

# Duomenų bazės

---

REKURSINĖS UŽKLAUSOS

# Lentelė „navigation“

---

Sukurkite DB lentelę „navigation“ kuri turėtų šiuos laukus:

id – automatinis skaičius, unsigned

pid – skaičius, unsigned pagal nutilėjimą 0

name – varchar, 16 simbolių

link – varchar, 32 simbolių

# Duomenų patalpinimas

---

Įkelkite pagrindinius meniu punktus į DB:

- Darbuotojai,
- Projektai, projektai.php
- Statistika,

Sukelkite sub punktus prie punkto darbuotojai:

- Darbuotojų sąrašas, darbuotojai\_list.php
- Darbo užkokesčiai, darbuotojai\_uzmokesciai.php

Sukelkite sub punktus prie punkto Statistika:

- Projektų statistika, statistika\_projektai.php
- Darbuotojų statistika, statistika\_darbuotojai.php

# Duomenų paėmimas ir atvaizdavimas

---

Sukurkime f-ją (get navigation) kuri iš DB rekursiškai surinktų visą meniu į masyvą

Sukurkime f-ją kuri atvaizduotų meniu punktus rekursiškai (kaip sąrašą).

# MySQL

---

TRIGGERIAI, EVENT

# MySQL triggeriai gali būti iškviesti:

---

Įvykiais:

- INSERT
- UPDATE
- DELETE

Laikas

- BEFORE
- AFTER

Kiekvienai lentelei, vienam įvykiui ir laikui galime turėti tik po vieną triggerį

# Konstantos OLD , NEW

---

OLD – įrašas kuris buvo prieš atnaujinimą

NEW – įrašas kuris bus po atnaujinimo

INSERT turi tik NEW konstantą

UPDATE turi NEW ir OLD konstantas

DELETE turi tik OLD konstantas

# Trigerio struktūra

---

Tipinis trigerio užrašymas:

```
CREATE TRIGGER `vardas`  
  BEFORE UPDATE ON `lentele`  
  FOR EACH ROW  
    SET NEW.salary = 0
```



# Keleto užklausų trigeriai

---

Norint vienam trigeriui sukurti keletą užklausų reikia naudoti BEGIN ... END struktūrą.

```
CREATE TRIGGER `del` BEFORE DELETE ON `test_1`  
FOR EACH ROW  
BEGIN  
DELETE FROM test_2 WHERE i=OLD.id;  
INSERT INTO la (`val`) VALUES (1);  
END
```

# Užduotis

---

Sukurkime triggerį kuris mėginant ištrinti pareigas tuo pačiu ištrintų ir darbuotojus dirbančius tose pareigose.

# Trigeriai nutraukiantys užklausas

---

Vienas iš būdų nutraukti vykdomą užklausą SIGNAL SQLSTATE. Jos sintaksė:

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Zinute';
```

Žemiau pateiktas trigerys neleis ištrinti įrašo su ID = 3

```
CREATE TRIGGER `del` BEFORE DELETE ON `test_1`  
  FOR EACH ROW BEGIN  
  IF (OLD.id =3) THEN  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Zinute';  
  END IF;  
END
```

# Užduotis

---

Sukurkime triggerį kuris neleisų įterpti įrašo į darbuotojų lentelę su atlyginimu mažesniu už minimumą (380).

# Kintamieji

---

MySQL galima apsibrėžti kintamuosius, jie prasideda @ ženkle. Pvz.:

```
SET @PVM=21;
```

Kintamieji taip pat gali būti gauti iš lentelės:

```
SET @PVM=(SELECT val FROM nustatymai WHERE id='PVM');
```

Arba:

```
SELECT val FROM nustatymai WHERE id='PVM' INTO @PVM;
```

# Užduotis

---

Sukurkime lentelę „nustatymai“ turinčią du stulpelius:

id – char 16 (pirminis)

val – varchar 255

Įterpkime įrašą:

MMA, 380

Pakoreguokime triggerį kad tikrintų jog įterpiamo darbuotojo alga nebūtų mažesnė nei MMA įrašo reikšmė.

# MySQL

---

EVENTS

# MySQL įvykiai

---

Jei trigeriai iškviečiami įvykus kokiam nors įvykiui, tai EVENT SCHEDULER iškviečiami tam tikru laiku.

Jie gali būti iškviečiami rekursiškai, arba nustatytu laiku.

Rekursiškai iškviečiamiems įvykiams apibrėžiamas laikas, pvz.:

EVERY 1 SECOND – kiekvienos minutės pirmąją sekundę.

Įvykių vykdymas aprašomas taip pat kaip ir trigeriai.



# Užduotis

---

Sukurkite įvykį kuris kas vieną minutę suskaičiuotų darbuotojų skaičių ir patalpintų į lentelę nustatymai.