# Базы данных

**Практическая работа № 11 «Языки запросов – общая характеристика.»**

## **Цель:**

## Освоение возможностей однотабличного оператора SELECT.

## **Задание.**

## Выполнение запросов SELECT с вложенными запросами:

## Описание запросов SELECT, осуществляющих сравнение с результатом вложенного запроса;

## Описание запросов SELECT, осуществляющих проверку на принадлежность результатам вложенного запроса;

## Описание запросов SELECT, осуществляющих проверку на существование;

## Описание запросов SELECT, осуществляющих многократное сравнение.

## Выполнение многотабличных запросов SELECT с внутренним объединением таблиц (по каждому из перечисленных ниже видов оператора SELECT в отчете привести выражение запроса):

## Внутреннее объединение таблиц на основе точного совпадения значений двух и более столбцов;

## Внутреннее объединение таблиц с использованием отношения предок-потомок.

## Выполнение многотабличных запросов SELECT с внешним объединением таблиц (по каждому из перечисленных ниже видов оператора SELECT в отчете привести выражение запроса):

## Выполнение многотабличных запросов SELECT с левым внешним объединением;

## Выполнение многотабличных запросов SELECT с правым внешним объединением;

## Выполнение многотабличных запросов SELECT с полным внешним объединением.

## **Ход выполнения.**

**Создание таблицы "Клиенты"**

CREATE TABLE Клиенты (

id INTEGER PRIMARY KEY AUTOINCREMENT,

имя TEXT NOT NULL,

телефон TEXT

);

INSERT INTO Клиенты (имя, телефон) VALUES

('Иванов Иван', '123-456-7890'),

('Петрова Анна', '987-654-3210'),

('Сидоров Сергей', NULL),

('Кузнецова Елена', '456-789-0123'),

('Новиков Алексей', '321-654-9870');

**Создание таблицы "Услуги"**

CREATE TABLE Услуги (

id INTEGER PRIMARY KEY AUTOINCREMENT,

название TEXT NOT NULL,

стоимость REAL NOT NULL

);

INSERT INTO Услуги (название, стоимость) VALUES

('Аренда велосипеда', 500.00),

('Аренда самоката', 300.00),

('Аренда скейта', 400.00),

('Аренда роликов', 200.00),

('Аренда электросамоката', 800.00);

**Создание таблицы "Прокат"**

CREATE TABLE Прокат (

id INTEGER PRIMARY KEY AUTOINCREMENT,

клиент\_id INTEGER NOT NULL,

услуга\_id INTEGER NOT NULL,

дата\_начала TEXT NOT NULL,

дата\_окончания TEXT,

FOREIGN KEY (клиент\_id) REFERENCES Клиенты(id),

FOREIGN KEY (услуга\_id) REFERENCES Услуги(id)

);

INSERT INTO Прокат (клиент\_id, услуга\_id, дата\_начала, дата\_окончания) VALUES

(1, 1, '2024-11-01', '2024-11-02'),

(2, 2, '2024-11-05', '2024-11-05'),

(3, 3, '2024-11-10', NULL),

(4, 1, '2024-11-15', '2024-11-16'),

(5, 4, '2024-11-20', NULL);

**Запросы SELECT с вложенными запросами:**

**Сравнение с результатом вложенного запроса**

SELECT имя, телефон

FROM Клиенты

WHERE id = (SELECT клиент\_id FROM Прокат WHERE услуга\_id = 3);

**Проверка на принадлежность результатам вложенного запроса**

SELECT имя

FROM Клиенты

WHERE id IN (SELECT клиент\_id FROM Прокат WHERE услуга\_id IN (1, 2));

**Проверка на существование**

SELECT имя

FROM Клиенты

WHERE EXISTS (

SELECT 1

FROM Прокат

WHERE Клиенты.id = Прокат.клиент\_id AND Прокат.дата\_окончания IS NULL

);

**Многократное сравнение**

SELECT имя

FROM Клиенты

WHERE id IN (

SELECT клиент\_id

FROM Прокат

WHERE услуга\_id = (SELECT id FROM Услуги WHERE название = 'Аренда велосипеда')

AND дата\_окончания IS NULL

);

**Многотабличные запросы SELECT с внутренним объединением:**

**Объединение на основе точного совпадения значений двух столбцов**

SELECT Клиенты.имя, Услуги.название, Прокат.дата\_начала

FROM Прокат

INNER JOIN Клиенты ON Прокат.клиент\_id = Клиенты.id

INNER JOIN Услуги ON Прокат.услуга\_id = Услуги.id;

**Объединение на основе отношения "предок-потомок"**

SELECT Клиенты.имя, Услуги.название, Прокат.дата\_начала

FROM Прокат

INNER JOIN Клиенты ON Прокат.клиент\_id = Клиенты.id

INNER JOIN Услуги ON Прокат.услуга\_id = Услуги.id

WHERE Услуги.стоимость > 300;

**Многотабличные запросы SELECT с внешним объединением**

**Левое внешнее объединение**

SELECT Клиенты.имя, Услуги.название, Прокат.дата\_начала

FROM Клиенты

LEFT JOIN Прокат ON Клиенты.id = Прокат.клиент\_id

LEFT JOIN Услуги ON Прокат.услуга\_id = Услуги.id;

**Правое внешнее объединение** можно реализовать с помощью изменения порядка таблиц:

SELECT Клиенты.имя, Услуги.название, Прокат.дата\_начала

FROM Услуги

LEFT JOIN Прокат ON Услуги.id = Прокат.услуга\_id

LEFT JOIN Клиенты ON Прокат.клиент\_id = Клиенты.id;

**Полное внешнее объединение** можно смоделировать через объединение LEFT JOIN и RIGHT JOIN:

SELECT Клиенты.имя, Услуги.название, Прокат.дата\_начала

FROM Клиенты

LEFT JOIN Прокат ON Клиенты.id = Прокат.клиент\_id

LEFT JOIN Услуги ON Прокат.услуга\_id = Услуги.id

UNION

SELECT Клиенты.имя, Услуги.название, Прокат.дата\_начала

FROM Услуги

LEFT JOIN Прокат ON Услуги.id = Прокат.услуга\_id

LEFT JOIN Клиенты ON Прокат.клиент\_id = Клиенты.id;

Код:

