1.fe:
html css js ...

2.be:
lang:
php
java +jsp,servlets,jdbc +spring boot
python + django or flask or cherry py ....
....
3.db:
sql:oracle ,mysql,ms sql server .....
nosql:mongodb

port : mysql 3306
---

sql

Data: data is nothing but its a collection of information
ex:books login.

Database: Its collection of organized data.
ex:
bank database:
customer,transctions,loans,emi,....etc.

Database Management System(DBMS):
It is the collection of serveral types of data,with data maintaintainance logic and data accessing logic.
1.foxpro
2.dbase
3.msaccess

Relational Database Management System(RDBMS):
Relationals are established among the data.

Object Relational Database Management System(ORDBMS):
oops + ordbms

current database s/w: oracle,mysql
RDBMS software:
There are 2 types.
1.single user or file database.
2.multi user or server database.

Single user dbms:
this data accessed by one user at a time through it is placed in server.

ex:
microsoft excel
microsoft access

Multi user dbms:
data can be manipulated by more than one user at a time concurrent access is possible.
ex:

1.sybase sql server
2.oracle sql server
3.ms sql server
4.mysql server

SQL:
->It is not a case sensitive language.
->It is a user friendly language.
->It is also known as sequel.
->This language is not meant for programming.
->It is only lang used for db,from last 30 years.
->It user databases objects and db commands and in its the syntax.

Working with oracle sql:
1.Install oracle with oracle installer file.
(oracle 10g expression edition)

note:oracle runs on multi user environment.
there 2 types of users
1.DBA
2.Normal User.

->connect as dba
sql>connect system
enter password: ****
(or)
sql>connect system/orcl;

->create a normal user.
syntax: create user username identified by password;
 ex: sql>create user nk identified by nk;

->grant privilages.
syntax: grant connect,resource to username;
ex: sql>grant connect,resource to nk;

Note: A normal user can also be a dba.

convert dba to username:
syntax: grant dba to username;
sql>grant dba to nk;

->connect to oracle server from browser:
http://127.0.0.1:8080/apex

SQL Syntax:
There are 2 types of sql syntax
1.sql commands
2.sql objects

SQL Commands:
There are 4 types of sql commands.
1.DDL
2.DML

3.TCL
4.DCL

DDL: Data Definition Languagee:
     1.create
     2.drop
     3.alter
DML: Data Manipulation Language:
     1.insert
     2.delete
     3.update
TCL: Transaction Control Language:
      1.commit
      2.rollback
DCL: Data Control Language:
     1.grant
     2.revoke

Database Objects:
sql users following database objects
1.Tables
2.View
3.Synonym
4.Sequence

Table:
->It is the collection of rows(records) and columns(field)
->excel worksheet
->this is also known as entity.
->table = structure(ddl)+rows(dml)

creation of table:
create:
     it is used to create a table.
ex:
syntax: create table table_name(variable datatype(),var2 dt(),var3
dt(),...);
sql> create table dept102(dept_no number(2),dept_name
varchar2(5),location varchar2(5));

Display the structure of the table:
syntax: desc table_name;   desc or describe table_name;
ex: desc dept102;

Insert:
      this is used to insert the a record into a table.
ex:
syntax: insert into table_name values(va1,val2,'val3string',...);
sql>insert into dept102 values(1,'maths','vij');

note:use single codes(' ') to insert/represent nonnumeric data.

commit:
      to submit or save the work.

Display all inserted records:
select:
      to retrieve the record from table.
ex:
syntax: select * from table_name;
sql>select * from dept102;

**Display only dept name of the dept102 **
syntax: select col from tablename;
sql>select dept_name from dept102;

**Display few column of the dept102 **
syntax: select col1,col2 from tablename;
ex:select dept_name,location from dept102;

**Display all column of the dept102 **
syntax: select * from tablename;
sql>select * from dept102;

DataTypes in sql:
sql supports following in datatypes.
1.char : it allows alphabates(a-z)(0-9),sepcial characters.
         it offers fixed length (255).

