* **SQl** : Structured Query Language SQL is a database tool that is used to create and access the database to support software applications.
* **Database Management Systems (DBMS): -** A Special Software program that helps users creates and maintains a database.
  + Make it easy to manage large amounts of information.
  + Interacts with software applications
* **Database:** It is any collection of related information. Computer is great for storing databases.
* **DBMS allow you to perform the C.R.U.D operation and other administrative tasks**
* **C.R.U.D means C = Create, R = Read, U = Update and D = Delete**
* **Two Types of Databases:-**
* **Relational Databases (SQL):**
* Organize data into one or more tables
* Each table has columns and row
* A unique key identifies each row
* **Non-Relational (NoSQL/ Not just SQL):**
* Organize data is anything but a traditional table
* Key-value stores
* Documents(JSON, XML, etc)
* Graphs
* Flexible tables
* **Relational Database Management Systems (RDDBMS): -**
* Help Users Create and maintain a relational database
* E.g. – MySQL, Oracle, PostgreSQL, MariaDB, etc
* Structured Query Language (SQL)
  + Standardized language for interacting with RDBMS
  + Used to perform C.R.U.D operations as well as other administrative tasks (user management, security, backup, etc)
  + Used to define tables and structures
  + SQL code used on one RDBMS is not always portable to another without modification
* **Non-Relational Database Management Systems (NRDBMS):-**
* Help Users Create and maintain a non-relational database
* E.g. – MongoDB, DynamoDB, Apache Cassandra, Firebase, etc
* Implementation Specific
  + Any non-relational database falls under this category, so there’s no set language standard
  + Most NRDBMS will implement their own language for performing C.R.U.D and administrative operations on the database
* **Database Queries: -**
* Queries are requests made to the database management system for getting some specific information.
* Example: - A Google search is a query
* **SQL is actually a Hybrid language, it’s basically 3 types of languages in one**
* **Data Definition Language (DDL): -**
  + used to define the database structure such as tables
  + such as CREATE, ALTER and DELETE objects.
* **Data Control Language (DCL): -**
  + Used for controlling access to the data in the database
  + User and permissions management such as GRANT and REVOKE

**GRANT Command**: This command is used to provide database access to users other than the administrator.

**REVOKE command**: This command is used to provide database deny or remove access to database objects.

* **Data Manipulation Language (DML): -**
  + used to manipulate the data in records.
  + INSERT, UPDATE, and DELETE.
* **What is the difference between SQL and MySQL?**

|  |  |
| --- | --- |
|  | |
| * + **SQL** | * + **MySQL** |
| * + SQL is a standard language which stands for Structured Query Language based on the English language | * + MySQL is a database management system. |
| * + SQL is the core of the relational database which is used for accessing and managing database | * + MySQL is an RDMS (Relational Database Management System) such as SQL Server, Informix etc. |

**What do you mean by table and field in SQL?**

A table is an organized collection of data stored in the form of rows and columns. Columns can be categorized as vertical and rows as horizontal. The columns in a table are called **fields** while the rows can be referred to as **records**.

**Why do we use SQL constraints? Which constraints we can use while creating a database in SQL?**

Constraints are used to set the rules for all records in the table. If any constraints get violated then it can abort the action that caused it.

There are 5 major constraints are used in SQL, such as

* NOT NULL: That indicates that the column must have some value.
* UNIQUE: This constraint is used to ensure that each row and column has a unique value
* PRIMARY KEY: This constraint is used in association with NOT NULL and UNIQUE constraints such as on one or the combination of more than one column to identify the particular record with a unique identity.
* FOREIGN KEY: It is used to ensure the referential integrity of data in the table. It matches the value in one table with another using the PRIMARY KEY.
* CHECK: It ensures whether the value in columns fulfills the specified condition.
* **What are joins in SQL?**
* A JOIN clause is used to combine rows from two or more tables, based on a related column between them. There are 4 joins in SQL namely:

**Inner Join** : It is also known as SIMPLE JOIN which returns all rows from BOTH tables when it has at least one matching column

**Right Join** : This joins returns all rows from the RIGHT table and its matched rows from the LEFT table.

**Left Join** : This join returns all rows from the LEFT table and its matched rows from a RIGHT table.

**Full Join**: This joins returns all results when there is a match either in the RIGHT table or in the LEFT table.

**What is the difference between CHAR and VARCHAR2 datatype in SQL?**

Both Char and Varchar2 are used for characters datatype but varchar2 is used for character strings of variable length whereas Char is used for strings of fixed length. For example, char(10) can only store 10 characters and will not be able to store a string of any other length whereas varchar2(10) can store any length i.e 6,8,2 in this variable.

