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.

-- создание архивной таблицы

**CREATE** **TABLE** logs (

table\_name **VARCHAR**(60) **NOT** **NULL**,

id **BIGINT**(50) **NOT** **NULL**,

name\_value **VARCHAR**(60) **NOT** **NULL**,

created\_time **DATETIME** **NOT** **NULL**

) ENGINE = ARCHIVE;

-- триггер пользователи

**DROP** **TRIGGER** **IF** **EXISTS** users\_log;

**DELIMITER //**

**CREATE** **TRIGGER** users\_log **AFTER** **INSERT** **ON** users

**FOR** **EACH** **ROW**

**BEGIN**

**INSERT** **INTO** logs (table\_name, id, name\_value, created\_time)

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

**END** //

**DELIMITER ;**

-- триггер каталог

**DROP** **TRIGGER** **IF** **EXISTS** catalogs\_log;

**DELIMITER //**

**CREATE** **TRIGGER** catalogs\_log **AFTER** **INSERT** **ON** catalogs

**FOR** **EACH** **ROW**

**BEGIN**

**INSERT** **INTO** logs (table\_name, id, name\_value, created\_time)

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

**END** //

**DELIMITER ;**

-- триггер продукция

**DROP** **TRIGGER** **IF** **EXISTS** products\_log;

**DELIMITER //**

**CREATE** **TRIGGER** products\_log **AFTER** **INSERT** **ON** products

**FOR** **EACH** **ROW**

**BEGIN**

**INSERT** **INTO** logs (table\_name, id, name\_value, created\_time)

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

**END** //

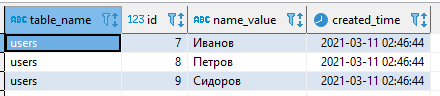
**DELIMITER ;**

**INSERT** **INTO** users (name, birthday\_at)

**VALUES** ('Иванов', '1900-01-01'),

('Петров', '1103-01-01'),

('Сидоров', '1103-01-01');

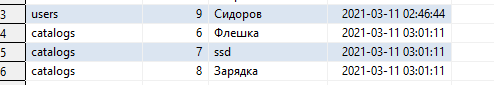


**INSERT** **INTO** catalogs (name)

**VALUES** ('Флешка'),

('ssd'),

('Зарядка');

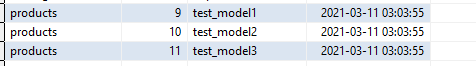


**INSERT** **INTO** products (name, description, price, catalog\_id)

**VALUES** ('test\_model1', 'Тестовое описание1', 4500.00, 8),

('test\_model2', 'Тестовое описание2', 780.00, 9),

('test\_model3', 'Тестовое описание3', 450.00, 10);



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.