Enter password: мой пароль 'IRONMAN11';

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

RABOTA S BAZAMI DANNIX

SHOW DATABASES; <--POKAZIVAET KAKIE EST BD NA DANNOM SERVERE;

SELECT DATABASE(); <--yznat tekyschyy Data Base;

SELECT USER(); <--vozvraschaet imia polzovatelia (y menia -> root@localhost)

CREATE DATABASE nazvanie\_novoi\_BD; <--sozdanie novoi BD.

sc query mysql80 <--NABRAT CHTOBI YZNAT ZAPYSCHEN LI MYSQL SERVER. Rabotaet v windows-konsoli.

**Chtobi ostanovit server, libo enable**

1)-nado libo vizvat task manager(ctrl + Alt + Delete), tam naiti mysqld.exe,

nazzimaesh pravoi klavishei i vibrat 'end task' <-- eta komanda viklychaet server. No kak vklychit

potom cherez etot task manager neponiatno.

2)-Libo naiti prilozzenie Windows services, naiti MySQL80 i ot tyda mozzno zapystit libo ostonovit.

3)-Esche odin metod no tolko ostanovki servera. 'Doiti' do direktorii gde lezzit papka bin, t.e.

C:\Program Files\MySQL\MySQL Server 8.0\bin> zatem nabrat mysqladmin -u root -p shutdown t.e.

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqladmin -u root -p shutdown;

4)-Takzze mozzno viklychit s pomosch`y workBench

Server naxoditsia na moei lokalnoi mashine.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

\c <-- otmeniaet komandy

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

It is important to know what the '>, ">, and `> prompts signify, because if you mistakenly enter an

unterminated string, any further lines you type appear to be ignored by mysql.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Переключения между БД** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

USE 'nazvanie BD' <-- pereklychaet na administrirovanie dannoi BD; **Если в названии БД или колонки или таблицы**

**есть пробел то используй обратный апостроф - `... ...`**

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

SELECT database(); <--pokazivaet kakaia BD ispolzyetsia;

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**VSTREOENNIE FYNKCII:**

select curdate(); <-- nepriviazano k kakoilibo BD, otrozzaet tekyschyy daty,pomogaet pri vichislenii (raboti) s datami.

select name, birth, curdate(), timestampdiff(year,birth,curdate()) as age from pet; <--timestampdiff imeet 3 argymenta

pervii argyment eto unit in which you want the result(dni,mesiaci,godi,chasi,min...), vtoroi i tretii argyment eto dati

iz kotorix i bydet vicheslena raznica.

SELECT name, birth, CURDATE(), TIMESTAMPDIFF(YEAR,birth,CURDATE()) AS age FROM pet ORDER BY name;<-- tozze samoe no s sortirovkoi

year(), month(), dayofmonth() <-- fynkcii dlia raboti s tipom dannix date, kotorie vozvraschayt sootvetstvyyschie tipi dannix,

t.e. delayt viborky i vozvraschayt god ili mesiac ili den.

SELECT name, birth FROM pet WHERE MONTH(birth) = 5; <--vibrat name, birth iz pet gde mesiac rozzdenia mai

regexp\_like() <-- fynkciia primeniaemaia s regyliarnimi virazzeniiami

COUNT(), MAX(), MIN(), SUM(), AVG() <-- primeniaetsia chasche s group by;

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**пример работы с типом данных date**

SELECT CURDATE(); ← Вернёт текущую дату

Если надо вставить значение даты то формат год-месяц-день

Iz tipa dannix date mozzno otdelno delat viborky po dniam, mesiacam, godam. T.e. esli y nas est data naprimer

1960-12-03 i mi xotim yznat tolko mesiac ili tolko den ili tolko god. Dlia etogo est vstroennie fynkcii:

year(), month(), dayofmonth(). PRIMER:

SELECT name, birth FROM pet WHERE MONTH(birth) = 5; <--vibrat name, birth iz pet gde mesiac rozzdenia mai

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**RABOTA S PATTERN MATCHING:**

SQL pattern matching enables you to use '\_' to match any single character and '%' to match an arbitrary

number(произвольное число)

Naprimer:

SELECT \* FROM pet WHERE name LIKE 'b%'; <--To find names beginning with b

SELECT \* FROM pet WHERE name LIKE '%fy'; <--To find names ending with fy

SELECT \* FROM pet WHERE name LIKE '%w%'; <--to find names containing a 'w'

SELECT \* FROM pet WHERE name LIKE '\_\_\_\_\_'; To find names containing exactly five characters, use five instance of '\_'

regexp\_like() <-- fynkciia primeniaemaia s regyliarnimi virazzeniiami, mozzno sokraschenno REGEXP ili RLIKE <-- eto sinonimi.

. <--simvoliziryet lyboi otdelno stoiaschii simvol

[...] <--matches any character within the brackets. For example, [abc] matches a, b, or c. To name a range of

characters, use a dash. [a-z] matches any letter, whereas [0-9] matches

any digit.

