

PRACTICAL 2

Manipulating data

A)performance of insert statement

B)use of update statement

C)use of delete statement

A) performance of insert statement

```
insert into dept values(10,'finance','london','15-jul-2008')
```

```
insert into dept values(20,'accounts','canada','05-feb-2003')
```

```
insert into dept values(30,'IT','uk','02-apr-2013')
```

```
insert into dept values(40,'marketing','usa','03-mar-2010')
```

```
insert into dept values(50,'HR','abc','05-apr-2008')
```

```
select *
```

```
from dept
```

```
insert into emp values(1,'allen',50000,'03-mar-2008',400,'101',10)
```

```
insert into emp values(2,'hayes',35000,'02-apr-2008',500,'102',20)
```

```
insert into emp values(3,'smith',75000,'03-mar-2005',200,null,10)
```

```
insert into emp values(4,'john',70000,'04-aug-2009',null,'100',null)
```

```
insert into emp values(5,'bob',35000,'05-sep-2013',100,'105',30)
```

```
select *
```

```
from emp
```

B) use of update statement

1) Display the details of emp table.

```
select *  
from emp
```

2) Display emp with doj.

```
select ename,doj  
from emp
```

3) Display those employee who are not belonging to any dept.

```
select ename  
from emp  
where dno is null
```

4) Assign 2% comm to those employees who are not belonging to any dept.

```
update emp  
set commission=commission+(commission*0.02)  
where dno=10
```

5) Increase the salary of employee by 10% who comm below 350.

```
update emp  
set commission=commission+(commission*0.02)  
where dno=10
```

6) Display emp with jobcode and decrease the salary by 2000 who are belonging to department 30 select ename,jobcode,salary-2000 modified salary.

```
from emp  
where dno=30
```

C) Use of delete statement

1) Remove those emp who joined on 05-mar-2013.

```
delete
```

```
from emp
```

```
where doj = '05-mar-2013'
```

2) Remove the emp from the table.

```
delete
```

```
from emp
```

3) Remove the emp table from the system

```
drop table emp
```

4) Create a new table from existing table

```
select *
```

```
into employee from emp
```

PRACTICAL 3

A) SQL basic statements

1) Display the details of employee

```
select *  
from employee
```

2) Display employee with corresponding salary and date of joinin

```
select ename,salary,doj  
from employee
```

3) Display those employees whose salary below 80000

```
select ename,salary  
from employee  
where salary<80000
```

4) Display employee with its annual salary

```
select ename,(salary*12)  
from employee
```

5) Display the department from employee table

```
select ename,dno  
from employee
```

6) Display departments without duplication

```
select distinct dno  
from employee
```

7) Display first two records of employee

```
select top 2 *  
from employee
```

8)Display those employees whose commisiion is in the range of 100-500

```
select ename,commission
from employee
where commission between 100 and 500
```

9)Display employees who are not getting commission

```
select ename,commission
from employee
where commission is null
```

10)Display employee whose name start with A

```
select ename
from employee
where ename like 'a%'
```

11)Display those employee with name end with b

```
select ename
from employee
where ename like '%b'
```

11)Display employee whose name comes in the range of A-P

```
select ename
from employee
where ename like '[a-p]%'
```

12)Display employee whose name not in the range of A-P

```
select ename
from employee
where ename like '[^a-p]%'
```

13)Display employee belonging to the department 10,20

```
select ename,dno
from employee
where dno in (10,20)
```

14)Display employee who are not from the department 20,30

```
select ename,dno
from employee
where dno not in (10,20)
```

15)Display employee who joined on 5th Sept 2015 neither belonging to dept 20 nor getting any condition

```
select ename,dno,dof,commission
from employee
where dno <>20 and dof='5-sep-2013' and commission is null
```

B)Sorting of data

1)Arrange the employee table with respect to its name

```
select *
```

```
from employee order by ename
```

2)Sort the employee according to job with commission multiplied by 10%

```
select ename,(commission*0.10)
```

```
from employee
```

```
order by jobcode
```

3)Sort the employee in descending order of salary

```
select ename,salary
```

```
from employee
```

```
order by salary desc
```

4)Sort the employee according to column no 4

```
select *
```

```
from employee
```

```
order by 4
```

