

Structural Query Language

what is data?

- * Data is nothing but a useful information

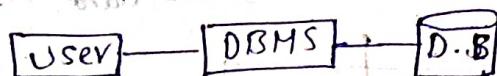
Data base:

It is a centralised location where we can store large amount of data.

Database management system :- (DBMS)

→ DBMS is a software which is responsible for all type of operations like fetching, removing, modifying and updating.

→ It acts as an Intermediator between the user and database.



→ DBMS helps the user to communicate with the Database.

SQL :-

→ Structured Query language is used to communicate with the Database.

→ We write Queries to communicate with the database in SQL.

Types of DBMS:-

1. RDBMS — Relational database management system

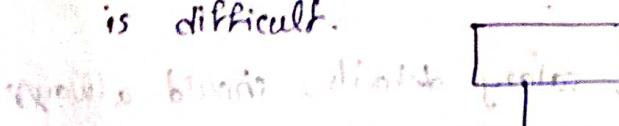
2. NDBMS — Network database management system.

3. HDBMS — Hierarchical DBMS

HDBMS :- It stores data in tree form.

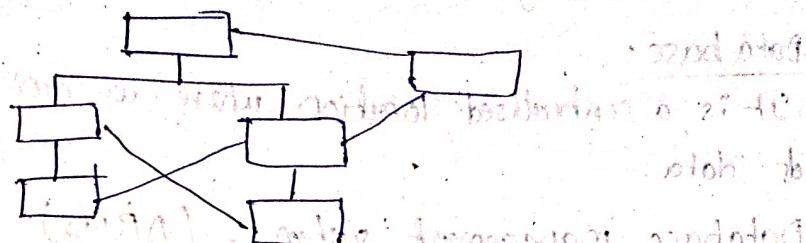
→ In HDBMS, the data will stored in the form of tree structure.

→ Here inserting the data is easy, but retrieving / fetching is difficult.



2. NDBMS :-

- In NDBMS, the data will be stored in the form of networks.
- Each data will be interconnected to each other.



→ Inserting the data is difficult and retrieving the data will be easy.

1. RDBMS :-

- In RDBMS, the data will be stored in the form of tables.
- In RDBMS, Inserting and retrieving the data is easy.

1	2	3	4	5

Objectives of DBMS :-

1. Mass storage
 2. Data protection
 3. Data redundancy
 4. Platform independent
 5. Improve Data sharing.
- Mass storage :-
- DBMS allows to store thousands of data into the database.
 - One can fetch the required data whenever needed.
2. Data protection Redundancy
- DBMS guarantees that there will be no data duplicity among all the records.
3. Data protection :-
- Data such as bank details, salary details, should always be secured.
 - DBMS provides a master level security for the data that will be stored in the database.

4. Platform independent :-

→ One can run DBMS on any platform.

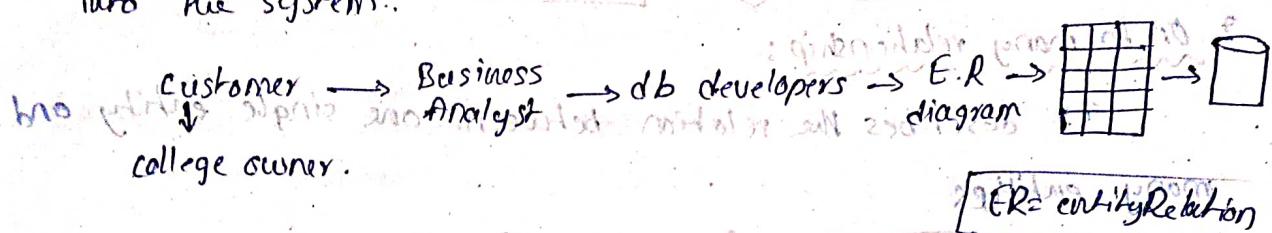
→ No particular platform is required to work on DBMS.

5. Improved Data sharing :-

→ DBMS will share the requested data more quickly, efficiently and secured.

Data models :-

Data models defines how the data is processed and stored into the system.



Entity Relationship diagram(ER) :-

In ER diagram, we should specify three components, they are 1. Entity 2. Attributes and 3. Relationships.

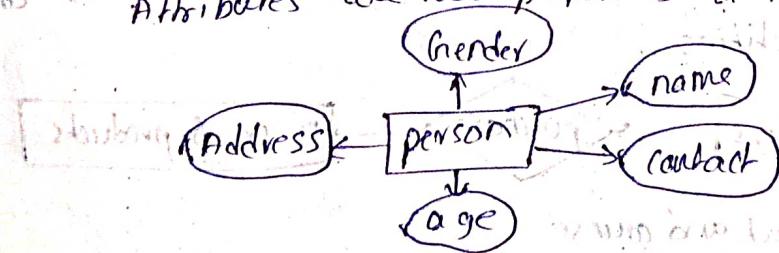
1. Entity :-

Entity is nothing but a living or non-living thing.

e.g. person, House

2. Attributes :-

Attributes are the properties of the entity.



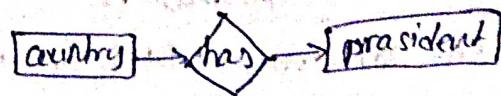
3. Relationship :-

Relationship is the connection or Association between two entities.



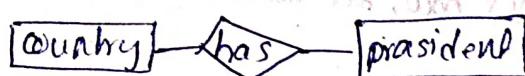
Types of Relationships:

1. one to one Relationship
2. one to many Relationship
3. many to one Relationship
4. many to many Relationship.



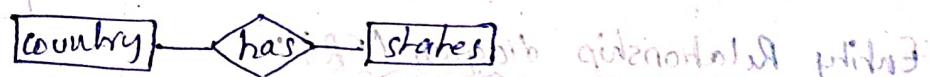
1. One to one Relationship :-

It describes the Relationship between one single entity and another single entity.



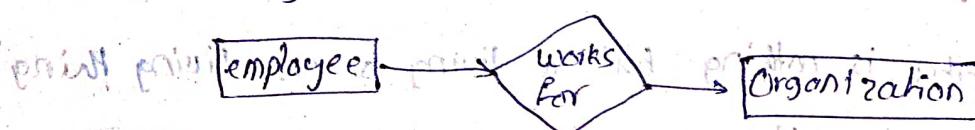
2. One to many relationship :-

It describes the relation between one single entity and many entities.



3. many to one relationship:-

It describes the relationship between many entities and one single entity.

