Data Definition Language SQL Data Types

.NET

*A software system used to maintain a relational database is called a Relational Database Management System (RDBMS). Many relational database systems use **Structured Query Language (SQL)** for querying and maintaining a database.*
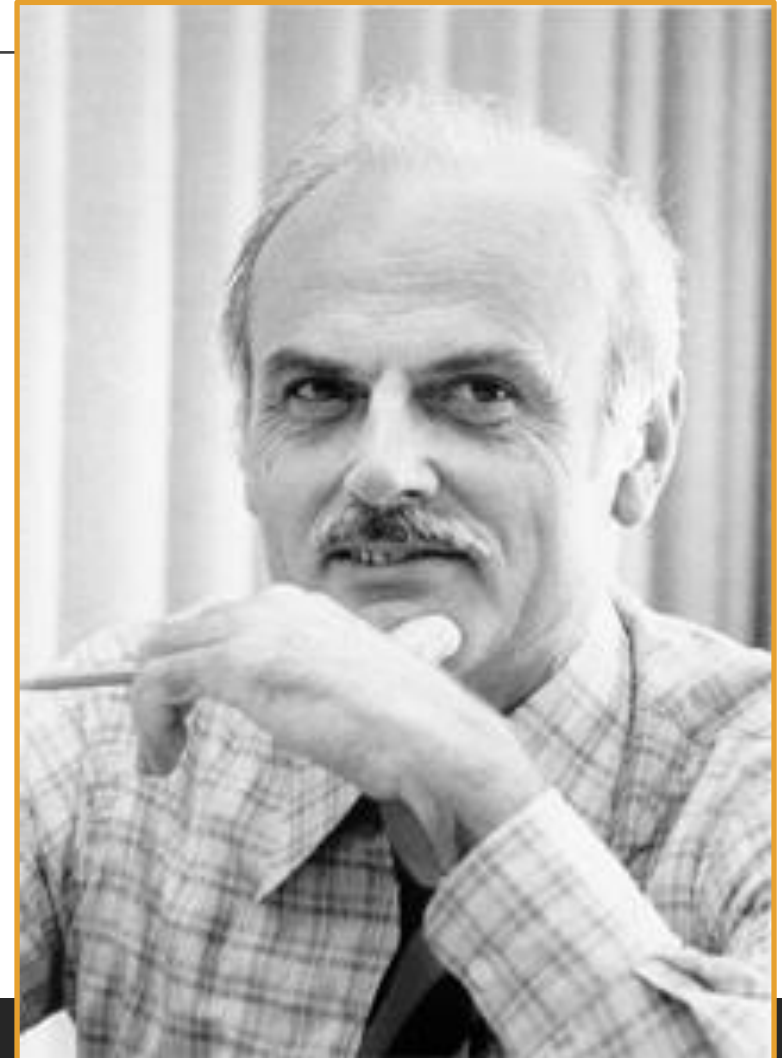
(RDBMS) Relational Database Management System – History

Relational databases are based on the relational model of data, as proposed by E. F. Codd in 1970.

