**УНИВЕРЗИТЕТ ,,ГОЦЕ ДЕЛЧЕВ” – ШТИП**

**ФАКУЛТЕТ ЗА ИНФОРМАТИКА**

**A picture containing calendar

Description automatically generated**

**СЕМИНАРСКА РАБОТА**

**Тема : MySQL**

Ментор : Проф. Цвета Мартиновска Банде

Изработил : Мартин Ристов - 102620

**ЈУНИ 2023**

**Содржина**

[**MySQL** 3](#_Toc137136886)

[**MySQL и Microsoft Access** 3](#_Toc137136887)

[**MySQL синтакса** 4](#_Toc137136888)

[**Работа во MySQL** 5](#_Toc137136889)

[**Користење на SELECT , COUNT и WHERE клаузулите** 5](#_Toc137136890)

[**Креирање на дата бази и табели и манипулирање со нив** 8](#_Toc137136891)

[**Заклучок** 13](#_Toc137136892)

[**Користена литература** 14](#_Toc137136893)

# **MySQL**

## **MySQL и Microsoft Access**

MySQL е систем на бази на податоци на вебот.Основно MySql датабазата ви овозможува да креирате релациони датабазни структури на веб-сервер со цел да чувате податоци или да ги автоматизирате процедурите.Доколку на MySQL размислуваме во споредба со Microsoft Access, MySQL и она што ги чува сите ваши табели, PHP актите какви кверија, и формите се во основа вебстрани со полиња во нив. Со сето ова комбинирано, можеме да креираме вистински спектакуларни проекти на веб.

MySQL е истотака со отворен код и е слободен за користење, и истотака спаѓа под GNU General Public License (GPL).Па така доколку сакаме да направиме наша веб страна или веќе имаме една, нашиот хост со сигурност би требало да подржува MySQL и PHP.Тие генерално се асоцирани со Unix/Linux базираните сервери.

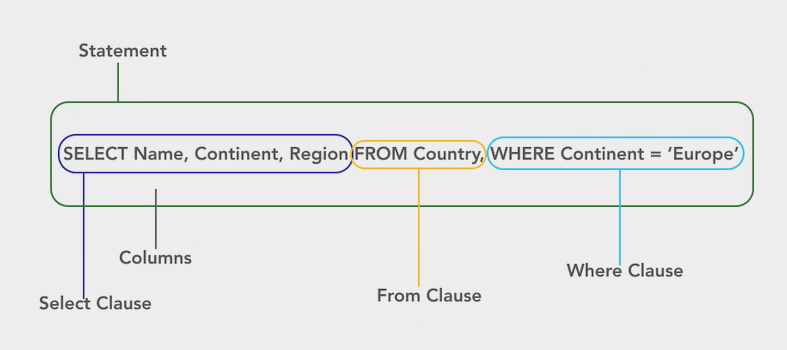
Доколку првпат се среќаваме со MySQL датабазите малце ќе ни биде чудна работата со нив, доколку сме навикнати на Microsoft Access и WYSIWYG интерфејсот. Тука за креирање на табели, ќе треба да го правиме тоа со користење на SQL наредби или со користење на друга опен сорс алатка наречена PhpMyAdmin, која ни дава лесен за користење интерфејс кој ни овозможува креирање на табели и извршување на кверија со потполнување на минимални информации.

Доколку ги споредуваме MySQL и Access ќе наидеме на многу разлики.Дедека со MySQL не е толку тешко да се справиме (откако табелата е направена - работата е попричично завршена),неговите способности се многу поголеми од она што Microsoft Access може да го понуди кога се работи за брзина и флексибилност.Тоа е едноставно подобар систем, бидејки Microsoft Access претставува само десктоп систем за бази на податоци.Крајно може да се каже дека Microsoft Access е добар доколку имаме мала фирма, но доколку имаме вебстрана која има 10.000 прегледи дневно и се извршуваат огромен број на кверија, ќе ни биде потребно нешто помоќно кое што може да ги изврши сите тие кверија ефикасно.

Истотака MySQL табелите се битно подобри и полуксузни. Пример доколку ни е потребно поле во кое можат да се сместат 4 милијарди карактери- за MySQL тоа не претставува никаков проблем, потребно е само да го искористиме LongText податочниот вид и работата е повеќе од завршена.Доколку ни е потребно полето да содржи многу карактери кој ке бидат осетливи на големина, LongBlob податочниот вид си ја врши работата како што треба. Доколку пак ни треба да зачуваме број помеѓи 0 и 18квинтилиони, не е проблем BigInt податочниот тип ке ја чува таа информација без никаков проблем.

