Analytics

What is a database schema?

What is a database schema?

A database schema defines how data is organized within a relational database; this is inclusive of logical constraints such as, table names, fields, data types and the relationships between these entities.

Schemas commonly use visual representations to communicate the architecture of the database, becoming the foundation for an organization's data management discipline. This process of database schema design is also known as data modeling.

These data models serve a variety of roles, such as database users, database administrators, and programmers. For example, it can help database administrators manage normalization processes to avoid data duplication. Alternatively, it can enable analysts to navigate these data structures to conduct reporting or other valuable business analyses. These diagrams act as valuable documentation within the database management system (DBMS), ensuring alignment across various stakeholders.

Database schema versus database instance

A database schema is considered the "blueprint" of a database which describes how the data may relate to other tables or other data models. However, the schema does not actually contain data.

A sample of data from a database at a single moment in time is known as a database instance. It contains all the properties that the schema describes as data values. Since database instances are just a snapshot at a given moment, they're likely to change over time, unlike database schemas.

Types of database schemas

While the term schema is broadly used, it is commonly referring to three different schema types—a conceptual database schema, a logical database schema, and a physical database schema.

- Conceptual schemas offer a big-picture view of what the system will contain, how it will be organized, and which business rules are involved. Conceptual models are usually created as part of the process of gathering initial project requirements.

- Logical database schemas are less abstract, compared to conceptual schemas. They clearly define schema objects with information, such as table names, field names, entity relationships, and integrity constraints—i.e. any rules that govern the database. However, they do not typically include any technical requirements.

- Physical database schemas provide the technical information that the logical database schema type lacks in addition to the contextual information, such as table names, field names, entity relationships, et cetera. That is, it also includes the syntax that will be used to create these data structures within disk storage.

Star schema vs. snowflake schema

In both logical schemas and physical schemas, database tables will have a primary key or a foreign key, which will act as unique identifiers for individual entries in a table. These keys are used in SQL statements to join tables together, creating a unified view of information. Schema diagrams are particularly helpful in showing these relationships between tables, and they enable analysts to understand the keys that they should join on. There are two additional types of schemas are also commonly referenced in the context of relational database management systems (RDBMS); these are known as star schemas and snowflake schemas.

While conceptual, logical, and physical schemas contain different levels of information about databases in their diagrams, star and snowflake schemas represent those relationships between entities differently. More specifically, a star schema is a type of relational database schema that is composed of a single, central fact table that is surrounded by dimension tables. This tends to be considered a simpler schema compared to the snowflake schema.

A snowflake schema consists of one fact table that is connected to many dimension tables, which can be connected to other dimension tables through a many-to-one relationship. This schema offers the advantage of low levels of data redundancy but is not as effective when it comes to query performance.

As the name implies, a star schema tends to look like a star whereas a snowflake schema tends to resemble a snowflake.

Benefits of database schemas

As big data continues to grow, database objects and schemas are critical to ensure efficiency in day-to-day company operations. If relational models are poorly organized and poorly documented, they will be harder to maintain, posing problems for both its users and the company.

Some key benefits of database schemas include:

- **Access and security:** Database schema design helps organize data into separate entities, making it easier to share a single schema within another database. Administrators can also control access through database permissions, adding another layer of security for more proprietary data. For example, a single schema may contain personally identifiable information (PII), which you would want to encrypt for privacy and security purposes.

- **Organization and communication:** Documentation of database schemas allow for more organization and better communication among internal stakeholders. Since it provides a common source of truth, it enables users to understand the logical constraints and methods of aggregation across tables.

- **Integrity:** This organization and communication also helps to ensure data validity. For example, it can help administrators manage normalization processes to avoid data duplication. It can also assist in monitoring compliance of the constraints in the schema's database design, enabling adherence to ACID properties (atomicity, consistency, isolation, durability).