SQL(Structured Query Language)

-used to communicate with Database.

-Data?

It is distinct piece of information or collection of facts related to any entity.

-Database?

It is a container where all data is stored. It is basically a systematic collection od data which is suppose to store and manipulate the data stored.

-Operations that can be performed on the database are – Insertion, Updating, Creation, Deleting.

* DBMS(Database Management System)
* A technology to store and retrieve data with utmost efficiency along with appropriate security measures.
* Features are—Security, Authorisation, CURD operation.
* Types—1. Hierarchical DBMS (parent-child relation)

2. Network DBMS

3. Relational DBMS(define relation in form of table ex-SQL)

4. Object-oriented DBMS(data is stored in form of object ex- post SQL )

* Advantages of DBMS
* Improve Data Sharing i.e. it is possible to use same database for two different application .
* Improve Data Security.
* Improve Data Redundancy.
* Minimized Data Inconsistency.
* Better interaction with Users.
* Improved Decision making.
* Increased end user productivity.
* Disadvantages of DBMS
* Increased cost.
* Complexity of backup and recovery.
* Frequent updates.
* RDBMS(Relational Database Management System)
* A DBMS which organise and store data by following relational model is known as RDBMS
* Relational Model is designed by E.F Cord.
* In relational model all the data is stored in form of relations.
* Table—combination of rows and columns.
* According to E.F. Cord, data in relational model should be logically organised and stored in form of tables.

Columns—It is also known as field or attribute. It is used to represent property of entity or table.

Rows—Rows are also called as records or tuples. It is used to represent all property of individual entity.

Cell—Intersection of row and column is known as cell. It is the smallest unit of table which is used to store the data.

🡪 In RDBMS, data inserted can be validated in two ways—

1. Constraints

* Used to limit the type of data that can go into a table to maintain data accuracy and integrity of data inside the table.
* Types of Constraints are—

1. Not Null—It restrict the column from having a NULL value. Cannot be applied on table level.
2. Unique—Ensure that a field or column will only have unique values i.e. will not have duplicate data. Can be applied to column level or table level.
3. Primary Key—Combination of unique and not null. It is important to have unique value and not important to have null value.

**Characteristic of primary key—**

* Column assigned as primary key cannot accept null value
* A table cannot have two primary key

Note—it is not mandatory to have primary key in the table, but it is highly recommended.

**4. Foreign Key**

* It is used to relate two tables.
* It can be applied to column level and table level.

**Characteristic of foreign key--**

- A table in which foreign key is present , it is known as child table.

-A table in which actually it belongs is known as parent table.

-Foreign key must and should be defined as primary key in its parent table.

-A column assigned as a foreign key can accept duplicate and null value.

- We cannot insert a value in a column defined as foreign key which is not present in the parent table column.

|  |  |  |
| --- | --- | --- |
| Id | name | Dept id  (foreign key) |
| 111 | palak | 10 |
| 222 | jenny | 20 |

|  |  |  |
| --- | --- | --- |
| Dept id  (primary key) | address | Desg |
| 10 | mumbai | Software engg |
| 20 | mumbai | Software engg |

Child table parent table

**5. Check**

- it is used to restrict the value of a column between a range.

-its like a condition checking before saving data into a column.

-can be applied to column as well as table level.

Difference between primary key and foreign key

Primary key Foreign key

* Do not accept null and duplicate -do accept null and duplicate value

value

* Table level - table and column level
* Primary key uniquely - used to link two table
* We can have only one primary key in table -we can have multiple foreign key

**Data Type—**

* Data type selection is usually dictated by nature of data and by intended use.
* Pay close attention to expected use of attributes for sorting and retrieval purpose.