Како што гледаме конкуренцијата на Microsoft Access бргу отпага при споредба со MySQL.И ова е токму една од причините зошто многу големи веб страни го користат MySQL за да се справат со милионите посети дневно, секако и е бесплатно плус!

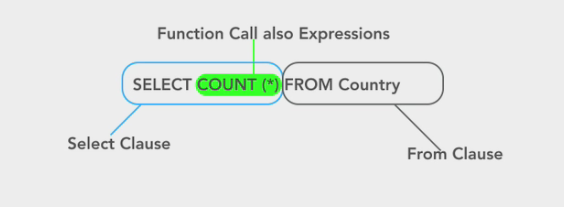
## **MySQL синтакса**

Како што рековме претходно MySQL е најпознатата верзија на SQL, пред се бидејки е бесплатна и со отворен код, достапна е за сите платформи, и е многу брза и флексибилна , а истотака е и доста безбедна.Истотака Тригерите кој се нова опција во MySQL од верзијата 5.0, овозможуваат да внесуваме одредена логика првнесувањето,обновувањето или бришењето на податоците.

MySQL синтакса

На сликата погоре можеме да видиме како изгледа синтаксата на MySQL. Значи најпрво со SELECT клаузулата ги селектираме колоните кој ни се потребни, наредно што е потребно е да специфираме од кој табели да ги селектираме тие колони со клаузулата FROM.На крај ставаме услов со кој определуваме кој податоци сакаме да бидат селектирани (условот во целост се однесува на редот), во случајов ги селектираме колоните Name,Continent и Region од табелата Country, каде што цредноста во во колоната Continent е Europe.

Во MySQL изјавите можеме да користиме и дефинирани функции.На сликата подолу е прикажано како изгледа тоа.



Користење на Функции 1

На сликата погоре е прикажано користењето на COUNT функцијата ,и целата изјава значи да се пребројат сите редови во табелата Country.

Од гореспоменатото можеме да заклучиме дека SQL изјавите можат да се состојат од клаузули,експресии и функции.

## **Работа во MySQL**

Во овој дел преку примери ке се запознаеме со основната работа во MySQL, преку постепено запознавање со сите клаузули.

## **Користење на SELECT , COUNT и WHERE клаузулите**

SELECT е една од најупотребуваните клаузули што се користат во SQL.

**Пример 01:**

Имаме табела Sveti\_Nikole ,и треба да ги прикажеме сите податоци во таа табела:

**SELECT \* FROM Sveti\_Nikole**

Овој едноставен ред на код значи Селектирај (**SELECT**) Се (**\***) Од (**FROM**) **Sveti\_Nikole.**

Значи \* значи се.

**Пример 02:**

Имаме за задача да пресметаме колку редици има во табелата Sveti\_Nikole:

**SELECT COUNT(\*) FROM Sveti\_Nikole**

Значи ова COUNT(\*) ни вели преброј ги сите редици, а останатиот дел од кодот кажува од каде.

**Пример 03:**

Од табелата Sveti\_Nikole (која се состои од колоните: Ime, Prezime, br\_index, prosek ....) треба да ги прикажеме само податоците од колоните Ime и Prezime :

**SELECT Ime, Prezime FROM Sveti\_Nikole**

сето ова на наш разбирлив македонски значи: Селектирај ги колоните Ime, Prezime. Значи синтаксата на SELECT наредбата е ова SELECT [имиња на колоните одвоени со запирка] FROM (имињата на табелите одвоени со запирка)

**Пример 04:**

Имаме табела Studenti (во која две колони kol1,во која се содржани имињата на студентите од 2ра година, и prezime во која се содржани презимињата на студентите од 2ра година). Имаме за задача да ни се прикажат тие податоци, но kol1 да ни ја пишува како Ime, и prezime како Student Prezime.

**SELECT kol1 AS Ime, prezime AS 'Student Prezime' FROM Studenti**

Ова значи селектирај (прикажи) ја kol1 Како Ime, prezime Како Student Prezime. Причината поради која Student Prezime  е ставено во единечни наводници е бидејки доколку ги изоставиме SQL-от ке го сфати тоа како **SELECT prezime AS Student** , a Prezime ке го земе како наредна команда која треба да се обработи, и бидејки таква команда не постои SQL-от ке јави грешка.