Edgar Frank "Ted" Codd (August 23, 1923 – April 18, 2003) was a British computer scientist and winner of the 1981 Turing Award.
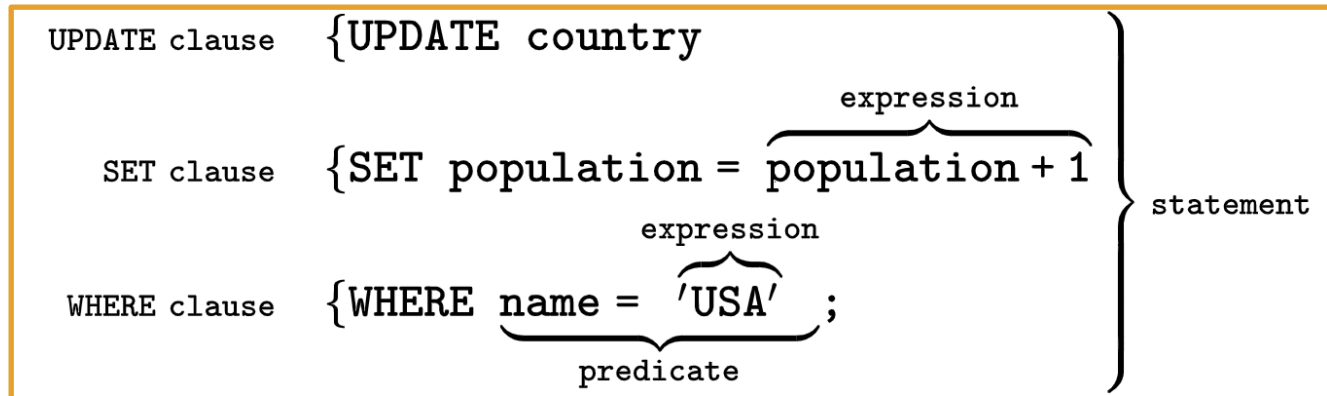
SQL (Structured Query Language)

SQL was originally based upon relational algebra and tuple relational calculus. SQL is a declarative language. We say what data we want, not how to get it. We cannot manage <u>how</u> SQL obtains the data.
The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control.

SQL consists of two main types of statements:

- Data Definition Language (DDL)
- Data Manipulation Language (DML).

```
UPDATE clause    {UPDATE country
                                            expression
    SET clause   {SET population = population + 1
                                expression
   WHERE clause  {WHERE name =  'USA'  ;
                                predicate
                                                    } statement
```
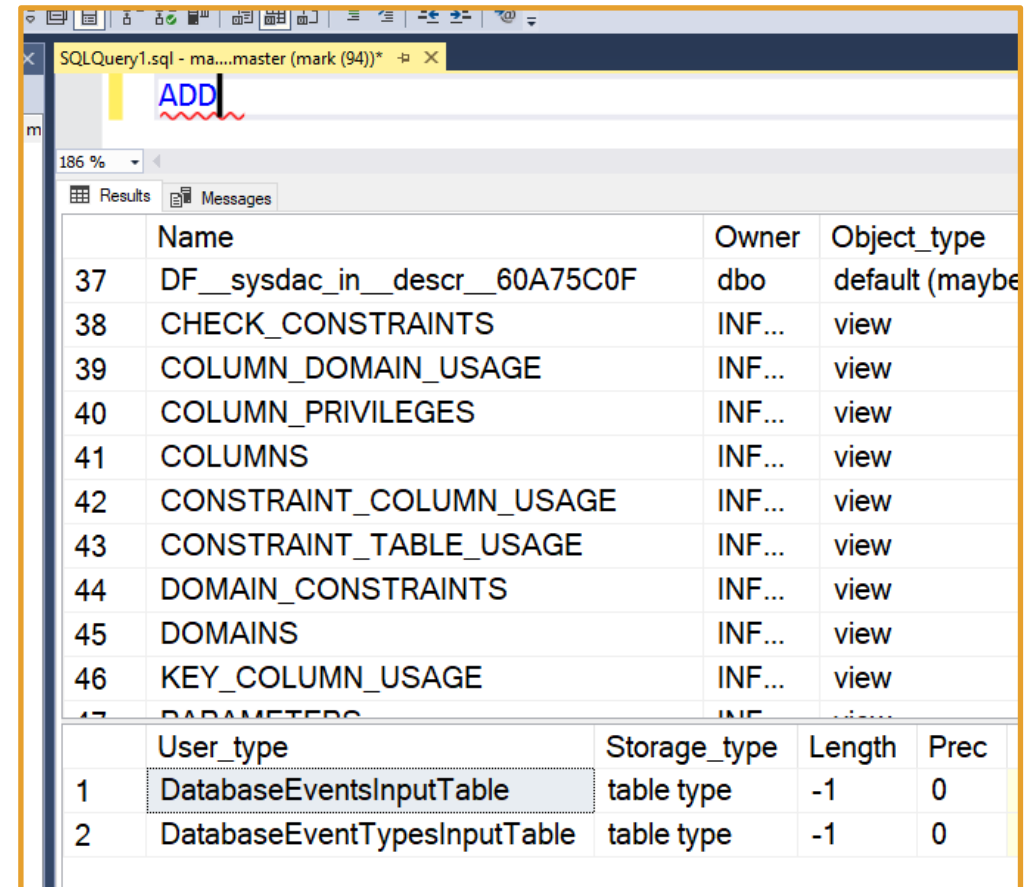
Microsoft and T-SQL

