Experiment -4

Study and Implementation of: Group By & having clause, Order by clause , Indexing, Views , Sub queries

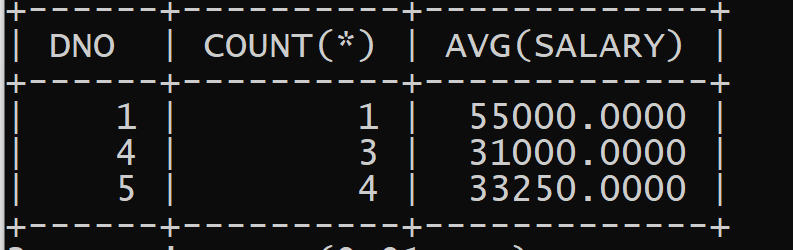
1. For each department, retrieve the department number, the number of

employees in the department, and their average salary.

**SYNTAX:**

select dno, count (\*), avg (salary) from employee group by dno;

**OUTPUT:**



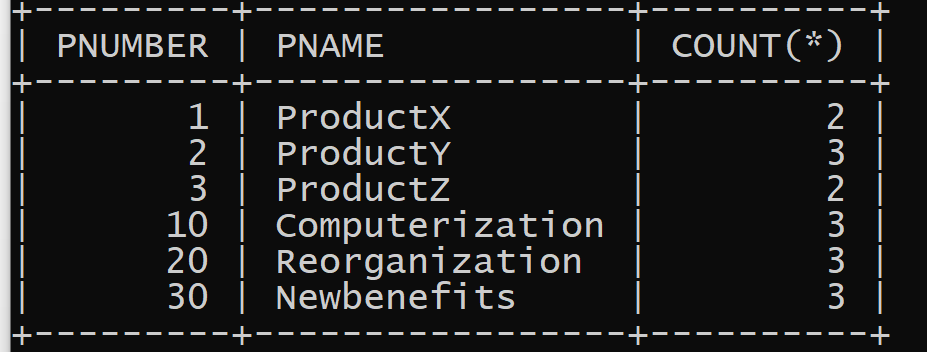
1. For each project, retrieve the project number, project name, and the number of

employees who work on that project.

**SYNTAX:**

select pnumber, pname, count (\*) from project, works\_on where pnumber=pno group by pnumber, pname;

**OUTPUT:**

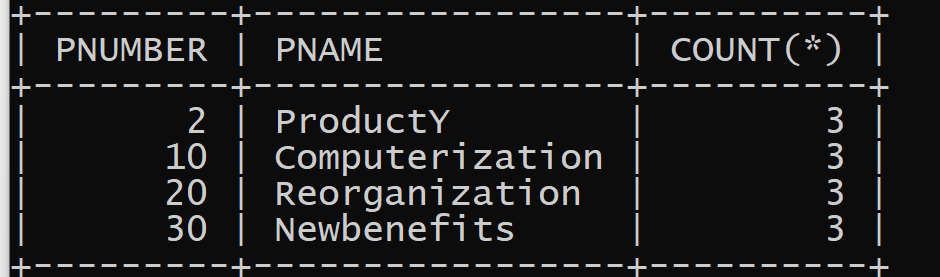


1. For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.

**SYNTAX:**

select pnumber, pname, count (\*) from project, works\_on where pnumber=pno group by pnumber, pname having count (\*) > 2;

