

## УПРАЖНЕНИЕ 1

### СЪЗДАВАНЕ НА БАЗА ОТ ДАННИ(СХЕМА) И ТАБЛИЦИ В MySQL

MySQL е open source релационна, управляваща бази от данни система (RDBMS), която използва SQL, най-популярния език за въвеждане, достъп и извършване на други процеси с данни в база данни. MySQL функционира в мрежова среда и използва архитектура от типа клиент/сървър т.е една централна програма работи като сървър и различни клиентски приложения се свързват към нея, за да изпращат заявки. MySQL може да работи с различни версии на Windows, Linux, UNIX, като връзката клиент-сървър не е ограничена само до среди, в които всички компютри използват една и съща операционна система. Клиентските програми могат да осъществяват връзка със сървърна, която е стартирана на същия или на различен хост.

Предлагат се много различни клиентски програми. Ето няколко от тях:

- MySQL Query Browser и MySQL Administrator, които представляват графични интерфейси на сървъра.
- mysql е програма с команден интерфейс, която функционира като текст-базиран интерфейс на сървъра. Използва се за заявки и преглед на резултатите в интерактивен вид от прозорец на терминал.
- mysqlimport ( за импортиране на файлове с данни), mysqldump (за създаване на резервни копия), mysqladmin(за администриране на сървъра), mysqlcheck(за проверка на целостта на файлове на базата от данни).
- MySQL Workbench – официалният инструмент от Oracle за връзка към сървъри и/или бази данни, манипулация на бази данни, изграждане на Entity/Relationship (същност/връзка) диаграми на бази данни и много други функции.

#### **1. Видове връзки. Създаване на връзка.**

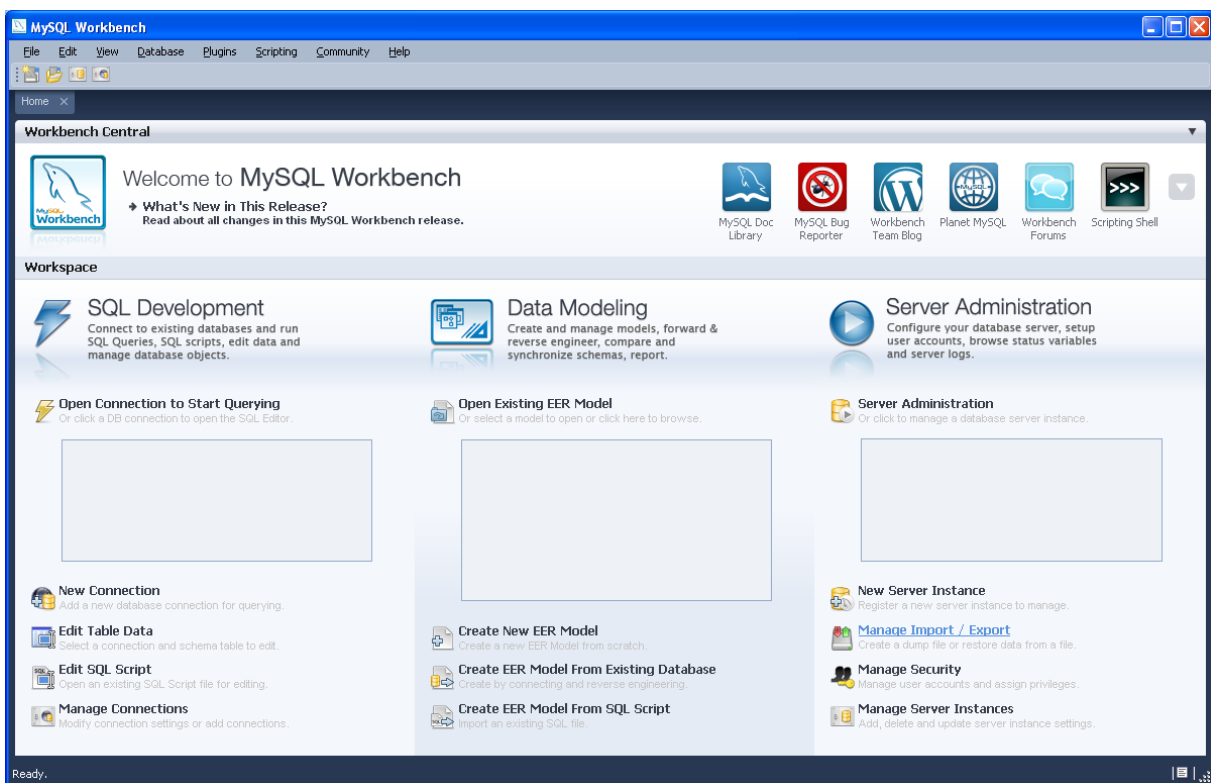
За да се свържем с даден MySQL сървър и някоя от неговите бази данни, трябва първо да създадем връзка към него. Връзката съдържа име или IP адрес на сървъра, номер на порт,

потребителско име и парола за достъп и други незадължителни параметри.

В нашата лаборатория потребителят за достъп е стандартният суперпотребител за MySQL – root, а паролата е празна.

**Внимание: При нас за улеснение паролата е оставена празна, което позволява достъп до базата данни без парола. Такава практика в реална система е опасна и се препоръчва винаги да се използват възможно най-сложни пароли.**

## 1.1 Създаване на връзка



### New Connection

The screenshot shows the 'Setup New Connection' dialog box. The 'Connection Name' field is 'Local Server'. The 'Connection Method' is 'Standard (TCP/IP)'. The 'Parameters' tab is selected, showing 'Hostname' as '127.0.0.1', 'Port' as '3306', 'Username' as 'root', and 'Password' with 'Store in Vault ...' and 'Clear' buttons. The 'Default Schema' field is empty. The 'Test Connection', 'Cancel', and 'OK' buttons are at the bottom right.

В полето **Connection Name** може да се сложи име, под което тази връзка да се запази и да се вижда за бързо избиране на това копие на клиентската програма (MySQL Workbench). Името не е задължително да съвпада с някой от параметрите на сървъра, а трябва да бъде описателно за потребителя.