https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15#tools-that-use-t-sql

T-SQL is central to using Microsoft SQL products and services. All tools that communicate with a SQL database send T-SQL commands. SQL works on top of T-SQL.

Some of the Microsoft tools that issue T-SQL commands are:
- SQL Server Management Studio (SSMS)
- SQL Server Data Tools (SSDT)
- Azure Data Studio

*You can type a T-SQL keyword in the SSMS Query Editor window and press F1 to get data about any T-SQL Keyword.
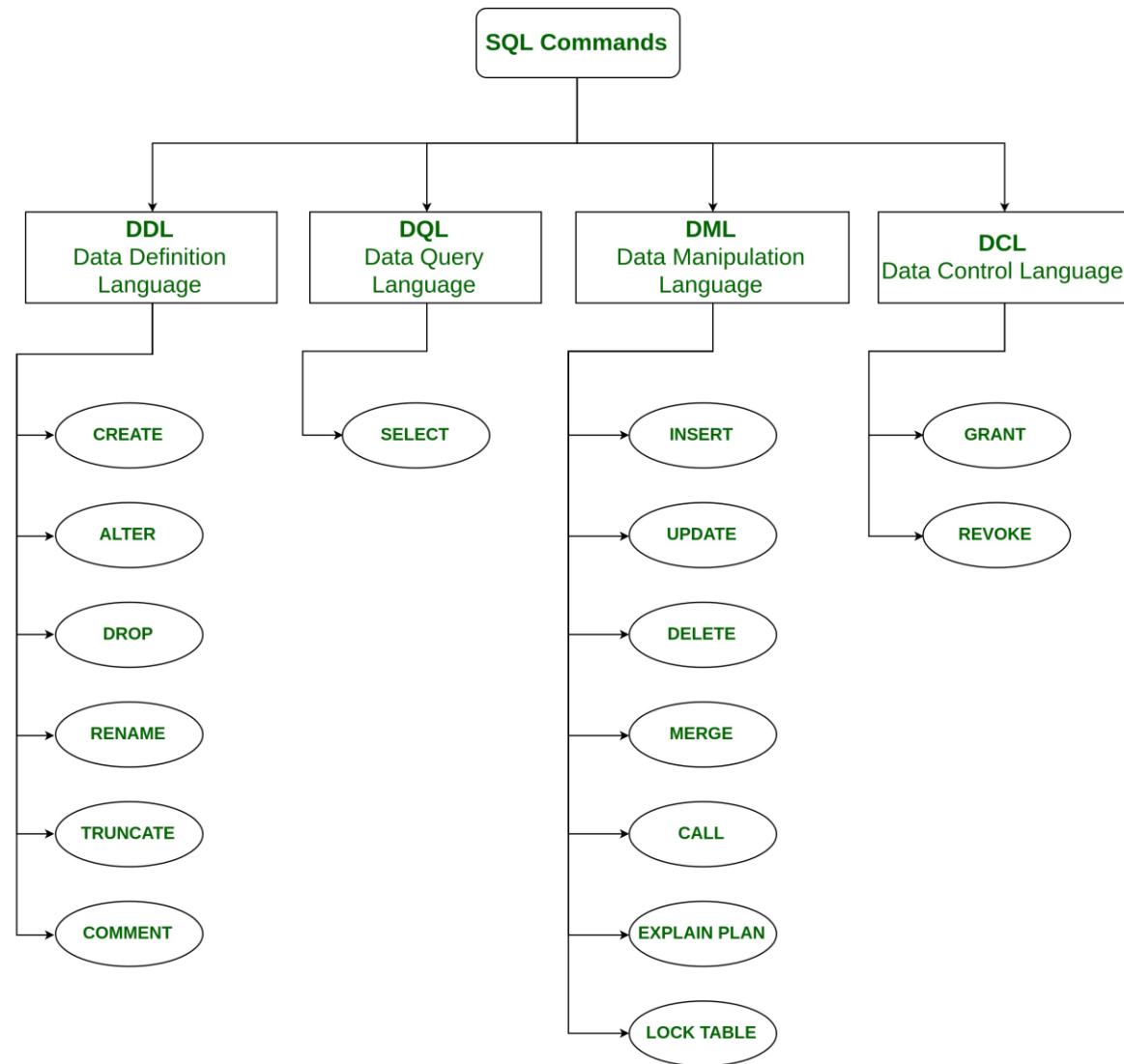
Types of SQL Commands