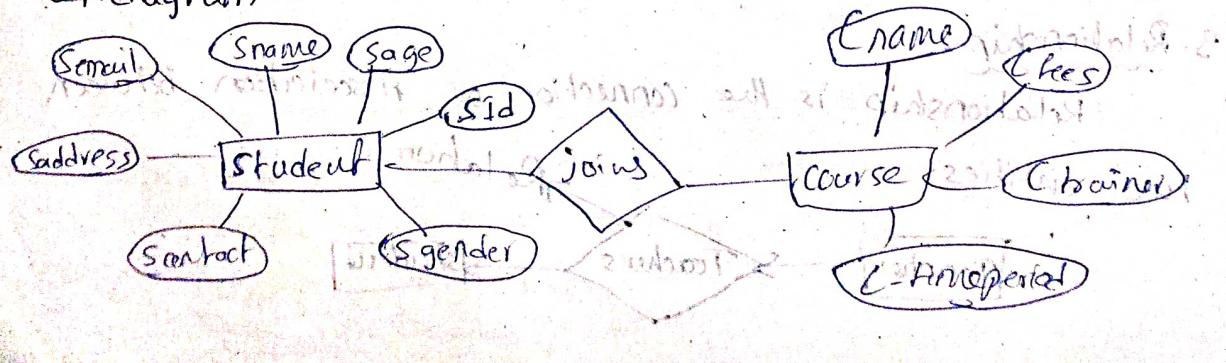


4. many to many Relationship:-

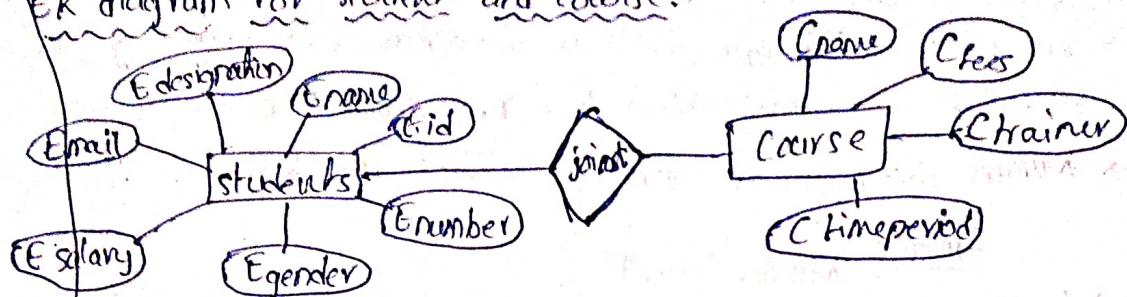
It describes the relationship between many entities and another n no. of entities.



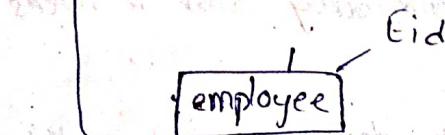
ER diagram for student and course.



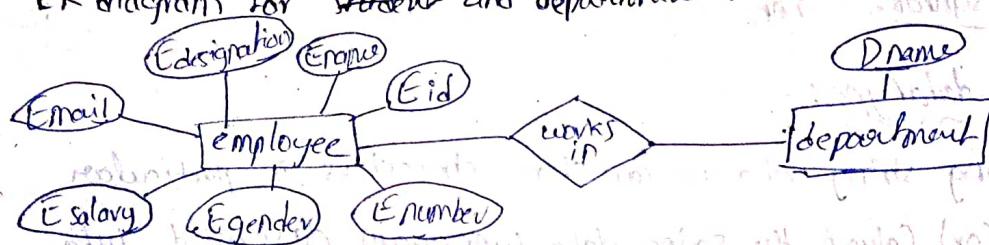
ER diagram for student and course:



ER diagram for employee and department:



ER diagram for student and department.



Data types:-

→ Data types defines what kind of data can be stored in each column of a table.

→ There are 3 main datatypes of SQL are as follows.

1. Number datatype (or) numeric datatype.

2. String datatype (or) capital datatype.

3. Date datatype.

1. Numeric datatype :-

→ The numeric datatype allows to store integers, floating points, positive number, negative number.

→ The numeric datatype established into 2 types

i). Number datatype.

ii). int datatype.

i) Number data type:-

- In number data type we can store the data according to the size but we cannot store the more than size.
- Number data type is used for control the data.

Syntax: Number(size)

ii) int data type:-

- int data type we no need to the size if suppose we want to store 4 digit, int datatype will automatically create 4 blocks of memory.
- There is a drawback in int datatype doesn't control on this size.

Syntax: int

2) String datatype:-

- by using string data we can store characters in particular purpose (or) column the string data type again classified into

3 types (i) character database datatype

(ii) Varchar database datatype

(iii) Varchar2 datatype

(i) Char data type:-

In that data type the left of memory will be wasted.

Syntax: char(size)

e.g: CHAR(5) ⇒

--	--	--	--	--

It will create 5 characters

(ii) Varchar data type:-

In varchar data type the left of blocks also we can use.

Syntax: varchar(size)

e.g: varchar(5)

It will create the 5 blocks.

iii) varchar2 data type:-
 → varchar and var char 2 both are same but varchar occurs 2000 bytes memory and varchar2 occupies 4000 bytes of memory

Syntax: `varchar2 (size)`

3) Date data type:-
 → Date data type is no need to give size.
 → The default date format of SQL is date, month, year.

Syntax: `date`

e.g. "DD-MM-YYYY"
 else "07-02-2001"

but SQL takes 07-Feb-2001 Automatically.

Sub languages of SQL

There are 5 types:-

1. DQL → Data Query Language (select) (Select, Insert, Update, Delete)
2. DDL → Data Definition language (Create, Alter, Drop, Truncate) (Create, Alter, Drop, Truncate)
3. DML → Data manipulation language (Insert, Update, Delete) (Insert, Update, Delete)
4. TCL → Transaction Controlling Language (Commit, save point, Rollback) (Commit, save point, Rollback)
5. DCL → Data controlling language. (Grant, Revoke) (Grant, Revoke)

- Data Query Language:-
 → Data query language is used to fetch data from the database
 → In DAL we use "select" command, this select command is used to inform the data base which data we are going to fetch.
 → From clause is used to inform the database from which table we are going to fetch the data.

• Syntax :- `select column name from Tablename;`

- ① Write a query to fetch employee name, salary, designation and commission

Select ENAME, SAL, JOB, COMM from EMP;

- ② Write a query to fetch department name, and department location from department table.

Select DNAME, LOC from DEPT;

- ③ A query to fetch all the data of employee table

Select * From EMP

"*" is an operator which is used to get all the data of the table.

Query :- Query is nothing but a request.

⇒ To display what are the tables present in the database

Select * From TAB;

tab → It is a keyword defines table

where clause :-

where clause is used to fetch particular records from the table

Syntax:

Select CN From TN where condition = 'condition';

- ① Write a query to fetch all the details of employee smith.

Select * From emp where ename = 'SMITH';

- ② Write a query to fetch salary and joindate of employee scott.

Select SAL, HIREDATE From EMP where ENAME = 'SCOTT';

- ③ Display all the details of research department.

Select * From DEPT where DNAME = 'RESEARCH';

Q) Display the department number of sales department.

Select DEPTNO from DEPT where DNAME = 'SALES'.

Note: show pagesize ; If page size 200
show linesize ; If page lines 200

→ set pages 200 lines 200

For clear the screen.

CL SCR

Information will receive of how many rows are

available & what information, what order, what type of

information is required & what is the row identifier.

All employees will have their details of personal details @

as follows:

[9131088 002-162 31MAY1982] (a)

employees will have their personal details will be given as

[9131082 012-162 10C 3 MAY 1982]

which will include employee number, name, date of birth, gender,

etc. etc.

[9131082 002-162 10C 3 MAY 1982]

and so on all other employees will have

[9131082 002-162 10C 3 MAY 1982]

information about their details of personal details.

With employees do similarly

so all employees will have their details of personal details.

[9131082 002-162 10C 3 MAY 1982]

Operators :-

→ An operator is a reserved word or character that performs some specific operations.

→ operators are classified into 6 types

1. Arithmetic operator
2. Relational operator
3. logical operator
4. String operator / Character operator
5. special operator
6. set operator

1. Arithmetic operator:-

An operator which is used to perform the mathematical operations like addition, subtraction, multiplication & division is known as Arithmetic operator.

