SQL   Sequel

Full form - Structured Query Language

Data - Varible (Temparary)
File (Permanent)

Text, Audio, Video, Database [[Oracle XE], MySQL, MS - SQL Server]

Table -

Sr.No	Name	Address
1	A	Sagar
2	B	indore

```
SQL > Connect
Enter user name :   System.
Enter Password :
Connected.
```

```
SQL > Create user [User name] identiyied by [Password.];
User Created.
```

Connect this user.

```
SQL> grant connect,resources to [User name];
Grant Succeeded.
SQL> connect   usorname / pass.;
Connected.
```

```
SQL > Create table [table name] (rollno. number(3),
                                 name  varchar(10));
   table created.
```

Data input in table.

```
SQL > insert into tablename values(101 , 'veer');
  1 row Createed.
```

Variables in Sql.

int ---→	number (size)	like	number(4);
Char ---→	Varchar(size)	like	Varchar(10);
Date ---→	date		

Show table.

SQL > Select * from tablename;

result show

According to ~~requirted~~ show.

SQL > Select Sr.No, name, Address. from tablename; ——→ etc.

According to name wise Search -

SQL > Select ~~Sr~~ * from tablename order by name; and order by Salary, address S.No. date desc etc.

SQL > Select * from tablename order by name desc; —Same Syllb

Show one name

SQL> ~~Show~~ Select * from tablename where name='veer'; and Salary, address, S.No. date etc. —Same Syntax

Condition wise check

SQL > Select * from tablename where Salary < 3000; and Salary > 3000 . <=, <> date != Same Syntax

null data Search.

SQL > Select * from tablename where Salary is null; - and name null, date null Address null etc isnull

Same to not null

SQL > Select * from tablename where Salary is not null; → —Same Syntax and operator.

SQL > Select * from tablename where address ~~name~~ = 'sagar' and Salary>60000 and etc
and like name and date

Or Operator

SQL > Select * from tablename where name = "veer" or address = 'sagar'; Or
and name salary etc.

1. Primary key.

a) No duplicates.
b) Not null
c) Max. 1 per table
d) Used to link a table with another.

2. Not null.

a) not null.

3. Check.

a) Conditional.

check row and Column Command.

```
SQL > desc tablename ;
```

Data permanent saved Command.

```
Sql > Commit ;
  Commit Complete.
```

data update Command.

```
SQL > update tablename set Salary = 20000 where name= 'veer' ;
```

and name, address, date etc updated. Same Syntax.

add row and Column, Command.

~~to~~ another variable used.

```
SQL > alter table tablename add (gender char) ;
```

drop row and Column Command.

```
SQL > alter table tablename drop column gender ;
```

---> etc. drop.

Modify row and Column. Command.

```
SQL> alter table tablename modify (name varchar(20)) ;
```

——> etc modify.

Delete Command.

SQL > delete from student where rollno. = 101 ; ⟶ etc. information.
tablename
row deleted.

SQL > delete from tablename ; → all row deleted.
all row deleted.

SQL > truncate table tablename ;
table truncated.

SQL > drop.table tablename.
table dropped.

SQL >
create table ~~net~~ account (acno. number(10) Primary key, name varchar(10) not null,
address varchar(10) not null, balance number(5)
check(balance >1000));

uses of Primary, notnull, check.

SQL > Create table employee (empcd number(3) primary key, name varchar(10) notnull,
Salary number(10) check (salary >15000),
depart varchar(12) check (depar in
('IT', 'HR'));

Connect to or more table command.

references
Keyword used for connected.

```
SQL> Create table Stock(pno number(3) primary key, pname varchar(10),
     company varchar(10));
```

after inserting.

PNO	pname	Company
1	Moniter	Dell
2.	Mouse	Logitech
3.	Keyboard	HP

```
SQL> Create table Sales(billno number(3) primary key, cname varchar(10) not null,
     pno number(3) references stock);
```

└→ table connected.

Conditions of connected.

1) Data type same like. number --→ number etc.

2) Same size of Data types like. $\underbrace{\text{number(10)}}$ ---→ $\underbrace{\text{number(10)}}$

3) references keyword used for connected.

```
SQL> insert into Sales values(101, 'veer', 2):
```

└→ Range of Parent primary key

if out of range parent key then error Primary key not found.

Joins

joins are used to query 2 or more tables using a single select statement.

1. Equi / Simple / Inner join - An equi displays the data from 2 or more tables, that is common in all the tables.

example - stock, Sales ~~tabb~~ tables.

SQL> Select stock.pno, pname, company, billno, cname from stock, sales where stock.pno = sales.pno;

Stock Sales.
P.NO = PNO

PNO	PNAME	COMPANY	BILLNO	CNAME
2	Mouse	Logitech	101	Veer
4	CPU	Intel	102	Hari

2. out join

i) Left out join - Displays all the data from the left / master tab irrespective of whether there is a corresponding entry in the right / child table.

Syntax like pno = pno(+);

example, Stack, Sales tables ;

```
SQL > Select stock.pno, pname, company, billno, cname from Stock,Sales where Stock.pno = Sales.pno(+);
```

PNO	PNAME	COMPANY	BILLNO.	CNAME
1.	Moniter	Dell		
2.	Mouse	Logitech	101	Veer
3.	Keyboard	HP		
4.	CPU	Intel	102	Hani

ii) Right Outer join :- Displays all the data from the child table irrespective of whether there is a corresponding entry in the parent table.

A right outer join should always return the same result as inner join. This is because it is not logically correct that the child table contains some records that are not parent in the parent tables.

Syntax = like pno(+) = pno ;

```
SQL > Select Stock.pno , pname, company, billno, cname from Stock. Sales where Stock.pno(+)
      = Sales.pno ;
```

PNO	PNAME	COMPANY	BILLNO	CNAME
2.	Mouse	Dell	101	Veer
4.	Keyboard	HP	102	Hani

Same result as inner join
Sample join.