Using ANSI SQL

by Sophia

WHAT'S COVERED

This lesson explores how popular relational database management systems implement and extend the SQL standards set forth by the American National Standards Institute (ANSI). Specifically, this lesson will cover:

1. SQL Standard

The American National Standards Institute (ANSI) has created a standard that defines how SQL is used in relational databases. In addition to defining a common syntax and semantics for querying, updating, and managing relational databases, the ANSI SQL standard also defines a way to make SQL commands portable and consistent across different database management systems (DBMS).

KEY CONCEPT

The standard specifies a number of basic SQL commands, such as SELECT, INSERT, UPDATE, DELETE, and others, along with the associated clauses and options. Data integrity, transaction management, data types, and other database operations are also defined. The ANSI SQL standard reduces vendor lock-in and promotes interoperability among different vendors by allowing developers and database administrators to write SQL code that works across various database platforms.

KEY CONCEPT

Vendor lock-in refers to a situation in which a customer or organization becomes heavily dependent on a particular vendor's products, services, or technologies to the extent that it becomes difficult or costly to switch to an alternative vendor. This dependency can result from various factors, including proprietary formats, unique features, custom integrations, or the high cost of migrating to a different vendor's solution.

Although ANSI SQL defines a common subset of SQL, many database vendors also provide proprietary extensions and features that go beyond the standard. For example, Microsoft SQL Server uses a TOP command to limit the number of rows returned in a query, and Oracle uses a ROWNUM command to do that same thing. Different database systems also have their own ways of auto-incrementing columns. MySQL uses AUTO_INCREMENT, and SQL Server uses IDENTITY. ANSI SQL doesn't include either of those, but it does have its own way of doing the same thing, with the GENERATED ALWAYS AS IDENTITY syntax. So, even though ANSI SQL ensures a certain degree of portability in terms of the basic command set, working with specific DBMS implementations will often require some customization and adjustment.

TERM TO KNOW

Vendor Lock-In
A situation in which a customer or organization becomes heavily dependent on a particular vendor's products, services, or technologies to the extent that it becomes difficult or costly to switch to an alternative vendor.

2. PostgreSQL and ANSI SQL Standards

PostgreSQL, an open-source relational database management system, aims to comply with ANSI SQL standards to provide SQL compatibility and interoperability. While not fully compliant with all ANSI SQL standards, PostgreSQL adheres to a significant portion of the standard. This makes it one of the most standards-compliant open-source databases available.

Some ways PostgreSQL complies with ANSI SQL standards include:

Basic SQL Syntax	PostgreSQL follows ANSI SQL syntax for fundamental SQL commands like SELECT, INSERT, UPDATE, DELETE, and others, ensuring portability and familiarity for developers. PostgreSQL supports a wide range of ANSI SQL data types, such as numeric, character, date/time, and boolean. It also supports more advanced data types like arrays, JSON, and **universally unique identifiers (UUIDs)**.
Transaction Management	PostgreSQL implements transaction management as per ANSI SQL standards, supporting ACID (atomicity, consistency, isolation, durability) properties to ensure data integrity. PostgreSQL allows the definition of primary keys, foreign keys, unique constraints, and check constraints, which are part of the ANSI SQL standard for data integrity.
Common Table Expressions (CTEs)	PostgreSQL supports **common table expressions (CTEs)**, a feature introduced in ANSI SQL that allows temporary result sets for complex queries. PostgreSQL supports ANSI SQL window functions, enabling advanced analytical queries and calculations over data partitions. Along with joins and aggregates, PostgreSQL provides various join methods and aggregate functions in accordance with ANSI SQL standards.

While PostgreSQL is known for its excellent ANSI SQL support, it's worth mentioning that it also offers many additional features and functionalities that go beyond the standard. This makes it a robust and powerful database system for a wide range of applications. Developers can leverage standard-compliant features for portability across different database systems, while also leveraging PostgreSQL's unique capabilities for specific use cases.

TERMS TO KNOW

Universally Unique ID (UUID)
A 128-bit identifier that is guaranteed to be unique across both space and time. It is often used as a primary key or unique identifier for records in a database, particularly in distributed and decentralized systems where ensuring uniqueness is crucial.

Common Table Expressions (CTEs)
A feature introduced in ANSI SQL that allows temporary result sets for complex queries.

3. Standards in the Real World

Databases often do not support pure ANSI SQL, for several reasons. They mostly have to do with the development of the database itself, the specific niche in which the database has been used, and the results of marketing or customer surveys on what they need out of a database.

IN CONTEXT

SQL databases have a long history, dating back to the 1970s. Different database vendors developed their own SQL versions before the ANSI SQL standardization efforts began. As a result, many databases already had established proprietary features and extensions that users widely adopted. Database vendors use proprietary extensions and features to differentiate their products in the market. These extensions may provide advanced functionality, better performance, or unique capabilities that are not part of the ANSI SQL standard. Vendors can attract customers and maintain a competitive edge by offering exclusive features.

Many organizations have large, complex legacy databases built on specific vendor platforms and rely heavily on proprietary SQL features. Migrating these systems to comply with the pure ANSI SQL standard could be time consuming, costly, and may require significant changes to existing applications.

Some databases optimize query processing and indexing using nonstandard SQL constructs. These optimizations may provide superior performance for specific workloads, making them more attractive for certain use cases. As technology advances, upcoming features and innovations may still need to be part of the ANSI SQL standard. Database vendors may implement these features to improve functionality, usability, and efficiency.

Over time, users become accustomed to the specific syntax and features of the database they use. Abruptly switching to a pure ANSI SQL standard could disrupt workflows and require retraining. Most features are developed because enough customers need a particular function, process, or command support. Developers lock themselves into specific technologies and go to great efforts to know all the nuances of those technologies and products. Changing a database or even removing a feature requires the code to evolve beyond the normal scope of change and might break a business workflow. Changing or altering the function of a database is a huge deal when it comes to business and manufacturing workflows.

While adhering to the ANSI SQL standard is beneficial for portability and interoperability, database vendors aim to strike a balance regarding compliance with the standard. They also aim to offer unique features that meet their user base's specific needs and preferences. As a result, most databases provide a mix of ANSI SQL-compliant features and vendor-specific extensions to cater to a broad range of use cases and user requirements.

SUMMARY

In this lesson, you learned that to ensure portability and consistency across different database platforms, PostgreSQL strives to adhere to ANSI **SQL standards**. The **PostgreSQL** database conforms to a significant part of the **ANSI SQL standard**, but it also includes additional capabilities that go beyond it. You learned about techniques for changing databases. SELECT, INSERT, UPDATE, and DELETE commands are supported, as are ANSI SQL data types, ACID properties for transaction management, and data integrity constraints. As part of the ANSI SQL standard, PostgreSQL offers various join methods, CTEs, window functions, and aggregate functions. You also learned about **standards in the real world**. The extensibility and openness of PostgreSQL make it a powerful and versatile database system that enables users to create custom functions, data types, and procedural languages, while maintaining ANSI SQL compatibility.

TERMS TO KNOW

Common Table Expressions (CTEs)
A feature introduced in ANSI SQL that allows temporary result sets for complex queries.

Universally Unique ID (UUID)
A 128-bit identifier that is guaranteed to be unique across both space and time. It is often used as a primary key or unique identifier for records in a database, particularly in distributed and decentralized systems, where ensuring uniqueness is crucial.

Vendor Lock-In
A situation in which a customer or organization becomes heavily dependent on a particular vendor's products, services, or technologies to the extent that it becomes difficult or costly to switch to an alternative vendor.