Q. Write a query to fetch salary for all the employees with increment of 300

A) SELECT ENAME, SAL+300 FROM EMP;

Q. Display the annual salary for all the employees

SELECT SAL*12 FROM EMP;

Q. display employee name, designation, salary with deduction of 300 for all the employees.

SELECT NAME, JOB, SAL-300 FROM EMP;

Q. display all the employee salary with hike of 30%.

SELECT SAL+(SAL*30/100) FROM EMP;

Q. write a query to fetch employee name, annual salary & commission of employee smith.

Select SNAME, SAL*12, COMM From EMP where

Operators :-

→ An operator is a reserved word or character that performs some specific operations.

→ operators are classified into 6 types

1. Arithmetic operator
2. Relational operator
3. Logical operator
4. String operator / character operator
5. Special operator
6. Set operator

1. Arithmetic operator:-

An operator which is used to perform the mathematical operations like addition, subtraction, multiplication & division is known as Arithmetic operator.

① Write a query to fetch salary for all the employees with increment of 300

A) SELECT ENAME, SAL+300 FROM EMP;

② Display the annual salary for all the employees

SELECT SAL*12 FROM EMP;

③ display employee name, designation, salary with deduction of 300 for all the employees.

SELECT NAME, JOB, SAL-300 FROM EMP;

④ display all the employee salary with hike of 30%.

SELECT SAL+(SAL*30/100) FROM EMP;

⑤ write a query to fetch employee name annual salary & comission of employee smith.

Select SAL,ENAME, SAL*12, COMM From EMP where
ENAME='smith' OR ENAME='SMITH';

- ⑥ Write a query to fetch employee name, designation, salary with hike of 50% & commission with increment of 300 For all the employee

Select ENAME, JOB, SAL + (SAL * 50 / 100), COMM + 300 from EMP;

- ⑦ Display name, employee hire date, designation, commission with deduction of 200 of employee scott

SELECT ENAME, HIREDATE, JOB, COMM - 200 from EMP;

where ENAME = 'SCOTT'

Relational operators: Relational operators are used to compare two values or two expressions. So that relational operators is also called as comparison operators.

, $=$, $>$, \geq , $=$, \neq (\leq)

- ① display all employee details who is getting salary more than 2000

Select * from EMP where SAL > 2000;

- ② list all the employee details who is getting salary atleast 1000

Select * from EMP where SAL \geq 1000;

- ③ display all the employee details who is getting exactly 3000 as salary

Select * from EMP where SAL = 3000;

- ④ write a query to fetch all employee details except those who are getting 3000 as salary.

Select * from EMP where SAL \neq 3000;

Select * from EMP where SAL $<>$ 3000;

⑤ Write a query to fetch all employee details who is getting commission less than 500. To implement this condition we can use

AND select * from EMP where COMMISSION < 500;

⑥ Display all the details of employee Allen

select * from EMP where ENAME = 'ALLEN';

Literals:-

→ Literal is nothing but a fixed data which might be available inside the table or from outside the table.

→ We have 3 types of literal

1. Number literal
2. String literal
3. Date literal

Note:

Used literals:- Oracle stores everything in numbers. So we can use literals to get specific format of output

(i) Number literal

e.g. select 620 from EMP;

(ii) String literal

e.g. select 'Good morning' From EMP;

(iii)

e.g. select '26-Nov' From EMP;

Dual

→ Dual is a dummy table which is automatically created by oracle database.

→ It consists of only one column and only one data/value

Use of Dual

* The use of dual table is to get the output only once.

Concatenation operator:

→ we have only one operator i.e., concatenation operator, || .

→ concatenation operator is used to combine the columns & literals.

Q) Write a query to combine employee names with respective designation.

select ENAME || ' ' || JOB From EMP;

Q) write a query to treat every employee as follows.

e.g. Good Morning Smith.

select 'Good Morning' || ENAME From EMP;

Q) write a query to display as follows.

e.g. Smith is working as a clerk and earning salary as 800.

select ENAME || ' is working as ' || DB || ' and earning ' || SAL From EMP;

Q) write a query to combine department name & location.

select DNAME || ' ' || LOC From DEPT;

Q) write a query to display the output as, smith is working department

selected ' smith is working under dept 20 ' From Dual;

Q) write a query to display the output as, scott is working as

an Analyst and joined the company on 19th April 87 and getting a salary of 3000, and working under dept 20.

select ' ENAME || ' is working as an ' || job || ' and joined the company on ' || hiredate || ' getting a salary of ' || SAL || ' and working under dept ' || DEPTNO , From EMP;

LOGICAL operators

Logical operators will fetch the data based on multiple conditions checking.

Types of logical operators are as follows:

1. AND operator

2. OR operator

3. NOT operator

AND: It will display the o/p when both the conditions are True / satisfied.

OR: If any one of the condition is True then OR operator will display the o/p.

NOT: It will fetch the data if none of the condition are True.

Q1) A query to fetch all the details of employee smith who is working in dept 20

Ans: Select * from EMP where ENAME = 'SMITH' AND DEPTNO = 20;

Q2) Write a query to fetch all the employee details except martin

Select * from EMP where ENAME != 'MARTIN';

Q3) Display all the dept information except sales dept.

Select * from dept where dname != 'sales';

Q4) Write a query to fetch all the details of employee who is not working as a salesman.

Select * from EMP where JOB != 'salesman';

Q5) A query to fetch all the details of employee smith and scott

Select * from EMP where ENAME = 'SMITH' OR ENAME = 'SCOTT'

- ⑥ Write a query to fetch all the employee details of 'JONES' as well as fetch of dept=10

Select * from EMP where ENAME = 'JONES' or DEPT NO = 10;

- ⑦ write a query to fetch all the details of employee ADAMS who is getting salary of 1100

Select * from EMP where ENAME = 'ADAMS' AND SAL = 1100;

- ⑧ query to fetch all the dept information except who are working in newyork location.

Select * from DEPT where LOC != 'newyork';

Special operators

Special operators are used to perform some specific operation
1. Between, 2. IS NULL, 3. IN, 4. Like

• Like: like operator is used to search a specified pattern. In like operator two wild characters: (%) (_) – underscore

Syntax: select Col name from Table name where condition like pattern;

- ① A query to fetch all the details of employee whose name is starting with 'S'

Select * from EMP where ENAME like 'S%';

- ② A query to fetch employee name, salary & commission of employee whose designation starts with 'C'

Select ENAME, SAL, COMM where JOB like 'C%';

③ Write a query to fetch job & designation & hire date of employees whose employee name starting with 'T' and 3rd character is 'R'.

Select * from EMP where ENAME like 'T_R%';

④ Write a query to fetch all the department information of employees whose location is starting with 'D' and ends with 'S'.

Select * from DEPT where LOC like 'D%_S';

⑤ Display the dept name of EMP whose location is ending with 'N'.

Select DNAME from DEPT where LOC like '%_N';

Syntax for like operator:

Select * from EMP where condition like pattern;

→ % is used to we don't know how many characters are there after given character.

→ _ is used to we know how many characters are there b/w the given characters.

Between operator

→ It is used to fetch the data while finding ranges.

Syntax:

Select * from Tablename where condition between high range and low range (or) vice versa;

① Display the details of employees who is getting salary in the range of 1000 & 3000.

Select * from emp where sal between 1000 and 3000;

② List all the employee details who is getting commission in the range of 500 and 1000.

Select * from emp where comm between 500 and 1000;

Q) Display the employee details who joins the company from 22-feb-81 to 23-may-87?

Select * from emp where Hiredate between '22-feb-81' and '23-may-87';

IS NULL's

→ IS NULL operator is used to fetch the null values.