В полето **Hostname** трябва да се запише името в мрежата или IP адресът на сървъра, до който изграждаме достъп. В нашата среда на всеки компютър има инсталирано копие на MySQL сървър и всеки ще работи локално на своето копие, затова в това поле се записва IP адрес 127.0.0.1 (local host).

В полето **Port** се записва номерът на TCP порта, на който отговаря MySQL сървърът. По подразбиране той е 3306.

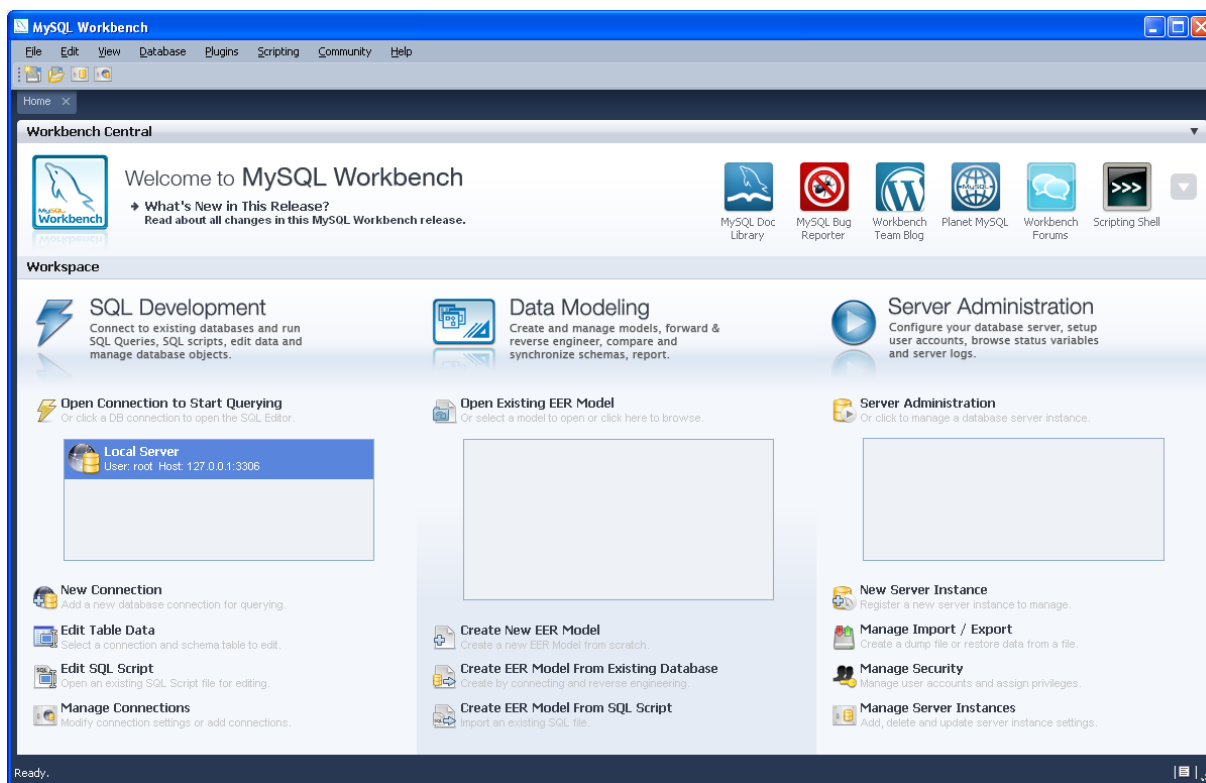
В полето **Username** се записва името на потребителя за достъп до сървъра – в нашия случай root.

В полето **Password** може да се зададе парола за потребителя, което в нашата среда не се изисква.

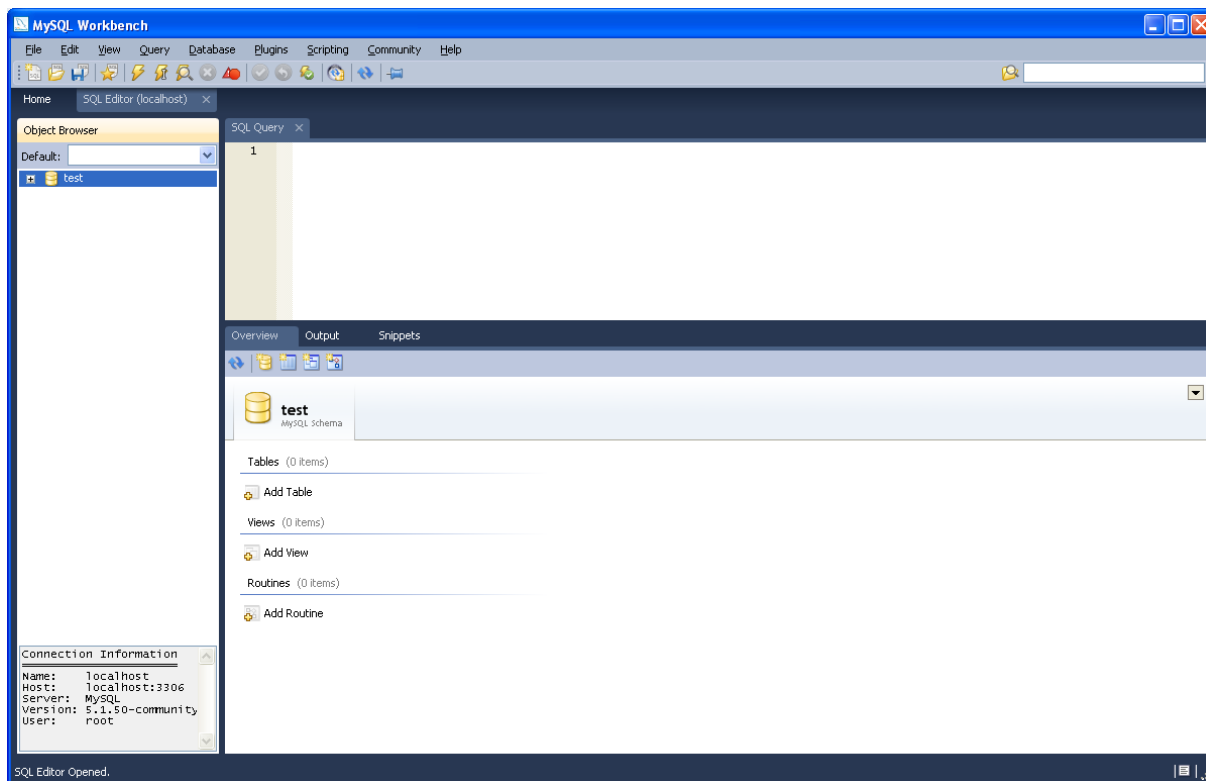
В полето **Default Schema** може да се зададе подразбиращата се база данни (схема), която да се избира при осъществяване на връзка към сървъра (ако има повече от една).

