
8.Retrieve or find the employee name, employee salary, department name for the employees whose department is located at Ggn.

OUTPUT:

1. Find the employee numbers who are working on the project named ‘cd’.

OUTPUT:

10. Find the project names which are ordered by ‘def’ client name.

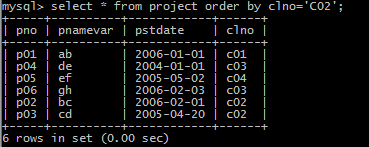
OUTPUT:

**Lab 6 – Relational algebraic operations using SQL**

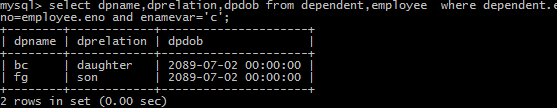
**Lab Outcome:** Students will be able to perform relational algebraic operations - set operations, restriction, projection, and join – using SQL statement to retrieve rows.

1. Find the complete details of the projects ordered by client no ‘C02’.

OUTPUT:

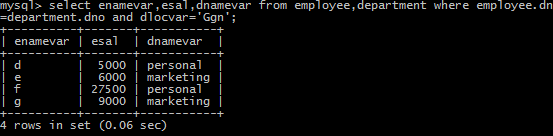


2.Find the dependent name, dependent relation and date of birth of dependents of employee named ‘c’.

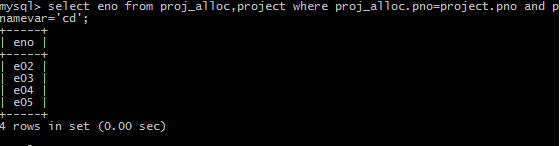
OUTPUT:

3.Retrieve or find the employee name, employee salary, department name for the employees whose department is located at Ggn.

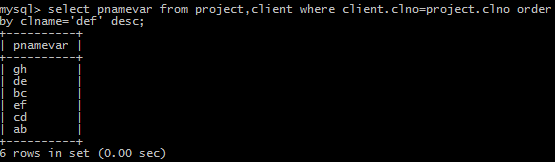
OUTPUT:



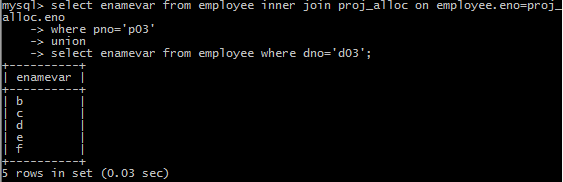
4.Find the employee numbers who are working on the project named ‘cd’.

OUTPUT:\_

5.Find the project names which are ordered by ‘def’ client name.

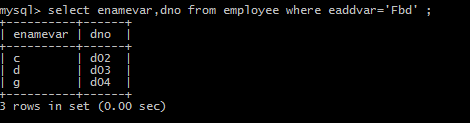
**OUTPUT:**

6.Find the employee name for the employees working for department ‘D01’ or who work on project number ‘p03’. (use set operation)

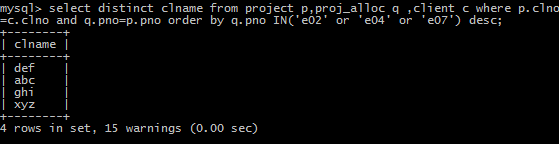
**OUTPUT:**

7.Find the employee name for the employees who live in ‘fbd’ and also have dependents. (use set operation)

**OUTPUT:**

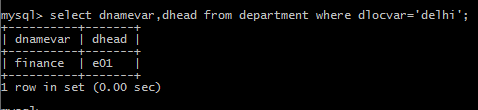
****

8.Find client names who have ordered projects which involve any of the employees ‘E02’ or ‘E04’ or ‘E07’.

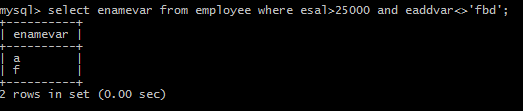
**OUTPUT:**

9.Find the department names and their corresponding department head names for the departments located at delhi.

