# Data-Definition Language

The SQL DDL (Data Definition Language) allows specification of not only a set of relations, but also the following information for each relation:

* The schema for each relation.
* The domain of values associated with each attribute.
* Integrity constraints.
* The set of indices for each relation.
* Security and authorization information.
* Physical storage structure on disk.

**Domain Types in SQL**

1. The SQL standard supports a variety of built-in domain types:
   * **char**(n) (or **character**(n)): fixed-length character string, with user-specified length.
   * **varchar**(n) (or **character varying**): variable-length character string, with user-specified maximum length.
   * **int** or **integer**: an integer (length is machine-dependent).
   * **smallint**: a small integer (length is machine-dependent).
   * **numeric**(*p, d*): a fixed-point number with user-specified precision, consists of *p* digits (plus a sign) and *d* of *p* digits are to the right of the decimal point. E.g., **numeric**(*3, 1*) allows 44.5 to be stored exactly but not 444.5.
   * **real** or **double precision**: floating-point or double-precision floating-point numbers, with machine-dependent precision.
   * **float**(n): floating-point, with user-specified precision of at least *n* digits.
   * **date**: a calendar date, containing four digit year, month, and day of the month.
   * **time**: the time of the day in hours, minutes, and seconds.
2. SQL allows arithmetic and comparison operations on various numeric domains, including, **interval** and *cast* (*type coercion*) such as transforming between *smallint* and *int*. It considers strings with different length are compatible types as well.
3. SQL allows **create domain** statement, e.g.,

**create domain** *person-name* **char**(20)

# Data Manipulation Language (DML)

1. **Data Manipulation** is:
   * **retrieval** of information from the database
   * **insertion** of new information into the database
   * **deletion** of information in the database
   * **modification** of information in the database
2. A DML is a language which enables users to access and manipulate data.

The goal is to provide efficient human interaction with the system.

1. There are two types of DML:
   * **procedural**: the user specifies what data is needed and how to get it
   * **nonprocedural**: the user only specifies what data is needed
     + Easier for user
     + May not generate code as efficient as that produced by procedural languages
2. A **query language** is a portion of a DML involving information retrieval only. The terms DML and query language are often used synonymously.

## The SQL SELECT Statement

The SELECT statement is used to select data from a database.

The result is stored in a result table, called the result-set.

### SQL SELECT Syntax

SELECT column\_name,column\_name  
FROM table\_name;

and

SELECT \* FROM table\_name;

## SELECT Column Example

The following SQL statement selects the "CustomerName" and "City" columns from the "Customers" table:

### Example

SELECT CustomerName,City FROM Customers;

## The SQL WHERE Clause

The WHERE clause is used to extract only those records that fulfill a specified criterion.

The WHERE clause is used to filter records.

### SQL WHERE Syntax

SELECT column\_name,column\_name  
FROM table\_name  
WHERE column\_name operator value;

## WHERE Clause Example

The following SQL statement selects all the customers from the country "Mexico", in the "Customers" table:

### Example

SELECT \* FROM Customers  
WHERE Country='Mexico';

## The SQL LIKE Operator

The LIKE operator is used to search for a specified pattern in a column.

### SQL LIKE Syntax

SELECT column\_name(s)  
FROM table\_name  
WHERE column\_name LIKE pattern;

## SQL LIKE Operator Examples

The following SQL statement selects all customers with a City starting with the letter "s":

### Example

SELECT \* FROM Customers  
WHERE City LIKE 's%';