

# Leksioni 13

Endri Raco

04 July, 2024





# Section 1



# Fillojmë me një tabelë

```
-- Krijë tabelën Klientë
CREATE TABLE Klientë (
  idKlienti int PRIMARY KEY IDENTITY(1,1),
  emri varchar(50),
  mbiemri varchar(50),
  email varchar(100) UNIQUE,
  telefoni varchar(20),
  adresa varchar(255),
  qyteti varchar(50),
  shteti varchar(50),
  kodi_postar varchar(10),
  data_regjistrimit DATE DEFAULT GETDATE()
);
```



# Fillojmë me një tabelë

```
-- Mbush tabelën Klientë me të dhëna shembull
```

```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
```

```
VALUES ('Ana', 'Hoxha', 'ana.hoxha@example.com', '0681234567', 'Rruga e Dibrës 12', 'Tiranë', 'Shqipëri', '1001')
```

```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
```

```
VALUES ('Maria', 'Anders', 'maria.anders@alfredsutterkiste.com', '0123456789', 'Obere Str. 57', 'Berlin', 'Germani', '10115')
```

```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
```

```
VALUES ('Ana', 'Trujillo', 'ana.trujillo@emparedadosyhelados.com', '9876543210', 'Avda. de la Constitución 2222', 'Medellín', 'Kolumbia', '0500110')
```

```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
```

```
VALUES ('Antonio', 'Moreno', 'antonio.moreno@taqueria.com', '1234567890', 'Mataderos 2312', 'México D.F.', 'Meksikë', '06702')
```

```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
```

```
VALUES ('Thomas', 'Hardy', 'thomas.hardy@aroundthehorn.com', '2345678901', '120 Hanover Sq.', 'London', 'UK', 'E1 1AN')
```

```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
```

```
VALUES ('Christina', 'Berglund', 'christina.berglund@berglundssnabbkop.com', '3456789012', 'Berguvsvägen 8', 'Luleå', 'Suedi', '971 82')
```

```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
```

```
VALUES ('Kardinali', 'Tom B. Erichsen', 'tom.kardinal@gmail.com', '454664446', 'Skagen 21', 'Stavanger', 'Norvegji', '4002')
```



# Deklarata INSERT INTO

Deklarata INSERT INTO përdoret për të futur të dhëna të reja në një tabelë.



# Sintaksa INSERT INTO

- Është e mundur të shkruhet deklarata **INSERT INTO** në dy mënyra:
  - ① Specifikoni emrat e kolonave dhe vlerat që do të futen:

```
INSERT NË emrin e tabelës (kolona 1, kolona 2, kolona 3,  
VALUES (vlera1, vlera2, vlera3, ...));
```



- 2 Nëse po shtoni vlera për të gjitha kolonat e tabelës, nuk keni nevojë të specifikoni emrat e kolonave në query-n SQL.
- Megjithatë, sigurohuni që rendi i vlerave të jetë në të njëjtin rend si kolonat në tabelë.





# Sintaksa INSERT INTO

- Këtu, sintaksa **INSERT INTO** do të ishte si më poshtë:

```
INSERT INTO table_name  
VALUES (vlera1, vlera2, vlera3, ...);
```



Deklarata e mëposhtme SQL fut një rekord të ri në tabelën “Klientë”:



```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, c  
VALUES ('Kardinali', 'Tom B. Erichsen', 'tom.kardinal@gmail.co
```



Zgjedhja nga tabela “Klientë” tani do të duket kështu:

```
SELECT * FROM Klientë;
```



