

Database Query Short Note

Kishor Roy
Assistant Programmer
Sonali Bank Limited
EX: Bsc In CSE, CUET

Type-01

Create table:

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

Select from table:

```
SELECT LastName, Address FROM Persons;
```

Insert into table:

```
INSERT INTO accounts(first_name,last_name,phone)  
VALUES ('Trinity','Knox','410-555-0197');
```

Update table Data:

```
UPDATE Customers  
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'  
WHERE CustomerID = 1;
```

SQL query to print details of the Workers whose FIRST NAME contains 'a'.

```
Select * from Worker where FIRST_NAME like '%a%';
```

SQL query to fetch the first 50% records from a table

```
SELECT *  
FROM WORKER  
WHERE WORKER_ID <= (SELECT count(WORKER_ID)/2 from Worker);
```

SQL query to show the last record from a table.

```
Select * from Worker where WORKER_ID = (SELECT max(WORKER_ID) from  
Worker);
```

Type-02

EmployeeInfo Table:

EmpID	EmpFname	EmpLname	Department	Project	Address	DOB	Gender
1	Sanjay	Mehra	HR	P1	Hyderabad (HYD)	01/12/1976	M
2	Ananya	Mishra	Admin	P2	Delhi(DEL)	02/05/1968	F
3	Rohan	Diwan	Account	P3	Mumbai(BOM)	01/01/1980	M
4	Sonia	Kulkarni	HR	P1	Hyderabad (HYD)	02/05/1992	F
5	Ankit	Kapoor	Admin	P2	Delhi(DEL)	03/07/1994	M

EmployeePosition Table:

EmpID	EmpPosition	DateOfJoining	Salary
1	Manager	01/05/2019	500000
2	Executive	02/05/2019	75000
3	Manager	01/05/2019	90000
2	Lead	02/05/2019	85000
1	Executive	01/05/2019	300000

Query to retrieve duplicate records from a table.

```
EmpID, EmpFname, Department COUNT(*)
FROM EmployeeInfo GROUP BY EmpID, EmpFname, Department
HAVING COUNT(*) > 1;
```

Query to find the third-highest salary from the EmpPosition table.

```

SELECT TOP 1 salary
FROM( SELECT TOP 3 salary
FROM employee_table
ORDER BY salary DESC) AS emp ORDER BY salary ASC;

```

Query to find the Nth highest salary from the table without using TOP/limit keyword.

```

SELECT Salary
FROM EmployeePosition E1
WHERE N-1 = (
    SELECT COUNT( DISTINCT ( E2.Salary ) )
    FROM EmployeePosition E2
    WHERE E2.Salary > E1.Salary );

```

Query to retrieve the list of employees working in the same department.

```

Select DISTINCT E.EmpID, E.EmpFname, E.Department
FROM EmployeeInfo E, Employee E1
WHERE E.Department = E1.Department AND E.EmpID != E1.EmpID;

```

Query to fetch top N records.

```

SELECT TOP N * FROM EmployeePosition ORDER BY Salary DESC;

```

Query to retrieve the last 3 records from the EmployeeInfo table.

```

SELECT * FROM EmployeeInfo WHERE
EmpID <=3 UNION SELECT * FROM
(SELECT * FROM EmployeeInfo E ORDER BY E.EmpID DESC)
AS E1 WHERE E1.EmpID <=3;

```

Query to find the maximum, minimum, and average of the employees.

```

SELECT Max(Salary),
Min(Salary),
AVG(Salary)
FROM EmployeePosition;

```

Type-03

Schema

employee(employee-name, street, city)
works(employee-name, company-name, salary)
company(company-name, city)
manages(employee-name, manager-name)

Give an expression in SQL for each of the following queries:

- a) **Find the names, street address, and cities of residence for all employees who work for 'First Bank Corporation' and earn more than \$10,000.**

```
select employee.employee-name, employee.street, employee.city
from employee, works
where employee.employee-name=works.employee-name
and company-name = 'First Bank Corporation' and salary > 10000)
```

- b) **Find the names of all employees in the database who live in the same cities as the companies for which they work.**

```
select e.employee-name
from employee e, works w, company c
where e.employee-name = w.employee-name and e.city = c.city
and w.company-name = c.company-name
```