5)Sort the employee whose salary is in the range of 30000 to 50000

```
select *
```

```
from employee
```

```
where salary between 30000 and 50000
```

```
order by salary
```

C)Single row functions

select abs(-50)

select abs(50) row

select ceiling(50.60)

select floor(50.60)

select log(10)

select log10(10)

select pi()

select power(50,2)

select rand()

select rand(20)

select round(567.865,2)

select square(2)

select sqrt(4)

select sin(30)

select sin(-20)

select sin(0)

select ascii('A')

select ascii('a')

select char(65)

select ename

from employee

where charindex('n',ename)>0

select substring('archee',2,2)

select len('archee')

select lower(ename)

from employee

select upper(ename)

from employee

select rtrim(' archee')


```
select reverse('arcee')  
select replicate('arcee',2)  
select replace('hello','l','o')
```

PRACTICAL 4

A) Displaying data from multiple tables

1) Find the details of emp and dept tables based on matching column (EQUI join/Inner join)

```
select *  
from emp e,dept d  
where e.dno=d.dno
```

```
select *  
from emp e  
inner join dept d  
on e.dno=d.dno
```

2) Display employee who works for department 10 and department located in London

```
select ename  
from emp,dept  
where emp.dno=10 and location='london'
```

3) Display those employees who have same date of joining (self join)

```
select e1.ename,e1.doj  
from emp e1, emp e2  
where e1.doj=e2.doj and  
e1.dno<>e2.dno
```

4) Display those employees who belong to different department (Non EQUI Join)

```
select distinct e1.ename  
from emp e1, emp e2  
where e1.dno<>e2.dno
```

5) Display the details of both tables based on matching column and also display the null values (Outer join)

```
select *  
from emp e  
left join dept d  
on e.dno=d.dno
```

```
select *  
from emp e  
right join dept d  
on e.dno=d.dno
```

```
select *  
from emp e  
full join dept d  
on e.dno=d.dno
```

B) Referential integrity error

- 1) inserting a new child record

```
insert into emp values(6,'vanshita',35000,'2015-12-30',200,105,100)
```

- 2) updating a parent record

```
update dept  
set dno=100  
where dno=20
```

- 3) removing the parent record

```
delete  
from dept  
where dno=10
```

PRACTICAL 5

A) Use of sql aggregate functions

1) Display the no of emp

```
select count(eno)
from emp
```

2) Display no of emp working for dept 10

```
select count(eno)
from emp
where dno=10
```

3) Display total salary of emp

```
select sum(salary)
from emp
```

4) Display lowest commission of emp

```
select min(commission)
from emp
```

5) Display highest salary of emp

```
select max(salary)
from emp
```

6) Display average commission of emp

```
select avg(commission)
from emp
```

7) Use all the aggregate function in one select statement

```
select count(*), sum(salary), min(commission), max(salary),avg(commission)
from emp
```

B) Aggregating data using group functions

- 1) Display salary according to dept

```
Select salary,dno  
From emp
```

- 2) Display total salary for each dept

```
select sum (salary)  
from emp  
group by dno
```

- 3) Display name of emp with its max salary

```
select max(salary),ename  
from emp  
group by ename
```

- 4) Display dept no with no of emp with salary more than 30000

```
select dno, count(eno)  
from emp  
where salary>30000  
group by dno
```

- 5) Display dept no with average salary beyond 20000

```
select dno,avg(salary)  
from emp  
group by dno  
having avg (salary)>20000
```

- 6) Display no of emp whose commission between 300-400 atleast 2 emp working for the dept

```
select dno,count(eno)  
from emp  
where commission between 300 and 400  
group by dno  
having count(eno)>=2
```

- 7) Display total salary of those emp whose name starts with A for each job having total salary below 50000 arrange the output according to job

```
select sum(salary),jobcode  
from emp  
where ename like (a%)  
group by jobcode  
having sum(salary)<50000
```

PRACTICAL 6

A) Use of set operators

1) Display the employee working for either department 10 or department 20

```
select ename
from employee
where dno=10
union
select ename
from employee
where dno=20
```

```
select ename
from employee
where dno=10
union all
select ename
from employee
where dno=20
```

2) Display common records of both tables

```
select dno
from employee
intersect
select dno
from dept
```

3) Display uncommon records of both tables

```
select dno
from employee
except
select dno
from dept
```