42|3ar3gu

NEW

SQLSERVER

RUN

queries.sql

```

1 -- Krijë tabelën Klientë
2 CREATE TABLE Klientë (
3     idKlienti int PRIMARY KEY IDENTITY(1,1),
4     emri varchar(50) NOT NULL,
5     mbiemri varchar(50) NOT NULL,
6     email varchar(100) NOT NULL UNIQUE,
7     telefoni varchar(20),
8     adresa varchar(255),
9     qyteti varchar(50),
10    shteti varchar(50),
11    kodi_postar varchar(10),
12    data_regjistrimit DATE DEFAULT GETDATE()
13 );
14
15 -- Mbush tabelën Klientë me të dhëna shembull
16
17 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
18 VALUES ('Ana', 'Hoxha', 'ana.hoxha@example.com', '0681234567', 'Rruga e Dibrës 12', 'Tiranë', 'Shqipëri', '1001');
19
20 -- Mbush tabelën Klientë me të dhëna shembull
21
22 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
23 VALUES ('Maria', 'Anders', 'maria.anders@alfredsfutterkiste.com', '0123456789', 'Obere Str. 57', 'Berlin', 'Germany', '12200');
24
25 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
26 VALUES ('Ana', 'Trujillo', 'ana.trujillo@comparadosyelados.com', '9876543210', 'Avenida de la Constitución 2222', 'México D.F.', 'Mexico', '0');
27
28 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
29 VALUES ('Antonio', 'Moreno', 'antonio.moreno@taqueria.com', '1234567890', 'Mataderos 2312', 'México D.F.', 'Mexico', '05023');
30
31 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
32 VALUES ('Thomas', 'Hardy', 'thomas.hardy@aroundthetown.com', '2345678901', '128 Hanover Sq.', 'London', 'UK', 'W1A 1DP');
33
34 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
35 VALUES ('Christina', 'Berglund', 'christina.berglund@berglundsnaabkop.com', '3456789012', 'Berguvsvägen 8', 'Luleå', 'Sweden', 'S-958 22');
36
37 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
38 VALUES ('Kardinali', 'Tom B. Erichsen', 'tom.kardinal@gmail.com', '4567890123', 'Stavanger', 'Norvegji', '4000');
39
40 SELECT * FROM Klientë

```

STOP

Input for the program (Optional)

Output:

idKlienti	emri	mbiemri
1	Ana	Hoxha
2	Maria	Anders
3	Ana	Trujillo
4	Antonio	Moreno
5	Thomas	Hardy
6	Christina	Berglund
7	Kardinali	Tom B. Erichsen

- A e keni vënë re që nuk kemi futur ndonjë numër në fushën e ID-së së klientit?
- Kolona **idKlienti** është një fushë me rritje automatike dhe do të gjenerohet automatikisht kur një rekord i ri futet në tabelë.



# Futja e të dhënave vetëm në kolonat e specifikuara

- Është gjithashtu e mundur që të futen të dhëna vetëm në kolona specifike.



# Futja e të dhënave vetëm në kolonat e specifikuara

- Deklarata e mëposhtme SQL do të fusë një rekord të ri, por fut vetëm të dhëna në kolonat “emri”, “qyteti” dhe “shteti”

(ID-ja e klientit do të përditësohet automatikisht):





```
-- Mbush tabelën Klientë me të dhëna shitesë, duke siguruar që email-i të jetë unik ose i bosh  
INSERT INTO Klientë (emri, qyteti, shteti)  
VALUES ('John', 'Wayne', 'Albania');
```



- Zgjedhja nga tabela “Klientë” tani do të duket kështu:

Output:

idKlienti	emri	mbiemri
1	Ana	Hoxha
2	Maria	Anders
3	Ana	Trujillo
4	Antonio	Moreno
5	Thomas	Hardy
6	Christina	Berglund
7	Kardinali	Tom B. Erichsen

  

idKlienti	emri	mbiemri
1	Ana	Hoxha
2	Maria	Anders
3	Ana	Trujillo
4	Antonio	Moreno
5	Thomas	Hardy
6	Christina	Berglund
7	Kardinali	Tom B. Erichsen
8	John	NULL



- Është gjithashtu e mundur që të futen shumë rreshta në një deklaratë.
- Për të futur shumë rreshta të dhënash, ne përdorim të njëjtën deklaratë **INSERT INTO**, por me vlera të shumta:



```
INSERT INTO Klientë (emri, mbiemri, adresa, qyteti, kodi_postar, shteti, email)
VALUES
('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway', 'cardinal.erichsen@norway.com'),
('Greasy Burger', 'Per Olsen', 'Gateveien 15', 'Sandnes', '4306', 'Norway', 'greasy.burger@norway.com'),
('Tasty Tee', 'Finn Egan', 'Streetroad 19B', 'Liverpool', 'L1 0AA', 'UK', 'tasty.tee@uk.com');

-- Kontrolllo të dhënat e futura përsëri
SELECT * FROM Klientë;
```



- Zgjedhja nga tabela “Klientë” tani do të duket kështu:



Output:

idKlienti	emri	mbiemri
1	Ana	Hoxha
2	Maria	Anders
3	Ana	Trujillo
4	Antonio	Moreno
5	Thomas	Hardy
6	Christina	Berglund
7	Kardinali	Tom B. Erichsen
8	Cardinal	Tom B. Erichsen
9	Greasy Burger	Per Olsen
10	Tasty Tee	Finn Egan



# Çfarë është një Vlerë NULL?

Një fushë me një vlerë NULL është një fushë pa vlerë.