2.varchar2: it is same as char varied in length.
            The max length is 2kb

3.long: it hold huge char data upto 2gb.

4.LOB's:Large Objects
        1.CLOB: char large object - uptp 4gb

5.number: it holds -ve and +ve ,int,floating numbers
ex:
a number(4)=1000
b number(7,2)=12345.12

6.date : it hold date and time values.
oracle sql users its date format as dd-mon-yy.

7. blob: binary large object (it holds unlimited binary data).

Build oracle sql based demo tables
[demobld.sql]->search in google.
copy emp table and dept table and past into your work area(user)

SQL> desc emp;
SQL> desc dept;

Query processing in sql:

->select command is known as query.
->query always returs the results.

There are 2 types of query processings
1.conditional query process
2.unconditional query process

unconditional query process
The process of reteriving all the records from database table.

ex:
Q:display all emp details.
SQL> select * from emp;

Q:display ename and their job details.
SQL> select ename,job from emp;

conditional query process
The process of retriving few records from database table.
This query should be return with select statement with where clause.

usage:
select ..... where codition;

where : it is a clause used to impose condition
ex:
Q:display all details of empono 7369
SQL> select * from emp where empno=7369;

Q:display all details of clerks
SQL> select * from emp where job='CLERK';

Q:display name of the manager
SQL> select ename from emp where job='MANAGER';

Operators:
1.arithmetic ops
2.relational ops
3.logical ops
4.special ops

Arithmetic ops:
+,-,*,/

dual table usage
dual is a system defined table that contains 1 record
sql>select 10+30 from dual;

40

***allocate 20% hra to every employee and display ename,sal,hra ***
SQL> select ename,sal,sal*.2 from emp;
SQL> select ename,sal,sal*.2 hra from emp;
Note: hra is alias name of sal*.2

Relational / comparision ops:
<,>,<=,>=,==,!=(or)<>

1.display emp details working in 20th dept
sql>select * from emp where deptno=20;
2.display enames and jobs of emps working in 20th dept
sql>select ename,job from emp where deptno=20;
3.display emp details working as clerk
sql>select * from emp where job='CLERK';

Logical operators:
these are used to compare more than one condition

1.and
2.or

and: it results if both the conditions are satisfied.

c1	c2	result
1	1	1
1	0	0
0	1	0
0	0	0

ex:
1.display emp details working in 10th dept and drawing sal above 2500
sql>slect * from emp where deptno=10th and sal>2500;

2.display emp details drawing sal between 2000 to 3000
sql>select * from emp where sal>=2000 and sal<=3000;

3.display all emps working as ANALYST and drawing sal above 1000
SQL> select * from emp where job='ANALYST' and sal>=1000;

OR:
It is a logical operator
it result if one condition is satisfied

1.display emps working as salemans, clerk
SQL> select * from emp where job='SALESMAN' or job='CLERK';

2.display all emps working 10th dept,30th dept
SQL> select * from emp where deptno=10 or deptno=30;

special ops:
1.in,not in
2.between,not between
3.like,not like
4.is null, isnot null