To anchor a pattern so that it must match the beginning or end of the value being tested, use ^ at the

beginning or $ at the end of the pattern

Primeri regexep\_like():

SELECT \* FROM pet WHERE REGEXP\_LIKE(name, '^b');<--To find names beginning with b, use ^ to match the beginning of the name

SELECT \* FROM pet WHERE REGEXP\_LIKE(name, 'fy$');<--To find names ending with fy

SELECT \* FROM pet WHERE REGEXP\_LIKE(name, 'w');<--To find names containing a w

SELECT \* FROM pet WHERE REGEXP\_LIKE(name, '^.....$');<--find names containing exactly five characters

SELECT \* FROM pet WHERE REGEXP\_LIKE(name, '^.{5}$'); <--tozze samoe no s operetorom '{n}'-repeat n times

v {} <--etix skobkax pishesh skolko raz povtorit nado kakyy-libo f-ciy.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**ПРОСМОТРЕТЬ ДАННЫЕ В ТАБЛИЦЕ (VIBORKA)**

Посмотреть данные в таблице можно через выполнение запроса SELECT \* FROM tableName;

Select \* from pet where name='Bowser'; <-- vibrat ty stroky iz takoi-to\_tablici gde name='Bowser';

Select \* from pet where birth>='1998-1-1'; <--vibrat te stroki gde znachenie stolbca birth>='1998-1-1'

SELECT \* FROM pet WHERE species = 'dog' AND sex = 'f'; <--zdes primenen sintaksis s and kotorii pozvoliaet

delat 'viborky' bolee yzko napravlennoi(specifichnoi dlia yzkogo kolichestva polei).

SELECT \* FROM pet WHERE species = 'snake' OR species = 'bird'; <--or imeet takoi-zze princip kak v Java operatori

sravneniia t.e. eto ili eto))).

SELECT \* FROM pet WHERE (species = 'cat' AND sex = 'm') OR (species = 'dog' AND sex = 'f');<--ispolzovanie parentheses

t.e. skobki imeet takoe-zze znachenie kak v Java.

SELECT name, birth FROM pet <--mozzno delat viborky cherez zapiatie, i vibirat nyzznie stolbci

SELECT DISTINCT owner FROM pet; <--primenenie distinct vernet tolko ynikalnie znacheniia v stolbce.

SELECT name, species, birth FROM pet WHERE species = 'dog' OR species = 'cat'; <-- combinaciia...

(SORTIROVKA)

SELECT name, birth FROM pet ORDER BY birth; <--order by oznachaet v otsortirovannom vosxodiaschem (ascending) poriadke

SELECT name, birth FROM pet ORDER BY birth DESC; <--desc korrektiryet sortirovky v nisxodiaschem (discending) poriadke.

SELECT name, species, birth FROM pet ORDER BY species, birth DESC; <--rezyltati viborki mozzno sortirovat v raznom pori-

adke. Zdes species sortiryytsia po ymolchaniy(vosxodiasche), birth nisxodiasche.Esli ykazanno DESC to dannaia

komanda deistvyet na stolbec ykazanni pered DESC. i ne zatragivaet drygyy viborky.

(Counting Rows)

SELECT COUNT(\*) FROM pet; <--counting all rows in the table;

SELECT owner, COUNT(\*) FROM pet GROUP BY owner;<-- group by-vozmet pervoe vstretivshaiasia ynikalnoe znachenie iz owner

i tak-kak ispolzyetsia instrykciia count poschetaet skolko vxozzdenii etogo ownera imeetsia v tablice pet, i

vivedet eto znachenie v colonke pod nazvaniem count(\*).Est sledyyschie instrykcii: count(\*),sum(),max(),min(),avg().

SELECT species, sex, COUNT(\*) FROM pet GROUP BY species, sex;<--zdes group by i po species i po sex

(inner join)

An INNER JOIN permits rows from either table to appear in the result if and only if both tables meet the conditions specified in the ON clause.

SELECT 'nazvanie\_colonok' FROM 'nazvanie\_tablici' INNER JOIN 'nazvanie\_vtoroi\_tablici' ON 'nazvanie\_kolonki\_pervoi\_tablici'=

'nazvanie\_kolonko\_vtoroi\_tablici';

Naprimer:

SELECT distinct product.maker, laptop.speed FROM product inner join laptop ON product.model=laptop.model WHERE laptop.hd>=10

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

—--------------------------------------------**Внесение данных в таблицу**—-------------------------------------------------------

INSERT INTO название\_таблицы (*column1*, *column2*, *column3*, ... ) VALUES (*value1*, *value2*, *value3*, ...); ← Specify both the column names and the values to be inserted:

Если вносимое значение является строкой и вней имеется симол одинарной кавычки → ‘ ← то его надо экранировать обратным слэшем, \’ ←так как написание значений в этой команде идёт в одинарных кавычках и программа падает, непонимая , что выполнять.