Types—

1. Numeric

* Bit 0 to 1
* Tiny int 0 to 255
* Small int -32,768 to 32,767
* Int
* Big int
* Decimal -10^38+1 to 10^38-1
* Numeric ‘’ ‘’
* Float
* real

1. String

* CHAR fixed length with max length of 8000 character
* VARCHAR variable length , max length 8000
* VARCHAR(max) variable length storage with provided max

character, not supported in sql

* TTEXT variable length storage with max size of 2GB data

Char datatype is of fixed memory allocation , in which memory is wasted

Varchar is variable memory allocation

1. Date and Time datatypes

* DATE stores date in format YYYY-MM-DD
* TIME HH:MI:SS
* DATETIME YYYY-MM-DD HH-MI-SS
* TIMESTAMP stores the number of seconds passed since the unix epoch
* YEAR stores year in 2-digit and 4-digit format

1. Miscellaneous datatype

-CLOB(character large object)

-BLOB(binary large object) Is use to store picture, audio & video

-XML

-JSON

**NOTE—while creating a table ,data type is mandatory to be assigned to the column but constraints are not mandatory but it is highly recommended**

**HISTORY**

-IBM was the first company to develop DBMS which follow Relational model and known as system arc.

-IBM developed a language to communicate with system arc or relational model. The language name was SEQUEL(Simple English Query Language)

-Due to its simplicity it was very popular at that time. In 1980, ANSI(American National Standard Institute ) acquired SEQUEL language and after doing some modification , it was then known as SQL and it was made standard language for RDBMS.

**NOTE-- is case insensitive**

**SQL is a language and MySql is an database application**

**SQL—Structured Query Language**

* A standardized programming language which is used for storing and managing data in databases.
* SQL was the first commercial language introduced for E.F Codd’s Relational model of database.
* With SQL, you can modify, add , update or delete rows of data , retrieve subsets of information from a database and many more
* Operations are written as statements.

**5 SQL Statement Categories—**

1. Data Definition Language(DDL) create, alter ,drop ,truncate and rename
2. Data Manipulation Language(DML)

Insert , update, delete

1. Data Control Language(DCL)
2. Transactional Control Language(TCL)
3. Data Query Language(DQL) select

* The SQL statement used to retrieve the data from database is known os DQL.
* Only one command to retrieve the data from database is:

Select

* We can retrieve data in three different ways:

1. Projection -- If we want to select some particular column from table then we use particular

SELECT column\_name from table\_name;

1. Selection -- if we want to select particular row from the table then we use selection

SELECT \*from table\_name;

1. Joins -- retrival of data from multiple table, at the same time, it is known as Joins.

SELECT \*from table\_name;

From is use to select the table from database and will put it for execution. We can pass table\_name as an argument.

Select is used to select columns from the table which is under execution and is also responsible to prepare the result set.

Select(\*) all the values .

Order of Execution—

1. From
2. select

* Clauses in SQL
* SQL statements consist of multiple clauses.
* This clause is a subprogram which accepts some arguments.

🡪 Where clause

- it is used to filter records.

- it is used to extract only those record that fulfil a specified condition.

- select insert

Syntax—

SELECT \*from table\_name

WHERE condition(column name=value);

Order of execution-

1. From table\_name
2. Where condition
3. Select

**Truncate v/s Delete**

**-**if we want to delete table and complete data, we **use** **truncate**

-If we want to delete a particular field of a table, we **use delete** command.

Note---we **use drop** to delete the structure of table and data also.

**EXPRESSION—**

-A statement which gives result or output is called expression.

C=a+b + is the operator, a and b are operands

* In sql operands are of two types –

1. Column
2. Literals -- actual value

select salary,salary+5000 from emp\_info;

salary and 5000 are the operands in which salary is column and 5000 is literal

literals are of three types—

1. Number
2. Character (+‘abc’) -it is case sensitive
3. Date (‘2019-08-08’)

12-08-19

**ALIAS—**

- Are used to give Temporary name to column or a table.

- It is often used to make column name readable

-->ALIAS COLUMN SYNTAX—

SELECT column\_name AS alias\_name FROM table\_name;

-->ALIAS TABLE SYNTAX—

SELECT column\_name(s) FROM table\_name AS alias\_name;

**DISTINCT—**

* It is used to return only distinct(different) values

SYNTAX--

SELECT DISTINCT column1, column 2,……. FROM table\_name;

* If more than one argument is passed to distinct, it will remove the combination of duplicate column.

