



**UNIVERSITETI AAB**  
**FAKULTETI SHKENCA KOMPJUTERIKE**  
Inxhinieria Softuerike

# Querit\_1 (Përzgjedhësit)

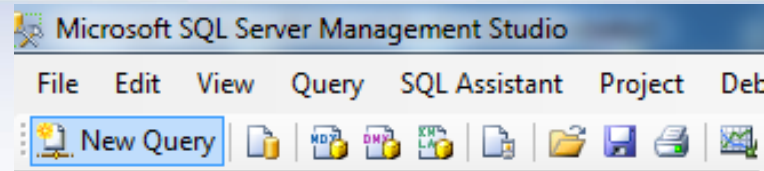
- Querit e thjesht
- Zgjedhja e disa kolonave
- Përdorimi i emrave të plotë dhe alias gjatë selektimit
- Përdorimi i kushtit WHERE për të zgjedhur disa rreshta
- Përdorimi i komandës DISTINCT etc.

# Structured Query Language - SQL

- Gjuha programuese e cila shfrytëzohet për krijimin, shfrytëzimin dhe mirëmbajtjen e database.
  - **DDL (data definition language)** – Definimi i nivelit konceptual dhe eksternal respektivisht specifikimi i shemave të të dhënave
  - **DML (data manipulation language)** – Realizimi i modifikimeve, lidhja në mes programeve aplikative dhe bazës, kryerja e operacioneve të ndryshme – lexim, regjistrim, fshirje, etj.
  - **QL (Query language)** – pjesë e gjuhës DML, shërben shfrytëzuesit të fundit për realizimin e kërkesave respektivisht “pyetsorëve” (query), pra, kërkimin, shfletimin, selektimin, etj. të të dhënave.
- DDL + DML + QL =SQL (Structured Query Language), vitet e vona 1980

# Hyrje në Queri

```
SELECT [DISTINCT] target-list  
FROM relation-list  
WHERE qualification
```



- SELECT klauzula e cila specifikon listen e attributeve dalëse
- DISTINCT opcion alternativ i cili përjashton paraqitjen e n-sheve (rreshtave) duplikate
- FROM klauzula e cila specifikon listen e tabelave respektivisht relacioneve nga do të merren të dhënat
- WHERE klauzulë opcionale e cila specifikon kushtin e caktuar për përzgjedhjen

# Operatorët për klauzulën WHERE

Operator	Description	Operator	Description
=	Equal	LIKE	Search for a pattern
<>	Not equal	AS	Alias
>	Greater than	IS NULL	
<	Less than	NOT	Negation
>=	Greater than or equal	HAVING	
<=	Less than or equal	JOIN	
DISTINCT / TOP		UNION	
AND / OR		Group BY	
BETWEEN	Between a range		
IN			

# Operatorët logjik

- Kombinimi I më shumë se një kriteri në WHERE  
AND dhe OR mundësojnë këtë  
NOT na mundëson po ashtu që ta ndryshojmë sjelljen e kriterin
- Kur të bëhet kombinimi I kriterëve; probleme mund të shfaqen.
- Këto probleme zgjidhen me përdorimine kllapave

# Kllapat

Rishkruajmë:

```
SELECT id, colour from tblDress  
WHERE (colour = 'RED'  
OR colour = 'BLUE')  
AND id LIKE '%5%'
```

Kjo më poshtë është më e lexueshme:

```
SELECT id, colour from tblDress  
WHERE ( colour = 'RED'OR colour = 'BLUE')  
AND id LIKE '%5%'
```

# AND

- AND kombinon të gjitha kriteret, pra të gjithë kriteret duhet të plotësohen:

Shema relacionale :

<u>LibriID</u>	ISBN	Titulli	AutoriID
----------------	------	---------	----------

```
SELECT  LibriID,  
        ISBN,  
        Titulli,  
        AutoriID
```

```
FROM  dbo.tblLibri
```

```
WHERE LibriID =2 and ISBN=42154
```

	LibriID	ISBN	Titulli	AutoriID
1	2	42154	Programming in C#	2



# OR (ose)

- Pra cilido kriter që plotësohet atëherë rreshtat na paraqiten:

```
SELECT  LibriID,  
        ISBN,  
        Titulli,  
        AutoriID
```

	LibriID	ISBN	Titulli	AutoriID
1	1	12422	Data Base System	1
2	2	42154	Programming in C#	2

```
FROM  dbo.tblLibri
```

```
WHERE AutoriID =2 or
```

```
       Titulli='Data Base System'
```

# TOP N / TOP N PERSENT

- Select Top 5, queri kthen saktë 5 rekorde për atë queri
- Select Top 5 \* from tblPuntori ;
  - Kthen sakt 5 rekorde
- Select Top 30 Percent, queri kthen 30% nga të dhënat që gjenden në tabel
- Select Top 30 Percent from tblPuntori;
  - Kthen perafersisht 3 rekorde nga 9 rekorde komplet