**What is the difference between DELETE and Truncate statements?**

|  |  |
| --- | --- |
| DELETE | Truncate |
| Delete command is used to delete a row in a table. | Truncate is used to delete all the rows from a table. |
| You can rollback data after using delete statement. | You cannot rollback data. |
| It is a DML command. | It is a DDL command. |
| It is slower than truncate statement. | It is faster. |

**What is a Unique key?**

Uniquely identifies a single row in the table.

Multiple values allowed per table.

Null values allowed.

**What is a Foreign key?**

Foreign key maintains referential integrity by enforcing a link between the data in two tables.

The foreign key in the child table references the primary key in the parent table.

The foreign key constraint prevents actions that would destroy links between the child and parent tables.

**What do you mean by data integrity?**

Data Integrity defines the accuracy as well as the consistency of the data stored in a database. It also defines integrity constraints to enforce business rules on the data when it is entered into an application or a database.

**What is the difference between clustered and non-clustered indexes?**

Answer: The differences between the two are as follows:

One table can have only one clustered index but multiple non-clustered indexes.

Clustered indexes can be read rapidly rather than non-clustered indexes.

Clustered indexes store data physically in the table or view whereas, non-clustered indexes do not store data in the table as it has separate structure from the data row.

**What do you mean by Denormalization?**

Denormalization refers to a technique which is used to access data from higher to lower forms of a database. It helps the database managers to increase the performance when we  combine data from various tables into a single table.

**What are Entities and Relationships?**

Entities:  A person, place, or thing in the real world about which data can be stored in a database. Tables store data that represents one type of entity. For example – A bank database has a customer table to store customer information. Customer table stores this information as a set of attributes (columns within the table) for each customer.

Relationships: Relation or links between entities that have something to do with each other. For example – The customer name is related to the customer account number and contact information, which might be in the same table. There can also be relationships between separate tables (for example, customer to accounts).

Or

The relationship can be defined as the connection between more than one table in the database.

There are 4 types of relationships:

One to One Relationship

Many to One Relationship

Many to Many Relationship

One to Many Relationship

**What is an Index ?**

An Index is a special structure associated with a table to speed up the performance of queries( allowing faster retrieval of records from the table) . The index can be created on one or more columns of a table.

**What is Normalization and what are the advantages of it?**

Normalization is the process of organizing data to avoid duplication and redundancy.

Some of the advantages are:

Better Database organization

More Tables with smaller rows

Efficient data access

Greater Flexibility for Queries

Quickly find the information

Allows easy modification

**Explain different types of Normalization.**

There are many successive levels of normalization. These are called normal forms. Each consecutive normal form depends on the previous one.The first three normal forms are usually adequate.

First Normal Form (1NF) – No repeating groups within rows means It removes all duplicate columns from the table.

Second Normal Form (2NF) –the relationship between tables using the primary key.

Third Normal Form (3NF) –  removes those columns which are not related through the primary key

**What are the properties of the transaction?**

Answer: Properties of the transaction are known as ACID properties. These are:

**Atomicity**: Ensures the completeness of all transactions performed. Checks whether every transaction is completed successfully or not. If not, then the transaction is aborted at the failure point and the previous transaction is rolled back to its initial state as changes are undone.

**Consistency**: Consistency ensures that the data must meet all the validation rules. In simple words,  you can say that your transaction never leaves the database without completing its state.

**Isolation**: Ensures that all transactions are performed independently and changes made by one transaction are not reflected on others.The main goal of isolation is concurrency control.

**Durability**:  Durability means that if a transaction has been committed, it will occur whatever may come in between such as power loss, crash or any sort of error.

