

Data:

It is a record that describes properties of an object.

Database:

It is a place or container used to store the data in an organised manner.

Database operations:

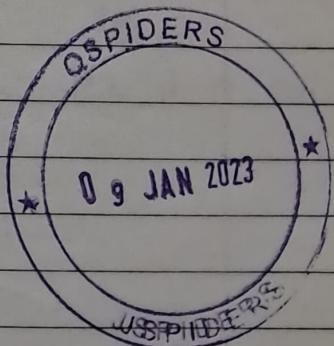
- Create
- Read.
- Update
- Delete.
- Database operation is also known as CRUD operation.

DBMS: (Database Management System)

- Software used to maintain and manage the database.
- To communicate with DBMS use query language
- DBMS provides security and authorization.

Types of DBMS:

- ① - RDBMS
- ② - Object oriented
- ③ - Network
- ④ - Hierarchical

① RDBMS (Relational Database Management System):

- Software used to maintain, manage & store the value in form of table.

## Data types

Data type is a type of data used to store in memory.

Type:

- 1. char    2. varchar    3. Large objd    4. date    5. Number.
- a. Binary large object
- b. char large object

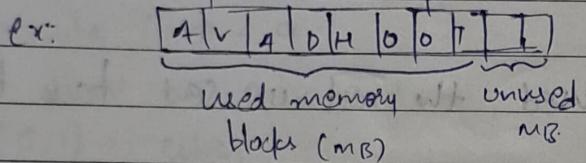
1. char:

- [A-Z], (a-z), (0-9), (\$-@)

- Syntax: char ( $\rightarrow$  length)

Ex: student name  $\rightarrow$  char (10)

10 blocks ~~for~~ allocated for each name from memory area



Initial point      used mB      Unused mB      Final point:

10                  8                  2                  10

10                  5                  5                  10

10                  6                  4                  10

If the initial and final points are the same then it is called fixed length memory allocation.

2. Varchar:

Syntax: varchar ( $\rightarrow$  length).

The unused memory blocks are left to the memory area.

The initial and final points are not the same.

This is known as variable length memory allocation.

Imp:- The max character length of char and varchar is 2000.

char data type is faster than varchar as unused memory blocks are sent back to memory area in varchar.

Variable Length:  
data type which is considered as variable length memory allocation  
and max size is 4000 memory blocks.

### 3. Large object:

#### a. Char Large Obj:

→ A-Z, a-z, 0-9, \$, @

→ Syntax - CLOB

→ length need not be mentioned as it is very large and unpredictable

→ Oracle database has a max. size of 4 GB

→ MySQL → 12-14 GB

→ If only 2GB is used, the rest is sent back to memory area.

#### b. Binary Large Obj:

→ It is a data type used to store audio, video, images, documents inside a database after converting into binary value using Java methods.

→ Syntax = BLOB

→ max size → 4GB

### 4. Date:

- Syntax: DD-MON-YYYY & DD-MON-YY

Ex: 14-AUG

14-AUG-23

14-AUG-1997

- first format used to denote the past and future years.

- second format used to denote the present years.

- Data type used to indicate date.

## 5. Number:

Assume a number: 24689.45

format: Number (P,F, scale)

- P stands for precision and means total number of digits
- No. of digits in decimal part is scale
- for above example, P=7 & Scale = 2
- according to format  $\rightarrow$  Number (7,2)
- Limit for precision - (1-38)
- Limit for scale - (0-127)

### Examples:

- a. Number (6) - 999999
- b. Number (6,3) - 999.999
- c. Number (1,0) - 9
- d. (2,4) - 0.0099
- e. (0,1)  $\rightarrow$  error.
- f. (2,5)  $\rightarrow$  0.00099

## \* CONSTRAINTS

- Set of conditions used to validate the data entered into the cell.
- Types:
  - a. Unique
  - b. Not null
  - c. Check
  - d. Primary Key
  - e. Foreign Key

### a. Unique:

- constraint used to avoid duplicate data entered into the cell.
- syntax: Unique (col. name)

b. Not null:

- Constraint that represents something in a col.
- Syntax : not null.
- Null is a keyword that represents nothing.
- $5 + \text{Null} = \text{Null}$  •  $\text{Null} + \text{Null} = \text{Error}$ .
- $0 + \text{Null} = \text{Null}$ .
- Not null occupies a memory block whereas null does not.

c. Check:

- It is a condition to validate a column.
- Syntax : check ( condition )

example

i. check ( $\text{sal} > 30000$  AND  $\text{sal} < 100000$ )

ii. check length (student contact) = 10;

d. Primary Key:

- It is a key used to uniquely identify a row in a table.
- characteristics:
  - It should be unique.
  - It should be not null.
  - It should be a combination of unique and not null.
  - In a table there should be only one Primary Key.
  - Primary key is present in parent table.
  - Primary key is not mandatory but design voice is preferable.
  - Primary key represents the table.

