

**Лабораторные работы по курсу
Базы данных**

**Лабораторная работа 3
«Использование объединяющих и вложенных запросов языка SQL»**

Москва, 2023

Оглавление

1. Теоретические сведения	3
1.1. Соединение таблиц	3
1.1.1. Неявное соединение таблиц	3
1.1.2. Соединение с помощью JOIN	4
1.1.3. Объединение, разность, пересечение таблиц	5
1.1.4. Пример соединения и объединения таблиц	6
1.2. Подзапросы.....	8
1.2.1. Некоррелированный запрос	8
1.2.2. Коррелированный запросы	9
2. Практическая часть	10
2.1. Задание 1.....	10
2.2. Задание 2.....	10
Список литературы.....	10

1. Теоретические сведения

В предыдущей лабораторной работе были рассмотрены примеры запросов на выборку данных из одной таблицы. Однако, существует большое число задач, когда требуется проанализировать информацию из нескольких таблиц. Для этого существуют операции соединения.

1.1. Соединение таблиц

1.1.1. Неявное соединение таблиц

Самым простым способом является неявное соединение таблиц, когда таблицы объединяются перекрестно. Другими словами, каждой строке одной таблицы будет совмещаться с каждой строкой второй таблицы. В данном случае мы получаем прямое (декартово) произведение двух таблиц.

На языке SQL для неявного соединения необходимо указать требуемые таблицы через запятую после оператора FROM.

Предположим, что нам необходимо вывести ФИО всех преподавателей и полное название подразделения, в котором они трудоустроены. Для этого соединим таблицы Структурное подразделение и Трудоустройство. Для этого возможно выполнить следующий запрос:

```
SELECT * FROM "Структурное подразделение", "Трудоустройство";
```

	Номер подразде bigint	Тип подразде character varying	Полное название text	Сокращенное названи character varying (20)	Руководитель character varying (40)	Номер струк bigint	Номер препс bigint	Номер трудо bigint	Ставка numeric (3,2)
1	1	Институт	Институт микро...	Институт МПСУ	Переверзев Алексей...	1	80001	152684	0.25
2	2	Институт	Институт систем...	Институт СПИНТех	Гагарина Лариса Ген...	1	80001	152684	0.25
3	3	Кафедра	Кафедра информ...	[null]	Хорев Анатолий Ана...	1	80001	152684	0.25
4	4	Кафедра	Кафедра инжене...	[null]	Соколова Татьяна Ю...	1	80001	152684	0.25
5	1	Институт	Институт микро...	Институт МПСУ	Переверзев Алексей...	2	80001	152685	0.25
6	2	Институт	Институт систем...	Институт СПИНТех	Гагарина Лариса Ген...	2	80001	152685	0.25
7	3	Кафедра	Кафедра информ...	[null]	Хорев Анатолий Ана...	2	80001	152685	0.25
8	4	Кафедра	Кафедра инжене...	[null]	Соколова Татьяна Ю...	2	80001	152685	0.25

В результате запроса будет выведены значения из двух таблиц, где каждому значению из второй будут соответствовать все значения из первой таблицы. Таким образом, будет выведено (число строк в первой таблице) * (число строк во второй таблице). В данном примере будет выведено 24 строки.

Аналогично добавим к данным двум таблицам ещё таблицу «Преподаватель».

```
SELECT * FROM "Структурное подразделение", "Трудоустройство", "Преподаватель"
```