**OUTPUT:**

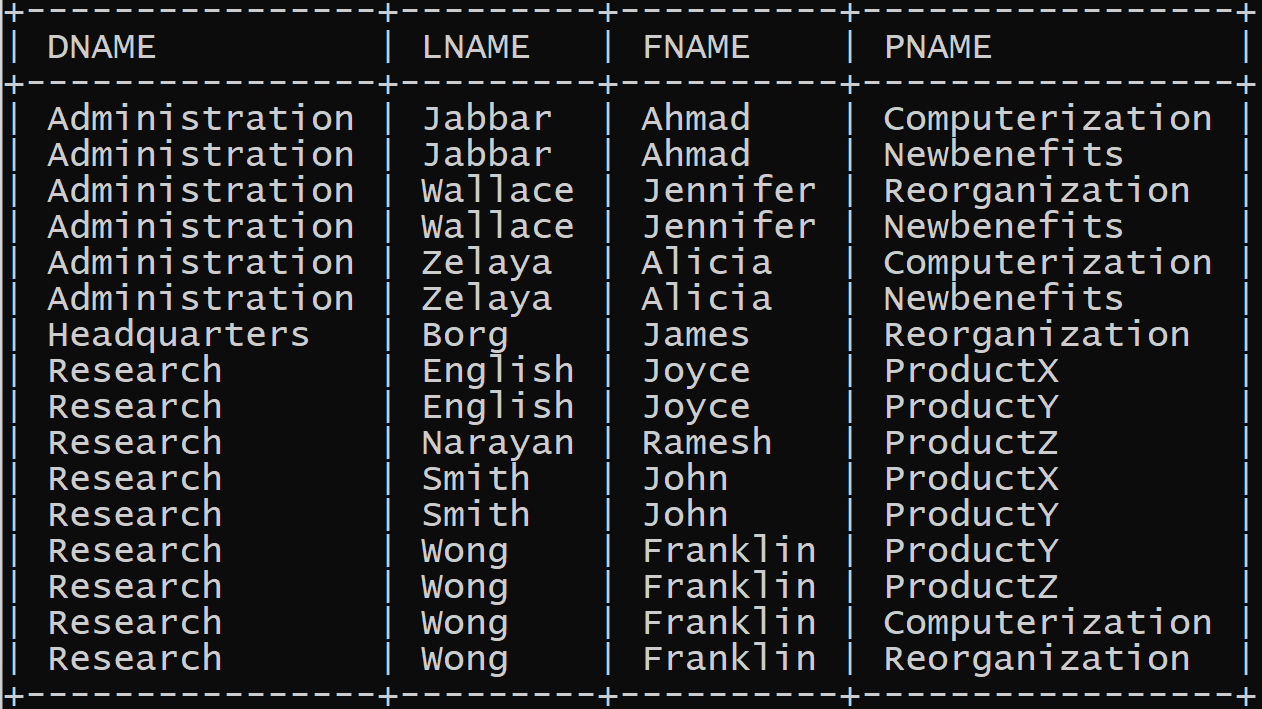


1. Retrieve a list of employees and the projects each works in, ordered by the employee's department, and within each department ordered alphabetically by employee last name.

**SYNTAX:**

select dname, lname, fname, pname from department, employee, works\_on, project where dnumber=dno and ssn=essn and pno=pnumber order by dname, lname;

**OUTPUT:**

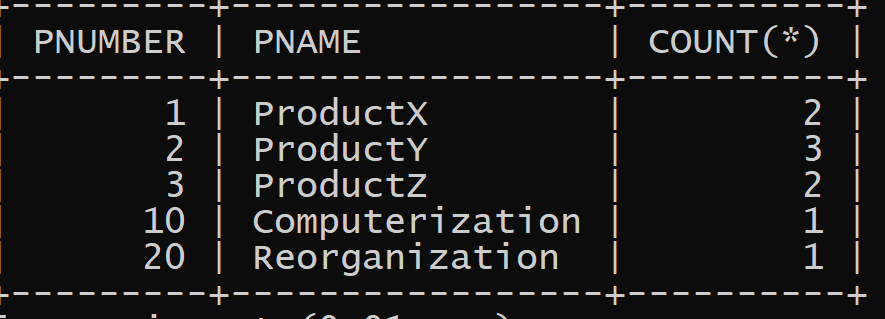


1. For each project, retrieve the project number, the project name, and the number of employees from department 5 who work on the project

**SYNTAX:**

select pnumber, pname, count (\*) from project, works\_on, employee where pnumber = pno and ssn = essn and dno = 5 group by pnumber, pname;

**OUTPUT:**



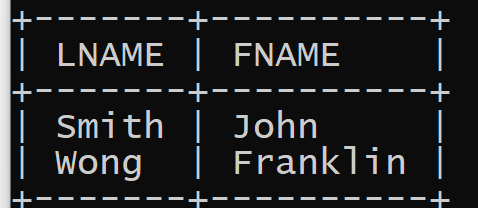
1. Retrieve the names of all employees who have two or more dependents.

