

DATABAZA

Databaza është një grumbull informacioni i organizuar në mënyrë të tillë që thjeshton aksesimin, menaxhimin dhe freskimin e të dhënave.

Databaza përbëhet nga **tabela** të cilat ruajnë informacione të ndryshme.

Çfarë është SQL?

Në qoftëse kuptoni çfarë është databaza, SQL është e thjeshtë.

SQL është akronim i fjalëve në gjuhën angleze **Structured Query Language**.

SQL përdoret për aksesimin dhe manipulimin e databazave.

MySQL është një **program** që kupton **SQL**.

SQL mundet:

- insertojë, updatojë, ose fshijë rekorde në një databazë.

- krijojë databazat, tabelat, të ruajë procedurat.

- të marri të dhënat nga një databazë etj.

SQL është **standart ANSI** (American National Standards Institute), por ka versione të ndryshme të gjuhës **SQL**. Shumica e këtyre gjuhëve kanë elemente shtesë të strukturës së gjuhës megjithatë komandat bazë janë të njëjta.

TABELAT E DATABAZËS

Një tabelë ruan dhe shfaq të dhënat në një format të strukturuar që konsiston në kolona dhe rreshta.

Databazat zakonisht përmbajnë shumë tabela, secila e projektuar për një qëllim specifik. Për shembull, imagjinoni një tabelë databazë që mban të dhënat e **id**, **lenda**, **nota**. Së pari, ndërtohen kolonat me emrat e tyre: **id**, **lenda**, **nota**.

Çdo tabelë ka fushën e vet të përcaktimit të të dhënave, në varësi të të dhënave që do të ruhen.

+ Options

id	lenda	Nota
1	HTML	10
2	CSS	8
1	HTML	10
2	CSS	8
3	CSS	7
4	JAVASCRIPT	7
5	SQL	10

Instrukcioni **CREATE TABLE** përdoret për krijimin e një table të re. Krijimi i një table përfshin emërtimin e tabelës, përcaktimin e kolonave dhe tipeve të të dhënave të secilës kolonë.

Sintaksa bazë për instruksionin **CREATE TABLE** është si më poshtë:

```
CREATE TABLE emri_tabelës
(
emri_kolones1 tipi_i_te_dhenes(permasa),
emri_kolones2 tipi_i_te_dhenes(permasa),
emri_kolones3 tipi_i_te_dhenes(permasa),
....
emri_kolonesN tipi_i_te_dhenes(permasa)
);
```

- **emri_kolones** specifikon emrin e kolonës që do të krijohet.
- **tipi_i_te_dhenes** specifikon çfarë tipi të dhëne do të mbajë kolona. Për shembull, përdoret int për numrat e plotë.
- Parametri **përmasa** specifikon gjatësinë maksimale të kolonës së tabelës.

Shënim: kolonat ndahen me presje.

Supozojmë se doni të krijoni tabelën 'përdorues' e cila konsiston në katër kolona: *përdoruesID*, *emri*, *mbiemri*, dhe *qyteti*.

Instrukcioni MySQL:

```
CREATE TABLE përdorues
(
përdoruesID int,
emri varchar(100),
```

```
mbiemri varchar(100),
qyteti varchar(100)
);
```

Varchar është tipi i të dhënës që ruan karakteret. Numri i karaktereve specifikohet në kllapa. Pra, fushat në shembullin e mësipërm, mund të mbajnë maksimumi tekste me 100 karaktere gjatësi.

Tipet e të dhënave

Tipi i të dhënave specifikon një tip të dhënës për një kolonë të caktuar.

Tipet më të zakonshme të të dhënave:

Numerike:

INT – Numrat e plotë.

FLOAT(M,D) – Numrat me presje notuese. Mund të përcaktohet edhe gjatësia (M) dhe numri i shifrave pas presjes (D).

DOUBLE(M,D) – Numrat me presje notuese. Mund të përcaktohet edhe gjatësia (M) dhe numri i shifrave pas presjes (D).

Data dhe Ora:

DATE - Data në formatin YYYY-MM-DD.

DATETIME – Një kombinim i datës dhe orës YYYY-MM-DD HH:MM:SS.

TIMESTAMP – Një timestamp, që nga mesnata e January 1, 1970

TIME – Ruan orën në formatin HH:MM:SS.

Tipi i stringjeve:

CHAR(M) – String karakteresh me gjatësi fikse maksimumi 255 bytes.

VARCHAR(M) – String karakteresh me gjatësi variablash. Vlera maksimale e përcaktuar në kllapa.

