МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО в ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

Технологии разработки и защиты баз данных

Тема: «Представления, триггеры, хранимые процедуры и транзакции»

Работу выполнила

Студент гр.4238

Бусов В.Р.

Принял

Преподаватель

Валова П.А.

Казань 2024

**Задание №1. Создание представлений**

CREATE VIEW v1 AS

SELECT clients.person, phone\_number FROM clients;

Данное представление позволяет нам получить краткую информацию для связи с пользователями

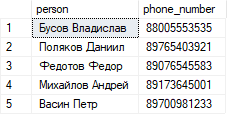


Рисунок 1 – результат вывода представления v1

CREATE VIEW v2 AS

SELECT

categories.title AS c\_title,

services.title AS s\_title,

categories.description as c\_desc

FROM

services JOIN categories ON

services.category = categories.cat\_id

WHERE services.cost <= 5000;

Данное представление позволяет нам получить информацию о связях между категориями и услугами

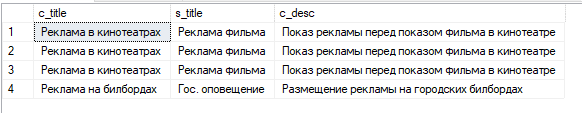


Рисунок 2 – результат вывода представления v2

CREATE VIEW v3 AS

SELECT apps.\*, applications\_history.exec\_id

FROM apps

LEFT JOIN applications\_history ON

apps.app\_id = applications\_history.app\_id;

Данное представление позволяет узнать полную информацию о заявке и о том, какой исполнитель взял ее в работу.

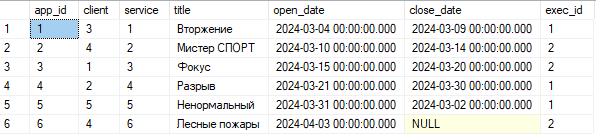


Рисунок 3 – результат представления v3

CREATE VIEW v4 AS

SELECT apps.\*, services.title as s\_title, services.description, services.cost

FROM apps JOIN services

ON apps.service = services.service\_id;

Данное представление дает информацию о заявке и о типе услуг, к которому она принадлежит

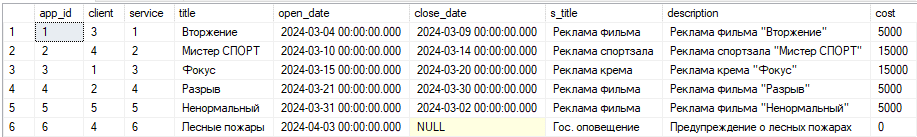


Рисунок 4 – результат представления v4

CREATE VIEW v5 AS

SELECT open\_date, close\_date

FROM apps

WHERE close\_date IS NOT NULL;

Данное представление даёт информацию о том, когда была открыта и закрыта заявка. Заявки, которые сейчас в работе, не учитываются

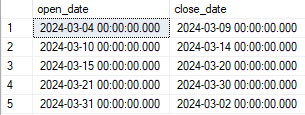


Рисунок 5 – результат представления v5

**Задание №2. Создание триггеров**

*trg\_DeleteCat*

CREATE TRIGGER trg\_DeleteCat

ON categories

AFTER DELETE

AS

BEGIN

DELETE FROM services WHERE category IN (SELECT cat\_id FROM deleted);

END

Данный триггер удаляет все типы услуг, которые относятся к удаленной категории

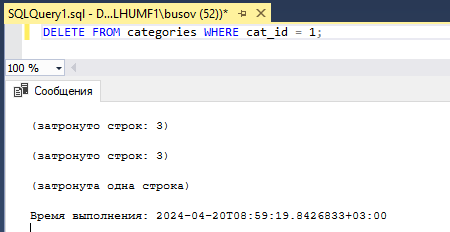


Рисунок 6 – результат работы триггера

*trg\_ChangeAppsOnCatDelete*

CREATE TRIGGER trg\_ChangeAppsOnCatDelete

ON services

AFTER DELETE

AS

BEGIN

UPDATE apps SET service = NULL WHERE service IN (SELECT service\_id FROM deleted);

END

Данный триггер заменяет тип услуги на Null у каждой заявки, которая относится к ней, после удаления услуги из таблицы services

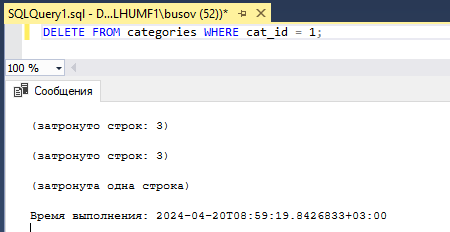


Рисунок 7 – результат работы триггера.

