ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ

ГОРОДА МОСКВЫ

Государственное бюджетное профессиональное

образовательное учреждение города Москвы

«Колледж малого бизнеса № 4»

(ГБПОУ КМБ № 4)

ЛАБОРАТОРНАЯ РАБОТА: отчет по лабораторной работе

Специальность: 09.02.07 Информационные системы и программирование
Форма обучения: очная
Студент(ка): Белов Кирилл Валерьевич
Группа: ИПО-21.24
Руководитель: Рыбаков Александр Сергеевич

Отчётная работа защищена с оценкой «___» _____

Москва, 2025 г.

Оглавление

Глоссарий по SQL 3
Лабораторная работа № 2 6
1. Цель работы 6
ПРИМЕРЫ 7
4. Задание 10
Лабораторная работа № 3 14
1. Цель работы 15
ПРИМЕРЫ 15
4 задание 19
Лабораторная работа № 4 24
1. Цель работы 24
ПРИМЕРЫ 25
4 Задание 33
Лабораторная работа № 5 38
1. Цель работы 38
ПРИМЕРЫ 39
4 задание 44
Лабораторная работа № 6 48
1. Цель работы 48
ПРИМЕРЫ 48
4 задание 54
Лабораторная работа № 7 58
1. Цель работы 58
ПРИМЕРЫ 58
4 задание 60
Лабораторная работа № 8 62
1. Цель работы 62
ПРИМЕРЫ 62
4 задания 68
Лабораторная работа № 9 74
1. Цель работы 74
ПРИМЕРЫ 74
4 задание 76
Лабораторная работа № 10 78

1. Цель работы 78
ПРИМЕРЫ 78
4 задание 81

Глоссарий по SQL

1. Команды выборки данных

SELECT - выбрать данные из таблицы

DISTINCT - выбрать только уникальные значения

TOP - выбрать первые N строк

AS - дать столбцу или таблице псевдоним

2. Фильтрация и сортировка

WHERE - отфильтровать строки по условию

ORDER BY - отсортировать результат

LIKE - поиск по шаблону

IN - проверить вхождение в список

BETWEEN - выбрать значения в диапазоне

IS NULL - проверить на пустое значение

3. Агрегатные функции

COUNT() - посчитать количество строк

SUM() - вычислить сумму

AVG() - вычислить среднее значение

MIN() - найти минимальное значение

MAX() - найти максимальное значение

4. Группировка

GROUP BY - сгруппировать строки

HAVING - отфильтровать группы

5. Соединение таблиц

JOIN - соединить таблицы

lNNER JOIN - внутреннее соединение (только совпадающие строки)

LEFT JOIN - левое соединение (все строки из левой таблицы)

RIGHT JOIN - правое соединение (все строки из правой таблицы)

FULL JOIN - полное соединение

CROSS JOIN - декартово произведение

6. Объединение запросов

UNION - объединить результаты запросов (уникальные значения)

UNION ALL - объединить результаты (все значения)

7. Подзапросы

Подзапрос в WHERE - фильтрация по результату другого запроса

Подзапрос в SELECT - вычисление значений

EXISTS - проверить существование строк

ANY / ALL - сравнение со всеми/любым значением подзапроса

8. Функции работы с данными

ROUND() - округление чисел

CEILING() / FLOOR() - округление вверх/вниз

UPPER() / LOWER() - преобразование регистра

LEN() - длина строки

SUBSTRING() - часть строки

REPLACE() - замена подстроки

CONVERT() / CAST() - преобразование типа данных

GETDATE() - текущая дата и время

DATEADD() - добавить интервал к дате

DATEDIFF() - разница между датами

9. Создание и изменение структуры (DDL)

CREATE TABLE - создать таблицу

ALTER TABLE - изменить таблицу

ADD COLUMN - добавить столбец

DROP COLUMN - удалить столбец

DROP TABLE - удалить таблицу

PRIMARY KEY - задать первичный ключ

FOREIGN KEY - задать внешний ключ

NOT NULL - запретить пустые значения

DEFAULT - значение по умолчанию

10. Работа с данными (DML)

INSERT INTO - добавить новые строки

UPDATE - изменить существующие строки

DELETE - удалить строки

VALUES - указать значения для вставки

11. Другие команды

SET - установить значение

FROM - указать таблицу для выборки

CASE - условное выражение

WHEN / THEN / ELSE - условия в CASE

END - завершение блока CASE

Лабораторная работа № 2

Команда SELECT

1. Цель работы

1. Изучение основ выборки данных.
2. Изучение конструкции DISTINCT | ALL.
3. Изучение сортировки данных.
4. Изучение конструкции TOP.
5. Изучение конструкции OFFSET и FETCH.

ПРИМЕРЫ

1

```sql
SELECT
        *
FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звани
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	медицина	1939
2	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	историк	1913
3	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
4	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
5	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910
6	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	геохимик	1908

Запрос успешно выполнен. (localdb)\MSSQL

2

Запрос 2.sql ..._22\User (51))

```sql
SELECT
        ФИО, Дата_рождения
FROM
        Академики
```

3

```sql
SELECT
 'Академик ' + ФИО + ', специализация: ' + Специализация AS Информация  FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Информация
1	Академик Аничков Николай Николаевич, специализаци...
2	Академик Бартольд Василий Владимирович, специали...
3	Академик Белопольский Аристарх Аполлонович, специ...
4	Академик Бородин Иван Парфеньевич, специализация:...
5	Академик Вальден Павел Иванович, специализация: хи...
6	Академик Вернадский Владимир Иванович, специализа...

Запрос успешно выполнен. (localdb)\MS

4

```sql
SELECT
        ФИО
        ,[Через год] = Год_присвоения_звания + 1
FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Через год
1	Аничков Николай Николаевич	1940
2	Бартольд Василий Владимирович	1914
3	Белопольский Аристарх Аполлонович	1904
4	Бородин Иван Парфеньевич	1903
5	Вальден Павел Иванович	1911
6	Вернадский Владимир Иванович	1909

Запрос успешно выполнен.

5

```sql
SELECT DISTINCT
    Специализация  FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Специализация
1	астрофизик
2	ботаник
3	геолог
4	геохимик
5	зоолог
6	историк

Запрос успешно выполнен.

6

```sql
SELECT
        *
FROM
        Академик
ORDER BY
        Год_присвоения_звания
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Карпинский Александр Петрович	1847-01-07 00:00:00.000	геолог	1889
2	Соболевский Алексей Иванович	1857-01-07 00:00:00.000	лингвист	1900
3	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
4	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
5	Ольденбург Сергей Фёдорович	1863-09-26 00:00:00.000	историк	1903
6	Коковцов Павел Константинович	1861-07-01 00:00:00.000	историк	1906

Запрос успешно выполнен. (localdb)\MSSQLLocal

7

```sql
SELECT
        *
FROM
        Академик
ORDER BY
        Специализация DESC
,ФИО ASC
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910
2	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	химик	1916
3	Курнаков Николай Семёнович	1860-12-06 00:00:00.000	химик	1913
4	Истрин Василий Михайлович	1865-02-22 00:00:00.000	филолог	1907
5	Перетц Владимир Николаевич	1870-01-31 00:00:00.000	филолог	1914
6	Павлов Иван Петрович	1849-09-26 00:00:00.000	физиолог	1907

Запрос успешно выполнен. (localdb)\MSS

8

```sql
SELECT TOP 2
        *
FROM
        Академик
ORDER BY
        ФИО ASC
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	медицина	1939
2	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	историк	1913

Запрос успешно выполнен. (localdb)\MSS

9

```sql
SELECT TOP 30 PERCENT
        *
FROM
        Академик
ORDER BY
        Год_присвоения_звания
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Карпинский Александр Петрович	1847-01-07 00:00:00.000	геолог	1889
2	Соболевский Алексей Иванович	1857-01-07 00:00:00.000	лингвист	1900
3	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
4	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
5	Ольденбург Сергей Фёдорович	1863-09-26 00:00:00.000	историк	1903
6	Коковцов Павел Константинович	1861-07-01 00:00:00.000	историк	1906

Запрос успешно выполнен. (localdb)\MSSQLLocalD

10

```sql
SELECT TOP 4 WITH TIES
        *
FROM
        Академик
ORDER BY
        Год_присвоения_звания
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Карпинский Александр Петрович	1847-01-07 00:00:00.000	геолог	1889
2	Соболевский Алексей Иванович	1857-01-07 00:00:00.000	лингвист	1900
3	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
4	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
5	Ольденбург Сергей Федорович	1863-09-26 00:00:00.000	историк	1903

11

```sql
SELECT
        *
FROM
        Академик
ORDER BY
        ФИО
        OFFSET 2 ROWS
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звани
1	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
2	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
3	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910
4	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	геохимик	1908
5	Виноградов Павел Гаврилович	1854-11-30 00:00:00.000	историк	1914
6	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	химик	1916

Запрос успешно выполнен. (localdb)\MSSQL

12

```sql
SELECT
        *
FROM
        Академик
ORDER BY
        ФИО
        OFFSET 2 ROWS
        FETCH NEXT 8 ROWS ONLY
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
2	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
3	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910
4	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	геохимик	1908
5	Виноградов Павел Гаврилович	1854-11-30 00:00:00.000	историк	1914
6	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	химик	1916

Запрос успешно выполнен. (localdb)\MSSQLLoc

4. Задание

1

```sql
SELECT ФИО,Специализация, Дата_рождения
FROM Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Специализация	Дата_рождения
1	Аничков Николай Николаевич	медицина	1885-11-03 00:00:00.000
2	Бартольд Василий Владимирович	историк	1869-11-15 00:00:00.000
3	Белопольский Аристарх Аполлонович	астрофизик	1854-07-13 00:00:00.000
4	Бородин Иван Парфеньевич	ботаник	1847-01-30 00:00:00.000
5	Вальден Павел Иванович	химик-технолог	1863-07-26 00:00:00.000
6	Вернадский Владимир Иванович	геохимик	1863-03-12 00:00:00.000

Запрос успешно выполнен.

2

```sql
SELECT
CONCAT(ФИО + N'О присвоении звания', Год_присвоения_звания) AS "О присвоении звания"
FROM Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	О присвоении звания
1	Аничков Николай НиколаевичО присвоении звания1939
2	Бартольд Василий ВладимировичО присвоении звания...
3	Белопольский Аристарх АполлоновичО присвоении зва...
4	Бородин Иван ПарфеньевичО присвоении звания1902
5	Вальден Павел ИвановичО присвоении звания1910
6	Вернадский Владимир ИвановичО присвоении звания19...

Запрос успешно выполнен. (localdb)\MSSQLLocall

3

```sql
SELECT
    ФИО,[Через 5 лет] = Год_присвоения_звания +5
FROM Академик;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Через 5 лет
1	Аничков Николай Николаевич	1944
2	Бартольд Василий Владимирович	1918
3	Белопольский Аристарх Аполлонович	1908
4	Бородин Иван Парфеньевич	1907
5	Вальден Павел Иванович	1915
6	Вернадский Владимир Иванович	1913

Запрос успешно выполнен.

