

**Difference between SQL Server and PostgreSQL**

Both of these being types of SQL a question may arise about what is the difference between both SQL Server vs PostgreSQL. Microsoft SQL server is a database management and analysis system which is mainly used for e-commerce, line of business and different data warehousing solutions. PostgreSQL, on the other hand, is an advanced object-relational database management system which provides support to the extended subset of SQL standards including different transactions, foreign keys, subqueries, triggers, and different user-defined types and functions.

**Head To Head Comparisons between SQL Server and PostgreSQL (Infographics)**

Below is the top 8 difference between SQL Server vs PostgreSQL

**Key Difference**

**Between SQL Server and PostgreSQL**

Both are popular choices in the market; let us discuss some of the major Difference:

* **CSV support:** Postgres is on top of the game when it comes to CSV support. It provides different commands like ‘copy to’ and ‘copy from’ which help in the fast processing of data. It also provides helpful error messages. If there is a slight issue with import it will throw an error and stop the import then and there. SQL server, on the other hand, does not support either importing the data or exporting it.
* **Cross-platform:** In today’s world for a language or any program to be cross-platform is really very important. With the advent of technology and IT industry, it has become very important to be accessible over Linux and UNIX as they are open source systems. SQL server is a Microsoft product is vendor locked and can be run only on Microsoft systems. Postgres can be used on Linux, BSD, Solaris and also Windows.
* **Procedural language features:** PL/PGSQL is a native procedural language provided by Postgres which has different modern features. It supports JSON data type and hence has ultimate power and loads of flexibility included in a single package. In addition to this, a user can easily use Python, Perl, R, Java, PHP with SQL as they are supported as procedural languages in Postgres. MS SQL server also provides procedural language support as an inbuilt feature. But this feature is a bit messy, slow and has poor features. It also has small errors with different bugs always occurring.
* **Regular expressions:** Postgres provides a huge number of regex as a basis for analytical work. SQL server, on the other hand, has like, substring, patindex which are not as good when compared to expressions provided by Postgres.

**SQL Server vs PostgreSQL Comparison Table**

Here are some of the Comparison:

|  |  |  |
| --- | --- | --- |
| **The Basis Of Comparison** | **SQL Server** | **PostgreSQL** |
| **Basic Difference** | SQL server is a database management system which is mainly used for e-commerce and providing different data warehousing solutions. | PostgreSQL is an advanced version of SQL which provides support to different functions of SQL like foreign keys, subqueries, triggers, and different user-defined types and functions. |
| **Updateable Views** | Views can be updatable even if 2 table views are updated. If the tables have different keys and the update statement does not involve more than one table then it will be updated automatically. The user can also make use of triggers to update complex views. | Views in PostgreSQL can be updated but not automatically unlike SQL server. The user must write rules against different views to update them. Also, complex views can be easily created. |
| **Computed Columns** | SQL server does provide computed columns but views are preferred over computed columns. Computed columns have a very limited use as they are not capable of holding different roll-ups. | PostgreSQL does not provide computed columns. PostgreSQL, on the other hand, has functional indexes which work just as a view. |
| **Replication** | SQL server can replicate all sorts of data. This can be log shipping, mirroring, snapshot, and transactional and merge etc. and can even have non-SQL Server windows-based subscribers. | Replication in Postgres is in the form of reports and is supposed to be least polished of the bunch. Although there are different third-party options to choose from the ones that are free and not free. PostgreSQL 8.4 or a higher slated version can have built-in replication feature. |
| **Support stored procedures and stored functions in different languages** | SQL server does support this feature. It can be done with any language which complies with CLR like VB, C#, Python, etc. TO get this done successfully user must first compile the code into all first. | Here there is no need to create a dull first. A user who has created the code can easily see what the code is doing. The server which is downside must host the language the environment is using. |
| **Dynamic actions in SQL** | SQL server does not support this feature. But instead of this user can use the stored procedure and call these from select statements so it is much more limiting than PostgreSQL. | PostgreSQL does provide this feature and just by using select statements a user can perform really all operations and retrieve and do all other jobs easily. |
| **Materialized Views** | Yes, it provides the facilities to run materialized views. The functioning though varies depending on where the query is being run. It can be SQL Express, Workgroup, etc. | Postgres does not provide facility to run materialized views. Instead of this, they have a module called mat views which helps in rebuilding any materialized view. |
| **Case sensitivity** | By default SQL server is considered to be case insensitive but if a user wants to change the same they can do it by going down to the column level. | By default, PostgreSQL is case sensitive and it is difficult to make it insensitive. Changes can be made in it but they are not exposed and are not ANSI compliant hence making it a delirious job to use it on MS Access, PHP Gallery, etc. where SQL is regarded to be case insensitive. |

**Conclusion**

In this SQL Server vs PostgreSQL article, we have seen Both SQL Server vs PostgreSQL are database management tools. They help in managing all data properly and efficiently. But when it comes to different features PostgreSQL is always at the upper hand. It is an advanced version of SQL and hence provides many additional features.  All these features are for free, unlike SQL server. Also, it is cross-platform and can be used with any operating system.

