



VILNIUSCODINGSCHOOL

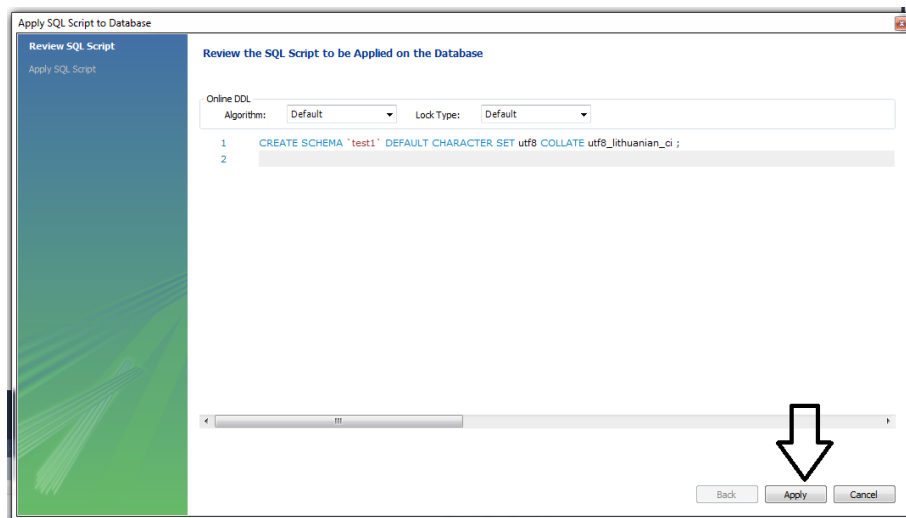
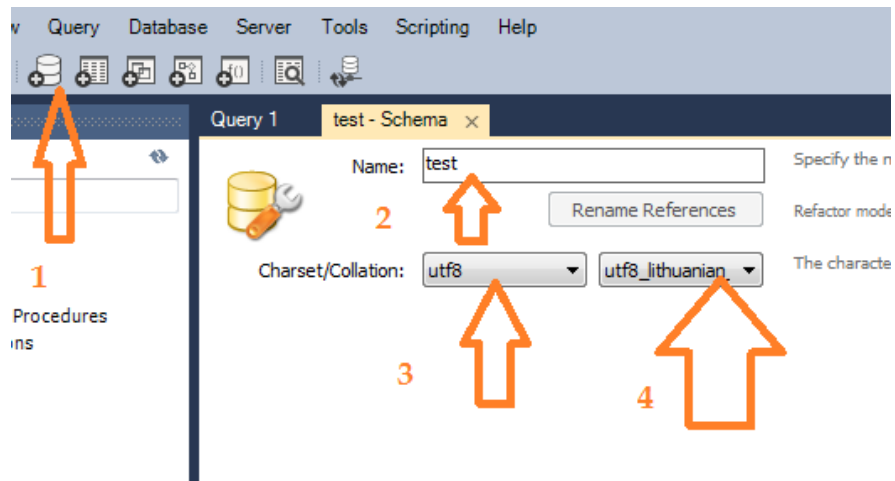
Duomenų Analitika. SQL Dalis



