

## ЛАБОРАТОРНАЯ РАБОТА №5

В рамках лабораторной работы №5 было необходимо изучить основные конструкции языка запросов SQL.

База данных, используемая в данной лабораторной, встречалась при выполнении лабораторных работ № 2, 4. Напомним её схему:

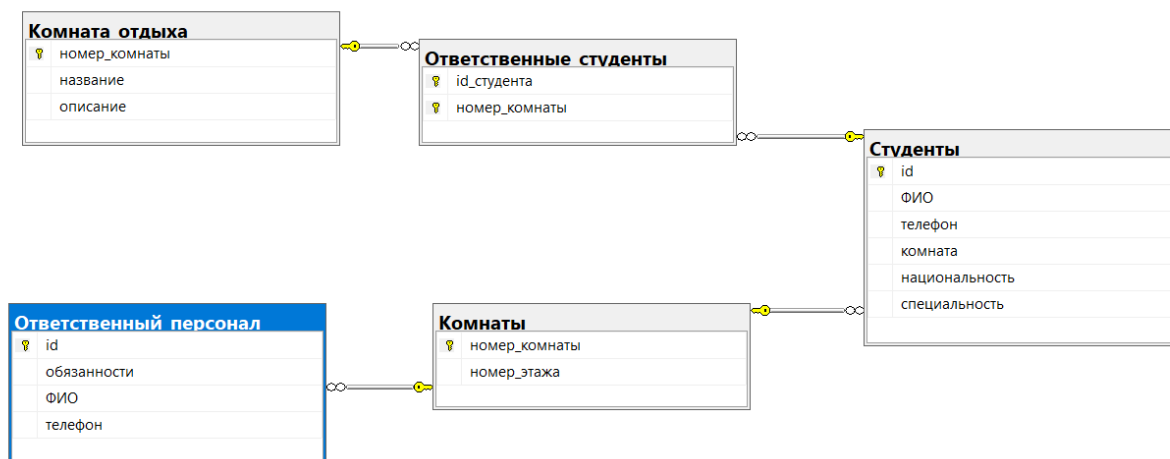


Рисунок 1. Модель данных из лабораторных работ № 2, 4

Теперь рассмотрим запросы, которые было необходимо написать в этой лабораторной:

```
-- Вычисляем номер этажа используя JOIN по номеру комнаты
SELECT s.id, s.ФИО, s.комната, k.номер_этажа
FROM dbo.Студенты s
JOIN Комнаты k ON s.комната = k.номер_комнаты
WHERE k.номер_этажа = 2
ORDER BY номер_комнаты ASC;
```

Код запроса, использующего INNER JOIN

Замечу, что необязательно писать ключевое слово **INNER**, т. к. по умолчанию будет использоваться именно он.

	id	ФИО	комната	номер_этажа
1	1	Иванов Иван Иванович	203	2
2	10	Смирнова Анастасия Сергеевна	204	2
3	2	Петров Петр Петрович	205	2
4	19	Титова Мария Сергеевна	205	2
5	11	Морозов Александр Владимирович	206	2
6	20	Сергеева Анастасия Александровна	207	2

Рисунок 2. Результат работы запроса, представленного выше

Далее было необходимо выполнить запрос с применением вычисляемых полей. Для этого давайте попытаемся вычислить то же самое, что и в прошлый раз – номер этажа, не используя JOIN. Это сделать не сложно, т. к.

все комнаты, находящиеся на 2 этаже будут иметь номер 2\*\*, на 3 – 3\*\* и т. д. Таким образом, вот как будет выглядеть наш запрос:

```
-- Вычисляем номер этажа без JOIN-а по номеру комнаты
SELECT s.id, s.ФИО, s.комната, (s.комната / 100) AS Этаж
FROM dbo.Студенты s
WHERE s.Комната >= 200
ORDER BY Этаж;
```

**Код запроса, использующего вычисляемые поля**

Результаты		Сообщения		
	id	ФИО	комната	Этаж
1	1	Иванов Иван Иванович	203	2
2	2	Петров Петр Петрович	205	2
3	10	Смирнова Анастасия Сергеевна	204	2
4	11	Морозов Александр Владимирович	206	2
5	19	Титова Мария Сергеевна	205	2
6	20	Сергеева Анастасия Александровна	207	2
7	12	Ковалева Екатерина Петровна	308	3
8	3	Сидорова Анна Павловна	307	3
9	4	Козлова Ольга Николаевна	408	4
10	13	Павлов Денис Алексеевич	409	4
11	14	Новикова Ольга Валерьевна	501	5
12	5	Михайлов Алексей Владимирович	510	5
13	6	Васильева Елена Сергеевна	609	6
14	15	Кузьмин Иван Дмитриевич	610	6
15	16	Волкова Анастасия Викторовна	708	7
16	7	Федоров Дмитрий Александрович	707	7
17	8	Александров Игорь Викторович	802	8