### c. Foreign Key:

- > It is a key used to make connections from one table to another one.
- > characteristics:
  - i. It can be unique.
  - ii. It can be not null.
  - iii. It is not a combination of unique and not null.
  - iv. Primary key of one table is foreign key of another table.
  - v. Foreign key is present in child table.
  - vi. Foreign key is also known as Referential Integrity Constraint.
  - vii. We can have multiple foreign keys.

### \* SQL statements:

- |             |           |              |              |               |
|-------------|-----------|--------------|--------------|---------------|
| 1. DDL      | 2. DML    | 3. TCI       | 4. LOCL      | 5. QPL        |
| a. Create   | a. Insert | a. Commit    | a. Grant     | a. Select     |
| b. Rename   | b. Update | b. Rollback  | b. Procedure | b. Projection |
| c. Alter    | c. Delete | c. Savepoint |              | c. Selection  |
| d. Truncate |           |              |              | d. Joining    |
| e. Drop     |           |              |              |               |

### I. Data Query Language: (DQL)

1. Select: Retrieval of ~~data from~~<sup>data</sup> table which is present in database.
2. Projection: Retrieval of ~~data~~<sup>data</sup> from table by providing column name.
3. Selection: Retrieval of data from table by providing both row and column name.
4. Joins: Retrieval of data from multiple table simultaneously.

## 1. Select:

Syntax: `From <tablename>`

- here 'from' is a clause.

'from' clause goes to the database ~~searches~~ for table name and puts it in the execution area.

## 2. Projection:

Syntax: `SELECT * / [distinct] columnname / expression [aliasname]`

- here 'select' is a clause.

'select' clause goes to the table present in execution area search for the column name and displays the d.p.

Q. WAPTD 'Ename' from 'Emp'.

→ `Select Ename  
from Emp;`

a. Asterisk (\*):

wild card character used to display all the details of the table.

Q. WAPTD all the details of the sal grade table.

→ `Select *  
from sal grade;`

~~Imp~~ Q. WAPTD all the tables present in database

→ `Select *  
from fat;`

## b. Expression:

By using two or more operand operands and considering some operation we can get some clp. ex:  $\text{sal} \times 12$ ;

Q. Emp name, sal, annual sal, 10% ↑ in monthly salary, 10% in commission

→ select ename, sal,  $\text{sal} \times 12$ ,  $\text{sal} \times 1.1$ ,  ~~$\text{sal} \times 0.1$~~ , comm  $\times 0.9$  from emp;

## c. Distinct:

It is a keyword used to a unique values @ the time of execution.

Q. Write a query to display unique salary from emp table.

→ select unique (sal)  
from emp;

## d. Alias name

It is an alternate name given to the column or expression @ the time of execution.

Methods to provide alias name

$\text{Sal} \times 12$  Annual

$\text{Sal} \times 12$  "Ann sal"

From As "Empname"

Q. Ename, job using suitable aliasname.

→ select ename "Emp name", job "designation"  
from emp;

## \* Queries on where clause

1. Select sal  
from emp  
where sal > '1500';
2. Select \*  
from emp  
where job = 'Manager';
3. Select ename, deptno  
from dept  
where loc = 'New York';
4. Select \*  
from emp  
where hiredate = '01-JAN-1995';
5. Select deptno, loc  
from dept  
where dept = 'Research';
6. Select \*  
from emp  
where sal > '2000';
7. Select sal, sal\*12, sal\*3  
from emp  
where sal > comm;
8. Select sal, sal\*12, sal\*3  
from emp  
where job = 'clerk';
9. Select \*  
from emp  
where deptno = 20;
10. Select sal, sal\*12, sal\*3  
from emp  
where job = 'Manager';
11. Select \*  
from emp  
where ename = 'Ford';
12. Select sal\*12, sal\*12+1000,  
sal + 1000  
from emp  
where deptno = 30;

### 3. Selection:

- Retrieval of data by providing both row and column name.
- 'where' clause :-
- It is used to filter the condition of rows.
- Syntax: where condition operator value;  
ex: where cname = 'Smith';
- Characteristics of where clause
- i) It executes row by row.
- ii) It gets executed after from clause.
- iii) In where clause we can take multiple conditions.
- iv) In where clause we can't write multi-row function.