4

```sql
SELECT DISTINCT Год_присвоения_звания
FROM Академик;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Год_присвоения_звания
1	1889
2	1900
3	1902
4	1903
5	1906
6	1907

Запрос успешно выполнен.

5

```sql
SELECT *
FROM Академик
ORDER BY Дата_рождения DESC;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	медицина	1939
2	Перетц Владимир Николаевич	1870-01-31 00:00:00.000	филолог	1914
3	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	историк	1913
4	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	химик	1916
5	Истрин Василий Михайлович	1865-02-22 00:00:00.000	филолог	1907
6	Марр Николай Яковлевич	1865-01-06 00:00:00.000	лингвист	1912

Запрос успешно выполнен. (localdb)\MSSC

6

```sql
SELECT *
FROM Академик
ORDER BY Специализация DESC, Год_присвоения_звания DESC, ФИО ASC;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910
2	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	химик	1916
3	Курнаков Николай Семёнович	1860-12-06 00:00:00.000	химик	1913
4	Перетц Владимир Николаевич	1870-01-31 00:00:00.000	филолог	1914
5	Истрин Василий Михайлович	1865-02-22 00:00:00.000	филолог	1907
6	Павлов Иван Петрович	1849-09-26 00:00:00.000	физиолог	1907

Запрос успешно выполнен. (localdb)\MS

7

```
SELECT TOP 1 *
FROM Академик
ORDER BY ФИО DESC;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Стеклов Владимир Андреевич	1864-01-09 00:00:00.000	математик	1912

Запрос успешно выполнен. (localdb)\M

8

```
SELECT TOP 1 ФИО
FROM Академик
ORDER BY Год_присвоения_звания ASC;
```

81 % 3 0

Подключено. (1/1)

9

```
SELECT TOP 10 PERCENT *
FROM Академик
ORDER BY ФИО ASC;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	медицина	1939
2	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	историк	1913

Запрос успешно выполнен. (localdb)\MSS

10

```
SELECT *
FROM Академик
WHERE Год_присвоения_звания IN (
    SELECT DISTINCT TOP 5 Год_присвоения_звания
    FROM Академик
    ORDER BY Год_присвоения_звания ASC
)
ORDER BY Год_присвоения_звания ASC;
```

81 % Проблемы не найдены.

Результаты Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Карпинский Александр Петрович	1847-01-07 00:00:00.000	геолог	1889
2	Соболевский Алексей Иванович	1857-01-07 00:00:00.000	лингвист	1900
3	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
4	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
5	Ольденбург Сергей Фёдорович	1863-09-26 00:00:00.000	историк	1903
6	Коковцов Павел Константинович	1861-07-01 00:00:00.000	историк	1906

Запрос успешно выполнен. (localdb)\MSSQLL

11

```
SELECT *
FROM Академик
ORDER BY Дата_рождения ASC
OFFSET 9 ROWS;
```

81 % Проблемы не найдены.

Результаты Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	геохимик	1908
2	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910
3	Ольденбург Сергей Фёдорович	1863-09-26 00:00:00.000	историк	1903
4	Стеклов Владимир Андреевич	1864-01-09 00:00:00.000	математик	1912
5	Марр Николай Яковлевич	1865-01-06 00:00:00.000	лингвист	1912
6	Истрин Василий Михайлович	1865-02-22 00:00:00.000	филолог	1907

Запрос успешно выполнен. (localdb)\MSS

12

```
SELECT *
FROM Академик
ORDER BY ФИО ASC
OFFSET 8 ROWS FETCH NEXT 2 ROWS ONLY;
```

81 % Проблемы не найдены.

Результаты Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания
1	Истрин Василий Михайлович	1865-02-22 00:00:00.000	филолог	1907
2	Карпинский Александр Петрович	1847-01-07 00:00:00.000	геолог	1889

Запрос успешно выполнен. (localdb)\MS

Лабораторная работа № 3

Фильтрация данных

1. Цель работы

1. Изучение основ фильтрации данных.
2. Изучение операций сравнения. 3.Изучение логических операторов.
4. Изучение BETWEEN.
5. Изучение LIKE.
6. Изучение NULL.
7. Изучение IN.

ПРИМЕРЫ

1.

```
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Площадь > 1000000
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Австралия	Канберра	7692000	25600000	Австралия
2	Алжир	Алжир	2381740	39813722	Африка
3	Ангола	Луанда	1246700	25831000	Африка
4	Аргентина	Буэнос-Айрес	2766890	43847000	Южная Америка
5	Боливия	Сукре	1098580	10985059	Южная Америка
6	Бразилия	Бразилиа	8516000	212600000	Южная Америка

Запрос успешно выполнен.

2.

```
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Население !> 1000000
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Вануату	Порт-Вила	12189	307815	Океания
2	Кирибати	Южная Тарава	811	119449	Океания
3	Маршалловы Острова	Маджуро	181	58413	Океания
4	Микронезия	Паликир	702	115023	Океания
5	Науру	Ярен	21	10824	Океания
6	Палау	Нгерулмуд	459	17901	Океания

Запрос успешно выполнен.

3.

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Континент = 'Африка'
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Столица	Площадь	Население	Континент
1	Алжир	Алжир	2381740	39813722	Африка
2	Ангола	Луанда	1246700	25831000	Африка
3	Ботсвана	Габороне	600370	2209208	Африка
4	Буркина-Фасо	Уагадугу	274200	19034397	Африка

Запрос успешно выполнен.

4.

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Континент != 'Европа'
```

81 % 7 0

Результаты Сообщения

	Название	Столица	Площадь	Население	Континент
1	Австралия	Канберра	7692000	25600000	Австралия
2	Алжир	Алжир	2381740	39813722	Африка
3	Ангола	Луанда	1246700	25831000	Африка
4	Аргентина	Буэнос-Айрес	2766890	43847000	Южная Америка
5	Боливия	Сукре	1098580	10985059	Южная Америка
6	Ботсвана	Габороне	600370	2209208	Африка

Запрос успешно выполнен. (localdb)\MSSQLLocalDB (15

5.

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       (Население > 1000000) AND (Площадь < 100000)
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Столица	Площадь	Население	Континент
1	Германия	Берлин	35788	83100000	Европа

Запрос успешно выполнен.

6.

```
        ,Площадь
        ,Население
        ,Континент
FROM
        Страна
WHERE
        (Континент = 'Европа') AND (Население > 10000000)
        OR
        (Континент = 'Азия') AND (Население > 50000000)
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Великобретания	Лондон	242495	66650000	Европа
2	Германия	Берлин	35788	83100000	Европа
3	Индия	Нью-Дели	3287000	1380000000	Азия
4	Китай	Пекин	9597000	1412000000	Азия
5	Франция	Париж	643801	67000000	Европа
6	Япония	Токио	377975	125800000	Азия

Запрос успешно выполнен.

7.

```
SELECT Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
FROM
        Страна
WHERE
        (Население BETWEEN 10000000 AND 100000000)
        AND
        (Площадь >= 100000) AND (Площадь <= 200000)
```

81 % 2 0

Результаты | Сообщения

Название	Столица	Площадь	Население	Континент

8.

```
SELECT Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
FROM
        Страна
WHERE
    Название BETWEEN 'Бенин' AND 'Ватикан'  ORDER BY
        Название
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Боливия	Сукре	1098580	10985059	Южная Америка
2	Ботсвана	Габороне	600370	2209208	Африка
3	Бразилия	Бразилиа	8516000	212600000	Южная Америка
4	Буркина-Фасо	Уагадугу	274200	19034397	Африка
5	Вануату	Порт-Вила	12189	307815	Океания

9.

```
SELECT Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
FROM
        Страна
WHERE
        Название LIKE 'С%'
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Самоа	Апиа	2842	198414	Океания
2	Соломоновы Острова	Хониара	28896	686878	Океания
3	США	Вашингтон	9834000	331000000	Северная Америка

Запрос успешно выполнен.

10.

3.1.sql - Уче...09\User (52))*

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
   Название LIKE '_а%я'
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Папуа — Новая Гвинея	Порт-Морсби	462840	8935000	Океания

11.

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Название LIKE '__[аоу]%'
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Бразилия	Бразилиа	8516000	212600000	Южная Америка
2	Науру	Ярен	21	10824	Океания
3	США	Вашингтон	9834000	331000000	Северная Америка
4	Франция	Париж	643801	67000000	Европа
5	Япония	Токио	377975	125800000	Азия

12.

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Название LIKE '[А-Г]%'
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Австралия	Канберра	7692000	25600000	Австралия
2	Алжир	Алжир	2381740	39813722	Африка
3	Ангола	Луанда	1246700	25831000	Африка
4	Аргентина	Буэнос-Айрес	2766890	43847000	Южная Америка
5	Боливия	Сукре	1098580	10985059	Южная Америка
6	Ботсвана	Габороне	600370	2209208	Африка

Запрос успешно выполнен.

13.

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Название LIKE '[^А-Г,^С]%'
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Столица	Площадь	Население	Континент
1	Индия	Нью-Дели	3287000	1380000000	Азия
2	Кирибати	Южная Тарава	811	119449	Океания
3	Китай	Пекин	9597000	1412000000	Азия
4	Маршалловы Острова	Маджуро	181	58413	Океания
5	Микронезия	Паликир	702	115023	Океания
6	Науру	Ярен	21	10824	Океания

14.

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Столица IS NULL
```

81 % Проблемы не найдены.

Результаты Сообщения

Название	Столица	Площадь	Население	Континент

15.

```sql
SELECT Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
       Страна
WHERE
       Континент IN ('Европа', 'Азия', 'Африка')
```

столбцы Континент(varchar, null)

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Столица	Площадь	Население	Континент
1	Алжир	Алжир	2381740	39813722	Африка
2	Ангола	Луанда	1246700	25831000	Африка
3	Ботсвана	Габороне	600370	2209208	Африка
4	Буркина-Фасо	Уагадугу	274200	19034397	Африка
5	Великобретания	Лондон	242495	66650000	Европа
6	Германия	Берлин	35788	83100000	Европа

4 задание

1.

```sql
SELECT TOP 5 Название, Столица
FROM Страна
ORDER BY Площадь DESC;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица
1	Россия	Москва
2	США	Вашингтон
3	Китай	Пекин
4	Бразилия	Бразилиа
5	Австралия	Канберра

2.

```sql
SELECT Название
FROM Страна
WHERE Континент = 'Африка' AND Население <= 1000000;
```

81 % Проблемы не найдены.

Результаты | Сообщения

Название

3.

```sql
FROM Страна
WHERE Население > 5000000
  AND Площадь < 100000
  AND Континент != 'Европа';
```

81 % Проблемы не найдены.

Результаты | Сообщения

Название

4.

```sql
SELECT Название
FROM Страна
WHERE (Континент IN ('Южная Америка', 'Северная Америка') AND Население > 20000000)
   OR (Континент = 'Африка' AND Население > 30000000);
```

Проблемы не найдены.

Результаты | Сообщения

	Название
1	Алжир
2	Аргентина
3	Бразилия
4	Венесуэла
5	США

5.

```sql
SELECT Название
FROM Страна
WHERE Население BETWEEN 10000000 AND 100000000
  AND Площадь <= 500000;
```

81 %

Проблемы не найдены.

Результаты | Сообщения

	Название
1	Буркина-Фасо
2	Великобретания
3	Германия

6.

```sql
SELECT Название
FROM Страна
WHERE Название NOT LIKE 'К%';
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название
1	Австралия
2	Алжир
3	Ангола
4	Аргентина
5	Боливия
6	Ботсвана
7	Бразилия
8	Буркина-Фасо
9	Вануату
10	Великобретания
11	Венесуэла
12	Германия

Запрос успешно выполнен.

7.

```sql
SELECT Название
FROM Страна
WHERE SUBSTRING(Название, 3, 1) = 'а'
  AND SUBSTRING(Название, LEN(Название)-1, 1) = 'и';
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название
1	Бразилия
2	Франция

8.

```sql
SELECT Название
FROM Страна
WHERE SUBSTRING(Название, 2, 1) IN ('а', 'е', 'ё', 'и', 'о', 'у', 'ы', 'э', 'ю', 'я');
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название
1	Боливия
2	Ботсвана
3	Буркина-Фасо
4	Вануату
5	Великобретания
6	Венесуэла
7	Германия
8	Кирибати
9	Китай
10	Маршалловы Острова
11	Микронезия

9.

```sql
SELECT Название
FROM Страна
WHERE Название LIKE '[К-П]%';
-- или для некоторых SQL:
-- WHERE Название >= 'К' AND Название < 'Р';
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название
1	Кирибати
2	Китай
3	Маршалловы Острова
4	Микронезия
5	Науру
6	Новая Зеландия
7	Палау
8	Папуа — Новая Гвинея

10.

```sql
SELECT Название
FROM Страна
WHERE Название LIKE '[А,В,Г]%' AND Название NOT LIKE 'Б%';
-- или
WHERE Название >= 'А' AND Название < 'Д' AND Название NOT LIKE 'Б%';
```

11.

```sql
SELECT Название
FROM Страна
WHERE Столица IS NOT NULL;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название
1	Австралия
2	Алжир
3	Ангола
4	Аргентина
5	Боливия
6	Ботсвана
7	Бразилия
8	Буркина-Фасо
9	Вануату
10	Великобретания

12.

```sql
SELECT Название
FROM Страна
WHERE Континент IN ('Африка', 'Южная Америка', 'Северная Америка');
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название
1	Алжир
2	Ангола
3	Аргентина
4	Боливия
5	Ботсвана
6	Бразилия
7	Буркина-Фасо
8	Венесуэла
9	США

Лабораторная работа № 4

Типы данных и встроенные функции

1. Цель работы

1. Изучить основные типы данных.

2. Изучить встроенные функции для работы со строками.
3. Изучить встроенные функции для работы с числами.
4. Изучить встроенные функции для работы с датами и временем.
5. Изучить встроенные функции преобразования данных.
6. Изучить CASE и IIF.

ПРИМЕРЫ

1.