1.in,not in:
these are used to compare for group of values
ex:
SQL> select * from emp where deptno in (10,30);

```
SQL> select * from emp where job in ('CLERK','MANAGER');
SQL> select * from emp where job NOT in ('CLERK','MANAGER');

2.between,not between:
this is used to test for range
usage:
SQL> select * from emp where sal between 2000 and 3000;
SQL> select * from emp where sal not between 2000 and 3000;

get all emp details working in 10,30 depts and working as clerk,manager
SQL> select * from emp where deptno in (10,30) and job in
('CLERK','MANAGER');

3.like,not like:
these are used to compare for pattern formed with wild card chard such -
,%
'-' ->It replaces multiple character.
'%' ->It replaces single char.

Q:Display enames whose names starts with 'S'
SQL> select * from emp where ename like 'S%';

Q:Display enames whose names ends with 'S'
SQL> select * from emp where ename like '%S';

Q:Display enames whose names contains 2nd char as 'A'
SQL> select * from emp where ename like 'A%';

DML COMMANDS:
1.Insert :This is used to  insert one record into a table.

Q.Insert into all column of table.
SQL> insert into dept102 values(11,'css','viz');

Q.Insert into few columns of table
SQL> insert into dept102(dept_name,location) values ('js','tn');

2.Delete : This is used to delete the records.
a)unconditional deletion.
b)conditional deletion.

a)unconditional deletion:
The process of deleting all the records from a table.
ex:Delete from table name.
clear data from screen -> [cl scr]

b)conditional deletion:The process of deleting one or few records from
table.
ex:Delete from table_name where colname = value;
delete from dept102 where dept_no='2';

3.Update:
This is to edit or modify the tables data.
a)unconditional  updation.
```

b)conditional updation.

a)unconditional updation:
Q: Increase all emps sal by 100 dallars.
SQL> update emp set sal=sal+100;

b)conditional updation:
the process of updating one / few records of a table.
Q:promote 7788 empno as manager.
sql>update emp set job='MANAGER' where empno=7788;

DDL commands:
CREATE-SELECT:
This used to create a new tables based on enitity strucure/table data.
->create a table emp3 as select * from emp;->structure rows
create table emp4 as select * from emp where deptno=**[table condition]
->structure,without rows.

iNSERT-SELECT:
->Insert into emp5(select * from emp);

Truncate:
->This is to delete all the records of a table unconditional
->It is a perminent deletion.

Delete	Truncate
->few rows,all	all
->temperary deletion	perminent deletion.

Types of Query's:
1.simple query
2.complex query

Simple Query:
1.select-where
2.select-distinct,order by
3.select-functions

Complex Query:
1.Sub Query
2.corelated sub query
  ----contraints---
3.joins
4.set operators.

Functions:
This is to retrive table data in various passion.
types of functions.
1.single row function.
2.group function.

Single Row Functin:
This function returns a value for every record.

types of single row functions:
1.char/str
2.numeric
3.date
4.conversion
5.millaneous ->(only oracle sql)

char function:
SQL> select upper(ename) from emp;
SQL> select upper('kumar') from dual;

dual:
It is a system defined table. That contains only one row.

SQL> select lower('HELLO') from dual;
SQL> select initcap('hello') from dual;
SQL> select length('welcome') from dual;

Q:display every emp name along with name length?
SQL> select ename,length(ename) from emp;

Q:display all emp name whose names contains 6 char length?
SQL> select ename from emp where length(ename)=6;

SQL> select chr(65) from dual;
chr() -> It returns char equavalence of given ascii values
SQL> select ascii('A') from dual;

Instr(str,search pattern)
Instr(str,patstr):It returns pastr string position as >='1' if found other whose returns '0'

SQL> select instr('java','v') from dual;
SQL> select instr('python','a') from dual;

Substr(str,start,end):
It returns a partial string b/w start & ending position.
SQL> select substr('welcome',1,4) from dual;

Soundex() function:
It returns similar string based on pronounciation.
SQL> SELECT ENAME FROM EMP WHERE SOUNDEX(ENAME)=SOUNDEX('SMITH');

trim():
It returns the string by ignoring,unwanted blank spaces.
SQL> select length(trim('nitish')) from dual;

Lpad():this is used to decorate a string with pad of chars at left side of a string.
SQL> select lpad('sql',8,'*') from dual;

Rpad():same as above but right side of the string.
SQL> select rpad('sql',8,'*') from dual;