# Çfarë është një Vlerë NULL?

- Nëse një fushë në një tabelë është opsionale, është e mundur të futni një rekord të ri ose të përditësoni një rekord pa shtuar një vlerë në këtë fushë.
- Atëherë, fusha do të ruhet me një vlerë NULL.



# Çfarë është një Vlerë NULL?

- Një vlerë NULL është e ndryshme nga një vlerë zero ose një fushë që përmban hapësira.
- Një fushë me një vlerë NULL është një fushë që është lënë bosh gjatë krijimit të rekordit!



# Si të Testoni për Vlera NULL?

- Nuk është e mundur të testoni për vlera NULL me operatorët e krahasimit, të tillë si `=`, `<`, ose `<>`.
- Ne do të duhet të përdorim operatorët `IS NULL` dhe `IS NOT NULL`.



```
SELECT emrat_e_kolonave  
FROM emri_i_tabelës  
WHERE emri_i_kolonës IS NULL;
```

# Sintaksa IS NOT NULL

```
SELECT emrat_e_kolonave  
FROM emri_i_tabelës  
WHERE emri_i_kolonës IS NOT NULL;
```



- Operatori IS NULL përdoret për të testuar për vlera bosh (vlera NULL).
- SQL-ja e mëposhtme rendit të gjithë klientët me një vlerë NULL në fushën “adresa”:



# Tabela

```
-- Krijo tabelën Klientë
CREATE TABLE Klientë (
    idKlienti int PRIMARY KEY IDENTITY(1,1),
    emri varchar(50),
    mbiemri varchar(50),
    email varchar(100) UNIQUE,
    telefoni varchar(20),
    adresa varchar(255),
    qyteti varchar(50),
    shteti varchar(50),
    kodi_postar varchar(10),
    data_regjistrimit DATE DEFAULT GETDATE()
);
```



```
-- Mbush tabelën Klientë me të dhëna shembull, përfshirë disa me fusha adresa bosh
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
VALUES
('Ana', 'Hoxha', 'ana.hoxha@example.com', '0681234567', 'Rruga e Dibrës 12', 'Tiranë', 'Shqipëri', '1001'),
('Maria', 'Anders', 'maria.anders@alfredsutterkiste.com', '0123456789', NULL, 'Berlin', 'Gjermani', '12209'),
('Ana', 'Trujillo', 'ana.trujillo@emparedadosyelados.com', '9876543210', 'Avda. de la Constitución 2222', 'México', 'Meksikë'),
('Antonio', 'Moreno', 'antonio.moreno@taqueria.com', '1234567890', 'Mataderos 2312', 'México D.F.', 'Meksikë'),
('Thomas', 'Hardy', 'thomas.hardy@aroundthehorn.com', '2345678901', NULL, 'London', 'MB', 'WA1 1DP'),
('Christina', 'Berglund', 'christina.berglund@berglundssnabbkop.com', '3456789012', 'Berguvsvägen 8', 'Luleå', 'Sueda'),
('Kardinali', 'Tom B. Erichsen', 'tom.kardinal@gmail.com', '454664446', 'Skagen 21', 'Stavanger', 'Norvegji', '4005'),
('John', 'Wayne', NULL, NULL, NULL, 'Wayne', 'Albania', NULL);
```





# Testimi për Vlera NULL

```
SELECT emri, mbiemri, adresa  
FROM Klientë  
WHERE adresa IS NULL;
```



STDIN

Input for the program ( Optional )

Output:

emri	mbiemri
-----	
Maria	Anders
Thomas	Hardy
John	Wayne



# Testimi për Vlera NOT NULL

```
SELECT emri, mbiemri, adresa  
FROM Klientë  
WHERE adresa IS NOT NULL;
```



STDIN

Input for the program ( Optional )

Output:

emri	mbiemri
-----	
Ana	Hoxha
Ana	Trujillo
Antonio	Moreno
Christina	Berglund
Kardinali	Tom B. Erichsen

# Deklarata SQL UPDATE

Deklarata UPDATE përdoret për të modifikuar rekordet ekzistuese në një tabelë.



# Sintaksa UPDATE

```
UPDATE emri_i_tabelës  
SET kolona1 = vlera1, kolona2 = vlera2, ...  
WHERE kushti;
```



- Bëni kujdes kur përditësoni rekordet në një tabelë!
- Vini re klauzolën WHERE në deklaratën UPDATE. Klauzola WHERE specifikon cilat rekorde do të përditësohen.
- Nëse e lini jashtë klauzolën WHERE, të gjitha rekordet në tabelë do të përditësohen!



