

Bahan Ajar Perekaman Data

Materi Pelengkap Modul

Diklat Fungsional Pranata Komputer
Tingkat Terampil
Kementerian Keuangan
Angkatan 2 Tahun 2019

Jimmy Ludin, SST., M.Si.

Pusat Pendidikan dan Pelatihan
Badan Pusat Statistik
Tahun 2019

KONSEP DASAR PEMBUATAN DATABASE

Database merupakan kumpulan data yang saling berhubungan yang disimpan sedemikian rupa tanpa adanya redundancy (perulangan) yang tidak perlu. Database terbentuk dari sekelompok data-data yang memiliki jenis atau sifat yang sama. Contohnya: data mahasiswa, data dosen, data mata kuliah, data karyawan, dll. Untuk dapat membuat suatu database yang baik, diperlukan suatu rancangan yang tepat dan sesuai kebutuhan. Konsep dasar dalam pembuatan database antara lain:

a. Entitas dan Relationship

Hal yang sangat mendasar dan harus dipahami adalah pemodelan entitas dan relationship. Entitas adalah berbagai hal dalam dunia nyata yang informasinya disimpan dalam database. Sebagai contoh, kita dapat menyimpan informasi tentang dosen yang mengajar mahasiswa. Dalam hal ini, dosen dan mahasiswa merupakan entitas.

Relationship hubungan antara entitas. Sebagai contoh, dosen mengajar mahasiswa. Mengajar merupakan relationship antara entitas dosen dan entitas mahasiswa.

Relationship terdiri dari 3 derajat yang berbeda, yaitu:

- 1. One-to-one menghubungkan secara tepat dua entitas dengan satu kunci (key). Misalnya seorang mahasiswa memiliki satu dosen wali.*
- 2. One-to-many (many-to-one) merupakan hubungan antar entitas dimana kunci (key) pada satu tabel muncul berkali-kali pada tabel lainnya. Misalnya banyak mahasiswa memiliki satu dosen wali.*
- 3. Many-to-many merupakan hubungan antar entitas dimana kunci utama pada tabel pertama dapat muncul beberapa kali pada tabel kedua, dan sebaliknya. Misalnya seorang mahasiswa dapat mengambil banyak matakuliah, dan satu matakuliah bias dimabil oleh banyak mahasiswa.*

b. Relasi atau Tabel

Relasi merupakan tabel yang mewakili entitas, dimana didalamnya terdapat kolom-kolom yang merupakan attribute dari entitas. Penamaan tabel juga disesuaikan dengan nama dari entitas agar mudah dipahami. Contoh tabel pegawai merupakan tabel yang mewakili entitas pegawai.

c. Kolom atau Attribute

Kolom atau attribute merupakan bagian dari tabel yang mewakili ciri dari suatu entitas. Contoh entitas mahasiswa. Maka attribute yang ada antara lain NIM, nama, dan kelas.

d. Kunci (key)

Kunci atau key atau primary key merupakan suatu nilai dalam sebuah tabel yang dapat digunakan untuk mengidentifikasi suatu baris dalam tabel. Contoh dalam tabel mahasiswa, didalamnya terdapat kolom NIM, nama, dan kelas. Untuk mengidentifikasi suatu baris dalam tabel, maka cukup menggunakan NIM.

RELASI ANTAR TABEL

A. Kunci (Key)

Sebelum membahas mengenai relasi, faktor penting yang menyangkut relasi antar table adalah kunci (key). Kunci (key) merupakan satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (record) dalam tabel secara unik. Macam-macam kunci dibedakan menjadi:

1. Kunci Primer

Kunci Primer merupakan sebuah field pada tabel yang merupakan identitas bagi field-field lainnya. Kunci Primer ini tidak boleh memiliki record kembar. Kunci Primer biasanya berada pada tabel induk

2. Kunci Sekunder

Kunci Sekunder merupakan kebalikan dari kunci primer. Kunci Sekunder biasanya berada pada tabel anak. Kedua kunci ini saling berhubungan, dan karena saling berhubungan maka tipe data yang dipakai harus sama.

B. Jenis-Jenis Relasi

Secara garis besar, jenis-jenis relasi dibagi menjadi empat, yaitu:

1. One – To One (Satu Ke Satu)

Relasi ini adalah relasi yang menghubungkan sebuah record pada table induk ke tepat sebuah record pada table anak, akan tetapi relasi ini relative jarang dipergunakan karena jika record pada table anak hanya cocok dengan satu record pada table induk, mengapa record pada table anak tersebut tidak dijadikan satu pada table induk dengan menambahkan field-field table anak pada table induk?.