- c) **Find the names of all employees in the database who live in the same cities and on the same streets as do their managers.**

```
select p.employee-name
from employee p, employee r, manages m
where p.employee-name = m.employee-name and m.manager-name =
r.employee-name
and p.street = r.street and p.city = r.city
```

- d) **Find the names of all employees in the database who do not work for 'First Bank Corporation'. Assume that all people work for exactly one company.**

```
select employee-name from works where company-name <> 'First Bank Corporation'
```

- e) **Find the names of all employees in the database who earn more than every employee of 'Small Bank Corporation'. Assume that all people work for at most one company.**

```
select employee-name from works
where salary > all (select salary from works where company-name =
'Small Bank Corporation')
```

- f) **Assume that the companies may be located in several cities. Find all companies located in every city in which 'Small Bank Corporation' is located.**

```
select s.company-name from company s
where not exists
((select city from company where company-name = 'Small Bank
Corporation')
except (select city from company t where s.company-name = t.company-
name))
```

- g) **Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company.**

```
select employee-name
from works t
where salary > (select avg(salary)
from works s where t.company-name = s.company-name)
```

- h) **Find the name of the company that has the smallest payroll.**

```
select company-name from works group by company-name
having sum(salary) <= all (select sum(salary)
from works group by company-name)
```

Type-04

Table – EmployeeDetails

EmpId	FullName	ManagerId	DateOfJoining	City
121	John Snow	321	01/31/2014	Toronto
321	Walter White	986	01/30/2015	California
421	Kuldeep Rana	876	27/11/2016	New Delhi

Table – EmployeeSalary

EmpId	Project	Salary	Variable
121	P1	8000	500
321	P2	10000	1000
421	P1	12000	0

SQL query to fetch the EmpIds that are present in both the tables – ‘EmployeeDetails’ and ‘EmployeeSalary.’

```
SELECT EmpId FROM
EmployeeDetails
where EmpId IN
(SELECT EmpId FROM EmployeeSalary);
```

Type-05

Sample Table – Worker

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
-----------	------------	-----------	--------	--------------	------------

Sample Table – Bonus

WORKER_REF_ID	BONUS_DATE	BONUS_AMOUNT
---------------	------------	--------------

Sample Table – Title

WORKER_REF_ID	WORKER_TITLE	AFFECTED_FROM
---------------	--------------	---------------

SQL query to fetch the list of employees with the same salary.

```
Select distinct W.WORKER_ID, W.FIRST_NAME, W.Salary
from Worker W, Worker W1
where W.Salary = W1.Salary
and W.WORKER_ID != W1.WORKER_ID;
```

Query to print the name of employees having the highest salary in each department.

```
SELECT t.DEPARTMENT, t.FIRST_NAME, t.Salary
FROM ( SELECT MAX (Salary) AS TotalSalary, DEPARTMENT
      FROM Worker
      GROUP BY DEPARTMENT)
AS TempNew
Inner Join Worker t on TempNew.DEPARTMENT=t.DEPARTMENT
and TempNew.TotalSalary=t.Salary;
```

SQL query to show only odd rows from a table.

```
SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) <> 0;
```

SQL query to show only even rows from a table.

```
SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) = 0;
```

SQL query to show one row twice in results from a table.

```
select FIRST_NAME, DEPARTMENT from worker W where W.DEPARTMENT='HR'
union all
select FIRST_NAME, DEPARTMENT from Worker W1 where W1.DEPARTMENT='HR';
```


SQL query to fetch the first 50% records from a table.

```
SELECT *  
FROM WORKER  
WHERE WORKER_ID <= (SELECT count(WORKER_ID)/2 from Worker);
```

Query for fetching the no. of workers in each department in descending order

```
SELECT DEPARTMENT, count(WORKER_ID) No_Of_Workers  
FROM worker  
GROUP BY DEPARTMENT  
ORDER BY No_Of_Workers DESC;
```

SQL Query for cloning a new table from another table

```
SELECT * INTO WorkerClone FROM Worker;
```

