

Факултет инжењерских наука Универзитета у Крагујевцу



Назив студијског програма: Машинско инжењерство

Ниво студија: Основне академске студије

Модул: Информатика у инжењерству

Предмет: Базе података

Број индекса: 190/2016

Стефан И. Милуновић

Систем за управљање базама података-MySQL

-завршни рад-

Комисија за преглед и одбрану:

1. Проф. др Ерић Милан-ментор

2. _____

3. _____

Датум одбране: _____

Оцена: _____



Универзитет у Крагујевцу
Факултет инжењерских наука



Основне студије: Основне академске студије

Студијско подручје: Машинско инжењерство-Информатика у инжењерству

Име и презиме: Стефан Милуновић

Број индекса: 190/2016

ПРИЈАВА ЗАВРШНОГ РАДА

Тема рада: *Систем за управљање базама података-MySQL*

Задатак: У оквиру овог завршног рада кандидат треба да проучи и презентира основне карактеристике, предности и мане, MySQL окружења за управљање базама података. Радом дати одговоре на питања зашто користити (перформансе, стабилност, поузданост, брзина итд.), како функционише, које су карактеристике MySQL-а. На конкретном примеру приказати рад у MySQL-у

Крагујевац, 15.06.2020

Ментор:


Проф. др Милан Д. Ерић



Изјава о самосталној изради рада

Изјављујем да сам овај завршни рад урадио самостално, служећи се стеченим знањем и искуством као и наведеном литературом.

Стефан Милуновић 190/2016

(потпис)

Резиме

Идеја овог завршног рада је да се проуче и презентирају основне карактеристике, предности и мане, MySQL окружења за управљање базама података. Дати одговоре на питања зашто користити, како функционише и које су карактеристике MySQL-а. У првом делу приказане су главне карактеристике MySQL-а као и структура SQL језика.

У другом делу је објашњено зашто користити MySQL и објашњене су предности и мане MySQL-а.

Пример се налази у трећем делу. Примером је представљен рад MySQL-а, у примеру је приказана употреба опције JOIN са 3 табеле.

У овом раду је представљен рад у окружењу MySQL.

Кључне речи: MySQL,табеле,предности,мане

Summary

The idea of this final paper is to study and present the basic characteristics, advantages and disadvantages of MySQL database management environments. Give answers to the questions why to use, how it works and what are the characteristics of MySQL. The first part shows the main characteristics of MySQL as well as the structure of the SQL language.

The second part explains why to use MySQL and explains the advantages and disadvantages of MySQL.

The example is in the third part. The example presents the work of MySQL, the example shows the use of the JOIN option with 3 tables.

This paper presents work in the MySQL environment.

Key words: MySQL, table, advantages, disadvantages

Садржај

1. Увод	1
2. Зашто користити MySQL ?	1
3. Како је дошло до појаве MySQL-а?	2
4. Главне карактеристике MySQL:	2
5. Структура језика	5
6. Конвенције које ће бити коришћене у овом материјалу.....	10
7. Зашто MySQL уместо неке друге базе података ?	25
8. Предности и мане MySQL-а	27
9. Пример	35
10. ЗАКЉУЧАК	38
11. ЛИТЕРАТУРА	39

1. Увод

MySQL је систем за управљање SQL базама података. База података је структурно повезан скуп података. ВР може бити било шта од обичне листе за куповину преко пописа слика уметничке галерије па све до големе количине података у корпорацијским мрежама. За додавање, приступање и обрађивање података у БП потребан је систем за управљање базама података као што је MySQL. Будући да се компјутери одлично сналазе при обради великих количина података, системи за управљање базама података играју централну улогу у рачунарству, као самостални алати или као део других апликација. MySQL се добро сналази и са релацијским базама података. Релацијска база података је база података која податке похрањује у одвојене табеле уместо у једну табелу. Ово омогућује бржи и флексибилнији рад са базама података. Табеле су повезане дефинисаним релацијама што омогућује комбиновање података из неколико табела у случају да постоји захтев за то. Реч SQL у MySQL значи "*Structured Query Language*" (структурни језик за претраживање) – најчешћи стандардизован језик за приступање базама података.[1]

MySQL је *open-source* производ, што значи да је дозвољено свакоме да га користи и прилагођава за своје потребе. Свакоме је омогућен *download* MySQL са Интернета и коришћење без обавезе плаћања. MySQL спада под GLP (*GNU General Public License*). Такође постоји MySQL *Enterprise* верзија програма која није бесплатна и која нуди додатне опције напредним корисницима.[1]

2. Зашто користити MySQL ?

Врло је брз, поуздан и лаган за коришћење. Такође има врло практичне додатне опције развијене у блиској сарадњи са корисницима. MySQL је оригинално развијан за манипулацију врло великих база података, много је бржи од постојећих решења и успешно се користи у високо захтевним окружењима. Приступачност, брзина и сигурност чине MySQL врло погодним за приступање базама података преко Интернета.[2]

Ако Вам треба брза, поуздана и флексибилна база података, рад са MySQL-ом биће остварење снова без обзира на примену.[2]

3.Како је дошло до појаве MySQL-а?

Људи који су основали данашњи MySQL-AB су првобитно желели да за повезивање табела у њиховим БП користе mSQL уз помоћ њихових бржих (ISAM) рутина. Али после исцрпних тестирања долазе до закључка да mSQL једноставно није довољно брз нити довољно флексибилан за њихове потребе. Ово је резултовало појавом новог CQL интерфејса за БП са скоро идентичним API-ем (*Application programming interface*) као код mSQL-а.[3]

Како је MySQL добио своје име није у потпуности јасно. Сви основни фолдери и велика количина lib-ова и алата су имали префикс “\my” више од десет година. Са друге стране ћерка *Michael "Monty" Widenius-a* се зове “My”. Тако да чак ни запосленици компаније не знају која је прича истинита.[3]

4.Главне карактеристике MySQL:

Следећа листа описује неке од важнијих карактеристика MySQL-а:

- Програм је оспособљен за потпуно искоришћење вишејезгарних процесора.
- C, C++, Elixir, Java, Perl, PHP, Python и Tcl APIs.
- Ради на много различитих платформи.
- Могућ је избор куцања од: потписаних или непотписаних интеропера дужине 1,2,3,4 и 8 Бајтова FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET и ENUM типова.
- Веома брзо придруживање коришћењем оптимизованог вишеструког повезника.
- Потпуна операциона и функцијска подршка у деловима SELECT и WHERE упитника.
- SQL функције су имплементиране кроз високо оптимизовану класу библиотеке и морају бити брзе колико год могу.
- Потпуну подршку за SQL GROUP BY и ORDER BY клаузуле.
- Подршка за LEFT OUTER JOIN и RIGHT OUTER JOIN са ANSI SQL и ODBC синтаксама.
- Могућност мешања табела из различитих база података у истом упитнику.

- Привилегије и *password*-ом заштићен систем, јер је *password* осигуран и тек када се конектујете на сервер он врши енкрипцију.
- ODBC (*Open-DataBase-Connectivity*) подржава Win32.
- Веома брзе *B-tree* диск табеле са индексом компресије.
- До 32 индекса по табели је дозвољено. Сваки индекс може да се састоји од 1 до 16 колона или делова колона. Максимална дужина индекса је 500 Бајтова.
- Записи фиксиране и промењиве дужине.
- Рукује са већим базама података.
- Све колоне имају дефаултне вредности.
- Користи GNU Automake, Autoconf и libtool за портабилност.
- Пише у C и C++. А тестиран је са широким спектром различитих компајлера.
- Веома брз систем за проналажење локације.
- Нема цурења меморије. Тестирано са комерцијалним детектором за цурење меморије.
- Користи *myisamchk* веома брз програм за проверавање табела, оптиматизацију и поправке.
- Потпуна подршка за неколико различитих карактера.
- Сви подаци су сачувани у датом карактеру.
- Сва сортирања се врше према датом карактеру који се употребљава. Могуће је мењање карактера тек када се стартује MySQL сервер.
- Alias у табелама и рововима су дозвољени по SQL92 стандарду.
- DELETE, INSERT, REPLACE и UPDATE враћају колико год је потребно колона уназад.
- Имена функција не поклапају се са именима колона и табела.
- У свим MySQL програми могуће је дозвати помоћ и остале асистент програме, такође је могућа и онлајн помоћ.
- Сервер може пружити својим клијентима упозорења о грешкама на мноштву језика.
- Клијенти могу да се конектују на MySQL сервер користећи TCP/IP сокете, UNIX сокете или NT.