В полето **Advanced** могат да се зададат допълнителни, незадължителни параметри за връзка.

С бутона **Test Connection** може да се провери валидността на зададените параметри, а с **OK** да се повърди създаването.



На фигурата се вижда вече създадена връзка.



След отваряне на връзката се виждат схемите (базите от данни), съществуващи на този сървър. По подразбиране в MySQL всяка новосъздадена връзка има една празна база данни с име **test**.

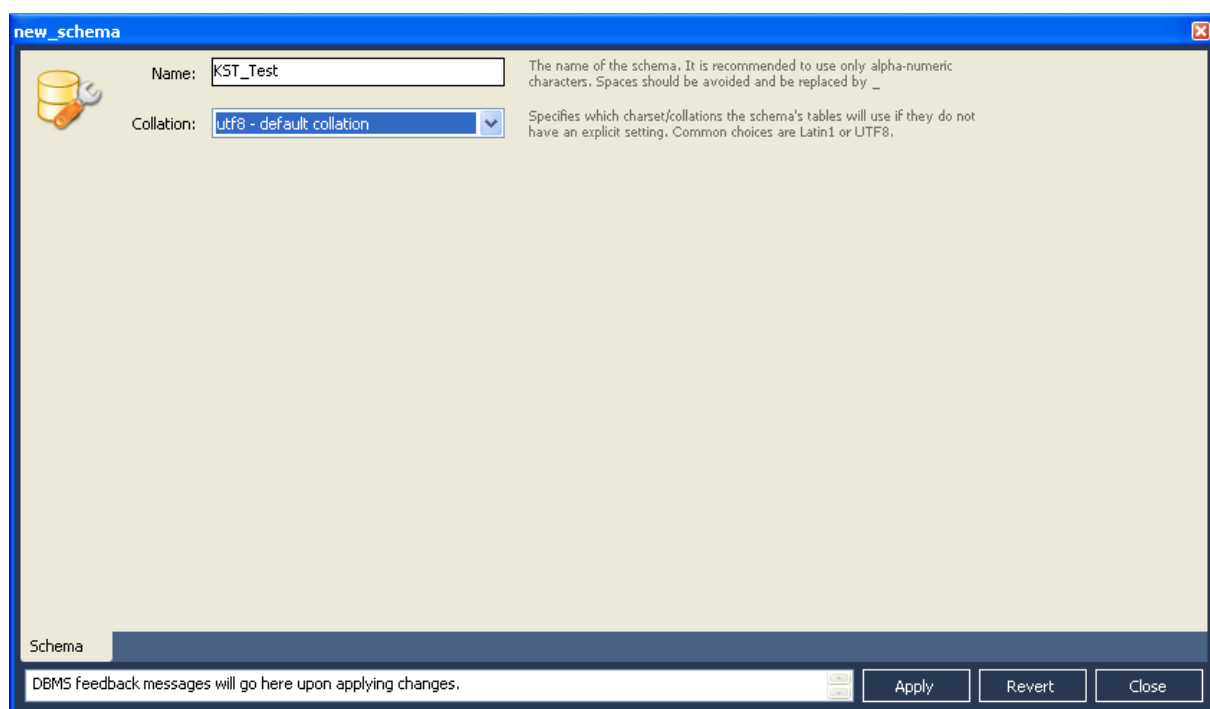
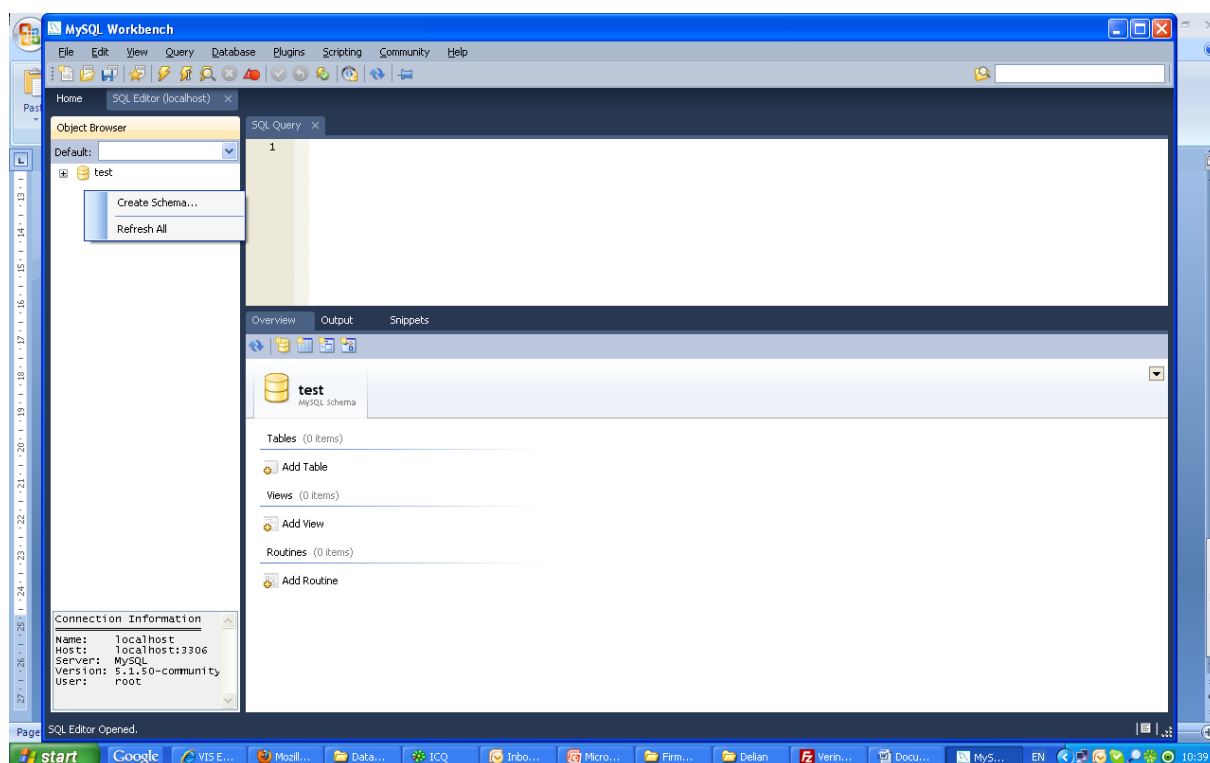
## **2. Общи свойства на базите от данни. Свойства на таблиците**