BLOB - "Binary Large Objects" përdoren për ruajtjen e sasive të mëdha të dhënash binare, si imazhet ose tipet e tjera të të dhënave.

TEXT – Sasi e madhe të dhënash tekst.

Zgjedhja korrekte e tipit të të dhënës për kolonën e tabelës është çelësi për një dizajnim të mirë databaze.

Specifikimi i rregullave për të dhënat në tabela

Për shembull në përcaktimin e kolonës 'emri' nuk lejohen vlerat boshe:

emri varchar(100) NOT NULL

NOT NULL – Kolona nuk mund të përmbajë vlerë boshe.

UNIQUE – Nuk lejohet futja e të dhënave të dublikuara. Një tabelë mund të ketë më shumë se një kolonë tabele unike.

PRIMARY KEY – Krijon një indeks unik për aksesim më të shpejtë të tabelës.

CHECK – Kontrollon nëse një vlerë është e vlefshme ose jo në një shprehje logjike.

DEFAULT – Gjatë ndërfutjes së të dhënave në tabelë, nëse nuk vendoset asnjë vlerë në tabelë vendoset vlera DEFAULT.

AUTO INCREMENT - Auto-increment lejon gjenerimin e një numri unik kur shtohet një record i ri në tabelë. Apriori, vlera startuese për

AUTO_INCREMENT është 1, dhe do të shtohet me 1 sa herë shtohet një rekord.

Çelësat Primarë

Një çelës primar është një fushë në tabelë e cila identifikon në mënyrë unike rreshtat(rekordet) e tabelës.

Karakteristikat kryesore të çelësve primarë:

- Duhet të kenë një vlerë unike për çdo rresht.
- Nuk mund të kenë vlera **NULL** (boshe).

Për shembull, tabela më poshtë përmban një record për çdo emër . Numri unik **ID** është zgjedhja e duhur për t'u zgjedhur si çelës primarë i tabelës, sepse është i mundur rasti kur janë dy persona me të njëjtat rezultate në të njëjtën lëndë.

+ Options

id	lenda	Nota
1	HTML	10
2	CSS	8
1	HTML	10
2	CSS	8
3	CSS	7
4	JAVASCRIPT	7
5	SQL	10

- Tabelat kanë vetëm nga **një** çelës primarë secila.
- Vlera e çelësit primarë duhet të jetë të ndryshme për secilin rast.

Çelësat primarë mund të përcaktohen gjatë krijimit të tabelës, duke përdorur fjalën kyçe **PRIMARY KEY** dhe duke vendosur brenda kolonave emrin e kolonës që do të jetë **çelësi primar**.

```
CREATE TABLE klasa
(
id int,
lenda varchar(100),
nota int(2),
PRIMARY KEY(id)
);
```

Komandat bazë SQL

Mënyra më e thjeshtë për të përdorur MySQL dhe PHPMyAdmin është të instaloni program si **XAMPP** ose **WAMP**. Këto mjete janë të thjeshta dhe pa kosto.

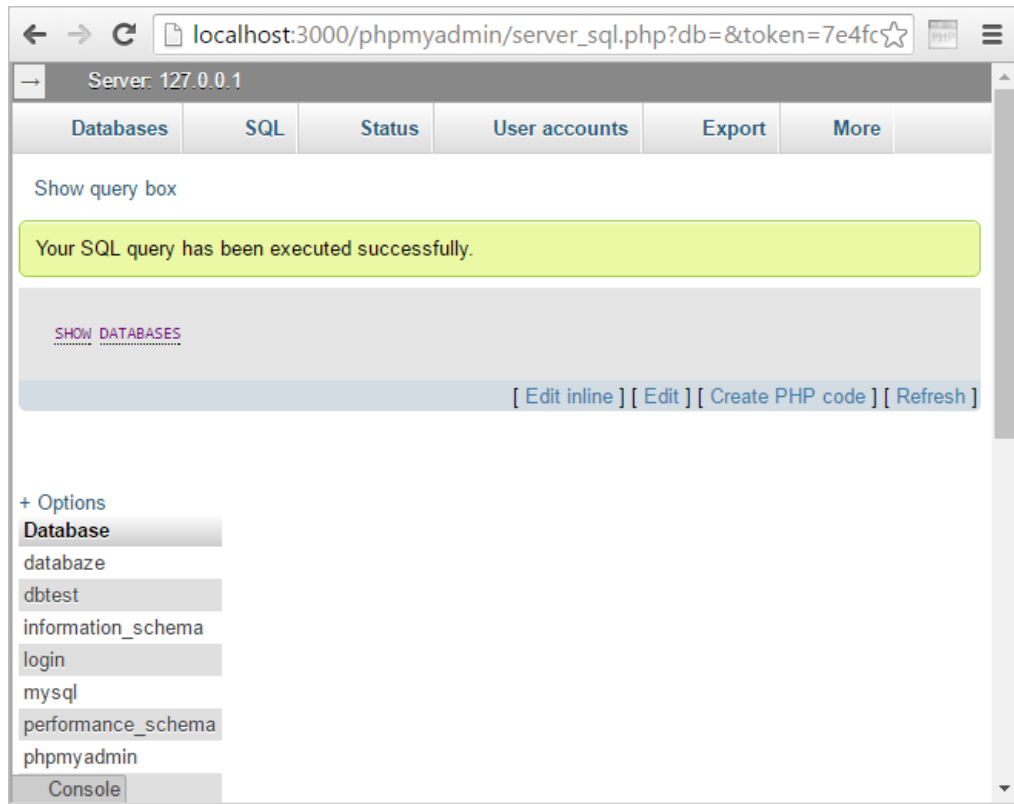
Komanda SQL **SHOW** shfaq të dhënat që ndodhen në databazë dhe në tabela.