Type-06

SQL Join

Employee Table:

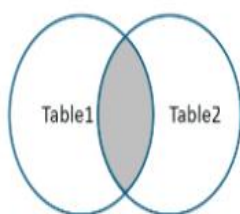
EmpID	EmpFname	EmpLname	Age	EmailID	PhoneNo	Address
1	Vardhan	Kumar	22	vardy@abc.com	9876543210	Delhi
2	Himani	Sharma	32	himani@abc.com	9977554422	Mumbai
3	Aayushi	Shreshth	24	aayushi@abc.com	9977555121	Kolkata
4	Hemanth	Sharma	25	hemanth@abc.com	9876545666	Bengaluru
5	Swatee	Kapoor	26	swatee@abc.com	9544567777	Hyderabad

Project Table:

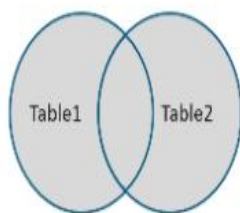
ProjectID	EmpID	ClientID	ProjectName	ProjectStartDate
111	1	3	Project1	2019-04-21
222	2	1	Project2	2019-02-12
333	3	5	Project3	2019-01-10
444	3	2	Project4	2019-04-16
555	5	4	Project5	2019-05-23
666	9	1	Project6	2019-01-12
777	7	2	Project7	2019-07-25
888	8	3	Project8	2019-08-20

Client Table:

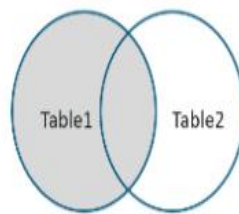
ClientID	ClientFname	ClientLname	Age	ClientEmailID	PhoneNo	Address	EmpID
1	Susan	Smith	30	susan@adn.com	9765411231	Kolkata	3
2	Mois	Ali	27	mois@jsq.com	9876543561	Kolkata	3
3	Soma	Paul	22	soma@wja.com	9966332211	Delhi	1
4	Zainab	Daginawala	40	zainab@qkq.com	9955884422	Hyderabad	5
5	Bhaskar	Reddy	32	bhaskar@xyz.com	9636963269	Mumbai	2



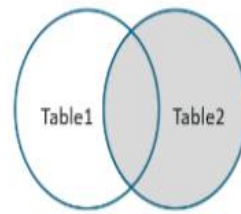
INNER JOIN



FULL JOIN



LEFT JOIN



RIGHT JOIN

Inner Join:

```
SELECT Employee.EmpID, Employee.EmpFname, Employee.EmpLname,  
Projects.ProjectID, Projects.ProjectName  
FROM Employee  
INNER JOIN Projects ON Employee.EmpID=Projects.EmpID;
```

Output:

EmpID	EmpFname	EmpLname	ProjectID	ProjectName
1	Vardhan	Kumar	111	Project1
2	Himani	Sharma	222	Project2
3	Aayushi	Shreshth	333	Project3
3	Aayushi	Shreshth	444	Project4
5	Swatee	Kapoor	555	Project5

Full Join:

```
SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID  
FROM Employee  
FULL JOIN Projects  
ON Employee.EmpID = Projects.EmpID;
```

Output:

EmpFname	EmpLname	ProjectID
Vardhan	Kumar	111
Himani	Sharma	222
Aayushi	Shreshth	333
Aayushi	Shreshth	444
Hemanth	Sharma	NULL
Swatee	Kapoor	555
NULL	NULL	666
NULL	NULL	777
NULL	NULL	888

Left Join:

```
SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID,  
Projects.ProjectName  
FROM Employee  
LEFT JOIN  
ON Employee.EmpID = Projects.EmpID ;
```

Output:

EmpFname	EmpLname	ProjectID	ProjectName
Vardhan	Kumar	111	Project1
Himani	Sharma	222	Project2
Aayushi	Shreshth	333	Project3
Aayushi	Shreshth	444	Project4
Swatee	Kapoor	555	Project5
Hemanth	Sharma	NULL	NULL

Right Join:

```
SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID,  
Projects.ProjectName  
FROM Employee  
RIGHT JOIN  
ON Employee.EmpID = Projects.EmpID;
```