- MySQL специјална команда SHOW може се користити за повратак информација о базама података, табелама и индексима. EXPLAIN команда се може користити за детерминисање како оптимизер решава упитник.[3]



Слика 1. Лого MySQL-а.[1]

5. Структура језика

Овде ће бити поменута правила за писање следећих елемената SQL исказа приликом коришћења MySQL-а:[4]

- Литерарне вредности, као што су стрингови и бројеви
- Идентификатори, као што су називи база, табела и колона
- Коментари
- Резервисане речи

5.1. Литерарне вредности

У литерарне вредности спадају стрингови, бројеви, хексадецималне вредности, боолеан вредности и NULL.[4]

5.2. Стрингови

Стринг је низ карактера ограђен било једноструким наводницима ('), било двоструким наводницима ("). Примери:[2]

'стринг'

"јос један стринг"

У оквиру стринга одређене секвенце имају специјално значење. Свака од ових секвенци почиње backslash карактером (\) који је познат као escape карактер. MySQL препознаје следеће escape секвенце:[2]

\'	Једноструки наводник(')
\"	Двоструки наводник (")
\b	Backspace карактер
\n	Прелазак у нову линију
\t	Tab карактер
\\	Backslash карактер (\)

За све остале есцапе секвенце бацксласх карактер се игнорише. На пример, \ц се интерпретира као “х”.

Ове секвенце су осетљиве на велика и мала слова. На пример, \б се интерпретира као бацкспаце, док се “\В” третира као “В”.

Постоји више начина укључивања наводника унутар стринга:

- Једноструки наводник унутар стринга који је ограничен једноструким наводницима може бити написан као "
- Двоструки наводник унутар стринга који је ограничен двоструким наводницима може бити написан као ""
- Коришћењем есцапе секвенци
- Једноструки наводник унутар стринга који је ограничен двоструким наводницима не захтева посебан третман (не захтева дуплирање или писање есцапе секвенце). Исто тако двоструки наводник унутар стринга који је ограничен једноструким наводницима не захтева посебан третман.[5]

5.3.Бројеви

Цели бројеви се представљају низом цифара. Реални бројеви користе тачку (.) као децимални сепаратор. Било који тип бројева може имати знак + или знак - на почетку ради означавања позитивних или негативних вредности респективно.[4]

Примери исправно написаних целих бројева: 1232

0

-31

Примери исправно написаних реалних бројева:

128.48

-35.785e+10 120.00

Цели бројеви се могу користити у израчунавањима са реалним бројевима и тада се третирају као еквивалентни реални бројеви.[4]

5.4. Боолеан вредности

Константама TRUE и FALSE одговарају вредности 1 и 0 респективно. Називи константи се могу писати и великим и малим словима.[3]

```
mysql> SELECT TRUE, true, FALSE, false;
```

-> 1, 1, 0, 0

5.5. NULL вредности

NULL вредност значи “нема података”. NULL се може писати и великим и малим словима. Треба имати на уму да је NULL вредност различита од вредности 0 за нумеричке типове података или празног стринга за стринг типове података.[3]

5.6. Називи база, табела, индекса, колона и алиас-и

Називи база, табела, индекса, колона и алиас-и су идентификатори.

Следећа табела дефинише максималне дужине за сваки тип идентификатора.[2]

Идентификатор	Максимална дужина
База података	64
Табела	64
Колона	64
Индекс	64
Алиас	255

Постоје одређена ограничења по питању карактера који се могу појављивати у идентификаторима:

- Дозвољено је користити наводнике у идентификаторима, али је најбоље то избегавати
- Називи база, табела и колона неби требало да се завршавају спаце карактером
- Називи база не могу садржати карактере / \ . или карактере који нису дозвољени у Windows-у приликом давања назива фолдерима
- Називи табела не могу садржати карактере / \ . или карактере који нису дозвољени у Windows-у приликом давања назива фајловима

Идентификатор може бити са наводницима или без наводника. Ако је идентификатор резервисана реч или садржи специјалне карактере корисник мора да пише идентификатор са наводницима сваки пут када га користи. Знак навода за идентификатор је знак backtick (`). (Изузетак: реч која иде након тачке у квалификатору мора бити идентификатор, па не мора бити под наводницима ако је резервисана реч). Специјални карактери су они који се не налазе у скупу алфанумеричких карактера тренутног сета карактера, _ и \$.

```
mysql> SELECT * FROM `select` WHERE `select`.id > 100;[4]
```

5.7. Квалификатори идентификатора

MySQL дозвољава називе који се састоје од једног идентификатора или од више идентификатора. Ако назив има више идентификатора неопходно је да они буду раздвојени тачком.[4]

У MySQL колона се може референцирати на један од следећих начина:

Референца колоне	Значење
<i>col_name</i>	Колона <i>col_name</i> из било које табеле коришћене у исказу која има колону са тим називом
<i>tbl_name.col_name</i>	Колона <i>col_name</i> из табеле <i>tbl_name</i> подразумеване базе података
<i>db_name.tbl_name.col_name</i>	Колона <i>col_name</i> из табеле <i>tbl_name</i> базе података <i>db_name</i>

5.8. Осетљивост идентификатора на велика и мала слова

У MySQL-у базама података на серверу одговарају фолдери у оквиру дата фолдера (C:\Program Files\MySQL\MySQL Server 5.0\data). Свакој табели базе одговара барем један фајл у фолдеру базе. Према томе осетљивост на велика и мала слова код оперативног система одређује осетљивост на велика и мала слова код назива база и табела. Ово значи да називи база и табела нису осетљиви на велика и мала слова у Њиндоњс-у.[4]

Иако називи база и табела нису осетљиви на велика и мала слова неби требало на различитим местима у оквиру истог исказа у називу базе или табеле користити различита слова (велика или мала).[4]

```
mysql> SELECT * FROM moja_tabela WHERE MOJA_TABELA.kol=1;
```

Називи колоне и индекса и алиаси колоне нису осетљиви на велика и мала слова. Називи окидача су осетљиви на велика и мала слова.[3]

Алиаси табела нису осетљиви на велика и мала слова.

Најбоље је усвојити доследну конвенцију, као што је да се приликом креирања и референцирања база и табела користе називи са малим словима. [3]

5.9. Синтакса коментара

MySQL сервер подржава три стила за коментаре:

- Од карактера # до краја линије[2]
- Од секвенце -- до краја линије. Неопходно је да након друге црте постоји барем један размак или контролни карактер (space, tab, нова линија, итд.).[2]
- Од секвенце /* до следеће /* секвенце. Ова синтакса дозвољава коментар у више редова, пошто почетна и крајња секвенца не морају да буду у истом реду.[2]

Следи пример који демонстрира сва три стила:

```
mysql> SELECT 1+1; # Овај коментар се наставља до краја линије
```

```
mysql> SELECT 1+1; -- Овај коментар се наставља до краја линије
```

```
mysql> SELECT 1 /* ово је коментар у линији */ + 1;
```

```
mysql> SELECT 1+
```

```
/*
```

```
ово је
```

```
коментар у више линија
```

```
*/
```

```
1;
```

5.10. Третман резервисаних речи у MySQL-у

У резервисаним речима језика SQL не прави се разлика између малих и великих слова. То је стандардно у свим системима за рад са базама података.[1]

Дозвољено је да се називи функција користе као идентификатори. На пример, ABS је прихватљиво као назив колоне. Међутим, треба знати да приликом позивања функција није дозвољено постојање размака између назива функције и отворене заграде која следи.[1]

```
mysql> CREATE TABLE abs (val INT);
```

```
mysql> CREATE TABLE abs(val INT);
```

У другом случају ће се јавити грешка у синтакси пошто исказ онда покушава да позове функцију ABS().