**SYNTAX:**

select lname, fname from employee where (select count (\*) from dependent where

ssn=essn) >= 2;

**OUTPUT:**

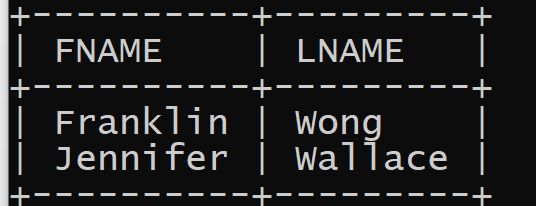


1. List the names of managers who have least one dependent.

**SYNTAX:**

select fname, lname from employee where exists (select \* from dependent where ssn=essn) and exists ( select \* from department where ssn=mgrssn );

**OUTPUT:**

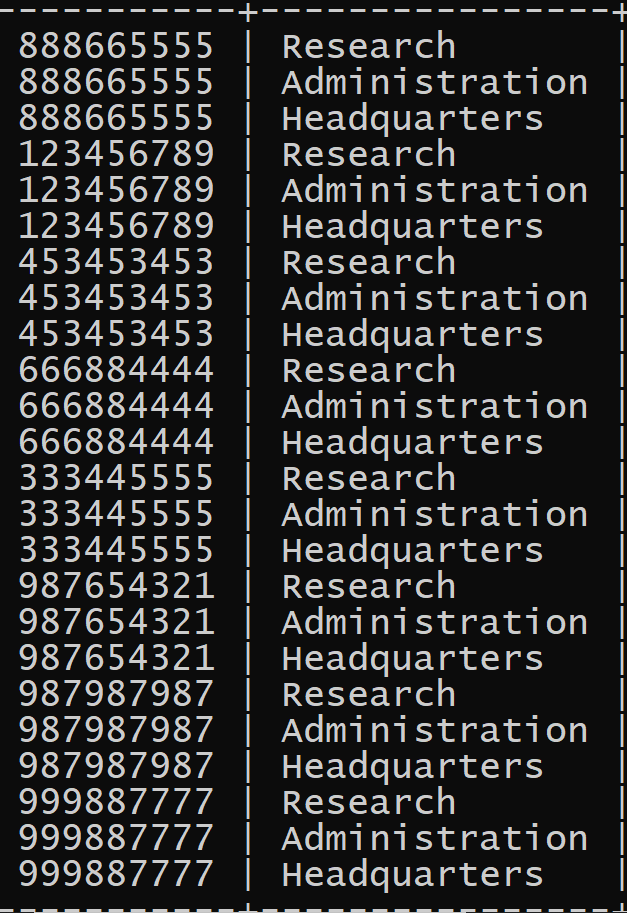


1. Select all combination of EMPLOYEE SSN and DEPARTMENT DNAME in the database

**SYNTAX:**

select ssn, dname from employee, department;

**OUTPUT:**



1. Select the CROSS PRODUCT of the EMPLOYEE and DEPARTMENT relations

**SYNTAX:**

select \* from employee cross join department;

**OUTPUT:**

A screen shot of a computer

Description automatically generated A screen shot of a computer screen

Description automatically generated

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

**SYNTAX:**

select dname,lname,fname,pname from department,employee,works\_on,project where dnumber=dno and ssn=essn and pno=pnumber order by dname desc,lname asc,fname asc;

**OUTPUT:**

A screen shot of a computer

Description automatically generated

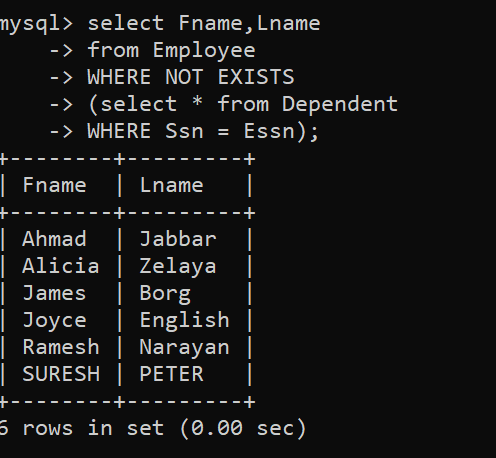
1. Retrieve the name of each employee who have no dependent.

