Introduction to SQL

Before we dive into this let's take a look at some basic terminology related to SQL.

Data:

The term "data" is used to refer to any and all of the constituent elements that make up a database. A database table is often arranged into columns that define the data types stored within.

Database:

A database is a collection of structured data stored electronically, often on a computer. Maintaining a database is often the responsibility of a database management system (DBMS).

RDBMS:

Relational database management systems (RDBMS) allow users to edit, query, and manage databases using a relational model.

SQL:

Abbreviation "SQL" stands for "Structured Query Language" in the technical meaning.Interacting with databases can be done using SQL programming language.

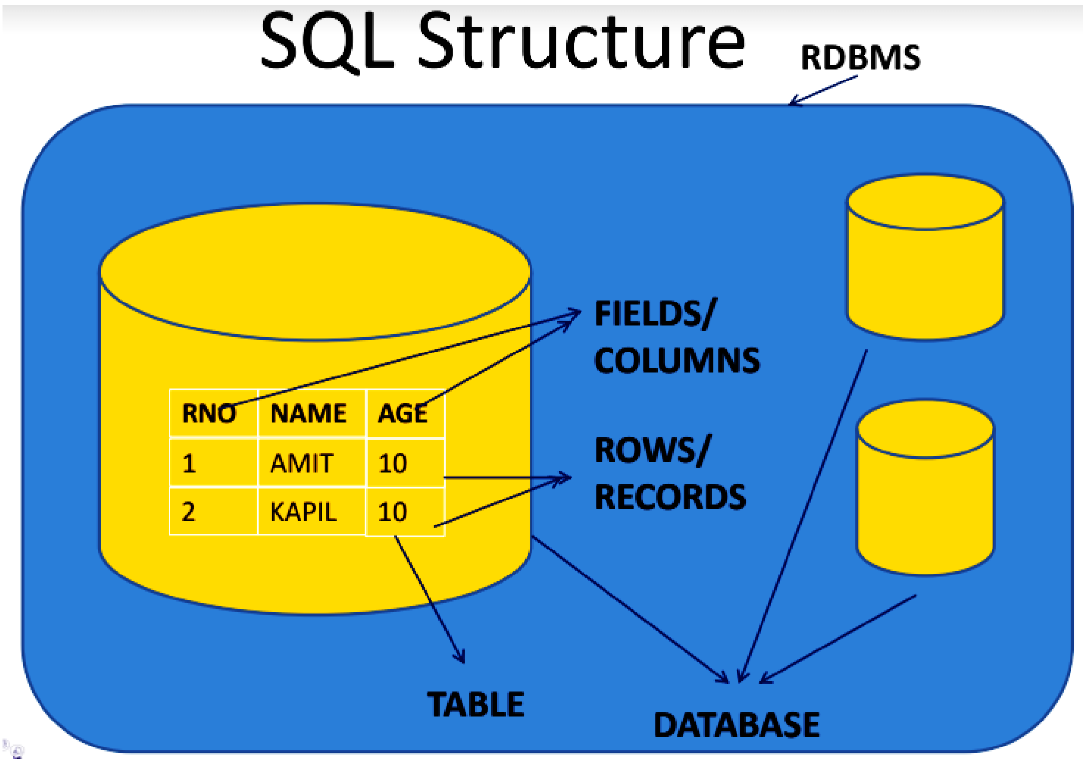
According to the American National Standards Institute, SQL is the standard language for relational database management systems (ANSI).

SQL instructions can be used to alter or obtain data from a database. Some of the most popular SQL-based relational database management systems are Oracle, Sybase, Microsoft SQL Server, Access, and Ingres.

In spite of the fact that most database systems use SQL, many have their own proprietary extensions that are often exclusively utilised on their system. In databases, the most common SQL commands, such as "Select," "Insert" and "Update," can be used for almost any operation. As a bonus, this course includes SQL Interpreter so you may study and practise all of these instructions.

Why SQL?