**OPERATORS—**

* An opretors is reserved word or a character used primarily in an SQL statements.

TYPES—

* Arithmetic operator
* Relational/ comparison operator
* Logical operator
* Special operator

1. Additional operators—

+, -, \*, /, %

1. Relational operator—

=, !=, <>(check if two values are equal or not, if not equal then it returns true), > , < , >=, <=

1. Logical operators—

AND( allow the existence of multiple conditions in WHERE clause)

A& B are the two condition

Case 1: if both the condition are true , then output will be true,

Case 2: if A is true , then it will check the B condition. If it is False, then output will be false

Case 3: if A is false then it will not check the B condition , the output will be false

Case 4: if both A & B are false the result is false.

SYNTAX—

Select \* from table\_name where column\_name=value && column\_name=value;

OR( “ “ ),

Case 1: if the first condition is true then the result will be true without evaluating the condition.

Case 2: if the first condition is false then it will check the second condition and the result will depend on it.

SYNTAX—

SELECT \* FROM table\_name WHERE column\_name=value or column\_name=value;

If both and & or are used together then first and is executed and then or.

NOT( will alter the result)

SELECT \* FROM table\_name WHERE NOT column\_name=value;

If three opertors are used then or is ecxecuted first then and & then or.

4.Special Operators---

1. **IN** operator--- it is used to evaluate multiple values

Syntax--- SELECT \* FROM table\_name WHERE column\_name IN(list of values);

2. **NOT IN** ----it is multi value operator, it will exclude the multiple values specified

Syntax--- SELECT \* FROM table\_name WHERE column\_name NOT IN(list of values);

3.**BETWEEN** –it is used for searching based on range of values

Syntax—SELECT \* FROM table\_name WHERE column\_name BETWEEN lower\_limit and upper\_limit;

4.**NOT BETWEEN**

Syntax—SELECT \* FROM table\_name WHERE column\_name NOT BETWEEN lower\_limit and upper\_limit

5.**IS** – to check the null value. If record is null, it will give true.

And if it is not null, then it will give false

Syntax—SELECT \* FROM table\_name WHERE column\_name IS null;

6.**NOT NULL**

Syntax—SELECT \* FROM table\_name WHERE column\_name NOT IS null;

7.**LIKE**

-like operator is used to perform pattern matching.

-syntax SELECT \* FROM table\_name WHERE column\_name LIKE ‘pattern’;

Character are of two type—

1. Ordinary character. i.e Normal alphabets

2. Special character(which perform special operation)

1. % (percentage) matches 0 to ‘n’ character, used when we don’t know how many character are there

2. \_(underscore) matches exactly one character. Represent only one character of string

Syntax—SELECT \* FROM table\_name WHERE column\_name LIKE ‘%/\_’;

NOT LIKE--

It is used to select all the string which donot match with the given string

Syntax—SELECT \* FROM table\_name WHERE column\_name Not LIKE ‘%/\_’;

**FUNCTIONS----**

-block of code which perform certain task.

-Function attributes—

---Input argument, Function name, Return type

-SQL predefined function—

---Aggregate function/ multiple row function

---Scalar function/ single row function

--**AGGREGATE FUNCTION—**

-

**input**

FUNCTION

* In aggregate function we pass multiple inputs but the output is one

Type of aggregate function—

1. AVG()

Syntax—SELECT AVG(column\_name) from table\_name;

1. COUNT()
2. MAX()
3. MIN()
4. SUM()

--**SCALAR FUNCTION—**

-for every input we get corresponding result.

input output

16-08-19

Scalar functions—

Divide into—

1. Case manipulation single row function

--UPPER()

--LOWER()

--INITCAP() in a string first letter is capital.

-not supported in mysql. Only supported in oracle.

2. Character manipulation single row function

--LENGTH() ---gives the length of each value in a column.

--CONCAT() – it is used to concat two strings

Syntax-- SELECT CONCAT(column\_name1, column\_name)

--REVERSE()

--SUBSTR() --function use to extract a string from a given string. It excepts three argument.

