SQL COMMANDS

(1) Select data commands :-

i) Select Data : All
select * from companyData

ii) Select Data : One or few columns
select name, location, salary from companyData

iii) Select Data : Column Alias
select name, location, salary as "INCOME" from companyData

iv) Select Data : where keyword
select * from companyData where salary > 15000
select * from companyData where id > 2 and id <= 4

v) Select Data : Between keyword
select * from companyData where salary between 10000 and 15000

(2) Delete data commands :-

1) Delete command —

```
delete from CompanyData where id = 3
```

11) Truncate command —

```
truncate table CompanyData
```

111) Drop command —

```
drop table CompanyData
```

(3) Create/Insert/Update :-

1) Create table in Database —

```
create table testtable
(
 Id number(5),
name varchar(20),
salary number(6),
location varchar(20)
);
```

11) Insert data into table —

```
insert into TestTable
values(5, 'John', 5000, 'NY');
```

111) Update data into table —

```
update TestTable set name = 'kyle'
where id = 5
```

* DDL - (Data Definition Language) :-

(I) CREATE
(II) ALTER
(III) DROP

} CAD

* DML - (Data Manipulation Language) :-

(I) SELECT
(II) INSERT
(III) UPDATE
(IV) DELETE

} SIUD

* DCL (Data Control Language) :-

i) GRANT — Gives a priviledge to user.
ii) REVOKE — Takes back priviledge granted from user.

* Distinct :-

select distinct paycode from empinfo

* Count :-

select count(*) from customers

* AND :-

select * from customers where city = 'Paris' AND postalcode = '75012'

* OR :-

select * from customers where city = 'Paris' AND postalcode = '75012' or postalcode = '75016'

* desc :-

select * from customers order by companyname desc

* asc :-

select * from customers order by city asc

* like :-

select * from customers where city like 'Lon%'

** % → A substitute for zero or more characters

** _ → A substitute for a single character

** [charlist] → Sets and ranges of characters to match

* in :-

```
select * from customers where
city in ('Stranbourg', 'Marseille')
```

* between :-

```
select * from customers where
customerID between 'ALFKI' and
'BERGS'
```

* not between :-

```
select * from customers where
customerID not between 'ALFKI'
and 'BERGS'
```

* Select Into :-

```
select * into neworders from
orders where 1=0
```

* Insert Into Select :-

```
insert into neworder(employeeID,
OrderDate)
select embaloyeeID, orderDate from
orders
```

APPIUM INTRODUCTION

* Select Top :-

```
select top 2 * from customers
```

* Aliases :-

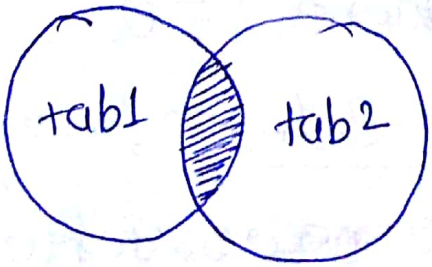
```
select customername as customer,
contactname as [Contact Person]
from customers
```

* SQL Join :-

1. INNER JOIN (Simple Join)
2. LEFT JOIN (LEFT OUTER JOIN)
3. RIGHT JOIN
4. FULL JOIN

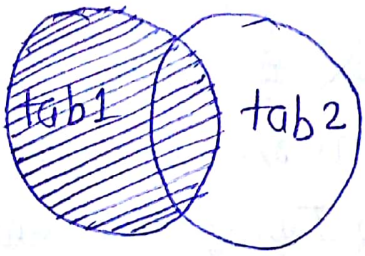
---

1. INNER JOIN :-



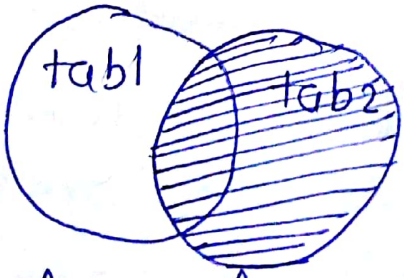
```
select customers.customername, orders.orderID from customers
INNER JOIN orders
ON customers.customerID = orders.customerID
ORDER BY customers.customername
```