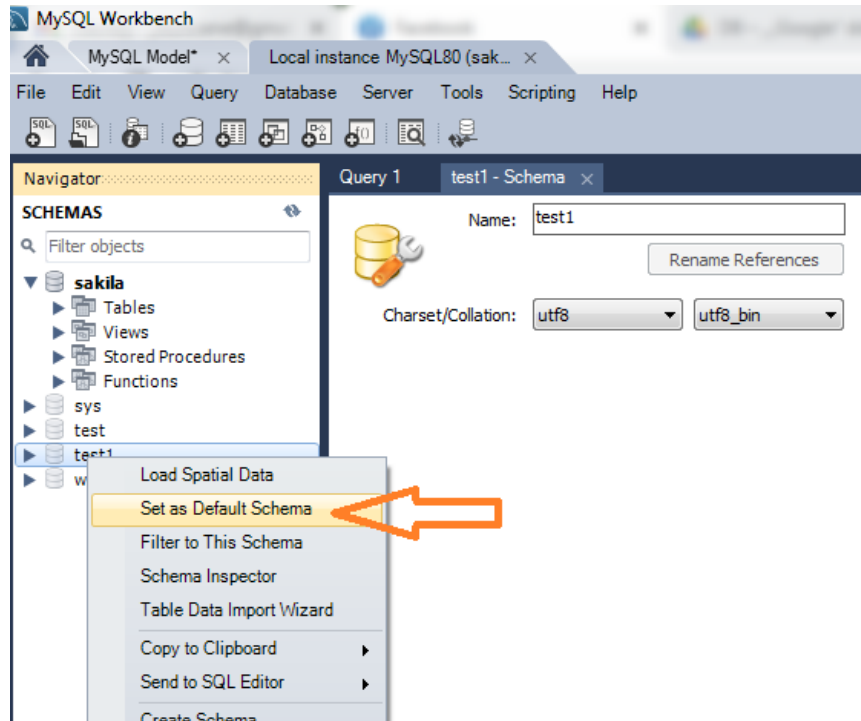
Duomenų analitika su SQL

- Duomenų bazių struktūra
- Duomenų ištraukimas iš duomenų bazių:
 - Ištraukimo sintaksė
 - Duomenų filtravimas
 - Papildomų duomenų susikūrimas
 - Lentelių sujungimas (ryšiai tarp jų)
- Duomenų tipai
- Lentelių sukūrimas ir užpildymas duomenimis

How to Create a MySQL Database with MySQL Workbench



How to Create a MySQL Database with MySQL Workbench



Lentelės Sukūrimas



- Kad nereiktų kiekvieną kartą nurodyti duomenų bazės pavadinimą, script'o pradžioje galima naudoti:

```
USE duomenų_bazės_pavadinimas;
```

- ```
CREATE TABLE lenteles_pavadinimas (
 stulpelio_pavadinimas1 duomenų_tipas apribojimais,
 stulpelio_pavadinimas2 duomenų_tipas apribojimais, ...);
```

- Pavyzdys:

```
CREATE TABLE Klientai (
 id INT NOT NULL PRIMARY KEY
 ,vardas_pavarde VARCHAR(255)
 ,kompanija VARCHAR(100))
```

- Eilės tvarka stulpelio\_pavadinimas1 duomenų\_tipas apribojimais dalyje yra svarbi!

Klientai

| Id | vardas_pavarde        | kompanija         |
|----|-----------------------|-------------------|
| 1  | Justinas Narušis      | Vilniaus Vandenys |
| 2  | Kristina Sakalaitė    | KPMG              |
| 3  | Albinas Juozapavičius | Cisco             |

# Lentelės Apribojimai (*constraints*)



Darbuotojas

- Pristatysiu kelis pagrindinius apribojimus (*constraints* arba *modifiers*) duomenims, kurie yra nurodomi kuriant lenteles:

- **NOT NULL** – nurodo, kad stulpelio reikšmės turi būti ne tuščios (NULL).
- **PRIMARY KEY** – nurodo, kuris lentelės stulpelis bus pirminis raktas. Šios reikšmės yra unikalios ir negali būti NULL.
- **FOREIGN KEY** – nurodo dviejų lentelių sąryšį. Ji galima naudoti tik jau sukurtai lentelei.

| Id | vardas     | pavarde      | amzius | miesto_id |
|----|------------|--------------|--------|-----------|
| 1  | Žygimantas | Kazakevičius | 30     | 6         |
| 2  | Augustas   | Butkus       | 15     | 3         |
| 3  | Mindaugas  | Žukauskas    | 51     | 1         |

- Pvz.:

```
CREATE TABLE Darbuotojas (
 id INT NOT NULL PRIMARY KEY
 ,vardas VARCHAR(55) NOT NULL
 ,pavarde VARCHAR(55) NOT NULL
 ,amzius INT
 ,miesto_id INT
 ,FOREIGN KEY (miesto_id) REFERENCES
 Miestai(miesto_id));
```

