Database Management Systems

Database Structuring and Querying with SQL

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
and
Centre for Artificial Intelligence and Machine Learning
Indian Statistical Institute, Kolkata

March, 2022

Basics of SQL

SQL or structured query language is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS). SQL uses a combination of relational algebra and relational calculus constructs. Note that, SQL is a declarative (non-procedural) language.

Basics of SQL

SQL or structured query language is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS). SQL uses a combination of relational algebra and relational calculus constructs. Note that, SQL is a declarative (non-procedural) language.

SQL is not only for querying, rather it also helps in defining the structure of the data, modifying the data and specifying the security constraints.

Basics of SQL

SQL or structured query language is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS). SQL uses a combination of relational algebra and relational calculus constructs. Note that, SQL is a declarative (non-procedural) language.

SQL is not only for querying, rather it also helps in defining the structure of the data, modifying the data and specifying the security constraints.

Note: The SQL keywords are case-insensitive, however, they are often written in uppercase. In some setups, table and column names are case-sensitive.

SQL functionalities

- **Data-definition language (DDL)** – provides commands for defining relation schemas, deleting relations, and modifying relation schemas.

SQL functionalities

- **Data-definition language (DDL)** – provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
- **Data-manipulation language (DML)** – includes commands to work on attributes, insert tuples into, delete tuples from, and modify tuples in the database.

SQL functionalities

- **Data-definition language (DDL)** – provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
- **Data-manipulation language (DML)** – includes commands to work on attributes, insert tuples into, delete tuples from, and modify tuples in the database.
- **View definition** – includes commands for defining views.

SQL functionalities

- **Data-definition language (DDL)** – provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
- **Data-manipulation language (DML)** – includes commands to work on attributes, insert tuples into, delete tuples from, and modify tuples in the database.
- **View definition** – includes commands for defining views.
- **Transaction control** – includes commands for specifying the beginning and ending of transactions.

SQL functionalities

- **Data-definition language (DDL)** – provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
- **Data-manipulation language (DML)** – includes commands to work on attributes, insert tuples into, delete tuples from, and modify tuples in the database.
- **View definition** – includes commands for defining views.
- **Transaction control** – includes commands for specifying the beginning and ending of transactions.
- **Embedded SQL and dynamic SQL** – embeds SQL statements into general-purpose programming languages.

SQL functionalities

- **Data-definition language (DDL)** – provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
- **Data-manipulation language (DML)** – includes commands to work on attributes, insert tuples into, delete tuples from, and modify tuples in the database.
- **View definition** – includes commands for defining views.
- **Transaction control** – includes commands for specifying the beginning and ending of transactions.
- **Embedded SQL and dynamic SQL** – embeds SQL statements into general-purpose programming languages.
- **Integrity** – includes commands for specifying integrity constraints that the data stored in the database must satisfy.

SQL functionalities

- **Data-definition language (DDL)** – provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
- **Data-manipulation language (DML)** – includes commands to work on attributes, insert tuples into, delete tuples from, and modify tuples in the database.
- **View definition** – includes commands for defining views.
- **Transaction control** – includes commands for specifying the beginning and ending of transactions.
- **Embedded SQL and dynamic SQL** – embeds SQL statements into general-purpose programming languages.
- **Integrity** – includes commands for specifying integrity constraints that the data stored in the database must satisfy.
- **Authorization** – includes commands for specifying access rights to relations and views.

History

"An SQL query goes into a bar, walks up to two tables and asks, 'May I join you?'." – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.

History

"An SQL query goes into a bar, walks up to two tables and asks, 'May I join you?'." – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.

History

"An SQL query goes into a bar, walks up to two tables and asks, 'May I join you?'." – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.
1987: IBM published its own corporate SQL standard Systems Application Architecture Database Interface (SAA-SQL).

History

"An SQL query goes into a bar, walks up to two tables and asks, 'May I join you?'." – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.
1987: IBM published its own corporate SQL standard Systems Application Architecture Database Interface (SAA-SQL).
1989: ANSI published an extended version SQL-89.

History

"An SQL query goes into a bar, walks up to two tables and asks, 'May I join you?'." – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.
1987: IBM published its own corporate SQL standard Systems Application Architecture Database Interface (SAA-SQL).
1989: ANSI published an extended version SQL-89.
1992: A major extended version SQL-92 was published.

History

“An SQL query goes into a bar, walks up to two tables and asks, ‘May I join you?’.” – Anonymous.