```
select
ФИО
        ,LEN(ФИО) AS Количество_символов
FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Количество_символов
1	Аничков Николай Николаевич	26
2	Бартольд Василий Владимирович	29
3	Белопольский Аристарх Аполлонович	33
4	Бородин Иван Парфеньевич	24
5	Вальден Павел Иванович	22
6	Вернадский Владимир Иванович	28
7	Виноградов Павел Гаврилович	27
8	Ипатьев Владимир Николаевич	27
9	Истрин Василий Михайлович	25
10	Карпинский Александр Петрович	29
11	Коковцов Павел Константинович	29
12	Курнаков Николай Семёнович	26

2.

```sql
SELECT
        TRIM(ФИО) AS ФИО
        ,Дата_рождения
        ,Специализация
        ,Год_присвоения_звания
FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звани
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	медицина	1939
2	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	историк	1913
3	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
4	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
5	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910
6	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	геохимик	1908
7	Виноградов Павел Гаврилович	1854-11-30 00:00:00.000	историк	1914
8	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	химик	1916
9	Истрин Василий Михайлович	1865-02-22 00:00:00.000	филолог	1907
10	Карпинский Александр Петрович	1847-01-07 00:00:00.000	геолог	1889
11	Коковцов Павел Константинович	1861-07-01 00:00:00.000	историк	1906

3.

```sql
SELECT
        ФИО
    ,CHARINDEX('o',ФИО) AS Позиция_o     FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Позиция_o
1	Аничков Николай Николаевич	6
2	Бартольд Василий Владимирович	5
3	Белопольский Аристарх Аполлонович	4
4	Бородин Иван Парфеньевич	2
5	Вальден Павел Иванович	19
6	Вернадский Владимир Иванович	25
7	Виноградов Павел Гаврилович	4
8	Ипатьев Владимир Николаевич	21
9	Истрин Василий Михайлович	22
10	Карпинский Александр Петрович	26
11	Коковцов Павел Константинович	2

4.

```sql
SELECT
        ФИО
    ,LEFT(Специализация, 3) AS Спец_3     FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Спец_3
1	Аничков Николай Николаевич	мед
2	Бартольд Василий Владимирович	ист
3	Белопольский Аристарх Аполлонович	аст
4	Бородин Иван Парфеньевич	бот
5	Вальден Павел Иванович	хим
6	Вернадский Владимир Иванович	гео
7	Виноградов Павел Гаврилович	ист
8	Ипатьев Владимир Николаевич	хим
9	Истрин Василий Михайлович	фил
10	Карпинский Александр Петрович	гео

5.

```sql
SELECT
        ФИО
    ,SUBSTRING(Специализация, 2, 4) AS Спец_2_5     FROM
        Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Спец_2_5
1	Аничков Николай Николаевич	едиц
2	Бартольд Василий Владимирович	стор
3	Белопольский Аристарх Аполлонович	стро
4	Бородин Иван Парфеньевич	отан
5	Вальден Павел Иванович	имик
6	Вернадский Владимир Иванович	еохи
7	Виноградов Павел Гаврилович	стор
8	Ипатьев Владимир Николаевич	имик
9	Истрин Василий Михайлович	илол
10	Карпинский Александр Петрович	еоло
11	Коковцов Павел Константинович	стор

6.

```sql
SELECT
        ФИО
        ,Дата_рождения
        ,REPLACE(Специализация, 'лингвист', 'языковед') AS Спец
        ,Год_присвоения_звания
FROM
        Академик
```

81 % Проблемы не найдены.

Результаты Сообщения

	ФИО	Дата_рождения	Спец	Год_пр
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	медицина	1939
2	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	историк	1913
3	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903
4	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902
5	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910
6	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	геохимик	1908
7	Виноградов Павел Гаврилович	1854-11-30 00:00:00.000	историк	1914
8	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	химик	1916
9	Истрин Василий Михайлович	1865-02-22 00:00:00.000	филолог	1907
10	Карпинский Александр Петрович	1847-01-07 00:00:00.000	геолог	1889

7.

```sql
SELECT
        ФИО
        ,Дата_рождения
        ,UPPER(Специализация) AS Спец
        ,Год_присвоения_звания
FROM
        Академик
```

81 % Проблемы не найдены.

Результаты Сообщения

	ФИО	Дата_рождения	Спец	Год_присвоения_звани
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	МЕДИЦИНА	1939
2	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	ИСТОРИК	1913
3	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	АСТРОФИЗИК	1903
4	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	БОТАНИК	1902
5	Вальден Павел Иванович	1863-07-26 00:00:00.000	ХИМИК-ТЕХНОЛОГ	1910
6	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	ГЕОХИМИК	1908
7	Виноградов Павел Гаврилович	1854-11-30 00:00:00.000	ИСТОРИК	1914
8	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	ХИМИК	1916
9	Истрин Василий Михайлович	1865-02-22 00:00:00.000	ФИЛОЛОГ	1907
10	Карпинский Александр Петрович	1847-01-07 00:00:00.000	ГЕОЛОГ	1889
11	Коковцов Павел Константинович	1861-07-01 00:00:00.000	ИСТОРИК	1906

8.

```sql
SELECT
        ФИО
        ,REVERSE(ФИО) AS ФИО_Обр
FROM
        Академик
        Название
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	ФИО_Обр
1	Аничков Николай Николаевич	чивеалокиН йалокиН вокчинА
2	Бартольд Василий Владимирович	чиворимидалВ йилисаВ дьлотраБ
3	Белопольский Аристарх Аполлонович	чивоноллопА хратсирА йиксьлополеБ
4	Бородин Иван Парфеньевич	чивеьнефраП навИ нидороБ
5	Вальден Павел Иванович	чивонавИ леваП недьлаВ
6	Вернадский Владимир Иванович	чивонавИ римидалВ йиксданреВ
7	Виноградов Павел Гаврилович	чиволирваГ леваП водаргониВ
8	Ипатьев Владимир Николаевич	чивеалокиН римидалВ веьтапИ
9	Истрин Василий Михайлович	чиволйахиМ йилисаВ ниртсИ
10	Карпинский Александр Петрович	чивортеП рднаскелА йикснипраК
11	Коковцов Павел Константинович	чивонитнатснoК леваП воцвокоК

9.

```sql
SELECT DISTINCT
        REPLICATE(Специализация, 4) AS Спец_4
FROM
Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Спец_4
1	астрофизикастрофизикастрофизикастрофизик
2	ботаникботаникботаникботаник
3	геологгеологгеологгеолог
4	геохимикгеохимикгеохимикгеохимик
5	зоологзоологзоологзоолог
6	историкисторикисторикисторик
7	лингвистлингвистлингвистлингвист
8	математикматематикматематикматематик
9	медицинамедицинамедицинамедицина
10	физиологфизиологфизиологфизиолог
11	филологфилологфилологфилолог

10.

```sql
SELECT
        ABS(COS(PI()))   AS Косинус_Пи
       ,ABS(SIN(PI()))   AS Синус_Пи
       ,ABS(TAN(PI()))   AS Тангенс_Пи
       ,ABS(COT(PI()))   AS КоТангенс_Пи
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Косинус_Пи	Синус_Пи	Тангенс_Пи	КоТангенс_Пи
1	1	1,22464679914735E-16	1,22464679914735E-16	8,16561967659769E+15

11.

```sql
SELECT
        ROUND(123.456, 3)   AS Окр3
       ,ROUND(123.456, 2)   AS Окр2
       ,ROUND(123.456, 1)   AS Окр1
       ,ROUND(123.456, 0)   AS Окр0
       ,ROUND(123.456, -1)   AS Окр_1
       ,ROUND(123.456, -2)   AS Окр_2
       ,ROUND(123.456, -3)   AS Окр_3
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Окр3	Окр2	Окр1	Окр0	Окр_1	Окр_2	Окр_3
1	123.456	123.460	123.500	123.000	120.000	100.000	0.000

12.

```sql
SELECT
        CEILING(123.456)  AS Больше
       ,FLOOR(123.456)    AS Меньше
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Больше	Меньше
1	124	123

13.

```sql
SELECT
        SQRT(25)      AS Корень
       ,SQUARE(25)    AS Квадрат
,POWER(25, 3) AS Куб
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Корень	Квадрат	Куб
1	5	625	15625

14.

```sql
SELECT
        GETDATE() AS Сейчас
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Сейчас
1	2026-02-09 12:15:03.053

15.

```sql
SELECT
        DAY(GETDATE()) AS День
       ,MONTH(GETDATE()) AS Месяц
       ,YEAR(GETDATE()) AS Год
       ,DATEPART(HOUR, GETDATE()) AS Час
       ,DATEPART(MINUTE, GETDATE()) AS Минута
       ,DATEPART(SECOND, GETDATE()) AS Секунда
       ,DATEPART(QUARTER, GETDATE()) AS Квартал
       ,DATEPART(WEEK, GETDATE()) AS Неделя
       ,DATEPART(DAYOFYEAR, GETDATE()) AS День_года
       ,DATEPART(WEEKDAY, GETDATE()) AS День_недели
```

81 % Проблемы не найдены.

Результаты | Сообщения

	День	Месяц	Год	Час	Минута	Секунда	Квартал	Неделя	День_года	День_недели
1	9	2	2026	12	15	46	1	7	40	2

16.

```sql
SELECT
        DATEADD(DAY, -100, GETDATE()) AS День_100_Назад
```

81 % Проблемы не найдены.

Результаты | Сообщения

	День_100_Назад
1	2025-11-01 12:16:25.673

17.

```sql
SELECT
        DATEDIFF(DAY, '18950708', '19710412') AS Количество_прожитых_дней
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Количество_прожитых_дней
1	27671

18.

```sql
SELECT
        ФИО
    , CASE MONTH(Дата_рождения)         WHEN 3 THEN 'Весна'
            WHEN 4 THEN 'Весна'
            WHEN 5 THEN 'Весна'
            WHEN 6 THEN 'Лето'
            WHEN 7 THEN 'Лето'
            WHEN 8 THEN 'Лето'
            WHEN 9 THEN 'Осень'
            WHEN 10 THEN 'Осень'
            WHEN 11 THEN 'Осень'
            ELSE 'Зима'
            END AS Времени_года
FROM Академик
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Времени_года
1	Аничков Николай Николаевич	Осень
2	Бартольд Василий Владимирович	Осень
3	Белопольский Аристарх Аполлонович	Лето
4	Бородин Иван Парфеньевич	Зима
5	Вальден Павел Иванович	Лето
6	Вернадский Владимир Иванович	Весна
7	Виноградов Павел Гаврилович	Осень
8	Ипатьев Владимир Николаевич	Осень
9	Истрин Василий Михайлович	Зима
10	Карпинский Александр Петрович	Зима
11	Коковцов Павел Константинович	Лето
12	Кузнецов Николай Семёнович	Зима

19.

```sql
14  OR (MONTH(Дата_рождения)=8  AND DAY(Дата_рождения) <= 21) THEN 'Лев'
15              WHEN (MONTH(Дата_рождения)=8  AND DAY(Дата_рождения) >= 22)
16  OR (MONTH(Дата_рождения)=9  AND DAY(Дата_рождения) <= 23) THEN 'Дева'
17              WHEN (MONTH(Дата_рождения)=9  AND DAY(Дата_рождения) >= 24)
18  OR (MONTH(Дата_рождения)=10 AND DAY(Дата_рождения) <= 23) THEN 'Весы'
19              WHEN (MONTH(Дата_рождения)=10 AND DAY(Дата_рождения) >= 24)
20  OR (MONTH(Дата_рождения)=11 AND DAY(Дата_рождения) <= 22) THEN 'Скорпион'
21              WHEN (MONTH(Дата_рождения)=11 AND DAY(Дата_рождения) >= 23)
22  OR (MONTH(Дата_рождения)=12 AND DAY(Дата_рождения) <= 22) THEN 'Стрелец'
23              WHEN (MONTH(Дата_рождения)=12 AND DAY(Дата_рождения) >= 23)
24  OR (MONTH(Дата_рождения)=1  AND DAY(Дата_рождения) <= 20) THEN 'Козерог'
25              WHEN (MONTH(Дата_рождения)=1  AND DAY(Дата_рождения) >= 21)
26  OR (MONTH(Дата_рождения)=2  AND DAY(Дата_рождения) <= 19) THEN 'Водолей'
27              WHEN (MONTH(Дата_рождения)=2  AND DAY(Дата_рождения) >= 20)
28  OR (MONTH(Дата_рождения)=3  AND DAY(Дата_рождения) <= 20) THEN 'Рыбы'
29          END AS Знак_зодиака
30  FROM Академик
31
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Знак_зодиака
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	Скорпион
2	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	Скорпион
3	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	Рак
4	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	Водолей
5	Вальден Павел Иванович	1863-07-26 00:00:00.000	Лев
6	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	Рыбы
7	Виноградов Павел Гаврилович	1854-11-30 00:00:00.000	Стрелец
8	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	Скорпион
9	Истрин Василий Михайлович	1865-02-22 00:00:00.000	Рыбы
10	Карпинский Александр Петрович	1847-01-07 00:00:00.000	Козерог
11	Кокорцов Павел Константинович	1861-07-01 00:00:00.000	Рак

20.

```sql
1  SELECT
2          ФИО
3          ,Дата_рождения
4          ,Специализация
5          ,Год_присвоения_звания
6          ,IIF(Год_присвоения_звания - Year(Дата_рождения) <= 45, 'Молодой','Старый')
7  AS Возраст_при_присвоении
8  FROM Академик
9
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Дата_рождения	Специализация	Год_присвоения_звания	Возраст_при_присвоении
1	Аничков Николай Николаевич	1885-11-03 00:00:00.000	медицина	1939	Старый
2	Бартольд Василий Владимирович	1869-11-15 00:00:00.000	историк	1913	Молодой
3	Белопольский Аристарх Аполлонович	1854-07-13 00:00:00.000	астрофизик	1903	Старый
4	Бородин Иван Парфеньевич	1847-01-30 00:00:00.000	ботаник	1902	Старый
5	Вальден Павел Иванович	1863-07-26 00:00:00.000	химик-технолог	1910	Старый
6	Вернадский Владимир Иванович	1863-03-12 00:00:00.000	геохимик	1908	Молодой
7	Виноградов Павел Гаврилович	1854-11-30 00:00:00.000	историк	1914	Старый
8	Ипатьев Владимир Николаевич	1867-11-21 00:00:00.000	химик	1916	Старый
9	Истрин Василий Михайлович	1865-02-22 00:00:00.000	филолог	1907	Молодой
10	Карпинский Александр Петрович	1847-01-07 00:00:00.000	геолог	1889	Молодой
11	Кокорцов Павел Константинович	1861-07-01 00:00:00.000	историк	1906	Молодой

4 Задание

1.

