**SQL Interview Questions**

**Q. What is partial dependency ?**

**Ans.** A Partial dependency is a dependency where A is functionally dependent on B

( A -> B), but there is some attribute on A that can be removed from A and yet the dependency stills holds. For instance if the relation existed.

**Q. What is candidate key, alternate key and composite key ?**

**Ans.** A ***candidate key*** is one that can identify each row of a table uniquely. Generally a candidate key becomes the primary key of the table. If the table has more than one candidate key, one of them will become the primary key, and the rest are called ***alternate keys***.

A key formed by combining at least two or more columns is called ***composite key***.

**Q. What is Normalization ?**

**Ans.** ***Normalization*** is a process for assigning attributes to entities. It reduces data redundancies and helps eliminate the data anomalies.

Normalization works through a series of stages called normal forms:

1. First normal form (1NF)
2. Second normal form (2NF)
3. Third normal form (3NF)
4. Boyce-Codd Normal Form (BCNF)

The highest level of normalization is not always desirable.

**Q. What is First normal form (1NF) ?**

**Ans.** The term first normal form (1NF) describes the tabular format in which:

* All the key attributes are defined.
* There are no repeating groups in the table.
* All attributes are dependent on the primary key.

**Q. What is Second Normal Form (2 NF) ?**

**Ans.**

* It is in 1NF and
* It includes no partial dependencies; that is, no attribute is dependent on only a portion of the primary key.
* It is still possible for a table in 2NF to exhibit transitive dependency; that is, one or more attributes may be functionally dependent on nonkey attributes.

**Q. What is Third Normal Form (3NF) ?**

**Ans.** Create a separate table with attributes in a transitive functional dependence relationship.

* A table is in 3NF if:
* It is in 2NF and
* It contains no transitive dependencies.

**Q. Explain Boyce-Codd Normal Form (BCNF) ?**

**Ans.** A table is in BCNF if every determinant in that table is a candidate key. If a table contains only one candidate key, 3NF and BCNF are equivalent.

* A table is in Boyce-Codd normal form (BCNF) if every determinant in the table is a candidate key.
* A determinant is any attribute whose value determines other values with a row.
* If a table contains only one candidate key, the 3NF and the BCNF are equivalent.
* BCNF is a special case of 3NF.

**Q. What is Denormalization?**

**Ans.**

* Normalization is only one of many database design goals.
* Normalized (decomposed) tables require additional processing, reducing system speed.
* Normalization purity is often difficult to sustain in the modern database environment.
* The conflict between design efficiency, information requirements, and processing speed are often resolved through compromises that include denormalization.

**Q. What is Primary Key constraints ?**

**Ans.** The Primary key is used to uniquely identify each row in a table. It can either be part of the actual record itself , or it can be an artificial field.

* A primary key can consist of one or more fields on a table.
* When multiple fields are used as a primary key, they are called a composite key.

***Syntax***:

column name datatype [CONSTRAINT constraint\_name] PRIMARY KEY

***Example:***

CREATE TABLE Persons

(

Person\_Id int NOT NULL PRIMARY KEY,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Address varchar(255),

City varchar(255)

);

**Q. What is Foreign Key constraints ?**

**Ans.** A foreign key means that values in one table must also appear in another table.The referenced table is called the parent table while the table with the foreign key is called the child table. The foreign key in the child table will generally reference a primary key in the parent table.

A foreign key can be defined in either a CREATE TABLE statement or an ALTER TABLE statement.

***Syntax***:

CREATE TABLE table\_name

(

column1 datatype null/not null,

column2 datatype null/not null,

CONSTRAINT fk\_column

FOREIGN KEY (column1, column2, ... column\_n)

REFERENCES parent\_table (column1, column2, ... column\_n)

);

**Q. What is a Check constraint ?**

**Ans.** A Check constraint allows you to specify a condition on each row in a table.

* A Check constraint can be defined in either a CREATE TABLE statement or an ALTER TABLE statement.
* A check constraint can NOT be defined on a VIEW.
* The check constraint defined on a table must refer to only columns in that table. It can not refer to columns in other tables.
* A check constraint can NOT include a SUBQUERY.

***Syntax***:

CREATE TABLE table\_name

(

column1 datatype null/not null,

column2 datatype null/not null,

CONSTRAINT constraint\_name CHECK (column\_name condition) [DISABLE]

);

**Q. What is Unique constraints ?**

**Ans.** Unique constraints are like alternative primary key constraints. A unique constraint defines a column, or series of columns, that must be unique in value. You can have a number of unique constraints defined and the columns can have NULL values in them, unlike a column that belongs to a primary key constraint. If you need to add unique key constraints to a table after the fact, simply use the alter table command.

