Министерство образования Республики Беларусь Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

ОТЧЕТ

Лабораторная работа №3

по теме: «Интернет магазин продажи компьютерной техники различных производителей»

Выполнил: Соловьёв Д.Н. гр.753502

Проверил: Алексеев И.Г.

Минск

БГУИР 2020

**SQL Запросы**

1. Реализовать с использованием триггеров создание backup копии для каждого типа операции (update, delete) с таблицами учетных записей пользователей.

* *Создаем таблицу, в которой будут храниться backup копии для каждого типа операций*

CREATE TABLE `users\_backup` (

`id` smallint(5) unsigned NOT NULL AUTO\_INCREMENT,

`operation\_type` enum('Update','Delete') NOT NULL,

`user\_id` smallint(5) unsigned NOT NULL,

`change\_date` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`login` varchar(45) NOT NULL,

`password` varchar(128) NOT NULL,

`salt` varchar(64) NOT NULL,

`role` enum('Client','Admin','Provider') NOT NULL DEFAULT 'Client',

`status` enum('Active','Blocked') NOT NULL DEFAULT 'Active',

`block\_begin` datetime DEFAULT NULL,

`block\_end` datetime DEFAULT NULL,

`registration\_date` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`id\_name1` smallint(5) unsigned NOT NULL,

`id\_name2` smallint(5) unsigned NOT NULL,

`id\_name3` smallint(5) unsigned NOT NULL,

`id\_personal\_info` mediumint(8) unsigned DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=32 DEFAULT CHARSET=utf8

* 1. Создание backup копии для операции UPDATE
* *Создаём триггер*

CREATE TRIGGER user\_back\_delete\_trigger

BEFORE DELETE ON OnlineStoreDB.users

FOR EACH ROW

INSERT INTO OnlineStoreDB.users\_backup

(operation\_type, user\_id, change\_date, login, password, salt, role, status, block\_begin, block\_end,

registration\_date, id\_name1, id\_name2, id\_name3, id\_personal\_info)

VALUES

('Delete', OLD.id, NOW(), OLD.login, OLD.password, OLD.salt, OLD.role, OLD.status, OLD.block\_begin,

OLD.block\_end, OLD.registration\_date, OLD.id\_name1, OLD.id\_name2, OLD.id\_name3, OLD.id\_personal\_info);

* 1. Создание backup копии для операции DELETE
* *Создаём триггер*

CREATE TRIGGER user\_back\_update\_trigger BEFORE UPDATE ON OnlineStoreDB.users

FOR EACH ROW

INSERT INTO OnlineStoreDB.users\_backup

(operation\_type, user\_id, change\_date, login, password, salt, role, status, block\_begin, block\_end,

registration\_date, id\_name1, id\_name2, id\_name3, id\_personal\_info)

VALUES

('Update', OLD.id, NOW(), OLD.login, OLD.password, OLD.salt, OLD.role, OLD.status, OLD.block\_begin,

OLD.block\_end, OLD.registration\_date, OLD.id\_name1, OLD.id\_name2, OLD.id\_name3, OLD.id\_personal\_info);

1. Реализовать восстановление backup копии для каждого типа операции (update, delete) с таблицами учетных записей пользователей с учетом необходимых блокировок.

* *Получаем информацию о последней операции из таблицы c backup копиями*

SELECT \* FROM users\_backup

ORDER BY change\_date DESC

limit 1;

* 1. Восстановление backup копии операции типа UPDATE
* *Блокируем таблицы*

LOCK TABLE users WRITE, users\_backup READ;

* *Восстанавливаем учетную запись пользователя*

UPDATE users SET

password = '{\*}', salt = '{\*}', role = '{\*}',

status = '{\*}', block\_begin = '{\*}',

block\_end = '{\*}', id\_name1 = {\*},

id\_name2 = {\*}, id\_name3 = {\*},

id\_personal\_info = {\*}

WHERE id = '{\*}'

* *Удаляем данные, которые запились в таблицу backup копий, при восстановления учетной записи ранее*

DELETE FROM credentials\_backup

WHERE user\_id = {\*}

ORDER BY change\_date DESC limit 1

* *Разблокируем таблицы*

UNLOCK TABLES;

* 1. Восстановление backup копии операции типа DELETE
* *Блокируем таблицы*

LOCK TABLE users WRITE, users\_backup READ;

* *Восстанавливаем учетную запись пользователя*

INSERT INTO credentials(id, login, pass, status, role, block\_begin, block\_end,

registration\_date, id\_name1, id\_name2, id\_name3, id\_passport)

VALUES

({\*}, '{\*}', '{\*}', '{\*}', '{\*}', '{\*}', '{\*}', '{\*}', {\*}, {\*}, {\*}, {\*});

* *Разблокируем таблицы*

UNLOCK TABLES;

* *Удаляем из backup данные, которые восстановили*

DELETE FROM credentials\_backup WHERE id = {\*}