Numeric function:

```
SQL> select mod(10,7) from dual;

SQL> select sqrt(25) from dual;

SQL> select abs(-25) from dual;

SQL> select power(2,3) from dual;

SQL> select log(40,10) from dual;
```

round(),trunc():
There functions used to reduce the floating with required no.of decimals.

```
SQL> select round(3.14912,2) from dual;
SQL> select trunc(3.14912,2) from dual;
```

Ceil(),Floor():
It returns integers requivalent by taking floating number as of argument.

```
SQL> select ceil(3.01) from dual;
SQL> select floor(3.99) from dual;
```

Date():
It returns current date in oracle date formate.

```
SQL> select sysdate from dual;

SQL> select add_months(sysdate,2) maturilydate from dual;

SQL> select months_between(sysdate,'31/08/2000')/12 from dual;

SQL> select NEXT_DAY(sysdate,'MONDAY') from dual;

SQL> select LAST_DAY(sysdate) from dual;
```

Conversion functions:
1.to_char()
2.to_date()
3.to_number()

to_char():
it is used to convert given data to char or number to char.

```
SQL> select length(to_char(1243)) from dual;
```

to_char(date,formate):
It returns date in particular formate.

```
SQL> select to_char(sysdate,'dd') from dual;
```

mm->month in number
mon->abbr name of month
month->full name of month
yy->2 digit year
yyyy->4 digit year
year ->year in words

```
ddd->nth day from jan 1st
hh->hours
mi->minutes
ss->seconds
day->full name of weekday
dy->short name of weekday
w->week number
ww->week number from jan 1st.

Q:display current time.
SQL> select to_char(sysdate,'hh:mi:ss') from dual;
Q:display emps joined on monday.
SQL> select ename from emp where to_char(hiredate,'dy')='mon';

Q:display emps joined in 'jan' month.
->SQL> select ename from emp where to_char(hiredate,'mon')='jan';

to_date():
This to convert char formate date  to oracle sql date.
SQL> select to_date('08/10/22','dd/mm/yy') from dual;

to_number():This is to convert char date into numeric formate.
SQL> select to_number('123')from dual;

Mislaneous functions:
user:
SQL> select user from dual;
user -> it returns work area name.

rownum:
SQL> select rownum,ename from emp;

2.Group Function/Aggregate Functions:
They return a value for group of rows.
1.sum()
2.min()
3.max()
4.avg()
5.count()

SQL> select sum(sal) from emp;
SQL> select min(sal) from emp;
SQL> select max(sal) from emp;
SQL> select avg(sal) from emp;
SQL> select count(sal) from emp;
SQL> select sum(comm) from emp;
It returns know value of sound by excluding null values.

Q:Display how many managers are working
SQL> select count(*) from emp where job='MANAGER';
--------------------------
Clauses:
1.where
2.unique/distinct
```

3.order by
4.group by
5. having

Unique/Distinct:
it is used to refer values for once though they repeated in table.
SQL> select unique job from emp;
SQL> select count(unique job) from emp;
SQL> select distinct job from emp;
SQL> select count(distinct job) from emp;

Order by:
This is to retrieve data either in a asc order or dsc order.
SQL> select * from emp order by ename;

Q: display all emp name in desc order
SQL> select * from emp order by ename desc;

Q: display all emp name in asc order
SQL> select * from emp order by ename asc;

Q: display all 10th dept employee details sorted order of their sal.
SQL> select * from emp where deptno=10 order by sal;

SQL> select unique sal from emp where deptno=20 order by sal;

order by clause with more than one column
sorting on 2nd column is done by only on repatitive rows.
SQL> select ename,sal from emp order by sal,ename;

Group by:
This consider entire table into logical groups.
Aggregate functions can be get from logical, groups.

->sum,max,min,count,avg.

Q:display emp count in department wise?
SQL> select deptno,count(*) from emp group by deptno;

Note:groupped column & aggregate function retrievel is possible.