### Q. WAP TO cname, sal if they are not getting a sal of 3000.

→ select cname, sal

from emp

where sal != 3000;

- \* Operators: ~~in~~ ~~and~~ ~~or~~ ~~not~~ ~~with~~ ~~if~~
- 1. Arithmetic operators : (+, -, \*, /)
- 2. Comparison: (>, <, >=, <=)
- 3. Relational: (>, <, =, !=)
- 4. Concatenation (||)
- 5. Special operator

i) IN, NOT IN

ii) Is, Is NOT

iii) Between, Not Between

iv) Like, Not Like

### G. Subquery operator

All, Any, Some, Exists, Not Exists

Exists, not Exists

\* AND                    OR

AND	OR
1 0 0	1 0 0
1 1 1	1 0 1
1 0 0	1 1 1
0 1 0	0 1 1
0 0 0	0 0 0

Q. Ename, sal, job if they are getting sal > 2000 in dept no 20

Select ename, sal, job

from emp

where sal > 2000 AND Deptno = 20;

Q. Ename, sal if they are working in deptno 30 and hirdate is before 1985

Select ename, sal

from emp

where deptno = 30 OR hirdate < '01-JAN-1985';

Q. all the details if they are working as analysts and sal > 2500 or working deptno 20.

Select \*

from emp

where (job = 'analysts' AND sal > 2500) OR (deptno = 20);

\* IN operator:

- It is a multivalue operator which takes single value in the LHS and multiple values in the RHS.

- Syntax: col name IN (v<sub>1</sub>, v<sub>2</sub>, ..., v<sub>n</sub>)

OR operator.

ex: `ename, job IN (10, 20, 30);`

- ♀ ename, job if they are working as analyst or manager or salesman in dept no. 10 or 20 or 30.
- Select ename, job  
from emp  
where (job IN ('Analyst', 'Manager', 'Salesman')) AND deptno IN (10, 20, 30);
- ♀ ename, job, dept no if the employee is not working in dept no 10, 30
- Select ename, job, dept no  
from emp  
where deptno != 10 AND deptno != 30;

#### \* NOT IN OPERATOR:

- multivalue operator which takes single value in LHS and multiple values in RHS
- syntax: col\_name NOT IN (v<sub>1</sub>, v<sub>2</sub>, ..., v<sub>n</sub>)  
AND operator

Assignment:

Equal

In.

- It takes single value in LHS - Single value in LHS and multiple values in RHS.
- It is faster It is slower.
- It is a single value operator It is a multi-value operator.

### \* IS operator

- Operator used to check NULL or NOT NULL values.  
Syntax:- Col name IS NULL / NOT NULL.

### \* LIKE operator:

- Operator used to check pattern matching.  
Syntax:- Col name Like '\_\_\_\_\_'

% ←                  \_ → (Underscore)  
 • Any char      • Any char only  
 any no. of time      one time.

Ex:- WAP TO查找出 job where name starts with A

Select ename, job  
from emp

where ename Like 'A%' ;

name ends with A

Select ename, job

from emp

where ename Like '%A' ;

### \* Escape character (!):

- It is a character used to convert wildcard character into special characters.

- For example we have '%' character in Like operator and has a defined function.

- Suppose a name starts with '%' and we need to print that name, then we need the '%' present in name behave like a special character and not a wildcard character.

- The escape (!) character convert the '%' into a special char.

- For that we use the escape (!) character.

ex:- where ename like '% %' escape('!');

#### \* BETWEEN operator:

- Operator to check ranges from lower value towards higher value.

Syntax: col\_name Between x and y.

here both x and y gets included by default. ( $\geq x, \leq y$ ).

Sal between 3000 AND 5000; i.e.  $sal \geq 3000$  AND  $sal \leq 5000$ .

#### \* NOT BETWEEN operator:

- Operator to check ranges from lower value towards higher value.

Syntax: col\_name Not Between x and y.

here x and y are excluded.

e.g. Sal not between 3000 AND 5000.

i.e.  $sal < 3000$  AND  $sal > 5000$

## \* FUNCTIONS

- It is a block of code used to generate some output.
- Types of functions:
- i. Built in func.
- ii. User defined func.

### i. Types of built in func:

- a. Single row func b. Multi row func

#### a. Single Row func (Scalar func)

- It is a func which takes in no. of i/p and generates 'n' no. of o/p.
- It executes row by row.