# Përditësimi i Një Rekordi

Deklarata SQL më poshtë përditëson klientin e parë ( $\text{idKlienti} = 1$ ) me një kontakt të ri dhe një qytet të ri.

```
UPDATE Klientë  
SET mbiemri = 'Schmidt', qyteti = 'Frankfurt'  
WHERE idKlienti = 1;
```





- Zgjedhja nga tabela “Klientë” tani do të duket kështu:

```

1 -- Krijë tabelën Klientë
2 CREATE TABLE Klientë (
3     idKlienti int PRIMARY KEY IDENTITY(1,1),
4     emri varchar(50),
5     mbiemri varchar(50),
6     email varchar(100) UNIQUE,
7     telefoni varchar(20),
8     adresa varchar(255),
9     qyteti varchar(50),
10    shteti varchar(50),
11    kodi_postar varchar(10),
12    data_regjistrimit DATE DEFAULT GETDATE()
13 );
14
15 -- Mbush tabelën Klientë me të dhëna shembull, përfshirë disa me fusha adresa bosh
16 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
17 VALUES
18 ('Ana', 'Hoxha', 'ana.hoxha@example.com', '0681234567', 'Rruga e Dibrës 12', 'Tiranë', 'Shqipëri', '1001'),
19 ('Maria', 'Anders', 'maria.anders@alfredsfutterkiste.com', '0123456789', NULL, 'Berlin', 'Gjermani', '12209'),
20 ('Ana', 'Trujillo', 'ana.trujillo@esparedadosyelados.com', '0876543210', 'Avda. de la Constitución 2222', 'México D.F.', 'Meksikë', '05821'),
21 ('Antonio', 'Moreno', 'antonio.moreno@taqueria.com', '1234567890', 'Mataderos 2312', 'México D.F.', 'Meksikë', '05823'),
22 ('Thomas', 'Hardy', 'thomas.hardy@aroundthehorn.com', '1234567890', NULL, 'London', 'ng', 'Vall 100'),
23 ('Christina', 'Berglund', 'christina.berglund@berglunds snabbkop.com', '3456789012', 'Berguvsvägen 8', 'Luleå', 'Suedi', 'S-958 22'),
24 ('Kardinali', 'Tom B. Erichsen', 'tom.kardinal@gmail.com', '414664446', 'Skagen 21', 'Stavanger', 'Norvegji', '4006'),
25 ('John', 'Wayne', NULL, NULL, 'Wayne', 'Albania', NULL);
26
27
28 UPDATE Klientë
29 SET mbiemri = 'Scheidt', qyteti = 'Frankfurt'
30 WHERE idKlienti = 1;
31
32
33 SELECT * FROM Klientë;

```

STDIN

Input for the program (Optional)

Output:

idKlienti	emri	mbiemri
1	Ana	Scheidt
2	Maria	Anders
3	Ana	Trujillo
4	Antonio	Moreno
5	Thomas	Hardy
6	Christina	Berglund
7	Kardinali	Tom B. Erichsen
8	John	Wayne

- Është klauzola **WHERE** që përcakton se sa rekorde do të përditësohen.
- Deklarata SQL më poshtë do të përditësojë mbiemrin në “Juan” për të gjitha rekordet ku shteti është “Meksikë”:



# Përditësimi i Rekordeve të Shumta

```
UPDATE Klientë  
SET mbiemri = 'Juan'  
WHERE shteti = 'Meksikë';
```



- Zgjedhja nga tabela “Klientë” tani do të duket kështu:



Output:

idKlienti	emri	mbiemri
1	Ana	Hoxha
2	Maria	Anders
3	Ana	Juan
4	Antonio	Juan
5	Thomas	Hardy
6	Christina	Berglund
7	Kardinali	Tom B. Erichsen
8	John	Wayne

# Paralajmërim për Përditësimin

- Bëni kujdes kur përditësoni rekordet.
- Nëse e lini jashtë klauzolën WHERE, të gjitha rekordet do të përditësohen!



