

ЛАБОРАТОРНАЯ РАБОТА №7

Тема: Раздел FROM предложения SELECT. Виды соединений. Внешние соединения. Источники данных на основе внутренних подзапросов.

Цель: Пробрести навыки использования различных видов соединений при создании SQL запросов.

В данной лабораторной работе используются база данных HR, созданная в рамках первой лабораторной работы. Также используется таблица TB_ELEKTROSTAL_2018, которая создана и наполнена данными в процессе выполнения лаб. работы №1.

Дополнительно используется удаленная база данных проекта stackoverflow.com. Структура этой базы данных описана в методических указаниях к выполнению лабораторной работы № 3. Напомним, адрес ресурса <http://data.stackexchange.com/stackoverflow/query/new>.

Рассмотрим две таблицы (отношения), описанные в методических указаниях к выполнению лабораторной работы 6.

Таблица 1.

Отношение «tb_city» (Города)

| shifrStr | name |
|----------|-----------|
| 1 | Луганск |
| 2 | Юбилейный |
| 3 | Видный |

Таблица 2.

Отношение «tb_street» (Улицы)

| shifrStr | shifrCit | tame |
|----------|----------|------------|
| 1 | 1 | Оборонная |
| 2 | 1 | Ватутина |
| 3 | 2 | Шахтерский |
| 4 | - | Артема |

Эти две таблицы будут использоваться для иллюстрации различных видов соединений (разновидностей оператора JOIN).

JOIN — оператор языка SQL, который является реализацией операции соединения реляционной алгебры. Входит в предложение FROM операторов SELECT, UPDATE и DELETE.

Операция соединения, как и другие бинарные операции, предназначена для обеспечения выборки данных из двух таблиц и включения этих данных в один результирующий набор. Отличительными особенностями операции соединения являются следующие:

- в схему таблицы-результата входят столбцы обеих исходных таблиц (таблиц-операндов), то есть схема результата является «сцеплением» схем операндов;
- каждая строка таблицы-результата является «сцеплением» строки из одной таблицы-операнда со строкой второй таблицы-операнда.

Определение того, какие именно исходные строки войдут в результат и в каких сочетаниях, зависит от типа операции соединения и от явно заданного условия соединения. Условие соединения, то есть условие сопоставления строк исходных таблиц друг с другом, представляет собой логическое выражение (предикат).

При необходимости соединения не двух, а нескольких таблиц, операция соединения применяется несколько раз (последовательно).

Описание оператора

```
FROM
  Table1
  {INNER | {LEFT | RIGHT | FULL} OUTER | CROSS } JOIN
  Table2
  {ON <condition> | USING (field_name [, ... n])}
```

В большинстве СУБД при указании слов LEFT, RIGHT, FULL слово OUTER можно опустить. Слово INNER также в большинстве СУБД можно опустить.

В общем случае СУБД при выполнении соединения проверяет условие (предикат) condition. Если названия столбцов по которым происходит

соединение таблиц совпадают, то вместо ON можно использовать USING. Для CROSS JOIN условие не указывается.

Для перекрёстного соединения (декартова произведения) CROSS JOIN в некоторых реализациях SQL используется оператор «запятая» (,):

```
FROM  
    Table1,
```

Виды оператора JOIN

INNER JOIN

Оператор внутреннего соединения INNER JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является симметричным.

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

Тело результата логически формируется следующим образом. Каждая строка одной таблицы сопоставляется с каждой строкой второй таблицы, после чего для полученной «соединённой» строки проверяется условие соединения (вычисляется предикат соединения). Если условие истинно, в таблицу-результат добавляется соответствующая «соединённая» строка.

Описанный алгоритм действий является строго логическим, то есть он лишь объясняет результат, который должен получиться при выполнении операции, но не предписывает, чтобы конкретная СУБД выполняла соединение именно указанным образом. Существует множество способов реализации операции соединения, например, соединение вложенными циклами (англ. inner loops join), соединение хэшированием (англ. hash join), соединение слиянием (англ. merge join). Единственное требование состоит в том, чтобы любая реализация логически давала такой же результат, как при применении описанного алгоритма.

Для иллюстрации принципов работы операции соединения в SQL часто используют диаграммы Венна. На этой диаграмме множество строк таблицы обозначают кругом. Множество строк, представляющих результат операции

соединения, выделяют цветом. Применительно к операции внутреннего соединения (inner join) на примере таблиц tb_street (левая таблица) и tb_city (правая таблица) диаграмма Венна показана на рис. 1.