<https://www.educba.com/sql-server-vs-postgresql/>

# **PostgreSQL vs. SQL Server (MSSQL) - Extremely Detailed Comparison**

Pankil Shah July 30, 2020



## ****Overview****

PostgreSQL and SQL Server (or MSSQL) are two widely used relational databases. Although they share a number of core traits, there are major differences between them. In this article, we provide a detailed rundown of the similarities and differences between PostgreSQL and SQL Server. Among the most significant distinctions is that PostgreSQL is open source, while SQL Server is owned and licensed by Microsoft. In addition, you will learn about differences between the two systems when it comes to licensing and cost, ease of use, SQL syntax and compliance, data types, available features, performance, and security, among many others. Over 40 topics are covered in head-to-head comparisons. It will be particularly useful for organizations who are thinking of making the switch from a commercial to an open-source database, but need more information on the possible trade-offs and advantages of the two systems. However, it is intended for anyone who is curious to learn more about relational databases. We conclude that SQL Server has historically been popular with organizations that rely on other Microsoft products, but PostgreSQL has risen to the top of the field not only because of the advantages of going open source but also for its robust features and active community of users.

## ****What is the difference between PostgreSQL and SQL Server licensing? Comparison of PostgreSQL vs. MSSQL Server licensing model****

#### **PostgreSQL**

PostgreSQL is an open source database released under the PostgreSQL License, an Open Source Initiative Approved License. The use of PostgreSQL for any purpose, including commercial, is free. Under the PostgreSQL Global Development Group, PostgreSQL is available as free and open source software in perpetuity.

#### **SQL Server**

Microsoft SQL Server is available through commercial license and can be licensed on a per-core model or server and client access level (CAL) model. MSSQL is offered in two main editions, Enterprise Edition and Standard Edition, to meet the performance and price requirements of organizations and individuals. Licensing costs range from from $3,586 for the Standard Edition to $13,748 for the Enterprise edition (for  two cores); for the server and CAL model it runs $899 for the server plus $209 per user. A free version is available to students and developers for building and testing.

## ****What are the release update differences between  PostgreSQL and SQL Server? Compare the release updates of PostgreSQL and MSSQL****

#### **PostgreSQL**

PostgreSQL was created in 1986 at the University of California, Berkeley, and first released in 1989. It has undergone several major updates since then, and the project still maintains regular releases under an open-source license. The current version of Postgres is version 12,  released in October 2019, with regular minor releases since  then. Previous major versions are supported for five years after their initial release.

#### **SQL Server**

SQL Server was developed by Microsoft and first released in 1989, and new releases occur regularly. The current version, Microsoft SQL Server 2019, was released in November 2019. Previous versions continue to receive support from SQL Server 2012 onward. Extended support for recent versions is offered for 10 years, with an optional premium assurance paid extension after that for up to 16 years.

## ****Which of PostgreSQL or SQL Server is easier to use? Compare the ease of use of PostgreSQL vs. MSSQL****

* PostgreSQL is an advanced object-relational database management system that uses Structured Query Language (SQL) in addition to its own procedural language, PL/pgSQL. PostgreSQL is easy-to-use with a full stack of RDBMS database features and capabilities for handling data. It can be easily installed on Linux environments.
* SQL Server is a Relational Database Management System (RDBM) developed and operated by Microsoft. It uses a variant of Structured Query Language (SQL) called T-SQL (for Transact-SQL). It can run on Linux operating systems with Kubernetes support or on Windows. Users describe it as easy to use and reliable, with strong .NET compatibility.

## ****What are the syntax differences between PostgreSQL and SQL Server? Compare PostgreSQL vs. MSSQL Server Syntax****

**PostgreSQL vs. SQL Server Syntax Comparison Table**

|  |  |  |
| --- | --- | --- |
| **Syntax** | **PostgreSQL** | **SQL Server** |
| **SELECT** | **col1, col2** | **Select [col1], [col2]** |
| **Aliases for columns and tables** | **SELECT AVG(col1) AS avg1** | **SELECT AVG(col1)=avg1** |
| **Working with dates** | **CURRENT\_DATE() CURRENT\_TIME() EXTRACT()** | **GETDATE() DATEPART()** |

## ****What are the data type differences between PostgreSQL and SQL Server? Compare data types in PostgreSQL vs. MSSQL****

**PostgreSQL vs. SQL Server Data Types Comparison Table**

|  |  |  |
| --- | --- | --- |
| **Data type** | **PostgreSQL** | **SQL Server** |
| **64-bit integer** | **BIGINT** | **BIGINT** |
| **Fixed length byte string** | **BYTEA** | **BINARY(n)** |
| **1, 0 or NULL** | **BOOLEAN** | **BIT** |
| **Fixed length char string, 1 <= n <= 8000** | **CHAR(n)** | **CHAR(n)** |
| **Variable length char string, 1 <= n <= 8000** | **VARCHAR(n)** | **VARCHAR(n)** |
| **Variable length char string, <= 2GB** | **TEXT** | **VARCHAR(max)** |
| **Variable length byte string , 1 <= n <= 8000** | **BYTEA** | **VARBINARY(n)** |
| **Variable length byte string , <= 2GB** | **BYTEA** | **VARBINARY(max)** |
| **Variable length Unicode UCS-2 string** | **VARCHAR(n)** | **NVARCHAR(n)** |
| **Variable length Unicode UCS-2 data, <= 2GB** | **TEXT** | **NVARCHAR(max)** |
| **Variable length character data, <= 2GB** | **TEXT** | **TEXT** |
| **Variable length Unicode UCS-2 data, <= 2GB** | **TEXT** | **NTEXT** |
| **Double precision floating point number** | **DOUBLE PRECISION** | **DOUBLE PRECISION** |
| **Floating point number** | **DOUBLE PRECISION** | **FLOAT(p)** |
| **32 bit integer** | **INTEGER** | **INTEGER** |
| **Fixed point number** | **NUMERIC(p,s)** | **NUMERIC(p,s)** |
| **Date includes year, month, and day** | **DATE** | **DATE** |
| **Date and time with fractional seconds** | **TIMESTAMP(p)** | **DATETIME, DATETIME2(p)** |
| **Date and time with time zone** | **TIMESTAMP(p) WITH TIME ZONE** | **DATETIMEOFFSET(p)** |
| **Date and time** | **TIMESTAMP(0)** | **SMALLDATETIME** |
| **Unsigned integer, 0 to 255 (8 bit)** | **SMALLINT** | **TINYINT** |
| **UUID (16 byte)** | **CHAR(16)** | **UNIQUEIDENTIFIER** |
| **Automatically updated binary data** | **BYTEA** | **ROWVERSION** |
| **Currency amount (32 bit)** | **MONEY** | **SMALLMONEY** |
| **Variable length binary data, <= 2GB** | **BYTEA** | **IMAGE** |
| **Geometric types** | **POINT, LINE, LSEG, BOX, PATH, POLYGON, CIRCLE** | **GEOMETRY** |

## ****What are the geographic data differences between PostgreSQL and SQL Server? Compare geographic data in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL does not have a native data type for geographic data. The open-source [PostGIS](https://postgis.net/" \t "_blank) resource offers support for geographic objects.

#### **SQL Server**

SQL Server has the geography data type for storing geographic spatial data.

## ****What are the case sensitivity differences between PostgreSQL and SQL Server? Compare index types in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL is case sensitive for evaluating strings. The LOWER() function allows users to convert strings to all lower case for evaluation purposes (there is also a similar UPPER() function). By default, PostgreSQL converts table and column names to lowercase, unless those names are placed in quotes. The citext module provides a case insensitive string data type citext for comparing values.

#### **SQL Server**

SQL Server is case insensitive by default. The case sensitivity can be changed by adjusting the SQL Server’s collation settings. The collation settings for case sensitivity can be set at the database or column level.

## ****What are the index type differences between PostgreSQL and SQL Server? Compare index types in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL offers a number of options for index types, including B-tree, hash, Generalized Search Tree (GiST), Space Partitioned GiST, Generalized Inverted Index (GIN), and Block Range Index (BRIN). In addition, it supports expression indexes (indexes created with an expression or function rather than a column value) and partial indexes (indexes of part of a table).

#### **SQL Server**

SQL Server offers clustered and non-clustered indexes. Clustered indexes sort and data rows in the table or view based on key values (columns in the index definition). A table can have only one clustered index. Nonclustered indexes are stored separately from table data, and each key value entry has a pointer to the data. MSSQL creates these automatically when PRIMARY KEY and UNIQUE constraints are defined on table columns. The UNIQUE constraint creates a nonclustered index, while the PRIMARY KEY creates a clustered index unless one already exists.

## ****What are the replication differences between PostgreSQL and SQL Server? Compare replication in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL has Primary-Secondary replication. Replication can be synchronous or asynchronous. Asynchronous replication is accomplished through the use of write-ahead logs (WALs) to share changes with the replica nodes. Streaming replication allows standby servers to be updated more immediately by streaming the WALs as they are created, rather than waiting for the file to be filled. Logical replication follows a publish and subscribe model; this method of replication is called logical because changes are based on the data’s replication identity (for example, a primary key) rather than its physical location. Physical replication deals with files and directories, without regard for the contents within those physical locations. PostgreSQL does not natively offer multi-master replication, but some third-party tools offer multi-master replication solutions.

#### **SQL Server**

SQL Server replication duplicates data from a Publisher server to a Subscriber offers three types of replication:

* transactional replication, for server-to-server environments, where changes are delivered from the publisher to the subscriber as they occur;
* merge replication, for server-to-client environments or in situations where conflicts might occur, where data can be changed and tracked on either the publisher or subscriber and later synchronized;
* snapshot replication, for when data is updated infrequently or does not need to be changed incrementally, where data is duplicated exactly as it appears at a specific moment.

Replication in SQL Server can be synchronous-commit or asynchronous commit. The  Enterprise edition offers peer-to-peer replication, as an alternative solution to multi-master replication.

## ****What are the differences in clustering between PostgreSQL and SQL Server? Compare clustering in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL allows clusters of servers but does not natively support multi-master or active/active clusters. Tools such as [repmgr](https://repmgr.org/" \t "_blank) allow for easy maintenance of PostgreSQL clusters.

#### **SQL Server**

SQL Server offers Windows Server Failover Clustering, which can be configured for both active/passive and active/active nodes. The Standard edition only supports two nodes for clusters; additional nodes require an upgrade to the Enterprise edition.

## ****What are the differences in high availability between PostgreSQL and SQL Server? Compare the high availability in PostgreSQL vs MSSQL****

#### **PostgreSQL**

PostgreSQL offers a number of solutions to ensure high availability for users, including shared disk failover, write-ahead log shipping, data partitioning, and multiple replication methods. Tools like [EDB Postgres Failover Manager](https://www.enterprisedb.com/products/postgresql-automatic-failover-manager-cluster-high-availability) provide automatic failover to ensure high availability by monitoring for and identifying database failure.

#### **SQL Server**

SQL Server includes a number of high availability tools in its various editions. These include replication, log shipping, and failover clusters. Its Always On availability groups, offered with the Enterprise edition, provide automatic failover when certain conditions are met.

## ****What are the “views” differences between PostgreSQL and SQL Server? Compare the “views” in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL supports views—virtual tables that do not store data themselves. Updatable views are supported, but updates do not occur automatically unless it meets the following conditions:

1. The query of that view must have precisely one section in the FROM clause and this can be a table or another updatable view.
2. The selection list must not contain any window function any aggregate functions or any set-returning function.
3. The query must not contain one of the following clauses at the top level: HAVING, LIMIT, DISTINCT, WITH, INTERSECT, EXCEPT, OFFSET AND LIMIT.

Views created with simple queries can be updated; ones created with complex queries cannot, but complex views can be updated by using rules. Materialized Views are also supported; the data in materialized views can be updated using the REFRESH MATERIALIZED VIEW statement.

#### **SQL Server**

SQL Server views can be used for security purposes to restrict user access to data. Both user-defined and system-defined views are supported. Views can be automatically updated using triggers. The data in a view can be updated when the modifications are made to a column from a single underlying base table and are referenced directly. Materialized views are known in SQL Server as Indexed Views; unlike materialized views in other relational databases, indexed views are synched to the underlying data and are thus updated automatically.

## ****What are the trigger differences between PostgreSQL and SQL Server? Compare the triggers in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL has advanced triggers. Supported triggering events are AFTER, BEFORE, and INSTEAD OF, and they can be used for INSERT, UPDATE, and DELETE events. Functions can be used to execute a complex SQL when the trigger gets invoked. PostgreSQL can execute this dynamically.

#### **SQL Server**

SQL Server offers triggers for different types of database events:

* DML Triggers: for a data manipulation language (DML) specific event, such as inserting, updating, or deleting records. These triggers fire on events irrespective to the number of rows affected.
* DDL Triggers: for data definition language (DDL) events, such as CREATE, DROP, or ALTER statements. These are useful for preventing or auditing changes to the database schema.
* Logon Triggers: for logon events, such as when a user session is established. These triggers fire after successful authentication and before establishing the user session. They are useful for auditing and controlling login activity.

## ****What are the stored procedures differences between PostgreSQL and SQL Server? Compare the stored procedures in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL supports stored procedures as user-defined functions with a RETURN VOID clause. Stored procedures are supported in various languages in addition to standard SQL syntax.

#### **SQL Server**

SQL Server supports stored procedures for languages supported by Microsoft .NET framework (common runtime languages or CLR, like VB, C#, or Python).

## ****What are the query differences between PostgreSQL and SQL Server? Compare the query in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL offers the PL/pgSQL procedural programming language. Additional functionalities to standard SQL in PostgreSQL include advanced types and user-defined types, extensions and custom modules, JSON support, and additional options for triggers and other functionality.

#### **SQL Server**

SQL Server uses T-SQL, which has a similar query syntax to standard SQL. T-SQL includes additional support for strings and data processing, local variables, and procedural programming.

## ****What are the full-text search differences between PostgreSQL and SQL Server? Compare full-text search in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL offers advanced functionality for full-text search. It uses full-text indexing and dictionaries for faster searches. Preprocessed text documents are stored as a tsvector data type, and processed queries are stored as the tsquery type. Preprocessing parses text documents into linguistic units known as lexemes, which allows you to find case insensitive variants of a word.

#### **SQL Server**

SQL Server offers full-text search as an optional component. Queries are run against a full-text index, and searches can be based on particular language rules. Searches are performed on columns or text data types (including char, varchar, nchar, nvarchar, text, ntext, image, xml, or varbinary(max) and FILESTREAM) using the T-SQL commands CONTAINS to match words and phrases and FREETEXT to match meaning. Thesaurus files can be used to help find synonyms of search terms.Full-text searches in SQL Server are not case sensitive.

## ****What are the regular expression differences between PostgreSQL and SQL Server? Compare regular expressions in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL has three methods for evaluating regular expressions: LIKE, SIMILAR TO, and POSIX regular expressions.

#### **SQL Server**

SQL Server does not natively support regular expression evaluation; similar but limited results can be achieved using the T-SQL functions LIKE, SUBSTRING, and PATINDEX.

## ****What are the partitioning differences between PostgreSQL and SQL Server? Compare the partitioning in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL offers built-in support for range, list and hash partitioning. Range partitioning groups a table is into ranges defined by a partition key column or set of columns—for example, by date range. List partitioning breaks a table into groups by explicitly listing predefined key values that appear in each partition.

[EDB Postgres Advanced Server](https://www.enterprisedb.com/products/edb-postgres-advanced-server-secure-ha-oracle-compatible) also supports Interval Partitioning, which automatically creates the interval partitions as data arrives without causing deadlocks.

#### **SQL Server**

SQL Server supports table and index partitioning. The data is partitioned horizontally and maps groups of rows into individual partitions. All partitions of a single index or table must reside in the same database, and the table or index is treated as a single entity for queries and updates.

## ****What are the table scalability differences between PostgreSQL and SQL Server? Compare the table scalability in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL has several indexing and two types of partitioning options to improve data operations and query performance on a scalable table. Table partitions and Indexes can be placed in separate tablespaces on different disk file systems, which can greatly improve table scalability. Postgres does not support horizontal table partitioning, but several commercially developed products are available.

#### **SQL Server**

SQL Server contains scalability enhancements to the on-disk storage for memory-optimized tables. The current versions offer multiple concurrent threads to persist memory-optimized tables, multithreaded recovery and merge operations, dynamic management views. Scaling in SQL Server can be easily achieved through sharding.

## ****What are the table statement differences between PostgreSQL and SQL Server? Compare the table statements in PostgreSQL vs. MSSQL****

|  |  |  |
| --- | --- | --- |
| **Truncate** | **In PostgreSQL, TRUNCATE removes all rows from a set of tables. It is faster than DELETE because it does not scan the tables first, and disk space is reclaimed immediately, without the need for a subsequent VACUUM operation. This is useful on large tables.**  **TRUNCATE customers;** | **In SQL Server TRUNCATE TABLE removes all rows from a table or specified partitions of a table, similar to a DELETE statement with no WHERE clause. TRUNCATE TABLE works faster than DELETE and uses fewer resources because it does not log individual row deletions.**  **TRUNCATE TABLE   customers**  **WITH (PARTITIONS (2, 4, 6 to 8));** |
| **Inheritance** | **PostgreSQL supports object-oriented programming features, including the use of inheritance.**  **CREATE TABLE country (**  **Name   text,**  **area real,**  **population real);**  **CREATE TABLE capitals (**  **city text) INHERITS (country);**  **When INHERITS is applied to a table, it inherits all the fields and properties of the parent table, which helps speed up development and improve readability.** | **SQL Server is not an object-oriented database and does not support table inheritance. However, a similar outcome is achievable through the use of DDL Triggers.** |
| **Nested** | **PostgreSQL does not explicitly support nesting data. It does support arrays of arbitrary types, which has an equivalent effect:**  **CREATE TYPE BeerType AS (**  **name CHAR(25),**  **kind CHAR(15),**  **percentage NUMERIC(2, 0)**  **);**  **CREATE TABLE BeerDrinkers (**  **name CHAR(35),**  **address AddrType,**  **beers BeerType[]**  **);** | **In SQL Server, a nested table can be created when two source tables contain a defined relationship, where items in one table can be related to those in the other. This can be a unique identifier shared by both tables. Nested tables can be useful for analyzing data.** |

## ****What are the compliance differences between PostgreSQL and SQL Server? Compare the compliance in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL is an object-relational database management system (ORDBMS) designed to emphasize extensibility and standards compliance. It is ACID-compliant.  [EDB Postgres Advanced Server](https://www.enterprisedb.com/products/edb-postgres-advanced-server-secure-ha-oracle-compatible) is also HIPAA, GDPR, and PCI compliant.

#### **SQL Server**

SQL Server is a relational database management system (RDBMS) with an emphasis on security and performance. It is ACID-compliant.

## ****What are the column differences between PostgreSQL and SQL Server? Compare the columns in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL introduced a new constraint feature in version 10 called GENERATED AS IDENTITY. This is an SQL-compliant variant of the SERIAL column that allows you to assign a unique value to an identity column automatically.

For a SERIAL column to have a unique constraint or be a primary key, it must now be specified, just like other data types. Unique identifier columns are created using the data types smallserial, serial, and bigserial, similar to auto-increment features in other databases.

#### **SQL Server**

SQL Server's identity column property creates an identity column for a table for generating key values for rows. Two values are specified when it is created: seed (initial value for the first row) and increment (amount to increase value over the previous row). By default, both the seed and incremental values are 1.  Each table can only contain one identity column. Uniqueness of the values are not guaranteed unless PRIMARY KEY or UNIQUE constraints are imposed.

## ****What are the computed column differences between PostgreSQL and SQL Server? Compare the computed column differences in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL uses the term generated columns for computed columns. This feature was recently introduced with version 12. Generated columns can be physically stored when they are marked STORED; otherwise, they are not stored and known as virtual.

