

3.4 Data Manipulation Language (DML)

Data Manipulation Language (DML) is used to manipulate (select, insert, update, delete) data.

3.4.1 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. The SQL SELECT syntax:

```
SELECT column_name(s) FROM table_name
```

```
SELECT * FROM table_name
```

3.4.2 The SQL SELECT DISTINCT Statement

In a table, some of the columns may contain duplicate values. This is not a problem; however, sometimes you will want to list only the different (distinct) values in a table.

The DISTINCT keyword can be used to return only distinct (different) values. The syntax is

```
SELECT DISTINCT column_name(s) FROM table_name
```

3.4.3 The WHERE Clause

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

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name operator value
```

Example:

```
SELECT * FROM Persons WHERE City='Sandnes'
```

Note: SQL uses single quotes around text values (most database systems will also accept double quotes). Although, numeric values should not be enclosed in quotes.

The operators allowed in the WHERE clause are:

Operator	Description
=	Equal
<>	Not equal
>	Greater than

<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	If you know the exact value you want to return for at least one of the columns

3.4.4 The AND & OR Operators

The AND operator displays a record if both the first condition and the second condition is true.

Example

```
SELECT * FROM Persons WHERE FirstName='Tove' AND LastName='Svendson'
```

This will select only the persons with the first name equal to "Tove" AND the Last name equal to "Svendson":

The OR operator displays a record if either the first condition or the second condition is true.

Example

```
SELECT * FROM Persons WHERE FirstName='Tove' OR FirstName='Ola'
```

This will select only the persons with the first name equal to "Tove" OR the first name equal to "Ola":

3.4.5 The ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set by a specified column. It sorts the records in ascending order by default. If you want to sort the records in a descending order, you can use the DESC keyword. The syntax is:

```
SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC
```

3.4.6 The INSERT INTO Statement

The INSERT INTO statement is used to insert a new row in a table. The syntax is:

```
INSERT INTO table_name VALUES (value1, value2, value3,...)
```

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

3.4.7 The UPDATE Statement

The UPDATE statement is used to update existing records in a table. The SQL UPDATE syntax is:

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

Note: Notice the WHERE clause in the UPDATE syntax. The WHERE clause specifies which record or records that should be updated. If you omit the WHERE clause, all records will be updated!

3.4.8 The DELETE Statement

The DELETE statement is used to delete rows in a table. The SQL DELETE Syntax is:

```
DELETE FROM table_name
WHERE some_column=some_value
```

Note: Notice the WHERE clause in the DELETE syntax. The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted!

Activity B

Use the customer table shown in table 7.6 to answer the following SQL statements, displaying the resulting record sets

- i. SELECT * FROM customers
- ii. SELECT CompanyName, ContactName FROM customers
- iii. SELECT * FROM customers WHERE companyname LIKE 'a%'
- iv. SELECT CompanyName, ContactName FROM customers WHERE CompanyName > 'a'
- v. SELECT CompanyName, ContactName FROM customers WHERE CompanyName > 'g' AND ContactName > 'g'

Table 7.6: Customers table
Source: Microsoft Northwind database sample

CompanyName	ContactName	Address	City
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin

Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå
Centro comercial Moctezuma	Francisco Chang	Sierras de Granada 9993	México D.F.
Ernst Handel	Roland Mendel	Kirchgasse 6	Graz
FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	C/ Morazarzal, 86	Madrid
Galería del gastrónomo	Eduardo Saavedra	Rambla de Cataluña, 23	Barcelona
Island Trading	Helen Bennett	Garden House Crowther Way	Cowes
Königlich Essen	Philip Cramer	Maubelstr. 90	Brandenburg
Laughing Bacchus Wine Cellars	Yoshi Tannamuri	1900 Oak St.	Vancouver
Magazzini Alimentari Riuniti	Giovanni Rovelli	Via Ludovico il Moro 22	Bergamo

3.4.9 The LIKE Operator

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

```
SELECT column_name(s)
FROM table_name
WHERE column_name LIKE pattern
```

Example: If we want to select the persons living in a city that starts with "s" from the table 7.6; We use the following SELECT statement:

```
SELECT * FROM Persons WHERE City LIKE 's%'
```

The "%" sign can be used to define wildcards (missing letters in the pattern) both before and after the pattern.

3.4.10 The IN Operator

The IN operator allows you to specify multiple values in a WHERE clause. The syntax is:

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1,value2,...)
```

Example: if we want to select the persons with a last name equal to "Hansen" or "Pettersen" from the Persons table; We use the following SELECT statement:

```
SELECT * FROM Persons
WHERE LastName IN ('Hansen','Pettersen')
```

3.4.11 The BETWEEN Operator

The BETWEEN operator selects a range of data between two values. The values can be numbers, text, or dates. The SQL BETWEEN Syntax is:

```
SELECT column_name(s)
FROM table_name
WHERE column_name
BETWEEN value1 AND value2
```

Example: If we want to select the persons with a last name alphabetically between "Hansen" and "Pettersen" from the persons table. We use the following SELECT statement:

```
SELECT * FROM Persons
WHERE LastName
BETWEEN 'Hansen' AND 'Pettersen'
```