***Syntax***:

CREATE TABLE table\_name

(

column1 datatype Unique,

column1 datatype,

column1 datatype

);

**Q. What is difference between primary key and unique key?**

**Ans.** Difference between Primary Key & Unique Key

|  |  |
| --- | --- |
| **Primary Key** | **Unique Key** |
| Primary Key can't accept null values. | Unique key can accept only one null value. |
| By default, Primary key is clustered index and data in the database table is physically organized in the sequence of clustered index. | By default, Unique key is a unique non-clustered index. |
| We can have only one Primary key in a table. | We can have more than one unique key in a table. |
| Primary key can be made foreign key into another table. | In SQL Server, Unique key can be made foreign key into another table. |

**Q. Explain NOT NULL constraint ?**

**Ans.** The NOT NULL constraint allows a column not to accept NULL values. This specify that NOT NULL Constraint enforce the field to accept a value. Simply, neither you insert or update a new record without adding a value to this field.

***Syntax***:

CREATE TABLE table\_name

(

column1 datatype NOT NULL,

column1 datatype NULL/NOT NULL,

column1 datatype

);

**Q. What is DEFAULT constraint ?**

**Ans.** The DEFAULT constraint provides a default value to a column when the INSERT INTO statement does not provide a specific value.

***Syntax***:

CREATE TABLE table\_name

(

column1 datatype Unique,

column1 datatype,

column1 datatype DEFAULT default\_vaule

);

**Q. What is Table in SQL ?**

**Ans.** In relational database systems (DBS) data are represented using tables (relations). A query issued against the DBS also results in a table.

* A table is uniquely identified by its name and consists of rows that contain the stored information,each row containing exactly one tuple (or record).
* A table can have one or more columns.
* A column is made up of a column name and a data type, and it describes an attribute of the tuples.
* The structure of a table, also called relation schema, thus is defined by its attributes.
* The type of information to be stored in a table is defined by the data types of the attributes at table creation time.

**Q. What are the Basic data types in SQL ?**

**Ans.**

***char(n):*** Fixed-length character data (string), n characters long. The maximum size for n is 255 bytes (2000 in Oracle8). Note that a string of type char is always padded on right with blanks to full length of n. (+ can be memory consuming).

Example: char(40)

***varchar2(n):*** Variable-length character string. The maximum size for n is 2000 (4000 in Oracle8). Only the bytes used for a string require storage. Example: varchar2(80)

***number(o, d):*** Numeric data type for integers and reals. o = overall number of digits, d = number of digits to the right of the decimal point.

Maximum values: o =38, d= ??84 to +127. Examples: number(8), number(5,2)

Note that, e.g., number(5,2) cannot contain anything larger than 999.99 without resulting in an error. Data types derived from number are int[eger], dec[imal], small int and real.

***date:*** Date data type for storing date and time.

The default format for a date is: DD-MMM-YY. Examples: '13-OCT-94', 07-JAN-98'

***long:*** Character data up to a length of 2GB. Only one long column is allowed per table.

**Q. What is the difference between Varchar and Varchar2 ?**

**Ans.** The difference between Varchar and Varchar2 is both are variable length but only 2000 bytes of character of data can be stored in Varchar where as 4000 bytes of character of data can be store in varchar2.

**Q. What is the difference between LONG and LONG RAW ?**

**Ans.** We can use the LONG datatype to store variable-length character strings. The LONG datatype is like the VARCHAR2 datatype, except that the maximum length of a LONG value is 32760 bytes.

WE can use the LONG RAW datatype to store binary data (or) byte strings. LONG RAW data is like LONG data, except that LONG RAW data is not interpreted by PL/SQL. The maximum length of a LONG RAW value is 32760 bytes.

**Q. Explain about CREATE TABLE Statement ?**

**Ans.** The SQL command for creating an empty table has the following form:

***Syntax:***

CREATE TABLE <table\_name>

(

<column 1> <data type> [not null] [unique] [<column constraint>],

. . . . . . . . .

<column n> <data type> [not null] [unique] [<column constraint>],

[<table constraint(s)>]

);

* For each column, a name and a data type must be specified and the column name must be unique within the table de\_nition.
* Column definitions are separated by colons.
* There is no difference between names in lowercase letters and names in uppercase letters. In fact, the only place where upper and lower case letters matter are strings comparisons.
* A not null constraint is directly specified after the datatype of the column and the constraint requires defined attribute values for that column, different from null.
* The keyword unique specifies that no two tuples can have the same attribute value for this column.
* Unless the condition not null is also speci\_ed for this column, the attribute value null is allowed and two tuples having the attribute value null for this column do not violate the constraint.

***Example:***

CREATE TABLE Person

(

name VARCHAR(30),

social-security-number number(10),

age number,

city VARCHAR(30),

gender number(1),

Birthdate DATE

);

**Q. How to Create a Table from an Existing Table?**

**Ans.** The most common way to create a table is with the CREATE TABLE command. However, some database management systems provide an alternative method of creating tables, using the format and data of an existing table.

This method is useful when you want to select the data out of a table for temporary modification. It can also be useful when you have to create a table similar to the existing table and fill it with similar data.