**CREATE** **TABLE** **table** (

…,

computed\_column GENERATED ALWAYS **AS** (expression) STORED

);

Generated columns cannot have an identity definitions or be part of a partition key; they can only reference the current row and cannot use subqueries. Values cannot be specified using INSERT or UPDATE, but the keyword DEFAULT is accepted.

#### **SQL Server**

SQL Server computed columns are not physically stored in a table unless the column is marked with the PERSISTED property; the column can only be persisted when the value is deterministic, or always returns the same result.

**ALTER** **TABLE** **table**

**ADD** computed\_column **AS** expression [PERSISTED];

If the computed column is deterministic and an acceptable data type, it can be used as a PRIMARY KEY or index, but it cannot be used as a DEFAULT or FOREIGN KEY constraint. Values cannot be specified using INSERT or UPDATE.

## ****What are the differences between PostgreSQL and SQL Server when deleting tables? Compare the differences when deleting table data in PostgreSQL vs. MSSQL****

You can delete data from a table in PostgreSQL using the DELETE statement:

**DELETE** **FROM** **table**

**WHERE** condition;

The DELETE FROM clause specifies the table, and the rows to delete are specified by using the condition in the WHERE clause. The WHERE clause is optional, but if you omit it, the statement will delete all rows in the table.

The DELETE statement removes rows from a table in SQL SERVER:

**DELETE** **FROM** **table**

**WHERE** condition;

The name of the table from which the rows are to be deleted is specified in the FROM clause, and the rows to delete are specified by the condition in the WHERE clause. The WHERE clause is optional, but if you skip it, all rows from the table will be removed.

You can specify the number or percent of random rows that will be deleted, by using the TOP clause.

**DELETE** TOP 10 **FROM** **table**;

This statement removes 10 random rows from the table. Because rows are stored in unspecified order, we do not know which 10 rows will be deleted. Similarly, you can delete the 10 percent of random rows:

**DELETE** TOP 10 PERCENT **FROM** **table**;

## ****What are the differences of integers between PostgreSQL and SQL Server? Compare the integers in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

There are three kinds of integers in PostgreSQL:

1. SMALLINT  (small integer, a 2-byte type with a range from -32,768 to 32,767)
2. INT (integer, a 4-byte type with a range from -2,147,483,648 to 2,147,483,647)
3. BIGINT (a large-range integer:  -9223372036854775808 to 9223372036854775807)

#### **SQL Server**

SQL SERVER supports standard SQL integer types BIGINT, INT, SMALLINT, and TINYINT. The range and storage size of each type is as follows:

|  |  |  |
| --- | --- | --- |
| **PostgreSQL vs. SQL Server Integers Comparison Table** | | |
| **Data type** | **Range** | **Storage** |
| **BIGINT** | **-263 (-9,223,372,036,854,775,808) to 263-1 (9,223,372,036,854,775,807)** | **8 Bytes** |
| **INT** | **-231 (-2,147,483,648) to 231-1 (2,147,483,647)** | **4 Bytes** |
| **SMALLINT** | **-215 (-32,768) to 215-1 (32,767)** | **2 Bytes** |
| **TINYINT** | **0 to 255** | **1 Byte** |

## ****What are the boolean type differences between PostgreSQL and SQL Server? Compare the boolean types in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

The PostgreSQL Boolean data type can have 3 states:

1. TRUE,
2. FALSE, and
3. NULL.

#### **SQL Server**

In SQL SERVER the BIT data type is used to represent true/false boolean data. A BIT field's value is either 1, 0, or null.

## ****What are the NoSQL capability differences between PostgreSQL and SQL Server? Compare the NoSQL capabilities in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL, like many other relational databases, has added support for JSON data, the most common format for semi-structured data stored in NoSQL systems. But because SQL is the only way to interact with a PostgreSQL database, it should not be considered NoSQL.

#### **SQL Server**

SQL Server has native JSON functions that enable you to parse JSON documents using standard SQL language. You can store JSON documents in SQL Server and query that JSON data just as you would in a NoSQL database. Still, because SQL Server is an SQL database, it should not be considered NoSQL.

## ****What are the security differences between PostgreSQL and SQL server? Compare the security in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL supports SSL (Secure Sockets Layer) connections to encrypt client-server communications. You can enable SSL by setting the ssl parameter in the postgresql.conf file. To meet an enterprise need, [EDB Postgres Advanced Server](https://www.enterprisedb.com/products/edb-postgres-advanced-server-secure-ha-oracle-compatible) includes additional built-in auditing features that capture more detailed data, integrated password policy management capabilities and data redaction.

#### **SQL Server**

SQL Server offers a range of features and functions to prevent security threats, because each application is unique in its security needs. The SQL Server security framework manages access to securable entities through authentication and authorization. SQL Server has support for a hierarchy of encryption options and supports TLS (transport layer security) for encrypting network traffic.

## ****What are the analytical function differences between PostgreSQL and SQL server? Compare the analytical functions in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL supports various analytical functions, which perform aggregation on a set of rows. There are two types of analytical functions: window functions and aggregate functions. Aggregate functions perform aggregation and return a single aggregate value for a set of rows (like sum, avg, min, or max). Window functions return a single aggregate value for each of the rows.