```sql
SELECT ФИО
FROM академик
ORDER BY LEN(ФИО);
```

81 % 3 0

Результаты Сообщения

	ФИО
1	Павлов Иван Петрович
2	Марр Николай Яковлевич
3	Вальден Павел Иванович
4	Бородин Иван Парфеньевич
5	Истрин Василий Михайлович
6	Аничков Николай Николаевич
7	Курнаков Николай Семёнович
8	Насонов Николай Викторович
9	Перетц Владимир Николаевич
10	Стеклов Владимир Андреевич

2.

```sql
SELECT TRIM(ФИО) AS ФИО_без_пробелов
FROM академик;
```

81 % Проблемы не найдены.

Результаты Сообщения

	ФИО_без_пробелов
1	Аничков Николай Николаевич
2	Бартольд Василий Владимирович
3	Белопольский Аристарх Аполлонович
4	Бородин Иван Парфеньевич
5	Вальден Павел Иванович
6	Вернадский Владимир Иванович
7	Виноградов Павел Гаврилович
8	Ипатьев Владимир Николаевич
9	Истрин Василий Михайлович
10	Карпинский Александр Петрович

3.

```sql
SELECT ФИО, CHARINDEX('ов', ФИО) AS Позиция_ов
FROM академик
WHERE CHARINDEX('ов', ФИО) > 0;
```

81 % Проблемы не найдены.

Результаты Сообщения

	ФИО	Позиция_ов
1	Аничков Николай Николаевич	6
2	Бартольд Василий Владимирович	26
3	Белопольский Аристарх Аполлонович	30
4	Вальден Павел Иванович	19
5	Вернадский Владимир Иванович	25
6	Виноградов Павел Гаврилович	9
7	Истрин Василий Михайлович	22
8	Карпинский Александр Петрович	26
9	Коковцов Павел Константинович	4
10	Курнаков Николай Семёнович	7
11	Марр Николай Яковлевич	16

4.

```sql
SELECT ФИО, RIGHT(Специализация, 2) AS Последние_две_буквы
FROM академик;
```

81 % Проблемы не найдены.

Результаты Сообщения

	ФИО	Последние_две_буквы
1	Аничков Николай Николаевич	на
2	Бартольд Василий Владимирович	ик
3	Белопольский Аристарх Аполлонович	ик
4	Бородин Иван Парфеньевич	ик
5	Вальден Павел Иванович	ог
6	Вернадский Владимир Иванович	ик
7	Виноградов Павел Гаврилович	ик
8	Ипатьев Владимир Николаевич	ик
9	Истрин Василий Михайлович	ог
10	Карпинский Александр Петрович	ог
11	Коковцов Павел Константинович	ик
12	Курнаков Николай Семёнович	ик

5.

```sql
-- 5. Вывести список академиков, ФИО в формате фамилия и инициалы
SELECT
  CONCAT(
    LEFT(ФИО, CHARINDEX(' ', ФИО) - 1), -- Фамилия
    ' ',
    LEFT(SUBSTRING(ФИО, CHARINDEX(' ', ФИО) + 1, LEN(ФИО)), 1), -- Инициал имени
    '.',
    LEFT(SUBSTRING(ФИО, CHARINDEX(' ', ФИО, CHARINDEX(' ', ФИО) + 1) + 1, LEN(ФИО)), 1)
    '.'
  ) AS Фамилия_и_инициалы
FROM академик
WHERE ФИО LIKE '% % %';
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Фамилия_и_инициалы
1	Аничков Н.Н.
2	Бартольд В.В.
3	Белопольский А.А.
4	Бородин И.П.
5	Вальден П.И.
6	Вернадский В.И.
7	Виноградов П.Г.
8	Ипатьев В.Н.
9	Истрин В.М.
10	Карпинский А.П.

6.

```sql
SELECT DISTINCT
  Специализация,
  REVERSE(Специализация) AS Обратный_вид
FROM академик;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Специализация	Обратный_вид
1	астрофизик	кизифортса
2	ботаник	кинатоб
3	геолог	голоег
4	геохимик	кимихоег
5	зоолог	голооз
6	историк	киротси
7	лингвист	тсивгнил
8	математик	китаметам
9	медицина	аницидем
10	физиолог	голоизиф

7.

```sql
DECLARE @фамилия NVARCHAR(50) = 'Туринге';
DECLARE @возраст INT = 20;

SELECT REPLICATE(@фамилия + ' ', @возраст) AS Фамилия_повтор;
```

81 % 2 0

Результаты | Сообщения

	Фамилия_повтор
1	Туринге Туринге Туринге Туринге Туринге Туринге ...

8.

```
SELECT
  ROUND(ABS(SIN(RADIANS(45))), 2) AS Абсолютное_значение;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Абсолютное_значение
1	0

9.

```
DECLARE @конец_семестра DATE = '2026-06-30';
SELECT DATEDIFF(DAY, GETDATE(), @конец_семестра) AS Дней_до_конца_семестра;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Дней_до_конца_семестра
1	141

10.

```
DECLARE @дата_рождения DATE = '2008-08-21';
SELECT DATEDIFF(MONTH, @дата_рождения, GETDATE()) AS Месяцев_с_рождения
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Месяцев_с_рождения
1	210

11.

```sql
SELECT
  ФИО,
  Год_рождения,
  CASE
    WHEN (Год_рождения % 4 = 0 AND Год_рождения % 100 != 0) OR (Год_рождения % 400 = 0)
    THEN 'Високосный'
    ELSE 'Не високосный'
  END AS Високосность
FROM (
  SELECT
    ФИО,
    YEAR(Дата_рождения) AS Год_рождения
  FROM академик
) AS t;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	ФИО	Год_рождения	Високосность
1	Аничков Николай Николаевич	1885	Не високосный
2	Бартольд Василий Владимирович	1869	Не високосный
3	Белопольский Аристарх Аполлонович	1854	Не високосный
4	Бородин Иван Парфеньевич	1847	Не високосный
5	Вальден Павел Иванович	1863	Не високосный
6	Вернадский Владимир Иванович	1863	Не високосный
7	Виноградов Павел Гаврилович	1854	Не високосный
8	Ипатьев Владимир Николаевич	1867	Не високосный
9	Истрин Василий Михайлович	1865	Не високосный
10	Карпинский Александр Петрович	1847	Не високосный
11	Коковцов Павел Константинович	1861	Не високосный

12.

```sql
SELECT DISTINCT
  Специализация,
  CASE
    WHEN LEN(Специализация) > 7 THEN 'длинный'
    ELSE 'короткий'
  END AS Длина_названия
FROM академик;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Специализация	Длина_названия
1	астрофизик	длинный
2	ботаник	короткий
3	геолог	короткий
4	геохимик	длинный
5	зоолог	короткий
6	историк	короткий
7	лингвист	длинный
8	математик	длинный
9	медицина	длинный
10	физиолог	длинный
11	филолог	короткий
12	химик	короткий

Лабораторная работа № 5

Агрегатные функции

1. Цель работы

1. Изучить основы агрегации данных.
2. Изучить функцию MAX.
3. Изучить функцию MIN.
4. Изучить функцию SUM.

5. Изучить функцию AVG.
6. Изучить функцию COUNT.
7. Изучить группировки данных.
8. Изучить применение фильтрации в группировке данных.

ПРИМЕРЫ

1.

```
SELECT
    MAX(Площадь) AS Макс_площадь  FROM
        Страна
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Макс_площадь
1	17100000

2.

```
SELECT
    MIN(Население) AS Мин_население FROM
        Страна
WHERE
        Страна
        Страна
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Мин_население
1	2209208

3.

```sql
SELECT
    SUM(Население) AS Суммарное_население FROM
        Страна
WHERE
        Континент = 'Северная Америка' OR
        Континент = 'Южная Америка'
```

81 % Проблемы не найдены.

Результаты Сообщения

	Суммарное_население
1	629460696

4.

```sql
SELECT
 ROUND(AVG(CAST(Население AS FLOAT)), 2) AS Среднее_население FROM
        Страна
WHERE
        Континент != 'Европа'
```

% Проблемы не найдены.

Результаты Сообщения

Среднее_население
141593375,41

5.

```sql
SELECT
        COUNT(*) AS Количество
FROM
        Страна
WHERE
        LEFT(Название, 1) = 'C'
```

81 % Проблемы не найдены.

Результаты Сообщения

	Количество
1	3

6.

```sql
SELECT
    COUNT(DISTINCT Континент) AS Количество_Континентов FROM
        Страна
```

81 % Проблемы не найдены.

Результаты Сообщения

	Количество_Континентов
1	8

7.

```sql
SELECT
    MAX(Население) - MIN(Население) AS Разница FROM
        Страна
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Разница
1	1411989176

8.

```sql
SELECT
        Континент
, COUNT(Название) AS Количество_Стран FROM
        Страна
GROUP BY
        Континент
ORDER BY
Количество_Стран DESC
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Континент	Количество_Стран
1	Океания	13
2	Южная Америка	4
3	Африка	4
4	Европа	3
5	Азия	3
6	Австралия	1
7	Европа/Азия	1
8	Северная Америка	1

9.

```sql
SELECT
        LEFT(Название, 1) AS Первая_буква
    , COUNT(Название) AS Количество_Стран FROM
        Страна
GROUP BY
    LEFT(Название, 1) ORDER BY
        Первая_буква
```

81 % Проблемы не найдены.

Результаты Сообщения

	Первая_буква	Количество_Стран
1	А	4
2	Б	4
3	В	3
4	Г	1
5	И	1
6	К	2
7	М	2
8	Н	2
9	П	2
10	Р	1

10.

```sql
SELECT
        Континент
    , AVG(CAST(Население AS FLOAT) / Площадь) AS Сред_Плотность FROM
        Страна
GROUP BY
        Континент
HAVING
        AVG(CAST(Население AS FLOAT) / Площадь) > 100
```

81 % Проблемы не найдены.

Результаты Сообщения

	Континент	Сред_Плотность
1	Азия	299,930424274635
2	Европа	900,309276603367
3	Океания	152,49668130958

11.

```sql
SELECT
        Континент
        , CASE
        WHEN Континент IN ('Европа', 'Азия') THEN FLOOR(SUM(Население) * 1.2)

        WHEN Континент IN ('Северная Америка', 'Африка') THEN FLOOR(SUM(Население) * 1.5)
        ELSE FLOOR(SUM(Население) * 1.7)
    END AS Суммарное_Население FROM
        Страна
GROUP BY
        Континент
```

81 % Проблемы не найдены.

Результаты Сообщения

	Континент	Суммарное_Население
1	Австралия	43520000
2	Азия	3501360000
3	Африка	130332490
4	Европа	260100000
5	Европа/Азия	249390000
6	Океания	28172592
7	Северная Америка	496500000
8	Южная Америка	507383183

12.

```sql
SELECT
        Континент
FROM
        Страна
GROUP BY
        Континент
HAVING MAX(Население) <= 1000 * MIN(Население)
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Континент
1	Австралия
2	Азия
3	Африка
4	Европа
5	Европа/Азия
6	Океания
7	Северная Америка
8	Южная Америка

13.