***Syntax***:

*CREATE TABLE* new\_table\_name

(

field1, field2, field3) AS (SELECT field1, field2, field3

FROM old\_table\_name <WHERE...>

);

This syntax allows you to create a new table with the same data types as those of the fields that are selected from the old table. It also allows you to rename the fields in the new table by giving them new names.

**Q. Explain INSERT Statement in SQL ?**

**Ans.** The insert statement will insert data into a defined table in the database. Because it is usually easier to open the table in SQL. While inserting a row, if you are adding value for all the columns of the table you need not specify the column's name in the sql query. But you need to make sure the order of the values is in the same order as the columns in the table.

***Syntax***:

INSERT INTO <table> (<column i, . . . , column j>)

VALUES (<value i, . . . , value j>);

**Q. Explain UPDATE Statement in SQL ?**

**Ans.** This statement is used to change values of existing data in the database.

***Syntax***:

UPDATE table\_name

SET columnname1 = value1 [, columname2 = value2]...

WHERE search\_condition;

This statement checks the WHERE clause first. For all records in the given table in which the WHERE clause evaluates to TRUE, the corresponding value is updated.

**Q. Explain DELETE Statement in SQL ?**

**Ans.** The delete statement is used to delete data from a table. Be careful when using this command, it is easy to delete more data than intended.

***Syntax***:

DELETE FROM table\_name [WHERE condition];

If the where clause is omitted, all rows are deleted from the table. Depending on the use of the DELETE statement's WHERE clause, SQL can do the following:

* Delete single rows
* Delete multiple rows
* Delete all rows
* Delete no rows

**Q. Explain ALTER TABLE statement in SQL ?**

**Ans.** The ALTER TABLE statement enables the database to change the structure of a table after it has been created. It is possible to modify the structure of a table even if rows have already been inserted into this table. The ALTER TABLE command can do the below things:

* Add a column to an existing table
* Modify a column that already exists

***Syntax***:

ALTER TABLE table\_name <ADD column\_name data\_type |

MODIFY column\_name data\_type>

**Q. Explain DROP TABLE statement in SQL ?**

**Ans.** The drop statement is used to remove a table from the database.Drop is also used to drop views and indexes and table data is removed from the database.

***Syntax***:

DROP TABLE table\_name;

**Q. Explain TRUNCATE TABLE in SQL ?**

**Ans.** The TRUNCATE TABLE command is used to delete complete data from an existing table.

***Syntax***:

TRUNCATE TABLE table\_name;

**Note**: Truncate will remove the data permanently we cannot rollback the deleted data.

**Q. What is the difference between DELETE and TRUNCATE in SQL ?**

**Ans.** Rollback is possible after DELETE but TRUNCATE remove the table permanently and can’t rollback. Truncate will remove the data permanently we cannot rollback the deleted data.

**Q. Explain DISTINCT clause in SQL ?**

**Ans.** The DISTINCT clause allows you to remove duplicates from the result set.

***Syntax***:

SELECT DISTINCT column(n) FROM table\_name WHERE [condition]

ORDER BY column;

**Q. Explain WHERE Clause in SQL ?**

**Ans.** The WHERE Clause is used when you want to retrieve specific information from a table excluding other irrelevant data. Table name is the place holder for the name of the table, and condition is the place holder for the where clause. Not only is the where clause optional, but the where clause can contain multiple conditions by using the AND keyword.

***Syntax***:

SELECT col1, col2... FROM table\_ name WHERE search\_condition;

**Q. Explain ORDER BY Clause in SQL ?**

**Ans.** The SQL ORDER BY clause is used to sort the data in ascending or descending order, based on one or more columns. Some database sorts query results in ascending order by default.

***Syntax***:

SELECT [distinct] <column(s)>

from table\_name [ WHERE <condition> ]

[ ORDER BY <column(s) [ASC|DESC]> ]

**Q. Explain GROUP BY Clause in SQL ?**

**Ans.** The GROUP BY clause can be used in a SELECT statement to collect data across multiple records and group the results by one or more columns.

***Syntax***:

SELECT column(n) FROM table\_name

WHERE [ conditions ]

GROUP BY column(n)

ORDER BY column(n)

The GROUP BY clause must before ORDER BY clause if ORDER BY clause exist.

**Q. Explain HAVING Clause in SQL ?**

**Ans.** The HAVING clause is used in combination with the GROUP BY clause. It can be used in a SELECT statement to filter the records that a GROUP BY returns.

***Syntax***:

SELECT column(n) FROM table\_name

WHERE [ conditions ]

GROUP BY column(n) HAVING conditions ORDER BY column(n)

When WHERE, GROUP BY and HAVING clauses are used together in a SELECT statement, the WHERE clause is processed first, then the rows that are returned after the WHERE clause is executed are grouped based on the GROUP BY clause.

**Q. What is the difference between HAVING clause and WHERE clause ?**

**Ans.**

* Having Clause is basically used only with the GROUP BY function in a query.
* WHERE Clause is applied to each row before they are part of the GROUP BY function in a query.