# Lentelės Užpildymas Duomenimis



- `INSERT INTO lenteles_pav (stulpelio1_pav, stulpelio 2 _pav,...)  
VALUES (reiksme1, reiksme2,...), (reiksmen, reiksmen+1), ...;`
- Pavyzdys:  

```
INSERT INTO Klientai (id, vardas_pavarde, kompanija)
VALUES (1, 'Justinas Narušis', 'Vilniaus Vandenys'),
 (2, 'Marytė Januškienė', 'Apranga Group');
```
- Pastaba / Priminimas: skaičiai niekada neturi būti rašomi kabutėse, bet tekstas ir datos – būtinai turi būti kabutėse.
- Jei norite pamatyti bet kurios lentelės informaciją, t.y. stulpelių pavadinimus, duomenų tipus, apribojimus ir pan., naudokite užklausą: `DESCRIBE lenteles_pavadinimas.`

# Laikinos Lentelės Sukūrimas 1



- Laikina lentelė nėra saugoma duomenų bazėje, ji tiesiog gyvuoja per sesiją, kada ji būna sukurta ir tik ją nutraukus (atsijungus nuo serverio) ji automatiškai išsitrina. Taigi tokia lentelė yra naudinga tada, kai jos reikia tik tam tikru metu. Ją galima sukurti

```
CREATE TEMPORARY TABLE lenteles_pav (stulpelio_pav duomenų_tipas
apribojimai, ...);
```

- Pavyzdys:

```
CREATE TEMPORARY TABLE Klientai (
 id INT NOT NULL PRIMARY KEY
 ,vardas VARCHAR(55) NOT NULL
 ,pavarde VARCHAR(55) NOT NULL);
```



# Laikinos Lentelės Sukūrimas 2



- Taip pat galima sukurti laikiną lentelę pagal jau esančią lentelę ir jos duomenis:

```
CREATE TEMPORARY TABLE laik_lenteles_pav
SELECT *
FROM lenteles_pav
```

Čia **SELECT** sakinyss gali atrodyti kaip tik norite (naudojant visas jo dalis, tokias kaip **WHERE** ir pan.).

# Duomenų Lentelėse Atnaujinimas



- Turint jau sukurtą ir duomenimis užpildytą lentelę, gali prireikti pakeisti tam tikras reikšmes stulpeliuose, pvz., padarėte klaidą įrašant duomenis. Tai galite padaryti **UPDATE** sakinio pagalba:

```
UPDATE lenteles_pav
SET stulpelio1_pav = reiksme1, stulpelio2_pav = reiksme2,...
WHERE sąlyga(-os);
```

# Duomenų Lentelėse Atnaujinimas. Pavyzdžiai



```
UPDATE Darbuotojas
SET pavarde = 'Zukauskas'
WHERE id = 3;
```

```
UPDATE Darbuotojas
SET amzius = 32, miesto_id = 2
WHERE id = 1;
```

Darbuotojas

| id | vardas     | pavarde      | amzius | miesto_id |
|----|------------|--------------|--------|-----------|
| 1  | Žygimantas | Kazakevičius | 30     | 6         |
| 2  | Augustas   | Butkus       | 15     | 3         |
| 3  | Mindaugas  | Žukauskas    | 51     | 1         |

# Duomenų ir Lentelių Ištrynimas VISAM LAIKUI 1



- Norint ištrinti vieną ar daugiau eilučių iš lentelės (ištrintos eilutės nebegrąžinamos), naudojamas sakiny:

```
DELETE FROM lenteles_pav
WHERE sąlyga (-os);
```

- Pavyzdžiai:

```
DELETE FROM Darbuotojas
WHERE id > 3;
```

- Norint ištrinti visus duomenis iš lentelės, rašome:

```
TRUNCATE TABLE lenteles_pav;
```