- RDBMS data may be accessed and updated more easily with SQL.
- Numerous instructions may be executed using SQL, allowing us to carry out a wide range of database-related tasks.
- Additionally, SQL is employed to keep track of the database transactions.
- You can use SQL UNION to combine two different SELECT statements.
- Data can be retrieved by using the SQL JOIN statement, which joins two or more tables.
- Query languages like SQL are often used on interactive websites that house large databases of user and product information. A database is at the heart of every website's backend. The majority of this data is retrieved and stored using SQL.
- Relational databases like Microsoft SQL Server, Oracle database and MYSQL are all compatible with SQL. It also allows for a great deal of customization and control over which tables in the database can be accessed and modified.
- Integration of scripting languages with SQL is simple. It's because of this that SQL is so useful, as it allows for easy queries to handle considerably larger data sets.
- As a result, it is employed by data scientists and analysts, as well as by machine learning enthusiasts.

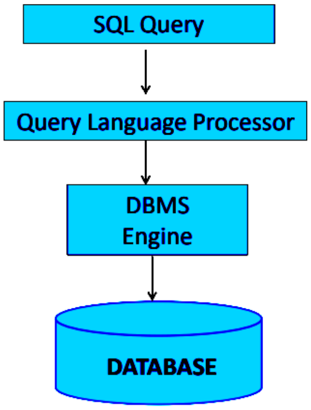


SQL Structure

RNO	NAME	AGE
1	AMIT	10
2	KAPIL	10

Process flow:

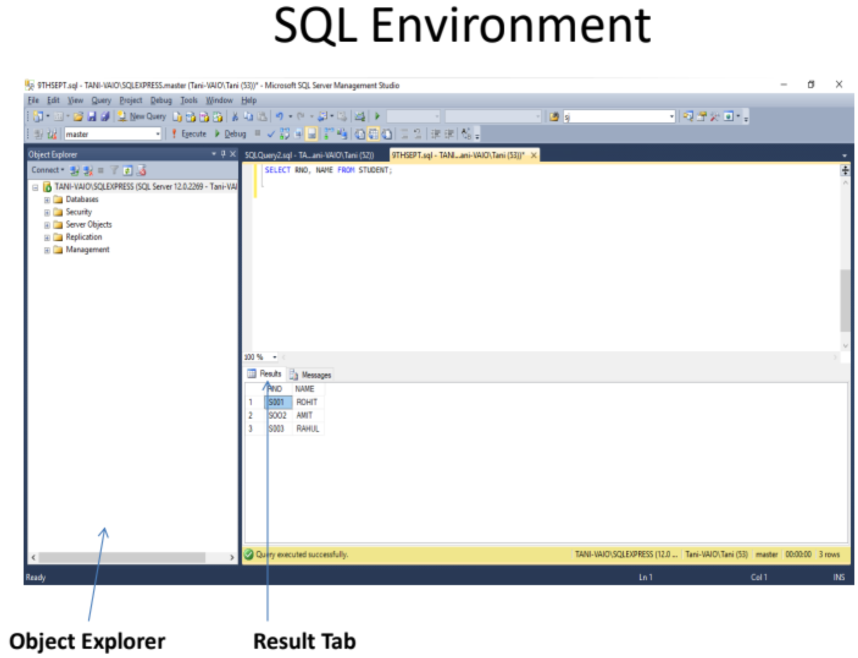
When you run a SQL query, the query is sent to a SQL Server for evaluation. Database management, load balancing, transaction management, and other functions are all performed by SQL Server. Using a physical database table, the SQL server returns the result. If the query and SQL server load are high, this process may take a few minutes.



SQL Environment:

The following are the components in a SQL environment:

- Editor
- Result
- Message
- Object Explorer



SQL Environment

9THSEPT.sql - TANI-VAIO\SQLEXPRESS.master (Tani-VAIO\Tani (53))* - Microsoft SQL Server Management Studio

```
SELECT RNO, NAME FROM STUDENT;
```

```
   RNO   NAME
1  S001  ROHIT
2  SOO2  AMIT
3  S003  RAHUL
```

Query executed successfully.