Q:diplay avg sal of all jobs?
SQL> select job,avg(sal) from emp group by job;

Q:diplay highest sal of each dept expt 20th ,display all dept in asc?
SQL> select deptno,max(sal) from emp where deptno=20 group by deptno
order by deptno;

Having:
It is a conditional clauses used to test the condition after grouping
the rows

Q:display depts in which >3 emps are working?
SQL> select deptno,count(*) from emp group by deptno having count(*)>3;

Q:display all depts where max(sal)>2000 except manager salary in asc order of deptno?
SQL> select deptno,max(unique sal) from emp where job!='MANAGER' group by deptno having max(sal)>2000 order by deptno;

Complex Query:
  Sub Query:
       A query with in another query one query is known as parent/outer query and
       another one is known as inner /sub query.

need
to retrieve complex results.

Usage:
select .... where ....(select ......);
select .... where .... ->parent/outer query
(select ......);   ->sub/inner query

Note:
->First sub query is executed independently.
->parent query is executed based on the results given by sub query.

Q:diplay emp name who is getting highest sal?
SQL> select ename from emp where sal=(select max(sal) from emp);

Q:display emp name who is getting min sal?
SQL> select ename from emp where sal=(select min(sal)from emp);

Q:display all emp details drawing ,sal < avg(sal) of the company
SQL> select * from emp where sal<(select avg(sal) from emp);

Q:display 2nd max sal?
SQL> select max(sal) from emp where sal< (select max(sal) from emp);

***sub query that returns more than one value ***
Q:display emp drawing avg(sal) of deptno 10 & 20 departments
SQL> select ename from emp where sal in (select avg(sal) from emp where deptno in (10,20));

nested sub query
A sub query with in another sub query one query is known as inner most query the other query is known as inner query and another query is known as outer query.

syntax:
select ... (select ...(select ...));

Q:display ename drawing 2nd highest sal?
SQL> select ename from emp where sal=(select max(sal)from emp where sal<(select max(sal) from emp));

Q:list the emp who are senior to king?
SQL> select * from emp where hiredate<(select max(hiredate) from emp where ename ='KING');

SQL> select * from emp where hiredate<(select max(hiredate) from emp where ename ='king');

Co-related sub query:
simple sub-query is independent where as co-related sub query is dependent on parent query table.

Q:display emp details drawing sal<avg(sal) from their working departments.
SQL> select ename,deptno,sal from emp e where sal<(select avg(sal) from emp where deptno=e.deptno);

Q:display emp details drawing sal max(sal) in their working departments.
SQL> select deptno,ename,sal from emp e where sal=(select max(sal) from emp where deptno=e.deptno);

Data Integrity:
The process of maintaining a rule based data. That is data to be maintained based ib some business rules.
ex:
product Id should be unic
gender to be Male or Female

Contrains:
These are used to impose business rules of to implement data integrity.
There are 3 types of contrains:
1.Domine Integrate constrains
2.Entity Integrity constrains
3.Referential Integrity constrains

Domain Integrity constrains:
1.Not null
2.check

Not Null Constrains:
This constrain doesn't allow null values.
ex:impose not null constraint.
create table product (pid number(3) not null,pname varchar2(30));
validate constraint
Insert into product (700,'pen');-->No-violation
insert into product (null,'pencil');-->violation