В MySQL базите от данни се съхраняват на едно място, наречено „директория за данни“ („data directory“).

1. Сървърът представя всяка база от данни посредством поддиректория на директорията за данни. Тази поддиректория се нарича „директория на базата от данни“ следователно директорията за данни е родителска за всички директории на базите от данни.
2. Всяка директория на базата от данни има подразбиращ се набор от знакове и кодации. Свойствата на базите от данни се задават при създаването им и те се съхраняват във файла db.opt в същата директория.
3. Базите от данни не могат да се влагат. Една база от данни не може да съдържа друга такава.

Според терминологията на MySQL друго наименование на „база от данни“ е „схема“ (schema).

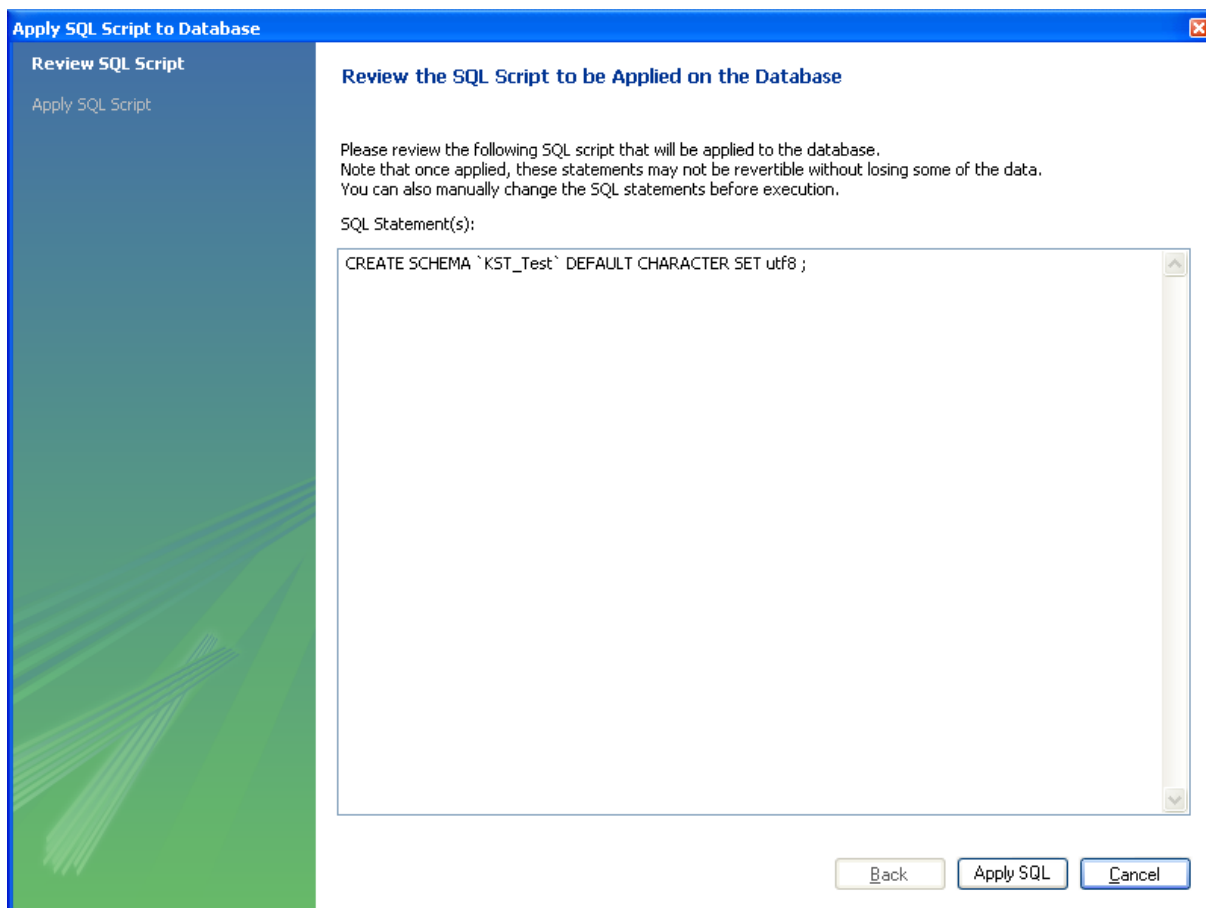
### **2.1. Създаване на схема:**



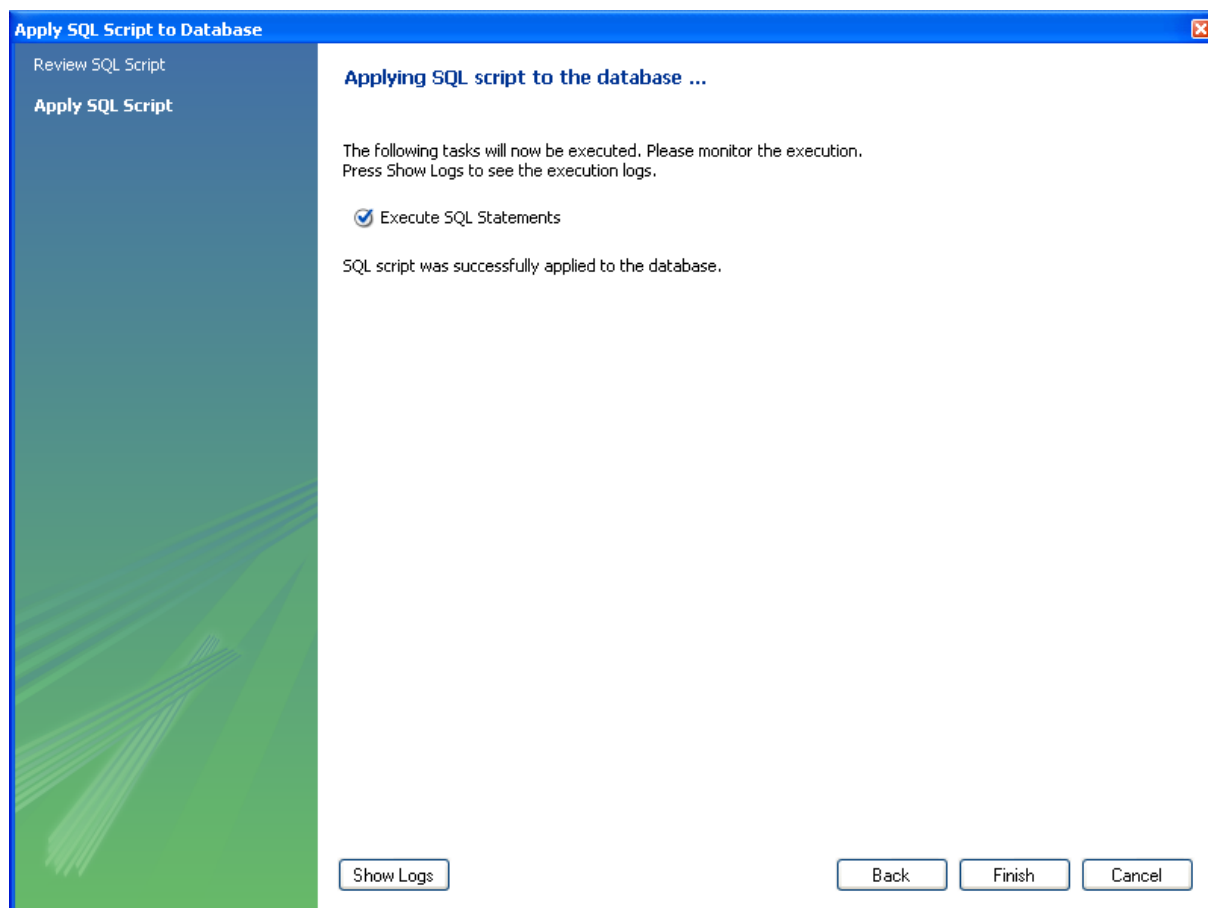
При създаване на схема трябва задължително да се зададе името, под което тя ще се вижда в сървъра.

При нужда може да се промени подразбиращата се колация на сървъра (набор от символи) на базата данни. Колацията по подразбиране за сървъра се задава при инсталирането му.

За да има възможност за работа със символи на кирилица колацията на базата данни трябва да бъде UTF8 или CP-1251. Да се създаде таблица Student със следната структура:



След потвърждаване на параметрите, програмата MySQL Workbench превръща нашите команди в SQL скрипт, като ни дава възможност да разгледаме и при нужда да редактираме параметрите му.



## Свойства на таблиците

Таблиците имат физическа и логическа структура. Логически погледнато всяка таблица се състои от редове и колони. Таблиците могат да са празни, но трябва да имат поне една колона. За да се подобри бързодействието на заявките, таблиците могат да бъдат и индексирани. Индексите стават все по-важни с нарастване на таблиците. Работата с индекси е предмет на следващо упражнение.

Физически погледнато, всяка таблица е асоциирана с един или повече файлове върху диска. Всяка таблица има собствен форматен файл в директорията на базата от данни. Той се създава от сървъра и съдържа дефиницията или структурата на таблицата. Името на форматния файл е същото като името на таблицата плюс разширеноето `frm`.

## Създаване на таблица

MySQL предлага няколко начина на създаване на таблици:

1. Празна таблица се създава или като се зададе изрично дефиницията ѝ, или като се използва дефиницията на съществуваща таблица.



