Q1: What are the various constraints in SQL? Exaplin any five.

Ans: Constraints in SQL Server are predefined rules and restrictions that are enforced in a single column or multiple columns, regarding the values allowed in the columns, to maintain the integrity, accuracy, and reliability of that column’s data. In other words, if the inserted data meets the constraint rule, it will be inserted successfully. If the inserted data violates the defined constraint, the insert operation will be aborted.

There are six main constraints that are :

**NOT NULL:**

A NOT NULL constraint in SQL is used to prevent inserting NULL values into the specified column, considering it as a not accepted value for that column. This means that you should provide a valid SQL NOT NULL value to that column in the INSERT or UPDATE statements, as the column will always contain data.

## UNIQUE Constraints:

## The UNIQUE constraint in SQL is used to ensure that no duplicate values will be inserted into a specific column or combination of columns that are participating in the UNIQUE constraint and not part of the PRIMARY KEY.

## PRIMARY KEY Constraint:

## The SQL PRIMARY KEY constraint combines between the UNIQUE and SQL NOT NULL constraints, where the column or set of columns that are participating in the PRIMARY KEY cannot accept a NULL value. If the PRIMARY KEY is defined in multiple columns, you can insert duplicate values on each column individually, but the combination values of all PRIMARY KEY columns must be unique.

## Foreign Key Constraint:

## FOREIGN KEY is used to relate two tables. FOREIGN KEY constraint is also used to restrict actions that would destroy links between tables.

**CHECK CONSTRAINT:**

Check constraint is used to restrict the value of a column between a range. It performs check on the values, before storing them into the database. Its like condition checking before saving data into a column.

Q2: What is Pattern matching in SQL and how it is done?

Ans: SQL pattern matching allows you to search for patterns in data if you don't know the exact word or phrase you are seeking. This kind of [SQL](https://www.lifewire.com/sql-fundamentals-1019780) query uses [wildcard](https://www.lifewire.com/searching-for-part-of-word-in-google-3482306) characters to match a pattern, rather than specifying it exactly. For example, you can use the wildcard "C%" to match any string beginning with a capital C.

#### LIKE Operator

The LIKE operator provides standard pattern matching in SQL that is always used after a WHERE clause. It matches any pattern based on some conditions provided using the wildcard characters.

Some of the commonly used wildcard characters in MySQL are as follows:

* '%' represents zero or more characters.
* '\_' represents exactly 1 character.

Now, if we use the wildcard characters D%s, it will search for all such patterns that have a D character, in the beginning, a s character at the end, and 0 or any number of characters in between them.

Q3: What is a checkpoint and when does it occur?

Ans: **Database Checkpoint** process is one of the most important part of SQL Server Architecture. Before you read this article, you need to understand the significance of the transaction log file.

I would recommend reading the article “[What is SQL Server Transaction Log](https://dbtut.com/index.php/2018/05/31/what-is-sql-server-transaction-log/)“.

I wanted to write this article because many people have question marks about the working principle of sql server.

You might think this article is a little theoretical. But I want to insist on reading anyone who wants to deepen on SQL Server. First you should understand SQL Server write ahead logging mechanism to understand checkpoint.

**1. Automatic —**This type of checkpoints occur behind the scenes and depend on the recovery interval server configurations. The value is measured in minutes, and the default value is 1 minute (cannot be set lower). The checkpoint will complete in the time that minimizes the impact to performance.

Under the SIMPLE recovery model, an automatic checkpoint is also triggered when the transaction log is 70% full.

**2. Indirect —**This type of checkpoints also occur behind the scenes according to the user-specified database recovery time settings. Starting from [SQL Server 2016 CTP2](https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2016) the default value for this type of checkpoint is 1 minute. It means that a database will use indirect checkpoints. For older SQL Server versions the default is 0. This means that a database will use automatic checkpoints, whose frequency depends on the recovery interval setting of the SQL Server instance. Microsoft recommends 1 minute for most systems.

**3. Manual —**Occurs while executing the T-SQL CHECKPOINT command.

**checkpoint\_duration** is an integer used to define the amount of time in which a checkpoint should complete. This parameter also governs how many resources are assigned to the checkpoint operation. If the parameter is not specified, the checkpoint will complete in the time that minimizes impact to performance.

**4. Internal —** Some SQL Server operations issue this type of checkpoints to ensure that disk images match the current transaction log state. These are checkpoints that are performed  when a certain operation takes place:

* A data file is added or removed
* A database shutdown occurs (for whatever reason)
* A backup or database snapshot is created
* A DBCC command is run that creates a hidden database snapshot (or e.g. DBCC\_CHECKDB, DBCC\_CHECKTABLE).

Q4: What is E-R model?

Ans: Database is absolutely an integral part of software systems. To fully utilize ER Diagram in database engineering guarantees you to produce high-quality database design to use in database creation, management, and maintenance. An ER model also provides a means for communication.

Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: **The major entities within the system scope**, and the **inter-relationships among these entities**.

And that's why it's called "Entity" "Relationship" diagram (ERD)!

When we talk about entities in ERD, very often we are referring to business objects such as people/roles (e.g. Student), tangible business objects (e.g. Product), intangible business objects (e.g. Log), etc. "Relationship" is about how these entities relate to each other within the system.

## When to draw ER Diagrams?

* **Database design** - Depending on the scale of change, it can be risky to alter a database structure directly in a DBMS. To avoid ruining the data in a production database, it is important to plan out the changes carefully. ERD is a tool that helps. By drawing ER diagrams to visualize database design ideas, you have a chance to identify the mistakes and design flaws, and to make corrections before executing the changes in the database.
* **Database debugging** - To debug database issues can be challenging, especially when the database contains many tables, which require writing complex SQL in getting the information you need. By visualizing a database schema with an ERD, you have a full picture of the entire database schema. You can easily locate entities, view their attributes and identify the relationships they have with others. All these allow you to analyze an existing database and to reveal database problems easier.
* **Database creation and patching** - Visual Paradigm, an ERD tool, supports a database generation tool that can automate the database creation and patching process by means of ER diagrams. So, with this ER Diagram tool, your ER design is no longer just a static diagram but a mirror that reflects truly the physical database structure.
* **Aid in requirements gathering** - Determine the requirements of an information system by drawing a conceptual ERD that depicts the high-level business objects of the system. Such an initial model can also be evolved into a physical database model that aids the creation of a relational database, or aids in the creation of process maps and data flow modes.

Q5: What is denormalization in DBMS?

Ans: Denormalization is an approach to speeding up read-oriented data retrieval performance in a [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database), where the database administrator selectively adds back specific instances of redundant [data](https://searchdatamanagement.techtarget.com/definition/data) after the [data structure](https://searchsqlserver.techtarget.com/definition/data-structure) has been [normalized](https://searchsqlserver.techtarget.com/definition/normalization). A denormalized database should not be confused with a [database](https://searchsqlserver.techtarget.com/definition/database) that has never been normalized.

Using normalization in SQL, a database will store different but related types of data in separate logical [tables](https://whatis.techtarget.com/definition/table), called relations. When a [query](https://searchsqlserver.techtarget.com/definition/query) combines data from multiple tables into a single result table, it is called a join. The performance of such a join in the face of complex queries is often the occasion for the administrator to explore the denormalization alternative.

## Techopedia explains Denormalization

In many cases, denormalization involves creating separate tables or structures so that queries on one piece of information will not affect any other information tied to it. For instance, where more global data variables such as customer names are tied together with single purchases in a purchase history, a database administrator will want to make sure that work done on an item purchased will not incorrectly affect the entire customer account. Therefore, database handlers will separate the two pieces of information, sometimes with redundant data, so that they can be worked on separately.

Where denormalization comes in is that adding redundant data allows for more sophisticated search results. Some examples that are typically given to explain this include situations where database handlers want to find prior addresses, purchase histories, or anything else about a customer or client that doesn’t address the specific present state of that account. This is where having redundant data can allow databases to give different results based on exactly what the user is asking for. Again, having this redundant data can also improve performance based on the specific ways that a database searches for a particular item. Challenges involved in denormalization include documenting the process carefully to avoid some kinds of anomalies that can occur as a result of data mismatch.

Q6: What is normalization in DBMS?

Ans: **Normalization** is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. There are two primary advantages of having a highly normalized data schema:

1. **Increased consistency**. Information is stored in one place and one place only, reducing the possibility of inconsistent data.
2. **Easier object-to-data mapping**. Highly-normalized data schemas in general are closer conceptually to object-oriented schemas because the object-oriented goals of promoting high cohesion and loose coupling between classes results in similar solutions (at least from a data point of view).

You typically want to have highly normalized operational data stores (ODSs) and data warehouses (DWs).

The primary disadvantage of normalization is slower reporting performance. You will want to have a [denormalized](http://agiledata.org/essays/dataNormalization.html" \l "Denormalization) schema to support reporting, particularly in data marts.

Here are the most commonly used normal forms:

## First normal form (1NF)

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

## Second normal form (2NF)

A table is said to be in 2NF if both the following conditions hold:

* Table is in 1NF (First normal form)
* No non-prime attribute is dependent on the proper subset of any candidate key of table.

An attribute that is not part of any candidate key is known as non-prime attribute.

## Third Normal form (3NF)

A table design is said to be in 3NF if both the following conditions hold:

* Table must be in 2NF
* [Transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) of non-prime attribute on any super key should be removed.

An attribute that is not part of any [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:

* X is a [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) of table
* Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as prime attribute.

## Boyce Codd normal form (BCNF)

It is an advance version of 3NF that’s why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every [functional dependency](https://beginnersbook.com/2015/04/functional-dependency-in-dbms/) X->Y, X should be the super key of the table.