Check constraint:
This is to impose business rule on a column.
create a table for book data maintainance with following contraints.
do not allow null value for bname column.
bid should be with in of 1000 to 2000.

```
create table books (bid number(4) check(bit between 1000 and 2000 ),bname
varchar2(30) not null);

insert into books values(2610,'the way');->no voilation.
insert into books (2611,'the way');->voilation of check.
insert into books values(2610,Null);->voilation of not null.

Q:create table for bank customer allow acctype as c or s.
create table customer (accno number(4) not null,acctype char(1)
check(acctype in ('c','s')));

->test for voilation:
Insert into customer values(1234,'s');

**Entity Integrity Constraints**
These constrains consider entaires table data for voilation check.
These constrains used to avoid data redendence(data duplication).
1.Unique
2.Primary Key

1.Unique:
This constrain doesn't allow duplicates
ex:
create table product(pro number(4) unique);
insert into product values(100); -> not-voilation
insert into product values(100); -> voilation.
Note:
The draw back of unique is it allows null values.

2.primary key:
It is doesn't allow null vaules and duplicate values.
primary key:unique+not null
It combines the both features of both unique and not null constraints.
the limitation of an primary key is for an entire table only one primary
key is allowed.
ex:
create table consumer (cno numbers(4) primary key,name varchar2(8));
insert into consumer values(1221,'aaa'); -> not voilated
insert into consumer values(1221,'bbb'); -> voilated
insert into consumer values(null,'aaa'); -> voilated

***Types of Imposing constraints on columns***
1.impose the constrains at column level without name.
2.impose the constrains at column level with name.
3.impose the constrains at table level without name.
4.impose the constrains at table level with name.
5.impose the constrains using alter table.

Referential Integrity constraint:
Inorder to impose relations amoung the tables

books:        member:
bid             mid
bname           mname
```

price address

Issues:
bid
mid
idate

These are used to established relation among the table thses relations is known as parent-child relation shop or master-details relationship.

Contraints:
1.primary key.
2.reference key.
3.foreigon key.

In order to establish a relation two tables are required
1.master table
2.detail table
Note: a key/coloumn is required to impose the relation.

steps in:
1.create a master & impose key column with primary key.
2.insert few records in master table.
3.create a detail & impose key column with reference key.
4.insert few records in detail table and test for voilation.

ex:
create master-detail relationship for gas agency table.
1.consumer.
2.booking.

consumer:
create table consumer (cons_id number(4) primary key,conname varchar(10));
insert into consumer values(1212,'smith');
insert into consumer values(1213,'allen');

Detail table:-
create table bookings(cons_id number(4) references consumer(cons_id),bdate date);
insert into bookings values(1212,sysdate); ->normal
insert into bookings values(121,sysdate); ->voilated

Constraint name:
every constraint is imposed with a name.This name is used for further purpose
1.to remove the constraint
2.disable/enable the constraint

**Impose the constraints at column level with name **
->create table books(bid numberr(4) constraint pk1 primary key ,bname varchar2(30) constraint nn not null);

**table level definition **

->to impose more than one constraint to a column
->this is to imposite primary key and unique key.

Table level definition:
constraints imposed at the end of the table is known as TLD.
EX:
create table for product details, allow product Id with primary key and pid should be > 8999.
->impose pk with col level
->impose check const with table level with col name.
**table level definition with out name **
create table product (pid number(4) primary key,pname varchar2(30) not null,constraint ch1 check(pid)>3999);

->constraint pk1 primary key(id);
->constraint un1 unique(nid);

Note: not null constraint should be impose only at column level.

composite primary keey/composite unique:
If primary|unique key imposed on more than one column.
ex:
one emp is eligible for vehloan and houloan.
7788 veh_loan
7788 housing_loan
7788 veh_loan ->voilated.

Note:
these contraints are imposed only at a table level.
->create table emp_loans(eid number(2),ename varchar2(10) not null,loan_name varchar2(20),constraint cm1 primary key(eid,loan_name));

->Inserrt into emp_loans(11,'mia','housing');
->Inserrt into emp_loans(11,'mia','vechicle');
->Inserrt into emp_loans(11,'mia','housing');-> voilated.

Alter table usage:
This is used to add the constrains to disable the constraints and to enable the constraints.