SQL Commands

- **DDL**
  Data Definition Language
  - CREATE
  - ALTER
  - DROP
  - RENAME
  - TRUNCATE
  - COMMENT

- **DQL**
  Data Query Language
  - SELECT

- **DML**
  Data Manipulation Language
  - INSERT
  - UPDATE
  - DELETE
  - MERGE
  - CALL
  - EXPLAIN PLAN
  - LOCK TABLE

- **DCL**
  Data Control Language
  - GRANT
  - REVOKE

Data Definition Language

Data Definition Language (DDL) statements define the structure of the DB. DDL statements create, alter, or drop the data structures (tables) of a database.

- ALTER - Modifies a table definition by altering, adding, or dropping columns and constraints.
- CREATE - creates a new database
- DROP - Removes one or more table definitions and all data, indexes, triggers, constraints, and permission specifications for those tables.

SQL – *Create* and *Drop* an empty DB

CREATE DATABASE *databasename*;

DROP DATABASE *databasename*;

*Deleting a database will result in complete loss of information stored in the database!

SQL – *Create* and *Drop* a table

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ....
);
```

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

```
DROP TABLE table_name;
```

```
DROP TABLE Shippers;
```

*Deleting a table will result in complete loss of information stored in the table!

SQL with SQL Server

In SQL Server, every table must be in a schema.

```
CREATE SCHEMA Poke;
GO
```

```
--CREATE TABLE Poke.Pokemon;
CREATE TABLE Poke.Pokemon (
    PokemonId INT NOT NULL IDENTITY(1000, 1),
    Name NVARCHAR(50) NOT NULL,
    Height DECIMAL(6,2) NULL,
    TypeId INT NOT NULL FOREIGN KEY REFERENCES Poke.Type (TypeId),
    DateModified DATETIME2 NOT NULL DEFAULT (GETDATE()),
    CONSTRAINT CK_Height_Nonnegative CHECK (Height IS NULL OR Height >= 0)
);
```

Create a Table in SQL Server

To create a table, you must provide:

1. a name for the table
2. name of each column
3. data type of each column
4. a unique primary key.