**PostgreSQL supports the following window functions:**

|  |  |
| --- | --- |
| **Function** | **Description** |
| **CUME\_DIST** | **Return the relative rank of the current row** |
| **DENSE\_RANK** | **Rank the current row within its partition without gaps** |
| **FIRST\_VALUE** | **Return a value evaluated against the first row within its partition** |
| **LAG** | **Return a value from a specified physical offset row before the current row within the partition** |
| **LAST\_VALUE** | **Return a value evaluated against the last row within its partition** |
| **LEAD** | **Return a value from a row that is offset rows after the current row within the partition** |
| **NTILE** | **Divide rows in a partition as equally as possible then assign each row an integer from 1 to the argument value** |
| **NTH\_VALUE** | **Return a value evaluated against the nth row in an ordered partition** |
| **PERCENT\_RANK** | **Return the relative rank of the current row** |
| **RANK** | **Rank the current row within its partition with gaps** |
| **ROW\_NUMBER** | **Number the current row within its partition starting from 1.** |

#### **SQL Server**

In SQL Server, analytic functions can return multiple rows for each group of rows that the function is performed on. They can be used to compute moving averages, running totals, percentages or top-N results within a group.

SQL Server supports the following analytic functions:

|  |  |
| --- | --- |
| **Function** | **Description** |
| **CUME\_DIST (Transact-SQL)** | **Calculate the cumulative distribution of a value within a group** |
| **FIRST\_VALUE (Transact-SQL)** | **Return the first value in an ordered set of values** |
| **LAG (Transact-SQL)** | **Return value of a previous row to compare values without requiring a self-join** |
| **LAST\_VALUE (Transact-SQL)** | **Return the last value in an ordered set of values** |
| **LEAD (Transact-SQL)** | **Return value of a subsequent row to compare values without requiring a self-join** |
| **PERCENTILE\_CONT (Transact-SQL)** | **Calculate a percentile based on continuous distribution of column values** |
| **PERCENTILE\_DISC (Transact-SQL)** | **Calculate a percentile based on discrete distribution of column values** |
| **PERCENT\_RANK (Transact-SQL)** | **Calculate relative rank of a row within group** |

## ****What are the dynamic action differences between PostgreSQL and SQL Server? Compare dynamic  actions in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL does not support dynamic actions. Most similar operations can be achieved using select statements.

#### **SQL Server**

SQL Server does not support dynamic actions. Some similar operations can be achieved using stored procedures.

## ****What are the administration and GUI tools differences between PostgreSQL and SQL server? Compare the administration with GUI tools in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL can be administered through a GUI using Oracle’s SQL Developer, pgAdmin, OmniDB, DBeaver, and [Postgres Enterprise Manager](https://www.enterprisedb.com/products/postgresql-enterprise-manager-best-gui-tools-database-management). Other GUI tools used for monitoring health and performance include Nagios, Zabbix, Cacti and [EDB Postgres](https://www.enterprisedb.com/products/edb-postgres-database-product-overview-features-specifications). SQLECTRON is a cross-platform option that is free and open source; it is compatible with a number of SQL databases including SQL Server.

#### **SQL Server**

SQL Server can be administered through a GUI on Windows using SQL Server Management Studio (SSMS), which is free. SQL Operations Studio is a free, opensource, cross-platform GUI for Mac. SQLECTRON is a cross-platform option that is free and open source; it is compatible with a number of SQL databases including PostgreSQL.

## ****What are the performance differences between PostgreSQL and SQL server? Compare the performance of PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL offers speed and performance across data sets of all sizes, and it regularly outperforms other databases in both online transaction processing (OLTP) and online analytical processing (OLAP) speeds. It offers multi-version concurrency control (MVCC), allowing multiple transactions to be processed simultaneously, with fewer deadlocks than SQL Server. PostgreSQL offers many tools and parameters that allow users to monitor and optimize database performance.

#### **SQL Server**

SQL Server prides itself in the speed of its analytical and transaction processing. However, because the SQL Server user agreement prohibits the publication of benchmark testing without Microsoft’s prior written approval, head-to-head comparisons with other database systems are rare. Among features SQL Server highlights for optimizing performance and speed is its In-Memory OLTP, which takes advantage of in-memory data tables that perform better than writing directly to disk. The SQL Server Standard edition has some performance limitations for memory, partitioning, indexing, and other functionalities that require upgrading to the Enterprise version.

## ****What are the concurrency differences between PostgreSQL and SQL Server? Compare concurrency in PostgreSQL vs. MSSQL.****

#### **PostgreSQL**

PostgreSQL has well-developed multi-version concurrency control (MVCC) for handling multiple procedures at one time. MVCC provides snapshots of database info to avoid showing inconsistencies caused by simultaneous transactions or locking of data that occurs in other database systems. It uses Serializable Snapshot Isolation (SSI) to guarantee transaction isolation.

#### **SQL Server**

SQL Server has a less fully developed multi-version concurrency control system and by default relies on locking of data to prevent errors from simultaneous transactions.  It offers an optimistic concurrency feature, which assumes that such conflicts only rarely appear; instead of locking a row, it is checked against a cached version to detect if any change has occurred.