Већина резервисаних речи су забрањене у стандардном SQL-у за називе колоне и табела.[1]

6. Конвенције које ће бити коришћене у овом материјалу

Резервисане речи у SQL-у нису осетљиве на велика и мала слова. У овом материјалу ће резервисане речи бити писане великим словима.[5]

У опису синтакси угласте заграде ([]) указују на опционе речи или клаузуле. [5]

`DROP TABLE [IF EXISTS] tbl_name;`

Када елемент синтаксе садржи већи број алтернатива, алтернативе су раздвојене вертикалним линијама (|). Када једна од алтернатива може бити изабрана, алтернативе су смештене унутар угластих заграда. Када једна од алтернатива мора бити изабрана, алтернативе су смештене унутар витичастих заграда ({}). [5]

(...) указује на изостављање дела исказа, са намером да се проследи краћа верзија комплексније синтаксе. (...) такође указује да се претходни елемент синтаксе исказа може понављати. [5]

6.1. Data Definition Language (DDL)

Језик за дефинисање података (Data Definition Language (DDL)) омогућава формирање структуре базе података. [3]

Прављење базе података – CREATE DATABASE синтакса

Након пројектовања структуре базе података, наредни корак је, сасвим логично, издавање команде MySQL-у за прављење нове базе података. То се ради помоћу SQL-ове команде CREATE DATABASE, на следећи начин: [3]

`CREATE DATABASE [IF NOT EXISTS] db_name;`

CREATE DATABASE креира базу података са датим називом. Ако већ постоји база података са датим називом, а није употребљено IF NOT EXISTS јавиће се грешка. [3]

Креирање базе података пословање: [3]

`CREATE DATABASE пословање;`

Ако корисник жели да провери да ли је ова команда успешно извршена може да изда команду:

`SHOW DATABASES;`

Требало би да се назив нове базе података појави у списку база података на серверу. Називе база података је најбоље писати малим словима. У називима база не треба користити српске карактере. У називима база не треба користити размаке. Ако се жели ефекат размака треба користити доњу црту (_). [3]

Сада на серверу постоји празна база података.

6.2. Бирање базе података – USE синтакса

USE db_name;

USE db_name исказ каже MySQL-у да користи базу са датим називом као подразумевану (текућу) базу за исказе који следе. База са датим називом остаје подразумевана до краја сесије или док се не проследи неки други USE исказ.[4]

Проглашавање неке базе подразумеваном (текућом) помоћу USE исказа не спречава корисника да приступа табелама из осталих база података на серверу.[4]

6.3. Прављење табеле – CREATE TABLE синтакса

DROP DATABASE [IF EXISTS] db_name;

DROP DATABASE briše sve tabele u bazi podataka sa datim nazivom i nakon toga briše i samu bazu podataka. Treba biti jako oprezan sa ovim iskazom!

IF EXISTS se koristi da bi se sprečila pojava greške ako na serveru ne postoji baza podataka sa datim nazivom.

DROP DATABASE vraća broj tabela koje su bile obrisane.[3]

6.4. Pravljenje tabele - CREATE TABLE sintaksa

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name (create_definition,...)

[table_option ...]

или:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name

[(create_definition,...)] [table_option ...] select_statement

или:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name

{ LIKE old_tbl_name | (LIKE old_tbl_name) }

create_definition:

column_definition

[ON UPDATE reference_option]

reference_option:

RESTRICT | CASCADE | SET NULL | NO ACTION

table_option:

{ENGINE|TYPE} [=] engine_name

| AUTO_INCREMENT [=] value

| AVG_ROW_LENGTH [=] value

| [DEFAULT] CHARACTER SET charset_name

| CHECKSUM [=] {0 | 1}

| COLLATE collation_name

| COMMENT [=] 'string'

| CONNECTION [=] 'connect_string'

| DATA DIRECTORY [=] 'absolute path to directory'

| DELAY_KEY_WRITE [=] {0 | 1}

| INDEX DIRECTORY [=] 'absolute path to directory'

| INSERT_METHOD [=] { NO | FIRST | LAST }

| MAX_ROWS [=] value

| MIN_ROWS [=] value

| PACK_KEYS [=] {0 | 1 | DEFAULT}

| PASSWORD [=] 'string'

| ROW_FORMAT [=]

{DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}

| UNION [=] (tbl_name[,tbl_name]...)

select_statement:

[IGNORE | REPLACE] [AS] SELECT ...

| [CONSTRAINT [*symbol*]] PRIMARY KEY [*index_type*] (*index_col_name*,...)

| {INDEX|KEY} [*index_name*] [*index_type*] (*index_col_name*,...)

| [CONSTRAINT [*symbol*]] UNIQUE [INDEX|KEY]

[*index_name*] [*index_type*] (*index_col_name*,...)

| {FULLTEXT|SPATIAL} [INDEX|KEY] [*index_name*] (*index_col_name*,...)

| [CONSTRAINT [*symbol*]] FOREIGN KEY

[*index_name*] (*index_col_name*,...) [*reference_definition*]

| CHECK (*expr*)

column_definition:

col_name data_type [NOT NULL | NULL] [DEFAULT *default_value*] [AUTO_INCREMENT]
[UNIQUE [KEY] | [PRIMARY] KEY]

[COMMENT '*string*'] [*reference_definition*]

index_col_name:

col_name [(*length*)] [ASC | DESC]

index_type:

USING {BTREE | HASH}

reference_definition:

REFERENCES *tbl_name* [(*index_col_name*,...)] [MATCH FULL | MATCH PARTIAL |
MATCH SIMPLE]

[ON DELETE *reference_option*][1]

CREATE TABLE креира табелу са датим називом. Табела се креира у подразумеваној (текућој) бази. Грешка се јавља ако табела постоји, ако не постоји подразумевана (текућа) база или ако база не постоји.[1]

Назив табеле може бити специфициран као *дб_наме.тбл_наме* да би се креирала табела у одређеној бази података. Ово функционише независно од тога да ли постоји подразумевана (текућа) база, под условом да база са датим називом постоји.[1]

Приликом креирања табеле се може користити TEMPORARY кључна реч. Табела креирана са овом кључном речи је видљива само у текућој конекцији (сесији) и бива аутоматски обрисана након прекида конекције.[1]

Кључна реч IF NOT EXIST спречава појављивање грешке ако табела са истим називом постоји.[1]

Називе табела је најбоље писати малим словима. У називима табела не треба користити српске карактере. У називима табела не треба користити размаке. Ако се жели ефекат размака треба користити доњу црту (_).[1]

Помоћу одредбе *db_name.tbl_name* може се направити нова табела која има исту шему као нека друга табела.[1]

У команди CREATE TABLE декларишу се унутар заграда потребне колоне, њихови типови података и друге информације које се тичу структуре табеле. Најједноставнија дефиниција колоне састоји се само од назива колоне и типа података у колони.[1]

Команда CREATE TABLE се може завршити командом SELECT. SELECT је SQL-ова команда која омогућава учитавање редова из једне или више табела. Помоћу ове опције се може напунити нова табела подацима које учитава задата команда SELECT.[1]

Типови података

data_type представља тип података приликом дефинисања колона табеле.

MySQL подржава велики број типова података који се могу разврстати у више категорија: нумерички типови, типови за датум и време и знаковни или текстуални типови.[3]

Нумерички типови се користе за складиштење бројева. BIT[(M)]

Служи за дефинисање бинарног типа података. М дефинише број битова и може бити од 1 до 64. Уколико се М изостави подразумева се да је 1.[3]

TINYINT[(M)] [UNSIGNED] [ZEROFILL]

Веома мали цели бројеви. SIGNED опсег је од -128 до 127. UNSIGNED опсег је од 0 до 255.

М представља ширину приказивања за целобројне типове података. Ширина приказивања се користи за приказивање целобројних вредности чија је ширина мања од специфициране ширине за колону додавањем спаце карактера (размака) са леве стране. Ширина приказивања не ограничава опсег вредности који се може приказати у колони. Када се користи заједно са опционим атрибуту ZEROFILL врши се додавање нула (водеће нуле) а не спаце карактера са леве стране. На пример, ако је колона декларисана са INT(5) ZEROFILL, вредност 4 се представља са 00004.[3]

Сви целобројни типови података могу имати опциони атрибут UNSIGNED. Када се наведе овај атрибут могуће је уносити само ненегативне бројеве, а горња граница дозвољеног опсега бива повећана, при чему величина опсега остаје иста.[3]

Ако се за колону са нумеричким типом података наведе атрибут ZEROFILL, MySQL аутоматски додаје UNSIGNED атрибут.[3]

BOOL,BOOLEAN

Ови типови су синоними за TINYINT (1). false (нетачно) одговара вредности 0. true (тачно) одговара било којој вредности различитој од нуле.[3]

SMALLINT[(M)] [UNSIGNED] [ZEROFILL]

Мали цели бројеви. SIGNED опсег је од -32768 до 32767. UNSIGNED опсег је од 0 до 65535.[3]

MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]