File Edit View Query Project Debug Tools Window Help
New Query
master
Execute
Debug
Object Explorer
Connect
TANI-VAIO\SQLEXPRESS (SQL Server 12.0.2269 - Tani-VAI
Databases
Security
Server Objects
Replication
Management
SQLQuery2.sql - TA...ani-VAIO\Tani (52))
9THSEPT.sql - TANI...ani-VAIO\Tani (53))*
100 %
Results
Messages
TANI-VAIO\SQLEXPRESS (12.0 ... | Tani-VAIO\Tani (53) | master | 00:00:00 | 3 rows
Ready
Ln 1
Col 1
INS
Object Explorer
Result Tab

Types of SQL Statements:

Data Definition Language (DDL)

Data definition statements are used to define the database structure or table.

Statement	Description
CREATE	Create a new database/table.
ALTER	Modifies the structure of the database/table.
DROP	Deletes a database/table.

Data Manipulation Language (DML)

Data manipulation statements are used for managing data within table objects.

Statement	Description
INSERT	Insert data into a table.
UPDATE	Updates existing data with new data within a table.
DELETE	Deletes the records rows from the table.

Data Query Language(DQL)

Data Query statements are used to select data within the database.

Statement	Description
SELECT	Retrieve data from the table.

Data Control Language (DCL)

Data control statements are used to give privileges to access limited data.

Statement	Description
GRANT	Gives privileges to users for accessing database data.
REVOKE	Take back for given privileges.

RDBMS Concepts:

Create Statement:

Syntax:

CREATE DATABASE Statement
CREATE DATABASE <DATABASE EID>;

Example:

```
create database test;
use test;
```

Syntax:

```
CREATE TABLE Statement
CREATE TABLE table_EID(
column1 datatype (Size),
column2 datatype (Size),
column3 datatype,
…..
.....
columnN data type);
```

Example:

```
CREATE TABLE Statement
CREATE TABLE EMP (
RNO char (5),
EID CHAR (20),
AGE INT ,
CLASS CHAR(10),
);
```

INSERT Statement:

Syntax:

```
INSERT INTO table_EID ( column1,
column2....columnN) VALUES ( value1,
value2....valueN);
```

Example:

```
INSERT INTO TABLE_EID VALUES
(value1,value2,value3,...valueN);
```

SQL SELECT Statement

Syntax:

- SELECT column1, column2....columnN FROM table_EID

Example:

- SELECT * FROM table_EID;

SQL Data Types

Numeric Data Types:

DATA TYPE	FROM	TO
Int	-2,147,483,648	2,147,483,647
Small Int	-32,768	32,767
Decimal	-10^38 +1	10^38 +1
Money	-922,337,203,685,477.5808	+922,337,203,685,477.5807
Float	-1.79E + 308	1.79E + 308

Character Data Types:

DATA TYPE	FROM	TO
Char	Char	Maximum length of 8,000 characters.
Varchar	Varchar	Maximum length of 8,000 characters.
Text	text	maximum length of 2,147,483,647 characters.

Date & Time Data Types:

DATA TYPE	FROM	TO
Datetime	Jan 1, 1753	Dec 31, 9999
smalldatetime	Jan 1, 1900	Jun 6, 2079
Date	Stores a date like June 30, 1991	
Time	Stores a time of day like 12:30 P.M.	