Për shembull, komanda **SHOW DATABASES** liston databazat e menaxhuara nga serveri.

Hapni **PHPMyAdmin**, dhe klikoni në tabin **SQL** për të hapur një dritare të thjeshtë ku të shkruani komandat SQL.

Shkruani **SHOW DATABASES** dhe klikoni "Go". Programi do t'ju njoftojë se SQL-ja juaj është ekzekutuar me sukses (*Your SQL query has been executed*

successfully). Do të listohen të gjithë databazat e PHPMyAdmin si në figurën më poshtë:



Komanda **SHOW TABLES** përdoret për shfaqjen e të gjithë tabelave në databazën MySQL të selektuar.

Rezultati:

Server: 127.0.0.1 » Database: databaze

Structure SQL Search Query Export Import More
[Edit inline] [Edit] [Create PHP code] [Refresh]

+ Options
Tables in database
nxenes

Query results operations
Print view Create view

Bookmark this SQL query
Label: Let every user access this bookmark

Console

Si shembull, është krijuar databaza, **databaze** me tabelën e quajtur **nxënës**.

Komanda **SHOW COLUMNS** shfaq informacione përreth kolonës së një table. Shembulli më poshtë shfaq kolonat të **tabelës nxenes**:

SHOW COLUMNS FROM nxenes

Rezultati:

+ Options

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
emri	varchar(11)	NO		NULL	
mbiemri	varchar(11)	NO		NULL	
mosha	int(11)	NO		NULL	

SHOW COLUMNS shfaq vlerat e mëposhtme për secilën kolonë të tabelës:

Field: emri kolonës

Type: tipi i të dhënës së kolonës

Key: tregon se ku është indeksuar kolona

Default: vlera default shenjuar kolonës

Extra: informacione shtesë

Komanda **SELECT** përdoret për selektimin e të dhënave nga një databazë.

Rezultati ruhet në një tabelë, e cila quhet **result-set**. Për përdorimin e komandës **SELECT**, specifikojmë edhe emrat e kolonave prej të cilave duam të marrim informacion.

Sintaksa e komandës **SELECT**:

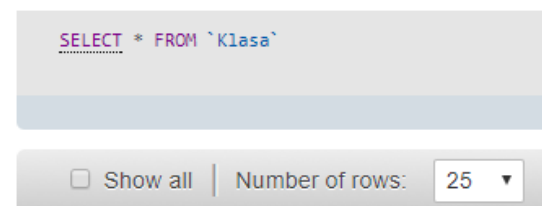
SELECT lista_kolonave

FROM emri_tabelës

- **lista_kolonave** përfshin një ose më shumë kolona

- **emri_tabelës** emri i tabelës së cilës i përket kolonat

Më poshtë paraqiten të dhënat nga tabela **klasa**:



+ Options

id	lenda	Nota
1	HTML	10
2	CSS	8
1	HTML	10
2	CSS	8
3	CSS	7
4	JAVASCRIPT	7
5	SQL	10

Komanda e mëposhtme SQL selekton **lenden** nga tabela **klasa**:

SELECT `lenda` FROM `Klasa`


```
SELECT `lenda` FROM `klasa`
```

Show all | Number of rows: 25 ▼

+ Options

lenda

HTML

CSS

HTML

CSS

CSS

JAVASCRIPT

SQL

Queri-të shumëfishe

SQL lejon shumë query ose shumë komanda në të njëjtën kohë.

