Практическое задание по теме “Оптимизация запросов”

1. Создайте таблицу logs типа Archive. Пусть при каждом создании записи в таблицах users, catalogs и products в таблицу logs помещается время и дата создания записи, название таблицы, идентификатор первичного ключа и содержимое поля name.

**Решение:**

CREATE TABLE IF NOT EXISTS `logs`

(

date timestamp,

table\_name VARCHAR(20),

id VARCHAR(20),

name VARCHAR(20)

) ENGINE = ARCHIVE DEFAULT CHARSET = utf8;

DROP TABLE IF EXISTS products;

CREATE TABLE products (

id SERIAL PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255) COMMENT 'Название',

description TEXT COMMENT 'Описание'

) COMMENT = 'Товарные позиции';

-- \*\*\*\*\*\*\*\*\*\* TRIGGER ON products \*\*\*\*\*\*\*\*\*\*

DROP TRIGGER IF EXISTS log\_products;

delimiter //

CREATE TRIGGER log\_products AFTER insert ON products

FOR EACH ROW

BEGIN

INSERT INTO logs (date, table\_name, id, name)

VALUES (NOW(), 'products', NEW.id, NEW.name);

END//

delimiter ;

-- \*\*\*\*\*\*\*\*\*\* TRIGGER ON users \*\*\*\*\*\*\*\*\*\*

DROP TRIGGER IF EXISTS watchlog\_users;

delimiter //

CREATE TRIGGER watchlog\_users AFTER INSERT ON users

FOR EACH ROW

BEGIN

INSERT INTO logs (created\_at, table\_name, str\_id, name\_value)

VALUES (NOW(), 'users', NEW.id, NEW.name);

END //

delimiter ;

-- \*\*\*\*\*\*\*\*\*\* TRIGGER ON catalogs \*\*\*\*\*\*\*\*\*\*

DROP TRIGGER IF EXISTS watchlog\_catalogs;

delimiter //

CREATE TRIGGER watchlog\_catalogs AFTER INSERT ON catalogs

FOR EACH ROW

BEGIN

INSERT INTO logs (created\_at, table\_name, str\_id, name\_value)

VALUES (NOW(), 'catalogs', NEW.id, NEW.name);

END //

delimiter ;

# Test

-- Новые данные добавлены в таблицы catalogs, users, products

SELECT \* FROM logs;

1. (по желанию) Создайте SQL-запрос, который помещает в таблицу users миллион записей.

Практическое задание по теме “NoSQL”

1. В базе данных Redis подберите коллекцию для подсчета посещений с определенных IP-адресов.
2. При помощи базы данных Redis решите задачу поиска имени пользователя по электронному адресу и наоборот, поиск электронного адреса пользователя по его имени.
3. Организуйте хранение категорий и товарных позиций учебной базы данных shop в СУБД MongoDB.

**Решение:**

**На лекции была, почитала методичку, но задание не делала, простите**