② LEFT JOIN :-




```
SELECT customers.customername,
orders.orderID from customers
LEFT JOIN orders
ON customers.customerID = Orders.CustomerID
ORDER BY Customers.CustomerName
```

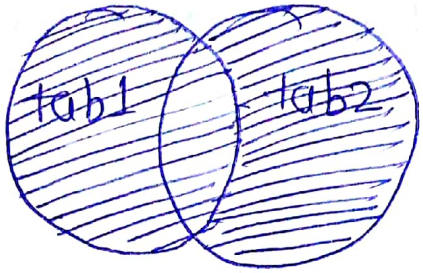
③ RIGHT JOIN :-




```
select orders.orderID, employees.firstname
from orders
RIGHT JOIN employees
```

ON orders.empID = employees.employee-eID
order by orders.orderID

④ FULL JOIN :-



Select customers.customerName,
orders.orderID from customers
FULL OUTER JOIN orders
ON customers.customer-ID = Orders.
customerID
ORDER BY customers.customername

Important Questions

* Select max(salary) from EmployeeNew

* For top 5 emp having max sal

```
Select FirstName, LastName, Salary
from EmployeeNew E1
where 5 > (select count(*) from
EmployeeNew E2 where E2.salary >
E1.salary) order by salary desc
```

* Assume a schema of Emp(Id, Name, DeptId), Dept(Id, Name)

  10 records → Emp table
  5 records → Dept table

  select * from Emp, Dept

  Result → 50 rows

  As a cartesian product or cross join, which is the default whenever the 'where' clause is omitted.

*

Id	Name	Sex	Salary
1	A	M	25000
2	B	F	15000
3	C	M	10000
4	D	F	14000

Swap all F and M values with a single update query and no intermediate temp table.

Update salaries set sex = CASE sex WHEN 'M' THEN 'F' Else 'M' END

↓ remaining will 'M'

* test_a
10
20
30
40
50

test_b
10
30
50

Query to fetch values in table ~~that~~ test_a that are and not in test_b without using the NOT keyword.

Select * from test_a
except
select * from test_b

** In oracle, the minus keyword is used instead.

* Table tbl —
1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1

Query to add 2 where number is 0 and add 3 where number is 1

update TBL set number = case when number > 0 then number + 3 else number + 2 end

* select all the even number records —

select * from table where id % 2 = 0

select all the odd number records —

select * from table where id % 2 != 0

* Triggers :- Triggers are stored programs/ procedure, which are automatically executed or fired when some events occur.

Triggers are, in fact, written to be executed in response to any of the following events —

1) A DML — (DELETE, INSERT or DROP)
2) A DDL — (CREATE, ALTER or DROP)
3) A database operation — (SERVER —

ERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN)

** Triggers can be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers —

i) Generating some derived column values automatically

ii) Enforcing referential integrity

iii) Event logging and storing information on table access.

iv) Auditing

v) Synchronous replication of tables

vi) Imposing security authorizations

vii) Preventing invalid transactions.

* Describe DML and DDL :—

DDL :— DDL is short name of Data Definition Language which deals with database schemas and

descriptions, of how the data should reside in the database.

I) CREATE — To create database and its objects like(table, index, views, store procedure, functions and triggers)

II) ALTER — Alters the structure of the existing database

III) DROP — Delete objects from the database

IV) TRUNCATE — Remove all records from a table, including all spaces allocated for the records are removed.

V) COMMENT — Add comments to the data dictionary.

VI) RENAME — Rename an object

DML — DML is short name of Data Manipulation Language which deals with data manipulation, and includes most common SQL statements such as SELECT, INSERT, UPDATE, DELETE etc., and it is used to store, modify, retrieve, delete and update data in database.

I) SELECT — Retrieve data from a database

II) INSERT — Insert data into a table

III) UPDATE — Update existing data within a table

IV) DELETE — Delete all records from a database table.

* Select highest top 5 salary of the employee from the table —

```
select top 5 salary, first, last
FROM employeeneee
ORDER BY salary Desc
```

* Find second highest salary of the employee —

```
Select max(salary)
FROM employeenew
where salary<(select max(salary)
from employeenew)
```