```
CREATE TABLE dbo.Products
   (ProductID int PRIMARY KEY NOT NULL,
   ProductName varchar(25) NOT NULL,
   Price money NULL,
   ProductDescription varchar(max) NULL)
GO
```

```
CREATE TABLE dbo.Products
   (ProductID int PRIMARY KEY NOT NULL,
   ProductName varchar(25) NOT NULL,
   Price money NULL,
   ProductDescription varchar(max) NULL)
GO
```

SQL – String Data Types

Data Type	Description
CHAR(n)	Fixed-length up to n, 0 to 255, Default 1
VARCHAR(n)	Variable length up to n. 0 to 65535
NCHAR(n)	Fixed-length, Unicode string
NVARCHAR(n)	variable-length Unicode string. (Use this unless you nee to use something else)

There are a many functions to deal with strings:
- LEN(), SUBSTRING(), CHARINDEX(), REPLACE(), LOWER(), UPPER()

SQL – Integer Data Types

Data Type	Description
BIT	It's a bit. Values 1 to 64. Default 1
TINYINT	Signed = -128 to 127. unsigned = 0 to 255
BOOL	Max 255 bytes. 0 = false, 1 = true
SMALLINT	Max 65535 bytes
INT	signed = 2147483648 to 2147483647. unsigned = 0 to 4294967295 (Use this unless you need something else)

SQL – Float Data Types

Data Type	Description
FLOAT(size,d)	size = total digits. d = number of digits after the decimal point
FLOAT(p)	If *p* is from 0 to 24, the data type becomes FLOAT(). If *p* is from 25 to 53, the data type becomes DOUBLE()
DOUBLE(size, d)	size = total digits. d = number of digits after the decimal point.
DECIMAL(*size, d*)	An exact fixed-point number. size = total digits(default 10, max 65). d = number of digits after the decimal point(default 0, max 30).

SQL – Date and Time Data Types

https://docs.microsoft.com/en-us/sql/t-sql/data-types/date-and-time-types?view=sql-server-ver15

Data Type	Description
DATE	Format: YYYY-MM-DD. From '1000-01-01' to '9999-12-31'
DATETIME(fsp)	Format: YYYY-MM-DD hh:mm:ss. Add *DEFAULT* and *ON UPDATE* in the column definition to get automatic initialization and updating to the current date and time
TIMESTAMP(fsp)	The number of seconds since the Unix epoch. Format: YYYY-MM-DD hh:mm:ss. Automatic initialization and updating with DEFAULT CURRENT_TIMESTAMP and ON UPDATE CURRENT_TIMESTAMP in the column definition.
TIME(fsp)	hh:mm:ss. From '-838:59:59' to '838:59:59'
YEAR	1901 to 2155, and 0000. MySQL 8.0 does not support year in two-digit format.
DATETIMEOFFSET	For storing intervals of time. Use *YEAR()* to extract parts of the dates/times, DATEPART(YEAR FROM '2019-01-01') or DATEPART(YEAR, '2019-01-01')

SQL – Currency Data Types

Data Type	Description
MONEY	From -922,337,203,685,477.5808 to 922,337,203,685,477.5807 prints with '$'
SMALLMONEY	From -214,748.3648 to 214,748.3647 prints with '$'

SQL – Attribute Constraints

https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-table-table-constraint-transact-sql?view=sql-server-ver15

NOT NULL	column does not accept NULL as a value
NULL	column explicitly accepts NULL as a value. (NULL will be the default value)
PRIMARY KEY	value must be unique within this column
UNIQUE	implies NOT NULL and UNIQUE, and by default sets a *CLUSTERED INDEX*.
FOREIGN KEY	by default sets a NONCLUSTERED INDEX
CHECK	enforces that some expression is true for every row
DEFAULT	configures a default value for that column
IDENTITY	this sets up an auto-incrementing default, AND prevents anyone from inserting their own value

ALTER Table

ALTER TABLE

- modifies a table definition by altering, adding, or dropping columns and constraints.
- reassigns and rebuilds partitions or disables and enables constraints and triggers.

```
ALTER TABLE Poke.Pokemon ADD
     Active BIT NOT NULL DEFAULT 1;
```

```
ALTER TABLE Addresses
DROP COLUMN ZipCode;
```

Definitions

Clustered / NonClustered Index - A clustered index defines the order in which data is physically stored in a table. Table data can be sorted in only way, therefore, there can be only one clustered index per table. In SQL Server, the primary key constraint automatically creates a clustered index on that particular column.