**Рисунок 3. Результат работы запроса, представленного выше**

Далее имеем несколько запросов, использующих внешние JOIN-ы и выборки:

```
-- Ответственные студенты ниже 5 этажа
SELECT *
FROM (
SELECT id, ФИО, комната, номер_комнаты AS комната_отдыха
FROM Студенты
LEFT JOIN Ответственные_студенты ON id = id_студента
) AS Студенты_Комнаты_Отдыха
WHERE комната < 500
ORDER BY комната_отдыха;
```

```
-- Список заселённых комнат ниже 5 этажа
SELECT номер_комнаты
FROM Комнаты
LEFT JOIN Студенты ON номер_комнаты = комната
GROUP BY номер_комнаты
HAVING номер_комнаты < 500
ORDER BY номер_комнаты;
```

**Код запроса, использующего LEFT JOIN и запроса, осуществляющего выборку данных после JOIN-а**

	id	ФИО	комната	комната_отдыха
1	10	Смирнова Анастасия Сергеевна	204	NULL
2	11	Морозов Александр Владимирович	206	NULL
3	12	Ковалева Екатерина Петровна	308	NULL
4	13	Павлов Денис Алексеевич	409	NULL
5	19	Титова Мария Сергеевна	205	NULL
6	20	Сергеева Анастасия Александровна	207	NULL
7	1	Иванов Иван Иванович	203	100
8	2	Петров Петр Петрович	205	101

	номер_комнаты
1	201
2	202
3	203
4	204
5	205
6	206
7	207
8	208

**Рисунок 4. Результаты работы запросов, представленных выше**

Затем по заданию предполагается написание запросов, изменяющих данные в таблицах, представляем их код *(все запросы написаны таким образом, чтобы не изменять уже загруженные данные, т. е. на любое действие предполагается выполнение затем обратного ему)*:

```
-- Удаляем существующую запись
DELETE FROM Комната_отдыха
WHERE название = 'Зона релаксации';

-- И вставляем её же
INSERT INTO Комната_отдыха (номер_комнаты, название, описание)
VALUES
(110, 'Зона релаксации', 'Зона релаксации создана для тех, кто ищет уютное место,
чтобы расслабиться и забыть о всех проблемах, окунувшись в атмосферу покоя и
умиротворения.');
```

```
-- Обновляем запись работника
UPDATE Ответственный_персонал
SET ФИО = 'Василевский Василий Васильевич', телефон = '+375292543299'
WHERE обязанности = 6;
```

```
-- Возвращаем на место запись
UPDATE Ответственный_персонал
SET ФИО = 'Васильева Елена Сергеевна', телефон = '+375291033298'
WHERE обязанности = 6;
```

#### **Код запросов, обновляющих данные в таблицах**

Далее давайте создадим таблицу на основе данных двух других таблиц. В принципе, подход к этой задаче может быть разным: можно использовать JOIN-ы, можно UNION-ы. В целом, ничего не поменяется, от того, какую выборку данных мы будем использовать, поэтому давайте создадим таблицу заселённых комнат с указанием количества проживающих в ней в данный момент студентов *(комнаты до 5 этажа, чтобы уменьшить выборку)*:

```
-- Создание таблицы заселённых комнат с количеством проживающих в них студентов
SELECT номер_комнаты, COUNT(комната) AS количество_заселенных
INTO Заселённые_комнаты
FROM Комнаты
LEFT JOIN Студенты ON номер_комнаты = комната
```

```
GROUP BY номер_комнаты
HAVING номер_комнаты < 500;
```

Код запроса, создающего таблицу заселённых комнат

Результаты			Сообщения
	номер_комнаты	количество_заселенных	
1	201	0	
2	202	0	
3	203	1	
4	204	1	
5	205	2	
6	206	1	
7	207	1	
8	208	0	
9	209	0	
10	210	0	
11	301	0	
12	302	0	
13	303	0	
14	304	0	
15	305	0	
16	306	0	

Рисунок 5. Результат работы запроса, представленного выше

UNION позволяет объединить результаты запросов, при этом удаляя и идентичные результаты. Например, у нас в таблицах *Студенты* и *Ответственный персонал* есть *Иванов Иван Иванович* с *id* равным 1 (если взглянуть на запросы на вставку в отчёте лабораторной работы №4 то это можно увидеть), выполнив следующий запрос:

```
-- Объединение запросов
SELECT id, ФИО
FROM Студенты
WHERE национальность = 'белорус'
UNION
SELECT id, ФИО
FROM Ответственный_персонал;
```

Код запроса, использующего UNION

Видим следующий результат:

	id	ФИО
1	1	Иванов Иван Иванович
2	2	Петров Петр Петрович
3	3	Сидорова Анна Павловна
4	4	Козлова Ольга Николаевна
5	5	Михайлов Алексей Владимирович
6	6	Васильева Елена Сергеевна
7	7	Федоров Дмитрий Александрович
8	8	Александров Игорь Викторович
9	9	Кузнецова Мария Игоревна
10	10	Смирнова Анастасия Сергеевна
11	12	Ковалева Екатерина Петровна
12	14	Новикова Ольга Валерьевна
13	16	Волкова Анастасия Викторовна
14	18	Карпова Анастасия Игоревна
15	19	Титова Мария Сергеевна

Рисунок 6. Результат работы запроса, представленного выше

Как видно из результата, *Иванов Иван Иванович* у нас один.

Теперь необходимо сделать подзапрос во фразе WHERE:

```
-- Выбор ответственных студентов
SELECT id, ФИО, комната
FROM Студенты
WHERE id IN (
SELECT id_студента
FROM Ответственные_студенты
);
```

Код запроса, с подзапросом

	id	ФИО	комната
1	1	Иванов Иван Иванович	203
2	2	Петров Петр Петрович	205
3	3	Сидорова Анна Павловна	307
4	4	Козлова Ольга Николаевна	408
5	5	Михайлов Алексей Владимирович	510

Рисунок 7. Результат работы запроса, с подзапросом

Далее необходимо написать запросы для создания сущностей БД. Все скрипты из данной части были написаны в лабораторной работе №4, вот пара из них:

```
-- Данные запросы были созданы в предыдущей лабораторной --
-- Создание таблицы "Ответственный персонал"
--CREATE TABLE Ответственный_персонал (
--    id BIGINT PRIMARY KEY,
--    обязанности TINYINT NOT NULL,
--    ФИО VARCHAR(100) NOT NULL,
--    телефон CHAR(13) NOT NULL,
--);

-- Создание уникального индекса для номера этажа
--CREATE UNIQUE INDEX уникальный_индекс_обязанностей
--ON Ответственный_персонал (обязанности)
```

Код запросов создания сущностей БД

И наконец последний запрос на создание представления будет выглядеть следующим образом:

```
-- Создание представления
CREATE VIEW Представление_Студенты_Комнаты AS
SELECT *
FROM Студенты
JOIN Комнаты ON комната = номер_комнаты
WHERE комната >= 500
```

Код запроса, создающего представление

	id	ФИО	телефон	комната	национальность	специальность	номер_комнаты	номер_этажа
1	5	Михайлов Алексей Владимирович	+375295678901	510	русский	Информационные технологии	510	5
2	6	Васильева Елена Сергеевна	+375296789012	609	белорус	Физика	609	6
3	7	Федоров Дмитрий Александрович	+375297890123	707	русский	Химия	707	7
4	8	Александров Игорь Викторович	+375298901234	802	украинец	Биология	802	8
5	9	Кузнецова Мария Игоревна	+375299012345	903	белорус	Психология	903	9
6	14	Новикова Ольга Валерьевна	+375293556677	501	белорус	Информационные технологии	501	5
7	15	Кузьмин Иван Дмитриевич	+375294667788	610	русский	Физика	610	6
8	16	Волкова Анастасия Викторовна	+375295778899	708	белорус	Химия	708	7
9	17	Егоров Игорь Александрович	+375296889900	803	украинец	Биология	803	8
10	18	Карпова Анастасия Игоревна	+375297900011	904	белорус	Психология	904	9

Рисунок 8. Результат работы запроса, создающего представление

Таким образом, в данной лабораторной работе я изучил основные конструкции языка SQL, столкнулся с основными проблемами, возникающими при разработке запросов, а также разобрал принципы действия некоторых сложных запросов.