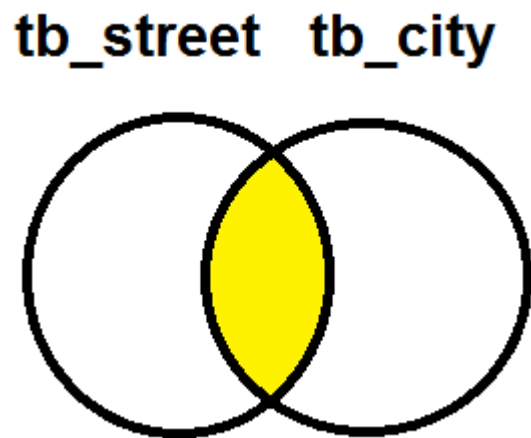


Рисунок 1. Диаграмма Венна для иллюстрации операции внутреннего (inner) соединения на примере таблиц tb_street и tb_city

```
SELECT * FROM tb_street  
INNER JOIN tb_city ON tb_street.shifrcit =  
tb_city.shifrcit
```

Выполнив этот запрос получим следующий результат.

Таблица 3.

| tb_city.shifrCit | tb_city.name | tb_street.shifrStr | tb_street.shifrcCit | tb_streer.Name |
|------------------|--------------|--------------------|---------------------|----------------|
| 1 | Луганск | 1 | 1 | Оборонная |
| 1 | Луганск | 2 | 1 | Ватутина |
| 2 | Юбилейный | 3 | 2 | Шахтерский |

OUTER JOIN

Соединение двух таблиц, в результат которого обязательно входят все строки либо одной, либо обеих таблиц.

LEFT OUTER JOIN

Оператор левого внешнего соединения LEFT OUTER JOIN соединяет две таблицы. Порядок таблиц для оператора важен, поскольку оператор не является симметричным.

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

Тело результата логически формируется следующим образом. Пусть выполняется соединение левой и правой таблиц по предикату (условию) p .

1. В результат включается внутреннее соединение (INNER JOIN) левой и правой таблиц по предикату p .
2. Затем в результат добавляются те записи левой таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие правой таблице, заполняются значениями NULL.

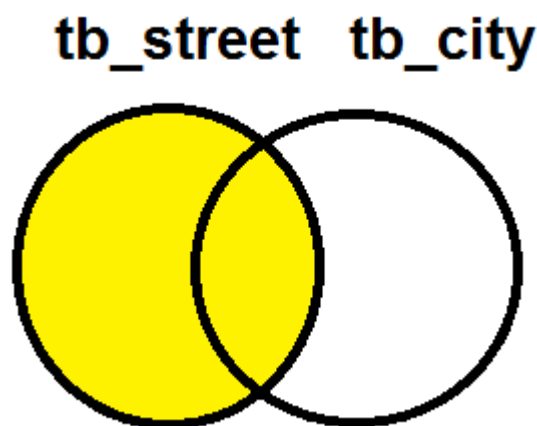


Рисунок 2. Диаграмма Венна для иллюстрации операции левого внешнего соединения (left outer join) на примере таблиц `tb_street` и `tb_city`

```
SELECT * FROM tb_street      -- Левая таблица
LEFT OUTER JOIN tb_city     -- Правая таблица
ON tb_street.shifrcit = tb_city.shifrcit
```

Результата выполнения запроса показан в табл.4

Таблица 4.

| tb_street.shifrStr | tb_street.shifrcit | tb_street.Name | tb_city.shifrcit | tb_city.name |
|--------------------|--------------------|----------------|------------------|--------------|
| 1 | 1 | Оборонная | 1 | Луганск |
| 2 | 1 | Ватутина | 1 | Луганск |

| | | | | |
|---|------|------------|------|-----------|
| 3 | 2 | Шахтерский | 2 | Юбилейный |
| 4 | null | Артема | null | null |

RIGHT OUTER JOIN

Оператор правого внешнего соединения **RIGHT OUTER JOIN** соединяет две таблицы. Порядок таблиц для оператора важен, поскольку оператор не является симметричным.

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

Тело результата логически формируется следующим образом. Пусть выполняется соединение левой и правой таблиц по предикату (условию) *p*.