Средње велики цели бројеви. SIGNED опсег је од -8388608 до 8388607. UNSIGNED опсег је од 0 до 16777215.[3]

INT[(M)] [UNSIGNED] [ZEROFILL]

Нормални цели бројеви. SIGNED опсег је од -2147483648 до 2147483647. UNSIGNED опсег је од 0 до 4294967295.[3]

INTEGER[(M)] [UNSIGNED] [ZEROFILL]

Овај тип је синоним за INT.[3]

BIGINT[(M)] [UNSIGNED] [ZEROFILL]

Велики цели бројеви. SIGNED опсег је од -9223372036854775808 до 9223372036854775807. UNSIGNED опсег је од 0 до 18446744073709551615.[3]

Апроксимирани нумерички типови су FLOAT и DOUBLE.[3]

FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]

Мали (једнострука прецизност) број са покретним зарезом. Дозвољене вредности су од -3.402823466E+38 до -1.175494351E-38, 0 и од 1.175494351E-38 до 3.402823466E+38.[2]

М је укупан број цифара, а Д је број цифара иза децималне тачке. Ако се М и Д изоставе вредности се смештају у складу са хардверским ограничењима. Број са покретним зарезом једноструке прецизности је тачан до приближно 7 децималних места.[2]

Коришћење овог типа података може довести до неочекиваних проблема пошто се сва израчунавања у MySQL-у раде са двоструком прецизношћу.[2]

DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]

Нормалан (двострука прецизност) број са покретним зарезом. Дозвољене вредности су од 1.7976931348623157E+308 до -2.2250738585072014E-308, 0 и од

2.2250738585072014E-308 до 1.7976931348623157E+308.[1]

М је укупан број цифара, а Д је број цифара иза децималне тачке. Ако се М и Д изоставе вредности се смештају у складу са хардверским ограничењима. Број са покретним зарезом двоструке прецизности је тачан до приближно 15 децималних места.[1]

DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL], REAL[(M,D)] [UNSIGNED] [ZEROFILL] [1]

Ови типови су синонимима за DOUBLE.[1]

Тачни нумерички типови су тип INTEGER и његове варијанте и DECIMAL. DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL].[1]

Број са фиксним зарезом. М је укупан број цифара (прецизност), а Д је број цифара иза децималне тачке (the scale). Децимална тачка и (за негативне бројеве) знак – се не рачунају у М. Ако је Д једнако нули бројеви немају децималну тачку и разломљени део. Максималан укупан број цифара (М) за овај тип података је 65. Максималан број цифара иза децималне тачке (Д) за овај тип података је 30. Ако се Д изостави, подразумевана вредност је 0. Ако се М изостави, подразумевана вредност је 10.[1]

Сва основна израчунавања (+, -, *, /) са овим типом података се раде са прецизношћу од 65 цифара.

Овај тип података се обично користи за рад са новчаним вредностима.[1]

DEC[(M[,D])] [UNSIGNED] [ZEROFILL], NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL], FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]

Ови типови су синоними за DECIMAL.[1]

DATE

Датум. Подржани опсег је од '1000-01-01' до '9999-12-31'. MySQL приказује вредности за овај тип података у формату 'YYYY-MM-DD'. [1]

DATETIME

Комбинација датума и времена. Подржани опсег је од '1000-01-01 00:00:00' до '9999-12- 31 23:59:59'. MySQL приказује вредности за овај тип података у формату 'YYYY-MM-DD HH:MM:SS'. [1]

TIMESTAMP

Ово је користан тип података за колоне табеле. Ако се у одређеном реду не зада вредност за колону овог типа у колону се уписује време када је ред додат табели или када је последњи пут измењен садржај реда. [2]

Опсег је од '1970-01-01 00:00:01' UTC до 2037. године. TIMESTAMP вредности су смештене као број секунди протеклих од епохе ('1970-01-01 00:00:00' UTC). TIMESTAMP вредност се враћа као стринг са форматом 'YYYY-MM-DD HH:MM:SS' и са ширином приказивања фиксираном на 19 карактера.[2]

TIME

Време. Подржани опсег је од '-838:59:59' до '838:59:59'. MySQL приказује вредности за овај тип података у формату 'HH:MM:SS'.[2]

YEAR[(2|4)]

Година као двоцифрени или четвороцифрени број. Подразумевани формат је четвороцифрени број. Код четвороцифреног формата подржани опсег је од 1901 до 2155 и 0000. Код двоцифреног формата подржани опсег је од 70 до 69, што одговара годинама од 1970. до 2069. MySQL приказује вредности за овај тип података у формату YYYY.[2]

[NATIONAL] CHAR(M) [CHARACTER SET charset_name] [COLLATE collation_name]

Стринг фиксне дужине коме приликом смештања увек бивају додати спаце карактери све до специфициране дужине. М представља дужину, односно број знакова. Опсег за М је од 0 до 255 карактера.[4]

Када се податак типа CHAR упише у колону табеле, он увек има дужину која је задата у дефиницији колоне. То се постиже допуњавањем податка у колони размацама. Ти размаци се аутоматски уклањају при учитавању податка из колоне типа CHAR.[4]

Очигледно је да подаци типа CHAR заузимају више простора на диску од еквивалентних знаковних вредности променљиве дужине. Предност им је што се подаци брже учитавају из табеле чије су све колоне фиксне дужине (рецимо CHAR или DATE).[4]

Декларацијама типова CHAR и VARCHAR може следити резервисана реч BINARY, што значи да се при поређењу знаковних вредности прави разлика између малих и великих слова. Подразумевани начин поређења је да се та разлика не прави.[4]

CHAR је скраћеница за CHARACTER. NATIONALCHAR (или одговарајућа скраћеница NCHAR) је стандардни начин да се у SQL-у дефинише да колона са типом података користи неки унапред дефинисани скуп карактера.[4]

CHARACTER SET атрибут специфицира скуп карактера који се користи. COLLATE атрибут специфицира упаривање за тај скуп карактера. Скуп карактера је скуп симбола и шифрирања. Упаривање је скуп правила за поређење карактера у скупу карактера.[4]

CHARSET је синоним за CHARACTER SET.[4]

CHAR [CHARACTER SET charset_name] [COLLATE collation_name] Ovo je sinonim za CHAR(1).[4]

[NATIONAL] VARCHAR(M) [CHARACTER SET charset_name] [COLLATE collation_name] Стринг варијабилне дужине. М представља максималну дужину података. Опсег за М је од 0 до 65535 карактера. VARCHAR је синоним за CHARACTER VARYING.[4]

BINARY(M)

BINARYTYP тип података је сличан CHAR типу података, али смешта бинарне стрингове уместо небинарних стрингова са карактерима.[4]

VARBINARY(M)

VARBINARY тип података је сличан VARCHAR типу података, али смешта бинарне стрингове уместо небинарних стрингова са карактерима.[4]

TINYBLOB

BLOB колона са максималном дужином од 255 бајтова. BLOB је велики бинарни објекат који може садржати променљиву количину података. BLOB колоне се третирају као бинарни стрингови.[4]

TINYTEXT [CHARACTER SET charset_name] [COLLATE collation_name] TEXT колона са максималном дужином од 255 карактера.[4]

BLOB[(M)]

BLOB је скраћеница од Binary Large Object (велики бинарни објекат). BLOB колона може да има максималну дужину од 65535 бајта.[4]

За овај тип се може дати опциона дужина М. Ако се дефинише MySQL креира колону која је најмањи BLOB тип довољно велики да прима вредности које су М бајтова дугачке.[4]

TEXT[(M)] [CHARACTER SET charset_name] [COLLATE collation_name] TEXT колона са максималном дужином од 65535 карактера.[4]

За овај тип се може дати опциона дужина М. Ако се дефинише MySQL креира колону која је најмањи TEXT тип довољно велики да прима вредности које су М карактера дугачке.[4]

Омогућава складиштење текстуалних података дужих од оног што може да стане у типове CHAR и VARCHAR.[4]

MEDIUMBLOB

BLOB колона са максималном дужином од 16777215 бајтова (16 MB).

MEDIUMTEXT [CHARACTER SET charset_name] [COLLATE collation_name] TEXT колона са максималном дужином од 16777215 карактера (16 MB).

LOB

BLOB колона са максималном дужином од 4294967295 бајтова (4 GB).

LONGTEXT [CHARACTER SET charset_name] [COLLATE collation_name] TEXT колона са максималном дужином од 4294967295 карактера (4GB).