**What do you mean by “Trigger” in SQL?**

Trigger in SQL is are a special type of stored procedures that are defined to execute automatically in place or after data modifications. It allows you to execute a batch of code when an insert, update or any other query is executed against a specific table.

**What is View in SQL?**

Answer: A View can be defined as a virtual table that contains rows and columns with fields from one or more tables.

**What are the different operators available in SQL?**

There are three operators available in SQL, namely:

Arithmetic Operators

Logical Operators

Comparison Operators

**Are NULL values same as that of zero or a blank space?**

A NULL value is not at all same as that of zero or a blank space. NULL value represents a value which is unavailable, unknown, assigned or not applicable whereas a zero is a number and blank space is a character.

**What is the difference between cross join and natural join?**

The cross join produces the cross product or Cartesian product of two tables whereas the natural join is based on all the columns having the same name and data types in both the tables.

**What is the need for group functions in SQL?**

Group functions work on the set of rows and returns one result per group. Some of the commonly used group functions are: AVG, COUNT, MAX, MIN, SUM, VARIANCE.

**What are transactions and their controls?**

Answer: A transaction can be defined as the sequence task that is performed on databases in a logical manner to gain certain results. Operations like Creating, updating, deleting records performed in the database come from transactions.

In simple words, we can say that a transaction means a group of SQL queries executed on database records.

There are 4 transaction controls such as

COMMIT: It is used to save all changes made through the transaction.

ROLLBACK: It is used to roll back the transaction. All changes made by the transaction are reverted back and the database remains as before.

SET TRANSACTION: Set the name of the transaction.

SAVEPOINT: It is used to set the point where the transaction is to be rolled back.

**What is the main difference between ‘BETWEEN’ and ‘IN’ condition operators?**

BETWEEN operator is used to display rows based on a range of values in a row whereas the IN condition operator is used to check for values contained in a specific set of values.

**What is the need of MERGE statement?**

This statement allows conditional update or insertion of data into a table. It performs an UPDATE if a row exists, or an INSERT if the row does not exist.

**What do you mean by recursive stored procedure?**

Recursive stored procedure refers to a stored procedure which calls by itself until it reaches some boundary condition. This recursive function or procedure helps the programmers to use the same set of code n number of times.

**What is CLAUSE in SQL?**

SQL clause helps to limit the result set by providing a condition to the query. A clause helps to filter the rows from the entire set of records.For example – WHERE, HAVING clause.

**What is the difference between ‘HAVING’ CLAUSE and a ‘WHERE’ CLAUSE?**

HAVING clause can be used only with SELECT statement. It is usually used in a GROUP BY clause and whenever GROUP BY is not used, HAVING behaves like a WHERE clause.

**How can you fetch common records from two tables?**

You can fetch common records from two tables using INTERSECT.

**What are the different set operators available in SQL?**

Some of the available set operators are – Union, Intersect or Minus operators.

**What are aggregate and scalar functions?**

Aggregate functions are used to evaluate mathematical calculation and returns a single value. These calculations are done from the columns in a table. For example- max(),count() are calculated with respect to numeric.

Scalar functions return a single value based on the input value. For example – UCASE(), NOW() are calculated with respect to string.

**What do you mean by Subquery?**

Answer: Query within another query is called as Subquery. A subquery is called inner query which returns output that is to be used by another query.

**What do you mean by Stored Procedures? How do we use it?**

Answer: A stored procedure is a collection of SQL statements that can be used as a function to access the database. We can create these stored procedures earlier before using it and can execute them wherever required by applying some conditional logic to it. Stored procedures are also used to reduce network traffic and improve performance.

**Define COMMIT?**

Answer: COMMIT saves all changes made by DML statements.

**How to select random rows from a table?**

Answer: Using a SAMPLE clause we can select random rows.

For Example, SELECT \* FROM table\_name SAMPLE(10);

**What is Referential Integrity?**

Answer: Set of rules that restrict the values of one or more columns of the tables based on the values of the primary key or unique key of the referenced table.

**Explain the difference between Rename and Alias?**

Answer: Rename is a permanent name given to a table or column whereas Alias is a temporary name given to a table or column.

