* Database *

• SQL - Structured Query Language.

* What is Database ?

1) A database is a systematic collection of data. Databases support electronic storage and manipulation of Data. also used into data management.

e.g - consider Facebook, It need to store, manipulate and present data related to members, their friends, member activities, messages, advertisements and many more.

* Relational database Management sim (RDBMS) – is one of the type of database.

- Which is (RDBMS) the most popular DBMS type in the market.
- RDBMS includes database like MySQL, Oracle, MongoDB, and Microsoft SQL server database. as well SQL.

* Popular Database Management Studio –

- MongoDB
- MsSQL
- Postgre SQL
- Oracle DB.
- Microsoft access
- SQL server

* Structured Query Language (SQL)

Introduction - SQL - stands for structured Query language.
- SQL is an Standard ANSI (1986) / ISO (1987) standard, Language for storing, manipulating and retrieving data in databases.
- SQL Keywords are NOT Case Sensitive.
- SQL is the standard language for dealing with Relational Database. like MySQL, oracle, MS SQL server etc.

What SQL can do ! - 1) SQL can execute queries against a database.
2) SQL can retriever data from database

* Latest version of Oracle - 11g

(3) SQL can insert records
(4) SQL can update records
(5) SQL can delete records
(6) SQL can create new database as well create new tables in dB.
(7) SQL can create stored procedures (SP).
(8) also create an views in database.
(9) SQL can set permissions on tables, procedures & views etc.

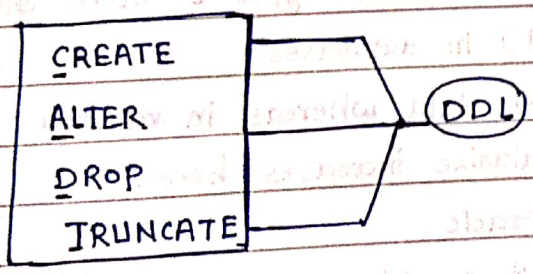
NOTE - The data in RDBMS is stored in database object called tables. A tables is collection of related data entries and it consist of rows and columns etc.

Interview
@ What are the types of command you had using in SQL ?
@ Explain DDL & DML ?

* SQL Command Types -

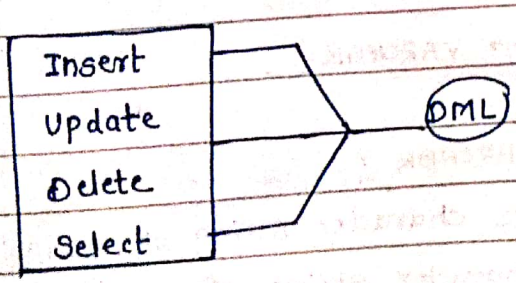
- DDL - (Data Definition Language) - Allow to work with structure/def?

trick Acidity



- DML - (Data Manipulation language) - To deal with the data itself
  ↳ used to

SUID



- DQL (Data Query language) - deals with data but to retrieve the data.

* Data Types In SQL -

- Characters / String :- • CHAR (20) — By default data size
  • VARCHAR (255) — By default data size
- Numbers / Numeric :- • INT (4 byte)
  • REAL FLOAT (4 byte)
  (Decimal)
- Times and Dates :- • Date (3 bytes)
  • DateTime (8 byte)
- Binary :- • Binary

@ Difference between varchar and varchar2 ?-

↳ varchar and varchar2 both are data types to store character / strings for particular columns (field) in databases.

- varchar allocate fixed size of data, whereas in varchar2 there is no limitation in datasize. datasize increases here.
- These are reserved by Oracle.
- Oracle says that, we don't consider as datatype as VARCHAR becz, It doesn't contains our properties that's why bydefault it considers as 'VARCHAR$_2$'.

VARCHAR ⟶ VARCHAR$_2$.

@ Difference bet^n. CHAR & VARCHAR ?

↳ VARCHAR is used to store character string of variable length. while CHAR is used to store character string of fixed length.

* Why VARCHAR$_2$ is 255 ?

↳ VARCHAR$_2$ (255) It means a string of 255 bytes, not characters.

* SQL Operators -

Types	Operator	Description
Arithmetic operator	+	- Add values
	−	- substract values
	*	- Multiplies values
	/	- divide left hand operand by right.
	% (Modulus)	- same like division but returns rmainder
Logical operator	AND	- The AND operator allows the existence of multiple conditions in SQL stml's WHERE clause.
	OR	- used to combine multiple conditions in SQL stmt's WHERE clause.