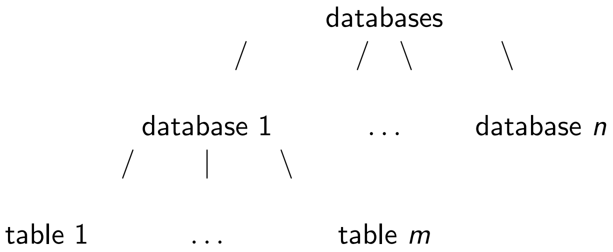
1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.
1987: IBM published its own corporate SQL standard Systems Application Architecture Database Interface (SAA-SQL).
1989: ANSI published an extended version SQL-89.
1992: A major extended version SQL-92 was published.
1999-2016: The versions SQL:1999, SQL:2003, SQL:2006, SQL:2008, SQL:2011 and SQL:2016 were published.

Standard conformance of SQL

Significant Features	**SQL:2008**	**SQL:2011**	**SQL:2016**
Truncation of table	Yes	Yes	Yes
INSTEAD OF trigger	Yes	Yes	Yes
XQuery regular expression	Yes	Yes	Yes
Partitioned JOIN	Yes	Yes	Yes
System-versioned tables	No	Yes	Yes
Time-sliced & sequenced queries	No	Yes	Yes
Temporal referential integrity	No	Yes	Yes
Temporal primary keys	No	Yes	Yes
Polymorphic table functions	No	No	Yes
Row pattern recognition	No	No	Yes
DECFLOAT data type	No	No	Yes
JSON data type	No	No	Yes

Data view through SQL

In practice, the databases (as a whole) comprises several separate database and each database consists of several tables.



Note: The MySQL Community Server can be downloaded from `https://dev.mysql.com/downloads/mysql`.

Principle structure of defining a table

A typical SQL query for defining a table appears as follows:

`create table` R `(`
$A_1 D_1, A_2 D_2, \ldots, A_k D_k$ `,`
$(IC_1), \ldots, (IC_n)$
`);`

Here, each A_i represents an attribute in the schema of relation R, each D_i denotes the data type of values in the domain of the corresponding attribute A_i, and IC_i symbolizes an integrity constraint. Some integrity constraints may also appear along with the data types.

Note: SQL is a freeform language.

The data types in SQL

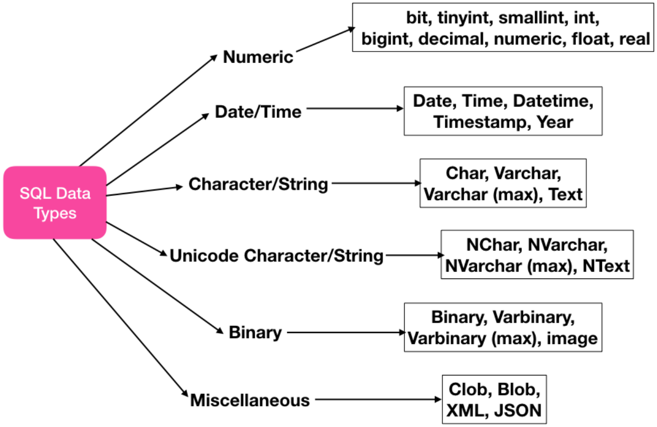

SQL Data Types
Numeric
bit, tinyint, smallint, int, bigint, decimal, numeric, float, real
Date/Time
Date, Time, Datetime, Timestamp, Year
Character/String
Char, Varchar, Varchar (max), Text
Unicode Character/String
NChar, NVarchar, NVarchar (max), NText
Binary
Binary, Varbinary, Varbinary (max), image
Miscellaneous
Clob, Blob, XML, JSON

Table creation with ease

Try this out!!!

SQLizer – Easily convert files into SQL databases

`https://sqlizer.io`

Deleting a table

A typical SQL query for deleting a table appears as follows:

```
drop table R;
```

Altering a table

A typical SQL query for altering a table by adding attributes appears as follows:

alter table R add A_i;

A typical SQL query for altering a table by deleting attributes appears as follows:

alter table R drop A_i;

Principle structure of manipulating a table

A typical SQL query for data manipulation appears as follows:

select $A_1, A_2, \ldots, A_m$
from $R_1, R_2, \ldots, R_n$
where P;

Here, each A_i represents an attribute, each R_i denotes a relation and P is a predicate.

Principle structure of manipulating a table

A typical SQL query for data manipulation appears as follows:

`select` $A_1, A_2, \ldots, A_m$
`from` $R_1, R_2, \ldots, R_n$
`where` P;

Here, each A_i represents an attribute, each R_i denotes a relation and P is a predicate.

- The `select` clause corresponds to the projection operation of the relational algebra.
- The `from` clause corresponds to the Cartesian-product operation of the relational algebra.
- The `where` clause corresponds to the selection predicate of the relational algebra.

Understanding the concepts in a better way

Try this out!!!

RAT – Relational Algebra Translator

`http://www.slinfo.una.ac.cr/rat/rat.html`