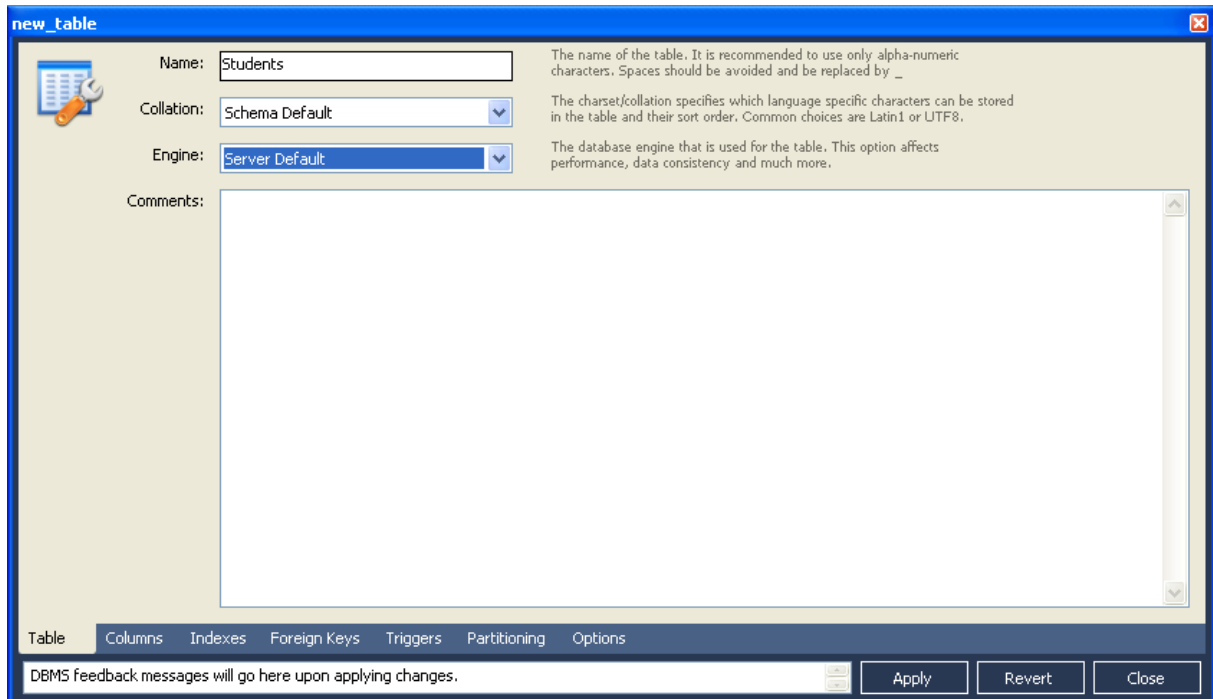
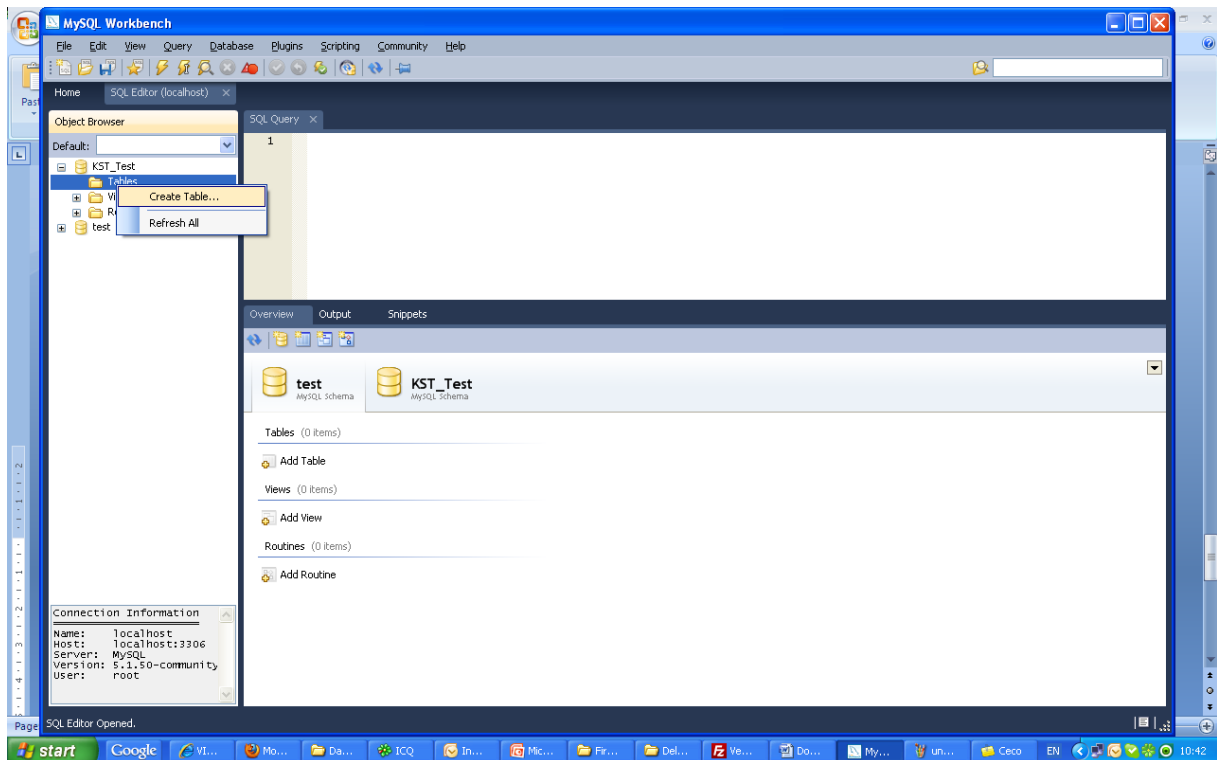
2. Може да се създаде таблица, попълнена с резултата от конструкция SELECT.

3. Може да се създаде временна таблица.

Всеки един от методите за създаване на таблици използва конструкцията CREATE TABLE.

create table име\_на\_таблицата (дефиниции \_на \_колони);

## Създаване на таблица



При създаването на таблица трябва да се зададе нейното име, колация и машина за обработка. Засега различните машини за обработка на таблиците са извън обхвата на дисциплината и ще се работи с подразбиращата се за сървъра машина.

## 2.2. Типове данни (MySQL)

### Целочислени

Тип	Значение	Размер
BIT (M)	0 или 1	1 бит
TINYINT	-127 ÷ 128 (0 ÷ 255)	1 байт
SMALLINT	-32768 ÷ 32767 (0 ÷ 65535)	2 байта
MEDIUMINT	-8388608 ÷ 8388607 (0 ÷ 16777215)	3 байта
INT	-2147483648 ÷ 2147483647 (0 ÷ 4294967295)	4 байта
BIGINT	-9223372036854775808 ÷ 9223372036854775807 (0 ÷ 18446744073709551615)	8 байта

- INT(4) – четирицифрено цяло
- ZEROFILL: 4 -> 0004
- NOT NULL
- AUTO\_INCREMENT
- SIGNED/UNSIGNED
- UNIQUE
- SERIAL = BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT  
UNIQUE

### Дробни

Тип	Значение	Размер
FLOAT	-3.40E+38 ÷ +3.40E+38	4 байта
DOUBLE	-2.22E308 ÷ +1.8E308	8 байта
DECIMAL(M,D)	M цифри (до 65), D до 30	9 цифри – 8 байта

- DOUBLE PRECISION = DOUBLE
- REAL = FLOAT или DOUBLE
- NUMERIC = DEC = DECIMAL
- DECIMAL(5,2): от -999.99 до 999.99

### Дати и време

Тип	Значение	Размер
DATETIME	'1000-01-01 00:00:00' ÷ '9999-12-31 23:59:59'	8 байта
DATE	'1000-01-01' ÷ '9999-12-31'	3 байта
TIMESTAMP	'1970-01-01 00:00:01' UTC ÷ '2038-01-19 03:14:07'	4 байта
TIME	'-838:59:59' ÷ '838:59:59'	3 байта
YEAR	1901 ÷ 2155	1 байт

### Текстови променливи

Тип	Значение	Размер
CHAR(N)	CHAR(4) – 4 байта	0-255
VARCHAR(N)	размер – 1 до N+1	0-255
BINARY(N)	BINARY(3) -> 'a\0\0'	0-255
VARBINARY	N+1, няма запълване	0-255
BLOB	Binary Large Object	2 <sup>16</sup> байта
TEXT	Текст	2 <sup>16</sup> байта

ENUM	('one', 'two', 'three') 64К ел.	1-2 б.
SET	До 64 стринга	1-8 б.

