Лабораторная работа 4
Многотабличные запросы

Реляционная база данных представляет собой совокупность взаимосвязанных таблиц. При решении большинства задач обработки данных необходимо извлекать информацию из нескольких таблиц. Запросы, в которых используется несколько таблиц, называют многотабличными.

Полное имя столбца состоит из имени таблицы и имени столбца, между которыми стоит точка, например **Employees.employee_id**. Если столбцы, разных таблиц, используемых в многотабличных запросах, имеют одинаковые имена, то необходимо указывать полные имена.

Условия соединения могут быть заданы или в предложении **WHERE** или в предложении **FROM**.

Условия соединения в предложении WHERE

Условия соединения в предложении WHERE в общем виде могут быть записаны следующим образом:

```
SELECT {список столбцов}
FROM  Таблица1, Таблица2, [Таблица3]……
WHERE Таблица1.столбец <операция соединения>
[ AND Таблица2.столбец < операция соединения> >
Таблица3.столбец]
```

Запрос 1. Вывести номера и названия отделов, расположенных в городе London**.**

```
SELECT  department_id, department_name, l.location_id
FROM  Departments d, Locations l
WHERE d.location_id=l.location_id
AND city = 'London';
```

В этом запросе кроме добавления условия соединения использованы псевдонимы таблиц. Так как столбец location_id есть в обеих таблицах то необходимо использовать полное имя l.location_id, l – псевдоним таблицы Locations.

Соединение по неэквивалентности.

В соединениях по неэквивалентности вместо операции = используются операции > ,< , BETWEEN и др.

Запрос 2. Для каждого сотрудника определить номера и названия товаров, которые он имеет право продавать. Это право определяется следующим правилом: рейтинг сотрудника должен быть больше или равен рейтинга товара.

```
SELECT employee_id, product_id, product_name, rating_e, rating_p
FROM Products, Employees
WHERE  rating_e >= rating_p
ORDER BY employee_id;
```

Условия соединения в предложении FROM

Для соединения таблиц в предложении **FROM** используется оператор **JOIN,** который имеет следующий синтаксис:

{таблица 1}{тип соединения} JOIN {таблица 2}
{условие соединения}

Таблицу расположенную слева от оператора JOIN ({таблица 1}) будем называть левой таблицей, а таблицу расположенную справа от оператора JOIN ({таблица 2}) будем называть правой таблицей. Можно создавать два типа соединений: внутренние и внешние.

Внутренние соединения

При использовании внутреннего соединения запрос будет выводить только те строки левой таблицы, которые имеют связанные строки в правой таблице.
Есть несколько вариантов определения внутреннего соединения в предложении JOIN.
Оператор **NATURAL JOIN** (Естественное соединение) имеет следующий синтаксис:

```
SELECT <список столбцов>
FROM <таблица 1> NATURAL JOIN <таблица 2>
```

Этот оператор соответствует операции соединения реляционной алгебры. При использовании этого оператора необходимо чтобы соединяемые таблицы имели один или несколько одноименных столбцов. Строки левой таблицы соединяются с теми строками правой таблицы, которые имеют совпадающие значения всех одноименных столбцов.

Запрос 3. Вывести названия населенных пунктов, номера и название отделов, которые в них расположены.

```
SELECT location_id, city, department_id, department_name
FROM Locations NATURAL JOIN Departments;
```

Другой способ определения внутреннего соединения реализует оператор **INNER JOIN**, который имеет следующий синтаксис:

{таблица 1}[INNER] JOIN {таблица 2}
{условие соединения}

Этот оператор является более общим способом реализации внутреннего соединения и требует указаний условий соединения. Служебное слово INNER

можно не указывать. Для определения условий соединения можно использовать следующие конструкции:

- **USING (имя столбца);**
- **ON ({таблица 1.имя столбца}{оператор соединения}{таблица 2.имя столбца}**

При использовании USING, таблицы должны иметь одноименный столбец, по которому будет осуществляться соединение. Конструкция USING позволяет осуществлять соединение по нескольким столбцам, в этом случае, в качестве параметра задается список столбцов. Строки левой таблицы соединяются с теми строками правой таблицы, которые имеют совпадающие значения всех столбцов из этого списка. Рассмотрим пример использования конструкции USING.

Запрос 4. Вывести названия населенных пунктов, номера и названия отделов, которые в них расположены.

```
SELECT location_id, city, department_id, department_name
FROM   Locations   JOIN  Departments USING (location_id);
```

Конструкция **ON** предоставляет намного больше возможностей. Она позволяет:

- осуществлять соединение по столбцам, имеющим разные имена в левой и правой таблице;
- позволяет осуществлять соединение по неэквивалентности.

Запрос 5. Вывести данные о заказах, которые оформил сотрудник 165

```
SELECT employee_id, order_id, customer_id, order_date
FROM Employees JOIN Orders ON (employee_id=salesman_id)
WHERE employee_id =165;
```

В одном запросе можно использовать разные способы соединения таблиц.

Запрос 6. Для сотрудников из отдела 80 определить общую сумму продаж.

```
SELECT employee_id, first_name, last_name, job_id,
SUM(quantity*unit_price) As Sales
FROM Employees JOIN Orders ON (employee_id=salesman_id)
     JOIN Order_Items  USING(order_id)
WHERE department_id =80
GROUP BY employee_id, first_name, last_name, job_id;
```

Внешнее соединение

При использовании внутреннего соединения, запрос выводит только те строки левой таблицы, которые связаны со строками правой таблицы. При решении некоторых задач необходимо выводить все строки таблиц участвующих в запросе. Для этого следует использовать внешние соединения. Существует три вида внешнего соединения:

- левое внешнее соединение;
- правое внешнее соединение:
- полное внешнее соединение.