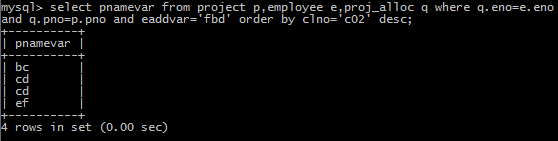
**OUTPUT:**



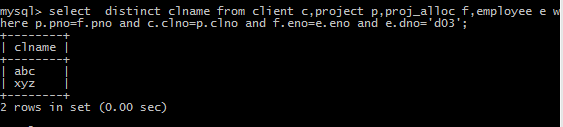
10.Find the employee name for the employees who have salary more than 25000 and who do not live in ‘fbd. (use set operation)

**OUTPUT:**

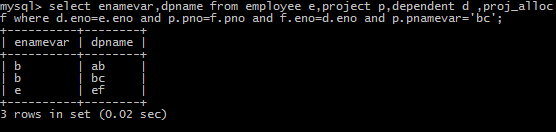
11.Find the project name for the projects which have been ordered by client c02 and have employees living in fbd working on them.

**OUTPUT:**

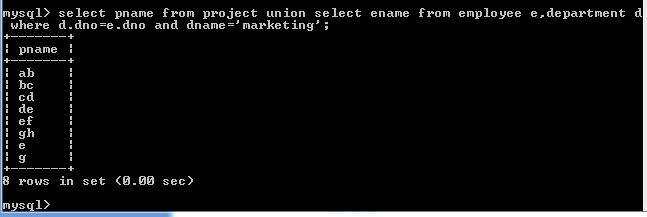
12.Find the client name of the clients who have ordered projects on which employees belonging to department d03 work.

**OUTPUT:**

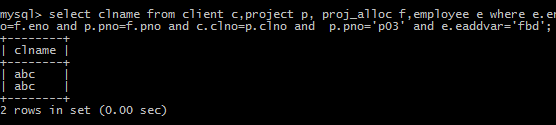
13.Find the employee name and their dependent name for the employees who are working on project named bc.

**OUTPUT:**

14.Find the project name for the projects who have either been ordered by client no c02 or have employees belonging to marketing Department working on them. (use set operation)

**OUTPUT:** ****

15.Find the client name who have ordered projects with pno p3 and who have employees living in fbd working on their ordered projects.

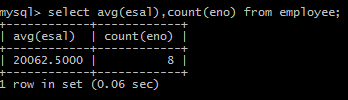
**OUTPUT:**

**Lab 7 – Aggregate, group by and outer join operations using SQL**

**Lab Outcome:** Students will be able to perform - aggregate, group by and outer join operations – using SQL statement to retrieve rows.

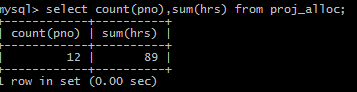
1. Find the average salary and total number of employees in the organization.

**OUTPUT:**



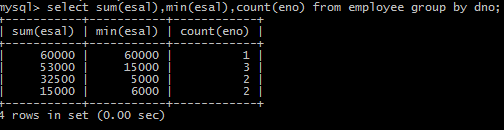
1. Find the total number of projects and total number of hours on which whole organization’s employees are working.

**OUTPUT:**

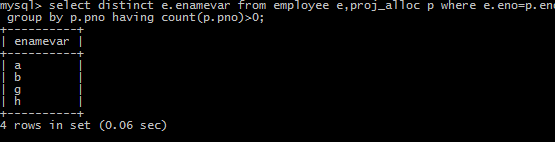
****

1. Find the total salary, minimum salary and total number of employees for each department number.

**OUTPUT:**

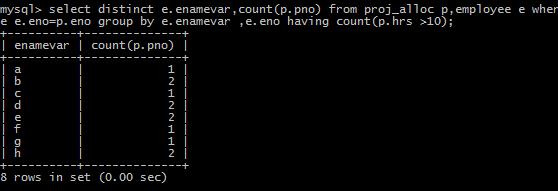
****

1. List the employee names who are working on more than one project.

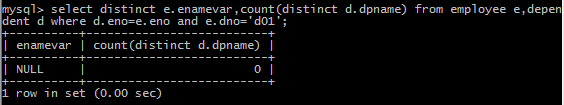
**OUTPUT:**

1. Find the employee names and their corresponding number of projects on which they are working for the employees who are working on more than 10 hours.

