

Управление данными

Лекция 5

Работа с выражениями. Запросы к нескольким таблицам

Свечников С.В.

Содержание занятия

- 01. Использование выражений
- 02. Упорядочение результатов запросов
- 03. Соединение таблиц

01. Использование выражений

Выражения в SELECT

SQL позволяет размещать среди выбираемых столбцов скалярные выражения и константы.

Пример

```
SELECT lname, city, year*12  
FROM person;
```

Выходные столбцы можно именовать с помощью предложения AS. AS не является обязательным параметром.

Пример

```
SELECT lname, city, year*12 AS 'Возраст в месяцах'  
FROM person;
```

Комментарии в результатах запроса

Разрешается вставлять в результат запроса комментарии. Комментарий будет повторяться для каждой строки, а не 1 раз для всей таблицы.

Пример

```
SELECT lname, 'родился в ', city,  
' , его возраст составляет ', year*12, ' месяцев'  
FROM person;
```

Условные выражения

Условное выражение **CASE** может принимать любое из нескольких значений в зависимости от того, какой из предикатов выполнен или с каким из значений обнаружено совпадение.

Выражение CASE имеет следующие формы:

- ☐ CASE
- ☐ NULLIF
- ☐ COALESCE

Пример

```
SELECT  
    CASE  
        WHEN Iname='Петров'  
        THEN 'Ветров'  
    END  
FROM person;
```

CASE

Если нам необходимо оставить строки с другими значениями в неизменном виде, используем предложение **ELSE**.

Пример

```
SELECT  
    CASE  
        WHEN Iname='Петров'  
        THEN 'Ветров'  
        ELSE Iname  
    END  
FROM person;
```

NULLIF

Форма **NULLIF** принимает в качестве аргументов 2 значения и если они совпадают – результатом будет NULL, в противном случае – первое из двух значений.

Пример

```
SELECT NULLIF(lname, 'Жуков'), fname  
FROM person;
```

COALESCE

Форма **COALESCE** последовательно просматривает список значений и выводит первое из них, не равное NULL.

Примеры

```
SELECT COALESCE(phone, pid)  
FROM person;
```

```
SELECT COALESCE(phone, 0)  
FROM person;
```

02. Упорядочение результатов запросов

Сортировка

Оператор **ORDER BY** позволяет изменять порядок вывода строк.

Пример

```
SELECT *  
FROM person  
ORDER BY Iname;
```

Существует 2 порядка сортировки:

- ☐ ASC – по возрастанию
- ☐ DESC – по убыванию

Пример

```
SELECT *  
FROM person  
ORDER BY Iname DESC;
```

Сортировка по множеству столбцов

Задать упорядочение можно сразу по нескольким столбцам и с разным порядком сортировки.

Пример

```
SELECT pid, city, lname  
FROM person  
ORDER BY city ASC, lname DESC;
```

В ORDER BY можно указать любое количество столбцов.

Кроме того, вместо имен столбцов в ORDER BY можно указать его порядковый номер.

Пример

```
SELECT pid, city, lname  
FROM person  
ORDER BY 3;
```

Сортировка групп

ORDER BY можно использовать совместно с GROUP BY для упорядочивания групп.

Пример

```
SELECT city, COUNT(city)
FROM person
GROUP BY city
ORDER BY 2;
```

Сортировка NULL-значений

Если в столбце имеются NULL-значения, то они либо будут предшествовать всем остальным значениям, либо располагаться после них.

Пример

```
SELECT lname, phone  
FROM person  
ORDER BY phone;
```

03. Соединение таблиц

Соединения

Соединение – способность определять отношения между таблицами и выводить информацию в соответствии с установленными отношениями.

Допускается установление связей между данными, которые не были связаны.

С помощью соединений можно устанавливать отношения между любым количеством таблиц и соединять разрозненные данные.

Именованние таблиц и столбцов

Имя столбца состоит из имени таблицы, разделительной точки и имени столбца.