# Pa Klauzolën WHERE

```
UPDATE Klientë  
SET mbiemri = 'Juan';
```





- Zgjedhja nga tabela “Klientë” tani do të duket kështu:



```
-- Krijë tabelën Klientë
```

```
CREATE TABLE Klientë (
  idKlienti int PRIMARY KEY IDENTITY(1,1),
  emri varchar(50),
  mbiemri varchar(50),
  email varchar(100) UNIQUE,
  telefoni varchar(20),
  adresa varchar(255),
  qyteti varchar(50),
  shteti varchar(50),
  kodi_postar varchar(10),
  data_regjistrimit DATE DEFAULT GETDATE()
);
```

```
-- Përbash të tabelën Klientë me të dhëna shembull, përfshirë disa me fusha adresa bosh
```

```
INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
```

```
VALUES
('Ana', 'Hoxha', 'ana.hoxha@example.com', '0681234567', 'Rruga e Dibrës 12', 'Tiranë', 'Shqipëri', '1001'),
('Maria', 'Anders', 'maria.anders@galfredsfutterkiste.com', '0123456789', NULL, 'Berlin', 'Gjermani', '12200'),
('Ana', 'Trujillo', 'ana.trujillo@esperadosyhelados.com', '9876543210', 'Avda. de la Constitución 222', 'México D.F.', 'Meksikë', '05021'),
('Antonio', 'Moreno', 'antonio.moreno@taqueria.com', '1234567890', 'Paraderos 2312', 'México D.F.', 'Meksikë', '05023'),
('Thomas', 'Hardy', 'thomas.hardy@aroundthehorn.com', '2345678901', NULL, 'London', 'Mb', 'M1 1DP'),
('Christina', 'Berglund', 'christina.berglund@berglunds snabbkop.com', '3456789012', 'Berguvsvägen 8', 'Luleå', 'Suedi', 'S-958 22'),
('Kardinali', 'Tom B. Erichsen', 'tom.kardinali@gmail.com', '456664444', 'Skagen 21', 'Stavanger', 'Norvegji', '4006'),
('John', 'Wayne', NULL, NULL, NULL, 'Wayne', 'Albania', NULL);
```

```
UPDATE Klientë
```

```
SET mbiemri = 'Juan';
```

```
SELECT * FROM Klientë;
```

STDN

Input for the program ( Optional )

Output:

idKlienti	emri	mbiemri
1	Ana	Juan
2	Maria	Juan
3	Ana	Juan
4	Antonio	Juan
5	Thomas	Juan
6	Christina	Juan
7	Kardinali	Juan
8	John	Juan

# Deklarata SQL DELETE

Deklarata DELETE përdoret për të fshirë rekordet ekzistuese në një tabelë.



# Sintaksa DELETE

```
DELETE FROM emri_i_tabelës WHERE kushti;
```



- Bëni kujdes kur fshini rekorde në një tabelë!
- Vini re klauzolën WHERE në deklaratën DELETE.
- Klauzola WHERE specifikon cilat rekorde do të fshihen.
- Nëse e lini jashtë klauzolën WHERE, të gjitha rekordet në tabelë do të fshihen!



- Deklarata SQL më poshtë fshin klientin “Alfreds Futterkiste” nga tabela Klientë:



# Shembulli DELETE

```
DELETE FROM Klientë WHERE emri='John' AND mbiemri='Wayne';
```



- Zgjedhja nga tabela “Klientë” tani do të duket kështu:





# Shembulli DELETE

```
1 -- Krijë tabelën Klientë
2 CREATE TABLE Klientë (
3     idKlienti int PRIMARY KEY IDENTITY(1,1),
4     emri varchar(50),
5     mbiemri varchar(50),
6     email varchar(100) UNIQUE,
7     telefoni varchar(20),
8     adresa varchar(255),
9     qyteti varchar(50),
10    shteti varchar(50),
11    kodi_postar varchar(50),
12    data_regjistrimit DATE DEFAULT GETDATE()
13 );
14
15 -- Mbush tabelën Klientë me të dhëna shembull, përfshirë disa me fusha adresa bosh
16 INSERT INTO Klientë (emri, mbiemri, email, telefoni, adresa, qyteti, shteti, kodi_postar)
17 VALUES
18 ('Ana', 'Hoxha', 'ana.hoxha@example.com', '0601234567', 'Rruga e Dibrës 12', 'Tiranë', 'Shqipëri', '1001'),
19 ('Maria', 'Anders', 'maria.anders@alfredfutterkiste.com', '0123456789', NULL, 'Berlin', 'Gjermani', '12289'),
20 ('Ana', 'Trujillo', 'ana.trujillo@emparedadosyelados.com', '9876543210', 'Avda. de la Constitución 2222', 'México D.F.', 'Meksikë', '05021'),
21 ('Antonio', 'Moreno', 'antonio.moreno@caqueria.com', '1234567890', 'Mataderos 2312', 'México D.F.', 'Meksikë', '05023'),
22 ('Thomas', 'Hardy', 'thomas.hardy@arounthehorn.com', '2345678901', NULL, 'London', 'WB', 'Wali IDP'),
23 ('Christina', 'Berglund', 'christina.berglund@berglundssnabbkoo.com', '3456789012', 'Berguvsvägen 8', 'Luleå', 'Suedi', 'S-958 22'),
24 ('Kardinali', 'Tom B. Erichsen', 'tom.kardinal@gmail.com', '456664446', 'Skagen 21', 'Stavanger', 'Norvegji', '4006'),
25 ('John', 'Wayne', NULL, NULL, NULL, 'Wayne', 'Albania', NULL);
26
27
28 DELETE FROM Klientë WHERE emri='John' AND mbiemri='Wayne';
29
30
31 SELECT * FROM Klientë;
32
```