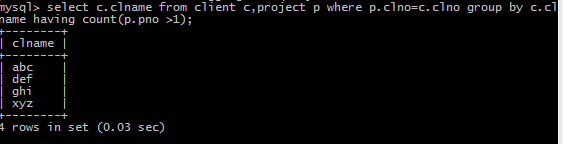
**OUTPUT:**



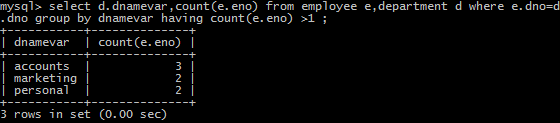
1. Find the employee names and their corresponding number of dependents for the employees belonging to department no ‘d01’.

**OUTPUT:**

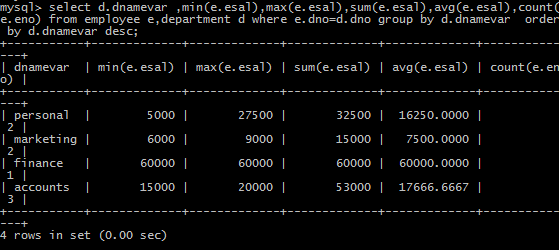
1. Find the client names for the clients who have ordered more than 1 project.

**OUTPUT:**

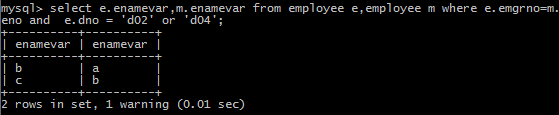
1. Find the department names and their corresponding number of employees for the departments having more than one employees and having employees with salary more than 15000.

**OUTPUT:**

1. Find the minimum salary, maximum salary, total salary, average salary, number of employees for each department name in order of department name in descending order.

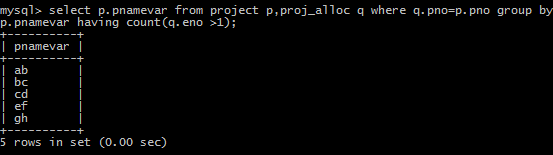
**OUTPUT:**

1. Find the employee name and their corresponding manager name for employees working for departments ‘d02’ or ‘d04’.

**OUTPUT:**

1. Find the project names which involve more than 1 employee.

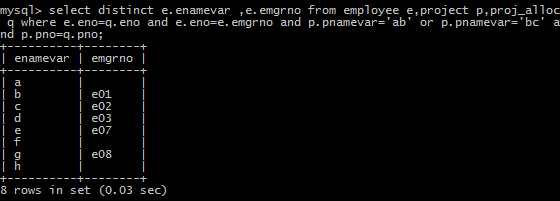
**OUTPUT:**

****

1. Find the employee names, dependent names (if any) for the employees belonging to department ‘finance’ or ‘personal’ or ‘accounts’.

**OUTPUT:**

1. Find each employee’s name and his corresponding manager’s name (if any) for the employees working on ‘ab’ or ‘bc’ projects.

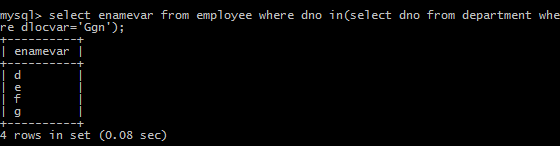
**OUTPUT:**

**Lab 8 – Nested query**

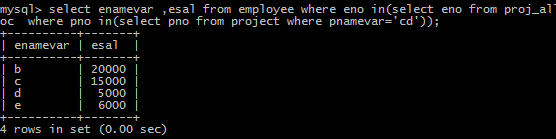
**SQL**

**Lab Outcome:** Students will be able to write nested queries (sub querying) – using SQL statement to retrieve rows.

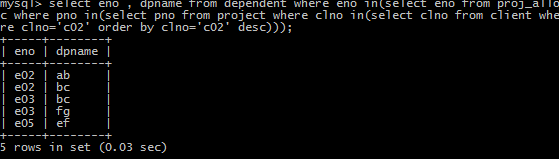
1. Find the employee names which belong to department located at ‘Ggn’.

**OUTPUT:**

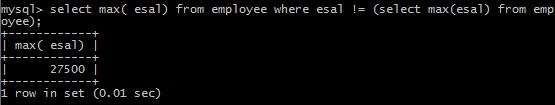
1. Find the employee names and their corresponding salary for the employees working on the project named ‘cd’.