Syntax: SELECT CN from TN where Condition is null;

① Write a query to fetch all the details of employee who is not getting commission?

Select * from emp where Comm is null;

② Write a query to fetch all the details of employee who are getting commission?

Select * from emp where Comm is not null;

③ Write a query to fetch the details of employee who don't have report managers

Select * from emp where MGR is null;

In Operator:

→ IN operator is used to fetch multiple data from the same column:

Syntax: Select CN from TN where Condition IN (values);

① Display all the details of employee smith, scott and martin?

Select * from emp where Ename IN ('smith', 'scott', 'martin');

② Display the details of employees who is working in 10 and 20

* Select * from emp where DeptNo IN (10, 20);

③ List all the employee details who is working as Salesman, Clerk, and Analyst

Select * from emp where Job IN ('Salesman', 'Clerk', 'Analyst');

④ Display the department information of accounting and research department

Select * from Dept where Dname IN ('Accounting', 'Research');

⑤ Display the department information of employee who is working in Chicago and Newyork?

Select * from Dept where Loc IN ('Chicago', 'Newyork');

As operator:- (or) Aliasing
[It is used to change the name of column & table]

* As operators is used to change the name of column of Table

* Here specifying of 'as' is optional

* Syntax:

Select column as Aliasing name from Table Name.

Q1 Write a query to change the name of the column from job to designation

Select JOB as designation from EMP;

Q2 Write a query to change the name of column from sal to salary

Select SAL as salary from EMP;

Q3 Write a query to change multiple columns

Select empname, job, desi from EMP;

Setoperator:
Set operators are used to perform operations like union, unionall, intersect, and minus

Syntax: (Select column from Table name set operator select column from Table name)

Note:

Set operator misguidedly use two select queries but it is used to combine two queries to perform them together

Ex:-

(1) Write a query to-

Display all employees whose job is not sales & not marketing

(Employee 'job' has option 'sales' & 'marketing')

① Union :- Union operator will fetch the unique values by removing the duplicates

select deptno from emp union select deptno from dept;

② Union all :- Union all operator will fetch all the data including the duplicates

select deptno from emp unionall select deptno from dept;

③ Intersection :- It is used to fetch the common values

select deptno from emp intersect select deptno from dept;

④ Minus :- Minus operator will remove the common values and it will fetch unique value from the first column.

select deptno from emp minus select deptno from dept;

o/p

no rows selected

select deptno from dept minus select deptno from emp;

o/p

deptno

Constraints :- These are the rules to maintain the integrity of the data.

* Constraints are nothing but a rules ~~but they~~ that we provide for a Table while creating.

* It is used to limit the type of data can be entered into the column of the table.

* The constraints are as follows

1. Unique Constraints
2. Not null
3. Default
4. Check
5. Primary Key
6. Foreign Key

1) Unique

- * It ensures all the values in a column are different.
- * It won't allow the duplicates.

2) Not null constraints

- * This constraint ensures that there will be ~~no~~ no null values of column of a Table.

3) Default Constraints

- * It sets a default value if no value is specified.

4) Check constraints

- * It allows the data into the column if the specified condition satisfied.

5) Primary key constraints

- * primary key constraint is used to identify each row in a table uniquely.
- * It is a combination of not null & unique.
- * For one table we can have only one primary key column.

6) Foreign key constraints

- * It is used to connect two tables.

- * For one table we can have multiple foreign key columns.

- * The foreign key columns of child table will take the reference of primary key columns of parent table and it is both the tables.

- * The foreign key column can contain both duplicates and null values.

DDL (Data Definition language)

- It is used to create tables, performing alterations to the created table and used to delete that tables
- The DDL commands are as follows

1. Create

2. Alter

3. Drop

4. Truncate

5. Rename

Create :-

This command is used to create our own tables in the database.

Syntax :-

```
create table Tablename (CN, datatype constraints, CR ---);
```

①

```
create table product (PID number(5) primary key, PName varchar(10)  
not null, price number(4));
```

②

Create table person name, age, gender

```
create table person (PID number(6) primarykey,
```

```
PName varchar(10) primarykey not null,
```

age number(5) not null,

gender varchar(10) not null);

Alter :-

Alter command is used to add, rename, drop and modify the columns in the created table.

① Alter add :-

Alter add is used to add new columns to the existing table.

Syntax

```
Alter table <table> add col_name (datatype,constraints)
```

② Alter table product add Exp-Date Date;

Alter Rename:

It is used to change name of the column.

Syntax:

`Alter Table TN Rename old col name to new col name;`

Ex:-

`Alter Table product Rename column price to cost;`

Alter drop:

It is used to delete particular columns for the created table.

Syntax

`Alter Table TN drop column (CN);`

Ex:-

`Alter Table product drop Exp-Date;`

Alter modify:

This command is used to change data type of a column at the same time we can change the size also.

SyntaxNote:

(we can change the data type of a column only when the column is empty)

Syntax

`Alter Table TN modify column datatype(size);`

Ex:-

`Alter Table product modify PNAME number(20);`

DROP :-

Drop is used to delete table from the Database.

Syntax

`Drop Table table name;`

Ex:-

`Drop Table product;`

Show Recycle bin:-

It is used to display whether the deleted table is present in the recycle bin or not.

Syntax:

Show recyclebin;

Flashback :-

It is used to restore the deleted table.

Syntax:

Flashback Table TN to before drop;

OR

Flashback Table product to before Drop;

Purge:-

It is used to delete the table permanently from the database.

Syntax:

purge Table TN;

or

purge Table products;

PERMANENT Recycle bin:-

It will delete multiple tables at a time in the recycle bin.

Syntax:

purge Recycle bin;

TRUNCATE:-

It is used to delete all the records from the table.

Syntax:

Truncate Table TN;

Foreign key:

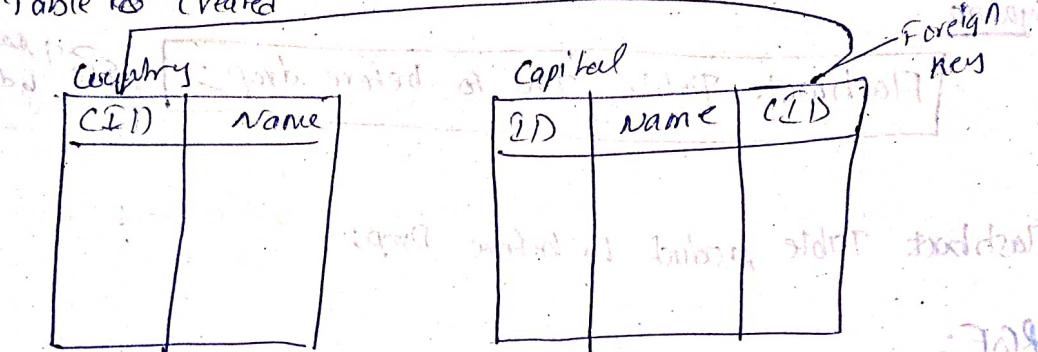
It is used to connect tables.

create table country (cid number(5) primarykey, name varchar(10) not null);

Table created.

create table capital (id number(8) primarykey, name varchar(10), reference country (cid));

Table is created.



Data Manipulation language :- (DML)

* DML commands are generally used to make changes on the database and it is responsible for all types of modification.

* DML commands are as follows.

1. Insert

2. Update

3. Delete

1) Insert:

Insert commands are used to insert values into the created Table.