* SQL commands

① **Create (DDL)** - It is SQL statement which is used to create new table into database.

Syntax - Create table tablename (column1 datatype (size), column2 datatype (size).....);

eg - create table practice (FN VARCHAR2(10), LN VARCHAR2(20), MB INT...);

② **Select (DML)** - used to view table, select/fetch data from table

syntax - select *from tablename ;

- select *from practice ;

→

(Select Query →)

(1) Fetch particular column OR two column

Syntax - Select column_name from tablename ;

eg - Select FN From practice ;

FN
-
-
-
-

(2) 2 column fetch (FN , LN)

syntax (eg) - select FN , LN from practice ;

FN	LN
-	-
-	-
-	-

◉ Where - clause - used to specify condition while fetching data from single table .

Syntax - Select *from table_name where condition ;

eg -(1) Fetch data of `Neha'.

Select *from practice where FN = 'Neha' ;

NOTE - AND & OR - operator used with where clause for fetching data with multiple condition .

(2) AND with where clause - & OR - with where clause.

eg - ~~Select *from practice where FN = 'Neha' AND LN = 'Joshi' ;~~

eg -

FN	LN	Gender	Location
Neha	Bedre	F	Pune
Neha	Joshi	F	Aurangabad

AND

Select *from practice where FN='Neha' AND LN = 'Joshi' ;

o/p

Neha	Joshi	F	Aurangabad

OR

Select *from Practice where FN = 'Neha' OR LN = 'Joshi' ;

Neha	Bedre	F	Pune
Neha	Joshi	F	Aurangabad

③ Insert into (DML) - used to insert a single / multiple records into table.

syntax - insert into tablename values (value1', 'value2' .....);

e.g - insert into practice values ('Aditya', 'Patki', 'M', 7, 'Nagpur');

④ Desc - used to describe the structure of table.

syntax - Desc table_name ;

e.g - Desc practice ;

⑤ Delete - (DML) - used to delete a single or multiple records from table (using - where clause with AND, OR)

- Syntax - Delete from table_name ;

e.g - Delete from practice ; .....o/p

FN	LN	Gender	city	CN

..... Here, does not delete table structure only delete records

- Delete single or multiple records

syntax - Delete from tablename where condition ;

- Delete from practice where FN = 'Neha' AND LN = 'Joshi';

⑥ Drop (DDL) - used to delete structure of table.

syntax - drop table table_name ;

e.g. - drop table practice ;

⑦ Alter (DDL) - used to add, drop, rename (column | table), Modify columns in an existing table.

(ADD) syntax - Alter table tablename add column_name datatype (size);

e.g - Alter table ~~t~~practice add MB Int ;

→

Alter (Drop) - used to delete particular column.

syntax - alter table table_name drop column column_name;

e.g - alter table practice drop column MB ;

(alter rename) - used to rename the column name or tablename;

- syntax - alter table tablename rename column old column to Newold column_name ; ------ rename columnname
  e.g - alter table practice rename column salary to payment ; (old) (New)
- rename tablename -
  syntax - alter table table_name rename to new table_name ;
  e.g - alter table practice rename to velocity ;

(alter modify) - used to modify column with datatype.

syntax - alter table table_name modify column_name datatype ;

e.g - alter table practice modify MN INT ;

⑧ Update (DML) - used to update existing records in a table.

syntax - update table_name set column_name = 'value' where column_name = 'value';

e.g - 1> update practice set salary = 80000 where FN = 'Manju';
2> update practice set salary = 100000 where FN = 'aniket' OR LN = 'Patil' ;

⑨ Between - operator used to selects values within a given range with a WHERE clause.

syntax - select * from tablename where column_name between value 1 AND value 2

e.g - Select * from practice where Mockresult between 15 AND 20 ;

asked In Interview

@ can you please right Query, there is one table & it contain one column i.e salary, I want to fetch data between 10K to 15K ?

↳ Query :- between

- Select *from table_name where column_name between value1 AND value2;

Ans. ⇒ select *from practice where salary between 10K AND 15K;
1000 AND 2000;
6000 AND 10000;

(10) Aggregate function / Parameters - used with select statements, they will return some numeric values.