	Номер подразде bigint	Тип подразде character varying	Полное название text	Сокращенное названи character varying	Руководитель character varying	Номер струк bigint	Номер препс bigint	Номер трудо bigint	Ставка numeric (3,2)	Номер препс bigint	Фамилия character varying	Имя character varying	Отчество character varying (3)
1	1	Институт	Институт микро...	Институт МП...	Переверзев...	1	80001	152684	0.25	80001	Петров	Иван	Иванович
2	1	Институт	Институт микро...	Институт МП...	Переверзев...	2	80001	152685	0.25	80001	Петров	Иван	Иванович
3	1	Институт	Институт микро...	Институт МП...	Переверзев...	1	80002	152986	0.25	80001	Петров	Иван	Иванович
4	1	Институт	Институт микро...	Институт МП...	Переверзев...	1	80003	153123	0.50	80001	Петров	Иван	Иванович
5	1	Институт	Институт микро...	Институт МП...	Переверзев...	3	80004	152273	0.60	80001	Петров	Иван	Иванович
6	1	Институт	Институт микро...	Институт МП...	Переверзев...	4	80005	152321	0.40	80001	Петров	Иван	Иванович
7	1	Институт	Институт микро...	Институт МП...	Переверзев...	1	80001	152684	0.25	80002	Попов	Андрей	Сергеевич
8	1	Институт	Институт микро...	Институт МП...	Переверзев...	2	80001	152685	0.25	80002	Попов	Андрей	Сергеевич

Очевидно, что такой результат запроса нас не устроит. Для того, чтобы отобразить только преподавателей, прикрепленных к месту работы, необходимо добавить условие отбора.

Выберем только те строки, где значения номера подразделения в таблицах «Структурное подразделение» и «Трудоустройство» и значения номера преподавателя в таблицах «Трудоустройство» и «Преподаватель» совпадают.

Итоговый запрос будет выглядеть следующим образом:

```
SELECT "Преподаватель"."Фамилия", "Преподаватель"."Имя",  
"Преподаватель"."Отчество", "Структурное подразделение"."Сокращенное название"  
FROM "Структурное подразделение", "Трудоустройство", "Преподаватель"
```

```
WHERE "Номер подразделения" = "Номер структурного подразделения" AND
"Трудоустройство"."Номер преподавателя" = "Преподаватель"."Номер
преподавателя"
```

	Фамилия character varying (30)	Имя character varying (30)	Отчество character varying (30)	Сокращенное название character varying (20)
1	Петров	Иван	Иванович	Институт МПСУ
2	Петров	Иван	Иванович	Институт СПИНТех
3	Попов	Андрей	Сергеевич	Институт МПСУ
4	Березина	Наталия	Владимировна	Институт МПСУ
5	Алексеев	Юрий	Алексеевич	[null]
6	Быкова	Елена	Николаевна	[null]

Для того, чтобы обращаться к атрибуту конкретной таблицы, необходимо указывать название таблицы, отделив его от названия атрибута точкой.

1.1.2. Соединение с помощью JOIN

Соединённая таблица — это таблица, полученная из двух других таблиц в соответствии с правилами соединения. Общий синтаксис описания соединённой таблицы:

```
T1 тип_соединения T2 [ условие_соединения ]
```

Кроме неявного соединения таблиц в языке SQL существует альтернативная форма записи операций соединения таблиц с помощью ключевого слова JOIN. Объединение происходит по столбцу, который есть в каждой из таблиц.

По типу соединения операторы JOIN подразделяются на внутренние и внешние — INNER JOIN и OUTER JOIN.

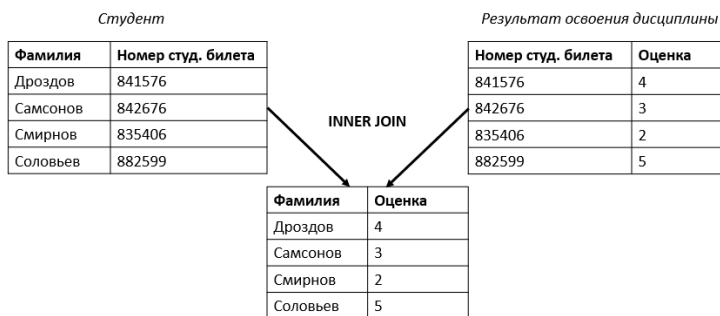
Внутреннее соединение - INNER JOIN используется для отбора строк из двух таблиц, в которых совпадают значения поля, по которому происходит объединение.