Syntax:

insert into TN values(v₁, v₂, v₃ ...);

(or)

insert into TN (col1, col2, ...) values(v₁, v₂, ...);

Note:

All the operation performed by DML code temporarily if we want to make it permanent we have to use a TCL command commit (transaction control) statement.

Update:

This command will update values in the table and for updation where condition is compulsory.

Syntax:

update TN set (N₁ = V₁, N₂ = V₂ -- where condition)

Ex-

select * from capital;

update capital set name = 'kathmandu' where ID = 102;

① update scott salary to 50000

select * from EMP;

update EMP set SAL = '50000' where ENAME = 'SCOTT';

② Update designation of dept ID 10 to sales man

update EMP set JOB = 'salesman' where deptno = 10;

③ Update the salary to 20000 and commission to 10000 for employee Allen.

Update EMP set SAL = 20000, COMM = 10000 where ENAME = 'ALLEN'

Delete :-

- * It is used to delete particular records from the table and where condition is compulsory for deletion.
- * If we don't specify where condition then the whole table will be affected.

Syntax

Delete from TN where condition;

ex:

Delete from Capital where name = 'Dhaka';

① Delete the details of employee martin.

Delete from EMP where ENAME = 'Martin';

② Delete all the details of Dept 10.

Delete from EMP where Dept NO = 10;

③ Delete the details of research Department.

Delete from dept where DNAME = 'research';

Note :-

To display ~~only~~ the column names we use

DESC TN;

It is used to fetch what are the columns present in Table.

③ Update the salary to 20000 and commission to 10000 of ENAME Allen.

Update EMP set SAL = 20000, COMM = 10000 where ENAME = 'ALLEN'

Delete :-

- * It is used to delete particular records from the table and where condition is compulsory for deletion.
- * If we don't specify where condition then the whole table will be affected.

Syntax

Delete from TN; where condition;

ex:

Delete from Capital where name = 'Dhaka';

① Delete the details of employee martin.

Delete from EMP where ENAME = 'Martin';

② Delete all the details of Dept 10.

Delete from EMP where Dept No = 10;

③ Delete the details of research Department.

Delete from dept where DNAME = 'research';

Note :

To display ~~by~~ the column names we use

DESCRIBE TN;

It is used to fetch what are the columns present in Table.

Difference b/w Truncate & Delete

Truncate	Delete
* Truncate is a DDL command	* Delete is a DML command
* It is used to Delete all the records from the table	* This is used to Delete particular records from the table
* Truncate command is faster than Delete	* Delete is slower than Truncate
* We cannot use where condition to Delete specific record in Truncate	* we have to use where condition to delete specific record in delete

Difference b/w char and varchar

Char	Varchar
* Char data type is used to store characters of fixed length	* Varchar is used to store characters of variable length
* In char data type, the left out memory block will be wasted	* In varchar the left out memory block will be reused
* In char data type, the limit of string characters is upto 255	* In varchar the limit of string characters is upto 65535
* Better performance than varchar	* Performance is not good as char data type

Difference b/w varchar and varchar2

Varchar	Varchar2
* It can store upto 2000 bytes of memory	* It can store upto 4000 bytes of memory
* Varchar occupies extra space for null values	* Varchar2 doesn't occupy space for null values
* Varchar is ANSI standards	* Varchar2 is ORACLE standard

Primary key and Foreign key Difference.

Primary key	Foreign key
* Primary key does not allow duplicates and null values.	* Foreign key allows both duplicate and null values.
* For a Table we can have only one primary key.	* For a table we can have multiple foreign keys.

DDL Rename:-

DDL rename is used to change name of the table.

Syntax:-

Rename old TN to new TN.

TCL (Transaction Controlling Language)

TCL commands are used to manage the changes that are made by DML commands and TCL command's are as follows:-

1. Commit
2. Save point
3. Roll back.

Commit:-

Commit is used to save the data permanently.

Syntax:-

Commit;

2. Save point:-

It is used to save the data temporarily so that we can easily roll back to the point.

Syntax:-

savepoint savepointname;

Savepoint A:

insert into STD values (20, 'B');

Savepoint B:

insert into STD values (30, 'C');

Savepoint C:

insert into STD values (40, 'D');

Roll Back:

It is used to get or restore the data to the last save point.

Syntax:

Rollback to savepointname;

Ex:-

Rollback to C;

DCL (Data Controlling language)

- * DCL commands are mainly used to deal with the rights & permissions of the database.
- * We have 2 DCL commands

1. Grant

2. Revoke

1. Grant:-

Grant is used to give permissions to the user.

Syntax:

**Grant privilegeName on objectname to User
or command TableName**

Ex:- Grant select on EMP to HR;

Grant Select, update on EMP to HR;

2) Revoke:

Revoke is used to get back the given permissions from the user.

Syntax:

Revoke privilegeName on objectname from User

Ex:-

Revoke select, update on EMP from HR;

Functions :-

In SQL we have predefined functions which are used to perform some specific operations according to the user requirement or developer requirements.

Functions are classified into two types.

1. single row function
2. multi row function (aggregate, group functions)

2) Multirow function :-

It generally takes multiple inputs and generate only one output. The multirow functions are as follows.

(i) Max Function :-

It is used to display maximum value among given values.

Syntax: `Select max(i/p) from TN;`

Ex: `Select max(SAL) from emp;`

(ii) Min Function :-

It is used to display minimum value among given values.

Syntax: `Select min(i/p) from TN;`

Ex: `Select min(SAL) from emp;`

(iii) Sum Function :-

Syntax: It is used to display the sum of all the input values.

`Select sum(i/p) from TN;`

Ex: `Select sum(SAL) from emp;`

(iv) Average Function :-

Syntax: It is used to display the average of all the input values.

`Select avg(i/p) from TN;`

Ex: `Select avg(SAL) from emp;`

O/P 2073.21429

(v) Count Functions:-

It is used to count number of rows present in columns of a table.

Syntax: `select count(9/p) from TN;]`

Ex: `select count(SAL) from emp;`

O/P: 11

Note: For multi row function we cannot use where condition at the same time we can't use any other column with multi row function.

(1) Single row Functions:-

* These are also called as scalar functions.

* It takes multiple inputs and displays the same number of outputs.

O/Ps

* There are again classified into 4 types

Single row Function

Number function

- least
- mod()
- abs()
- power()
- sqrt()
- round()
- ceil()
- floor()
- ASCII
- greatest
- fact

char function

- length
- upper
- lower
- substr
- Replace
- Reverse
- initcap
- trim

Date function

- sysdate
- current_date
- add_months
- Add_months

conversion function

- To_date
- To_date
- months-between
- Add_months

Notes

→ In oracle we select the standard library.

→ And we can't select the standard library as a table.

Number Function :-

1. mod() function:-

It is used to display the remainder of given inputs.

- Ex: ① select mod(25,4) from dual; o/p: 1
 ② select mod(25,0) from dual; o/p: 25

2. Absolute() function:-

It will convert the negative values into positive.

- Ex: select abs(-15) from dual; o/p: 15
 select abs(15) from dual; o/p: 15

3. power() function:-

It is used to display the power value of given inputs.

- Ex: select power(5, 2) from dual; o/p: 25;

4. square root() function:-

It is used to find the root value of given value.

- Ex: select sqrt(16) from dual; o/p: 4.

5. Floor() function :-

It will convert the decimal values into integers and

it will display the least value as output.

- Ex: ① select floor(12.1) from dual; o/p: 12

- ② select floor(12.9) from dual; o/p: 12.