Komandat e mëposhtme SQL lejojnë selektimin nga kolonat **emri** dhe **mbiemri** nga tabela:

```
SELECT lenda FROM klasa;
SELECT nota FROM klasa;
```

Mos harroni të ndani komandat nga njëra-tjetra me **pikëpresje**.

Case Sensitivity

SQL është case **insensitive** dmth e njëjta komandë SQL mund të shkruhet si me shkronja të mëdha ashtu edhe me të vogla.

Për shembull kodet e mëposhtme do të prodhojnë të njëjtin rezultat:

```
Select emri from nxenes;
SELECT Emri FROM nxenes;
select Emri FROM nxenes;
```

Megjithatë si rekomandim praktik është mirë që komandat SQL të shkruhen më shkronja të mëdha.

Rregullat e sintaksës

Një komandë SQL e vetme mund të shkruhet në një ose më shumë rreshta. Për më tepër, komandat shumëfishe SQL mund të kombinohen në një rresht të vetëm.

Hapsirat boshe dhe rreshtat shumëfishe injorohen në SQL.
Për shembull, query-a e mëposhtme është e saktë .

```
SELECT lenda  
FROM klasa;
```

Megjithatë, rekomandohet eliminimi i hapsirave dhe rreshtave të panevojshme.

Selektimi i disa kolonave

Siç u përmend edhe më lart, komanda SQL **SELECT** merr rekordet nga tabela nga databaza SQL.

Mund të selektohen disa kolona njëherësh

Tjeshht listohen emrat e kolonave , të ndara nga **presje**:

```
SELECT emri, mbiemri, mosha  
FROM nxenes;
```

Mos vendosni presje pas emrit të kolonës së **fundit**.

Selektimi i të gjithë kolonave

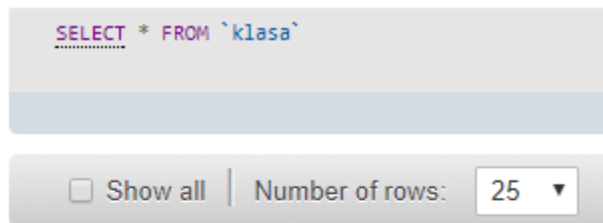
Për marrjen e të gjithë informacionit që ndodhet në tabelën tuaj, vendosni një shenjë yll (*) pas komandës **SELECT**, në vend të listimit të të gjithë emrave të

kolonave.

Komanda mëposhtme SQL selekton të gjithë kolonat në tabelën klasa:

```
SELECT * FROM klasa;
```

Rezultati:



+ Options

id	lenda	Nota
1	HTML	10
2	CSS	8
1	HTML	10
2	CSS	8
3	CSS	7
4	JAVASCRIPT	7
5	SQL	10

Në SQL, ylli do të thotë të **gjitha**.

Fjala kyçe DISTINCT

Në situatat në të cilat ju keni rekorde të dublikuara në tabelë, do të ishte më mirë të bashkonim të gjitha në një rekord të vetëm. Komanda SQL **DISTINCT** përdoret së bashku me komandën **SELECT** për eliminimin e rekordeve të dublikuara dhe lënien vetëm të atyre unike.

Sintaksa bazë e **DISTINCT** është si më poshtë :

```
SELECT DISTINCT emri_kolon1, emri_kolon2
FROM emer_tabele;
```

Theksojmë se janë emra të duplikuar. Komandat e mëposhtme SQL selektojnë vetëm vlerat që nuk janë të dublikuara nga kolona emri:

```
SELECT DISTINCT emri FROM nxenes;
```

Fjala kyçe LIMIT

Apriori, të gjitha rezultatet që plotësojnë kushtet e vendosuar në SQL do të kthehen. Megjithatë, ndonjëherë duhet të kthehet një numër i caktuar rekordesh. Në MySQL, kjo realizohet duke përdorur fjalën kyçe LIMIT.

Sintaksa:

```
SELECT lista kolonave  
FROM emri_tabeles  
LIMIT [numri_rekordeve];
```

Gjithashtu mund të zgjidhen disa rekorde nga një pjesë e caktuar. Në shembullin e mëposhtëm, zgjidhen **katër** rekorde, duke filluar nga pozicioni i të tretit:

```
SELECT ID , emri, mbiemri, mosha  
FROM nxenes LIMIT 3, 4;
```

Emrat e kualifikimit (Fully Qualified Names)

Në SQL, ju mund të përdorni emrin e një table para emrit të një kolone të ndarë nga një **pikë**.