## CHAR и VARCHAR

стойност	CHAR(4)	байта	VARCHAR(4)	байта
' '	'__ '	4	' '	1
'ab'	'ab__ '	4	'ab'	3
'abcd'	'abcd'	4	'abcd'	5
'abcdef'	'abcd'	4	'abcd'	5

## BLOB и TEXT

- TINYBLOB, TINYTEXT – до 28 байта
- BLOB, TEXT – до 216 байта
- MEDIUMBLOB, MEDIUMTEXT – до 224 байта
- LONGBLOB, LONGTEXT – до 232 байта

## ENUM

NULL	-
' '	0
'one'	1
'two'	2
'three'	3

## SET

- SET('one', 'two') NOT NULL:

- ☐ ' '
- ☐ 'one'
- ☐ 'two'
- ☐ 'one,two'

- До 64 елемента

## Добавяне на колони



**new\_table**

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
FakNo	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Born	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
GotDocuments	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
WeekHours	TIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Timestamp	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Zapisan	YEAR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Specialnost	ENUM('KST', 'KTT', 'AI...)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
IBAN	CHAR(22)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Uspeh	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Stipendia	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Details

Collation:

Table Default:

Comments:

DBMS feedback messages will go here upon applying changes.

Apply Revert Close

След потвърждение отново можем да видим и редактираме SQL Скрипта за създаване на таблицата:

**Apply SQL Script to Database**

Review SQL Script

Apply SQL Script

**Review the SQL Script to be Applied on the Database**

Please review the following SQL script that will be applied to the database.  
Note that once applied, these statements may not be revertible without losing some of the data.  
You can also manually change the SQL statements before execution.

SQL Statement(s):

```
CREATE TABLE `KST_Test`.`Students` (
  `FakNo` INT NOT NULL ,
  `Name` VARCHAR(45) NULL ,
  `Born` DATE NULL ,
  `GotDocuments` DATETIME NULL ,
  `WeekHours` TIME NULL ,
  `Timestamp` TIMESTAMP NULL ,
  `Zapisan` YEAR NULL ,
  `Specialnost` ENUM('KST', 'KTT', 'AIUT') NULL ,
  `IBAN` CHAR(22) NULL ,
  `Uspeh` FLOAT NULL ,
  `Stipendia` VARCHAR(45) NULL ,
  PRIMARY KEY (`FakNo`),
  UNIQUE INDEX `FakNo_UNIQUE` (`FakNo` ASC));
```

