



SQL Analytics: the Foundation of Business Intelligence

What is SQL Analytics?

SQL (pronounced “sequel”) is short for “structured query language.” As the name implies, it’s a programming language used to query and manage databases. Basically, it’s how data engineers, analysts, and developers “talk” to a database to find out what they want to know, pull data, and more. It’s the most basic way database professionals manipulate information, combining elements from different datasets into new tables that are greater than the sum of their parts, in terms of what they are able to show users.

No one chart or table has all the answers, and SQL analytics tools have been created since the rise of this powerful, versatile language to empower nontechnical to pull together information in a drag-and-drop environment that discreetly maps simple clicks to SQL code in the back end. However SQL still remains an essential tool in the hands of data engineers and business analysts. We'll dig into what makes SQL so useful, what it's used for, types of SQL databases, and finish by talking about the variety of SQL tools out there.

SQL: Why It's Great

Beyond its essential purpose, to pull data out of tables and combine it in new ways, one of SQL's other strengths is that it's open-source and easy to learn and use. Open-source languages are an essential part of a free and accessible internet and being able to use an open-source language like SQL to manipulate data helps level the playing field for database administrators and other professionals around the world. Data must be democratized to achieve ubiquity; if only those with powerful analytic tools and data science degrees can afford to manipulate it, there will always be a lag between what humanity knows and what it *could know*, with the right combination of datasets, ingenuity, and will.

SQL also handles databases of all sizes; it's database agnostic, for the most part. As long as you're using an SQL database (that is, a type of database built to work with SQL — more on that later), you'll be able to talk to it with SQL, no matter how much data is in there. If you're reading this and you work for an organization with a database, then someone in your company definitely works with SQL to pull insights out of data or prepare datasets for other people to dig into (you may even be that person!). As your data pipeline grows, you may think you need to learn new skills or buy new software, but stable and reliable SQL has you covered (though there are other things that an SQL analytics tool brings to the table).

By now you're probably thinking SQL sounds pretty cool! You're right. SQL is a powerful, versatile language and with the increase in the amounts of data that every company is capturing, buying, and sharing, there's a huge demand for SQL experts. (A quick Indeed search showed almost 100,000 full-time job listings with the term "SQL" in them at time of writing.) Analytics and BI tools are allowing nontechnical team members to dig into this data in a structured environment without writing SQL, but there are still some tasks that SQL is uniquely well-suited for and many database professionals still choose to take advantage of their SQL skills to optimize their analysis. Let's dig into those tasks right now.

Types of SQL Tasks

SQL is top dog when it comes to pulling data out of tables and mashing it up with information from other sources, creating new tables. Every SQL query begins with `SELECT`, as you're telling the system what information you're looking for. You can pull an entire column or columns, get a range of values, or even just one specific piece of information. Adding `JOINS` to your `FROM` clause within the `SELECT` statement allows you to bring together data across multiple tables. Other commands include `DELETE`, `CREATE`, `UPDATE`, and `ALTER`, which are how you edit your data in a variety of ways. SQL analytics tools can help nontechnical users do a lot of these activities, usually by dragging and dropping datasets or clicking on the specific attributes that they want to mash up.

Besides the nuts of bolts of what SQL actually *does*, it also functions in a wide array of broader categories, which we'll dig into on the next page.

- **Data Definition Language:** SQL can be used to define data structures like database schemas. The main statements used to do this are CREATE, ALTER, and DROP. This allows users to create, change, and remove elements from the databases they're working with (tables, indexes, users, etc.).
- **Data Manipulation Language:** As touched on above, SQL isn't just about finding and pulling data, it can also be used to change the database itself. Commands like INSERT, DELETE, and UPDATE affect the contents of a database, ensuring that users have only the information they need in the form that will work for them.
- **Transaction Control Language:** Once the data in a system has been manipulated, those changes aren't permanent until the session is closed out. That's where SQL's role as a TCL comes into play. The COMMIT command locks in recent changes, and ROLLBACK turns the clock back to the last committed save state. SAVEPOINT is a middle ground that a user can ROLLBACK to when needed.
- **Data Control Language:** Data security is one of the tech world's biggest challenges and issues. SQL can be used to safeguard datasets by GRANTing and REVOKEing user privileges at either the system level (ability to create sessions, tables, etc.) or object level (what queries, commands, etc. the user can perform).

SQL's versatility and power has helped make it one of the most prevalent, useful, and most learned languages in the tech world. Any transformation that you need to manipulate your database for analysis, SQL has you covered. Again: SQL analytics tools help bridge gaps for nontechnical users, but a skilled SQL expert can exercise total control over a database, maintaining it, preparing data for analysis, and more.

Databases that Work with SQL

After all this talk about how powerful and useful SQL is, it's worth digging into the types of relational database management systems (RDBMS) that work with SQL (or use a proprietary version thereof).

