

Современные промышленные СУБД Лабораторная работа №4

Лабораторная работа №4 Выборка данных из нескольких таблиц

Цель: научить использовать конструкцию JOIN для внешнего и внутреннего соединения двух или более таблиц.

Теоретический материал: перед выполнением лабораторной работы рекомендуется изучить лекцию №4 «Выборка данных», в которой изучается один из основных операторов языка SQL – SELECT, и объясняется назначение каждого из его разделов, а также рассматриваются вопросы соединения нескольких таблиц.

Требования к отчету: по результатам работы представить набор SQL-скриптов, решающих задачи из раздела «Самостоятельная работа».

Задание 1. Получите список товаров с указанием их категорий.

Указания к выполнению:

- 1. Так как в таблице *product* нормализованной базы данных хранится только идентификатор категории товара, то чтобы получить список товаров с указанием их категорий, необходимо соединить таблицы *product* и *type_product*.
- 2. Перед указанием поля в операторе **SELECT** необходимо будет использовать имена таблиц *product* и *type_product* для разрешения конфликтов, т.к. в обеих таблицах присутствуют поля id_type .

SELECT product.name, type_product.type, product.price FROM product JOIN type_product ON product.id_type = type_product.id_type;

Результат выполнения запроса показан на рис. 1.

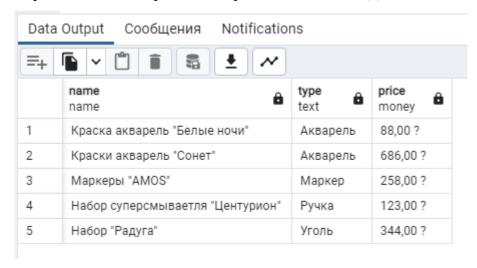


Рис. 1. Результат соединения двух таблиц – товары с указанием категории

Замечание. При таком соединении товары, для которых не указана их категория, не включаются в набор результатов.

Можно использовать «JOIN» вместо «INNER JOIN», так же получаемые таблицы одинаковы при любом варианте)

Задание 2. Создайте запрос, возвращающий имя заказчика, дату размещения заказа и наименование товара.

Указания к выполнению:

1. Выполните следующий код:

SELECT client.surname, client.name, client.patronymic, order_client.date_order, product.name
FROM client
LEFT JOIN order_client
ON client.id_client = order_client.id_client
LEFT JOIN product
ON order_client.id_product = product.id_product
ORDER BY date order

Результат выполнения запроса показан на рис. 2.

Data	Output Coo	бщения Not	tifications		
=+	• • •		~		
	surname name	name name	patronymic name	date_order a	name name
1	Сидоров	Дмитрий	Константинович	2019-04-03	Краска акварель "Белые ночи"
2	Петров	Алексей	Александрович	2019-11-29	Набор "Радуга"
3	Борисова	Анна	Николаевна	2021-05-14	Маркеры "AMOS"
4	Смит	Ричард	[null]	2021-12-12	Набор суперсмываетля "Центурион"
5	Стоун	Брэд	[null]	2022-05-23	Краска акварель "Белые ночи"
6	Кузнечиков	Дмитрий	Константинович	[null]	[null]
7	Жукова	Валерия	Васильевна	[null]	[null]
8	Уткина	Валерия	Александровна	[null]	[null]
9	Петров	Генадий	Федорович	[null]	[null]
10	Павлушин	Константин	Сергеевич	[null]	[null]

Рис. 2 Результат соединения трех таблиц (левое соединение) – клиенты, даты размещения ими заказов и наименование товара

Для лиц, не размещавших заказы, в полях *date_order* и *name* содержится значение **NULL**.

При *правом соединении* (ключевое слово **RIGHT JOIN**) в результат включаются все записи правой таблицы, независимо от того, есть для них соответствующая строка в левой таблице или нет (рис. 3). Измените рассмотренный запрос так, чтобы он выдавал такие же результаты при использовании левого соединения.

	name	name name	patronymic name	date_order date	name name
1	Сидоров	Дмитрий	Константинович	2019-04-03	Краска акварель "Белые ночи"
2	Петров	Алексей	Александрович	2019-11-29	Набор "Радуга"
3	Борисова	Анна	Николаевна	2021-05-14	Маркеры "AMOS"
4	Смит	Ричард	[null]	2021-12-12	Набор суперсмываетля "Центурион"
5	Стоун	Брэд	[null]	2022-05-23	Краска акварель "Белые ночи"
6	[null]	[null]	[null]	[null]	Краски акварель "Сонет"

Рис. 3 Результат соединения таблиц (правое соединение) – клиенты и даты

размещения ими заказов

Задание 3. С помощью перекрестного соединения можно перечислить все возможные категории продуктов в базе данных.

Оператор перекрёстного соединения, или декартова произведения CROSS JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является коммутативным.

Заголовок таблицы-результата является объединением (конкатенацией) заголовков соединяемых таблиц.

Тело результата логически формируется следующим образом. Каждая строка одной таблицы соединяется с каждой строкой второй таблицы, давая тем самым в результате все возможные сочетания строк двух таблиц.

Указания к выполнению:

SELECT DISTINCT product.id_product, type_product.type FROM product CROSS JOIN type_product

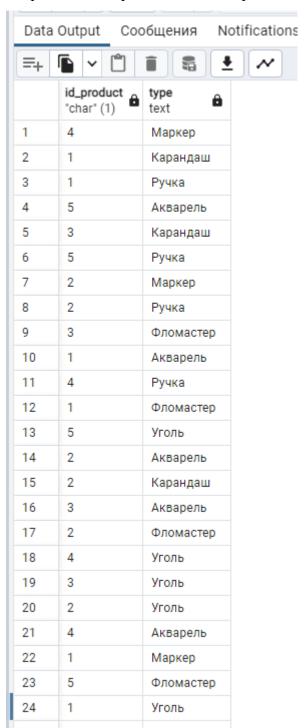


Рис.4 Результат перекрестного соединения двух таблиц.

Самостоятельная работа

- 1. Сконструируйте 3 запроса с использованием оператора Where и оператора Inner Join для базы данных, созданной в первой лабораторной работе.
- 2. Сконструируйте 2 запроса с использованием оператора Where.
- 3. Сконструируйте 2 запроса на объединение таблиц (обычное, левостороннее, правостороннее).

- 4. Предложите вариант перекрестного соединения, реализуйте его на SQL.
- 5. Сконструируйте 2 запроса с использованием подзапроса в условии.
- 6. Сконструируйте 2 запроса на объединение таблицы с собой.