**OUTPUT:**

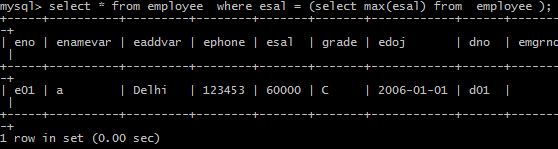
1. Find the employee no., their corresponding dependent names, relationship for the employees working on the project ordered by client ‘c02’.

**OUTPUT:**

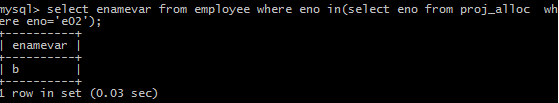
1. Find the second highest salary amongst all the employees.

**OUTPUT:**

**5.**Display details of employees whose salary is maximum.

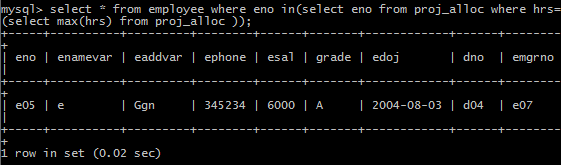
**OUTPUT:**

6.Find the employee names for the employees who are working on any of the projects on which employee ‘e2’ is working.

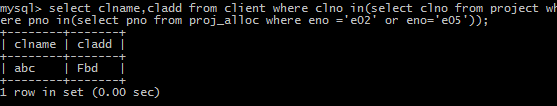
**OUTPUT:**

7.List details of employee(s) who are working for maximum hours.

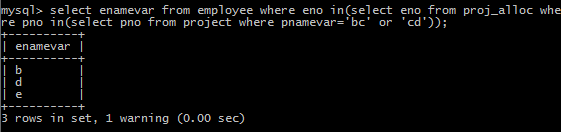
**OUTPUT:**



8.Find the client names and their addresses for the clients which have ordered for project on which employee ‘e02’ or ‘e05’ are working

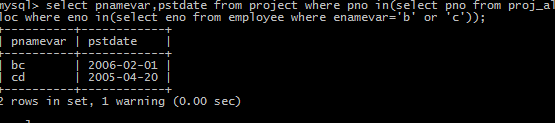
**OUTPUT:**

9.Find the employee names working either on ‘bc’ or ‘cd’ projects.

**OUTPUT:**

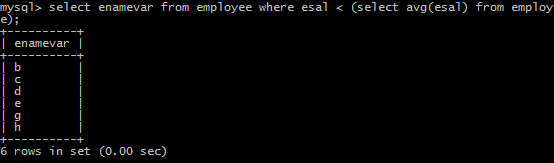
10.Find the project names and project starting date for the projects which involve employees named ‘b’ or ‘c’.

**OUTPUT:**

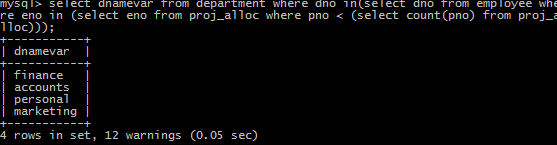


11.Find the employee names having their salary less than the average salary for all the employees.

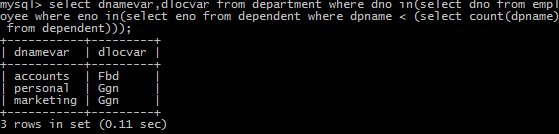
**OUTPUT:**



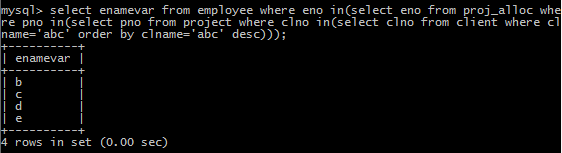
12.Find the department names which involve employees who are working on more than 1 project.

**OUTPUT:**

13.Find the department names, department location for the departments which involve employees having more than 1 dependent.

**OUTPUT:**

14.Find the employee names working on any of the projects ordered by client named ‘abc’.

**OUTPUT:**

15.Find the employee names and total hours they work on projects for those employees whose total hours is more than the average of the total hours of other employees.