Например, объединим таблицы «Студент» и «Результат освоения дисциплины». Каждая из этих таблиц содержит поле студенческого билета. Произведем по этому полю соединение.

```
SELECT "Студент"."Фамилия", "Студент"."Имя", "Результат освоения
дисциплины"."Оценка"
FROM "Студент"
INNER JOIN "Результат освоения дисциплины"
ON "Студент"."Номер студенческого билета" = "Результат освоения
дисциплины"."Студент"
```

	Фамилия character varying (30)	Имя character varying (30)	Оценка bigint
1	Дроздов	Марк	4
2	Дроздов	Марк	3
3	Самсонов	Максим	3
4	Самсонов	Максим	2
5	Смирнов	Ярослав	5
6	Смирнов	Ярослав	4
7	Соловьев	Сергей	2
8	Соловьев	Сергей	3

В результате выполнения запроса на экран будет выведена информация об оценках всех студентов института.



Внешнее соединение - OUTER JOIN можно разделить на правое (RIGHT), левое (LEFT) и полное (FULL).

Операция левого внешнего соединения возвращает кроме обычного результата соединения, строки из левого операнда, для которых не нашлось парного значения строки в правом операнде. Вместо значений атрибутов правого операнда будет указано неопределенное значение NULL.

Аналогично, операция правого внешнего соединения возвращает все строки, для которых не нашлось пары в первом операнде.

Полное внешнее соединение включает в себя все пересекающиеся строки и все непарные строки из обеих таблиц.

Аналогично неявному соединению, возможно произвести декартово произведение таблиц с помощью оператора JOIN. Для этого существует ключевое слово CROSS.

С помощью операций соединения возможно соединять несколько таблиц. Рассмотрим это на примере более сложного запроса. Необходимо вывести всех преподавателей и структурное подразделение, к которым они прикреплены.

Данные о преподавателях хранятся в таблице «Преподаватель». Данные об структурных подразделениях – в таблице «Структурное подразделение». Общих столбцов у данных таблиц нет, поэтому необходимо соединить их вместе с промежуточной таблицей – «Трудоустройство».

```
SELECT "Преподаватель"."Фамилия", "Преподаватель"."Имя",
"Преподаватель"."Отчество", "Структурное подразделение"." Полное название"
FROM "Преподаватель"
INNER JOIN "Трудоустройство" ON
"Трудоустройство"."Номер преподавателя" = "Преподаватель"."Номер
преподавателя"
INNER JOIN "Структурное подразделение" ON
"Структурное подразделение"."Номер подразделения" = "Трудоустройство"."Номер
структурного подразделения"
```

	Фамилия character varying (30)	Имя character varying (30)	Отчество character varying (30)	Полное название text
1	Петров	Иван	Иванович	Институт микроприборов и систем управления имени Л.Н. Преснухина
2	Петров	Иван	Иванович	Институт системной и программной инженерии и информационных техноло...
3	Попов	Андрей	Сергеевич	Институт микроприборов и систем управления имени Л.Н. Преснухина
4	Березина	Наталья	Владимировна	Институт микроприборов и систем управления имени Л.Н. Преснухина
5	Алексеев	Юрий	Алексеевич	Кафедра информационной безопасности
6	Быкова	Елена	Николаевна	Кафедра инженерной графики и дизайна

1.1.3. Объединение, разность, пересечение таблиц

Кроме соединения таблиц, когда в результате операции атрибуты одной таблицы будут добавлены к атрибутам другой существуют операции **объединения**. В данном случае число атрибутов не изменяется, но в итоговой таблице будут содержаться значения из нескольких таблиц. При объединении таблиц необходимо соблюдать условие, что тип

данных каждого столбца первой таблицы должен совпадать с типом данных соответствующего столбца во второй таблице. Имена столбцов в объединяемых таблицах не обязательно должны быть одинаковыми.

В языке SQL для объединения таблиц используется оператор UNION.

```
запрос1 UNION [ALL] запрос2
```

Т.к. оператор UNION объединяет строки из двух таблиц результатов, в объединенной таблице могут содержаться повторяющиеся строки. По умолчанию оператор UNION в процессе своего выполнения удаляет повторяющиеся строки.

Кроме операции объединения существуют операции *пересечения* (INTERSECT) и *разности* (EXCEPT). Операция пересечения оставляет только общие строки из двух таблиц, а операция разности удаляет из первой таблицы значения, содержащиеся во второй. Данные три операции в языке SQL тесно связаны с логическими операциями булевой алгебры.