**SQL Query For beginner level (ask for mostly interview)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ****WORKER\_ID**** | ****FIRST\_NAME**** | ****LAST\_NAME**** | ****SALARY**** | ****JOINING\_DATE**** | ****DEPARTMENT**** |
| 001 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 002 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 003 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 004 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 005 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| 006 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 007 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| 008 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |

|  |  |  |
| --- | --- | --- |
| ****WORKER\_REF\_ID**** | ****WORKER\_TITLE**** | ****AFFECTED\_FROM**** |
| 1 | Manager | 2016-02-20 00:00:00 |
| 2 | Executive | 2016-06-11 00:00:00 |
| 8 | Executive | 2016-06-11 00:00:00 |
| 5 | Manager | 2016-06-11 00:00:00 |
| 4 | Asst. Manager | 2016-06-11 00:00:00 |
| 7 | Executive | 2016-06-11 00:00:00 |
| 6 | Lead | 2016-06-11 00:00:00 |
| 3 | Lead | 2016-06-11 00:00:00 |

|  |  |  |
| --- | --- | --- |
| ****WORKER\_REF\_ID**** | ****BONUS\_DATE**** | ****BONUS\_AMOUNT**** |
| 1 | 2016-02-20 00:00:00 | 5000 |
| 2 | 2016-06-11 00:00:00 | 3000 |
| 3 | 2016-02-20 00:00:00 | 4000 |
| 1 | 2016-02-20 00:00:00 | 4500 |
| 2 | 2016-06-11 00:00:00 | 3500 |

CREATE TABLE Worker (

WORKER\_ID INT NOT NULL PRIMARY KEY,

FIRST\_NAME CHAR(25),

LAST\_NAME CHAR(25),

SALARY INT(15),

JOINING\_DATE DATETIME,

DEPARTMENT CHAR(25)

);

INSERT INTO Worker

(WORKER\_ID, FIRST\_NAME, LAST\_NAME, SALARY, JOINING\_DATE, DEPARTMENT) VALUES

(001, 'Monika', 'Arora', 100000, '14-02-20 09.00.00', 'HR'),

(002, 'Niharika', 'Verma', 80000, '14-06-11 09.00.00', 'Admin'),

(003, 'Vishal', 'Singhal', 300000, '14-02-20 09.00.00', 'HR'),

(004, 'Amitabh', 'Singh', 500000, '14-02-20 09.00.00', 'Admin'),

(005, 'Vivek', 'Bhati', 500000, '14-06-11 09.00.00', 'Admin'),

(006, 'Vipul', 'Diwan', 200000, '14-06-11 09.00.00', 'Account'),

(007, 'Satish', 'Kumar', 75000, '14-01-20 09.00.00', 'Account'),

(008, 'Geetika', 'Chauhan', 90000, '14-04-11 09.00.00', 'Admin');

CREATE TABLE Bonus (

WORKER\_REF\_ID INT,

BONUS\_AMOUNT INT(10),

BONUS\_DATE DATETIME,

FOREIGN KEY (WORKER\_REF\_ID)

REFERENCES Worker(WORKER\_ID)

ON DELETE CASCADE

);

INSERT INTO Bonus

(WORKER\_REF\_ID, BONUS\_AMOUNT, BONUS\_DATE) VALUES

(001, 5000, '16-02-20'),

(002, 3000, '16-06-11'),

(003, 4000, '16-02-20'),

(001, 4500, '16-02-20'),

(002, 3500, '16-06-11');

CREATE TABLE Title (

WORKER\_REF\_ID INT,

WORKER\_TITLE CHAR(25),

AFFECTED\_FROM DATETIME,

FOREIGN KEY (WORKER\_REF\_ID)

REFERENCES Worker(WORKER\_ID)

ON DELETE CASCADE

);

INSERT INTO Title

(WORKER\_REF\_ID, WORKER\_TITLE, AFFECTED\_FROM) VALUES

(001, 'Manager', '2016-02-20 00:00:00'),

(002, 'Executive', '2016-06-11 00:00:00'),

(008, 'Executive', '2016-06-11 00:00:00'),

(005, 'Manager', '2016-06-11 00:00:00'),

(004, 'Asst. Manager', '2016-06-11 00:00:00'),

(007, 'Executive', '2016-06-11 00:00:00'),

(006, 'Lead', '2016-06-11 00:00:00'),

(003, 'Lead', '2016-06-11 00:00:00');