Misc Data Types:

DATA TYPE	FROM	TO
Image	Maximum length of 2,147,483,647 bytes. (Variable length Binary Data)	

SQL Clauses:

SQL WHERE Clause:

```
SELECT column1, column2....columnN FROM
table_EID WHERE CONDITION;
```

SQL LIKEClause:

```
SELECT column1, column2....columnN FROM
table_EID WHERE column LIKE 'XXXX%‘
```

```
SELECT FROM table_EID WHERE column LIKE 'XXXX_'
```

There are two wildcards used in conjunction with the LIKE operator:
1. The percent sign (%)
2. The underscore (_)

SQL TOP Clause:

SELECT TOP number|percent column_EID(s) FROM
table_EID WHERE [condition]

SQL UPDATE Statement:

UPDATE table_EID
SET column1 = value1, column2 = value2
....columnN=valueN
[WHERE CONDITION];

SQL DELETE Statement:

- DELETE FROM table_EID WHERE
{CONDITION};

- DELETE FROM table_EID

- DELETE table_EID

SQL ALTER TABLE Statement:

- ALTER TABLE table_EID
ADD
column_EID {data_type};

- ALTER TABLE table_EID
DROP Column
column_EID ;

- ALTER TABLE table_EID
ALTER Column
column_EID {data_type};

SQL DROP TABLE Statement:

- DROP TABLE table_EID;
- DROP DATABASE database_EID;

SQL TRUNCATE TABLE Statement :

```
TRUNCATE TABLE table_EID;
```

SQL COMMIT Statement:

```
COMMIT;
```

SQL ROLLBACK Statement :

```
ROLLBACK;
```

SQL BETWEEN Clause:

```
SELECT column1, column2....columnN FROM table_EID WHERE column_EID BETWEEN val-1 AND val-2;
```

SQL IN Clause

```
SELECT column1, column2....columnN
FROM table_EID
WHERE column_EID IN (Val1, Val2... Valn);
```

SQL Like Clause

```
SELECT column1, column2....columnN FROM table_EID WHERE column_EID LIKE { PATTERN}
```

SQL COUNT Clause

```
SELECT COUNT(column_EID) FROM table_EID WHERE CONDITION;
```

SQL DISTINCT Clause

```
SELECT DISTINCT (column) FROM table_EID;
```

SQL ORDER BY Clause

```
SELECT column1, column2....columnN
FROM table_EID
WHERE CONDITION
ORDER BY column_EID {ASC|DESC};
```

SQL GROUP BY Clause

```
SELECT SUM(column_EID)
FROM table_EID
WHERE CONDITION
GROUP BY column_EID;
```

SQL HAVING Clause

```
SELECT SUM(column_EID)
FROM table_EID
WHERE CONDITION GROUP BY column_EID
HAVING (arithmetic function condition);
```

SQL Operators

In a SQL statement, an operator is a reserved word or character used to execute operations, such as comparisons and arithmetic calculations, in the WHERE clause of the statement. Operators can be used in a SQL statement to express conditions.

Type of Operators

- Arithmetic Operators (+, -, /, *, %)
- Comparison Operators (=, <>, !=, >, <, >= ,<=, !>, !<)
- Logical Operators (AND, OR, NOT)
- Other Operators (BETWEEN , IN, LIKE, IS NULL, DISTINCT, EXISTS)

Normalization

Database normalisation is the process of arranging data in a database in a way that is both effective and efficient. When saving data, we adhere to a set of rules, guidelines, and assertions.

The process of normalisation has two reasons:

- Data redundancy, such as keeping the same information in multiple tables, can be eliminated.
- Maintaining the integrity of data dependencies.

First Normal Form (1NF)

• Define the data items. This means looking at the data to be stored, organizing the data into columns, defining what type of data each column contains, and finally putting related columns into their own table.
• Ensure that there are no repeating groups of data
• Ensure that there is a primary key.

Second Normal Form (2NF)

• It should meet all the rules for 1NF
• There must be no partial dependencies of any of the columns on the primary key

Third Normal Form (3NF)

• It should meet all the rules for 2NF
• Tables should have relationships.

SQL Constraints:

A table constraint is a set of restrictions for a column of data in the table. Using these, you can restrict the type of data that can be entered into the table. This ensures the database's accuracy and reliability.