MySQL: First launched in 1995, MySQL is available as a free, open-source version or a paid enterprise version. It features a graphical user interface in addition to the command-line interface; a variety of third-party front-ends are also available for manipulating the system. One of its major claims to fame is the fact that large corporations like Facebook, Google, Adobe, and others use MySQL.

PostgreSQL: With over 30 years in the game, Postres (as it's often called) is well known in the database world. One of its main strengths is the fact that it can handle a wide variety of data types, including advanced ones like "array." One interesting facet of Postgres is how it manages concurrency: the database takes a "snapshot" of itself during every transaction. This lets the system make changes without affecting other transactions or requiring read locks and maintains ACID compliance.**Microsoft SQL Server:** Attached to one of the biggest technology companies in the world, it's no wonder that Microsoft SQL Server is one of the most-used databases out there. True to Microsoft's all-encompassing, product-driven ways, there are a few different products you can choose from to suit your specific needs. Additionally, Microsoft uses its proprietary T-SQL language for data manipulation and definition tasks, extending the typical SQL suite of commands.

Redshift: Speaking of giant names in the tech world, when it comes to cloud databases, it doesn't get any bigger than Amazon and AWS, and Redshift is one of their leading products. Redshift handles huge amounts of data and especially fast-moving live data. The proprietary version of SQL built for Redshift has been refined to help deal with these immense datasets.

Columnar-storage databases: As the name implies, the thing with these databases is that they store information in columns only, not columns and rows. This means that the system can read the data more quickly and deliver better compression. The corresponding drawback is that when updates are entered, they have to enter data across the entire row (for instance, customer data, where name, address, etc. are all stored in the same row), which can be time-consuming.

Whatever database option you choose, you'll still be able to use SQL to manipulate your information. That's definitely one of SQL's strengths: a lot of databases understand it and that puts a lot of power in the hands of a trained analyst or engineer.

SQL Analytics Tools

After all this talk about how awesome SQL is, it's worth covering SQL analytics tools. These are analytics and BI systems that allow even nontechnical users to harness some of the abilities usually associated with SQL to perform tasks and pull insights from data. The bottom line is that every company is becoming a data company. Organizations of all shapes and sizes in every industry are undergoing digital transformations where they put all their data on the cloud. Every department is capturing (and digging into) all the data it can and often creating or gaining access to cloud databases or warehouses of their own. These companies are also becoming more data-driven, as even frontline business users are expected to have data to back up their business decisions.

Not everyone is going to learn SQL. With SQL analytics tools, they don't have to. The most basic thing that these tools do is allow users to query datasets and get individual pieces of information to make decisions or guide processes. Instead of typing a query with SQL, they just open up the GUI and click around within a prepared dataset until they find what they're looking for. These systems are also pretty good at visualizations, so the information can be presented graphically in a pleasing way (think bar graphs, charts, etc).

Many SQL analytics tools allow users to create data models via a drag-and-drop interface, rather than using SQL to pull specific elements from various datasets to build new tables which they can then query. These data models have several advantages: First off, once the model is built, the builder can reuse it again and again for more queries. The model, linked with the underlying data, stays up to date as they change, but users don't have to send queries back to the source databases every time, speeding up results. Additionally, with multiple users sharing the same platform, they can also share data models, meaning that once a comprehensive data model is assembled, it starts providing value to a wide swath of the company, instead of requiring each department or team to build their own. Strong SQL analytics tools will also have easy-to-control security settings that allow users or admins to designate security access to various users, teams, etc., ensuring that everyone has access to the data they need to do their jobs, without putting sensitive information at risk.

SQL analytics tools are indeed mighty, but they'll never completely replace a skilled database administrator or analyst who knows SQL. For instance, when it comes to making changes to the underlying databases, these platforms don't really go there. They are more useful for end-user capabilities: pulling data, making new models, and sharing insights. Again, a huge strength is that they allow nontechnical users to answer their own data questions without IT assistance. You still need IT and data engineers to manage your databases, set up your pipeline, and maintain them.

Building with SQL and Beyond

SQL is the essence of how humans talk to databases. When it comes to querying, editing, adding, and deleting information, mixing that data in new ways, and a host of other activities, there will always be a place for SQL. Not everyone in your organization is going to learn SQL, and that's okay.

These nontechnical users can still dig into datasets, build models, perform ad hoc analysis, share insights, and much more with robust analytics and data platforms like Sisense. Sisense can also help developers build powerful cloud-native apps that put data, analytics, and insights directly in your customers' hands. Cloud-native architecture makes monetizing data, delivering seamless customer experiences, and empowering users of all kinds with the insights that matter to them is easier than ever (no matter which OS or cloud database you use).

The right SQL skills allow database pros to build amazing datasets. The right analytics platform makes everyone a builder: Data engineers can build robust data models that empower stakeholders across the entire organization to work without IT support. Analysts and business users can build reports to answer complex, game-changing questions. Developers can build the next wave of world-changing cloud-native apps. Whatever your company is building, build boldly with Sisense.

Explore Sisense