

DBMS Lab File

(2024)

|  |  |
| --- | --- |
| Made by – | |
| Name  Course  Sap Id | Satvik Raj  B. tech CSE  500119624 |

# Experiment 6: Use of Inbuilt functions and relational algebra operation

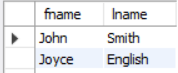
**Objective:** To understand the use of inbuilt function and relational algebra with sql query.

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.

I/O- select fname,lname from employee where ssn in (select essn from works\_on where pno =(select pno from project where pname='ProductX') and hours>10) and dno=5;



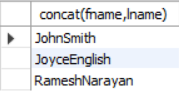
1. List the names of all employees who have a dependent with the same first name as themselves.

I/o- select fname,lname from employee where fname in (select dependent\_name from dependent where ssn=essn);

there is no employee with the same name as its dependent

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

I/o- select concat(fname,lname) from employee where super\_ssn=(select ssn from employee where fname="Franklin" and lname="Wong")



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

I/O- select fname,lname from employee where ((select count(\*) from project)=(select count(\*) from works\_on where ssn=essn));

No employee works on every project



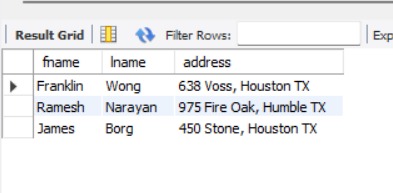
1. Retrieve the names of employees who do not work on any project.

I/O- select fname,lname from employee where ((select count(\*) from works\_on where ssn=essn)=0);

No employee is vacant or not working 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.

I/O- select fname,lname,address from employee where ssn in (select essn from works\_on where pno in (select pnumber from project where plocation=”Houston”) and DNO in (select Dnumber from dept\_locations where dlocation=”Houston”));



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

I/O- 

select lname from employee where ssn in (select mgr\_ssn from department) and ssn not in (select essn from dependent);

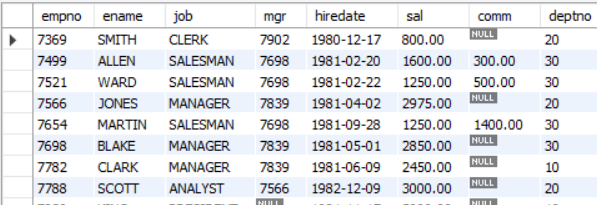
# Experiment 7: Use of Inbuilt functions and relational algebra operation

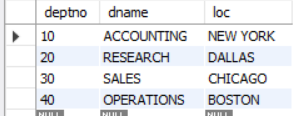
**Objective:** To understand the use of inbuilt function and relational algebra with sql query.

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

I/O- select \* from emp;

Select \* from dept;





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

1. Retrieve average salary of all employees.

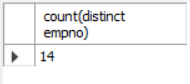
I/O-Select avg(sal) from emp;



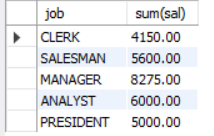
1. Retrieve the number of employees.

I/O- select count(ename) from emp;

1. Retrieve distinct number of employee.

I/O- select count(distinct empno) from emp;

1. Retrieve total salary of employee group by job.

select job,sum(sal) from emp group by job;

1. Display the employee information with maximum salary.

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



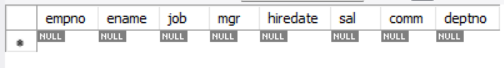
1. Find the highest paid employee in department 10.

select \* from emp where (sal = (select max(sal) from emp)) and deptno=10;

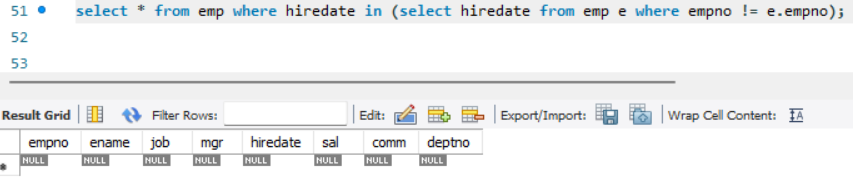


1. List the emps whose sal is equal to the average of max and minimum.

Select \* from emp where sal=(select (max(sal)+min(sal))/2 from emp);

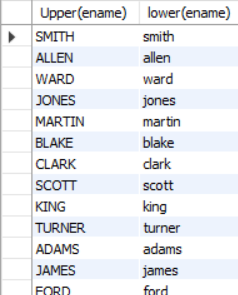


1. List the emps who joined in the company on the same date.



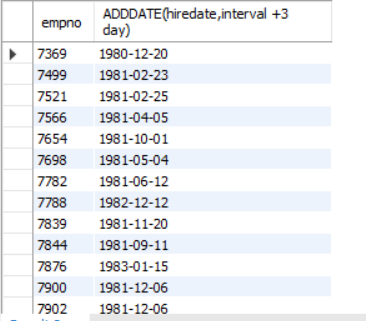
1. Display the employee names in upper and lower case.