(1) countC) - function returns the number of rows that matches.
syntax - select count (column_name) from Table_name;
e.g. select count (MN) from practice ;

(2) Avg C) - function return the avg value of a numeric column.
syntax - select Avg (CN) from TN;
e.g. select Avg (id) from practice ;

(3) Sum C) - function return the total sum of numeric column.
syntax - select sum (CN) from TN;
e.g. select sum (salary) from practice ;

(4) min C) - return the min value of a numeric column.
syntax - select min (CN) from TN;
e.g. select min (company) from practice ;

(5) max C) - return the max value of a numeric column.
syntax - select max (CN) from TN;
e.g. select max (company) from Practice ;

(11) Comparision Operator

(a) = (equal) - used to show all records equals to condition.

syntax - select *from table_name where column_name = value ;

e.g - select * from practice where salary = 70000;

(b) < (less than) - show all records less than condition.

syntax - select *from table_name where CN < value ;

e.g - select * from practice where salary < 45000;

(c) > (greater than) - show all records greater than coudition.

syntax - (e.g) - select *from practice where salary > 50000;

(d) <= (eless than or equal to) -

e.g - select *from practice where salary <= 75020;

(e) >= (greater than or equal to) -

e.g - select *from practice where salary >= 10000;

(12) Like - clause is used to find a value to similar values or patter using wildcard operators like %A, %A%, A% etc

Jump

syntax - select *from table_name where CN Like Pattern;

(a) A% - search pattern starting with char 'A'.

↳ syntax - select *from practice where FN Like A% ;

e.g - Aniket

(b) %A - search pattern ending with char A :

syntax (e.g) - select *from tapractice where FN like %A ;

e.g - Manjusha

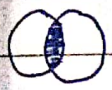
(c) %A% - search inbetween pattern

e.g - select * from practice where FN like %A%;

Manjusha
Pratiksha
Ankita

(13) Join - Joins are used to retrive data from multiple tables. It required two or more tables to performed JOIN.
- There are 5 different types of join -

(a) Inner Join - It returns all rows from multiple tables where two table contains common values in the join condition is met.



- Syntax -

select TN1.CN1, TN1.CN2, TN1.CN3, TN2.CN1, TN2.CN2, TN2.CN3
From TN1 a Inner join TN2 b On TN1.CN (Primary key) = TN2.CN (Foreign key)

- eg -

Select a.EID, a.FN, a.LN, b.DEGIS, b.CN from batchb2 a inner join velocity2 b on a.EID = b.Employeeid;

(b) Left join - returns all rows from the left-hand table specified in ON condition & only those rows from the other table where the joined fields are equal.

eg Select a.EID, a.FN, a.LN, b.DEGIS, b.CN from batchb2 a Left Join velocity2 b on a.EID = b.Employeeid;

(c) Right join - returns all rows from the right-hand table specified is ON cond^n & only those rows from the other table where the joined fields are equal.

eg - same example just replace left join to right join

(d) Full Join - returns all rows from the left-hand table & right-hand table.

eg - same e.g. just right full join

* SQL Constraints - used to specify rules for data in a table. Constraints can be used when the table is created with the create table statement.

- Different types of constraints :

1) NOT NULL - It forces a column to NOT NULL accept null values. Because, while creating a table, if value is not present, it take null value in CN.

eg - LN $varchar_2(20)$ not null

2) Unique - It ensures, all values in a column are different.

eg - EID INT Unique

3) Primary Key - used to uniquely identifies each record in a table. primary key must contain Unique values + NOT NULL.

primary key = unique + not null.

eg - Primary (EID)

4) Foreign key - is used to link two tables together. A foreign key is a field in one table that refers to the primary key in another table

eg - Create table batchb2(EID Int , FN $varchar_2(10)$ not null, LN $varchar_2(20)$ not null , MN Int , primary key (EID));

table 1

Create table velocity (Employeeid Int , CN $varchar_2(20)$, Desig $varchar_2(20)$, foreign key (Employeeid) reference batchb2(EID));

table 2

Computer Interviews Vol. Imp

classmate Date Page

* **Order by** - order by clause is used to sort the records in your result set.

- syntax - Select CN from TN order by CN ASC / DESC;

(ASC - Ascending order ↑ & DESC - descending order) ↓

- eg ① Select FN from vma order by FN ASC;
Select FN from vma order by FN DESC;
Select FN from vma order by FN;

② Select salary from vma order by salary Asc;
——————||—————— Desc;
select salary from vma order by salary.

* **Distinct** - used to return only distinct values. (different) or we can say return the unique values.

- It is used to remove duplicates. (How ?)

eg - select Distinct CN From TN;
select Distinct FN from vma;
select Distinct MN from vma;
Select Distinct *from vma;

@ Findout Second Highest Salary ?

↳ select max (salary) from vma where salary not in (select max (salary) from vma);

@ N^{th} Highest salary -

↳ Select min (salary) from (select distinct salary from vma order by salary desc) where rownum <= 2;