4) Display employees working for dept 10 but not for the dept 20

```
select dno
from employee
where dno=10
except
select dno
from employee
where dno=20
```

B) Date, time functions

```
select getdate()
```

```
select dateadd(day,30,getdate())
```

```
select datediff(day,'10-04-2024',getdate())
```

```
select datepart(month,getdate())
```

```
select isdate('10-04-2024')
```

```
select current_timestamp
```

```
select convert(varchar,getdate(),1)
```

```
select convert(varchar,getdate(),2)
```

```
select convert(varchar,getdate(),3)
```

```
select convert(varchar,getdate(),4)
```

```
select convert(varchar,getdate(),5)
```

```
select convert(varchar,getdate(),6)
```

```
select convert(varchar,getdate(),7)
```

```
select convert(varchar,getdate(),8)
```

```
select convert(varchar,getdate(),9)
```

```
select convert(varchar,getdate(),10)
```

PRACTICAL 7

A) Basic subquery

1) Display the details of both the tables based on matching column.

```
select *  
from emp  
where dno in (select dno from dept)
```

2) Display the employee name who works for dept 10 and dept located in London

```
select ename  
from emp  
where dno=10 and dno in (select dno from dept where location='London')
```

3) Display those employee who have same doj (outer reference/correlated subquery)

```
select ename  
from emp e1  
where doj in (select doj from emp e2 where e1.eno<>e2.eno)
```

4) Display those employees who belong to different department

```
select ename  
from emp  
where dno not in (select dno from emp)
```


B) Advance subquery

1) Display those employees who works for particular department

```
select ename,dno  
from employee  
where exists (select * from employee)
```

2) Display those employees who works for accounts department

```
select dname,ename  
from dept,employee  
where dname in (select dname from dept where dname='Accounts')
```

3) Display employees whose salary is greater than average salary

```
select ename,salary  
from employee  
where salary > (select avg(salary) from employee)
```

4) Display those employees who are not belonging to any department

```
select ename,dno  
from employee  
where not exists (select * from employee)
```

5) Display employee who are earning more than average salary for the dept where they are working

```
select ename  
from emp  
where salary > (select avg(salary) from emp( where dno in (select dno from dept))
```

6) Display employees whose salary is greater than maximum salary according to the dept

```
select ename,salary,dno  
from emp  
where salary > all(select max(salary) from emp group by dno)
```

7) Display employees whose commission is beyond the commission of employee whose id is 3

```
select ename,commission,eid  
from emp  
where commission > (select commission from emp where eno=3)
```

8) List employee getting salary more than some of the employee working for dept 30

```
select ename,salary
from emp
where salary>any(select salary from emp where dno=30)
```

9) Display emp whose earning salary more than every emp working for dept IT or dept Finance

```
select ename,salary
from emp
where salary >all(select salary from emp e ,dept d where dname='IT' or
dname='Finance')
```

10) List those emp who are in the same dept having O in their name with salary less than maximum salary

```
select ename,dno,salary
from emp
where dno in(select dno in dept where ename like '%o%' and salary<(select
max(salary) from emp))
```

11) Transfer all the emp who belongs to dept 30 to employee table

```
insert into employee(select * from emp where dno=30)
```

12) Increase 10% commission for employees whose dept is in UK

```
update emp
set commission=commission*0.1
where dno in(select dno from emp where location='UK')
```

13) Remove those emp who works for HR department

```
delete from emp
where dno in (select dno from dept where dname='HR')
```