# Duomenų ir Lentelių Ištrynimasis VISAM LAIKUI 2



- Norint ištrinti visą lentelę (-es) iš duomenų bazės (tuščią arba su duomenimis; lentelės atstatyti nebeįmanoma), rašome:

```
DROP TABLE lenteles1_pav, lenteles2_pav, ...;
```

- Tai yra labai rimtos funkcijos, kurios gali pridaryti daug žalos duomenų bazėms, todėl dažniausiai norint jas naudoti, Jums turės suteikti edit'inimo teises. Dažniausiai tokių teisių niekas nesuteikia, jei nesate tos duomenų bazės administratorius.

# ALTER TABLE



ALTER TABLE komanda naudojama pridėti/ištrinti/modifikuoti stulpelius lentelėje.

Consider the following table called **People**:

| ID | FirstName | LastName | City        |
|----|-----------|----------|-------------|
| 1  | John      | Smith    | New York    |
| 2  | David     | Williams | Los Angeles |
| 3  | Chloe     | Anderson | Chicago     |

The following SQL code adds a new column named **DateOfBirth**:

```
ALTER TABLE People ADD DateOfBirth
date;
```

Result:

| ID | FirstName | LastName | City        | DateOfBirth |
|----|-----------|----------|-------------|-------------|
| 1  | John      | Smith    | New York    | NULL        |
| 2  | David     | Williams | Los Angeles | NULL        |
| 3  | Chloe     | Anderson | Chicago     | NULL        |


## Dropping

The following SQL code demonstrates how to delete the column named *DateOfBirth* in the *People* table.

```
ALTER TABLE People
DROP COLUMN DateOfBirth;
```

The *People* table will now look like this:

| ID | FirstName | LastName | City        |
|----|-----------|----------|-------------|
| 1  | John      | Smith    | New York    |
| 2  | David     | Williams | Los Angeles |
| 3  | Chloe     | Anderson | Chicago     |

 The column, along with all of its data, will be completely removed from the table.

## Renaming

The ALTER TABLE command is also used to rename columns:

```
ALTER TABLE People
CHANGE FirstName name varchar(100);
```

This query will rename the column called *FirstName* to *name*.

Result:

| ID | name  | LastName | City        |
|----|-------|----------|-------------|
| 1  | John  | Smith    | New York    |
| 2  | David | Williams | Los Angeles |
| 3  | Chloe | Anderson | Chicago     |

## Renaming Tables

You can rename the entire table using the **RENAME** command:

```
RENAME TABLE People TO Users;
```

# Užduotys

- Sukurkite lentelės „customer“ kopiją kaip laikiną lentelę ir išsaugokite ją pavadinimu „Klientai“.
- Lentelėje „klientai“ pakeiskite klientės Linda Williams el.pašto adresą į [linda.williams@gmail.com](mailto:linda.williams@gmail.com).
- Lentelėje „klientai“ pakeiskite adresą kodą iš 10 į 21 klientei Jennifer Davis.
- Iš lentelės „klientai“ pašalinkite klientę Mary Smith.
- Iš lentelės „klientai“ pašalinkite klientus, kurių kodas yra intervale [4, 6].

# Užduotys

- Ištrinkite visus įrašus iš lentelės „klientai“.
- Ištrinkite lentelę „klientai“ visam laikui.
- Sukurkite lentelės „rental“ kopiją kaip ilgalaikę lentelę ir išsaugokite ją pavadinimu „Nuoma“.
- Pridėkite varchar(255) tipo stulpelį „skolininkai“ į lentelę „Nuoma“.



# Užduotys

- Pervadinkite lentelės „rental“ stulpelį „skolininkai“ į „negrąžinusieji“.
- Klientams, kurie dar nėra grąžinę išsinuomoto filmo, stulpelyje „negrąžinusieji“ įrašykite žodį „Taip“. Naudokite lentelę „rental“.
- Iš lentelės „rental“ ištrinkite stulpelį „negrąžinusieji“.