Операция	SQL операция	Логическая операция
Объединение	UNION	A OR B
Пересечение	INTERSECT	A AND B
Разность	EXCEPT	A AND NOT B

1.1.4. Пример соединения и объединения таблиц

Для закрепления материала рассмотрим еще один пример. Предположим, что имеются две таблицы, содержащие игроков двух команд. Два игрока входят в составы обеих из них. Применим к ним операции соединения.

Команда1		Команда2	
ID	Фамилия	ID	Фамилия
1	Герасимов	1	Лазарева
2	Полякова	2	Емельянов
3	Емельянов	3	Новиков
4	Миронова	4	Полякова

SQL запрос	Возвращенные значения		Комментарий
SELECT K1.Фамилия, K2.Фамилия FROM K1, K2;			Декартово произведение двух таблиц. Каждой игроку из первой команды сопоставляется каждый игрок из второй команды.
Или			
SELECT K1.Фамилия, K2.Фамилия FROM K1 CROSS JOIN K2			
SELECT K1.Фамилия, K2.Фамилия FROM K1 FULL OUTER JOIN K2 ON K1.Фамилия = K2.Фамилия			Отбор всех игроков из двух команд с возможностью совпадений записей в левой и правой таблицах. Если таковых нет, на пустой стороне вставляется NULL.

<pre>SELECT K1.Фамилия, K2.Фамилия FROM K1 LEFT OUTER JOIN K2 ON K1.Фамилия = K2.Фамилия</pre>	<table><thead><tr><th></th><th>Фамилия character varying (64)</th><th>Фамилия character varying (64)</th></tr></thead><tbody><tr><td>1</td><td>Герасимов</td><td>[null]</td></tr><tr><td>2</td><td>Полякова</td><td>Полякова</td></tr><tr><td>3</td><td>Емельянов</td><td>Емельянов</td></tr><tr><td>4</td><td>Миронова</td><td>[null]</td></tr></tbody></table>		Фамилия character varying (64)	Фамилия character varying (64)	1	Герасимов	[null]	2	Полякова	Полякова	3	Емельянов	Емельянов	4	Миронова	[null]	Отбираются все игроки из первой команды и игроки из второй, которые также играют за первую команду.
	Фамилия character varying (64)	Фамилия character varying (64)															
1	Герасимов	[null]															
2	Полякова	Полякова															
3	Емельянов	Емельянов															
4	Миронова	[null]															
<pre>SELECT K1.Фамилия, K2.Фамилия FROM K1 RIGHT OUTER JOIN K2 ON K1.Фамилия = K2.Фамилия</pre>	<table><thead><tr><th></th><th>Фамилия character varying (64)</th><th>Фамилия character varying (64)</th></tr></thead><tbody><tr><td>1</td><td>[null]</td><td>Лазарева</td></tr><tr><td>2</td><td>Емельянов</td><td>Емельянов</td></tr><tr><td>3</td><td>[null]</td><td>Новиков</td></tr><tr><td>4</td><td>Полякова</td><td>Полякова</td></tr></tbody></table>		Фамилия character varying (64)	Фамилия character varying (64)	1	[null]	Лазарева	2	Емельянов	Емельянов	3	[null]	Новиков	4	Полякова	Полякова	Отбираются все игроки из второй команды и игроки из первой, которые также играют за вторую команду.
	Фамилия character varying (64)	Фамилия character varying (64)															
1	[null]	Лазарева															
2	Емельянов	Емельянов															
3	[null]	Новиков															
4	Полякова	Полякова															
<pre>SELECT K1.Фамилия, K2.Фамилия FROM K1 INNER JOIN K2 ON K1.Фамилия = K2.Фамилия</pre>	<table><thead><tr><th></th><th>Фамилия character varying (64)</th><th>Фамилия character varying (64)</th></tr></thead><tbody><tr><td>1</td><td>Полякова</td><td>Полякова</td></tr><tr><td>2</td><td>Емельянов</td><td>Емельянов</td></tr></tbody></table>		Фамилия character varying (64)	Фамилия character varying (64)	1	Полякова	Полякова	2	Емельянов	Емельянов	Отбираются участники, играющие за обе команды						
	Фамилия character varying (64)	Фамилия character varying (64)															
1	Полякова	Полякова															
2	Емельянов	Емельянов															
<pre>SELECT Фамилия FROM K1 UNION SELECT Фамилия FROM K2</pre>	<table><thead><tr><th></th><th>Фамилия character varying (64)</th></tr></thead><tbody><tr><td>1</td><td>Лазарева</td></tr><tr><td>2</td><td>Герасимов</td></tr><tr><td>3</td><td>Полякова</td></tr><tr><td>4</td><td>Миронова</td></tr><tr><td>5</td><td>Новиков</td></tr><tr><td>6</td><td>Емельянов</td></tr></tbody></table>		Фамилия character varying (64)	1	Лазарева	2	Герасимов	3	Полякова	4	Миронова	5	Новиков	6	Емельянов	Объединение всех участников команд без повторяющихся значений.	
	Фамилия character varying (64)																
1	Лазарева																
2	Герасимов																
3	Полякова																
4	Миронова																
5	Новиков																
6	Емельянов																
<pre>SELECT Фамилия FROM K1 INTERSECT SELECT Фамилия FROM K2</pre>	<table><thead><tr><th></th><th>Фамилия character varying (64)</th></tr></thead><tbody><tr><td>1</td><td>Полякова</td></tr><tr><td>2</td><td>Емельянов</td></tr></tbody></table>		Фамилия character varying (64)	1	Полякова	2	Емельянов	Объединение всех участников команд.									
	Фамилия character varying (64)																
1	Полякова																
2	Емельянов																