14) Display those emp who joined after Hayes

```
select ename
from emp
where doj>(select doj from emp where ename='Hayes')
```

```
create table Worker(worker_id varchar(10),
                    first_name char(10),
                    last_name char(10),
                    salaries int,
                    joining_date datetime,
                    subject char(10))
```

```
create table Bonus(worker_ref_id varchar(10),
                   bonus_date datetime,
                   bonus_amount int)
```

```
create table Title(worker_ref_id varchar(5),
                   worker_title char(15),
                   affected_from datetime)
```

```
insert into Worker values('001','Monika','Arora',100000,'20-Feb-2014 09:00:00','HR')
insert into Worker values('002','Niharika','Verma',80000,'11-Jun-2014 09:00:00','Admin')
insert into Worker values('003','Vishal','Singhal',300000,'20-Feb-2014 09:00:00','HR')
insert into Worker values('004','Amitabh','Singh',500000,'20-Feb-2014 09:00:00','Admin')
insert into Worker values('005','Vivek','Bhati',500000,'11-Jun-2014 09:00:00','Admin')
insert into Worker values('006','Vipul','Diwan',200000,'11-Jun-2014 09:00:00','Account')
insert into Worker values('007','Satish','Kumar',75000,'20-Jan-2014 09:00:00','Create')
insert into Worker values('008','Geetika','Chauhan',90000,'11-Apr-2014 09:00:00','Admin')
```

```
insert into Bonus values('1','20-Feb-2016 00:00:00',5000)
insert into Bonus values('2','11-Jun-2016 00:00:00',3000)
insert into Bonus values('3','20-Feb-2016 00:00:00',4000)
insert into Bonus values('1','20-Feb-2016 00:00:00',4500)
insert into Bonus values('2','11-Jun-2016 00:00:00',3500)
```

```
insert into Title values('1','Acting','20-Feb-2016 00:00:00')
insert into Title values('2','Executive','11-Jun-2016 00:00:00')
insert into Title values('8','Executive','11-Jun-2016 00:00:00')
insert into Title values('5','Manager','11-Jun-2016 00:00:00')
insert into Title values('4','Asst.Boss','11-Jun-2016 00:00:00')
insert into Title values('7','Executive','11-Jun-2016 00:00:00')
insert into Title values('6','Lead','11-Jun-2016 00:00:00')
insert into Title values('3','Leaded','11-Jun-2016 00:00:00')
```

(1) Write an SQL query to fetch “FIRST_NAME” from Worker table using the alias name as <WORKER_NAME>.

```
select first_name as worker_name
from Worker
```

(2) Write an SQL query to fetch “FIRST_NAME” from Worker table in upper case.

```
select upper(first_name)
from Worker
```

(3) Write an SQL query to fetch unique values of DEPARTMENT from Worker table.

```
select distinct subject
from Worker
```

(4) Write an SQL query to print the first three characters of FIRST_NAME from Worker table.

```
select substring(first_name,1,3)
from Worker
```

(5) Write an SQL query to find the position of the alphabet ('a') in the first name column 'Amitabh' from Worker table.

```
select instr(first_name,binary,'a')
from Worker
where first_name='Amitabh'
```

(6) Write an SQL query to print the FIRST_NAME from Worker table after removing white spaces from the right side.

```
select rtrim(first_name)
from Worker
```

(7) Write an SQL query to print the DEPARTMENT from Worker table after removing white spaces from the left side.

```
select ltrim(subject)
from Worker
```

(8) Write an SQL query that fetches the unique values of DEPARTMENT from Worker table and prints its length.

```
select distinct len(subject)
from Worker
```

(9) Write an SQL query to print the FIRST_NAME from Worker table after replacing 'a' with 'A'.

```
select replace(first_name,'a','A')
from Worker
```

(10) Write an SQL query to print the FIRST_NAME and LAST_NAME from Worker table into a single column COMPLETE_NAME. A space char should separate them.

```
select concat(first_name,' ',last_name) as Complete_name
from Worker
```

(11) Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending.

```
select *  
from Worker  
order by first_name
```

(12) Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending and DEPARTMENT Descending.

```
select *  
from Worker  
order by first_name,subject desc
```

(13) Write an SQL query to print details for Workers with the first name as “Vipul” and “Satish” from Worker table.

```
select *  
from Worker  
where first_name in ('Vipul','Satish')
```

(14) Write an SQL query to print details of workers excluding first names, “Vipul” and “Satish” from Worker table.

```
select *  
from Worker  
where first_name not in ('Vipul','Satish')
```

(15) Write an SQL query to print details of Workers with DEPARTMENT name as “Admin”.

```
select *  
from Worker  
where subject='Admin'
```

(16) Write an SQL query to print details of the Workers whose FIRST_NAME contains ‘a’.

```
select *  
from Worker  
where first_name like '%a%'
```

(17) Write an SQL query to print details of the Workers whose FIRST_NAME ends with ‘a’.

```
select *  
from Worker  
where first_name like '%a'
```

(18) Write an SQL query to print details of the Workers whose FIRST_NAME ends with ‘h’ and contains six alphabets.

```
select *  
from Worker
```

where first_name like '____h'