Për shembull, komandat e mëposhtme janë të njëvlershme.

```
SELECT emri FROM nxenes;  
SELECT nxenes.emri FROM nxenes;
```

Kjo lloj sintakse quhet "**fully qualified name**" e kolonës.

Kjo mënyrë shkrimi është efiçente kur punojmë me shumë tabela të cilat mund të kenë emra kolonash të njëjta

Komanda Order By

ORDER BY përdoret për selektimin dhe radhitjen e të dhënave të marra. Shembulli i mëposhtëm rendit tabelen nxenes sipas kolonës emri.

```
SELECT * FROM klasa
ORDER BY nota ;
```

Rezultati:



+ Options

id	lenda	Nota
3	CSS	7
4	JAVASCRIPT	7
2	CSS	8
2	CSS	8
1	HTML	10
1	HTML	10
5	SQL	10

Sic e shikoni, rreshtat janë radhitur në rend rrites sipas kolonës nota.

Apriori, komanda **ORDER BY** rendit rezultatet sipas **rendit rritës**.

Radhitja e shumë kolonave

Komanda **ORDER BY** mund t'i radhisë të dhënat në varësi të shumë kolonave. Kur përdoret **ORDER BY** me më shumë se një kolonë, ndahet lista e kolonave me presje.

Për të radhitur sipas **emrit** dhe **moshës**:

```
SELECT * FROM nxenes
ORDER BY mbiemri, mosha;
```

Të dhënat radhiten fillimisht sipas kolonës së parë më pas të dytës, të tretës e kështu me radhë.

WHERE

WHERE përdoret për ekstratimin e atyre rekordeve të cilat përmbushin një kriter të caktuar.

Sintaksa përdorimit të **WHERE** :

```
SELECT lista_kolones  
FROM emri_tabeles  
WHERE kushti;
```

Për të selektuar një record specifik:

```
SELECT * FROM klasa  
WHERE ID = 5;
```

Rezultati:

```
SELECT * FROM klasa WHERE ID = 5
```

Show all | Number of rows: 25 ▼

+ Options

id	lenda	Nota
5	SQL	10

Operatorët SQL të krahasimit

Operatorët e krahasimit dhe Operatorët logjik përdoren bashkë me WHERE për të filtruar të dhënat e selektuara.

Operatori	Përshkrimi
=	barabartë
!=	JO barabartë
>	Më i madh
<	Më i vogël
>=	Më i madh ose i barabartë
<=	Më i vogël ose i barabartë
BETWEEN	midis vlerash të caktuara

Për shembull , ne mund të shfaqim listën e të gjithë emrave të nxënësve , me përjashtim të atyre me ID=4:

```
SELECT * FROM lenda
WHERE ID != 4;
```

Rezultati:

```
SELECT * FROM klasa WHERE ID != 4
```

Show all | Number of rows: 25 ▼

+ Options

id	lenda	Nota
1	HTML	10
2	CSS	8
1	HTML	10
2	CSS	8
3	CSS	7
5	SQL	10

Siç duket, rekordi me ID = 4 nuk është përfshirë në listë.

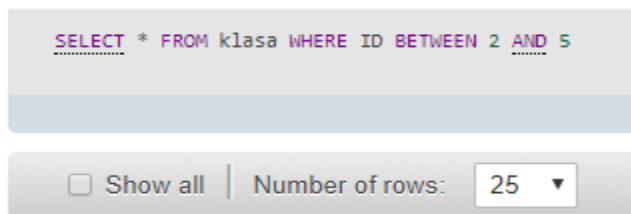
Operatori BETWEEN

Operatori BETWEEN selekton vlerat brenda një kufiri vlerash. Sintaksa e përdorimit të operatorit BETWEEN është si më poshtë :

```
SELECT emri_i_kolones
FROM emri_tabeles
WHERE emri_kolones BETWEEN vleres1 AND vleres2;
```

Shembull, selektimi i të gjithë rekordeve me ID që bien brenda segmentit 3 dhe 7:

```
SELECT * FROM klasa
WHERE ID BETWEEN 2 AND 5;
```



+ Options

id	lenda	Nota
2	CSS	8
2	CSS	8
3	CSS	7
4	JAVASCRIPT	7
5	SQL	10

Instruksioni i mëposhtëm SQL selekton të gjithë rekordet ku nota është e barabartë me '8'

```
SELECT* FROM klasa WHERE nota = '8';
```

Rezultati:


```
SELECT* FROM klasa WHERE nota = '8'
```

Show all | Number of rows: 25 ▼

+ Options

id	lenda	Nota
2	CSS	8
2	CSS	8