1. В результат включается внутреннее соединение (**INNER JOIN**) левой и правой таблиц по предикату *p*.
2. Затем в результат добавляются те записи правой таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие левой таблице, заполняются значениями **NULL**.

tb_street tb_city

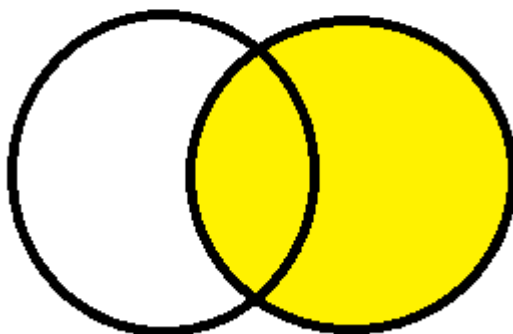


Рисунок 3. Диаграмма Венна для иллюстрации правого внешнего соединения (right outer join) на примере таблиц *tb_street* и *tb_city*

```
SELECT * FROM tb_street      -- Левая таблица
RIGHT OUTER JOIN tb_city    -- Правая таблица
ON tb_street.shifrcit = tb_city.shifrcit
```

Таблица 5.

| tb_street.shifrStr | tb_street.shifrcCit | tb_street.Name | tb_city.shifrCit | tb_city.name |
|--------------------|---------------------|----------------|------------------|--------------|
| 1 | 1 | Оборонная | 1 | Луганск |
| 2 | 1 | Ватутина | 1 | Луганск |
| 3 | 2 | Шахтерский | 2 | Юбилейный |
| null | null | null | 3 | Видный |

FULL OUTER JOIN

Оператор полного внешнего соединения FULL OUTER JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является симметричным.

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

Тело результата логически формируется следующим образом. Пусть выполняется соединение первой и второй таблиц по предикату (условию) *p*. Слова «первой» и «второй» здесь не обозначают порядок в записи (который неважен), а используются лишь для различения таблиц.

1. В результат включается внутреннее соединение (INNER JOIN) первой и второй таблиц по предикату *p*.
2. В результат добавляются те записи первой таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие второй таблице, заполняются значениями NULL.
3. В результат добавляются те записи второй таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие первой таблице, заполняются значениями NULL.

tb_street **tb_city**

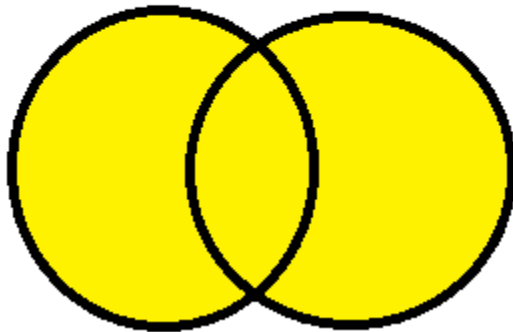


Рисунок 4. Диаграмма Венна для иллюстрации двустороннего внешнего соединения (full outer join) на примере таблиц tb_street и tb_city

```
SELECT * FROM tb_street  
FULL OUTER JOIN  
tb_city ON tb_street.shifrcit = tb_city.shifrcit
```

Таблица 6. Результат:

| tb_street.shifrStr | tb_street.shifrcit | tb_street.Name | tb_city.shifrcit | tb_city.name |
|--------------------|--------------------|----------------|------------------|--------------|
| 1 | 1 | Оборонная | 1 | Луганск |
| 2 | 1 | Ватутина | 1 | Луганск |
| 3 | 2 | Шахтерский | 2 | Юбилейный |
| 4 | null | Артема | null | null |
| null | null | null | 3 | Видный |

LEFT OUTER JOIN с ограничениями. (Левое подмножество)

На практике часто приходится отыскивать множество записей одной таблицы, не входящих в другую. Пусть, например, требуется отыскать множество улиц, для которых не указан город. Для решения такой задачи нужно дополнительно с операцией внешнего соединения использовать условие фильтрации: требуется проверить, что значение поля в другой таблиц содержит null.

tb_street tb_city

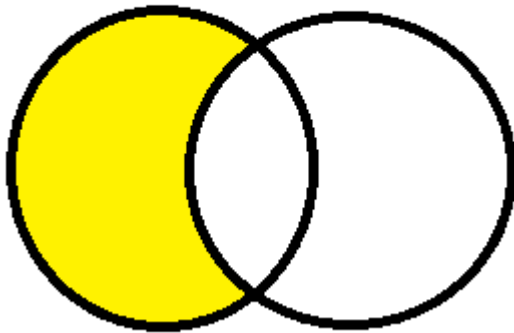


Рисунок 5. Диаграмма Венна для иллюстрации левого внешнего соединения (left outer join) с ограничениями на примере таблиц tb_street и tb_city

```
SELECT * FROM tb_street      -- Левая таблица
LEFT OUTER JOIN tb_city     -- Правая таблица
ON tb_street.shifrcit = tb_city.shifrcit
WHERE tb_city.shifrcit is null
```

Таблица 7.