6. ceil() function:-

It is also used to bind the decimal value into

integers but it will display the highest value as output.

- Ex: ① select ceil(12.1) from dual; o/p: 13

- ② select ceil(12.9) from dual; o/p: 13.

7. Round() function:-

It is a combination of both floor function and ceil function.

→ From 12.1 to 12.4 it will display least value as output.

→ From 12.5 to 12.9 it will display greatest value as output.

ex: select round(12.1) from dual;

o/p:

$$12.1 \Rightarrow 12$$

$$12.5 \Rightarrow 13$$

$$12.9 \Rightarrow 13$$

8. ASCII :-

It is used to find ASCII value of any characters as special characters.

ex: select ASCII('A') from dual;

o/p: 65

select ASCII('!') from dual;

o/p: 33

2. Char Function :-

These are used to deal with the characters as strings.

1. length:

It is used to find the length of the given string.

ex: select length('sai') from dual;

o/p: 3

2. Upper:

It is used to convert all the lower case characters into uppercase.

ex: select upper('Sai') from dual; o/p: SAI

3. lower ():

It is used to convert the upper case characters into lower case.

ex: select lower('SAI') from dual;

o/p: sai

Substring Function:-

This function is used to fetch a part of given string.

Syntax :

`substring (source, position);`

Ex:- `substring (source, position, length) From dual;`

① select substr ('welcome') From dual;

comes

② select substr ('welcome', 4, 2) From dual;

co

Replace Function:-

It is a function which is used to replace the old string with new string.

Syntax:

`Replace (input, oldchar, newchar);`

Ex:-

select Replace ('Developers ', 'ers ', 'ment') From dual;

development

select Replace ('Java1', 'ja', 'Vimal') From dual;

Vimal

Reverse Function:

It is used to reverse the given string.

Syntax:

`Reverse (input/source);`

Ex:-

① select reverse ('hello') From dual;

olleh

② select reverse (ENAME) From EMP;

InitCap :-

It is a function which is used to display the first character of each word in a sentence in upper case.

Syntax :-

InitCap (source/ Input)

Ex:-

select initcap ('we are developers') From dual;

Op:- We Are Developers

Trim Function:-

The use of trim function is to remove the starting space and ending space of given string

Trim (Input/source)

Ex:-

select trim (' Hai ') from dual;

Op:-

Hai

Date Functions :-

- * The date functions are mainly used to deal with the dates.
- * The date functions are as follows:
 - 1. system date
 - 2. current date
 - 3. months between

1. System date :-

It is used to display the date present in the oracle system.

Ex:-

① select sysdate From dual;

Op:- 17-12-22

② Current date function:-

It will display today's date.

Ex:-

select current_date From dual;

Op:- 17-12-22

Months-between:
It will display the no. of months present b/w the given dates

Syntax:

`Months-between (Date1, Date2);`

Ex:

① `select Months-between('17-Dec-22', '17-Mar-23') from dual;`

Op:

② `select Months-between('17-Mar-23', '17-Dec-22') from dual;`

Op:

③ `select Months-between('17-Mar-23', '17-Mar-23') from dual;`

Op:

Add months :
It will display the date after adding the no. of months to the input date.

Syntax:

`Add-Months (Date, months);`

Ex:

① `select Add-months ('17-DEC-22', 6) From dual;`

Op: 17-Jun-23

Conversion function:

Conversion function having one function. It converts one data type into another data type.

① `TO-date()`

The use of TO_DATE function is to convert the character form to date into SQL default format.

of date into SQL default format.

Syntax:

`TO-date (I/P, format);` and condition "a set of characters

Ex:

① `select TO-date ('DEC-17-2022', 'MM-DD-YYYY') From dual;`

Op:

17-DEC-2022

② `select TO-date ('Dec/17/2022', 'DD/MM/YYYY') From dual;`

Op:

17-DEC-2022

Sub Queries:-

A query inside another query is known as nested sub-query.

Syntax:

Select * from dept

select (N) from TN <where condition> (select CN from TN);
 (Outer query) (Inner query)

If we want to fetch data from one table based on the input of another table then we have to go for sub queries

Execution process:

First the inner query gets executed based on inner query output
 the outer query gets executed.

Types of sub queries:

Sub queries are classified into 3 types

1. Single row subquery
2. Multi row subquery
3. Correlated subquery.

1. Single row subquery:-

* If the inner query is displaying only one output then it is said to be "single row subquery".

* In single row subquery we have to use '=' operator b/w the outer query and inner query.

2. Multi row subquery:-

* If the inner query is displaying more than one output then it is said to be a "multirow subquery".

* In multirow subquery we have to use 'in' operator b/w the outer query and inner query.

- ① Write a query to fetch all the details of emp who is getting max salary.

$$\text{select * from emp where SAL} = (\text{select max(SAL) from emp})$$
- ② write a query to fetch all the details of emp who is getting min sal

$$\text{select * from emp where SAL} = (\text{select min(SAL) from emp})$$
- ③ Display the employee name, SAL, & designation of emp who is working in newyork location.

$$\text{select ENAME, SAL, JOB from emp where DEPTNO} = (\text{select DEPTNO from DEPT where LOC} = \text{'newyork'})$$
- ④ display dept information of employee JAMES.

$$\text{select * from DEPT where ENAME} = (\text{select * from EMP where ENAME} = \text{'JAMES'})$$
- ⑤ Get the emp details of accounting dept

$$\text{select * from emp where DEPTNO} = (\text{select DEPTNO from DEPT where ENAME} = \text{'ACCOUNTING'))$$
- ⑥ display the dept information of employees who is getting 3000 as salary.

$$\text{select * from DEPT where DEPTNO} \in (\text{select DEPTNO from emp where SAL} = 3000)$$
- ⑦ Display the employees whose dept name is ending with 'S'
- ⑧ display the dept name & location of all the salesmen
- ⑨ display the dept information of employees except clerk
- ⑩ display the emp name, SAL, designation of employees who is working in Dallas.
- ⑪ List the employees whose dept name starts with 'O' and 4th character is 'R'
- ⑫ display the details of dept whose SAL is in the range of 3000-5000
- ⑬ display the dept information of employees who is not getting commission
- ⑭ display the dept name & location of employees who are getting commission

- ① select * from emp where deptno (select deptno from dept where dname like 'u.s.') =
- ② select * from dept where deptno in (select deptno from emp where job = 'salesman')
- ③ select * from dept where deptno in (select deptno from emp where job = 'clerk')
- ④ select ename, sal, job from emp where deptno = (select deptno from dept where loc = 'DALLAS')
- ⑤ select * from emp where deptno in (select deptno from emp where ename like 'm%' or job like '%clerk')
- ⑥ select * from dept where deptno in (select deptno from emp where sal between 3000 and 5000)
- ⑦ select * from dept where deptno in (select deptno from emp where comm is null)
- ⑧ select * from dept where deptno in (select deptno from emp where comm is not null)
- ⑨ display the dept name of employees who has no reporting manager.
- ⑩ write a query to display all the employees whose name is same as scott.
- ⑪ list the employees whose salary is greater than miller
- ⑫ list the employees whose job is same as Johns job
- ⑬ write a query to fetch second highest salary.
- ⑭ list the employees who is located in chicago
- ⑮ display all the employees whose salary is greater than avg salary of dept 20

- ① Select dname from dept where deptno in (select deptno from emp
where mgr = 'reporting manager') ;
mgr is 'null' ;
- ② Select * from emp where deptno in (select deptno from emp where
deptno = (select deptno from emp where ename = 'scott')) ;
(or)
- ③ select * from emp where deptno = (select deptno from emp where
ename = 'scott') ;
- ④ Select * from emp where SAL > (select SAL from emp where Ename='miller') ;
- ⑤ Select Ename from emp where job = (select job from emp where Ename='jones') ;
- ⑥ Select max(SAL) from emp where SAL <= (select max(SAL)
from emp) ;
- ⑦ Select Ename from emp where SAL > (select AVG(SAL) from emp
where deptno = 20) ;

Rownum Function:

* Rownum function is used to limit the no of rows return by a query

Syntax

select colName from TN where rownum = value

* Rownum function will start counting from the first column
and we can use only \leq or $=$ operator
 \neq operator will work for fetching only the first row.