STDM

Input for the program ( Optional )

Output:

idKlienti	emri	mbiemri
1	Ana	Hoxha
2	Maria	Anders
3	Ana	Trujillo
4	Antonio	Moreno
5	Thomas	Hardy
6	Christina	Berglund
7	Kardinali	Tom B. Erichsen



- Është e mundur të fshini të gjitha rreshtat në një tabelë pa e fshirë tabelën.
- Kjo do të thotë që struktura e tabelës, atributet dhe indeksat do të mbeten të paprekura:



# Fshirja e të Gjitha Rekordeve

```
DELETE FROM emri_i_tabelës;
```



- Deklarata SQL më poshtë fshin të gjitha rreshtat në tabelën Klientë, pa e fshirë tabelën:



# Fshirja e të Gjitha Rekordeve

```
DELETE FROM Klientë;
```



- Zgjedhja nga tabela “Klientë” tani do të duket kështu:



# Fshirja e të Gjitha Rekordeve

STDIN

[Click to change the language](#)

Input for the program ( Optional )

Output:

```
idKlienti    emri                                     mbiemri
```



Për të fshirë plotësisht një tabelë, përdorni deklaratën DROP TABLE:

```
DROP TABLE Klientë;
```





- Klauzola **SELECT TOP** përdoret për të specifikuar numrin e rekordeve që do të kthehen.
- Klauzola **SELECT TOP** është e dobishme në tabelat e mëdha me mijëra rekorde.
- Kthimi i një numri të madh rekordesh mund të ndikojë në performancë.



# Sintaksa për SQL Server

```
SELECT TOP numri|përqindja kolona(t)  
FROM emri_i_tabelës  
WHERE kushti;
```



- Zgjedh vetëm tre rekordet e para të tabelës Klientë:



```
SELECT TOP 3 * FROM Klientë;
```



Output:

idKlienti	emri	mbiemri
1	Ana	Hoxha
2	Maria	Anders
3	Ana	Trujillo

- Zgjedh 50% të parë të rekordeve nga tabela Klientë:

```
SELECT TOP 50 PERCENT * FROM Klientë;
```



- Zgjedh tre rekordet e para nga tabela Klientë, ku shteti është “Gjermani”:



# Shtimi i Klauzolës WHERE

```
SELECT TOP 3 * FROM Klientë  
WHERE shteti='Gjermani';
```





Output:

idKlienti	emri	mbiemri
2	Maria	Anders

- Shtoni fjalën kyçe **ORDER BY** kur dëshironi të renditni rezultatin dhe të ktheni tre rekordet e para të rezultatit të renditur.



- Renditni rezultatin në mënyrë alfabetike të kundërt sipas emrit, dhe ktheni tre rekordet e para:

```
SELECT TOP 3 * FROM Klientë  
ORDER BY emri DESC;
```



Output:

idKlienti	emri	mbiemri
5	Thomas	Hardy
2	Maria	Anders
7	Kardinali	Tom B. Erichsen

- Një funksion agregues është një funksion që kryen një llogaritje mbi një grup vlerash dhe kthen një vlerë të vetme.



- Funksionet agreguese shpesh përdoren me klauzolën **GROUP BY** të deklaratës **SELECT**.
- Klauzola **GROUP BY** ndan grupin e rezultateve në grupe vlerash dhe funksioni agregues mund të përdoret për të kthyer një vlerë të vetme për secilin grup.



Funksionet agreguese më të përdorura në SQL janë:

- **MIN()** - kthen vlerën më të vogël brenda kolonës së zgjedhur
- **MAX()** - kthen vlerën më të madhe brenda kolonës së zgjedhur
- **COUNT()** - kthen numrin e rreshtave në një grup
- **SUM()** - kthen shumën totale të një kolone numerike
- **AVG()** - kthen vlerën mesatare të një kolone numerike



Funksionet agreguese injorojnë vlerat null (përveç **COUNT()**).





