# Базы данных

**Практическая работа № 10 «Организация ввода данных в базу данных.»**

## **Цель:**

## Освоение возможностей однотабличного оператора SELECT.

## **Задание.**

## Выполнение запросов выборки данных из таблиц с помощью оператора SELECT (по каждому из перечисленных ниже видов оператора SELECT в отчете привести выражение запроса):

## Выполнение оператора SELECT с предложением WHERE, в котором сравниваются значения двух выражений;

## Выполнение оператора SELECT с предложением WHERE, в котором осуществляется проверка на принадлежность диапазону;

## Выполнение оператора SELECT с предложением WHERE, в котором осуществляется проверка на членство в множестве;

## Выполнение оператора SELECT с предложением WHERE, в котором осуществляется проверка на соответствие шаблону;

## Выполнение оператора SELECT с предложением WHERE, в котором осуществляется проверка проверки на значение NULL.

## Выполнение запросов SELECT с агрегатными функциями:

## Выполнение агрегатной функции AVG();

## Выполнение агрегатной функции SUM();

## Выполнение агрегатной функции MIN();

## Выполнение агрегатной функции MAX();

## Выполнение агрегатной функции COUNT().

## Выполнение запросов SELECT, содержащие агрегатные функции, использующие операцию группировки:

## Выполнение запросов с группировкой GROUP BY;

## Выполнение запросов с группировкой с предложением HAVING;

## Выполнение запросов с группировкой с предложением ORDER BY;

## **Ход выполнения.**

**Создание базы данных и таблицы**

CREATE TABLE Прокат\_ПР10 (

id INTEGER PRIMARY KEY AUTOINCREMENT,

клиент TEXT NOT NULL,

услуга TEXT NOT NULL,

стоимость REAL NOT NULL,

дата\_начала TEXT NOT NULL,

дата\_окончания TEXT

);

*-- Наполнение таблицы данными*

INSERT INTO Прокат\_ПР10 (клиент, услуга, стоимость, дата\_начала, дата\_окончания) VALUES

('Иванов Иван', 'Аренда велосипеда', 500.00, '2024-11-01', '2024-11-02'),

('Петрова Анна', 'Аренда самоката', 300.00, '2024-11-05', '2024-11-05'),

('Сидоров Сергей', 'Аренда скейта', 400.00, '2024-11-10', NULL),

('Кузнецова Елена', 'Аренда велосипеда', 600.00, '2024-11-15', '2024-11-16'),

('Новиков Алексей', 'Аренда роликов', 200.00, '2024-11-20', NULL);

**1. SELECT с WHERE, сравниваются значения двух выражений**

SELECT \*

FROM Прокат\_ПР10

WHERE стоимость = 500.00;

**2. SELECT с WHERE, проверка на принадлежность диапазону**

SELECT \*

FROM Прокат\_ПР10

WHERE стоимость BETWEEN 300 AND 600;

**3. SELECT с WHERE, проверка на членство во множестве**

SELECT \*

FROM Прокат\_ПР10

WHERE услуга IN ('Аренда велосипеда', 'Аренда скейта');

**4. SELECT с WHERE, проверка на соответствие шаблону**

SELECT \*

FROM Прокат\_ПР10

WHERE клиент LIKE 'Иван%';

**5. SELECT с WHERE, проверка на значение NULL**

SELECT \*

FROM Прокат\_ПР10

WHERE дата\_окончания IS NULL;

**Выполнение запросов SELECT с агрегатными функциями:**

**6. AVG()**

SELECT AVG(стоимость) AS Средняя\_стоимость

FROM Прокат\_ПР10;

**7. SUM()**

SELECT SUM(стоимость) AS Общая\_стоимость

FROM Прокат\_ПР10;

**8. MIN()**

SELECT MIN(стоимость) AS Минимальная\_стоимость

FROM Прокат\_ПР10;

**9. MAX()**

SELECT MAX(стоимость) AS Максимальная\_стоимость

FROM Прокат\_ПР10;

**10. COUNT()**

SELECT COUNT(\*) AS Количество\_записей

FROM Прокат\_ПР10;

**Запросы SELECT с группировкой:**

**11. Группировка GROUP BY**

SELECT услуга, COUNT(\*) AS Количество\_услуг

FROM Прокат\_ПР10

GROUP BY услуга;

**12. Группировка с HAVING**

SELECT услуга, SUM(стоимость) AS Общая\_стоимость

FROM Прокат\_ПР10

GROUP BY услуга

HAVING Общая\_стоимость > 500;

**13. Группировка с ORDER BY**

SELECT услуга, AVG(стоимость) AS Средняя\_стоимость

FROM Прокат\_ПР10

GROUP BY услуга

ORDER BY Средняя\_стоимость DESC;

Краткие выводы:

* SQL позволяет выполнять разнообразные выборки данных с использованием условий фильтрации (WHERE), проверок диапазонов, множеств, шаблонов, а также на наличие NULL значений.
* Агрегатные функции, такие как AVG, SUM, MIN, MAX, и COUNT, предоставляют полезные инструменты для анализа данных.
* Группировка данных (GROUP BY) и использование условий фильтрации на группах (HAVING) помогают эффективно агрегировать данные и получать нужные значения.
* Упорядочивание данных с помощью ORDER BY делает вывод более наглядным и удобным для анализа.

**Код:**