```sql
SELECT
    COUNT(*) AS Количество   FROM
        Страна
WHERE
        Столица IS NULL
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Количество
1	0

14.

```sql
SELECT
        MAX(LEN(Название)) AS Дл_Название
        , MAX(LEN(Столица)) AS Дл_Столица
        , MIN(LEN(Название)) AS Кр_Название
    , MIN(LEN(Столица)) AS Кр_Столица FROM
        Страна
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Дл_Название	Дл_Столица	Кр_Название	Кр_Столица
1	20	12	3	4

15.

```sql
SELECT
        Континент
    ,AVG(CAST(Население AS FLOAT) / Площадь) AS Плотность FROM
        Страна
WHERE
    Площадь > 1000000 GROUP BY
        Континент
HAVING
        AVG(CAST(Население AS FLOAT) / Площадь) > 30
ORDER BY
        Плотность DESC
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Континент	Плотность
1	Азия	283,482513850937
2	Северная Америка	33,6587350010169

4 задание

1.

```sql
SELECT MIN(Площадь) AS Минимальная_площадь
FROM Страна;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Минимальная_площадь
1	21

2.

```sql
SELECT TOP 1 Название, Население
FROM Страна
WHERE Континент IN ('Южная Америка', 'Северная Америка')
ORDER BY Население DESC;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Население
1	США	331000000

3.

```sql
SELECT ROUND(AVG(CAST(Население AS FLOAT)), 1) AS Среднее_население
FROM Страна;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Среднее_население
1	134659037,9

4.

```sql
SELECT COUNT(*) AS Количество_стран
FROM Страна
WHERE Название LIKE '%ан' AND Название NOT LIKE '%стан';
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Количество_стран
1	0

5.

```sql
SELECT COUNT(DISTINCT Континент) AS Количество_континентов
FROM Страна
WHERE Название LIKE 'Р%';
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Количество_континентов
1	1

6.

```sql
SELECT
  MAX(Площадь) / NULLIF(MIN(Площадь), 0) AS Во_сколько_раз_больше
FROM Страна;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Во_сколько_раз_больше
1	814285,714285714

7.

```sql
SELECT Континент, COUNT(*) AS Количество_стран
FROM Страна
WHERE Население > 100000000
GROUP BY Континент
ORDER BY Количество_стран;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Континент	Количество_стран
1	Европа/Азия	1
2	Северная Америка	1
3	Южная Америка	1
4	Азия	3

8.

```sql
SELECT LEN(Название) AS Количество_букв, COUNT(*) AS Количество_стран
FROM Страна
GROUP BY LEN(Название)
ORDER BY Количество_стран DESC;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Количество_букв	Количество_стран
1	5	8
2	6	4
3	8	4
4	9	3
5	7	3
6	14	2
7	18	2
8	20	1
9	10	1

9.

```sql
SELECT
  Континент,
  SUM(Население * 1.1) AS Прогнозируемое_население
FROM Страна
GROUP BY Континент;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Континент	Прогнозируемое_население
1	Австралия	28160000.0
2	Азия	3209580000.0
3	Африка	95577159.7
4	Европа	238425000.0
5	Европа/Азия	161370000.0
6	Океания	18229324.3
7	Северная Америка	364100000.0
8	Южная Америка	328306765.6

10.

```sql
SELECT Континент
FROM Страна
GROUP BY Континент
HAVING
  MAX(Площадь) / NULLIF(MIN(Площадь), 0) <= 10000
  OR MIN(Площадь) = 0;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Континент
1	Австралия
2	Азия
3	Африка
4	Европа
5	Европа/Азия
6	Северная Америка
7	Южная Америка

11.

```sql
SELECT AVG(CAST(LEN(Название) AS FLOAT)) AS Средняя_длина_названий
FROM Страна
WHERE Континент = 'Африка';
```

81 % Проблемы не найдены.

Результаты Сообщения

	Средняя_длина_названий
1	7,75

12.

```
SELECT Континент
FROM Страна
WHERE Население > 1000000
GROUP BY Континент
HAVING AVG(CAST(Население AS FLOAT) / Площадь) > 30;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Континент
1	Азия
2	Европа
3	Северная Америка

Лабораторная работа № 6

Соединения таблиц

1. Цель работы

1. Изучить неявные соединения таблиц.
2. Изучить явные соединения таблиц.
3. Изучить внутреннее соединение таблиц.
4. Изучить внешнее соединение таблиц.
5. Изучить соединения таблиц со своей копией.

ПРИМЕРЫ

1.

```sql
SELECT
        Ф.Название AS Факультет
    , К.Название AS Кафедра FROM
    Факультет Ф, Кафедра К  WHERE
    Ф.Аббревиатура = К.Факультет ORDER BY
        Факультет, Кафедра
```

81 % Проблемы не найдены.

Результаты Сообщения

	Факультет	Кафедра
1	Естественные науки	Высшая математика
2	Естественные науки	Общая физика
3	Информационные технологии	Информационные системы
4	Информационные технологии	Прикладная информатика
5	Физико-математический	Математическое моделирование
6	Физико-математический	Экспериментальная физика

2.

```sql
SELECT
        Ф.Название AS Факультет
        , К.Название AS Кафедра
FROM
        Факультет Ф
        INNER JOIN Кафедра К ON Ф.Аббревиатура = К.Факультет ORDER BY
        Факультет, Кафедра
```

81 % Проблемы не найдены.

Результаты Сообщения

	Факультет	Кафедра
1	Естественные науки	Высшая математика
2	Естественные науки	Общая физика
3	Информационные технологии	Информационные системы
4	Информационные технологии	Прикладная информатика
5	Физико-математический	Математическое моделирование
6	Физико-математический	Экспериментальная физика

3.

```sql
SELECT
        Ф.Название AS Факультет
        , К.Название AS Кафедра
FROM
        Факультет Ф
  LEFT OUTER JOIN Кафедра К ON Ф.Аббревиатура = К.Факультет ORDER BY
        Факультет, Кафедра
```

81 % Проблемы не найдены.

Результаты Сообщения

	Факультет	Кафедра
1	Гуманитарные науки	NULL
2	Естественные науки	Высшая математика
3	Естественные науки	Общая физика
4	Информационные технологии	Информационные системы
5	Информационные технологии	Прикладная информатика
6	Физико-математический	Математическое моделирование
7	Физико-математический	Экспериментальная физика

4.

```sql
SELECT
        С.Фамилия
        , П.Направление
        , К.Название AS Кафедра
FROM
        Студент С
            INNER JOIN Специальность П ON С.Номер = П.Номер
            INNER JOIN Кафедра К ON П.Шифр = К.Шифр
```

81 % Проблемы не найдены.

Результаты Сообщения

	Фамилия	Направление	Кафедра
1	Сидоров	Информационные системы и технологии	Информационные сист
2	Кузнецов	Информационные системы и технологии	Информационные сист
3	Александров	Бизнес-информатика	Информационные сист
4	Дмитриев	Бизнес-информатика	Информационные сист
5	Васильев	Прикладная математика	Математическое модел
6	Николаев	Прикладная математика	Математическое модел
7	Иванов	Прикладная информатика	Прикладная информат
8	Петров	Прикладная информатика	Прикладная информат
9	Смирнов	Ядерные физика и технологии	Экспериментальная фи
10	Попов	Ядерные физика и технологии	Экспериментальная фи

5.

```sql
SELECT
        С.Фамилия
        , С.Должность
        , С.Зарплата
        , П.Фамилия AS Руководитель
FROM
        Сотрудник С
            INNER JOIN Сотрудник П ON С.Шеф = П.Таб_номер
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Фамилия	Должность	Зарплата	Руководитель
1	Прохоров П.П.	зав.кафедрой	35000.00	Прохоров П.П.
2	Семенов С.С.	преподаватель	25000.00	Прохоров П.П.
3	Петров П.П.	преподаватель	25000.00	Прохоров П.П.
4	Сидорова С.С.	инженер	15000.00	Семенов С.С.
5	Андреев А.А.	зав.кафедрой	35000.00	Андреев А.А.
6	Борисов Б.Б.	преподаватель	25000.00	Андреев А.А.
7	Глухов Г.Г.	инженер	20000.00	Андреев А.А.
8	Чернов Ч.Ч.	инженер	15000.00	Борисов Б.Б.
9	Басов Б.Б.	зав.кафедрой	35000.00	Басов Б.Б.

6.

```sql
SELECT
        С.Фамилия
FROM
        Студент С
            INNER JOIN Экзамен Э ON С.Рег_номер = Э.Рег_номер
GROUP BY
        С.Фамилия
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Фамилия
1	Александров
2	Васильев
3	Дмитриев
4	Иванов
5	Кузнецов
6	Николаев
7	Петров
8	Попов
9	Сидоров
10	Смирнов

7.

```sql
SELECT
        С.Фамилия
        , COUNT(Э.Оценка) AS [Количество экзаменов]
        , AVG(Э.Оценка) AS [Средний балл]
FROM
        Студент С
            INNER JOIN Экзамен Э ON С.Рег_номер = Э.Рег_номер
GROUP BY
        С.Фамилия
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Фамилия	Количество экзаменов	Средний балл
1	Александров	3	4
2	Васильев	1	5
3	Дмитриев	2	4
4	Иванов	4	4
5	Кузнецов	1	3
6	Николаев	1	3
7	Петров	2	4
8	Попов	2	4
9	Сидоров	2	3
10	Смирнов	2	3

8.

```sql
SELECT
        С.Фамилия
        , С.Зарплата
        , З.Стаж
FROM
        Сотрудник С
            INNER JOIN Зав_кафедрой З ON С.Таб_номер = З.Таб_номер
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Фамилия	Зарплата	Стаж
1	Прохоров П.П.	35000.00	15
2	Андреев А.А.	35000.00	18
3	Басов Б.Б.	35000.00	20
4	Волков В.В.	35000.00	10
5	Кузнецов К.К.	35000.00	18
6	Зверев З.З.	35000.00	8

9.

```sql
SELECT
    С.Фамилия    , П.Степень FROM
        Сотрудник С
  INNER JOIN Преподаватель П ON С.Таб_номер = П.Таб_номер WHERE
        П.Степень IN ('к.ф.-м.н.', 'д.ф.-м.н.')
```

10.

```sql
SELECT DISTINCT
        Д.Название AS Дисциплина
        , С.Фамилия
        , С.Должность
        , П.Степень
        , Э.Дата
      , Э.Аудитория FROM
        Экзамен Э
            INNER JOIN Дисциплина Д ON Э.Код = Д.Код
            INNER JOIN Сотрудник С ON Э.Таб_номер = С.Таб_номер
    INNER JOIN Преподаватель П ON Э.Таб_номер = П.Таб_номер ORDER BY
        Э.Дата
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Дисциплина	Фамилия	Должность	Степень	Дата	Аудитория
1	базы данных	Борисов Б.Б.	преподаватель	к.ф.-м.н.	2015-06-05	т506
2	информатика	Семенов С.С.	преподаватель	к.ф.-м.н.	2015-06-05	т505
3	информатика	Петров П.П.	преподаватель	к.т.н.	2015-06-07	ф419
4	информатика	Прохоров П.П.	зав.кафедрой	д.т.н.	2015-06-07	т506
5	информатика	Семенов С.С.	преподаватель	к.ф.-м.н.	2015-06-07	м425
6	моделирование	Сергеев С.С.	преподаватель	к.т.н.	2015-06-09	ф333
7	программирование	Зайцев З.З.	преподаватель	к.т.н.	2015-06-09	м424
8	математика	Кузнецов К.К.	зав.кафедрой	д.ф.-м.н.	2015-06-10	т506
9	моделирование	Басов Б.Б.	зав.кафедрой	д.т.н.	2015-06-10	э105

11.

```sql
SELECT
        С.Фамилия
      , COUNT(Э.Дата) AS [Количество экзаменов] FROM
        Экзамен Э
        INNER JOIN Сотрудник С ON Э.Таб_номер = С.Таб_номер GROUP BY
        С.Фамилия
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Фамилия	Количество экзаменов
1	Басов Б.Б.	2
2	Борисов Б.Б.	2
3	Зайцев З.З.	1
4	Зверев З.З.	1
5	Кузнецов К.К.	4
6	Петров П.П.	1
7	Прохоров П.П.	1
8	Романов Р.Р.	1

12.

```sql
SELECT
      С.Фамилия FROM
        Студент С
            LEFT OUTER JOIN Экзамен Э ON С.Рег_номер = Э.Рег_номер
WHERE
        Э.Рег_номер IS NULL
```

81 % Проблемы не найдены.

Результаты | Сообщения

Фамилия

4 задание

1.