Contoh. Daftar pelanggan sebuah toko online dengan alamat pengiriman berbeda dengan alamat penagihan tetapi ada beberapa kelemahan yaitu:

- a. Pelanggan yang membedakan alamat pengiriman dengan alamat penagihan tidak banyak.
- b. Kalau alamat pengiriman dengan alamat penagihan dijadikan dalam satu tabel, maka field penagihan akan banyak yang kosong karena hanya terisi sedikit saja.

- c. *Sebaiknya dibuat tabel sendirisendiri yaitu tabel alamat pengiriman dan table alamat penagihan untuk menyimpan informasi dan kemudian masingmasing tabel tersebut direlasikan.*
- d. *Satu record alamat pengiriman hanya akan terhubung dengan satu record nama pelanggan.*

2. *One – To – Many (Satu Ke Banyak)*

Relasi yang menghubungkan satu record pada satu tabel dengan beberapa record pada tabel lainnya. Contoh. Database Perpustakaan dengan beberapa kriteria yaitu:

- a. *Nama satu anggota perpustakaan hanya akan tercatat satu kali dalam table anggota.*
- b. *Akan tercatat berkalkali dalam tabel pinjam.*
- c. *Satu orang bisa meminjam buku sampai berkalkali.*

3. *Many – To – One (Banyak Ke Satu)*

Relasi kebalikan dari One To Many (satu ke banyak) atau biasa disebut dengan look up table relationship. Relasi ini tidak saling terhubung ke kunci primer pada kedua tabel. Contoh. Database Nama Propinsi dengan kriteria yaitu:

- a. *Tabel informasi propinsi di Indonesia dihubungkan ke tabel alamat pelanggan.*
- b. *Record kota yang ada pada alamat pelanggan berelasi dengan record propinsi.*
- c. *Banyak kota yang bisa masuk ke dalam sebuah propinsi.*

4. *Many – To – Many (Banyak Ke Banyak)*

- a. *Relasi many to many ini hampir tidak mungkin ada.*
- b. *Jika ada biasanya terjadi karena kesalahan dalam perancangan tabel.*

C. *Keakuratan Relasi (Referential Integrity)*

Dalam sebuah database, data yang sudah tersimpan biasanya akan mengalami perbaikan – perbaikan dalam jangka waktu tertentu, hal ini berpengaruh pada relasi sehingga setiap jangka waktu tertentu perlu adanya

perbaikan supaya relasi yang ada bisa akurat, keakuratan relasi ini dibagi menjadi dua yaitu:

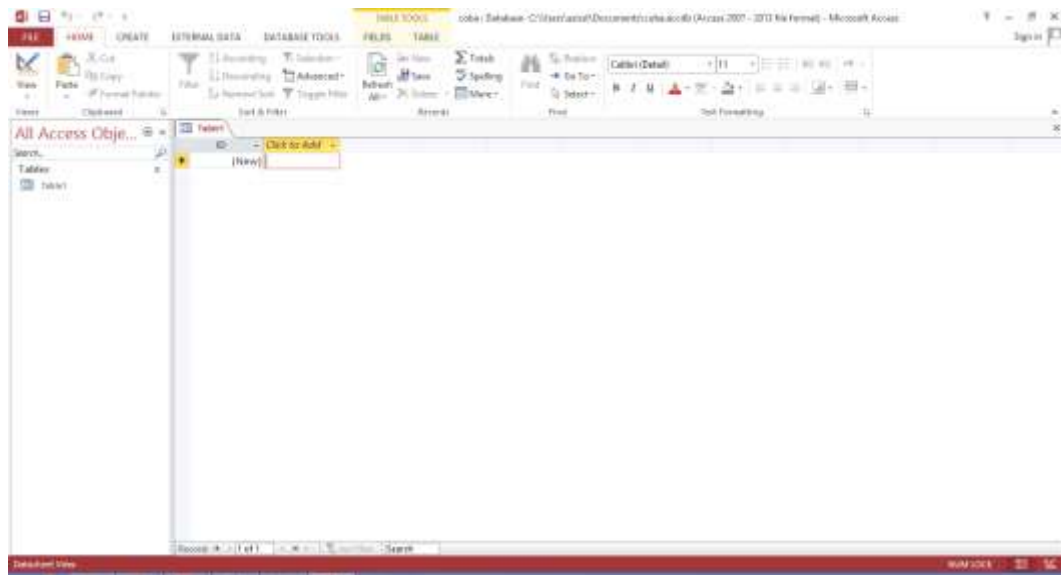
a. Cascade Delete Related Record : yaitu Penghapusan semua record.