**SYNTAX:**

select Fname,Lname from Employee WHERE NOT EXISTS (select \* from Dependent

WHERE Ssn = Essn);

**OUTPUT:**



1. Display fname, lname and address of employees who are working in research department.(USING JOINS)

**SYNTAX:**

select fname, lname, address from employee join department on dno = dnumber where

dname = 'research';

**OUTPUT:**

A black and white screen with white text

Description automatically generated

1. Create a view for following query:

**SYNTAX:**

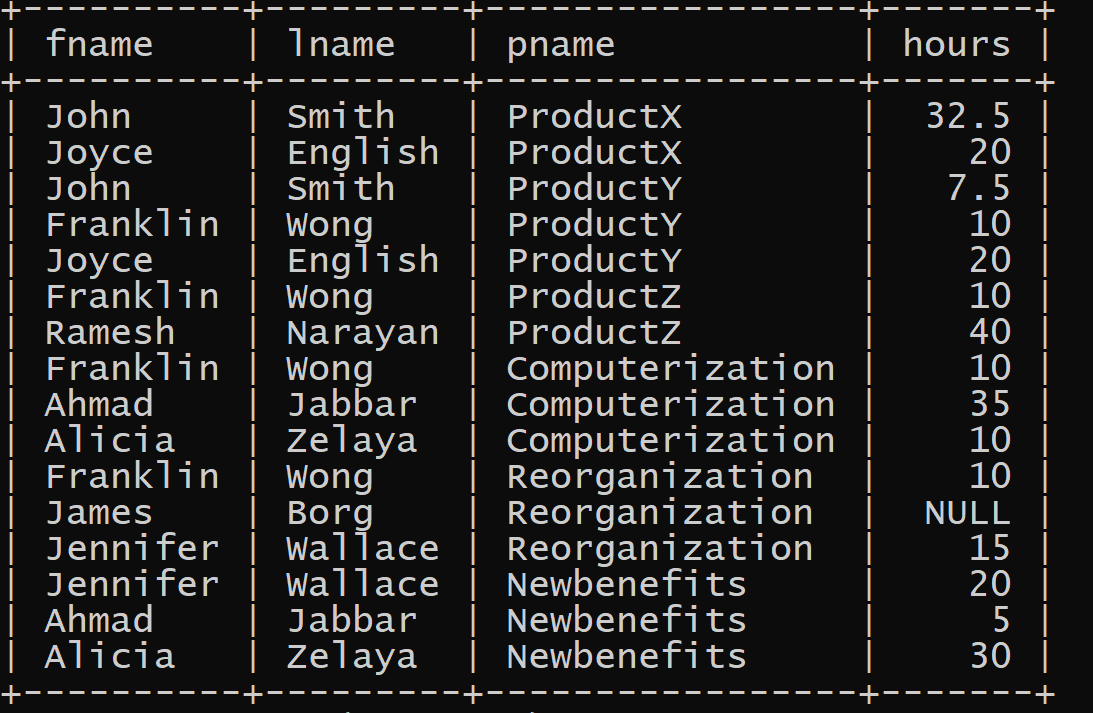
create view works\_on1 as select fname, lname, pname, hours from employee, project, works\_on where ssn = essn and pno = pnumber;

To display view:

**SYNTAX:**

select \* from works\_on1;

**OUTPUT:**



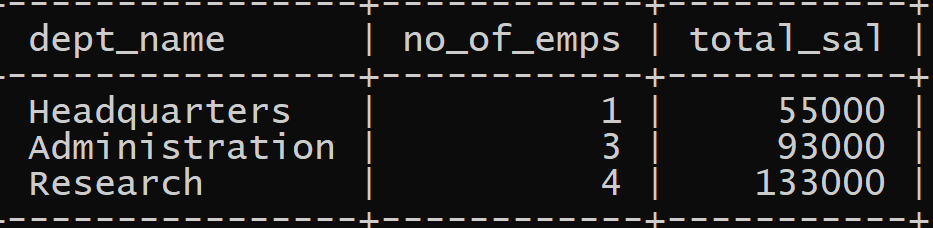
1. Write a query to create view:-

**create view dept\_info(dept\_name, no\_of\_emps, total\_sal) as select dname, count (\*), sum (salary) from department, employee where dnumber = dno group by dname;**

**SYNTAX:**

select \* from dept\_info;

**OUTPUT:**



1. Write a query to create index for any one existing table.

**SYNTAX:**

create index idx\_pname on employee (fname,lname);

Query OK, 0 rows affected (0.08 sec)

Records: 0 Duplicates: 0 Warnings: 0

There are many types of indexes in databases, including clustered, non-clustered, unique, composite, hash, B-tree, text, bitmap, and geospatial indexes:

**SYNTAX:**

show index from employee;

**OUTPUT:**

