CREATE DATABASE

- The CREATE DATABASE statement is used to create a database.

```
CREATE DATABASE database_name
```

- CREATE DATABASE Example

```
CREATE DATABASE my_db
```

- Create a Database According to Name_ID

CREATE TABLE

- Create a Table named Person like below where attribute's data type will be varchar(200) for last 4 column and for P_Id data type will be int

P_Id	LastName	FirstName	Address	City

CREATE TABLE

- The CREATE TABLE statement is used to create a table in a database.

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
column_name3 data_type,
....
);
```

SQL Server Data Types

Data type	Description
varchar(n)	Variable-length character string. Maximum 8,000 characters
int	Allows whole numbers between -2,147,483,648 and 2,147,483,647
float(n)	Floating precision number data from -1.79E + 308 to 1.79E + 308.
date	Store a date only. From January 1, 0001 to December 31, 9999
image	Variable-length binary data. Maximum 2GB
timestamp	Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable

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

INSERT data into Database

- The INSERT INTO statement is used to insert a new row in a table.
- Please remember Values are in single inverted commas'.
- The first form doesn't specify the column names where the data will inserted, only their values:

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

INSERT data into Database

- The second form specifies both the column names and the values to be inserted:

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

INSERT data into Database

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Tove	Bakken 2	
5	Tjessem	Jakob		

Insert Data Only in Specified Columns

It is also possible to only add data in specific columns.

The following SQL statement will add a new row, but only add data in the "P_Id", "LastName" and the "FirstName" columns:

```
INSERT INTO Persons (P_Id, LastName, FirstName)
VALUES (5, 'Tjessem', 'Jakob')
```

The "Persons" table will now look like this:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Johan	Bakken 2	Stavanger
5	Tjessem	Jakob		

SELECT Statement

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

```
SELECT column_name(s)
FROM table_name

Or

SELECT * FROM table_name
```

- select the content of the columns named "LastName" and "FirstName" from the table named Person.
- select all the content of the table named Person

SELECT * Example

Now we want to select all the columns from the "Persons" table.

We use the following SELECT statement:

```
SELECT * FROM Persons
```

Tip: The asterisk (*) is a quick way of selecting all columns!

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

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.
- Syntax

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

- Now we want to select only the distinct values from the column name "City" from the table named Person

SELECT DISTINCT Example

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

Now we want to select only the distinct values from the column named "City" from the table above.

We use the following SELECT statement:

```
SELECT DISTINCT City FROM Persons
```

The result-set will look like this:

City
Sandnes
Stavanger

WHERE Clause

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

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

- Select only the persons living in the city "Sandnes" from the table above

WHERE Clause

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	To specify multiple possible values for a column

WHERE Clause Example

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

Now we want to select only the persons living in the city "Sandnes" from the table above.

We use the following SELECT statement:

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

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes

Quotes Around Text Fields

SQL uses single quotes around text values (most database systems will also accept double quotes).

However, numeric values should not be enclosed in quotes.

For text values:

```
This is correct:

SELECT * FROM Persons WHERE FirstName='Tove'

This is wrong:

SELECT * FROM Persons WHERE FirstName=Tove
```

For numeric values:

```
This is correct:

SELECT * FROM Persons WHERE Year=1965

This is wrong:

SELECT * FROM Persons WHERE Year='1965'
```

The AND & OR Operators

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

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

AND Operator Example

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

Now we want to select only the persons with the first name equal to "Tove" AND the last name equal to "Svendson":

We use the following SELECT statement:

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

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
2	Svendson	Tove	Borgvn 23	Sandnes

OR Operator Example

Now we want to select only the persons with the first name equal to "Tove" OR the first name equal to "Ola":

We use the following SELECT statement:

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

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes

Combining AND & OR

You can also combine AND and OR (use parenthesis to form complex expressions).

Now we want to select only the persons with the last name equal to "Svendson" AND the first name equal to "Tove" OR to "Ola":

We use the following SELECT statement:

```
SELECT * FROM Persons WHERE
LastName='Svendson'
AND (FirstName='Tove' OR FirstName='Ola')
```

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
2	Svendson	Tove	Borgvn 23	Sandnes

ALTER TABLE Statement

- The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.
- To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
ADD column_name datatype
```

ALTER TABLE Statement

- To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name
DROP COLUMN column_name
```

- To change the data type of a column in a table, use the following syntax:

```
ALTER TABLE table_name
ALTER COLUMN column_name datatype
```

ALTER TABLE Statement

- Add a column named “Age" in the "Persons" table
- Change the data type “int” of the column named “age" in the "Persons" table.
- we want to delete the column named “Age" in the "Persons" table.

UPDATE Statement

- The UPDATE statement is used to update existing records in a table.

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

- update the person "Tjessem, Jakob" in the "Persons" table, set his address 'Nissestien 67' and City 'sandnes'
- Please remember if you forget to give where clause here , disaster will be happened in DB

The UPDATE Statement

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

SQL UPDATE Syntax

```
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!

```
UPDATE Persons
SET Address='Nissestien 67', City='Sandnes'
WHERE LastName='Tjessem' AND FirstName='Jakob'
```

The "Persons" table will now look like this:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Johan	Bakken 2	Stavanger
5	Tjessem	Jakob	Nissestien 67	Sandnes

DELETE Statement

- The DELETE statement is used to delete rows in a table.

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

- Delete the person "Tjessem, Jakob" in the "Persons" table.
- Please remember if you forget to give where clause here , disaster will be happened in DB

SELECT INTO Statement

- The SELECT INTO statement selects data from one table and inserts it into a different table.
- The SELECT INTO statement is most often used to create backup copies of tables.
- We can select all columns into the new table:

```
SELECT *
INTO new_table_name
FROM old_tablename
```

- Or

```
SELECT column_name(s)
INTO new_table_name
FROM old_tablename
```

DROP TABLE Statement

- The DROP TABLE statement is used to delete a table.

```
DROP TABLE table_name
```

- Drop the table named Person_custom

TRUNCATE TABLE Statement

- What if we only want to delete the data inside the table, and not the table itself?
- Then, use the TRUNCATE TABLE statement:

```
TRUNCATE TABLE table_name
```

- Truncate the table named Person_Backup