# BETWEEN

- Select \* From tblPuntori  
Where IDPuntori between 3 and 8
  - Rezultati do jetë ( 3,4,5,6,7 dhe 8)
- Ky queri do mund të realizohet edhe me anë të këtoj queri:
- Select \* From tblPuntori  
Where IDPuntori IN (3,4,5,6,7,8)

# Like ‘\_’

- Përdorim operatorin like kur deshirojm te specifikojm kerkimin,
- Me operatorin like ne kemi mundesi me ‘\_’ te zevendesojm nje shkronje:

- Select \*

From tblPuntori

Where Emri Like ‘\_A%’

Rezultati do jene te gjithë puntoret emrat e te cileve fillon me cfardo shkronje mirpo shkronja e dyte te jete sakte A.

# Like ‘%’

- Përdorim operatorin like kur deshirojm te specifikojm kerkimin,
- Me operatorin like ne kemi mundesi me ‘%’ te zevendesojm nje grup te shkronjave:

- `Select * from tblPuntori`  
`Where Emri Like ‘A%’`

Rezultati do jene te gjithë puntoret emrat e te cileve fillon me sakte me shkronjen ‘A’

# AS (alias)

- Kemi mundesi perdorimin e AS nese deshirojme te perdorim emer te ndryshem per rekordin ne klauzulen Select, I cili emrim eshte I perkoheshem:
- ```
Select (Sasi * Cmimi) as Vlera  
From tblArtikulli
```
- ```
Select p.PuntoriID,  
       p.Emri,  
       pa.Paga  
From tblPuntori as p, tblPaga as pa
```

# IS NULL

- Ne e dim se cdo rekord mund te kete vleren null.  
Per te kontrolluar keto vlera perdorim IS NULL.
- `Select * from tblPuntori`  
`Where KomunaID is null;`
  - Rezultati do paraqet te gjithë puntoret ku nuk kene te ruajtur komune
- `Select isnull(paga,0) as Paga`  
`From tblPaga`
  - Rezultati do paraqet per rekordin paga edhe nese nuk ka te dhene do kete vleren 0

# NOT

NOT inverton rregullon të dhënë në vijim

Titulli='Data Base System'

Ky kriter mund të invertohet në dy mënyra

- WHERE ISBN !=42154
- WHERE not ISBN =42154

	LibriD	ISBN	Titulli	AutoriD
1	1	12422	Data Base System	1

NOT nuk është shumë ndihmues në këtë shembull, por nëse kemi kritere më të komplikuar atëherë mund ta shohim roline tij



# NOT

- Operatori NOT eshte rezervuar per manipulime me operatoret logjik, ne te cilet eshte përdorur si:
  - NOT BETWEEN
  - NOT IN
  - NOT LIKE
  - IS NOT NULL
  - NOT EXIST
  - NOT UNIQUE

# Shembuj

- Select \* from tblPuntori  
Where komunaID NOT IN (3,4);
- Select \* from tblPuntori  
Where Paga not between 200 and 300
- Not like ( where rekord not like 'A%')
- Is not null (where rekord is not null)
- Not Exists ( where rekord not exists)
- Not Unique( where rekord not Unique)

# Prioritetet

Fjala është për prioritete te evaluimit të kriterëve të filtrimit

- Prioritet nuk merret nga rradhitja e kriterëve
- Kriteret kombinohen bashkë së pari AND, pastaj OR dhe përfundimisht NOT
- Konsideroni:

```
SELECT id, colour
From tblDress
WHERE colour = 'RED'           --Line 1
      OR colour = 'BLUE'      --Line 2
      AND id LIKE '%5%'       --Line 3
```

# Pa 'DISTINCT'

- Te paraqiten të gjith punetoret e regjistruar ne sistem:

```
select *  
from tblPunetori;
```

	NrPersonal	Emri	EmriPrindit	Mbiemri
1	1	Agim	N	Gashi
2	2	Edona	B	Shemsedini
3	3	Agim	H	Hasani

# Me 'DISTINCT'

- Te paraqiten të gjith punetoret e regjistruar ne sistem:

```
Select distinct Emri  
from tblPunetori;
```

	Emri
1	Agim
2	Edona

- Kur kemi një list të OR për të njëjtin atribut (kolonë), atëherë operatori IN ofron mënyrë më të lehtë të punës.

- Në vend të:

```
SELECT  Biblioteka, Adresa, Shenim
FROM    tblBiblioteka
WHERE   (Adresa = 'Prishtine') OR
        (Adresa = 'Ferizaj')
```

	Biblioteka	Adresa	Shenim
1	Biblioteka-1	Prishtine	2121
2	Biblioteka-3	Prishtine	2123
3	Biblioteka-4	Ferizaj	2124

- Mund ta bëjmë:

```
SELECT  Biblioteka, Adresa, Shenim
FROM    tblBiblioteka
WHERE   Adresa in( 'Prishtine','Ferizaj')
```

# Përzgjedhësit e ndërthurur

- Përzgjedhës të ndërthurur quhen përzgjedhësit të cilët në ndërtimin e vetë përmbajnë përzgjedhës.
- Përzgjedhësit e “brendëshëm” quhen nënpërzgjedhës (ang. subquery).

```
SELECT S.sname  
FROM Sailors S  
WHERE S.sid IN (SELECT R.sid  
FROM Reserves R  
WHERE R.bid=100)
```

```
SELECT R.sid  
FROM Reserves R  
WHERE R.bid=100
```

## Shema Relacionale:

Sailors (sid: integer, sname: string, rating: integer, age: real)

Boats (bid: integer, bname: string, color: string)

Reserves (sid: integer, bid: integer, day: date)

# Përzgjedhësit e ndërthurur

- SELECT S.sname  
FROM Sailors S  
WHERE EXISTS (SELECT \*  
FROM Reserves R  
WHERE R.bid=103 AND S.sid=R.sid)



# Përzgjedhësit e ndërthurur

```
SELECT S.sname,  
       (SELECT r.rid  
        FROM Reserves R  
        WHERE R.bid=103 AND S.sid=R.sid) as RID,  
FROM Sailors S
```

# ANY - ALL

- ANY – do të thotë plotësohet kushti për ndonjë element (në rastin e relacioneve për ndonjë n-she) dhe
- ALL – do të thotë plotësohet kushti për të gjithë elementet (në rastin e relacioneve të gjithë n-shet)

# ANY

- Gjej të dhënat për puntor me koefficientin më të madhe se sa koefficienti i ndonjërit nga puntoret me emrin Agim.

```
SELECT *  
FROM Puntori S  
WHERE S.paga > ANY (SELECT S2.paga  
FROM Puntori S2  
WHERE S2.emri='Agim')
```

# ALL

- Të gjenden të dhënat për punetoret me koefficientin më të larte se koefficienti i çdonjërit punetor me emrin Agim.

```
SELECT *  
FROM Puntor S  
WHERE S.paga > ALL (SELECT S2.paga  
FROM Puntor S2  
WHERE S2.emri ='Agim')
```

- Të gjenden të dhënat për puntorin me koefficient më të larte nga të gjithë puntoret tjerë.

```
SELECT *  
FROM Puntor S  
WHERE S.paga >= ALL (SELECT S2.paga  
FROM Puntor S2)
```

# Funksione agregate

- Funksione agregate janë funksione matematikore të cilët veprojnë në ndonjë attribute nga ndonjë grupë e të dhënave nga database-i.
- Funksionet agregate më shpesh të përdorur janë:
  - COUNT()
  - MIN()
  - MAX()
  - SUM()
  - AVG()

# COUNT

- COUNT([DISTINCT] \* | shprehje)
- Jep numrin e n-sheve të zgjedhura.
- NULL vlera nuk hynë në llogaritje.
- **SELECT COUNT(*expression*)**  
**FROM tables**  
**WHERE predicates;**

➤ Të gjendet numri i puntoreve me page mbi 350:

```
SELECT COUNT(*) as "Number of employees"  
FROM employees  
WHERE salary > 350;
```

# MAX

- MAX(shprehje)
- Vlerën maksimale të atributit (në kolonë) për n-shet e zgjedhura.
- NULL vlera nuk hynë në llogaritje.
- Të gjendet emri i puntoreve më të moshuar:

– SELECT S.emri, MAX (S.mosha)  
FROM tblPunetori S

Nuk funksionon

– SELECT S.emri, S.mosha  
FROM tblPunetori S  
WHERE S.mosha = (SELECT MAX (S2.mosha)  
FROM Sailors S2)



# MIN

- MIN(shprehje)
- Vlerën minimale të atributit (në kolonë) për n-shet e zgjedhura.
- NULL vlerat nuk hynë në llogaritje.
- *SELECT MIN(expression )*  
*FROM tables*  
*WHERE predicates;*

```
SELECT MIN(salary) as "Paga me e vogel"  
FROM employees;
```

# SUM

- SUM([DISTINCT] shprehje)

Funksioni llogarit shumën e vlerave të atributi ose shprehjes së dhënë për n-shet e zgjedhura.

NULL vlerat nuk hynë në llogaritje.

- *SELECT SUM(expression )*  
*FROM tables*  
*WHERE predicates;*
- SELECT SUM(salary) as "Total"  
FROM employees  
WHERE salary > 15000;

# AVG

- AVG([DISTINCT] shprehje)
- Vlera mesatare e atributit numerik ose shprehjes së dhënë për n-shet e relacionit.
- NULL vlera nuk hynë në llogaritje.
- `SELECT AVG(column_name)`  
`FROM table_name`
- Të gjendet mosha mesatare e puntoreve:
  - `SELECT AVG (mosha)`  
`FROM TblPunetori`

?