**Пример 05:**

Имаме табела UGD\_studenti (која ги содржи следниве колони: Ime, Prezime, Index, Godina, Fakultet ....). Задача ни е да ги прикажеме сите студенти кој учат на информатика заедно со нивните Имиња, Презимиња, Број На Индекс и Факултет.

**SELECT Ime, Prezime, Index, Fakultet FROM UGD\_studenti WHERE Fakultet = 'informatika'**

Ова значи Селектирај ги колоните Ime, Prezime, Index, Fakultet од UGD\_studenti каде вредноста во колоната Fakultet е "informatika"

**Пример 06:**

Имаме табела UGD\_studenti (која ги содржи следниве колони: Ime, Prezime, Index, Godina, Fakultet ....). Иаме за задача да ги прикажеме имињата на сите факултети и бројот на студенти во секој факултет. Имињата на факултетите ке бидат прикажани во Колоната fakultet, а бројот на студенти во колоната vkupno.

\*Размислуваме: треба најпрво да ја селектираме колоната Fakultet, бидејки така ни се бара, т.е да ги преброиме студентите по факултет. За да ги изброиме студентите ке ја искористиме колоната Index каде се сместени броевите на индексите на студентите (не користиме име или презиме бидејки има студенти со исти имиња и презимиња, а бројот на индекс е единствен како што е матичниот број).

**SELECT Fakultet AS fakultet, COUNT (Index) AS vkupno FROM UGD\_studenti GROUP BY Fakultet**

Овој пример треба да е повеќе од јасен, бидејки е искористено се од претходните.

## **Креирање на дата бази и табели и манипулирање со нив**

Креирањето на датабаза е поприлично едноставно за разлика од креирањето на табелите во нив. За да креирате датабаза едноставно впишете:

**CREATE DATABASE Nova\_Baza**

-Nova\_Baza е името на датабазата што сакаме да ја креираме.

Тоа е тоа.

**Пример 01:**

Да се креира табела UGD\_student која ке ги содржи следниве колони:

id - содржи реден број и мора да го има (несмее да биде NULL), автоматски да се инкрементира и да ни биде PRIMARY KEY

ime - ги содржи имињата на студентите со максимална должина од 25 карактери

prezime - ги содржи презимињата на студентите до 25 карактери

index - број на индекс на студентот

fakultet - првите 3 букви од името на факултетот (пр. информатика - inf)

**CREATE TABLE UGD\_student {**

**id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,**

**ime VARCHAR(25),**

**prezime VARCHAR(25),**

**index INTEGER,**

**fakultet CHAR(3)**

**};**

-првиот ред треба да е јасен (да напоменеме дека параметрите од табелата се поставуваат во блок {} )

-id INTEGER значи дека колоната id ке ни содржи целобројни вредности, полињата несмеат да бидат празни (NOT NULL) и вредноста автоматски ке се инкрементира (зголемува) (AUTO\_INCREMENT) и ова колона ке ни биде примарен клуч, што значи дека по неа ке се разликуваат сите останати

-разликата помегу CHAR и VARCHAR е тоа што кај VARCHAR вредноста во полињата може да биде во должина од 0-25 карактери, додека пак кај CHAR таа мора да е толку колку што сме навеле, односно во  нашиов случај, мора да е во должина од 3 карактери.

**Пример 02:**

Во претходно креираната табела (UGD\_student) да се вметне следниов ред (вредности):

 ime: Goce

prezime: Delcev

index: 007

fakultet: inf

**INSERT INTO UGD\_student**

**(ime, prezime, index, fakultet)**

**VALUES (**

**'Goce' ,**

**'Delcev' ,**

**'007' ,**

**'007'**

**)**

**Пример 03:**

Ја имаме табелата UGD\_student која ја креиравме претходно, и веке е пополнета со податоци. Наша задача е:

а) да ни се прикажат имињата, презимињата и броевите на индекс на сите студенти од факултетот за информатика што се викаат Martin

**SELECT ime, prezime, index, fakultet**

**FROM UGD\_student**

**WHERE ime = 'Martin'**

мислам дека го разбирате горе напишаното

б) да ни се прикажат имињата, презимињата , броевите на индекс и просекот на сите студенти од факултетот за информатика што имаат просек поголем од 8.0