```sql
SELECT s.Рег_номер, s.Фамилия, s.Номер, sp.Направление, k.Название, k.Факультет
FROM Студент s
JOIN Специальность sp ON s.Номер = sp.Номер
JOIN Кафедра k ON sp.Шифр = k.Шифр
WHERE k.Факультет = 'ит';
```

81 % Проблемы не найдены.

Результаты Сообщения

	Рег_номер	Фамилия	Номер	Направление	Название	Факультет
1	20101	Сидоров	09.03.02	Информационные системы и технологии	Информационные системы	ит
2	20102	Кузнецов	09.03.02	Информационные системы и технологии	Информационные системы	ит
3	10101	Иванов	09.03.03	Прикладная информатика	Прикладная информатика	ит
4	10102	Петров	09.03.03	Прикладная информатика	Прикладная информатика	ит
5	80101	Александров	38.03.05	Бизнес-информатика	Информационные системы	ит
6	80102	Дмитриев	38.03.05	Бизнес-информатика	Информационные системы	ит

2.

```sql
SELECT DISTINCT k.Факультет, k.Шифр, k.Название, c.Фамилия
FROM Кафедра k
JOIN Специальность sp ON k.Шифр = sp.Шифр
JOIN Сотрудник c ON k.Шифр = c.Шифр AND c.Должность = 'зав.кафедрой'
WHERE sp.Номер IS NOT NULL;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Факультет	Шифр	Название	Фамилия
1	ит	ис	Информационные системы	Андреев А.А.
2	фм	мм	Математическое моделирование	Басов Б.Б.
3	фм	эф	Экспериментальная физика	Зверев З.З.
4	ит	пи	Прикладная информатика	Прохоров П.П.

3.

```sql
SELECT с.Таб_номер, с.Фамилия, с.Шеф, r.Фамилия AS Руководитель
FROM Сотрудник с
LEFT JOIN Сотрудник r ON с.Шеф = r.Таб_номер AND с.Таб_номер != с.Шеф
ORDER BY с.Таб_номер;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Таб_номер	Фамилия	Шеф	Руководитель
1	101	Прохоров П.П.	101	NULL
2	102	Семенов С.С.	101	Прохоров П.П.
3	105	Петров П.П.	101	Прохоров П.П.
4	153	Сидорова С.С.	102	Семенов С.С.
5	201	Андреев А.А.	201	NULL
6	202	Борисов Б.Б.	201	Андреев А.А.
7	241	Глухов Г.Г.	201	Андреев А.А.
8	242	Чернов Ч.Ч.	202	Борисов Б.Б.
9	301	Басов Б.Б.	301	NULL

4.

```sql
SELECT s.Рег_номер, s.Фамилия, COUNT(e.ID) AS Количество_экзаменов
FROM Студент s
JOIN Экзамен e ON s.Рег_номер = e.Рег_номер
GROUP BY s.Рег_номер, s.Фамилия
HAVING COUNT(e.ID) >= 2;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Рег_номер	Фамилия	Количество_экзаменов
1	10101	Иванов	4
2	10102	Петров	2
3	20101	Сидоров	2
4	30101	Смирнов	2
5	30102	Попов	2
6	80101	Александров	3
7	80102	Дмитриев	2

5.

```sql
SELECT Таб_номер, Фамилия, Должность, Зарплата
FROM Сотрудник
WHERE Должность = 'инженер' AND Зарплата < 20000;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Таб_номер	Фамилия	Должность	Зарплата
1	153	Сидорова С.С.	инженер	15000.00
2	242	Чернов Ч.Ч.	инженер	15000.00

6.

```sql
SELECT DISTINCT s.Рег_номер, s.Фамилия
FROM Студент s
JOIN Экзамен e ON s.Рег_номер = e.Рег_номер
WHERE e.Аудитория = 't505';
```

81 % Проблемы не найдены.

Результаты | Сообщения

Рег_номер	Фамилия

7.

```sql
SELECT s.Рег_номер, s.Фамилия,
       COUNT(e.ID) AS Количество_экзаменов,
       ROUND(AVG(CAST(e.Оценка AS FLOAT)), 2) AS Средний_балл
FROM Студент s
JOIN Экзамен e ON s.Рег_номер = e.Рег_номер
GROUP BY s.Рег_номер, s.Фамилия
HAVING AVG(CAST(e.Оценка AS FLOAT)) >= 4;
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Рег_номер	Фамилия	Количество_экзаменов	Средний_балл
1	10101	Иванов	4	4,25
2	10102	Петров	2	4
3	30102	Попов	2	4,5
4	50101	Васильев	1	5
5	80101	Александров	3	4,67
6	80102	Дмитриев	2	4

8.

```sql
SELECT c.Фамилия, c.Зарплата, p.Степень
FROM Сотрудник c
JOIN Преподаватель p ON c.Таб_номер = p.[Таб_номер]
WHERE c.Должность = 'зав.кафедрой';
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Фамилия	Зарплата	Степень
1	Прохоров П.П.	35000.00	д.т.н.
2	Андреев А.А.	35000.00	д.ф.-м.н.
3	Басов Б.Б.	35000.00	д.т.н.
4	Волков В.В.	35000.00	д.т.н.
5	Кузнецов К.К.	35000.00	д.ф.-м.н.
6	Зверев З.З.	35000.00	д.ф.-м.н.

9.

```sql
SELECT с.Таб_номер, с.Фамилия, р.Звание, р.Степень
FROM Сотрудник с
JOIN Преподаватель р ON с.Таб_номер = р.[Таб_номер]
WHERE р.Звание = 'профессор';
```

81 % Проблемы не найдены.

Результаты Сообщения

	Таб_номер	Фамилия	Звание	Степень
1	101	Прохоров П.П.	профессор	д.т.н.
2	201	Андреев А.А.	профессор	д.ф.-м.н.
3	301	Басов Б.Б.	профессор	д.т.н.
4	401	Волков В.В.	профессор	д.т.н.
5	501	Кузнецов К.К.	профессор	д.ф.-м.н.
6	502	Романов Р.Р.	профессор	д.ф.-м.н.
7	601	Зверев З.З.	профессор	д.ф.-м.н.

10.

```sql
SELECT
    е.Код AS Код_дисциплины,
    с.Фамилия,
    с.Должность,
    р.Степень,
    е.Дата,
    е.Аудитория
FROM Экзамен е
JOIN Сотрудник с ON е.Таб_номер = с.Таб_номер
LEFT JOIN Преподаватель р ON с.Таб_номер = р.[Таб_номер]
WHERE е.Дата BETWEEN '2015-06-05' AND '2015-06-12'
ORDER BY е.Дата;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Код_дисциплины	Фамилия	Должность	Степень	Дата	Аудитория
1	102	Семенов С.С.	преподаватель	к.ф.-м.н.	2015-06-05	т505
2	102	Семенов С.С.	преподаватель	к.ф.-м.н.	2015-06-05	т505
3	202	Борисов Б.Б.	преподаватель	к.ф.-м.н.	2015-06-05	т506
4	202	Борисов Б.Б.	преподаватель	к.ф.-м.н.	2015-06-05	т506
5	102	Петров П.П.	преподаватель	к.т.н.	2015-06-07	ф419
6	102	Прохоров П.П.	зав.кафедрой	д.т.н.	2015-06-07	т506
7	102	Семенов С.С.	преподаватель	к.ф.-м.н.	2015-06-07	м425
8	205	Зайцев З.З.	преподаватель	к.т.н.	2015-06-09	м424
9	209	Сергеев С.С.	преподаватель	к.т.н.	2015-06-09	ф333
10	101	Кузнецов К.К.	зав.кафедрой	д.ф.-м.н.	2015-06-10	т506

11.

```sql
SELECT с.Фамилия, COUNT(е.ID) AS Количество_экзаменов
FROM Сотрудник с
JOIN Экзамен е ON с.Таб_номер = е.Таб_номер
GROUP BY с.Фамилия
HAVING COUNT(е.ID) > 3;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Фамилия	Количество_экзаменов
1	Кузнецов К.К.	4
2	Семенов С.С.	4

12.

```sql
SELECT s.Рег_номер, s.Фамилия
FROM Студент s
WHERE s.Рег_номер NOT IN (
    SELECT e.Рег_номер
    FROM Экзамен e
    WHERE e.Дата = '2015-06-05
);
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Рег_номер	Фамилия
1	30101	Смирнов
2	30102	Попов
3	50101	Васильев
4	50102	Николаев
5	80101	Александров
6	80102	Дмитриев

Лабораторная работа № 7

Объединение результатов нескольких запросов

1. Цель работы

1. Изучить основы объединения результатов запросов.
2. Изучить UNION.
3. Изучить EXCEPT.
4. Изучить INTERSECT.

ПРИМЕРЫ

1.

```
        ,Население
        ,Континент
FROM
        Страна
WHERE
        Площадь > 1000000
UNION
SELECT
            Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
FROM
        Страна
WHERE
        Население > 100000000
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Столица	Площадь	Население	Континент
1	Австралия	Канберра	7692000	25600000	Австралия
2	Алжир	Алжир	2381740	39813722	Африка
3	Ангола	Луанда	1246700	25831000	Африка
4	Аргентина	Буэнос-Айрес	2766890	43847000	Южная Америка
5	Боливия	Сукре	1098580	10985059	Южная Америка
6	Бразилия	Бразилиа	8516000	212600000	Южная Америка
7	Индия	Нью-Дели	3287000	1380000000	Азия
8	Китай	Пекин	9597000	1412000000	Азия
9	Россия	Москва	17100000	146700000	Европа/Азия
10	США	Вашингтон	9834000	331000000	Северная Америка
11	Япония	Токио	377975	125800000	Азия

2.

```
        ,Население
        ,Континент
FROM
        Страна
WHERE
        Площадь > 1000000
UNION ALL SELECT
            Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
FROM
        Страна
WHERE
        Население > 100000000
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Столица	Площадь	Население	Континент
1	Австралия	Канберра	7692000	25600000	Австралия
2	Алжир	Алжир	2381740	39813722	Африка
3	Ангола	Луанда	1246700	25831000	Африка
4	Аргентина	Буэнос-Айрес	2766890	43847000	Южная Америка
5	Боливия	Сукре	1098580	10985059	Южная Америка
6	Бразилия	Бразилиа	8516000	212600000	Южная Америка
7	Индия	Нью-Дели	3287000	1380000000	Азия
8	Китай	Пекин	9597000	1412000000	Азия
9	Россия	Москва	17100000	146700000	Европа/Азия
10	США	Вашингтон	9834000	331000000	Северная Америка
11	Бразилия	Бразилиа	8516000	212600000	Южная Америка

3.

```sql
SELECT
        Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
        Страна
WHERE
        Континент = 'Европа' AND
        CAST(Население AS FLOAT) / Площадь > 400
UNION
SELECT
        Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Индия	Нью-Дели	3287000	1380000000	Азия
2	Япония	Токио	377975	125800000	Азия
3	Германия	Берлин	35788	83100000	Европа

4.

```sql
SELECT
        Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
        Страна
WHERE
        Площадь > 1000000
EXCEPT
SELECT
        Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
```

81 % Проблемы не найдены.

Результаты | Сообщения

Название	Столица	Площадь	Население	Континент

5.

```sql
        Площадь > 1000000
INTERSECT
SELECT
        Название
       ,Столица
       ,Площадь
       ,Население
       ,Континент
FROM
        Страна
WHERE
        Население > 100000000
```

81 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Бразилия	Бразилиа	8516000	212600000	Южная Америка
2	Индия	Нью-Дели	3287000	1380000000	Азия
3	Китай	Пекин	9597000	1412000000	Азия
4	Россия	Москва	17100000	146700000	Европа/Азия
5	США	Вашингтон	9834000	331000000	Северная Америка

4 задание

1.

```sql
SELECT Название, Площадь, Население, Континент
FROM Страна
WHERE Площадь < 500
UNION ALL
SELECT Название, Площадь, Население, Континент
FROM Страна
WHERE Площадь > 5000000
ORDER BY Площадь;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Площадь	Население	Континент
1	Науру	21	10824	Океания
2	Тувалу	26	11792	Океания
3	Маршалловы Острова	181	58413	Океания
4	Палау	459	17901	Океания
5	Австралия	7692000	25600000	Австралия
6	Бразилия	8516000	212600000	Южная Америка
7	Китай	9597000	1412000000	Азия
8	США	9834000	331000000	Северная Америка
9	Россия	17100000	146700000	Европа/Азия

2.

```sql
SELECT Название, Площадь, Население, Континент
FROM Страна
WHERE Площадь > 1000000
EXCEPT
SELECT Название, Площадь, Население, Континент
FROM Страна
WHERE Население < 100000000;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Площадь	Население	Континент
1	Бразилия	8516000	212600000	Южная Америка
2	Индия	3287000	1380000000	Азия
3	Китай	9597000	1412000000	Азия
4	Россия	17100000	146700000	Европа/Азия
5	США	9834000	331000000	Северная Америка

3.