ENUM('value1','value2',...) [CHARACTER SET charset_name] [COLLATE collation_name]

Набрајање. Овај тип података омогућава задавање листе могућих вредности. Колона табеле може садржати само једну вредност, изабрану из листе вредности 'value1','value2',..., NULL или специјална ' ' грешка вредност. ENUM колона може имати максимално 65535 различитих вредности. ENUM вредности су интерно представљене као цели бројеви.[4]

пол ENUM ('м', 'ж')

SET('value1','value2',...) [CHARACTER SET charset_name] [COLLATE collation_name]

Скуп. Колона табеле може садржати једну или више вредности, а свака од њих мора бити изабрана из листе вредности 'value1','value2',... SET колона може имати максимално 64 члана. SET вредности су интерно представљене као цели бројеви.[4]

6.4.2. Остали атрибути

- Свака колона се може декларисати као NOT NULL или NULL, што значи да се не дозвољава да колона садржи вредност NULL (опција NOT NULL) или да се прихвата вредност NULL (опција NULL). Ако нису специфицирани ни NULL ни NOT NULL атрибут, колона се третира као да је NULL атрибут специфициран.[4]
- Колоне са целобројним типом података могу имати додатни атрибут AUTO_INCREMENT. Када се у индексирану AUTO_INCREMENT колону унесе вредност NULL (препоручено) или вредност 0, вредност у колони се подешава на следећу вредност у низу. Типично то је валуе+1 где је валуе највећа вредност у колони тренутно у табели. AUTO_INCREMENT низови почињу бројем 1. Може постојати само једна AUTO_INCREMENT колона по табели, мора бити индексирана и не може имати подразумевану (DEFAULT) вредност. AUTO_INCREMENT функционише исправно ако садржи само позитивне вредности.[4]
- DEFAULT клаузула специфицира подразумевану вредност за колону. Са једним изузетком, подразумевана вредност мора бити константа; не може бити функција или израз. Изузетак је да се може специфицирати CURRENT_TIMESTAMP као подразумевана вредност за TIMESTAMP колону. Колоне типа BLOB и TEXT не могу имати дефинисану подразумевану вредност.[4]
- Коментар за колону може бити специфициран помоћу COMMENT опције и може имати дужину до 255 карактера[4]

6.4.3. Ограничења (Constraints)

Ако се редовно сортирају подаци у табели по једној истој колони или колонама потребно је направити индекс за ту колону или те колоне. Индекс је интерна табела вредности која одржава ред, односно редослед записа. На овај начин, када је потребно да се сортирају подаци или када је потребно да се одређени податак пронађе брзо MySQL може да врши претраживање по индекс кључевима по познатом редоследу, што је боље од секвенцијалног претраживања по подацима.

Прављење индекса успорава уношење података; сваки нови запис, обрисани запис или промена вредности у индексираном пољу захтева промену индекса. Индексирани колоне треба користити само онда када су стварно неопходне.[4]

KEY (кључ) је нормално синоним за INDEX (индекс) и значи да ће задата колона (или колоне) бити индексирана. Индекси се користе да би се брзо пронашли редови са одређеном вредношћу у колони. Атрибут PRIMARY KEY се може специфицирати и као KEY када је дат у дефиницији колоне.[4]

UNIQUE (јединствени) индекс креира ограничење такво да све вредности у индексу (односно колони која је индексирана) морају бити различите. Уколико корисник покуша да дода нови ред у табелу који има исту вредност у колони за коју је дефинисан UNIQUE индекс као неки други ред јавиће се грешка. Ово ограничење се не односи на NULL вредности сем за БДБ машине за смештање података. За остале машине за смештање података UNIQUE индекс дозвољава вишеструке NULL вредности за колоне које могу садржати NULL вредности.[4]

PRIMARY KEY (примарни кључ) је јединствени индекс где све колоне кључа морају бити дефинисане као NOT NULL. Ако нису експлицитно декларисане као NOT NULL, MySQL их декларише имплицитно. Табела може имати само један примарни кључ. На овај начин се поставља ограничење над табелом. То значи да се уписом новог записа у колону која је примарни кључ не сме уписати вредност која већ постоји. За колону (или колоне) која је примарни кључ се аутоматски формира индекс.[4]

У креираној табели примарни кључ је смештен као први, затим долазе сви UNIQUE (јединствени) индекси и на крају сви нејединствени индекси.

Примани кључ може бити индекс над више колона. Међутим, није могуће креирати индекс над више колона коришћењем атрибута PRIMARY KEY у спецификацији колоне. Мора се користити посебна PRIMARY KEY(index_col_name, ... клаузула.[4]

У MySQL-у назив примарног кључа је PRIMARY. За остале индексе, ако им се не додели назив, индексу се даје назив прве индексиране колоне, заједно са опционим суфиксом (_2, _3, ...) да би био јединствен.[4]

За машине за складиштење које нису типа ИнноДБ је могуће приликом дефинисања колоне користити REFERENCES tbl_name(col_name) која нема стварни ефекат и служи само као подсетник или коментар кориснику да се намерава да колона која се тренутно дефинише указује на колону неке друге табеле.[4]

Неке машине за смештање података дозвољавају дефинисање типа индекса приликом креирања индекса.[4]

Спецификација за индекс_col_name се може завршавати са ASC или DESC. Ове кључне речи су дозвољене за будуће екстензије за специфицирање смештања вредности у индексу

по растућем или опадајућем редоследу. Тренутно се ове кључне речи игноришу; вредности у индексу су увек смештене по растућем редоследу.[4]

Могуће је креирати специјалне FULLTEXT индексе који се користе за Full-text (текстуална) претраживања. Само МуИСАМ машина за смештање података подржава FULLTEXT индексе. Они могу бити креирани само за CHAR, VARCHAR и TEXT колоне.[4]

Могуће је креирати SPATIAL индексе за просторне типове података (дозвољавају генерисање, смештање и анализу географских обележја). Ови типови података су подржани само у МуИСАМ табелама и индексирани колоне морају бити декларисане као NOT NULL. InnoDB табеле подржавају ограничења типа страни кључ (FOREIGN KEY). Страни кључ је скуп колона (једна или више колона) из једне табеле (секундарна табела или дете табела) чије вредности морају да се 'слажу', односно да одговарају вредностима примарног кључа у некој другој табели (примарна табела или родитељ табела). Дефинисањем страног кључа се у ствари успоставља релација између табела. Синтакса за дефинисање страног кључа код овог типа табела изгледа овако:[4]

```
[CONSTRAINT symbol] FOREIGN KEY [id] (index_col_name, ...) REFERENCES tbl_name (index_col_name, ...)
```

```
[ON DELETE {RESTRICT | CASCADE | SET NULL | NO ACTION}] [ON UPDATE {RESTRICT | CASCADE | SET NULL | NO ACTION}][4]
```

Приликом дефинисања страног кључа обе табеле морају бити ИнноДБ табеле и не смеју бити привремене (TEMPORARY) табеле.

InnoDB одбија било коју INSERT или UPDATE операцију која покушава да креира вредност за страни кључ у дете табели ако не постоји одговарајућа вредност кључа у родитељ табели. Акција коју предузима InnoDB за било коју UPDATE или DELETE операцију која покушава да промени или обрише вредност кључа у родитељ табели који има повезане редове у дете табели зависи од референцијалне акције специфициране коришћењем ON UPDATE и ON DELETE подклаузула у FOREIGN KEY клаузули. Када корисник покуша да обрише или промени ред у родитељ табели, а постоје један или више повезаних редова у дете табели, InnoDB подржава 5 опција везано за акцију која ће бити извршена:[4]

- CASCADE: Брисањем или изменом реда у родитељ табели се аутоматски бришу или мењају повезани редови у дете табели.

- **SET NULL:** Брисањем или изменом реда у родитељ табели се вредност у колони која је страни кључ дете табеле за повезане редове поставља на NULL. Ово важи само ако колоне које су страни кључ немају специфициран NOT NULL квалификатор.
- **NO ACTION:** У стандардном SQL-у NO ACTION значи да нема акције у смислу да неће бити могуће обрисати или променити вредност примарног кључа у родитељ табели ако постоји повезана вредност страног кључа у дете табели. InnoDB одбија операцију брисања или измене за родитељ табелу.
- **RESTRICT:** Одбија се операција брисања или измене за родитељ табелу.
- **SET DEFAULT:** Ова акција је препозната од стране парсера, али InnoDB одбија дефиниције табела које садрже ON DELETE SET DEFAULT или ON UPDATE SET DEFAULT клаузуле.