Contoh: Jika ada mahasiswa yang mengundurkan diri maka data mahasiswa tersebut dihapus semua recordnya bukan mengganti relasinya.

b. Cascade Update Related Record : yaitu Perbaharuan record.

MENU PADA MICROSOFT OFFICE ACCESS 2013

Berikut adalah tampilan awal dari halaman baru microsoft office access 2013. Didalamnya terdapat beberapa bagian yang mendasar. Ada title bar, menu bar, toolbar, sidebar dan field.



Pada title bar berisi tentang judul atau nama dari file. Pada menu bar terdapat beberapa menu yang dapat digunakan oleh user untuk memanajemen basis data.



Menu "FILE" yang mana didalamnya terdapat beberapa pilihan menu atau perintah info untuk merepair dan memproteksi password, new untuk membuat file yang baru, open untuk membuka file yang telah tersimpan dalam suatu direktori harddisk, save untuk melakukan penyimpanan, save as untuk melakukan penyimpanan dengan nama berbeda. Print untuk melakukan pengeprinan. Close untuk melakukan penutupan halaman yang sedang terbuka. Account untuk melihat keterangan akun yang sedang menggunakan dan juga keterangan

produk. Dan option berisi tentang pengaturan yang opsional yang ada dalam ms office access 2013.

Menu “HOME” yang berisi tool-tool yang dapat digunakan oleh user untuk mengedit format dari isi / data file basis data. Seperti dibawah ini.



Menu “CREATE” yang berisi tool-tool yang dapat digunakan oleh user untuk melakukan penambahan tabel, query, form baru dll. Seperti dibawah ini.



Menu “EXTERNAL DATA” yang berisi tool-tol yang dapat digunakan oleh user untuk melakukan penambahan data dari file lain.



Menu “DATABASE TOOLS” yang berisi tool-tool yang dapat digunakan oleh user untuk melakukan pembuatan atau menghubungkan database-database atau tabel-tabel yang telah dibuat.

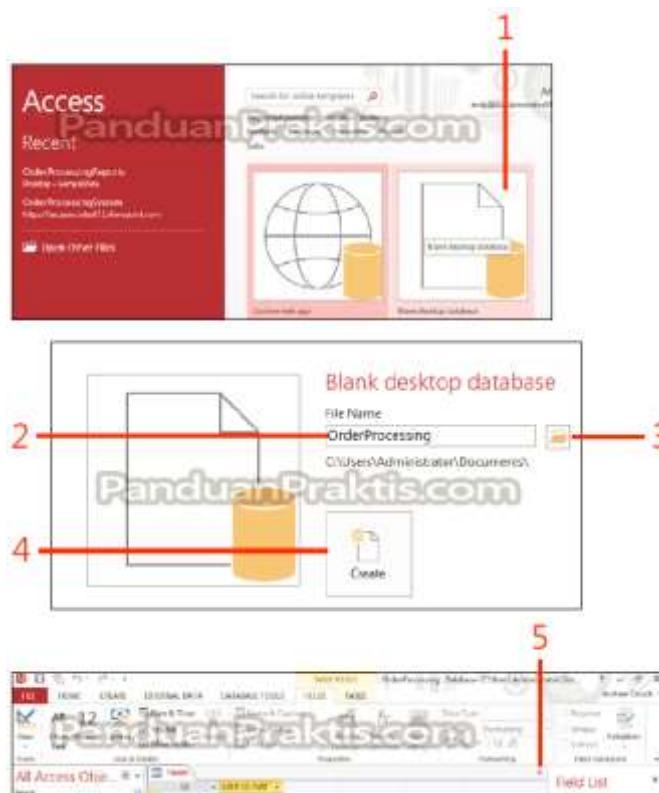


CARA MEMBUAT DATABASE DESKTOP KOSONG

Access menawarkan Anda fleksibilitas dalam membuat database desktop. Anda bisa menggunakan template database atau memulainya dengan database kosong yang kemudian Anda bisa mengimpor data yang sudah ada, atau benar-benar memulainya dengan database baru yang kemudian Anda sesuaikan dengan kebutuhan yang lebih spesifik.