① Write a query to fetch first 5 records from the table

select * from emp where Rownum \leq 6 ;
(or)

select * from emp where Rownum \leq 5 ;

② display top 10 records from the emp table.

select * from emp where rownum <= 10;

③ write a query to display only the first record from the table.

select * from emp where rownum = 1;

Distinct keyword:

* It is used to fetch the unique values by removing the duplicates.

Syntax:

select distinct CN from INR;

* For multiple columns distinct keyword will check each column.

* Value keyword removes the duplicates & Distinct keys temporary.

① Fetch the unique values of salaries.

select distinct SAL from emp;

② display the unique values of designation.

select distinct job from emp;

Order by clause:

* It is used to sort the data either in ascending or descending order.

* Order by clause can be used with select statement.

Syntax:

select CN from INR order by COLNAME ASC / DESC;

* If you are not specifying the order of the data to be sorted by default it will fetch the data in ascending order.

* It is also temporary.

① display salaries in ascending order.

select ename, sal from emp order by SAL ASC;

2nd syntax of subquery :-

select * from (select column from) where condition.

(Q) write a query to fetch 3rd highest salary.

select * from (select * from emp order by sal desc) where rownum=3
 (select * from (select * from emp order by sal desc) where rownum <= 3
 order by SAL ASC) where rownum <= 1

(Q) write a query to fetch unique as third highest SAL

select * from (select distinct SAL from emp order by
 SAL desc) where rownum=3 order by SAL ASC
 where rownum <= 1

Join

Joins

physical joins

logical joins

combined data

combination of data

→ cross joins

→ Equijoins

→ Inner join

→ self join

→ outer join

Cross joins

* cross joins is also called a cartesian joins

which is derived from set theory of mathematics

* In cross joins each value of first table will be comparing with all the values of 2nd table

* Cross joins will fetch the data if there is connection or no condition between columns

Syntax

select T₁.CN, T₂.CN --- from T₁ crossjoin T₂ where condition;

① Write a query to fetch the enames, dname, SAL and LOC

* select e.ename, e.SAL, D.Dname, D.loc from emp E, Dept D;

② display ename, sal, deptname, loc of employee 'smith'.

* select e.ename, e.sal, d.dname, d.loc from emp E, dept D
where E.ename = 'smith';

③ display emumber, designation, hiredate, loc of 'martin'.

* select e.empno, e.job, e.hiredate, d.loc from emp E,
dept D where E.ename = 'martin';

2) Equi joining:

Equi join is same as cross join but we have to specify the joining condition to remove the invalid combination of output.

structure

select T₁.CN, T₂.CN -- from T₁, T₂ where condition
and joining condition

Note :-

We have to write the joining condition between the common columns i.e., primary key column & foreign key column.

① Display name of emp, SAL, deptno, dname, deptno from dept table.

select e.ename, e.SAL, e.deptno, d.dname, d.deptno from
emp E, dept D where e.deptno = d.deptno

② display the ename, dname, deptno from both the tables of employee allen.

select e.ename, d.dname, e.deptno, d.deptno from
emp E, dept D where e.ename = 'Allen' AND
e.deptno = d.deptno;

Inner joins :-

Both equijoins and inner joins are same but the way of writing inner join query is different.

Syntax :-

select $T_1.CN, T_2.CN \dots$ from T_1 , innerjoin T_2 on
cond and J.C;

join condition is the condition which will be satisfied during joining condition

① want to fetch name of the employee, designation & department of emp.

martin.

select e.ename, e.job, d.dname from emp e, innerjoin dept d
on e.ename = 'martin' and e.deptno = d.deptno;

② display the empno, SAL, and loc of allen.

select e.empno, e.SAL, d.loc from emp e, innerjoin dept d
on e.ename = 'allen' and e.deptno = d.deptno;

③ display dname and commission of all the salesman.

select d.dname, e.comm from dept d, innerjoin emp e
on e.job = 'salesman' and e.deptno = d.deptno;

Self joins :- A table joining itself is known as self-join.

A table joining itself is known as self-join.

Syntax :-

select $T_1.CN, T_2.CN \dots$ from T_1, T_2 where condition
and joining condition.

① display ename, SAL, comm and deptno of employees whose SAL is same as martin's salary

select e1.ename, e1.SAL, e1.comm, e2.deptno where e1.ename =
martin and e1.SAL = e2.SAL;

② display empno, ename, and sal of employees whose job is same as Johns job.

A) select e2.empno, e2.ename, e2.sal from emp e1, emp e2 where e1.ename = 'JONES' and e2.job = e1.job

Outer joins

* It will fetch all the data from one table and only the matched data from another table.

* Outer joins are classified into 3 types

1. Left outerjoin
2. Right outerjoin
3. Full outerjoin

leftouterjoin

It will fetch all the data from the leftside table and only the matched data from rightside table.

Syntax:

select T1.CN, T2.CN -- from T1 leftjoin T2 on
T1.empno = T2.empno condition and join conditions;

①

select e.empno, e.deptno, d.deptno, d.dname from employee
leftjoin dept d on e.deptno = d.deptno;

Right outerjoin:

It will fetch all the data of Rightside table and only the matched data from leftside table.

Syntax:

select T1.CN, T2.CN -- from T1 rightjoin T2 on condition
and join conditions;

①

Select e.empno, e.deptno, d.deptno, d.dname from emp
 Right join dept d on E.deptno = d.deptno;

full outer join :-

It is used to fetch both matched and unmatched data from the tables.

syntax

Select T₁.CN, T₂.CN from T₁ FullJoin T₂ on condition and join conditions;

e.g.

Select e.empno, e.deptno, d.deptno, d.dname, from emp FullJoin dept d on e.deptno = d.deptno;

Group by clause :-

If group by clause is used to group the rows with same values.

syntax

Select CN from TN Group by CN;

Note :-

In the select statement or group by clause we have to use only the column names & multirow functions.

② WAP to fetch avg salary of each job

Select Avg(SAL), JOB from emp group by JOB;

③ display the no. of employees present in each dept

Select count(ename), deptno from emp group by deptno;

③ WAG to fetch the highest SAL in each dept.

select max(SAL), deptno from emp group by deptno;

Having clause

* The having clause is added to SQL because we cannot use the where condition for the multirow functions.

* multirow function is also called as aggregate function or group function

statement

[select CN(S) from TH Group by CN Having condition;]

① WAG to fetch the max salary of each dept where the max-salary should be greater than 3000

select max(SAL), deptno from emp group by deptno having

② display the no of employees present in each dept where the count of employees should be greater than 5

select count(ename), deptno from emp group by deptno having

③ display the sum of SAL of each job where the sum of SAL should be greater than 3500

select sum(SAL), job from emp group by job having

sum(SAL) > 3500

Index

Indices are used to get the data/output very quickly.

