

SQL

Sequel

Full form - Structured query Language

Data - Variable (Temporary)
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 identified by Password;

User Created.

Connect this user.

SQL > grant connect, resources to user name;

Grant Succeeded.

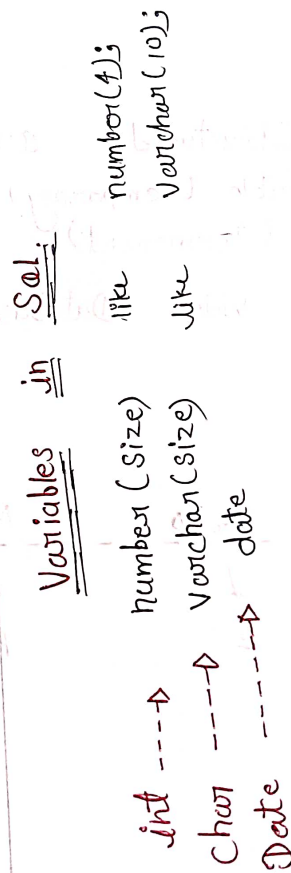
SQL > connect username / pass.;

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 created.



Show table.

SQL > Select * from tablename;

result show

According to requirement show.

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

According to name size search -

SQL > Select * from tablename order by name; and order by Salary, Address

SQL > Select * from tablename order by name desc; S.No. date desc etc. Same Syntax

Show one name

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

Condition size 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 Same to not null Address null etc is null

SQL > Select * from tablename where Salary is not null; —————> Same Syntax

and operator.
SQL > Select * from tablename where address = 'Sagar' and Salary > 60000 and like name ^ date etc

or Operator
SQL > Select * from tablename where name = 'veer' or address = 'Sagar'; 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.

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.
row deleted. tablename

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 account (acno. number(10) Primary key, name varchar(10) not null, address varchar(10) not null, balance number(5) null, check (balance > 1000));
uses of Primary, notnull, check.

SQL > create table employee (empid number(3) primary key, name varchar(10) not null, salary number(10) check (salary > 15000), department varchar(12) check (department in ('IT', 'HR')));

Connect to our more table command.

References

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

Keyword used
for connected.

after inserting.

pno	pname	Company
1	Monitor	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);

conditions of connected.

↳ table connected.

- 1) Data type same like. number → number etc.
- 2) Same size of data types like. number(10) → 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 tables.

SOL > Select stock.pno, pname, company, billno, cname from Stock, Sales where stock.pno = sales.pno;

	<u>PNO</u>	<u>PNAME</u>	<u>COMPANY</u>	<u>BILLNO</u>	<u>CNAME</u>
Stock P.No = P.No	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 table irrespective of whether there is a corresponding entry in the right / child table.

Syntax like pno = pno(+);

example, stock, sales tables ;

SAL > Select stock.pno, pname, company, billno, cname from stock, sales where stock.pno = sales.pno(+) ;

PNO	PNAME	COMPANY	BILLNO.	CNAME
1.	Monitor	Dell		
2.	Mouse	Logitech	101	Veer
3.	Keyboard	HP		
4.	cpu	Intel	102	Hari

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 present in the parent tables.

Syntax = like $pno(+) = pno$;

SAL > 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	Hari

Same result as inner join sample join.