```sql
SELECT Название, Площадь, Население, Континент
FROM Страна
WHERE Площадь < 500 AND Население < 100000;
```

81 % Проблемы не найдены.

Результаты Сообщения

	Название	Площадь	Население	Континент
1	Маршалловы Острова	181	58413	Океания
2	Науру	21	10824	Океания
3	Палау	459	17901	Океания
4	Тувалу	26	11792	Океания

Лабораторная работа № 8

Подзапросы

1. Цель работы

1. Изучить виды вложенных запросов.
2. Изучить некоррелирующие подзапросы.
3. Изучить коррелирующие подзапросы.
4. Изучить применение конструкции IN к подзапросам.
5. Изучить конструкцию ALL.
6. Изучить конструкцию ANY/SOME.
7. Изучить конструкцию EXISTS.

ПРИМЕРЫ

1

```
SELECT
            Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
            ,ROUND(CAST(Население AS FLOAT) * 100 /
            (
              SELECT
                  SUM(Население)
              FROM
                  Страна
            ), 3) AS Процент
FROM
        Страна
ORDER BY
        Процент DESC
```

Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент	Процент
1	Китай	Пекин	9597000	1412000000	Азия	36,671
2	Индия	Нью-Дели	3287000	1380000000	Азия	35,84
3	США	Вашингтон	9834000	331000000	Северная Америка	8,596
4	Бразилия	Бразилиа	8516000	212600000	Южная Америка	5,521
5	Россия	Москва	17100000	146700000	Европа/Азия	3,81
6	Япония	Токио	377975	125800000	Азия	3,267
7	Германия	Берлин	357588	83100000	Европа	2,158
8	Франция	Париж	643801	67000000	Европа	1,74
9	Великобритания	Лондон	242495	66650000	Европа	1,731

Запрос успешно выполнен. (localdb)\MSS

Список ошибок

2

2 присер.sql ...22\User (64)) | 1 пример.sql -..._22\User (52))

```sql
SELECT
            Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент        FROM
          Страна
WHERE
          Население > (
              SELECT
                  AVG(Население)
              FROM
                  Страна)
```

83 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Индия	Нью-Дели	3287000	1380000000	Азия
2	Китай	Пекин	9597000	1412000000	Азия

3

12 задачаа.sql..._22\User (74)) | 12 задача.sql ..._22\User (67)) | 10

```sql
SELECT
            Название
        ,Столица
        ,Площадь
        ,Население
            ,Континент
FROM
          (
          SELECT
              Название
        ,Столица
        ,Площадь
        ,Население
            ,Континент
          FROM
              Страны
```

83 % 13 0

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Алжир	Алжир	2381740	39813722	Африка
2	Ангола	Луанда	1246700	25831000	Африка
3	Бенин	Порто-Ново	112620	11167000	Африка
4	Буркина-Фасо	Уагадугу	274200	19034397	Африка

4

```sql
        ,Континент
            ,ROUND(CAST(Население AS FLOAT) * 100 /
            (
              SELECT
                  SUM(Население)
              FROM
                  Страна Б
              WHERE
                  А.Континент = Б.Континент
            ), 3) AS Процент
FROM
        Страна А
ORDER BY
        Процент DESC
```

Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент	Процент
1	Австралия	Канберра	7692000	25600000	Австралия	100
2	Бразилия	Бразилиа	8516000	212600000	Южная Америка	100
3	Россия	Москва	17100000	146700000	Европа/Азия	100
4	США	Вашингтон	9834000	331000000	Северная Америка	100
5	Китай	Пекин	9597000	1412000000	Азия	48,393
6	Индия	Нью-Дели	3287000	1380000000	Азия	47,296
7	Германия	Берлин	357588	83100000	Европа	38,339

5

```sql
SELECT
        Название
      ,Столица
      ,Площадь
      ,Население
      ,Континент        FROM
    Страна А WHERE
        Население > (
            SELECT
                AVG(Население)
            FROM
                Страна Б
            WHERE
                Б.Континент = А.Континент
                )
```

Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Индия	Нью-Дели	3287000	1380000000	Азия
2	Китай	Пекин	9597000	1412000000	Азия
3	Германия	Берлин	357588	83100000	Европа

6

```sql
        Название
      ,Столица
      ,Площадь
      ,Население
      ,Континент
FROM
    Страна  WHERE
        Континент IN (
            SELECT
            Континент        FROM
                Страна
            GROUP BY
            Континент        HAVING
                AVG(Население) > (
                    SELECT
                        AVG(Население)
                    FROM
                        Страна
                    )
        )
```

1 0

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Индия	Нью-Дели	3287000	1380000000	Азия
2	Китай	Пекин	9597000	1412000000	Азия
3	Япония	Токио	377975	125800000	Азия

7

SQLQuery16.s...2\User (61))* | SQLQuery15.s...22\User (58)

```sql
SELECT
        Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент
FROM
        Страна
WHERE
        Континент = 'Азия'
        AND
        Население > ALL (
            SELECT
            Население        FROM
            Страна           WHERE
                Континент = 'Европа'
```

83 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Индия	Нью-Дели	3287000	1380000000	Азия
2	Китай	Пекин	9597000	1412000000	Азия
3	Япония	Токио	377975	125800000	Азия

8

```sql
SELECT
        Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент
FROM
        Страны
WHERE
        Континент = 'Европа'
        AND
        Население > ANY (
            SELECT
            Население        FROM
                Страны
            WHERE
                Континент = 'Южная Америка'
        )
```

% 11 0

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
	Бельгия	Брюссель	30528	11250585	Европа
2	Великобритания	Лондон	244820	65341183	Европа

9

12 задачаа.sql..._22\User (74)) | 3 пример.sql -..._22\User (65)) | 8 пример.s

```sql
SELECT
        Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент
FROM
        Страны
WHERE
        Континент = 'Африка'
        AND
        EXISTS (
            SELECT
                *
            FROM
            Страны          WHERE
                Континент = 'Африка'
                AND
                Население > 10000000
        )
```

83 % | 10 | 0

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Алжир	Алжир	2381740	39813722	Африка
2	Ангола	Луанда	1246700	25831000	Африка
3	Бенин	Порто-Ново	112620	11167000	Африка
4	Ботсвана	Габороне	600370	2209208	Африка

Запрос успешно выполнен. | (localdb)\M

10

8 пример.sql -..._22\User (62)) | 9 пример.sql -..._22\User (54))

```sql
SELECT
        Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент
FROM
    Страны  WHERE
        Континент = (
            SELECT
            Континент       FROM
            Страны      WHERE
                Название = 'Вьетнам  '
        )
```

83 % 10 0

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Азербайджан	Баку	86600	9705600	Азия
2	Афганистан	Кабул	647500	29822848	Азия
3	Бангладеш	Дакка	144000	160221000	Азия
4	Бахрейн	Манама	701	1397000	Азия
5	Бутан	Тхимпху	47000	784000	Азия
6	Восточный Тимор	Дили	14874	1167242	Азия
7	Вьетнам	Ханой	329560	91713300	Азия

11

8 пример.sql -..._22\User (62)) | 9 пример.sql -..._22\User (54)) | 10 прим

```sql
SELECT
        Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент
FROM
    Страны  WHERE
        Население !> (
            SELECT
            Население       FROM
            Страны          WHERE
                Название = 'Болгария'
        )
```

83 % 10 0

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Албания	Тирана	28748	2866026	Европа
2	Бахрейн	Манама	701	1397000	Азия
3	Белиз	Бельмопан	22966	377968	Северная Америка
4	Болгария	София	110910	7153784	Европа
5	Ботсвана	Габороне	600370	2209208	Африка
6	Бутан	Тхимпху	47000	784000	Азия
7	Восточный Тимор	Дили	14874	1167242	Азия

12

SQLQuery21.s...2\User (55))* | 11 пример.sql ..._22\User (72)) | 10 пр

```sql
SELECT
        Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент
FROM
    Страна  WHERE
        Население = (
            SELECT
                MAX(Мин_Нас)
            FROM
                (
                    SELECT
                        MIN(Население) AS Мин_Нас
                    FROM
                    Страна                  GROUP BY
                        Континент
                ) A
        )
```

83 % Проблемы не найдены.

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	США	Вашингтон	9834000	331000000	Северная Америка

4 задания

1

SQLQuery22.s...2\User (63))* | 12 пример.sql ..._22\User (55))

```sql
SELECT
    Название,
    ROUND((Площадь * 100.0) / (SELECT SUM(Площадь)
    FROM Страна), 2)
    AS Процент_от_общей_площади
FROM Страна
ORDER BY Процент_от_общей_площади DESC;
```

83 % Проблемы не найдены.

Результаты | Сообщения

	Название	Процент_от_общей_площади
1	Россия	29,66
2	США	17,06
3	Китай	16,65
4	Бразилия	14,77
5	Австралия	13,34
6	Индия	5,7
7	Франция	1,12
8	Япония	0,66
9	Германия	0,62
10	Великобритания	0,42

2

SQLQuery23.s...2\User (56))* | 1 задача.sql -..._22\User (63)) | 12 пример.sql ..._22\Us

```sql
SELECT
    Название,
    Площадь,
    Население,
    ROUND(Население / Площадь, 2) AS Плотность
FROM Страна
WHERE
    Население / Площадь > (SELECT AVG(Население / Площадь) FROM Страна)
    AND Площадь > 0  -- избегаем деления на 0
ORDER BY Плотность DESC;
```

83 % Проблемы не найдены.

Результаты | Сообщения

	Название	Площадь	Население	Плотность
1	Индия	3287000	1380000000	419,84

3

S_BMESH_22\L...- dbo.Страны | S_BMESH_22\L...-

```sql
SELECT
    Название,
    Население
FROM Страны
WHERE
    Континент = 'Европа'
    AND Население < 5000000;
```

3 % 5 0

Результаты | Сообщения

	Название	Население
1	Албания	2866026

4

SQLQuery25.s...2\User (66))* | S_BMESH_22\L...- dbo.Страны

```sql
SELECT
    А.Название,
    А.Континент,
    ROUND((А.Площадь * 100.0) / (
        SELECT SUM(Б.Площадь)
        FROM Страны Б
        WHERE Б.Континент = А.Континент
    ), 2) AS Процент_от_континента
FROM Страны А
ORDER BY А.Континент, Процент_от_континента DESC;
```

83 % 9 0

Результаты | Сообщения

	Название	Континент	Процент_от_континента
1	Афганистан	Азия	50.970000000000
2	Вьетнам	Азия	25.940000000000
3	Бангладеш	Азия	11.340000000000
4	Азербайджан	Азия	6.820000000000
5	Бутан	Азия	3.700000000000
6	Восточный Тимор	Азия	1.170000000000
7	Бахрейн	Азия	0.060000000000
8	Алжир	Африка	51.600000000000
9	Ангола	Африка	27.010000000000
10	Ботсвана	Африка	13.010000000000
11	Буркина-Фасо	Африка	5.940000000000

5

QLQuery26.s...2\User (68))* | задача 5.sql -..._22\User (6

```sql
SELECT
    А.Название,
    А.Континент,
    А.Площадь
FROM Страны А
WHERE
    А.Площадь > (
        SELECT AVG(Б.Площадь)
        FROM Страны Б
        WHERE Б.Континент = А.Континент
    )
ORDER BY А.Континент, А.Площадь DESC;
```

% 11 0

Результаты | Сообщения

	Название	Континент	Площадь
1	Афганистан	Азия	647500
2	Вьетнам	Азия	329560
3	Алжир	Африка	2381740
4	Ангола	Африка	1246700
5	Великобритания	Европа	244820
6	Белоруссия	Европа	207595
7	Бразилия	Южная Америка	8511965

6

```
9   Общемировая_плотность AS (
10      SELECT AVG(Население / Площадь) AS Мир_плот
11      FROM Страны
12      WHERE Площадь > 0
13  )
14  SELECT
15      С.Название,
16      С.Континент,
17      ROUND(С.Население / С.Площадь, 2) AS Плотно
18  FROM Страны С
19  JOIN Средняя_плотность_по_континентам СП
20      ON С.Континент = СП.Континент
21  JOIN Общемировая_плотность ОП
22      ON СП.Ср_плотность > ОП.Мир_плотность
23  WHERE С.Площадь > 0
24  ORDER BY С.Континент, Плотность DESC;
```

83 % 19 0

Результаты Сообщения

	Название	Континент	Плотность
1	Бахрейн	Азия	1992
2	Бангладеш	Азия	1112
3	Вьетнам	Азия	278
4	Азербайджан	Азия	112
5	Восточный Тимор	Азия	78
6	Афганистан	Азия	46
7	Бутан	Азия	16

7