Cara membuat database desktop kosong:

1. Klik Blank Desktop Database.
2. Masukkan nama database Anda.
3. Pilih lokasi dimana file database tersebut akan disimpan.
4. Klik Create.
5. Ketika database kosong baru terbuka, Anda akan melihat bahwa Access telah membuat sebuah tabel, yaitu Table1. Saat ini, Anda bisa menutup tabel ini tanpa menyimpan perubahan pada tabel. Klik tombol Close untuk menutup tabel.



Daftar tipe data yang tersedia ketika Anda menambahkan field ke dalam tabel pada database desktop:

- *Short Text: Teks hingga 255 karakter. Pada Access versi sebelumnya, tipe data ini disebut dengan Text.*
- *Long Text: Sejumlah besar teks, hingga 65,536 karakter. Pada Access versi sebelumnya, tipe data ini disebut dengan Memo.*
- *Number: properti Field Size yang disediakan untuk Byte, Integer, Long Integer (default), Single, Double, Replication ID, Decimal.*
- *Date/Time: field untuk tanggal dan waktu.*
- *Currency: field untuk mata uang.*
- *AutoNumber: Sequential Automatic Number, Long Integer (default).*
- *Yes/No: True/False (defaultnya adalah False).*
- *OLE Object: bisa menyimpan gambar atau dokumen lain.*
- *Hyperlink: link/tautan ke internet dan dokumen lokal.*
- *Attachment: memungkinkan untuk menyimpan banyak dokumen.*
- *Calculated: field Calculated.*
- *Lookup: membuat foreign key lookup.*

Microsoft Access Data Types

Data type	Description	Storage
Text	Use for text or combinations of text and numbers. 255 characters maximum	
Memo	Memo is used for larger amounts of text. Stores up to 65,536 characters. Note: You cannot sort a memo field. However, they are searchable	
Byte	Allows whole numbers from 0 to 255	1 byte
Integer	Allows whole numbers between -32,768 and 32,767	2 bytes
Long	Allows whole numbers between -2,147,483,648 and 2,147,483,647	4 bytes
Single	Single precision floating-point. Will handle most decimals	4 bytes
Double	Double precision floating-point. Will handle most decimals	8 bytes
Currency	Use for currency. Holds up to 15 digits of whole dollars, plus 4 decimal places. Tip: You can choose which country's currency to use	8 bytes
AutoNumber	AutoNumber fields automatically give each record its own number, usually starting at 1	4 bytes
Date/Time	Use for dates and times	8 bytes
Yes/No	A logical field can be displayed as Yes/No, True/False, or On/Off. In code, use the constants True and False (equivalent to -1 and 0). Note: Null values are not allowed in Yes/No fields	1 bit
Ole Object OBjects)	Can store pictures, audio, video, or other BLOBs (Binary Large up to 1GB	
Hyperlink	Contain links to other files, including web pages	
Lookup Wizard down list	Let you type a list of options, which can then be chosen from a drop-4 bytes	

MS Access Functions

MS Access has many built-in functions.

This reference contains the string, numeric, and date functions in MS Access.

MS Access String Functions

<i>Function</i>	<i>Description</i>
<i>Asc</i>	<i>Returns the ASCII value for the specific character</i>
<i>Chr</i>	<i>Returns the character for the specified ASCII number code</i>
<i>Concat with &</i>	<i>Adds two or more strings together</i>
<i>CurDir</i>	<i>Returns the full path for a specified drive</i>
<i>Format</i>	<i>Formats a value with the specified format</i>
<i>InStr</i>	<i>Gets the position of the first occurrence of a string in another</i>
<i>InstrRev</i>	<i>Gets the position of the first occurrence of a string in another, from the end of string</i>
<i>LCase</i>	<i>Converts a string to lower-case</i>
<i>Left</i>	<i>Extracts a number of characters from a string (starting from left)</i>
<i>Len</i>	<i>Returns the length of a string</i>
<i>LTrim</i>	<i>Removes leading spaces from a string</i>
<i>Mid</i>	<i>Extracts some characters from a string (starting at any position)</i>
<i>Replace</i>	<i>Replaces a substring within a string, with another substring, a specified number of times</i>
<i>Right</i>	<i>Extracts a number of characters from a string (starting from right)</i>
<i>RTrim</i>	<i>Removes trailing spaces from a string</i>
<i>Space</i>	<i>Returns a string of the specified number of space characters</i>
<i>Split</i>	<i>Splits a string into an array of substrings</i>
<i>Str</i>	<i>Returns a number as string</i>
<i>StrComp</i>	<i>Compares two strings</i>
<i>StrConv</i>	<i>Returns a converted string</i>
<i>StrReverse</i>	<i>Reverses a string and returns the result</i>
<i>Trim</i>	<i>Removes both leading and trailing spaces from a string</i>
<i>UCase</i>	<i>Converts a string to upper-case</i>