## ****What are the adoption differences between PostgreSQL and SQL server? Compare the adoption in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL is the world's most advanced open source database. Businesses around the world are using PostgreSQL for mission critical workloads. The PostgreSQL community and a few companies such as Enterprise DB and 2ndQuadrant are making sure that PostgreSQL adoption continues to expand on a global level.

#### **SQL Server**

SQL Server is popular with enterprises that rely on Microsoft products. It saw an increase in market share over the past two decades as Microsoft pushed it with its Windows Servers. But with more and more enterprises making a shift to Open source in recent years, the popularity curve of SQL server is becoming more and more flat.

## ****What are the environment and stack differences between PostgreSQL and SQL server? Compare the environment and stack in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL is popular with the LAPP stack (Linux, Apache, PostgreSQL, and PHP/Python. The LAPP stack is growing in popularity; large-platform service providers like Amazon and VMware provide services with readily installed LAPP stack modules.

#### **SQL Server**

SQL Server is a popular component of the Microsoft stack. It consists of Microsoft technologies like Microsoft WPF, ASP.NET, SharePoint, and Office 365.

## ****What are the differences in scheduling tasks between PostgreSQL and SQL Server? Compare the scheduling tasks in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL doesn’t provide a built-in job scheduler like other SQL databases do. Recurring tasks require external tools like pgAgent, cron, or pg\_cron on Linux, and Task Scheduler or  SQLBackupAndFTP on Windows.

#### **SQL Server**

Tasks in the SQL Server can be scheduled using SQL Server Management Studio.

## ****What are the differences about data redaction between PostgreSQL and SQL Server? Compare the data redaction in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL does not support data redaction for limiting the display of sensitive data for certain users.  Enterprises looking for data redaction features to add more security can use EDB Postgres Advanced server, Oracle compatible fork of PostgreSQL.

#### **SQL Server**

SQL Server offers dynamic data masking (DDM). DDM limits sensitive data exposure by hiding it from users without the proper privileges. It complements other SQL Server security features like auditing, encryption, and row level security.

## ****What are the different limitations between PostgreSQL and SQL Server? Compare the limitations of PostgreSQL vs. MSSQL****

#### **PostgreSQL**

In addition to SQL and PL/pgSQL, PostgreSQL supports includes the procedural languages PL/Tcl, PL/Perl, and PL/Python in its distribution, and supports the external procedural languages PL/Java, PL/Lua, PL/R, PL/sh (Unix shell), and PL/JavaScript. It also supports user-defined functions in C-languages. It can be hosted on a wide range of server operating systems, including Linux, Mac, Windows, BSD, and Solaris. It can be deployed on Docker containers or Kubernetes.

#### **SQL Server**

In addition to T-SQL,  SQL Server supports languages that are compatible with the Microsoft .NET framework, including C#, Java, PHP, and Python. SQL Server must run on Linux or Windows operating systems. It can be deployed on Docker containers and on Kubernetes with Microsoft’s Azure Kubernetes Services.

## ****What are the access method differences between PostgreSQL and SQL server? Compare the access methods in PostgreSQL vs. MSSQL****

#### **PostgreSQL**

PostgreSQL is compatible with the following access methods, protocols, and APIs for gaining access to its data: ADO.NET, JDBC, ODBC, and the native C library. It also supports a streaming API for binary large objects (BLOBs).

#### **SQL Server**

SQL Server is compatible with the following access methods, protocols, and APIs for gaining access to its data: ADO.NET, JDBC, ODBC, OLE DB, and TDS.

## ****What are the bulk collect and binds differences between PostgreSQL and SQL server? Compare the bulk collect and binds in PostgreSQL vs. MSSQL****

|  |  |  |
| --- | --- | --- |
| **Bulk Collect** | **PostgreSQL does not have syntax for bulk collect, nor any close functional equivalent. Instead, you can create a temporary table with PL/PgSQL code or use a common table expression (CTE, or WITH query), if working within a single SQL statement.** | **There is no syntax for bulk collect in SQL Server. One alternative is to use a temporary table and a cursor.** |
| **Binds** | **PostgreSQL does not support bind variables, unlike some other relational databases like Oracle. Instead, PostgreSQL uses the PREPARE statement to achieve similar results.** | **SQL Server supports bind variables. Each parameter marker in an SQL statement must be bound to a variable before the statement can be executed using the [SQLBindParameter](https://docs.microsoft.com/en-us/sql/relational-databases/native-client-odbc-api/sqlbindparameter?view=sql-server-ver15" \t "_blank) function. Parameters can also be bound to arrays of program variables to process an SQL statement in batches. SQL Server also supports defining names for stored procedure parameters.** |

## ****What are the differences about synonyms between PostgreSQL and SQL server? Compare the synonyms in PostgreSQL vs. MSSQL****

SQL Server supports synonyms. Synonyms provide a layer of abstraction that protects a client application from changes made to base objects. A synonym belongs to a schema, and like other objects in a schema, its name must be unique. Binding is by name only; if a base object is modified, dropped, or replaced, the missing reference will only be found at run-time. PostgreSQL does not support synonyms.