# Funksionet SQL MIN() dhe MAX()

- Funksioni MIN() kthen vlerën më të vogël të kolonës së zgjedhur.
- Funksioni MAX() kthen vlerën më të madhe të kolonës së zgjedhur.



# Shembull për MIN()

Gjeni çmimin më të ulët në kolonën Çmimi:

```
SELECT MIN(Çmimi)
FROM Produkte;
```



```
-- Krijó tabelën Produkte
CREATE TABLE Produkte (
    ProduktID int PRIMARY KEY IDENTITY(1,1),
    EmriProduktit varchar(255) NOT NULL,
    FurnizuesiID int NOT NULL,
    KategoriaID int NOT NULL,
    Njesia varchar(255) NOT NULL,
    Çmimi decimal(10, 2) NOT NULL
);
```

```
-- Mbush tabelën Produkte me të dhëna shembull
INSERT INTO Produkte (EmriProduktit, FurnizuesiID, KategoriaID, Njesia, Çmimi)
VALUES
('Chais', 1, 1, '10 kuti x 20 qese', 18.00),
('Chang', 1, 1, '24 - 12 oz shishe', 19.00),
('Aniseed Syrup', 1, 2, '12 - 550 ml shishe', 10.00),
('Chef Anton's Cajun Seasoning', 2, 2, '48 - 6 oz kavanoza', 22.00),
('Chef Anton's Gumbo Mix', 2, 2, '36 kuti', 21.35);
```



# Shembull për MAX()

Gjeni çmimin më të lartë në kolonën Çmimi:

```
SELECT MAX(Çmimi)  
FROM Produkte;
```



Output:

-----  
22.00

# Përcaktoni Emrin e Kolonës (Alias)

- Kur përdorni `MIN()` ose `MAX()`, kolona e kthyer nuk do të ketë një emër përshkrues.
- Për t'i dhënë kolonës një emër përshkrues, përdorni fjalën kyçe `AS`:



# Përcaktoni Emrin e Kolonës (Alias)

```
SELECT MIN(Çmimi) AS ÇmimiMeIULët  
FROM Produkte;
```





---

Output:

ÇmimiMeIulët

-----

10.00

Këtu përdorim funksionin MIN() dhe klauzolën GROUP BY, për të kthyer çmimin më të ulët për secilën kategori në tabelën Produkte:



# Përdorni MIN() me GROUP BY

```
SELECT MIN(Çmimi) AS ÇmimiMeIULët, KategoriaID  
FROM Produkte  
GROUP BY KategoriaID;
```



Output:

ÇmimiMeIULët	KategoriaID
--------------	-------------

-----	-----
-------	-------

18.00	1
-------	---

10.00	2
-------	---

# Funksioni SQL COUNT()

- Funksioni COUNT() kthen numrin e rreshtave që përputhen me një kriter të specifikuar.



Gjeni numrin total të rreshtave në tabelën **Produkte**:

```
SELECT COUNT(*)  
FROM Produkte;
```

Output:

-----  
5

```
SELECT COUNT(emri_i_kolonës)  
FROM emri_i_tabelës  
WHERE kushti;
```



- Ju mund të përcaktoni një emër kolone në vend të simbolit yll ().
- Nëse përcaktoni një emër kolone në vend të (), vlerat NULL nuk do të numërohen.



Gjeni numrin e produkteve ku EmriProduktit nuk është null:



Gjeni numrin total të rreshtave në tabelën **Produkte**:

```
SELECT COUNT(EmriProduktit)
FROM Produkte;
```

Output:

-----  
5

Ju mund të shtoni një klauzolë WHERE për të specifikuar kushtet:



Gjeni numrin e produkteve ku çmimi është më i lartë se 20:

```
SELECT COUNT(ProduktID)
FROM Produkte
WHERE Çmimi > 20;
```



Output:

-----  
2

## Injoroni

Dublikatat

- Ju mund të injoroni dublikatat duke përdorur fjalën kyçe **DISTINCT** në funksionin **COUNT()**.
- Nëse specifikohet **DISTINCT**, rreshtat me të njëjtën vlerë për

 Endri Raço

Sa çmime të ndryshme ka në tabelën Produkte:

```
SELECT COUNT(DISTINCT Çmimi)
FROM Produkte;
```





Output:

-----

5

Jepni kolonës së numëruar një emër duke përdorur fjalën kyçe AS.



