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

ОТЧЕТ

Лабораторная работа №2

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) Авторизация

SELECT id FROM users WHERE login = '{\*}' AND password = '{\*}'

SELECT status, role, block\_begin, block\_end,

registration\_date, name1, name2, name3, id\_personal\_info FROM users

INNER JOIN names1 ON users.id\_name1 = names1.id

INNER JOIN names2 ON users.id\_name2 = names2.id

INNER JOIN names3 ON users.id\_name3 = names3.id

WHERE users.id = {\*} AND users.status = 'Active'

2) Управление пользователями

2.1) Создание пользователя

LOCK TABLE names1 WRITE

INSERT IGNORE INTO names1(name1) VALUES (\*)

UNLOCK TABLE

LOCK TABLE names2 WRITE

INSERT IGNORE INTO names1(name1) VALUES (\*)

UNLOCK TABLE

LOCK TABLE names3 WRITE

INSERT IGNORE INTO names1(name1) VALUES (\*)

UNLOCK TABLE

LOCK TABLE users WRITE

INSERT INTO users(login, password, salt, role, status, registration\_date, id\_name1, id\_name2, id\_name3)

VALUES ('{\*}', '{\*}', '{\*}', '{\*}', 'Active', NOW(),

(SELECT id FROM names1 WHERE name1='{\*}'),

(SELECT id FROM names2 WHERE name2='{\*}'),

(SELECT id FROM names3 WHERE name3='{\*}'))

UNLOCK TABLE

2.2) Удаление пользователя

DELETE FROM users WHERE login='{\*}'

2.3) Изменение прав пользователя

UPDATE users SET

role = '{\*}'

WHERE login = '{\*}'

2.4) Блокировка пользователя

UPDATE users SET

block\_begin = NOW(), block\_end = '2038-01-01', status = 'Blocked'

WHERE login = '{\*}'

2.5) Разблокировка пользователя

UPDATE users SET

block\_begin = NULL, block\_end= NULL, status='Active'

WHERE login = '{\*}'

2.6) Блокировка на определенное время с автоматическим снятием

Блокировки

UPDATE users SET

block\_begin = NOW(), block\_end = NOW() + INTERVAL {\*} SECOND ,

status='Blocked'

WHERE login = '{\*}'

DROP EVENT IF EXISTS unlock\_event\_{login}

CREATE EVENT unlock\_event\_{login}

ON SCHEDULE

AT NOW() + INTERVAL {seconds} SECOND

DO

UPDATE users SET

block\_begin = NULL, block\_end= NULL, status='Active'

WHERE login = '{\*}'

2.7) Поиск пользователей по различным критериям

SELECT users.id, login, status, role, block\_begin, block\_end, registration\_date,

name1, name2, name3, id\_personal\_info FROM users

INNER JOIN names1 ON users.id\_name1 = names1.id

INNER JOIN names2 ON users.id\_name2 = names2.id

INNER JOIN names3 ON users.id\_name3 = names3.id

WHERE

name1 like '%{\*}%'

AND name2 like '%{\*}%'

AND name3 like '%{\*}%'

AND role = '{\*}'

AND status = '{\*}'

3) Журналирование

3.1) Журналирование авторизации

CREATE TABLE `auth\_log` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`login` varchar(45) NOT NULL,

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

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

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

PRIMARY KEY (`id`),

UNIQUE KEY `id\_UNIQUE` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=26 DEFAULT CHARSET=utf8

INSERT INTO auth\_log(login, login\_date, status, role)

VALUES ('{\*}', NOW(), '{\*}', '{\*}')

3.2) Журналирование пользователей операции INSERT

CREATE TABLE `users\_insert\_log` (

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

`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=3 DEFAULT CHARSET=utf8

DROP TRIGGER user\_insert\_trigger;

CREATE TRIGGER user\_insert\_trigger

AFTER INSERT ON OnlineStoreDB.users

FOR EACH ROW

INSERT INTO OnlineStoreDB.users\_insert\_log

(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

(NEW.id, NOW(), NEW.login, NEW.password, NEW.salt,

NEW.role, NEW.status, NEW.block\_begin,

NEW.block\_end, NEW.registration\_date, NEW.id\_name1,

NEW.id\_name2, NEW.id\_name3, NEW.id\_personal\_info);

3.3) Журналирование пользователей операции UPDATE

CREATE TABLE `users\_update\_log` (

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

`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=34 DEFAULT CHARSET=utf8

DROP TRIGGER user\_update\_trigger;

CREATE TRIGGER user\_update\_trigger

AFTER UPDATE ON OnlineStoreDB.users

FOR EACH ROW

INSERT INTO OnlineStoreDB.users\_update\_log

(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

(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);

3.3) Журналирование пользователей операции DELETE

CREATE TABLE `users\_delete\_log` (

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

`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 DEFAULT CHARSET=utf8

DROP TRIGGER user\_delete\_trigger;

CREATE TRIGGER user\_delete\_trigger AFTER DELETE ON OnlineStoreDB.users

FOR EACH ROW

INSERT INTO OnlineStoreDB.users\_delete\_log

(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

(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);