MS Access Numeric Functions

<i>Function</i>	<i>Description</i>
<i>Abs</i>	<i>Returns the absolute value of a number</i>
<i>Atn</i>	<i>Returns the arc tangent of a number</i>
<i>Avg</i>	<i>Returns the average value of an expression</i>
<i>Cos</i>	<i>Returns the cosine of an angle</i>
<i>Count</i>	<i>Returns the number of records returned by a select query</i>
<i>Exp</i>	<i>Returns e raised to the power of a specified number</i>
<i>Fix</i>	<i>Returns the integer part of a number</i>
<i>Format</i>	<i>Formats a numeric value with the specified format</i>
<i>Int</i>	<i>Returns the integer part of a number</i>
<i>Max</i>	<i>Returns the maximum value in a set of values</i>
<i>Min</i>	<i>Returns the minimum value in a set of values</i>
<i>Randomize</i>	<i>Initializes the random number generator (used by Rnd()) with a seed</i>
<i>Rnd</i>	<i>Returns a random number</i>
<i>Round</i>	<i>Rounds a number to a specified number of decimal places</i>
<i>Sgn</i>	<i>Returns the sign of a number</i>
<i>Sqr</i>	<i>Returns the square root of a number</i>
<i>Sum</i>	<i>Calculates the sum of a set of values</i>
<i>Val</i>	<i>Reads a string and returns the numbers found in the string</i>

MS Access Date Functions

<i>Function</i>	<i>Description</i>
<i>Date</i>	<i>Returns the current system date</i>
<i>DateAdd</i>	<i>Adds a time/date interval to a date and then returns the date</i>
<i>DateDiff</i>	<i>Returns the difference between two dates</i>
<i>DatePart</i>	<i>Returns a specified part of a date (as an integer)</i>
<i>DateSerial</i>	<i>Returns a date from the specified parts (year, month, and day values)</i>
<i>DateValue</i>	<i>Returns a date based on a string</i>
<i>Day</i>	<i>Returns the day of the month for a given date</i>
<i>Format</i>	<i>Formats a date value with the specified format</i>
<i>Hour</i>	<i>Returns the hour part of a time/datetime</i>

<i>Minute</i>	<i>Returns the minute part of a time/datetime</i>
<i>Month</i>	<i>Returns the month part of a given date</i>
<i>MonthName</i>	<i>Returns the name of the month based on a number</i>
<i>Now</i>	<i>Returns the current date and time based on the computer's system date and time</i>
<i>Second</i>	<i>Returns the seconds part of a time/datetime</i>
<i>Time</i>	<i>Returns the current system time</i>
<i>TimeSerial</i>	<i>Returns a time from the specified parts (hour, minute, and second value)</i>
<i>TimeValue</i>	<i>Returns a time based on a string</i>
<i>Weekday</i>	<i>Returns the weekday number for a given date</i>
<i>WeekdayName</i>	<i>Returns the weekday name based on a number</i>
<i>Year</i>	<i>Returns the year part of a given date</i>

MS Access Some Other Functions

<i>Function</i>	<i>Description</i>
<i>CurrentUser</i>	<i>Returns the name of the current database user</i>
<i>Environ</i>	<i>Returns a string that contains the value of an operating system environment variable</i>
<i>IsDate</i>	<i>Checks whether an expression can be converted to a date</i>
<i>IsNull</i>	<i>Checks whether an expression contains Null (no data)</i>
<i>IsNumeric</i>	<i>Checks whether an expression is a valid number</i>