Syntax--- SELECT SUBSTR (string from which we get new string , starting position(number) which specify starting position in a string, length specify the length of new string) from table\_name;

--INSTR() –it is used to check if a substring if present in a given string

or not. If a substring is not present in a given string, then the function will return 0.

Syntax—SELECT INSTR(string (it is a string in which substring has to be searched), substring) from table\_name;

--REPLACE() – it is used to replace a character or substring in a given string

Syntax ---SELECT REPALCE(string, substr1(the letter which has to be replaced), substr2(letter with which it is to be replaced) from table\_name;

**GROUPBY------**

* The group by statement group rows that have the same values into summary rows.
* The group by statement is often used with aggregate function(COUNT, MAX, MIN,SUM, AVG) to group the result-set by one or more column.

Syntax—

SELECT column\_name FROM table\_name WHERE condition GROUP BY column\_name;

**Note**---Group by clause is executed row by row. After the execution of group by, the records are group. Therefore, all the clause will execute after the execution of group by clause. Group by clause will execute where clause, group by clause and then select

**HAVING CLAUSE—**

* The having clause is often used with the GROUP BY clause to filter groups based on a specified condition.
* In HAVING clause we can use only an column i.e used in group by clause

Syntax—SELECT column\_name FROM table\_name WHERE condition GROUP BY column\_name HAVING condition

Order of execution--From ,where, group by, having, select

**NOTE—**HAVING clause executed after the execution of group by caluse. Therefore it check the given condition after grouping.

-since it executes after grouping, HAVING clause wil execute group by group.

- HAVING clause use column used by group by.

**ORDER BY---**

-order by clause is used to sort the result in ascending and descending order .

- by default order by sort the records in ascending order.

Syntax—

SELECT column1, column2..FROM table\_name ORDER BY column1, column2 ASC|DESC;

Order of execution---from, where, select, order by

* Order by clause always execute after execution of select
* Order by clause should always be the last clause to be written in syntax.
* Order by clause by default sort the data in ascending order.
* If the group by clause is used, then in order by clause we can use aggregate or multirow function or only the column name which is used in the group by clause.

**SUB QUERY---**

-a query written inside a query is known as sub-query

- it is also known as nested query or inner query.

- sub-query is also another way to fetch data from multiple tables.

-inner query is written in parenthesis.

- always inner query should be executed first.

-And the output of inner query is taken as input for outer query.

-And the final result is obtained from the outer query.

**\*\*\*when to use sub-query in where clause?**

1. if the given condition contain unknown value, then we use sub-query to obtain the unknown value.
2. If the column to be selected and condition to be filtered for two different tables, we can use sub-query.

Type—

1. Single row subquery –

We can use operators such as >, < >=, < =

-Syntax—SELECT column1,column2..FROM table\_name WHERE column1 **operators** (SELECT column FROM table\_name WHERE expr1=value);

1. Multi Row subquery –

We can use operators such as NOT, IN

Syntax—SELECT column1,column2..FROM table\_name WHERE column1 **operators** (SELECT column FROM table\_name WHERE expr1=value);

**JOINS—**

* If we want to fetch data from two tables, we use joins.
* Merging of more than one table horizontally is called join.
* Atleast no. of table should be two.

Type—

1. Cross join/ Cartesian join

* In cartesian join, a record from one table is merged with each and every record of other table. Or join two table records from a left table will be merged with each n every record of right table such joint is known as cartesian join.
* Syntax—SELECT column\_name-list FROM table\_name1 CROSS JOIN table\_name1;

|  |  |
| --- | --- |
| A1 | B1 |
| a | 10 |
| b | 20 |

|  |  |
| --- | --- |
| A2 | B2 |
| c | 30 |
| d | 40 |

Table 1

Table 2

T1.T2

|  |  |  |  |
| --- | --- | --- | --- |
| A1 | B1 | A2 | B2 |
| a | 10 | c | 30 |
| a | 10 | d | 40 |
| b | 20 | c | 30 |
| b | 20 | d | 40 |