Output:

EmpFname	EmpLname	ProjectID	ProjectName
Vardhan	Kumar	111	Project1
Himani	Sharma	222	Project2
Aayushi	Shreshth	333	Project3
Aayushi	Shreshth	444	Project4
Swatee	Kapoor	555	Project5
NULL	NULL	666	Project6
NULL	NULL	777	Project7
NULL	NULL	888	Project8

Type-07


Query for fetching the details of workers with Salaries ≥ 5000 and ≤ 10000


WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
001	NIHARIKA	ARORA	20000	2013-02-25 09:00:00	HR
002	AYUSHI	GURONDIA	5000	2015-02-10 09:00:00	ADMIN
003	PRIYANSHA	CHOUKSEY	25000	2014-05-16 09:00:00	HR
004	APARNA	DESHPANDE	8000	2016-12-20 09:00:00	ADMIN
005	SHAFALI	JAIN	21000	2015-08-29 09:00:00	ADMIN
006	SUCHITA	JOSHI	20000	2017-02-12 09:00:00	ACCOUNT
007	SHUBHI	MISHRA	15000	2018-03-23 09:00:00	ADMIN
008	DEVYANI	PATIDAR	18000	2014-05-02 09:00:00	ACCOUNT

```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) As Worker_Name, Salary
FROM worker
WHERE WORKER_ID IN
(SELECT WORKER_ID FROM worker
WHERE Salary BETWEEN 5000 AND 10000);
```

Type-08

SQL UNION Examples

CUSTOMER	
Id	
FirstName	
LastName	
City	
Country	
Phone	

SUPPLIER	
Id	
CompanyName	
ContactName	
City	
Country	
Phone	
Fax	

```
SELECT 'Customer' As Type,
       FirstName + ' ' + LastName AS ContactName,
       City, Country, Phone
FROM Customer
UNION
SELECT 'Supplier',
       ContactName, City, Country, Phone
FROM Supplier
```

Type-09

Trigger:

Table:

```
CREATE TABLE "SUPPLIERS"
(
    "SUPPLIER_ID" NUMBER,
    "SUPPLIER_NAME" VARCHAR2(4000),
    "SUPPLIER_ADDRESS" VARCHAR2(4000)
)
```

Before Trigger

```
CREATE OR REPLACE TRIGGER "SUPPLIERS_T1"
BEFORE
insert or update or delete on "SUPPLIERS"
for each row
BEGIN
    Insert into Supplier_Log( SUPPLIER_ID, SUPPLIER_NAME, SUPPLIER_ADDRESS, MODIFY_TIME) VALUES
    (SUPPLIERS.SUPPLIER_ID, SUPPLIERS. SUPPLIER_NAME, SUPPLIERS. SUPPLIER_ADDRESS, SYSDATE)
END;
/
ALTER TRIGGER "SUPPLIERS_T1" ENABLE
/
```

After Trigger

```
CREATE OR REPLACE TRIGGER "SUPPLIERS_T2"
AFTER
insert or update or delete on "SUPPLIERS"
for each row
begin
    Insert into Supplier_Log( SUPPLIER_ID, SUPPLIER_NAME, SUPPLIER_ADDRESS, MODIFY_TIME) VALUES
    (SUPPLIERS.SUPPLIER_ID, SUPPLIERS. SUPPLIER_NAME, SUPPLIERS. SUPPLIER_ADDRESS, SYSDATE)
end;
/
ALTER TRIGGER "SUPPLIERS_T2" ENABLE
/
```

Function:

```
create or replace function adder(n1 in number, n2 in number)
return number
is
    n3 number(8);
begin
    n3 :=n1+n2;
    return n3;
end;
```

/

Sequence:

```
CREATE SEQUENCE id_seq  
  INCREMENT BY 10  
  START WITH 10  
  MINVALUE 10  
  MAXVALUE 100  
  CYCLE  
  CACHE 20;
```

PL/SQL program to retrieve emp table and then display the salary

```
declare  
v_sal number(10);  
begin select max(sal) into v_sal;  
from emp;  
dbms_output.put_line(v_sal);  
end;  
/
```