*trg\_ServiceUpdate*

CREATE TRIGGER trg\_ServiceUpdate

ON apps

AFTER DELETE

AS

BEGIN

DELETE FROM applications\_history

WHERE app\_id IN (SELECT app\_id FROM deleted);

END

Данный триггер, при удалении заявки, удаляет ее из истории выполнения

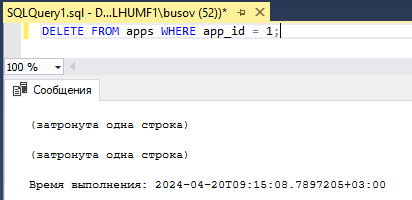


Рисунок 8 – результат работы триггера

*trg\_ExecDelete*

CREATE TRIGGER trg\_ExecDelete

ON executors

AFTER DELETE

AS

BEGIN

DELETE FROM applications\_history

WHERE exec\_id IN (SELECT exec\_id FROM deleted);

END

Данный триггер удаляет из истории все заявки, над которыми работал исполнитель, которого удалили

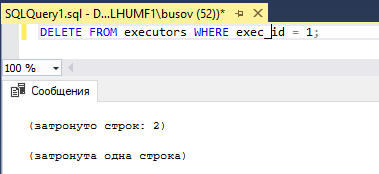


Рисунок 9 – результат работы триггера

*trg\_ClientsDelete*

CREATE TRIGGER trg\_ClientsDelete

ON clients

AFTER DELETE

AS

BEGIN

DELETE FROM apps

WHERE client IN (SELECT client\_id FROM deleted);

END

Данный триггер удаляет заявки клиента, который был удален

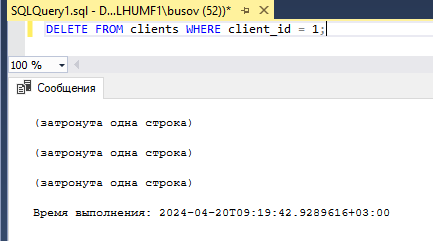


Рисунок 10 – результат работы триггера

**Задание №3. Создание процедур**

CREATE PROCEDURE p1 AS

SELECT phone\_number, person FROM clients;

Данная процедура позволяет нам получить краткую информацию для связи с пользователями

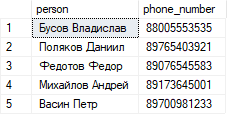


Рисунок 11 – результат вывода процедуры p1

CREATE PROCEDURE p2 AS

BEGIN

SELECT

categories.title AS c\_title,

services.title AS s\_title,

categories.description as c\_desc

FROM

services JOIN categories ON

services.category = categories.cat\_id

WHERE services.cost <= 5000;

END

Данная процедура позволяет нам получить информацию о связях между категориями и услугами

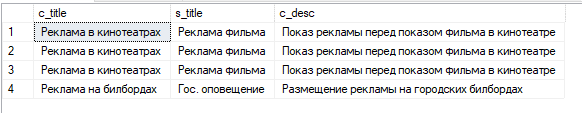


Рисунок 12 – результат вывода процедуры p2

CREATE PROCEDURE p3 AS

BEGIN

SELECT apps.\*, applications\_history.exec\_id

FROM apps

LEFT JOIN applications\_history ON

apps.app\_id = applications\_history.app\_id;

END

Данная процедура позволяет узнать полную информацию о заявке и о том, какой исполнитель взял ее в работу.

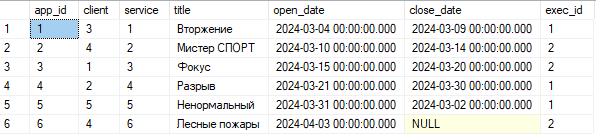


Рисунок 13 – результат процедуры p3

CREATE PROCEDURE p4 AS

BEGIN

SELECT apps.\*, services.title as s\_title, services.description, services.cost

FROM apps JOIN services

ON apps.service = services.service\_id;

END

Данная процедура дает информацию о заявке и о типе услуг, к которому она принадлежит

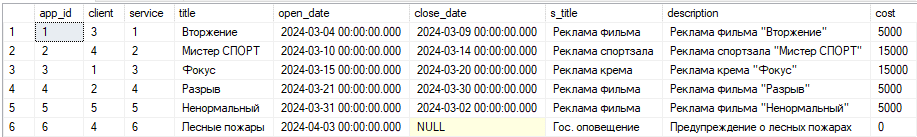


Рисунок 14 – результат процедуры p4

CREATE PROCEDURE p5 AS

BEGIN

SELECT open\_date, close\_date

FROM apps

WHERE close\_date IS NOT NULL;

END

Данная процедура даёт информацию о том, когда была открыта и закрыта заявка. Заявки, которые сейчас в работе, не учитываются

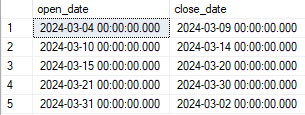


Рисунок 15 – результат процедуры p5