Syntax

`create index Indexname on TN (CN1, CN2, ...)`

Advantages of creating Index

1. we can save the time by creating index because we can get the output faster.
2. The database performance will be increased.

e.g.

`create index I2 on emp (Job, SAL)`

Views:-

* views are used to maintain a logical copy of data that is

Required

* It is used to display the necessary data by hiding the unnecessary data.

Advantages

- * provides security for the data.
- * Increases the database performance

Syntax

`create view Viewname as select CN from TN;`

e.g.

How to connect system? After opening the oracle terminal type

`>>> conn system`

enter password: connected.

`>>> grant create view to scott;`

Grant succeeded.

Now open sqlplus and enter the command

can start

enter password : * * * * *
connected.

SQL create views v1 AS select hiredate, comm from
view created.

Questions on functions

- ① WAP to fetch the details of odd salaries.
- ② WAP to fetch first 3 characters of the emp names.
- ③ display deptnames whose loc consist of atleast 7 characters.
- ④ display the unique values of designation & prints its length.
- ⑤ display all the students name in lower case from student table.
- ⑥ display the query to combine first name column, last name column into single, complete table known as EMP Worker table.
- ⑦ calculate 'H�dt'('C') present in string 'Accounting'.
- ⑧ display the emp names whose job consist of atleast 6 characters.
- ⑨ display the empno, name of employees whose job consist of man as last 3 characters.
- ⑩ select SAL from emp where MOD(SAL, 2) = 1;
- ⑪ select substr(ename, 1, 3) from emp;
- ⑫ select dname from dept where length(loc) >= 7;
- ⑬ select distinct job, length(job) from emp;
- ⑭ select lower(^{ename}) from emp;
- ⑮ select empno || deptno from emp;
- ⑯ select ename from emp where length(job) >= 6;
- ⑰ select empno, ename from emp where job like '1% man';

Q) Select empno, ename from emp where

$\text{substr}(\text{job}, \text{length}(\text{job}) - 2, \text{length}(\text{job})) = 'MAN'$

(or)

Select empno, ename from emp where $\text{substr}(\text{job}, -3) = 'MAN'$

Q) Select length('Accounting') - length(Replace('Accounting', 'c', ''))
from dual;

Normalization :-

- * It is a process of organizing the data in the database.
- * It is used to remove the duplicate data in the database.
- * It is also used to eliminate insertion, deletion, update anomalies.
- * normalization divides the larger tables into smaller tables and join them by using relationship

Types of normalization

1. First normal form
2. Second normal form
3. Third normal form

i) First normal form:

A table is a first normal form if and if only

- (i) There are only single value attributes.
- (ii) All the CN are unique.
- (iii) Attribute domain doesn't change.

To Char :-

It is used to convert/fetch user date format from system date / current date.

Syntax

```
select to_char('inputdate', 'Dateformat');
```

① select to_char(sysdate, 'yyyy') from dual;

o/p: 2023

② select to_char(sysdate, 'year') from dual;

o/p: Two Thousand Twenty Three.

③ select to_char(sysdate, 'yy') from dual;

o/p: 23

④ select to_char(sysdate, 'day') from dual;

o/p: Sunday.

⑤ select to_char(Hiredate, 'day') from emp;

⑥ select to_char(Month, 'month') from dual;

o/p: June

⑦ select to_char(sysdate, 'dd') from dual;

o/p: 25

⑧

Comments

1) Single line Comments:

we can use `-- message/content`

for single line comments

2) Multiline Comments

we can use `(* line1
line 2
line 3 *)`

for multiline comments

3) Inline Comments

we can use `select * from /*Custom*/`

Query for the no. of Columns in the table

```
Select count(*) as NumberofColumns from ALL_TAB_COLUMNS
where Table_Name = TN;
```

Query to fetch what are the columns present in the Table

```
Select COLUMN_NAME from ALL_TAB_COLUMNS where
Table_Name = TN;
```

Datatypes of each Column in Table.

```
Describe TN;
```

Delete specified Column of the Table

```
Alter Table TN drop column CN;
```

- ③ List all the sales man in dept 30
- ④ List all the sales man in dept 30 and having salary > 1500
- ⑤ Display all the details of employees whose name starts with a vowel
- ⑥ Display the details of employee whose name consists of at least 2 'L's
- ⑦ List all the employees whose name does not ends with 'es'
- ⑧ Display the dept information of employees whose working location ends with 'N' or 'K'
- ① select * from EMP where JOB = 'SALESMAN' AND DEPTNO = 30;
- ② select * from EMP where JOB = 'SALESMAN' AND DEPTNO = 30
AND SAL > 1500;
- ③ select * from emp where ename like 'A%.' or ename like 'E%.'
or ename like 'I%.' or ename like 'O%.' or ename like 'U%.'
- ④ select Ename from emp where Ename not like '%.es';
- ⑤ Select * from EMP where ENAME NOT like '%.ES';

- ① QTF dname of emp's who is getting highest comm?
- ② Display Name & Job of emp's who is getting highest salary in Accounting dept?
- ③ Display the details of Smith's manager's manager?
- ④ Display the emp details who is getting sal more than scott & less than king?
- ⑤ What is the dname of emp's who is getting 5th highest salary?
- ⑥ Display the Dname of Allen's Managers?
- ⑦ Display the Dept info of emp's who is getting salary more than scott and job not same as jones?

- ① Select DNAME from DEPT where DEPTNO = (select DEPTNO from EMP where comm = (select max(comm) from Emp));
- ② DNAME of employees who is getting highest commission
display Name, Job of employees who is getting highest salary in
- ③ Accounting dept.
- ④ select ENAME, JOB, SAL, DEPTNO from emp where SAL = (select MAX(SAL) from emp where Deptno = (select DEPTNO from DEPT where DNAME = 'Accounting'))
- ⑤ Display details of Smith's Manager
select * from emp where JOB = 'Manager' and Deptno = (select Deptno from emp where ENAME = 'SMITH');
- ⑥ Display details of employees who is getting SAL greater than 'scott' and less than 'king'
select * from emp where SAL > (select SAL from emp where ENAME = 'SCOTT') and SAL < (select SAL from emp where ENAME = 'KING');
- Display employee details who is getting SAL less than 'SCOTT' and less greater than 'king'.
select * from emp where SAL > (select SAL from emp where ENAME = 'KING') or SAL < (select SAL from emp where ENAME = 'SCOTT');
- ⑦ Display name of employee of getting 5th highest salary.
select * from (select * from (select ENAME, SAL from EMP order by SAL DESC) where Rownum <= 5) order by SAL ASC) where Rownum = 1;

⑥ Dname of Allen's manager.

select Dname from DEPT where DEPTNO = (select Deptno from emp where ename job = 'MANAGER' and DEPTNO = (select deptno from emp where ename = 'Allen'));

⑦ Display dept info of employees who is getting SAL more than scott and Job not same as Jones.

select * from dept where deptno in (select Deptno from emp where sal > (select sal from emp where ename = 'scott')) and job != (select job from emp where ename = 'Jones'));

⑧ Display dname of emp who is getting 5th highest salary.

select Dname from Dept where Deptno = (select Deptno from (select * from (select sal, deptno from emp order by sal desc) where Rownum <= 5 order by sal) where rownum = 1);

⑨ WAP to fetch all emp details whose location same as Adams.

select * from emp where deptno = (select deptno from emp where ename = 'ADAMS');

10) Fetch odd salaries without using function

select * from emp where substr(sal, -1) in (1, 3, 5, 7, 9);