```
2       Название,
3       Континент,
4       Население
5   FROM Страны
6   WHERE
7       Континент = 'Южная Америка'
8       AND Население > ALL (
9           SELECT Население
10          FROM Страны
11          WHERE Континент = 'Африка'
12      );
```

83 % 9 0

Результаты Сообщения

	Название	Континент	Население
1	Аргентина	Южная Америка	43847000
2	Бразилия	Южная Америка	206081432

8

```sql
SELECT
    Название,
    Континент,
    Население
FROM Страны
WHERE
    Континент = 'Африка'
    AND Население > ANY (
        SELECT Население
        FROM Страны
        WHERE Континент = 'Южная Америка'
    );
```

Результаты | Сообщения

	Название	Континент	Население
1	Алжир	Африка	39813722
2	Ангола	Африка	25831000
3	Бенин	Африка	11167000
4	Буркина-Фасо	Африка	19034397

9

```sql
SELECT *
FROM Страны
WHERE
    Континент = 'Африка'
    AND EXISTS (
        SELECT 1
        FROM Страны
        WHERE
            Континент = 'Африка'
            AND Площадь > 2000000
    );
```

Результаты | Сообщения

	Название	Столица	Площадь	Население	Континент
1	Алжир	Алжир	2381740	39813722	Африка
2	Ангола	Луанда	1246700	25831000	Африка
3	Бенин	Порто-Ново	112620	11167000	Африка
4	Ботсвана	Габороне	600370	2209208	Африка
5	Буркина-Фасо	Уагадугу	274200	19034397	Африка

10

```sql
SELECT
    Название,
    Континент
FROM Страны
WHERE
    Континент = (SELECT Континент
        FROM Страны
        WHERE Название = 'Фиджи')
    AND Название != 'Фиджи';
```

Название	Континент

11

```sql
SELECT
    Название,
    Население
FROM Страны
WHERE
    Население <= (
        SELECT Население
        FROM Страны
        WHERE Название = 'Фиджи'
    )
ORDER BY Население DESC;
```

Название	Население

12

SQLQuery33.s...2\User (74))* | 12 задача.sql ..._22\User (67)

```sql
    С.Название,
    С.Площадь,
    С.Население
FROM Страны С
JOIN Мин_площадь_по_континентам МП
    ON С.Континент = МП.Континент
    AND С.Площадь = МП.Мин_площадь
WHERE
    С.Население = (
        SELECT MAX(Население)
        FROM Страны
        WHERE
            Континент = С.Континент
            AND Площадь = МП.Мин_площадь
    )
ORDER BY С.Континент;
```

83 % | 19 | 0

Результаты | Сообщения

	Континент	Название	Площадь	Население
1	Азия	Бахрейн	701	1397000
2	Африка	Бенин	112620	11167000
3	Европа	Албания	28748	2866026
4	Северная Америка	Белиз	22966	377968
5	Южная Америка	Венесуэла	912050	31028637

Лабораторная работа № 9

Основы DDL

1. Цель работы

1. Изучить создание базы данных.
2. Изучить удаление базы данных.
3. Изучить создание таблиц.
4. Изучить удаление таблиц.

ПРИМЕРЫ

1.

```
CREATE TABLE Управление
(
 ID INT ,  Вид VARCHAR(20)
)
```

81 % Проблемы не найдены.

Сообщения

```
Выполнение команд успешно завершено.

Время выполнения: 2026-02-09T18:10:21.5439981+03:00
```

2.

```
DROP TABLE Управление
```

81 % 1 0

Сообщения

```
Выполнение команд успешно завершено.

Время выполнения: 2026-02-09T18:11:27.312
```

3.

```sql
CREATE TABLE Управление
(
        ID  INT UNIQUE,
        Вид     VARCHAR(20) NOT NULL
)
```

81 % Проблемы не найдены.

Сообщения

```
Выполнение команд успешно завершено.

Время выполнения: 2026-02-09T18:11:55.2037558+03:00
```

4.

SQLQuery3.sq...9\User (61))*

```sql
CREATE TABLE Управление
(
        ID  INT CHECK (ID < 200),
        Вид     VARCHAR(20) DEFAULT 'Президентская республика'
)
```

5.

SQLQuery3.sq...9\User (61))*

```sql
CREATE TABLE Управление
(
        ID  INT PRIMARY KEY IDENTITY(5,3),
        Вид     VARCHAR(20)
)
```

4 задание

1.

1.sql - (lo...\USER (113))*

```sql
CREATE TABLE Управление_Белов (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    Название_отдела NVARCHAR(100) NOT NULL,
    Руководитель NVARCHAR(100),
    Дата_создания DATE DEFAULT GETDATE(),
    Бюджет DECIMAL(15,2) DEFAULT 0.00
);
```

68 % Проблемы не найдены. Стр:

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2026-02-11T23:56:01.2893565+03:00

2.

2.sql - (lo...\USER (118))*

```sql
CREATE TABLE Страны_Белов (
    Код_страны CHAR(3) PRIMARY KEY,
    Название NVARCHAR(100) NOT NULL,
    Столица NVARCHAR(100),
    Площадь DECIMAL(12,2) CHECK (Площадь > 0),
    Население INT CHECK (Население >= 0),
    Континент NVARCHAR(50) NOT NULL,
    Дата_вступления_ООН DATE
);
```

68 % Проблемы не найдены. Стр: 2,

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2026-02-11T23:54:51.3058055+03:00

3.

3.sql - (lo...\USER (119))*

```sql
CREATE TABLE Цветы_Белов (
    ID INT UNIQUE,
    Код_цветка INT IDENTITY(1,1) PRIMARY KEY,
    Название NVARCHAR(100) NOT NULL,
    Семейство NVARCHAR(100),
    Класс NVARCHAR(50) DEFAULT 'Двудольные',
    Период_цветения NVARCHAR(50),
    Высота_см DECIMAL(6,2)
);
```

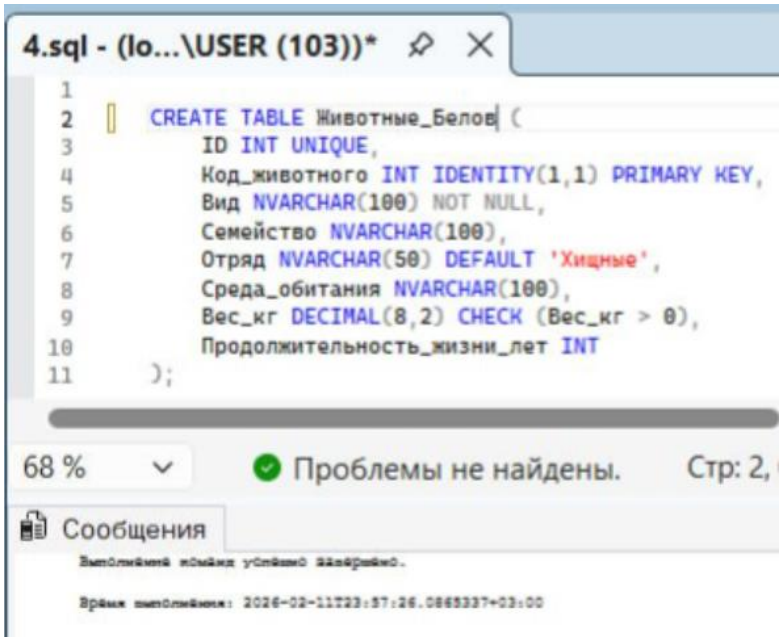
68 % Проблемы не найдены. Стр: 2, С

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2026-02-12T00:01:19.3540426+03:00

4.



Лабораторная работа № 10

Основы DML

1. Цель работы

1. Изучить команду INSERT.
2. Изучить команду UPDATE.
3. Изучить команду DELETE.
4. Изучить команду TRUNCATE TABLE.
5. Изучить конструкцию SELECT … INTO.

ПРИМЕРЫ

1.

```sql
INSERT INTO Ученик
    (Фамилия, Предмет, Школа, Баллы) VALUES
        ('Маркин', 'Физика', 'Гимназия', 96)
```

81 % Проблемы не найдены.

Сообщения

```
(Затронута 1 строка)

Время выполнения: 2026-02-09T18:22:49.7134785+03:00
```

2.

```sql
INSERT INTO Ученик
    (Фамилия, Предмет, Школа, Баллы) VALUES
        ('Никишина', 'Химия', 'Лицей', 77),
        ('Андреев', 'Математика', 'Школа №18', NULL)
```

81 % Проблемы не найдены.

Сообщения

```
(Затронуто строк: 2)

Время выполнения: 2026-02-09T18:24:25.4752867+03:00
```

3.

```sql
UPDATE
    Ученик  SET
        Баллы = 87
    WHERE
        Фамилия = 'Андреев'
```

81 % Проблемы не найдены.

Сообщения

```
(Затронута 1 строка)

Время выполнения: 2026-02-09T18:24:53.5985206+03:00
```

4.

```
UPDATE
        Ученик
    SET
        Школа = 'Школа №31', Предмет = 'Математика'
    WHERE
        Фамилия = 'Никишина'
```

81 % Проблемы не найдены.

Сообщения

```
(Затронуто строк: 2)

Время выполнения: 2026-02-09T18:25:49.0317712+03:00
```

5.

```
UPDATE
    Ученик  SET
        Баллы = Баллы - 5
    WHERE
        Предмет = 'Математика'
```

81 % Проблемы не найдены.

Сообщения

```
(Затронуто строк: 6)

Время выполнения: 2026-02-09T18:26:58.8823139+03:00
```

6.

```
DELETE FROM
        Ученик
    WHERE
        Школа = 'Школа №18'
```

81 % Проблемы не найдены.

Сообщения

```
(Затронута 1 строка)

Время выполнения: 2026-02-09T18:28:07.6137079+03:00
```

7.

```sql
SELECT
        ID
        ,Фамилия
        ,Предмет
        ,Школа
        ,Баллы
INTO
        Лицеисты
FROM
        Ученик
    WHERE
        Школа = 'Лицей'
```

81 % Проблемы не найдены.

Сообщения

```
(Затронуто строк: 7)

Время выполнения: 2026-02-09T18:28:32.7613725+03:00
```

8.

```sql
TRUNCATE TABLE Лицеисты
```

81 % Проблемы не найдены.

Сообщения

```
Выполнение команд успешно завершено.

Время выполнения: 2026-02-09T18:28:55.8134706+03:00
```

4 задание

1.

```sql
INSERT INTO Ученик (Фамилия, Предмет, Школа, Баллы)
VALUES ('Трошков', 'Химия', 'Школа %18', NULL);
```

81 % Проблемы не найдены.

Сообщения

```
(Затронута 1 строка)

Время выполнения: 2026-02-09T18:31:15.5855207+03:00
```

2.

```sql
INSERT INTO Ученик (Фамилия, Предмет, Школа, Баллы)
VALUES
    ('Смирнов', 'Физика', 'Лицей', 85),
    ('Кузнецова', 'Математика', 'Гимназия', 91),
    ('Попов', 'Химия', 'Школа №25', 73);
```

81 % Проблемы не найдены.

Сообщения

```
(Затронуто строк: 3)

Время выполнения: 2026-02-09T18:32:30.6985582+03:00
```

3.

```sql
UPDATE Ученик
SET
    Школа = 'Школа №21',
    Предмет = 'Математика',
    Баллы = 56
WHERE Фамилия = 'Трошков';
```

81 % Проблемы не найдены.

Сообщения

```
(Затронута 1 строка)

Время выполнения: 2026-02-09T18:33:06.9740670+03:00
```

4.

```sql
UPDATE Ученик
SET Баллы = Баллы * 1.1
WHERE Предмет = 'Химия' AND Баллы < 60;
```

81 % Проблемы не найдены.

Сообщения

```
(Затронуто строк: 0)

Время выполнения: 2026-02-09T18:33:38.9351300+03:00
```

5.

```
DELETE FROM Ученик
WHERE Школа = 'Школа %21';
```

81 % Проблемы не найдены.

Сообщения

```
(Затронута 1 строка)

Время выполнения: 2026-02-09T18:34:24.7517082+03:00
```

6.

```
CREATE TABLE Гимназисты (
    ID INT PRIMARY KEY IDENTITY(1,1),
    Фамилия NVARCHAR(50) NOT NULL,
    Предмет NVARCHAR(50),
    Баллы DECIMAL(5,2)
);

INSERT INTO Гимназисты (Фамилия, Предмет, Баллы)
SELECT Фамилия, Предмет, Баллы
FROM Ученик
WHERE Школа LIKE '%Гимназия%' AND Баллы >= 60;
```

81 % Проблемы не найдены.

Сообщения

```
(Затронуто строк: 5)

Время выполнения: 2026-02-09T18:35:00.4546172+03:00
```

7.

```
TRUNCATE TABLE Гимназисты;
```

31 % Проблемы не найдены.