**OUTPUT:**

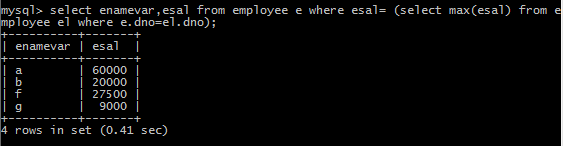
**Lab 8 – Nested co related query and views**

**SQL and Inner/Natural/Outer/Cross Join**

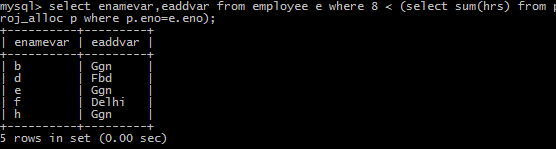
**SQL**

**Lab Outcome:** Students will be able to write nested co related queries (sub querying) and create views – using SQL statement to retrieve rows and also use appropriate kind of join for combining tuples from two or more relations

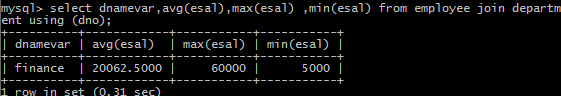
1. Find the employee names and the salary drawn for the employees drawing maximum salary in each department.

**OUTPUT:**

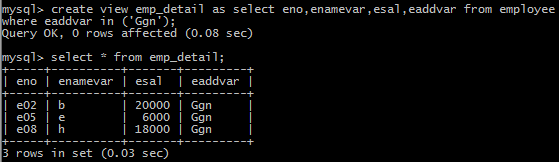
2.Find the employee name and address of the employees who are working for more than a total of 8 hours.

**OUTPUT:**

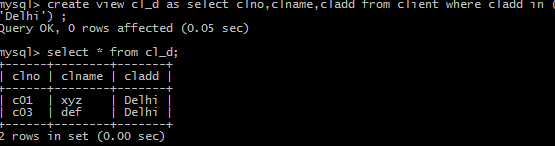
3.Create a view for retrieving department name, average salary, maximum salary and minimum salary drawn by its employees.

**OUTPUT:**

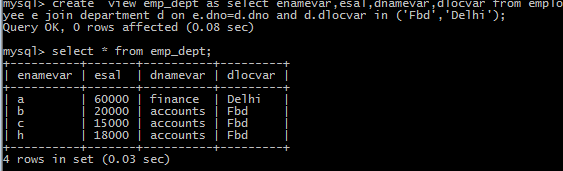
4.Create a view that gives information about employee number, employee name, salary and address with address ‘Ggn’. Also display the contents of the view created.

**OUTPUT:**

1. Create a view that gives information about client number, client name and address with address ‘Delhi’ (with restriction of Delhi clients only). Also display the contents of the view created.

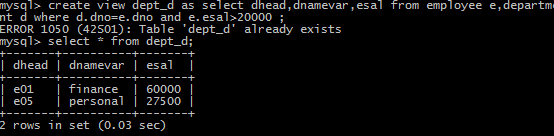
**OUTPUT:**

6.Create a view which is based on the details regarding employee name, salary, dept name, dept. location, for the employees who are located at Fbd or Delhi. Also display the contents of the view created.

**OUTPUT:**

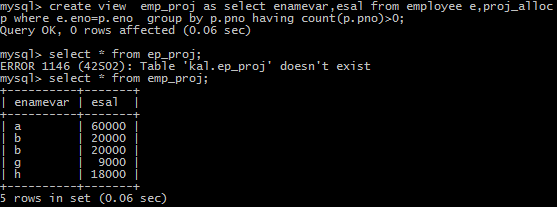
7.Create a view which is based on the details regarding the Department head names, their corresponding department names, their salary (for the department heads having salary more than Rs. 20000) for the departments having average salary more than 15000. Also display the contents of the view created.

**OUTPUT:**



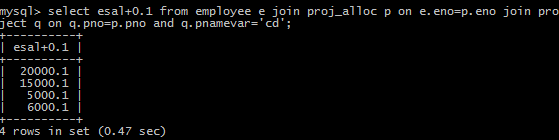
8.Create a view which is based on the details regarding the employee’s name, employee’s salary (in order of their salary) for the employees who are working on more than one the projects Also display the contents of the view created.