9-9-19

2. Inner Join—

* Joining two tables with the help of join column is known as inner join.

|  |  |
| --- | --- |
| A1 | B1 |
| a | 10 |
| b | 20 |
| c | 50 |

|  |  |
| --- | --- |
| A2 | B2 |
| b | 30 |
| c | 40 |
| d | 60 |

Table 1

Table 2

T1.A1 = T2.A1

|  |  |  |  |
| --- | --- | --- | --- |
| A1 | B1 | A2 | B2 |
| b | 20 | b | 30 |
| c | 50 | c | 40 |

* Syntax -- SELECT \* FROM TABLE1 INNER JOIN on Table2 on condition.(mysql)
* SELECT \* FROM table1, table2 WHERE condition;(oracle)

3.Outer join

- Types of outer join

1. Left outer join

2. Right outer join

3. Full outer join

1. Left outer join

* Left outer join gives inner join + unmatched record of left table.
* Unmatched records—the records who do not have pair in opposite table w.r.t join.

|  |  |
| --- | --- |
| A1 | B1 |
| a | 10 |
| b | 20 |
| c | 50 |
| e | 70 |

|  |  |
| --- | --- |
| A2 | B2 |
| b | 30 |
| c | 40 |
| d | 60 |

Table 1

Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| A1 | B1 | A2 | B2 |
| b | 20 | b | 30 |
| c | 50 | c | 40 |
| a | 10 | null | null |
| e | 70 | null | null |

* Syntax—

SELECT \* FROM table1 LEFT OUTER JOIN table2 on condition;(mysql)

SELECT \* FROM table1, table2 Where tablw1.column\_name=table2.column\_name(+); (oracle)

1. Right outer join

* Right outer join gives inner join + unmatched record of right table.

|  |  |
| --- | --- |
| A1 | B1 |
| a | 10 |
| b | 20 |
| c | 50 |
| e | 70 |

|  |  |
| --- | --- |
| A2 | B2 |
| b | 30 |
| c | 40 |
| d | 60 |

Table 1

Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| A1 | B1 | A2 | B2 |
| b | 20 | b | 30 |
| c | 50 | c | 40 |
| Null | Null | D | 60 |
|  |  |  |  |

* Syntax—

SELECT \* FROM table1 RIGHT OUTER JOIN table2 on condition;(mysql)

SELECT \* FROM table1, table2 Where tablw1.column\_name=table2.column\_name(+); (oracle)

1. Full Outer join

* It gives inner join as well as all the records from both the tables who do not have any pair in common.

Syntax— SELECT \* FROM table1 FULL OUTER JOIN table2 on condition;(mysql) but this is not supported

**KEY IN RDBMS ---**

* Key Attributes-
* A key attribute is used to find an attribute using which we can uniquely determine a record in a table.
* Non- Key attributes-
* All the attributes except key attributes are non- key attributes.
* Prime key attributes-
* A key attributes which is chosen to be a main attribute to determine the record uniquely in the table.
* Non-prime key attribute-
* All the attributes except prime key attributes are non-prime attributes.
* Composite key-
* The combination of two or more attributes which determines the record uniquely is known as composite key.
* Foreign Key-
* Foreign key is used to relate two table.
* It can take null and duplicate values.

**FUNCTIONAL DEPENDENCY –**

* In a functional dependency a relation exist such that an attribute determine another attribute uniquely.
* Example--- R ->{ x, y }

x is determinant and is unique and y is dependent

**Types of functional dependency---**

1. **Total Functional Dependency**

* If all the attributes of a relation is determined by a key attribute is known as total functional dependency.
* For example— R -> { a, b, c, d } where a,b,c,d are the attributes

->if a is key attribute then all other are dependent on a .

(a)->b, (a)->c, (a)-> d

1. **Partial Functional Dependency**

* Relation is said to have partial dependency if-

1. If it consist of composite key attribute
2. There exist a dependency such that a attribute can be determined by another attribute that is part of composite key

* For example—R->{a, b, c, d}

->here a and b are the composite key and c and d are partially dependent on both a and b.