<pre>SELECT Фамилия FROM K1 ЭКСПЕРТ SELECT Фамилия FROM K2</pre>	<table><tr><td></td><td>Фамилия character varying (64) </td></tr><tr><td>1</td><td>Герасимов</td></tr><tr><td>2</td><td>Миронова</td></tr></table>		Фамилия character varying (64)	1	Герасимов	2	Миронова	Выбор игроков, играющих за первую команду, но не играющих за вторую.
	Фамилия character varying (64)							
1	Герасимов							
2	Миронова							
<pre>SELECT Фамилия FROM K2 ЭКСПЕРТ SELECT Фамилия FROM K1</pre>	<table><tr><td></td><td>Фамилия character varying (64) </td></tr><tr><td>1</td><td>Лазарева</td></tr><tr><td>2</td><td>Новиков</td></tr></table>		Фамилия character varying (64)	1	Лазарева	2	Новиков	Выбор игроков, играющих за вторую команду, но не играющих за первую.
	Фамилия character varying (64)							
1	Лазарева							
2	Новиков							

1.2. Подзапросы

Результатом выполнения запроса является набор кортежей, оформленный в виде таблицы. Данную таблицу возможно вывести на экран для просмотра или использовать в других запросах. В таком случае, запрос используемый внутри другого запроса называют вложенным запросом или подзапросом.

Существуют два типа подзапросов:

Некоррелированный подзапрос – оператор SELECT вложенный в другой запрос SQL, не связанный с внешним запросом (он может быть выполнен отдельно от него).

Коррелированный подзапрос – оператор SELECT вложенный в другой запрос SQL, и ссылающийся на один или несколько столбцов внешнего запроса.

Рассмотрим типы запросов на примерах.

1.2.1. Некоррелированный запрос.

Выведем всех преподавателей, чей оклад больше среднего. Для этого создадим подзапрос, вычисляющий среднюю зарплату всех преподавателей вуза. Далее, используя скалярный результат этого подзапроса найдем все большие значения.

```
SELECT
"Преподаватель"."Фамилия", "Преподаватель"."Имя", "Преподаватель"."Должность"
FROM "Преподаватель"
WHERE "Преподаватель"."Оклад"::numeric >
(
    SELECT AVG("Преподаватель"."Оклад"::numeric)
    FROM "Преподаватель"
)
```

	Фамилия character varying (30)	Имя character varying (30)	Должность character varying (20)
1	Попов	Андрей	профессор
2	Березина	Наталия	профессор
3	Wittgenstein	Ludwig	профессор