Одговарајуће колоне које чине страни и примарни кључ морају имати сличне типове података да би могле да буду поређене без конверзије података. Величина и знак целобројних типова морају бити исти. Дужине стринг типова не морају бити исте. [4]

7. Зашто MySQL уместо неке друге базе података ?

Превише база података је присутно на тржишту да бисмо их упоређивали у детаље али оно што издваја MySQL базу података од других база података је то што је јефтинија и има много боље перформансе што ће други произвођачи тешко надмашити.[1]

7.1.Перформансе:

MySQL је веома брз програм. Oracle, Microsoft и IBM тврде да продају најбрже базе података на свету, што не мора да значи јер данашњи *benchmark* програми и резултати могу доказати и потврдити квалитете које произвођач и спонзор гарантују иако се у пракси не показују тако. На сајту MySQL можете да видите поређења MySQL базе података са другим тренутно актуелним базама и можете видети да је MySQL далеко надмашује остале базе у свим погледима.[2]

benchmark резултати се могу наћи на:

www.mysql.com/information/benchmarks.html

Брзина је одувек била главна одлика MySQL базе података. Нови додаци се стављају у MySQL једино кад се увере да то неће нашкодити перформансама база података. Ово значи да се новине и додаци ретко и споро стављају у MySQL прилично спорије него што би корисници желели, али све је то за добробит корисника и MySQL, једноставно жеља да програм буде брз и без грешака.[2]

7.2.Цена:

Цена је нешто што је најлакше поредити. За многе сврхе MySQL је бесплатна апликација. GPL омогућава коришћење софтвера, пре него оригиналног кода, и дистрибуцију софтвера другим корисницима али под условом да буду привржени GPL. Под неким околностима ако желите да редистрибуирате MySQL као део комерцијалног производа

потребна вам је комерцијална дозвола. Главно правило MySQL центра за дистрибуцију је " *if you are free, so are we; if you are commercial, so are we.*"[2]

7.3.Стабилност:

Људи који праве MySQL одувек су се хвалили да је стабилност њихових производа главна карактеристика и приоритет. Све верзије MySQL се праве у бинарној фази и као такве се пуштају у рад са тестовима и MySQL тест базе. Ови тестови функционалности омогућавају проналажење грешака а грешкама које су већ пронашли не дозвољавају да се понове. Ствараоци MySQL посебну пажњу морају да обрате на проналажење и исправљање грешака јер сваки њихов даљни рад би био онемогућен ако претходне грешке нису исправљене.[1]

7.4.Лакоћа коришћења:

Никаква компликована подешавања и процедуре нису потребне за покретање MySQL. MySQL ради како се каже право из кутије. Почетна подешавања су намештена на минимално коришћење хард диска и меморије. Ако желимо напреднија подешавања мораћемо да искључимо почетна подешавања. Да би нам још олакшали у MySQL пакет су стављени и примери неких функција и радова.[1]

7.5.Карактеристике:

Поређење карактеристика зависи од тога која карактеристика нама највише одговара. MySQL има неке карактеристике као што су потпуно претраживање текста, понављање и подршка за масивне табеле, све ове опције или недостају или су избачене из нискобуџетних понуђених програма. Међутим MySQL недостају сачуване процедуре и врсте погледа који су стандардни у високобуџетним решењима а јављају се чак и у неким јефтинијим верзијама. Свесни овога ствараоци MySQL ће у наредним решењима овог програма убацити дата решења. Неке од карактеристика које поседује MySQL , на пример могућност закључавања редова, немају ни најскупљи програми који се налазе на тржишту.[1]

8.Предности и мане MySQL-а

Предности:

- Брзина
- Поузданост
- Мала захтевност за системским ресурсима
- Флексибилно побољшавање перформанси
- Рад на различитим платформама
- Подржава велику количину програмских језика
- ODBC
- Бесплатно или повољно лиценцирање
- Повољна комерцијална подршка
- Јака подршка корисника
- Доступност изворног кода

Мане:

- Недостатак неких могућности SQL-а
- Недостатак озбиљнијих тестирања на одређеним платформама
- Компликовано руковање изворним кодом

8.1.Предности MySQL-а:

8.1.1Брзина:

Срце MySQL кода је писано у потпуности од нуле, а основни циљ је био постизање што бољих перформанси. Оно што је уствари и довело до настанка MySQL-а је чињеница да је Monty Widenius био толико фрустриран спорошћу других програма на тржишту да је одлучио да напише MySQL. Велики број нових корисника је било скептично и није веровало да је MySQL толико бржи од конкуренције, чак су доводили у питање да ли је MySQL способан да задовољи и њихове основне потребе. Људи из MySQL-а су предложили да се направи *benchmark* програм који ће истестирати могућности MySQL-а и показати шта све он уствари може. Резултати тестова су били толико невероватни да су се нови корисници без поговора одлучивали за MySQL. Велики број програма за вођење система база података може да прими између 1000 и 2000 упита у секунди на уобичајном x86 хардверу (dual Pentium III 800MHz 1Gb ram). MySQL је способан да прими 13000 упита у секунди на quad Pentium III 700MHz извлачећи рекорд уз помоћ примарног кључа из табеле са милион рекорда. Тако да на питање „Да ли је MySQL довољно брз за моје потребе?“ одговор би „Да, и вероватно много бржи од онога што вам је потребно.“ Наравно као и сваки други програм MySQL може да буде спор у одређеним ситуацијама, до овога може доћи на пример ако не обраћамо пажњу на тачност команде када шаљемо упите. Ово се може избећи схватањем начина рада сервера.[5]

8.1.2.Поузданост:

MySQL је стекао репутацију да може да ради данима, па чак и месецима без интервернције корисника – након основног подешавања. Наравно повремено се јављају проблеми или долази до откривање грешака унутар програма али то је уобичајно и за било који други сервер база података. Права је реткост да „падне“ MySQL сервер, али када дође до тога врло лако се враћа у претходно стање. Ова репутација коју је стекао MySQL привукла је пажњу неколицине озбиљних корисника који су одлучили да са комерционалних програма пређу на MySQL. Неки од њих су *Yahoo! Finance, Cisco, Texas Instruments, the United States Census Bureau, NASA, Novell, Blue World Communications, Motorola* и многи други. Развојни тим је веома фокусиран на поузданост и стабилност. Било је случајева да су нове верзије MySQL-а одгађане због минорних грешака у програму које су биле откривене, али нису биле отклоњене. У случају откривања веће грешке развој тренутне верзије би се обуставио и кретало би се од нуле, а развојни тим објавио би јавно извињење свим корисницима.[5]

8.1.3.Мала захтевност за системским ресурсима:

MySQL ће максимално искористити ресурсе које му дате. Наравно са што више ресурса располаже перформансе ће бити боље, али минимални ресурси неће учинити MySQL неупотребљивим као што се догађа се неким другим серверима база података. MySQL сасвим коректно ради и на конфигурацијама типа Pentium 166Mhz и 32Mb PAM-a, али било је искустава и са конфигурацијама које су имале и слабије спецификације од горе наведене. Минимум PAM меморије који је довољан за покретање MySQL-a је између 2 и 3MB-a па је теоретски могуће покренути MySQL и на системима са 4MB PAM-a. [4]

8.1.4Флексибилно побољшавање перформанси:

Проверено је у пракси да MySQL веома добро скалира перформансе на системима са до 4 процесора и до 4GB-a PAM-a и у потпуности искоришћава могућности системских ресурса. Такође је утврђено да се добро сналази са табелама које имају неколико милијарди записа. MySQL се без проблема носи са чак до 1500 корисника у исто време без значајнијег пада перформанси. Врло је вероватно да крајњи лимити скалирања перформанси MySQL-a уопште нису ни утврђени. У тренутку писања овог текста MySQL девелопери нису успели да испитају хардверске и софтверске границе, тако да се највише ослањају на извештаје њихових корисника о понашању MySQL-a под оптерећењем. Ако би икад захтевност прерасла могућности сервера, увек остаје могућност повезивања више сервера у јединствен систем и тако разделити оптерећење. [4]

8.1.5.Рад на различитим платформама:

MySQL има могућност рада на различитим архитектурама и оперативним системима. Оперативни системи који се фреквентно користе су Linux, Windows, Solaris и FreeBSD. MySQL такође ради и на Irix, HP-UX, AIX, SCO, Tru64, OpenBSD, NetBSD, и Mac OS X.[5]

8.1.6.Подржава велику количину програмских језика:

Када радите на развоју апликација за управљање базама података, једна од највећих брига вам је да ли ће те моћи повезати вашу апликацију са сервером база података користећи одређени програмски језик. И на овом пољу се MySQL издиже изнад конкуренције пошто програмери са MySQL-ом могу комуницирати користећи C/C++, PHP, Perl, Java, Python, TCL, Ruby и Eiffel и Eiffel. [5]

8.1.7.ODBC

Поред велике подршке за програмске језике, MySQL такође има подршку за ODBC. Ово пружа програмеру могућност да креира апликације користећи *Open Database Coonnectivity* (ODBC) стандард. Подршка за ODBC такође пружа могућност да се MySQL користи са другим менаџерима база података који имају ODBC. Неки од тих апликација су *Microsoft Acces*, *Microsoft Excel*, *Crystal Reports* и многи други. MySQL захваљујући ODBC-у може да се користи у комбинацији са *Visual Basic-om*, *Delphi-em*, *ASP-om (Active Server Pages)*, *ColdFusion-om*, *BorlandBuilder-om* и многим другим развојним алатима и окружењима. [5]

8.1.8.Бесплатно или повољно лиценцирање:

MySQL се дистрибуира под одредбама GPL-а (General Public License) које су прописали људи из FSF-а (Free Software Foundation) .Према овој лиценци корисник може да користи с софтвер потпуно бесплатно како за некомерцијалне сврхе тако и за комерцијалне сврхе. Ово значи да се MySQL дистрибуира корисницима потпуно бесплатно, осим када се корисник одлучује за MySQL *Enterprise* за који је потребно плаћање додатне лиценце. У случају да је лиценца потребна да ли из разлога да организација у којој се користи софтвер не дозвољава коришћење GPL производа или се корисник одлучи за *Enterprise* решење лиценцирање се може затражити од MySQL АВ-а и то по веома повољним условима. Лиценца за један сервер износи 200\$, али цена драстично пада повећавањем броја лиценци. Лиценце се издају по серверу и не ограничавају број корисника.[3]

8.1.9.Повољна комерцијална подршка:

За оне који желе да користе MySQL за захтевније послове веома је битна квалитетна корисничка подршка. Чак и да не дође до проблема корисник ће се осећати много сигурније ако буде знао да постоји неко коме се може обратити ако којим случајем до проблема дође. Да овакве подршке нама корисницима би било много теже одредити се за MySQL. MySQL АВ пружа разнолику подршку по веома разумним ценама укључујући и свакодневну телефонску подршку. Цене се крећу од 1500\$ за основни пакет па све до 48000\$ за делуц пакет. Пошто на ваша питања одговарају сами девелопери нећете морати да причате са неколико људи пре него што добијете одговоре на ваша питања.[4]

8.1.10.Јака подршка корисника:

MySQL је више од обичне базе података, оснивачи MySQL-а су се одувек трудили да заједници њихових корисника дају што више. Корисници су на то одговорили лојалношћу и преданошћу. Каког ефекта ово има на ентерприсе корисника. Најочигледнији ефекат је да поред стандардне подршке коју пружа MySQL додатну подршку може добити од осталих корисника. Ако би сте поставили неко питање везано за MySQL на неком форум или IRC каналу који има везе са базама података врло је вероватно да ће те у врло кратком року добити одговоре. За разлику од класичне подршке за тачност ових одговора MySQL АБ не гарантује али и корисничким комунама се могу наћи велики експерти који дају одличне савете. Овакав вид подршке можете упоредити са пецањем, ако знате да пецате можете добити бесплатну рибу из језера, ако не знате пецати или вам се једноставно не иде до језера увек можете отићи у продавницу и купити рибу. Још једна од добрих страна корисничких комуна је чињеница да врло лако можете наћи експерте који би били вољни да раде за вас. Још један од добрих аспеката је “сигурност за опстанак”. Велика брига корисника софтвера мањих компанија је будућност тог производа и саме компаније. Чак и да MySQL АБ престане са радом корисници би уз помоћ изворног кода одржали програм у животу. [2]

8.1.11.Доступност изворног кода:

Могућност приступа MySQL-овом изворном коду је веома велика предност посебно за кориснике који у свом стручном кадру имају искусне C/C++ програмере. Ово вам омогућује да вршите разна прилагођавања, побољшања, измене, поправке грешака без потребе да чекате на издавача софтвера да то учини. Још једна од предности отвореног кода је много мања вероватноћа да ће доћи до грешака у коду или до лошег кода, јер отворени код може свако да види и много је мање вероватно да ће неки програмер желети да сви виде да му се поткрала грешка у коду.[3]

8.2.Мане MySQL-а:

8.2.1.Недостатак неких могућности SQL-а:

Ово представља вишеструке проблеме у повезивању постојећих апликација са MySQL-ом. Ако ваша апликација користи неку од особина коју MySQL нема, мораћете да репрограмирате те делове апликације пре него што је повежете са MySQL-ом. У неким случајевима ово може бити исцрпљујући задатак. Многи зналци из ИТ света сматрају да неке опције које MySQL тренутно нема се под хитно морају додати. Многи програмери су навикли да користе те опције и оне представљају есенцијални део њихово програмерског репертоара. Овај недостатак MySQL-а понекад зна да одвраћа потенцијалне кориснике. Реакција корисника би се могла упоредити са човеком који је свој цео живот возио аутомобил са аутоматским мењачем и сада мора да пређе на мануелни.[5]

С друге стране искуснији MySQL корисници су навикли да живе са овим недостацима чак и уживају у изазову да заобиђу ове недостатке. Изазвано нужношћу ситуације програмери често откривају и много елегантнија решења од оних које су до сад користили. Пронашли су начине да искористе могућности MySQL-а до пуног потенцијала и без обзира на недостатке залажу се за распрострањивање MySQL-а. Исто као што возач који користи мануелни мењач увек постиже већу максималну брзину од онога са аутоматским мењачем. Због ових недостатака професионални корисници имају подељено мишљење око употребе MySQL-а. И сада се поново поставља питање да ли је MySQL довољно добар?[5]

У реалности MySQL је више него добар за изградњу базе података у комбинацији са вештим и креативним девелоперима. Проблем се јавља када су програмери чврсто везани за неки други програм. Њихово залагање у раду са MySQL-ом не би било ни приближно залагању када би радили са неким другим програмом. Ако је овакво стање у радној организацији из чисто економских разлога не би имало смисла форсирати раднике да користе MySQL. С друге стране раднике би могао да обрадује рад са другим програмом и да за врло краatak период нађу начин да преброде ове недостатке. MySQL АВ је свестан ових недостатака и раде на томе да их у наредних неколико година отклоне.[5]

8.2.2.Недостатак озбиљнијих тестирања на одређеним платформама:

MySQL AB има веома стриктне стандарде и своје производе увек пропушта кроз неколико захтевних тестова, али ту се тестирање завршава. Следећа серија тестова се јавља када нову верзију програма са интернета преузме и по 20000 корисника дневно инсталира на различитим системима и прилагоди за рад са њиховим базама података. У току овог процеса могу се открити разне грешке и системски проблеми који једноставно нису могли да буду установљени унутар компаније. Грешке се пријављују MySQL AB-у и обично се отклањају веома брзо, али MySQL AB јако зависи од feedback-а корисника.[5]

Као резултат свега овога можемо уочити да стабилност и перформансе MySQL-а на одређеној платформи зависе од броја корисника на одређеној платформи. Што је већи број корисника мање су шансе да ће нека критична грешка остати неоткривена.[5]

Иако се MySQL AB труди да њихов производ буде што више cross-platform и да не фаворизује неку платформу испред осталих, разлика у броју корисника на одређеним платформама доводи до тога да MySQL боље ради на неким платформама. Према одређеним прорачунима око 40% корисника користи MySQL на x86 Linux платформи, 25% корисника га користи на Windows платформи, 15% одлази на FreeBSD, 15% на SPARC Solaris и 5% на све остале платформе.[5]