Левое внешнее соединение

Синтаксис:

```
{таблица 1}LEFT [OUTER] JOIN {таблица 2}
{условие соединения}
```

Запрос будет выводить все строки левой таблицы и те строки правой таблицы, которые связаны со строками правой таблицы. Если строка левой таблицы не связана со строками правой таблицы, то столбцы правой таблицы, для этой строки, будут иметь значение NULL.

Запрос 7. Вывести названия населенных пунктов, находящихся в стране country_id = 'UK', и названия отделов, которые в них расположены.

```
SELECT l.location_id, city, department_name
FROM  Locations l LEFT OUTER JOIN  Departments d
      ON (l.location_id = d.location_id)
WHERE country_id = 'UK'
```

Правое внешнее соединение

Синтаксис:

```
{таблица 1} RIGHT [OUTER] JOIN {таблица 2}
{условие соединения}
```

Запрос будет обрабатывать все строки правой таблицы и те строки левой таблицы, которые связаны со строками правой таблицы. Если строка правой таблицы не связана со строками левой таблицы, то столбцы левой таблицы, для этой строки, будут иметь значение NULL.

Запрос 8. Вывести названия населенных пунктах ,имеющих почтовые индексы: '00989', '3095','M5V 2L7','80925', и названия отделов, расположенных в этих городах, если они есть.,

```
SELECT city, department_name
FROM   Departments  RIGHT JOIN Locations USING (location_id)
WHERE postal_code IN ('00989', '3095','M5V 2L7','80925')
ORDER BY city, department_name DESC;

city   |department_name |
-------+----------------+
Bern   |                |
Munich |Public Relations|
Roma   |                |
Toronto|Marketing       |
```

Полное внешнее соединение

Синтаксис:

```
{таблица 1} FULL [OUTER] JOIN {таблица 2}
{условие соединения}
```

Запрос будет анализировать все строки, как правой таблицы, так и левой таблицы.

Запрос 9. Необходимо вывести данные о заказах, которые были оформлены в период с 10.05.17 по 31.05.17. Данные должны содержать информацию о сотруднике, который оформил заказ, и о содержимом заказа.

```
SELECT salesman_id,o.order_id,order_date,item_id,product_id,
quantity
FROM Orders o FULL join order_items oi on o.order_id=oi.order_id
WHERE order_date BETWEEN '10.05.17' AND '31.05.17';
```

```
salesman_id|order_id|order_date|item_id|product_id|quantity|
-----------+--------+----------+-------+----------+--------+
           |      19|27-05-2017|      1|        38|      53|
           |      20|27-05-2017|      1|        26|     105|
       153|      34|12-05-2017|      1|        15|     141|
       151|      21|27-05-2017|       |          |        |
       159|      41|12-05-2017|       |          |        |
       155|      44|21-05-2017|       |          |        |
       145|       3|26-05-2017|       |          |        |
```

Декартово произведение таблиц

Синтаксис:

```
{таблица 1} CROSS JOIN {таблица 2}
```

При выполнении этой операции каждая строка левой таблицы соединяется с каждой строкой правой таблицы. Напомним, что при отсутствии условий соединения автоматически осуществляется декартово произведение таблиц, и

как правило, это является ошибкой. CROSS JOIN следует применять в тех случаях, когда вы сознательно используете эту операцию.

Запрос 10. Для каждого сотрудника определить товары, которые он не продавал.

```
SELECT employee_id, product_id
FROM   Employees e CROSS JOIN Products p
WHERE p.product_id NOT IN
( SELECT DISTINCT product_id
  FROM ORDERS JOIN ORDER_ITEMS USING (ORDER_ID)
  WHERE employee_id = e.employee_id)
ORDER BY employee_id;
```

В этом запросе, сначала, с помощью декартова произведения генерируются все возможные пары значений employee_id, product_id, а потом исключаются строки содержащие значения product_id, товаров, которые продавал сотрудник.

Самосоединение таблицы

Самосоединением называется операция, при которой строка таблицы соединяется с другими строками этой же таблицы. Синтаксис и правила соединения остаются такими же, как при соединении нескольких таблиц.

Запрос 11. Вывести данные об однофамильцах.

```
SELECT emp1.employee_id, emp1.first_name, emp1.last_name,
emp1.job_id
FROM Employees emp1 JOIN Employees emp2 ON
   (emp1.last_name=emp2.last_name
    AND emp1.employee_id<>emp2.employee_id)
ORDER BY emp1.last_name;
```

В этом примере следует обратить внимание на то, что необходимо указывать полные имена столбцов, так как формально мы используем две разные таблицы.

Задание

Задача 1. Вывести название отдела, которым руководит менеджер 108 и название города, в котором расположен отдел.

Задача 2.. Вывести названия отделов и названия товаров, которые продавали сотрудники этих отделов.

Вариант а: вывести только те отделы, сотрудники которых продавали товары.
Вариант б: вывести все отделы.

Задача 3. Вывести даты продаж, осуществленных в ноябре 2019 года, и общую сумму продаж за каждую дату.

Задача 4. Выведите количество сотрудников и суммарную зарплату сотрудников, работающих в каждом городе. Должны быть выведены данные обо всех городах из таблицы Locations.

Задача 5. Вывести данные о сотрудниках, у которых сумма продаж более чем в 50 раз больше зарплаты, которую они получают.

Задача 6.. Для каждого отдела определите отношение суммы всех продаж выполненных сотрудниками этого отдела суммарной заработной плате этого отдела.

Задача 7.. Выведите данные зарплате сотрудников, с итоговыми строками, которые содержат суммарную зарплату по каждой должности, отделу и городу. Исключить данные о сотрудниках, которые работают в США (country_id ='US').