Выведем всех студентов, которые имеют хотя бы одну оценку 2 за один из экзаменов. Для этого создадим подзапрос, выбирающих всех двоечников и используя их номера студенческого билета найдем их фамилию и имя.

```
SELECT "Фамилия", "Имя"
FROM "Студент"
WHERE "Студент"."Номер студенческого билета" IN(
    SELECT "Результат освоения дисциплины"."Студент"
```



```
FROM "Результат освоения дисциплины"
WHERE "Результат освоения дисциплины"."Оценка" = 2
)
```

1.2.2. Коррелированный запросы

Создадим аналогичный запрос с поиском всех студентов, имеющих хотя бы одну оценку 2, но с помощью коррелированного запроса. Обратите внимание, что во вложенном запросе происходит обращение к таблице «Студент», не указанной после ключевого слова FROM данного подзапроса. Если мы попытаемся запустить отдельно от основного данный подзапрос, то в результате будет получена ошибка.

```
SELECT "Фамилия", "Имя"
FROM "Студент"
WHERE 2 = ANY
(
    SELECT "Результат освоения дисциплины"."Оценка"
    FROM "Результат освоения дисциплины"
    WHERE "Результат освоения дисциплины"."Студент" = "Студент"."Номер
студенческого билета"
)
```

	Фамилия character varying (30)	Имя character varying (30)
1	Самсонов	Максим
2	Соловьев	Сергей
3	Маслов	Михаил
4	Коровина	Мария
5	Сахарова	Милана
6	Родионова	Арина
7	Королева	Ксения

Создадим запрос, выводящий средний балл каждого из студентов. Для этого создадим подзапрос, вычисляющий средний балл и ссылающийся на атрибут "Студент"."Номер студенческого билета" из внешнего запроса.

```
SELECT "Фамилия", "Имя", (
    SELECT AVG("Результат освоения дисциплины"."Оценка")
    FROM "Результат освоения дисциплины"
    WHERE "Результат освоения дисциплины"."Студент" = "Студент"."Номер
студенческого билета"
) AS "Средняя оценка"
FROM "Студент"
```

	Фамилия character varying (30)	Имя character varying (30)	Средняя оценка numeric
1	Дроздов	Марк	3.5000000000000000
2	Самсонов	Максим	2.5000000000000000
3	Смирнов	Ярослав	4.5000000000000000
4	Соловьев	Сергей	2.5000000000000000
5	Селиванов	Дмитрий	4.0000000000000000
6	Егоров	Артём	4.0000000000000000
7	Грачев	Андрей	3.5000000000000000
8	Маслов	Михаил	2.5000000000000000

2. Практическая часть

2.1.Задание 1.

Разработайте 5 **осмысленных** запросов к базе данных, используя приведенные в данной лабораторной работе материалы. Вариант выбирается в соответствии с номером по списку.

	Вариант 1					Вариант 2					Вариант 3					Вариант 4				
Ключевое слово	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
INNER JOIN	+	+				+	+				+	+				+	+			
LEFT OUTER JOIN			+					+												
RIGHT OUTER JOIN													+					+		
UNION				+					+					+					+	
EXCEPT					+					+										
INTERSECT															+					+

2.2. Задание 2.

Разработайте 4 **осмысленных** запроса к базе данных, 2 из которых внутри себя содержать коррелированный подзапрос и 2 некоррелированный.

Список литературы

- [1] Документация к PostgreSQL 15.1, 2022.
- [2] «Исходный код СУБД postgres,» [В Интернете]. Available: <https://github.com/postgres/postgres>. [Дата обращения: 30 01 2023].
- [3] Е. Рогов, PostgreSQL изнутри, 1-е ред., Москва: ДМК Пресс, 2023, р. 662 .
- [4] Б. А. Новиков, Е. А. Горшкова и Н. Г. Графеева, Основы технологии баз данных, 2-е ред., Москва: ДМК пресс, 2020, р. 582.
- [5] Е. П. Моргунов, PostgreSQL. Основы языка SQL, 1-е ред., Санкт-Петербург: БХВ-Петербург, 2018, р. 336.