Back Apply SQL Cancel

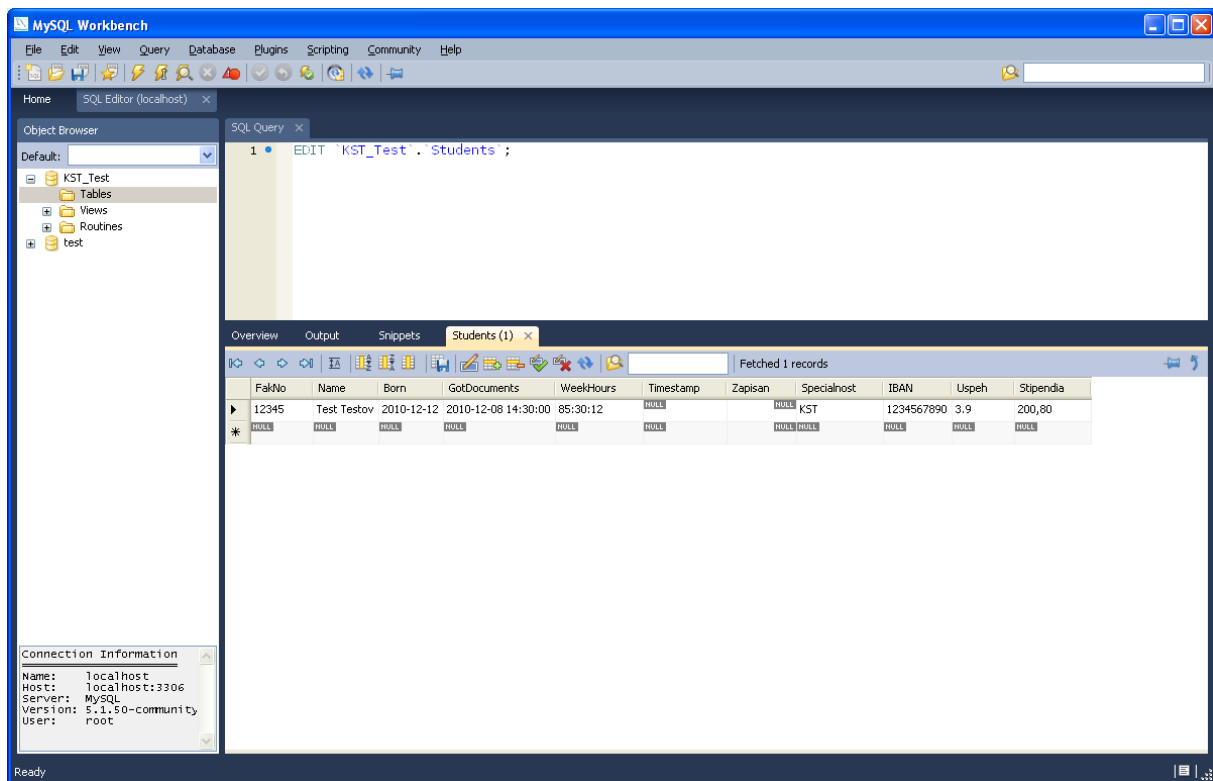
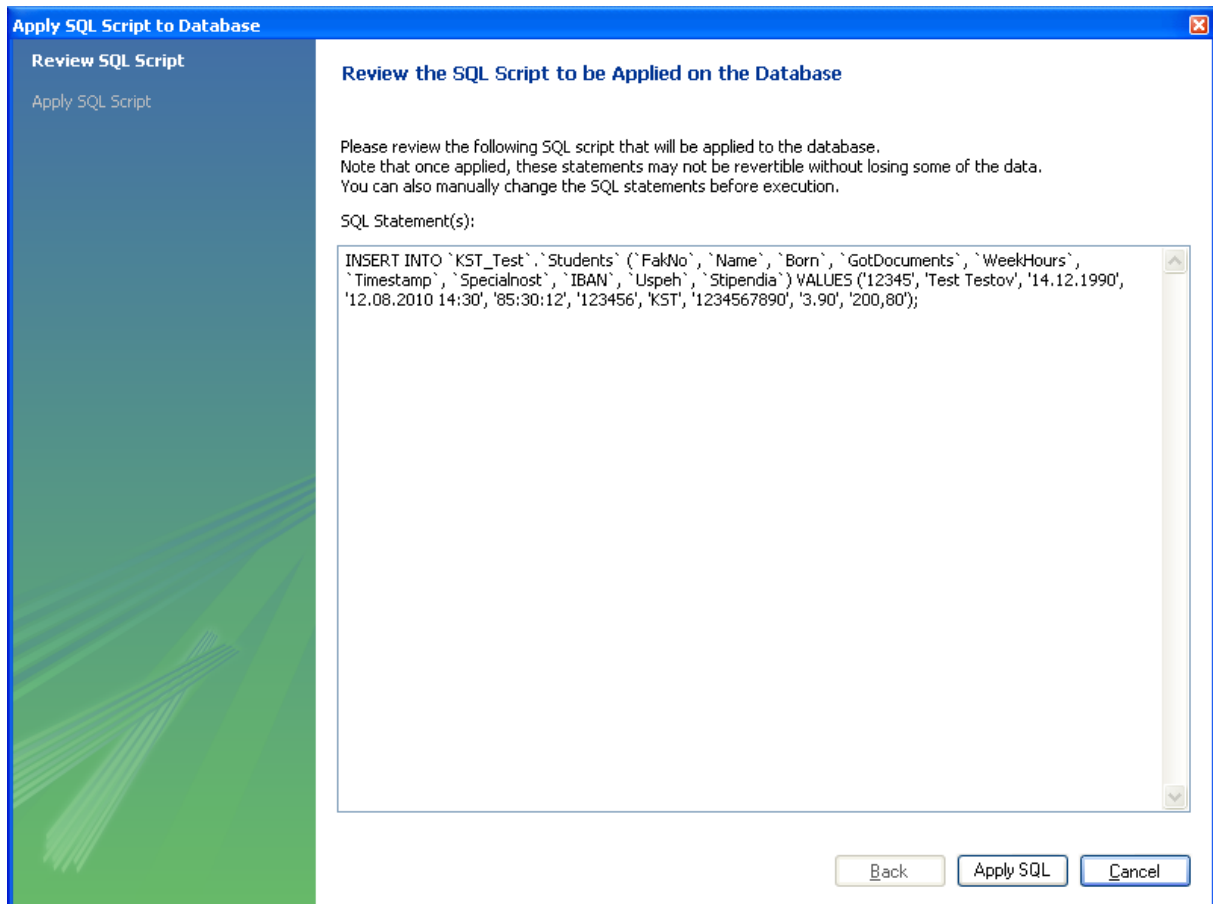
The screenshot displays the MySQL Workbench application window. The top menu bar includes File, Edit, View, Query, Database, Plugins, Scripting, Community, and Help. Below the menu is a toolbar with various icons for file operations, editing, and database management. The main interface is divided into three panes:

- Object Browser (Left):** Shows the database structure for 'KST\_Test'. It includes a 'Default:' dropdown and a tree view with folders for 'Tables', 'Views', 'Routines', and 'test'.
- SQL Query (Top Center):** Contains a single SQL statement: `1 • EDIT KST_Test . Students ;`
- Results (Bottom Center):** Displays the output of the query. The tab is labeled 'Students (1)\*'. The status bar indicates 'Fetched 0 records, inserted 1'. The results table has the following columns: FakNo, Name, Born, GotDocuments, WeekHours, Timestamp, Zapisan, Specialhost, IBAN, Uspeh, and Stipendia. The data row shows: 12345, Test Te..., 14.12..., 12.08.2010 14:30, 85:30:12, 123456, NULL, KST, 123456..., 3.90, 200,80[. Below the data row, there are asterisks and the word 'NULL' under each column header, indicating that the data is not yet committed.

At the bottom left, the 'Connection Information' pane shows the following details:

- Name: localhost
- Host: localhost:3306
- Server: MySQL
- Version: 5.1.50-community
- User: root

The status bar at the very bottom indicates 'SQL Editor Opened.'.





## Задачи

1. Да се създаде нова конекция към локалната машина с име Local Server, потребител root и празна парола.
2. Да се създаде база от данни(схема) с наименование KST\_Test и колагия utf8.
3. Да се създаде таблица Studenti със следната структура:

Studenti (

**FakNo** *int unique, primary key,*

**Ime** *varchar(45),*

**Stipendia** *decimal, 10.2,*

**Prekysnal** *binary,*

**Avtobiografii** *text,*

**Roden** *date,*

**PodalDokumenti** *datetime,*

**Zapisan** *year,*

**Specialnost** *ENUM('KST','KTT','AIUT','E'),*

**UchebenDen** *SET('Ponedelnik', 'Vtornik', 'Sriada', 'Chetvartak', 'Petak', 'Sabota', 'Nedelia'),*

**Uspeh** *decimal 3.2*

)

4. Да се попълнят данни за няколко студента в таблицата, чрез ръчно модифициране на полетата в средата на MySQL Workbench.
5. Да се изтрият създадените бази данни и връзки.