Jepni kolonës emrin “Numri i Rekordeve”:

```
SELECT COUNT(*) AS [Numri i Rekordeve]  
FROM Produkte;
```

Output:

Numri i Rekordeve

-----

5

- Këtu përdorim funksionin COUNT() dhe klauzolën GROUP BY, për të kthyer numrin e rekordeve për secilën kategori në tabelën Produkte:



```
SELECT COUNT(*) AS [Numri i Rekordeve], KategoriaID  
FROM Produkte  
GROUP BY KategoriaID;
```



Output:

Numri i Rekordeve	KategoriaID
-------------------	-------------

-----	
-------	--

2	1
---	---

3	2
---	---

# Funksioni SQL SUM()

Funksioni SUM() kthen shumën totale të një kolone numerike.





# Tabela

```
-- Krijo tabelën DetajetPorosi
CREATE TABLE DetajetPorosi (
  DetajetPorosiID int PRIMARY KEY IDENTITY(1,1),
  PorosiaID int NOT NULL,
  ProduktiID int NOT NULL,
  Sasia int NOT NULL
);
```



```
-- Mbush tabelën DetajetPorosi me të dhëna shembull
INSERT INTO DetajetPorosi (PorosiaID, ProduktiID, Sasia)
VALUES
(10248, 11, 12),
(10248, 42, 10),
(10248, 72, 5),
(10249, 14, 9),
(10249, 51, 40);
```



Ktheni shumën e të gjitha fushave **Sasia** në tabelën **DetajetPorosi**:

```
SELECT SUM(Sasia)  
FROM DetajetPorosi;
```



Output:

-----

76

```
SELECT SUM(emri_i_kolonës)  
FROM emri_i_tabelës  
WHERE kushti;
```

Ju mund të shtoni një klauzolë WHERE për të specifikuar kushtet:



Ktheni shumën e fushës Sasia për produktin me ProduktiID 11:

```
SELECT SUM(Sasia)
FROM DetajetPorosi
WHERE ProduktiID = 11;
```



Output:

-----  
12



Jepni kolonës së përmbledhur një emër duke përdorur fjalën kyçe AS.



Jepni kolonës emrin “totali”:

```
SELECT SUM(Sasia) AS totali  
FROM DetajetPorosi;
```



Output:

totali

-----

76

Këtu përdorim funksionin SUM() dhe klauzolën GROUP BY, për të kthyer shumën e Sasia për secilin PorosiaID në tabelën DetajetPorosi:



```
SELECT PorosiaID, SUM(Sasia) AS [Sasia Totale]  
FROM DetajetPorosi  
GROUP BY PorosiaID;
```



Output:

PorosiaID	Sasia Totale
10248	27
10249	49

# SUM() me Një Shprehje

- Parametri brenda funksionit SUM() mund të jetë gjithashtu një shprehje.
- Nëse supozojmë se secili produkt në tabelën DetajetPorosi kushton 10 dollarë, mund të gjejmë fitimet totale në dollarë duke shumëzuar çdo sasi me 10:



Përdorni një shprehje brenda funksionit SUM():

```
SELECT SUM(Sasia * 10)
FROM DetajetPorosi;
```





---

Output:

-----  
760

Ne gjithashtu mund të bashkojmë tabelën DetajetPorosi me tabelën Produkte për të gjetur shumën aktuale, në vend që të supozojmë se është 10 dollarë:



*-- Krijo tabelën Produkte*

```
CREATE TABLE Produkte (  
    ProduktiID int PRIMARY KEY IDENTITY(1,1),  
    EmriProduktit varchar(255) NOT NULL,  
    Çmimi decimal(10, 2) NOT NULL  
);
```



```
-- Mbush tabelën Produkte me të dhëna shembull
INSERT INTO Produkte (EmriProduktit, Çmimi)
VALUES
('Produkt1', 10.00),
('Produkt2', 20.00),
('Produkt3', 15.00);
```

```
-- Krijo tabelën DetajetPorosi
CREATE TABLE DetajetPorosi (
    DetajetPorosiID int PRIMARY KEY IDENTITY(1,1),
    PorosiaID int NOT NULL,
    ProduktiID int NOT NULL,
    Sasia int NOT NULL
);
```



```
-- Mbush tabelën DetajetPorosi me të dhëna shembull
INSERT INTO DetajetPorosi (PorosiaID, ProduktiID, Sasia)
VALUES
(1, 1, 2),
(1, 2, 1),
(2, 3, 3);
```



Bashkoni tabelën DetajetPorosi me Produkte, dhe përdorni SUM() për të gjetur shumën totale:

```
SELECT SUM(Çmimi * Sasia) AS ShumaTotale  
FROM DetajetPorosi  
LEFT JOIN Produkte ON DetajetPorosi.ProduktiID = Produkte.ProduktiID;
```



Output:

ShumaTotale

-----  
85.00