По нивоу квалитета оних преосталих 5% ће увек заостајати за осталима. Наравно не треба мислити да MySQL не функционише како треба на тим платформама. Чак по информацијама које се добијају од корисника установљено је да се MySQL сасвим добро сналази и на тим платформама, што је уствари последица добро написаног изворног кода. Добро написан изворни код има далеко веће шансе да ради без икаквих проблема и на платформама које званично нису тестиране. MySQL AB се наравно труди да прошири платформе на којима врши тестове и да провери стабилност и системима као што су AIX и HP-UX. Можемо очекивати да како се компанија буде ширила да ће ова мана у потпуности нестати.[5]

8.2.3.Компликовано руковање изворним кодом:

Раније је већ напоменуто колико је значајно што је омогућен приступ изворном коду програма. Имати изворни код дефинитивно омогућује велику флексибилност у раду корсника. Али кад радите са кодом којег је неко други написао прво морате да га разумете да би вам био од значаја. Код MySQL сервера је релативно тешко схватити, чак и искуснији C/C++ програмери имају проблема са разумевањем овог кода. Ретко ко се уопште усуђује да мења код. Постоје два разлога због којих је овај код толико комплексан. Прво то је код сервера база података што значи да је код оптимизован да што боље организује податке на диску, минимализује употребу диска, убрза приступ меморији и расчлањује упите.[4]

Овакав код небитно колико је добро написан би био веома комплексан. Други разлог је због такозваног “ефекта генијалног кода”. Саму основу изворног кода написао је Montz Widenius који је у програмерским круговима добио надимак “Моцарт компјутерског програмирања”. Дobar програмер би уживао гледајући његов код исто толико колико би човек са добрим слухом уживао у Моцартовим делима. Да би неко у потпуности разумео овај код мора да има исту инспирацију какву је имао Монту у том тренутку и да схвати како и зашто је тај код написан тако. Чак и најбољи програмери су ово налазили као веома велик изазов. Наравно овај изазов се може пребродити, па је развојни тим MySQL-а успео да овлада том виртуозношћу.[3]

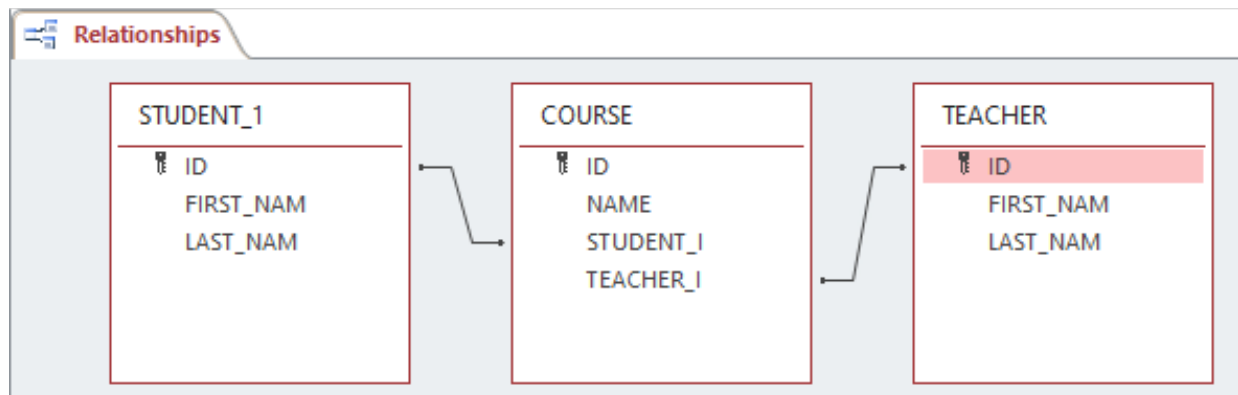
9.Пример

У овом примеру ћемо показати могућност коришћења функције JOIN на 3 табеле.

```
SELECT
  student.first_name,
  student.last_name,
  course.name
FROM student
JOIN student_course
  ON student.id = student_course.student_id
JOIN course
  ON course.id = student_course.course_id;
```

Слика 2. Код примера.

Прво формирамо три табеле преко којих ћемо да прикажемо функцију JOIN.



Слика 3. Везе табела примера.

У горе наведеном ER дијаграму можете да видите табеле, њихове колоне, типове података колона и референце између табела. На пример, постоји референца између табела студент и курс студент - сваки студент може бити повезан са више редова у табели student_course.

student – Садржи информације о студенту:

id – ID студента.

first_nam – Име студента.

last_nam – Презиме студента.

student_course – Садржи информацију на који курс иде студент:

student_id – ID студента.

course_id – ID курса.

course – Садржи информације о курсу:

id – ID курса.

name – Име курса.

student_i- Студент ID.

teacher_i – ID професора курса.

teacher - Садржи информације о предавачима:

id – ID предавача.

first_nam – Име предавача.

last_nam – Презиме предавача.

Прикази табела:

STUDENT

STUDENT			
	ID	FIRST_NAM	LAST_NAM
+	1	ANDRIJA	ANDRIC
+	2	MARKO	MARKOVIC
+	3	PETAR	PETROVIC

Табела 1. Приказ примера табеле STUDENT.

STUDENT_COURSE

STUDENT_COURSE	
STUDENT.ID	COURSE.ID
1	1
3	2
2	3

Табела 2. Приказ примера табеле STUDENT_COURSE.

COURSE

COURSE				
ID	NAME	STUDENT_I	TEACHER_I	
1	ENGLESKI	1	1	
2	PROGRAMIRANJE	3	2	
3	NEMACKI	2	3	

Табела 3. Приказ примера табеле COURSE.

TEACHER

TEACHER			
ID	FIRST_NAM	LAST_NAM	
1	NIKOLA	NIKOLIC	
2	STEFAN	STEFANOVIC	
3	JOVAN	JOVANOVIC	

Табела 4. Приказ примера табеле TEACHER.

JOIN

STUDENT_COURSE					
STUDENT.ID	COURSE.ID	FIRST_NAM	LAST_NAM	NAME	TEACHER_I
1	1	ANDRIJA	ANDRIC	ENGLESKI	1
3	2	PETAR	PETROVIC	PROGRAMIRANJE	2
2	3	MARKO	MARKOVIC	NEMACKI	3
*(New)	(New)				

Табела 5. Приказ JOIN-а.

10.3 АКЉУЧАК

MySQL je vrlo sposoban relacioni klijent/server sistem baza podataka. Veoma je siguran i kompatibilan je sa velikim brojem aplikacija. Takođe odnos cena/kvalitet je fenomenalan i to ne samo zato što je MySQL besplatan, nego zato što su njegove hardverske potrebe veoma skromne. Ali iznad svega treba istaći da u svetu Linux operativnih sistema MySQL se sve više koristi od strane drugih aplikacija za vođenje njihovih internih baza podataka bez obzira da li se radi o poboljšanju logovanja podataka, upravljanju kolekcijama MP3 fajlova, e-mail-ova, adresa ili nečega drugog. MySQL kod Linux aplikacija zauzima onu ulogu koju ima Jet Engine kod Microsoft-ovih operativnih sistema, ali u većini slučajeva MySQL pruža bolje performanse od Jet Engine-a.

Zahvaljujući ODBC interfejsu, MySQL se sad koristi i u svetu Windows operativnih sistema kao direktna zamena Jet Engine-u. Osim sa tehničke strane MySQL je u prednosti nad ostalim open source sistemima baza podataka jer je rasprostranjeniji od ostalih. Treba dodati da je MySQL mnogo podrobnije testiran i ispitan na različitim platformama, a i mnogo je lakše naći developera koji ima iskustvo sa MySQL-om nego sa bilo kojim drugim open source sistemom baza podataka. Ipak MySQL još uvek ne može da se nosi sa „velikim momcima“ iz sveta komercijalnih baza podataka. Malo je verovatno da će te odabrati MySQL ako on nema neke mogućnosti koje su vama potrebne.

11.ЛИТЕРАТУРА

1. MySQL Administrator's Guide and Language Reference (2nd Edition) by MySQL AB (Paperback - May 7, 2006)
2. MySQL Enterprise Solutions by Alexander ""Sasha"" Pachev (Paperback - Feb 4, 2003)
3. MySQL: Essential Skills by John Horn, Michael Grey, and LLC Interstate Software (Paperback - Jun 10, 2004)
4. PHP & MySQL For Dummies 3rd edition (For Dummies (Computer/Tech)) by Janet Valade (Paperback - Nov 20, 2006)
5. The Definitive Guide to MySQL 5, Third Edition (Definitive Guide) by Michael Kofler (Paperback - Sep 23, 2005)