**OUTPUT:**

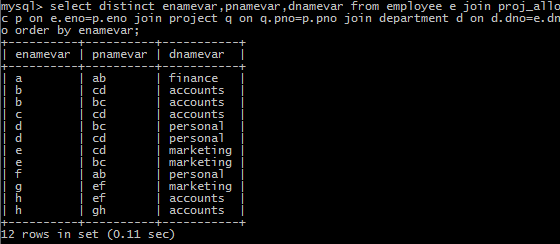
****

9.Show the resulting salaries if every employee working on the ‘cd’ project is given a 10 percent raise.

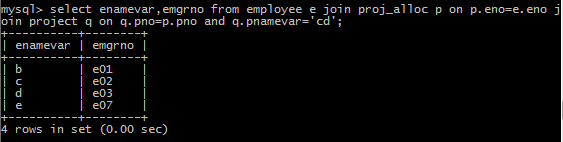
**OUTPUT:**



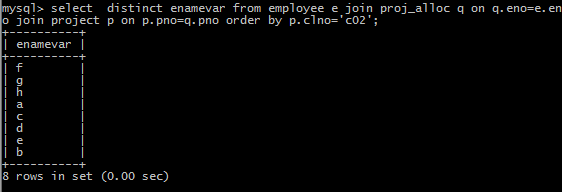
10.Retrieve a list of employees and the projects they are working on, ordered by department name and, within each department, ordered alphabetically by employee name.

**OUTPUT:**

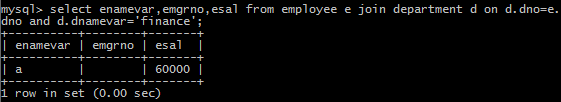
11.Retrieve the employee name along with manager name (if any) for the employees working on project name ‘cd’.

**OUTPUT:**

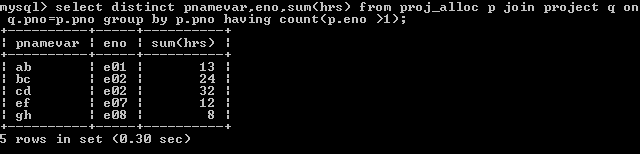
12.Retrieve the name of each employee who works on any of the projects ordered by client number ‘c02’.

**OUTPUT: **

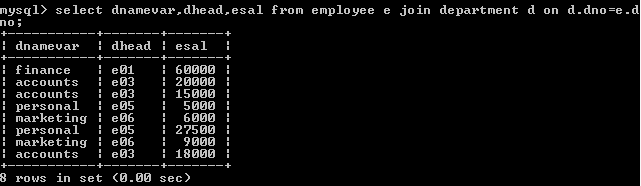
13.List the employee name, manager name (if any), and employee salary for each employee who works in the ‘finance’ department.

**OUTPUT:**

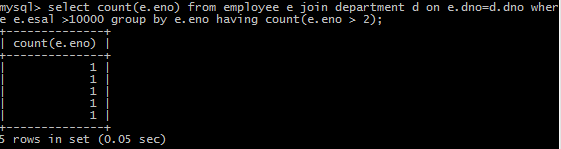
14List the project name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

**OUTPUT: **

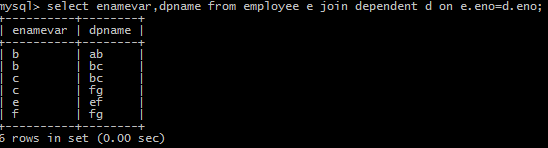
15.Display the department name, dept. head name, and manager’s salary for every department.

**OUTPUT: **

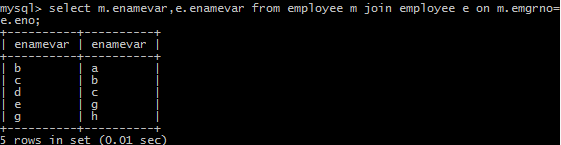
16.Count the total number of employees whose salaries exceed 10,000 in each department, but only for departments where more than two employees work.

**OUTPUT: **

17.Display employee name along with his dependent name, if any.

**OUTPUT: **

\18.Display the manager name along with their subordinate employee name.

**OUTPUT: **