**SELECT ime, prezime, index, fakultet, prosek**

**FROM UGD\_student**

**WHERE prosek >= 8.0**

в) да ни се прикажат имињата, презимињата и броевите на индекс на сите студенти чие што  презиме започнува на буквата N

**SELECT ime, prezime, index, fakultet, prosek**

**FROM UGD\_student**

**WHERE prezime LIKE 'N%'**

последниот ред ни е важен. Тука ја користиме "командата" LIKE, што значи селектираните колони од табелата UGD\_student каде prezime е слично со стрингот 'N%' ,т.е започнува со N и има други букви натака кој не ни се битни, т.е има било што понатаму (N.... (Nakov, Nikolov, Naumov ....))

-доколку сакаме да ги прикажеме сите презимиња што завршуваат на V пишуваме:WHERE prezime LIKE '%V'

-доколку пак сакаме да ги испечатиме сите имиња кој во себе било каде ја содржат буквата M:

**WHERE prezime LIKE '%М%'**

**Пример 04:**

Имаме Табела UGD\_student, која ја имаме креирано и потполнето претходно. Задача ни е:

a)да ги прикажеме сите студенти со нивните имиња, презимиња, броеви на индекс и факултети кој учат на факултетите: информатика, медицина и филолошки.

**SELECT ime, prezime, index, fakultet**

**FROM UGD\_student**

**WHERE fakultet IN ( 'inf' , 'med' , 'fil')**

значи од табелата UGD\_student да ни се прикажат податоците од селектираните колони, каде (WHERE) во колоната (IN) fakultet е содржана една од вредностите inf, med, fil

б)да се надополни претходната задача со тоа што ке ни се прикажат истите податоци, но за кој е исполнет условот, prosek>=9.0

**SELECT ime, prezime, index, fakultet**

**FROM UGD\_student**

**WHERE fakultet IN ( 'inf' , 'med' , 'fil')**

**AND prosek >= 9.0**

в) од табелата UGD\_student да се прикажат имињата презимињата, броевите на индекс, просекот и факултетот, на студентите од факултетот за информатика кој имаат просек помегу 8.0 и 9.0, и филолошкиот факултет кој имаат просек помегу 8.5 и 9.5 (студентите кој имаат услов за стипендија А), и податоците да се сортираат по факултет.

**SELECT ime, prezime, index, fakultet, prosek**

**FROM UGD\_student**

**WHERE fakultet = 'inf'**

**AND prosek >= 8.0 OR prosek <= 9.0**

**AND WHERE fakultet = 'fil'**

**AND prosek >= 8.5 OR prosek <= 9.5**

**ORDER BY fakultet**

**Пример 05:**

Имаме Табела UGD\_student, која ја имаме креирано и потполнето претходно. Задача ние:

1)да ги апдејтуваме (промениме) следниве податоци:

за id = 14

prosek = 8.2

status = red    (red - редовен , von - вонреден)

**UPDATE UGD\_student SET prosek = 8.2 , status = 'red' WHERE ID = 14**

б) да го избришеме целиот ред каде id=15

**DELETE FROM UGD\_student WHERE id = 15**

# **Заклучок**

Од сето ова споменато погоре, можеме да заклучиме дека базите на податоци играат огромна улога во ИТ сверата околу нас. Нивната примена е огромна и масивна, и без нивното постоење денеска многу работи не би биле такви ко што се. Пример не би имале вебсајтови кој што во исто време можат да примат милиони корисници, и во делови од секундата на сите да им ги достават бараните податоци.Или пак ебанкарството би било незамисливо без нивното постоење.Истотака видовме и дека работата со нив е поприлично едноставна, а сепак многу корисна, и поради тоа, како што е неопходно секој поединец кој што работи во ИТ сверата (програмер,ит инженер...) да го познава англискиот, истотака во денешно време се бара да знае да работи со бази на податоци.

**Користена литература**

Книгата по бази на податоци

<http://www.freebsddiary.org/mysql-capi.php>

<http://ucl.ac.uk/archaeology/cisp/database/manual/node1.html>

<http://code.google.com/edu/tools101/mysql.html>

<http://hr.wikipedia.org/wiki/Baza_podataka>

<http://php.com.hr/66>