Following are commonly used constraints available in SQL:

- PRIMARY
- UNIQUE
- NOT NULL
- DEFAULT
- CHECK
- FOREIGN

NOT NULL Constraint:

The default value for a column is NULL. Constraints are needed to prevent a column from having a NULL value, so we must define such a constraint on this column.

```
CREATE TABLE SALES(
ID INT NOT NULL,
EID VARCHAR (20) NOT NULL,
AGE INT NOT NULL,
ADDRESS CHAR (25) ,
SALARY DECIMAL (18, 2)
);

ALTER TABLE SALES
ALTER COLUMN SALARY DECIMAL (18, 2) NOT NULL;
```

DEFAULT Constraint:

The DEFAULT constraint provides a default value to a column when the INSERT INTO statement does not provide a specific value

```
CREATE TABLE SALES(
ID INT NOT NULL,
EID VARCHAR (20) NOT NULL,
AGE INT NOT NULL,
ADDRESS CHAR (25) ,
SALARY DECIMAL (18, 2) DEFAULT 5000.00
);
ALTER TABLE SALES
ADD CONSTRAINT DSAL DEFAULT 5000.00 FOR SALARY;
ALTER TABLE SALES
DROP CONSTRAINT DSAL;
```

UNIQUE Constraint:

The UNIQUE constraint provides a unique value to a column.

```
CREATE TABLE SALES(
ID INT NOT NULL,
EID VARCHAR (20) NOT NULL,
AGE INT NOT NULL UNIQUE,
ADDRESS CHAR (25) ,
SALARY DECIMAL (18, 2) DEFAULT 5000.00
);
ALTER TABLE SALES
ADD CONSTRAINT <CONSTRAINT EID > UNIQUE (AGE);
ALTER TABLE SALES
ADD CONSTRAINT myUniqueConstraint UNIQUE(AGE, SALARY);
ALTER TABLE SALES
DROP CONSTRAINT myUniqueConstraint;
```

CHECK Constraint:

The CHECK Constraint enables a condition to check the value being entered into a record. If the condition evaluates to false, the record violates the constraint and it s not entered the table.

```
CREATE TABLE SALES(
ID INT NOT NULL,
EID VARCHAR (20) NOT NULL,
AGE INT NOT NULL CHECK (AGE > 18),
ADDRESS CHAR (25) ,
SALARY DECIMAL (18, 2) DEFAULT 5000.00
);

ALTER TABLE SALES
ADD CONSTRAINT ckAge CHECK (AGE > 18);

ALTER TABLE SALES
DROP CONSTRAINT ckAge;
```

PRIMARY KEY Constraint:

A primary key is a field in a table which uniquely identifies the each rows/records in a database table. Primary keys must contain unique values. A primary key column cannot have NULL values.

A table can have only one primary key which may consist of single or multiple fields. When multiple

fields are used as a primary key, they are called a composite key.

```
CREATE TABLE SALES(
ID INT NOT NULL,
EID VARCHAR (20) NOT NULL,
AGE INT NOT NULL,
ADDRESS CHAR (25) ,
SALARY DECIMAL (18, 2) ,
PRIMARY KEY (ID)
);

ALTER TABLE SALES
ADD CONSTRAINT pkID PRIMARY KEY (ID);
ALTER TABLE SALES
```

```
DROP CONSTRAINT pkID;
```

FOREIGN KEY Constraint:

A foreign key is a key used to link two tables together. This is sometimes called a referencing key.

```
CREATE TABLE SALES(
ID INT NOT NULL,
EID VARCHAR (20) NOT NULL,
AGE INT NOT NULL,
ADDRESS CHAR (25) ,
PRIMARY KEY (ID)
);
CREATE TABLE ORDERS(
OID INT NOT NULL,
CUST_ID INT REFERENCES SALES (ID),
ODATE DATE,
QTY INT,
PRICE INT
);
ALTER TABLE ORDERS
ADD CONSTRAINTS FKID FOREIGN KEY (CUST_ID) REFERENCES SALES (ID);
```