SQL Quick Reference from W3Schools

SQL Statement	Syntax
AND / OR	<i>SELECT column_name(s)</i> <i>FROM table_name</i> <i>WHERE condition</i> <i>AND/OR condition</i>
ALTER TABLE	<i>ALTER TABLE table_name</i> <i>ADD column_name datatype</i> <i>or</i> <i>ALTER TABLE table_name</i> <i>DROP COLUMN column_name</i>
AS (alias)	<i>SELECT column_name AS column_alias</i> <i>FROM table_name</i> <i>or</i> <i>SELECT column_name</i> <i>FROM table_name AS table_alias</i>
BETWEEN	<i>SELECT column_name(s)</i> <i>FROM table_name</i> <i>WHERE column_name</i> <i>BETWEEN value1 AND value2</i>
CREATE DATABASE	<i>CREATE DATABASE database_name</i>
CREATE TABLE	<i>CREATE TABLE table_name</i> <i>(</i> <i>column_name1 data_type,</i> <i>column_name2 data_type,</i> <i>column_name3 data_type,</i> <i>...</i> <i>)</i>
CREATE INDEX	<i>CREATE INDEX index_name</i> <i>ON table_name (column_name)</i>

or

*CREATE UNIQUE INDEX index_name
ON table_name (column_name)*

*CREATE VIEW CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition*

*DELETE DELETE FROM table_name
WHERE some_column=some_value*

or

*DELETE FROM table_name
(Note: Deletes the entire table!!)*

*DELETE * FROM table_name
(Note: Deletes the entire table!!)*

DROP DATABASE DROP DATABASE database_name

*DROP INDEX DROP INDEX table_name.index_name (SQL Server)
DROP INDEX index_name ON table_name (MS Access)
DROP INDEX index_name (DB2/Oracle)
ALTER TABLE table_name
DROP INDEX index_name (MySQL)*

*DROP TABLE DROP TABLE table_name
IF EXISTS (SELECT * FROM table_name WHERE id = ?)
BEGIN
--do what needs to be done if exists
END
ELSE
BEGIN
--do what needs to be done if not
END*

*GROUP BY SELECT column_name, aggregate_function(column_name)
FROM table_name*

WHERE column_name operator value
GROUP BY column_name

HAVING *SELECT column_name, aggregate_function(column_name)*
FROM table_name
WHERE column_name operator value
GROUP BY column_name
HAVING aggregate_function(column_name) operator value

IN *SELECT column_name(s)*
FROM table_name
WHERE column_name
IN (value1,value2,..)
INSERT INTO INSERT INTO table_name
VALUES (value1, value2, value3,...)
or

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

INNER JOIN *SELECT column_name(s)*
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name=table_name2.column_name

LEFT JOIN *SELECT column_name(s)*
FROM table_name1
LEFT JOIN table_name2
ON table_name1.column_name=table_name2.column_name

RIGHT JOIN *SELECT column_name(s)*
FROM table_name1
RIGHT JOIN table_name2
ON table_name1.column_name=table_name2.column_name

FULL JOIN *SELECT column_name(s)*
FROM table_name1
FULL JOIN table_name2

	<i>ON table_name1.column_name=table_name2.column_name</i>
<i>LIKE</i>	<i>SELECT column_name(s)</i> <i>FROM table_name</i> <i>WHERE column_name LIKE pattern</i>
<i>ORDER BY</i>	<i>SELECT column_name(s)</i> <i>FROM table_name</i> <i>ORDER BY column_name [ASC DESC]</i>
<i>SELECT</i>	<i>SELECT column_name(s)</i> <i>FROM table_name</i>
<i>SELECT *</i>	<i>SELECT *</i> <i>FROM table_name</i>
<i>SELECT DISTINCT</i>	<i>SELECT DISTINCT column_name(s)</i> <i>FROM table_name</i>
<i>SELECT INTO</i>	<i>SELECT *</i> <i>INTO new_table_name [IN externaldatabase]</i> <i>FROM old_table_name</i> <i>or</i> <i>SELECT column_name(s)</i> <i>INTO new_table_name [IN externaldatabase]</i> <i>FROM old_table_name</i>
<i>SELECT TOP</i>	<i>SELECT TOP number percent column_name(s)</i> <i>FROM table_name</i> <i>TRUNCATE TABLE</i>
<i>UNION</i>	<i>TRUNCATE TABLE table_name</i> <i>SELECT column_name(s) FROM table_name1</i> <i>UNION</i> <i>SELECT column_name(s) FROM table_name2</i>
<i>UNION ALL</i>	<i>SELECT column_name(s) FROM table_name1</i> <i>UNION ALL</i> <i>SELECT column_name(s) FROM table_name2</i>
<i>UPDATE</i>	<i>UPDATE table_name</i> <i>SET column1=value, column2=value,...</i> <i>WHERE some_column=some_value</i>

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

Source : https://www.w3schools.com/sql/sql_quickref.asp