Примеры

person.lname

person.pid

person.year

Указывать имя таблицы не обязательно, но при обращении к нескольким таблицам могут возникнуть проблемы, если столбцы этих таблиц имеют одинаковые имена.

Естественное соединение

Самый распространенный тип соединения – это **естественное соединение (natural join)**. Они основываются на уже существующих в БД соединениях.

Примеры

```
SELECT pid, lname, fname, name  
FROM person, gruppa  
WHERE group_id=gid;
```

```
SELECT person.pid, person.lname, person.fname, gruppa.name  
FROM person, gruppa  
WHERE person.group_id= gruppa.gid;
```

Если не указать соединения, то получим так называемое **Декартово произведение** – произведение всех столбцов одной таблицы на столбцы другой таблицы.

Составные предикаты

Составные предикаты состоят из предиката соединения и обычного предиката.

Для повышения эффективности обычный предикат указывается первым.

Пример

```
SELECT pid, lname, fname, city, name  
FROM person, группа  
WHERE city IN ('Москва', 'Самара')  
AND group_id=gid;
```



Обычный предикат



Предикат соединения

Эквисоединения

Соединения, в которых используются предикаты с равенствами, называются **эквисоединениями**.

Примеры

```
SELECT pid, lname, fname, name  
FROM person, grupp  
WHERE group_id=gid;
```

```
SELECT pid, lname, fname, name  
FROM person, grupp  
WHERE gid>pid;
```

Соединение более 2 таблиц

Можно конструировать запросы, соединяющие более 2 таблиц.

Пример

```
SELECT pid, lname, fname, name, lector  
FROM person, gruppа, course  
WHERE person.group_id=gruppа.gid  
AND course.group_id=gruppа.gid;
```

Соединение таблицы с собой

Можно соединять таблицу со своей копией.

При соединении таблицы с собой имена всех столбцов повторяются вместе с префиксом таблицы. Для того, чтобы обойти эту проблему таблице можно присвоить временное имя – **алиас**.

Пример

```
SELECT a.pid, a.lname, b.lname, b.mf  
FROM person a, person b  
WHERE a.mf=b.mf;
```

В результатах запроса каждая комбинация показана дважды. Для исключения дублирования можно сравнить записи по ID.

Пример

```
SELECT a.pid, a.lname, b.lname, b.mf  
FROM person a, person b  
WHERE a.mf=b.mf  
AND a.pid<b.pid;
```

Алиасы

Алиасы можно использовать не только, когда объединяется таблица и ее копия.

Пример

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p, группа g  
WHERE p.group_id= g.gid;
```

Комбинирование алиасов и имен таблиц

Можно конструировать соединения, содержащие как различные таблицы, так и алиасы одной таблицы.

Пример

Поиск пар групп, занимающихся у одного преподавателя.

```
SELECT a.gid, a.name, b.name, lector  
FROM gruppa a, gruppa b, course  
WHERE a.gid<>b.gid  
AND group_id=a.gid  
AND a.name>b.name;
```

Внешние соединения

Предположим, что в таблице person у одного из студентов не задан group_id. При выполнении запроса:

```
SELECT pid, lname, fname, name  
FROM person, grupp  
WHERE group_id=gid;
```

этот студент не будет показан.

Представленное соединение, исключающее несовпадающие строки, называется внутренним соединением.

Для того, чтобы в результатах запроса были представлены данные с NULL-значениями необходимо использовать внешние соединения.

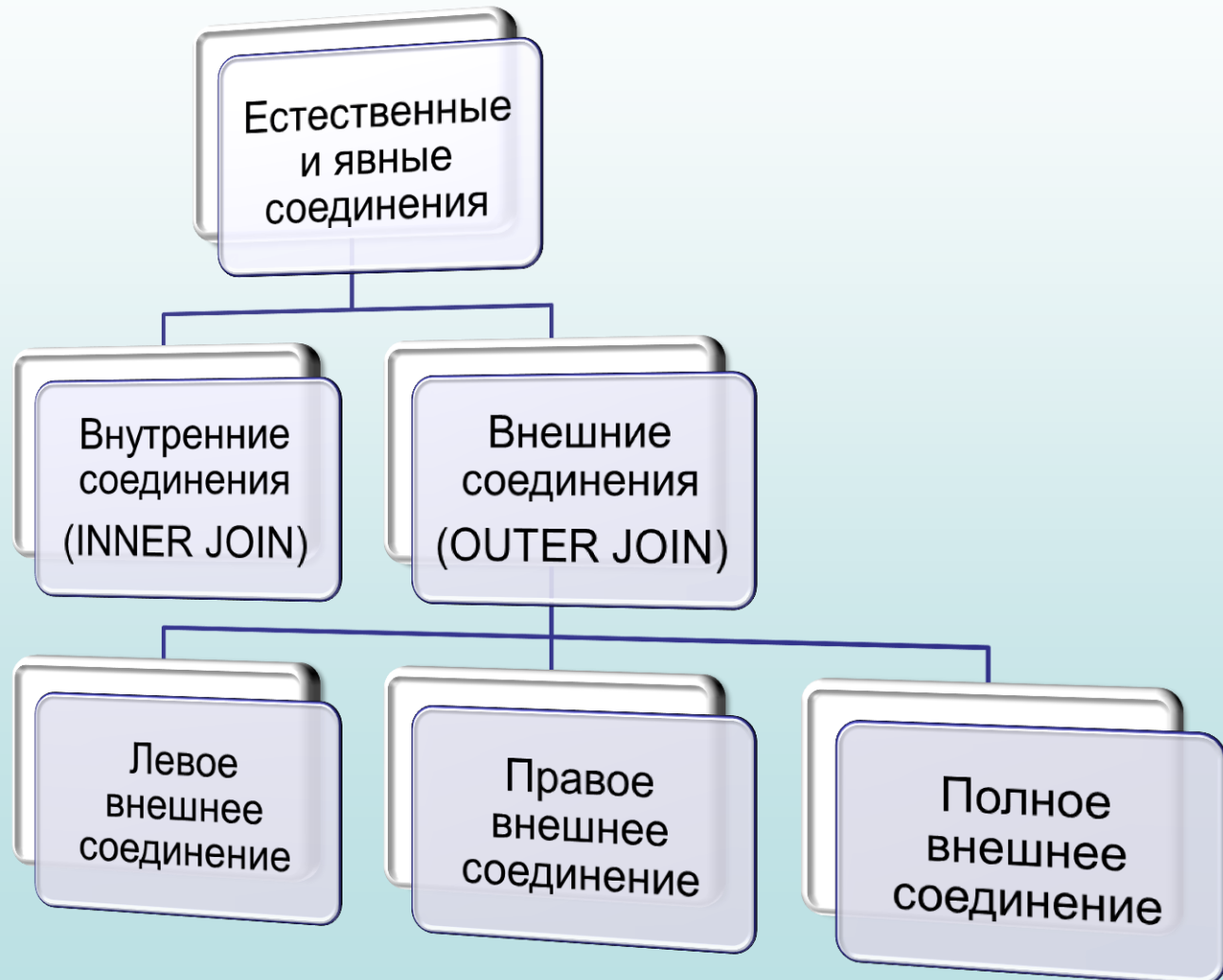
Специальные операторы соединения

В SQL определены специальные операторы, облегчающие выполнение некоторых распространенных операций соединения.



Естественные и явные соединения

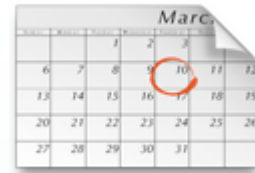
Естественные и явные соединения в свою очередь могут быть разделены на следующие категории.



Вопросы



Домашнее задание



- ☐ Изучить главы 9 и 10 книги М.Грабера Введение в SQL
- ☐ Подготовиться к контрольной работе по лекции

Контроль



Для выполнения контрольного теста используем ссылку:

app.startexam.com/Center/Web/student

Короткая ссылка:

clck.ru/2Undm

Пароль на тест:

1240