(ab)->c, d

(ab)->c

(ab)->d

1. **Transitive Functional Dependency-**

* Relational is said to be transitive functional dependency if there exist a relation such that an attribute is determined by non-key attributes which is in term determined by key attribute.
* For example—R-> {a, b, c, d }

a(key attribute)->b, b->c, a->c also

**NORMALIZATION—**

1. Dataredundancy i.e. same data is repeated many times

2. The side effect that occur during performing DML(insert, update, delete) operation is called anomaly.

Therefore to overcome this problem, we use normalization.

- the process of decomposing the table into smaller table in order to remove redundancy and anomaly by identifying dependencies is known as normalization.

Or

process of reducing table in normal form is known as normalization.

**Normal Form—**

* A state of table without table redundancies and anomalies is known as normal form.
* Types of normal form-

1. 1 Normal Form (1NF)
2. 2 Normal Form(2NF)
3. 3 Normal Form(3NF)
4. BCNF
5. 4 Normal Form(4NF)
6. **1NF—**

* A table is said to be in 1st normal form if it satisfy the following condition-

1. Table should not have duplicates rows
2. Every cell in the table should be single value(atomic value)
3. **2NF—**

* A table is said to be in 2nd normal form if it satisfy the following condition-

1. Table should follow 1NF

2. Table should not have partial functional dependency(composite key)

**Note**—if the table consist of partial functional dependency, the attributes which are responsible are removed from the table.

1. **3NF—**

* The table is said to be 3rd NF is following condition are satisfied—

1. It should follow 2NF

2. The table should not have transitive functional dependency

**Note—**1NF, 2 NF and 3NF are based on key attributes and functional dependency of relational schema.

**ER (Entity Relational) Diagram—**

* ER diagram describe the structure of database with the help of diagram which is known as ER diagram.
* An ER model is blueprint of database that can later be implemented as database.
* Component of ER Diagram—

1. Entity 🡪

2. Attribute 🡪 primary key 🡪

3. Relationship 🡪

🡪 Single line is used to connect entity and attribute.

Id name studying college\_id

name

student

college

email address

mobile no

one to many is college to student (1:N)

many to one is student to college (N:1)

one to one is student having Aadhar card , pan card. (1:1)

**Relationships—**

1. One to one
2. One to many
3. Many to one
4. Many to many

**VIEWS—**

* View are logical or virtual tables that can be created on the existing table.
* View do not occupy memory.
* Syntax—CREATE VIEW view\_name AS SELECT \* FROM table\_name;
* SELECT \* FROM view;

Table->view right click -create view

VIEW `ust`.`EmployeeView`

AS

SELECT \* FROM emp\_info where designation in ('tester', 'software developer'); 1st execute this

select \* from EmployeeView; and then execute this

**Note**—any DML operation performed on view the base table will be reflected with the changes.

**TCL(Transaction Control Language)—**

1. **Rollback**

In mysql we should set auto commit false and then perform the rollback.

-set autocommit=0;

1. Commit
2. Savepoint—

* Savepoint usually runs with rollback.
* Syntax—SAVEPOINT name;

Perform any operation

To rollback to the savepoint—rollback to savepointname;

**STORE PROCEDURES—**

**-**A stored procedure is a collection or group of SQL statements, stored in the database’s data dictionary and called from either a remoye program , another stored procedure or the command line

Or

* A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.
* Stored procedures are commonly called SP’s.
* Syntax—

DELIMITER $$

Create procedure <procedure Name>

As

Begin

<SQL Statement>;

End $$

DELIMITER;

ALTER

-TO add new column

ALTER TABLE table\_name ADD column\_name column-definition;

* To drop column

ALTER TABLE table\_name DROP COLUMN column\_name;

TRUNCATE

TRUNCATE table table\_name

RENAME

-to change the table name

RENAME TABLE table\_name to new\_table\_name

INSERT

INSERT INTO table\_name(column\_name, column\_name) values(values,values);

UPDATE

UPDATE table\_name set column\_name=value;

DELETE

DELETE from table\_name;