***Adding the constraints ***
->Not nulls
alter table emp_loans modify(loanname varchar2(20) not null);
->pk|unique|check:
alter table student add primary key(regno);
alter table student add constraint un1 unique(name);
alter table student add constraint ck1 check(regno>10);

->reference key:
create table customer(cid number(4) primary key);
create table trans(cid number(4));
alter table trans add constraint rel1 foreign key(cid) reference customer(cid);

Remove/drop constraint
alter table trans drop constraint ch1;

***disble the constraint ***
alter table trans disable constraint rel1;

***enable the constraint ***
alter table trans enable constraint rel1;

JOINS:
Joins:
this is usedd to retrieve comm data spreadedd over among table more than one table.
these are 3 types of joins.
1.Inner join
2.outer join
3.self join

Inner Join:
It is used to get common data spreded over among the tables.This should be imposed with join condition.
->select * from emp,dept;
It is results in cartition products.
->Inner joint to interduced to avoid carticial product.
->It should be imposed with join condition where as join condition is imposed with common column of both thee table.

Q:Display emp name along with their working dept namee.
->select e.ename,d.dname from emp e,dept d where e.deptno=d.deptno;

subq:Q: display revenue spent on sales dept.
->select sum(sum) from emp where deptno=(select deptno from dept where dnamee='sales');

Q:display emps name working as a managers in location newyork.
->select e.ename,d.deptno,d.dname,e.job from emp e,dept d where e.deptno=d.deptno and e.job='MANAGER' and d.loc='NEW YORK';

Q:display the highest paid emp location
->select d.loc from dept d,emp e where d.deptno= e.deptno and sal=(select max(sal) from emp);

Q:display all the emps details along with their working dept details
->select e.*,d.* from emp e,dept d where e.deptno=d.deptno;
->select ename,d.* from emp e,dept d where e.deptno=d.deptno;

Outer Join:
It is an extension to inner join.
it retriven comm and uncommon data from one table.

left outer join:
select ..... from table1 t1, table2 t2 where t1.id(+)=t2.id;

right outer join:
select ..... from table1 t1, table2 t2 where t1.id=t2.id(+);

note:
comm and uncomm data will be retrieven from oppostie table for (+) symbol.

Q:display all dept details along with emp details working in those depts.
->select e.*,d.* from emp e,dept d where e.deptno(+)=d.deptno;

Self Join:
Its a joins impose with in a single tableby considering logical as two tables.
select ..... from emp.e1,emp.e2 from emp e1,emp e2 where e1.col1=e2.col2;

select e.empno,m.ename from emp e,emp m where e.mgr= m.empno;

Set Operators:
These are known as relationaloperators used to access relational data from more than one table.
These are similar to simple mathematical sets:
1.Union all
2.Union
3.Intersect
4.Minus.

create table tab1(no number(3));
insert into tab1 values (10);
insert into tab1 values (20);

create table tab2(no number(3));
insert into tab2 values (10);
insert into tab2 values (20);
insert into tab2 values (70);

Union all:
select no from tab1 union all select no from tab2;

Union:
select no from tab1 union select no from tab2;
union same as union all but dosen't include duplicate.

Intersect:
It retrieves common data.

minus:
It retrieves disting | uncommon data of tab1.

Database Objects| sql Objects:
1.View
2.Synonym

3.Sequence

View:
It is imaginary table,offers window access to a table

need of views:
This is to control privilages on granted objects.

```
create view v1 as select * from tab1;
```

note:
all DML manipulations are allowed on view.

read only views:
It offers only select usage.

```
create view v2 as select * from emp readonly;
```

conditional views:
It offers condition access

```
create view v3 as select * from v1 where no !=50;

drop view v3; [drop]
```

Synonym:
It offers additional name to a table and provides user friendly usage.

```
create synonym e for emp;
select * from e;
drop synonym e;
```

Sequence: It only supports in oracle sql.

```
create sequence s1; [default options]
create sequence s2 start with 1000 increment by 1 max value 1005;
```

usage of the sequence:
pseudo columns is used to get values from sequence.

```
select s1.nextval from dual;
select s1.currval from dual;
```

----------the end----------