### b. Multi-row func (Aggregate func):

- It takes 'n' no. of ip and generates only 1 o/p.
- Types of multi-row func:  
Max(), Min(), Sum(), Avg(), Count()
- It executes group by group.

### \* Group By clause:

- Clause used to create the group or group the record!

Syntax: Group by col-name/ expression.

group by expression.

### Functionality of group by clause:

- It goes to the table under execution area & searches for the column name, execute row by row to start creating group.

### Characteristics:

- It is used to create group.
- It executes after from clause ~~by~~ but if there is a condition given in the question then it gets executed after where clause.
- Along with group by clause the written expression is known as group by expression.
- After group by clause ~~every~~ every clause gets executed group by group.
- By putting the normal column name into the group by clause, it converts into group automatically, so we can write it with multi-row func in select clause.

from → where → group by → having → select → order by

Page No.

Date

- \* HAVING clause
- used to filter the group.

Syntax: HAVING multirow function condition;

characteristics:

- It executes group by group
- It executes after from clause but if there is where clause condition & group by clause it will execute after it.
- In having clause we can take multiple multirow func<sup>n</sup> condition
- In having clause we can write grouped column condition.

Ex: wanted max sal, min sal in each dept if they are getting some sal and min sal < 2000.

→ Select max(sal), min(sal), deptno  
from emp  
where sal is not null

group by dept no

having min(sal) < 2000;

Int

#### \* ORDER BY CLAUSE:

- It is a clause used to sort the values.
- Syntax: order by column name / expression / multirow function / alias name DESC / ASC.
- DESC stands for descending order and ASC for ascending order.

characteristics:

- It is used to sort the value.
- It executes after select clause.
- In order by clause we can take multirow function.
- By default order by clause executes in ascending order.

\* \* path of execution:

from → where → groupby → having → select → orderby.

### \* Subquery:

- Query inside another query is known as Subquery.
- There are 2 kinds of Subqueries:
  - a) Inner query
  - b) Outer query.
- The inner query gets executed first.
- The inner query is the unknown value.
- OLP of inner query is the ref to the outer query.
- OLP of inner query is called partial OLP.
- OLP of outer query is called complete OLP.
- Outer query is dependent on the inner query while inner query is independent.
- Inner query is called Subquery while outer query is called main query.

Ex: Wanted ename, sal from emp if they are getting sal more than ALLEN.

→ Select ename, sal

from emp

where sal > (Select sal

from emp

where ename = 'ALLEN');

Wanted ename where there is only one 'S' in the name.

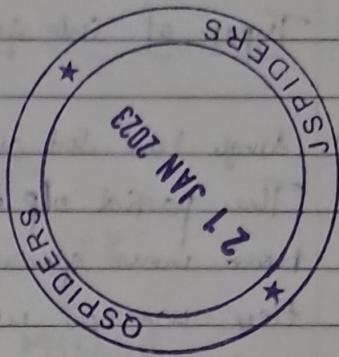
Select ename

from emp

where ename like '%.S%' and ename not like '%.S%.S%';

## In \* Nesting of query.

Ex:   
WAP TO second highest sal.  
 select max(sal)  
 from emp  
 where sal < (select max(sal)  
 from emp);



NOTE: It is possible to nest 255 sub queries.

### \* Conditions of Sub-queries:

- i. Inner query select clause col name should be same as outer query where clause col name.
- ii. Inner query select clause col name data type should be same as outer query where clause col name data type.
- iii. Inner query select clause should not have multiple column names to be compared with outer query where clause col name.

## Insp

- P. WAP TO Ename, sal, job if they're working in Research dept.
- Select Ename, sal, job  
 from Emp  
 where deptno = (Select deptno  
 from dept  
 where dept = 'Research'));
- Q. Ename, Loc if emp name is King  
 Select Ename, Loc  
 from dept  
 where deptno = (Select deptno from Emp where Ename = 'King');

\* Types of sub-queries:

i) Single row sub query

ii) Multirow subquery.

- The partial op is single from inner query.
- Here we can use relational and comparison operators.
- The partial op is multiple from inner query.
- Here we have to use sub-query and special operators.

\* ALL operator.

- It is a multi-value operator which takes single value in LHS and multiple value in RHS along with comparison operator.

- It is same as 'AND' operator.

Q. Find Ename, job, sal who are having sal more than all the managers.

→ Select Ename, job, sal  
from Emp  
where sal > ALL (Select sal  
from Emp  
where job = 'Manager');

\* ANY Operator.

- It is a multi-value operator which takes single value in LHS and multiple values in the RHS along with comparison operator.

- ANY operator is same as 'OR' operator.