Select Upper(ename),lower(ename) from emp;



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

Select empno, adddate(hiredate,interval +3 day) from emp;



# Experiment 8: 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:

**I/O- Student (sid, sname, age)**

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

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

create table student (

sid int primary key,

sname varchar(20),

age int

);

create table Matches (

mid varchar(20) primary key,

mname varchar(50),

venue varchar(50)

);

create table play(

sid int, mid varchar(20),

pday date ,

primary key (sid,mid),

foreign key (sid) references student(sid),

foreign key (mid) references matches(mid)

);

1. Populate all the tables.

I/o- insert into student values

(1,'Amit',20),

(2,'Ajay',20),

(3,'Anu',19),

(4,'Ash',18);

INSERT INTO matches (mid, mname, venue) VALUES

('B10', 'ODI Final', 'Mumbai'),

('B11', 'Ranji', 'Delhi'),

('B12', 'Knights', 'Delhi'),

('B13', 'Knicks', 'Mumbai');

insert into play values

(1,'B10','2024-05-12'),

(2,'B10','2024-05-12'),

(2,'B11','2024-06-13'),

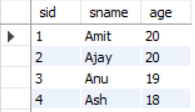
(3,'B11','2024-06-12'),

(3,'B12','2024-07-14'),

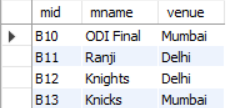
(4,'B13','2024-08-18'),

(4,'B12','2024-09-22');

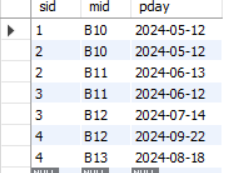
Select \* from student;



Select \* from matches;



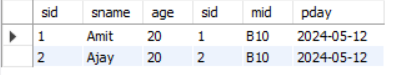
Select \* from play;



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

I/O- Select \* from student

Inner join play on student.sid=play.sid where(play.mid=’B10’)



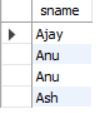
1. Find the name of matches played by Amit.

I/O-select mname from matches inner join play on matches.mid=play.mid where play.sid in (select sid from student where sname='Amit') ;



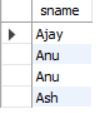
1. Find the names of students who have played a match in Delhi.

I/O- select sname from student s inner join play p on s.sid=p.sid where p.mid in (select mid from matches where venue = 'Delhi');



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

I/O- select distinct sname from student s inner join play p on s.sid=p.sid ;

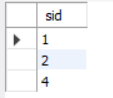


1. Find the ids and names of students who have played two different matches on the same day.

SELECT DISTINCT p.sid, sname FROM play p INNER JOIN student s on p.sid = s.sid WHERE day IN (SELECT day FROM play WHERE p.sid = sid AND p.mid <> mid);

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

select s.sid from student s inner join play p on s.sid=p.sid where p.mid in (select mid from matches where venue = 'Mumbai' or 'Delhi');



1. Find the average age of students.

Select avg(age) from student;



# EXPERIMENT-9: 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 |  |

CREATE TABLE EMPLOYEES (

Employee\_id CHAR(10) PRIMARY KEY,

First\_Name CHAR(30) NOT NULL,

Last\_Name CHAR(30) NOT NULL,

DOB DATE,

Salary NUMBER(25) NOT NULL,

Department\_id CHAR(10)

);

INSERT INTO EMPLOYEES VALUES ('E001', 'John', 'Doe', TO\_DATE('1985-06-15', 'YYYY-MM-DD'), 75000, 'D001');

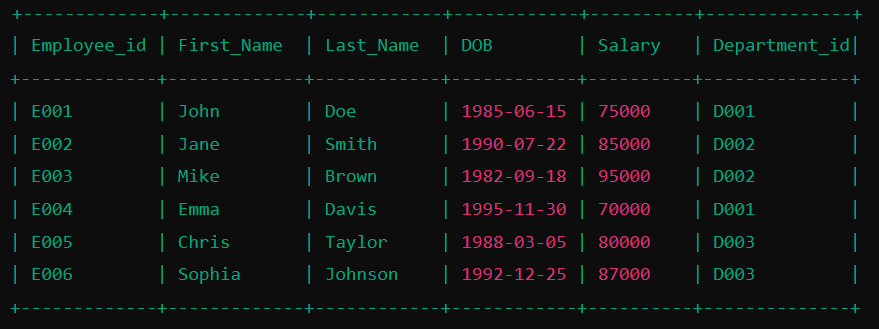
INSERT INTO EMPLOYEES VALUES ('E002', 'Jane', 'Smith', TO\_DATE('1990-07-22', 'YYYY-MM-DD'), 85000, 'D002');

INSERT INTO EMPLOYEES VALUES ('E003', 'Mike', 'Brown', TO\_DATE('1982-09-18', 'YYYY-MM-DD'), 95000, 'D002');

INSERT INTO EMPLOYEES VALUES ('E004', 'Emma', 'Davis', TO\_DATE('1995-11-30', 'YYYY-MM-DD'), 70000, 'D001');

INSERT INTO EMPLOYEES VALUES ('E005', 'Chris', 'Taylor', TO\_DATE('1988-03-05', 'YYYY-MM-DD'), 80000, 'D003');

INSERT INTO EMPLOYEES VALUES ('E006', 'Sophia', 'Johnson', TO\_DATE('1992-12-25', 'YYYY-MM-DD'), 87000, 'D003');



2. View-Related Queries

**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.

CREATE VIEW emp\_view AS

SELECT Employee\_id, Last\_Name, Salary, Department\_id

FROM EMPLOYEES;



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

ALTER TABLE EMPLOYEES MODIFY (Last\_Name NULL, Salary NULL);

INSERT INTO emp\_view (Employee\_id, Last\_Name, Salary, Department\_id)

VALUES ('E007', 'Miller', NULL, 'D004');

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

UPDATE emp\_view

SET Salary = 60000

WHERE Employee\_id = 'E007';

DELETE FROM emp\_view

WHERE Employee\_id = 'E007';

DROP VIEW emp\_view;

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

CREATE VIEW salary\_view AS

SELECT Employee\_id, First\_Name, Last\_Name, (Salary \* 12) AS Annual\_Salary

FROM EMPLOYEES

WHERE Department\_id = 'D002';

SELECT \* FROM salary\_view;