**Создание таблицы (table) / удаление**

CREATE TABLE IF NOT EXISTS TABLE\_NAME (column\_name datatype, column\_name datatype);

CREATE TABLE nazvanie\_tablici ('nazvanie\_stolbca' VARCHAR(20), 'nazvanie\_stolbca' VARCHAR(20),

'nazvanie\_stolbca' VARCHAR(20), 'nazvanie\_stolbca' CHAR(1), 'nazvanie\_stolbca' DATE, 'nazvanie\_stolbca' DATE);

DROP TABLE название\_таблицы;

create table event (name varchar(20), date date, type\_of\_event varchar(15), remark varchar(255));

SHOW TABLES <-- pokazivaet syschestvyyschie tablici v BD

**describe nazvanie\_tablici**; <--vivodit dannyy table.(shabky).

—----------------------------------------**IZMENENIE DANNIX V TABLICE—-------------------------------------**

UPDATE nazvanie\_tablici SET nazvanie\_stolbca\_gde\_proizvodim\_izmeneniia = 'vnosimie\_izmeneniia' WHERE name = 'Bowser';

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**УДАЛЕНИЕ ДАННЫХ ИЗ ТАБЛИЦЫ**

TRUNCATE nazvanie\_tablici <--stiraet vse dannie is tablici

delete from nazvani\_tablici <--stiraet vse dannie iz tablici. **(DELETE FROM page\_information WHERE name='intputForAnAnswer';)**

DROP TABLE nazvanie\_tablici; ← УДАЛЕНИЕ ТАБЛИЦЫ. ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

ISPRAVLENIE RAZNIX OSHIBOK:

--------Y menia ne rabotala zapis v bd iz faila, kak reshit?

1)-nado zdelat skritie papki vidimimi

2)-v dokymente my.ini (eto configyracionni fail)

a)-v razdele [client] ykazat loose-local-infile=1

b)-v razdele [mysqld] ykazat local\_infile=1

--------Не определялся путь (path)

komanda --> LOAD DATA LOCAL INFILE '\Users\user\Desktop\pet.txt' into table pet; ← ne chitalas.

reshenie--> LOAD DATA LOCAL INFILE '/Users/user/Desktop/pet.txt' into table pet; ←надо перевернуть слэш.

--------Sozdannaia tablica otrozzalas vsia razorvannaia. Ia ispolzoval zapis s faila i komandy:

LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/event.txt' into table event;

potom ia poproboval

LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/event.txt' into table event LINES TERMINATED BY '\r\n';

t.e. dobavil LINES TERMINATED BY '\r\n'; <-- esli nabiraesh tekst v windows redaktore to nado tak

propisivat. O nashol takoe ob`iasnenie - eto poxody sviazano s tem chto sam redaktor teksta gde

nabrana informaciia ispolzyet simvoli \r,\n-dli perenosa stroki i eto kak-to konfliktyeyt s reda-

ktorom v MySQL. V obschem kogda zalivaesh table iz faila to nado propisivat LINES TERMINATED BY '\r\n';

takim obrazom tablici otrozzaytsia korrktno.

—-----Не мог перейти в консоле на бд для редактирования. Для перехода использовал:

**use computer firm** ← выдавало ошибку по синтаксису, мол нет такой бд. **Причина- пробел в названии**.

Решение: заключить название БД в обратные апострофы, это актуально для любого названия где есть пробел (название БД, столбца, таблицы)

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

RABOTA S FAILAMI :

[ ZAPIS S FAILA V MYSQL TABLICY ]

Chtobi ne vipolniat komandi pri povtornom sozdanii kakix-to dannix mozzno sozdat fail i v faile zapisat

znachenie kazzdogo stolbca cherez odno nazzatie na Tab (no ne cherez probel).

Naprimer eta shabka tablici: name | owner | species | sex | birth | death

sozdaem fail i vnosim Whistler Gwen bird null 1997-12-09 null

Vnosim dannie v stroky,sledyia zadannomy poriadky nazvaniia stolbcov, ochen vazzno vnosit inf. cherez Tab.

OOOCHEN KLASSNO SOZDAESH TAK KAK-BI BAZOVYY VERSIY TABLICI.

Primer komandi:

LOAD DATA LOCAL INFILE '/pyt/nazvanie\_faila.txt' into table nazvanie\_tablici;

mysql> LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/event.txt' into table event;

LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/event.txt' into table event LINES TERMINATED BY '\r\n';

tyt ykazana LINES TERMINATED BY '\r\n'; v konce dlia korrektnogo otobrazzenia tablici.

[ZAPIS SODERZZIMOGO TABLICI NA FAIL]

Nelzia zapisivat kyda xochesh, est specialnoe mesto dlia exportnix failov eto papka Uploads.

Polnii pyt mozzet bit takim-> C:\ProgramData\MySQL\MySQL Server 8.0\Uploads

Komanda kotoraia zapisivaet fail:

SELECT \* FROM nazvanie\_tablici INTO OUTFILE 'pyt(lytshe skopirovat iz Uploads)/filename.txt';

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

SELECT \* FROM pet INTO OUTFILE '/Users/user/Desktop/someFail.txt';

SELECT \* FROM tablename INTO OUTFILE "C:\\ProgramDATA\\MySQL Server 8.0\\Uploads\\filename.csv"