**fetch “FIRST\_NAME” from Worker table using the alias name as <WORKER\_NAME>.**

Ans : select first\_name as worker\_name from worker;

**fetch “FIRST\_NAME” from Worker table in upper case**

Ans : select upper(first\_name) from worker;

**fetch unique values of DEPARTMENT from Worker table**

Ans : select distinct department from worker;

**print the first three characters of FIRST\_NAME from Worker table**

Ans :select substr(first\_name, 1 ,3) from worker;

**find the position of the alphabet (‘a’) in the first name column ‘Amitabh’ from Worker table.**

Ans : select instr(first\_name, 'a') from worker where first\_name = 'Amitabh';

**print the FIRST\_NAME from Worker table after removing white spaces from the right side.**

Ans: select rtrim(first\_name) from worker;

**fetches the unique values of DEPARTMENT from Worker table and prints its length.**

Ans: select distinct length(department) from worker;

**print the FIRST\_NAME from Worker table after replacing ‘a’ with ‘A’.**

Ans : select replace(first\_name,'a','A') from worker;

**print the FIRST\_NAME and LAST\_NAME from Worker table into a single column COMPLETE\_NAME. A space char should separate them.**

Ans : SELECT (FIRST\_NAME || ' ' ||LAST\_NAME) AS 'CUSTOMER NAME' FROM worker;

Or Select CONCAT(FIRST\_NAME, ' ', LAST\_NAME) AS 'COMPLETE\_NAME' from Worker;

**print all Worker details from the Worker table order by FIRST\_NAME Ascending.**

Ans : select \* from worker order by first\_name;

**print all Worker details from the Worker table order by FIRST\_NAME Ascending and DEPARTMENT Descending.**

Ans : select \* from worker order by first\_name asc , department desc;

**print details for Workers with the first name as “Vipul” and “Satish” from Worker table.**

Ans : select \* from worker where first\_name in ('Satish','vipul');

**Or exclude vipul and satish then**

Ans : select \* from worker where first\_name not in ('Satish','vipul');

**print details of the Workers who have joined in Feb’2014.**

Ans : select \* from worker where year(joining\_date) = 2014 and month(joining\_date) = 2;

**fetch the no. of workers for each department in the descending order.**

Ans : select count(worker\_id),department from worker group by department order by department desc;

**print details of the Workers who are also Managers.**

Ans : SELECT DISTINCT W.FIRST\_NAME, T.WORKER\_TITLE FROM Worker W

INNER JOIN Title T

ON W.WORKER\_ID = T.WORKER\_REF\_ID

AND T.WORKER\_TITLE in ('Manager');

**Join 3 table**

Ans : select \* from worker W

inner join title T ON W.WORKER\_ID = T.WORKER\_REF\_ID

inner join bonus b on b.WORKER\_REF\_ID = T.WORKER\_REF\_ID;

**fetch duplicate records having matching data in some fields of a table.**

Ans : select worker\_title, affected\_from, count(\*) from title

group by worker\_title, affected\_from

having count(\*) > 1;

**to show only odd rows from a table.**

Ans : Select \* from worker where (worker\_id % 2) = 1;

**to show only even rows from a table.**

Ans : Select \* from worker where (worker\_id % 2) = 0;

**to clone a new table from another table.**

Ans : CREATE TABLE new\_table AS SELECT \* FROM worker;

**to determine the nth (say n=5) highest salary from a table.**

Ans : SELECT Salary FROM Worker ORDER BY Salary DESC LIMIT n-1,1;

**to determine the nth highest salary without using TOP or limit method.**

Ans : SELECT Salary FROM Worker W1

WHERE n-1 = (SELECT COUNT( DISTINCT ( W2.Salary ) ) FROM Worker W2

WHERE W2.Salary >= W1.Salary);

**to fetch nth max salaries from a table.**

Ans : SELECT distinct Salary from worker a WHERE n >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;