(19) Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.

```
select *  
from Worker  
where salary between 100000 and 500000
```

(20) Write an SQL query to print details of the Workers who have joined in Feb'2014.

```
select *  
from Worker  
where year(joindate)=2014 and  
      month(joindate)='2'
```

(21) Write an SQL query to fetch the count of employees working in the department 'Admin'.

```
select count(*)  
from Worker  
where subject='Admin'
```

(22) Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000.

```
select first_name  
from Worker  
where salary>= 50000 and  
      salary<= 100000
```

(23) Write an SQL query to fetch the no. of workers for each department in the descending order.

```
select count(*),subject  
from Worker  
group by subject  
order by count(*) desc
```

(24) Write an SQL query to print details of the Workers who are also Managers.

```
select *  
from Worker,Title  
where worker_ref_id=worker_ID and  
      worker_title='Manager'
```

(25) Write an SQL query to fetch duplicate records having matching data in some fields of a table.

```
select joindate,salary,count(*)  
from Worker  
group by joindate,salary  
having count(*)>1
```

(26) Write an SQL query to show only odd rows from a table.

```
select *  
from Worker  
where worker_ID%2=1
```

(27) Write an SQL query to show only even rows from a table.

```
select *  
from Worker  
where worker_ID%2=0
```

(28) Write an SQL query to clone a new table from another table.

```
select *  
into Worker1  
from Worker
```

(29) Write an SQL query to fetch intersecting records of two tables.

```
select worker_ID  
from Worker  
intersect  
select worker_ref_id  
from Bonus
```

(30) Write an SQL query to show records from one table that another table does not have.

```
select worker_ref_id  
from Title  
except  
select worker_ref_id  
from Bonus
```

(31) Write an SQL query to show the current date and time.

```
select current_timestamp
```

(32) Write an SQL query to show the top n (say 10) records of a table.

```
select top 5 *  
from Worker
```

(33) Write an SQL query to determine the nth (say n=5) highest salary from a table.

```
select top n *  
from Worker  
order by salary desc
```

(34) Write an SQL query to determine the 5th highest salary without using TOP or limit method.

```
select distinct salary
from Worker
order by salary desc
offset 4 rows
fetch next 1 row only
```

(35) Write an SQL query to fetch the list of employees with the same salary.

```
select w1.first_name
from Worker w1,Worker w2
where w1.worker_ID<>w2.worker_ID and
      w1.salary=w2.salary
```

(36) Write an SQL query to show the second highest salary from a table.

```
select max(salary)
from Worker
where salary<(select max(salary)
              from Worker)
```

(37) Write an SQL query to show one row twice in results from a table.

```
select *
from Worker
union all
select *
from Worker
```

(38) Write an SQL query to fetch intersecting records of two tables.

```
select worker_ID
from Worker
intersect
select worker_ref_id
from Bonus
```

(39) Write an SQL query to fetch the first 50% records from a table.

```
select top 50 percent *
from Worker
```

(40) Write an SQL query to fetch the departments that have less than five people in it.

```
select count(*),subject
from Worker
group by subject
having count(*)<5
```

(41) Write an SQL query to show all departments along with the number of people in there.

```
select count(*) as No_of_Worker,subject
```


from Worker
group by subject

(42) Write an SQL query to show the last record from a table.

select top 1 *
from Worker
order by worker_ID desc

(43) Write an SQL query to fetch the first row of a table.

select top 1 *
from Worker

(44) Write an SQL query to fetch the last five records from a table.

select top 5 *
from Worker
order by worker_ID desc

(45) Write an SQL query to print the name of employees having the highest salary in each department.

select first_name
from Worker
where salary in (select max(salary)
 from Worker
 group by subject)

(46) Write an SQL query to fetch three max salaries from a table.

select top 3 *
from Worker
order by salary desc

(47) Write an SQL query to fetch three min salaries from a table.

select top 3 *
from Worker
order by salary

(48) Write an SQL query to fetch nth max salaries from a table.

select top n *
from Worker
order by salary desc

(49) Write an SQL query to fetch departments along with the total salaries paid for each of them.

select subject,sum(salary)
from Worker

group by subject

(50) Write an SQL query to fetch the names of workers who earn the highest salary.

```
select top 1 max(salary),first_name  
from Worker  
group by first_name  
order by first_name
```