**Or min salaries**

Ans : SELECT distinct Salary from worker a WHERE n >= (SELECT count(distinct Salary) from worker b WHERE a.Salary >= b.Salary) order by a.Salary desc;

**to fetch the list of employees with the same salary.**

Ans : Select distinct W.WORKER\_ID, W.FIRST\_NAME, W.Salary

from Worker W, Worker W1

where W.Salary = W1.Salary

and W.WORKER\_ID != W1.WORKER\_ID;

**to fetch the first 50% records from a table**

Ans :SELECT \* FROM WORKER

WHERE WORKER\_ID <= (SELECT count(WORKER\_ID)/2 from Worker);

**to show the last record from a table**

Ans : Select \* from Worker where WORKER\_ID = (SELECT max(WORKER\_ID) from Worker);

**Or first record**

Ans : Select \* from Worker where WORKER\_ID = (SELECT min(WORKER\_ID) from Worker);

**Join more than 2 table**

SELECT country.country\_name\_eng, city.city\_name, customer.customer\_name

FROM country

INNER JOIN city ON city.country\_id = country.id

INNER JOIN customer ON customer.city\_id = city.id;

**Display the total salary of each employee adding the Salary with Variable value.**

SELECT EmpId,

Salary+Variable as TotalSalary

FROM EmployeeSalary;

**Window Function**

Find average salary of employees for each department and order employees within a department by age.

SELECT Name, Age, Department, Salary,

AVERAGE(Salary) OVER( PARTITION BY Department ORDER BY Age) AS Avg\_Salary

FROM employee;

**Window functions** applies aggregate and ranking functions over a particular window (set of rows). OVER clause is used with window functions to define that window. OVER clause does two things :

Partitions rows into form set of rows. (PARTITION BY clause is used)

Orders rows within those partitions into a particular order. (ORDER BY clause is used)

**Aggregate Window Function** : Various aggregate functions such as SUM(), COUNT(), AVERAGE(), MAX(), MIN() applied over a particular window (set of rows) are called aggregate window functions.

**window partition (PARTITION BY)** - This clause is used to divide or breaks the rows into partitions.

**window ordering (ORDER BY)** - This clause is used to specify the order of the rows within a partition

**window frame (ROWS)** - A frame is the subset of the current partition in window functions.

SELECT SUM(profit) AS total\_profit FROM sales;

+--------------+

| total\_profit |

+--------------+

| 7535 |

+--------------+

SELECT country, SUM(profit) AS country\_profit

FROM sales

GROUP BY country

ORDER BY country;

+---------+----------------+

| country | country\_profit |

+---------+----------------+

| Finland | 1610 |

| India | 1350 |

| USA | 4575 |

+---------+----------------+

SELECT

year, country, product, profit,

SUM(profit) OVER() AS total\_profit,

SUM(profit) OVER(PARTITION BY country) AS country\_profit

FROM sales

ORDER BY country, year, product, profit;

+------+---------+------------+--------+--------------+----------------+

| year | country | product | profit | total\_profit | country\_profit |

+------+---------+------------+--------+--------------+----------------+

| 2000 | Finland | Computer | 1500 | 7535 | 1610 |

| 2000 | Finland | Phone | 100 | 7535 | 1610 |

| 2001 | Finland | Phone | 10 | 7535 | 1610 |

| 2000 | India | Calculator | 75 | 7535 | 1350 |

| 2000 | India | Calculator | 75 | 7535 | 1350 |

| 2000 | India | Computer | 1200 | 7535 | 1350 |

| 2000 | USA | Calculator | 75 | 7535 | 4575 |

| 2000 | USA | Computer | 1500 | 7535 | 4575 |

| 2001 | USA | Calculator | 50 | 7535 | 4575 |

| 2001 | USA | Computer | 1200 | 7535 | 4575 |

| 2001 | USA | Computer | 1500 | 7535 | 4575 |

| 2001 | USA | TV | 100 | 7535 | 4575 |

| 2001 | USA | TV | 150 | 7535 | 4575 |

+------+---------+------------+--------+--------------+----------------+

<https://www.w3schools.com/sql/default.asp>