| tb_street.shifrStr | tb_street.shifrcCit | tb_street.Name | tb_city.shifrcCit | tb_city.name |
|--------------------|---------------------|----------------|-------------------|--------------|
| 4 | null | Артема | null | null |

RIGHT OUTER JOIN с ограничениями. (Правое подмножество)

Для демонстрации этого вида соединений рассмотрим запрос, который выводит список городов, в которых не расположено ни одной улицы.

tb_street tb_city

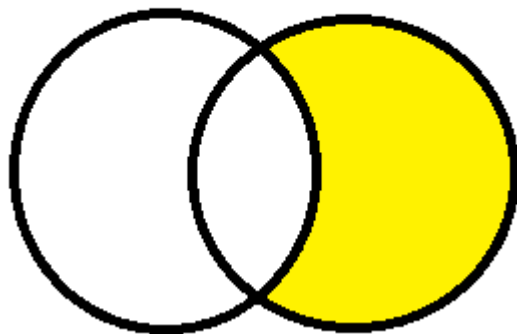


Рисунок 6. Диаграмма Венна для иллюстрации правого внешнего соединения (right outer join) с ограничениями на примере таблиц tb_street и tb_city

```
SELECT * FROM tb_street      -- Левая таблица
RIGHT OUTER JOIN tb_city    -- Правая таблица
ON tb_street.shifrcit = tb_city.shifrcit
WHERE tb_street.shifrcit is null
```

Таблица 8.

| tb_street.shifrStr | tb_street.shifrcCit | tb_street.Name | tb_city.shifrCit | tb_city.name |
|--------------------|---------------------|----------------|------------------|--------------|
| null | null | null | 3 | Видный |

FULL OUTER JOIN с ограничениями. (Все кроме пересечения)

Иногда требуется найти записи как левой таблицы так и правой таблицы, не имеющие пары в другой таблице. Для этого можно использовать вид соединения full outer join с ограничениями. Пусть требуется найти как улицы, не относящиеся ни к одному городу, так и города, не содержащие улиц. Запрос, представленный ниже, решает поставленную задачу.

tb_street tb_city

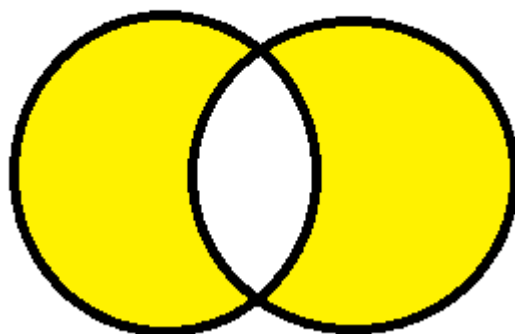


Рисунок 7. Диаграмма Венна для иллюстрации полного внешнего соединения (full outer join) с ограничениями на примере таблиц tb_street и tb_city

```
SELECT * FROM tb_street      -- Левая таблица
```

```
RIGHT OUTER JOIN tb_city    -- Правая таблица
ON tb_street.shifrcit = tb_city.shifrcit
WHERE tb_street.shifrcit is null
AND tb_city.shifrcit is null
```

Таблица 9.

| tb_street.shifrStr | tb_street.shifrcCit | tb_street.Name | tb_city.shifrCit | tb_city.name |
|--------------------|---------------------|----------------|------------------|--------------|
| null | null | null | 3 | Видный |
| 4 | null | Артема | null | null |

Порядок выполнения работы

Порядок выполнения работы.

Составить следующие запросы

- Составить запрос к БД stackoverflow. Результат запроса должен содержать две колонки. Первая колонка - имя пользователя, создавшего этот пост. Вторая колонка - содержание поста. Таблица Posts должна быть расположена слева от операции join. Указание: необходимо использовать left outer join операцию соединения таблиц.
- Составить запрос как в первом случае только таблица Posts должна быть расположена справа от операции join.
- Составить запрос к БД stackoverflow. Результат запроса должен содержать список 10 первых пользователей, которые не создали ни одного поста. Схему (список колонок) запроса определите самостоятельно. Сортировка строк также не требуется. Пусть это будут 10 случайно отобранных пользователей, не создавших ни одного поста. Для составления запроса обязательно требуется использовать операцию join.
- Составить запрос к БД stackoverflow. Результат запроса должен содержать список 10 первых постов (таблица Posts), к которым нет ни одного комментария (таблица Comments). Схему (список колонок) запроса определите самостоятельно. Сортировка строк также не требуется. Пусть

это будут 10 случайно отобранных пользователей, не создавших ни одного поста.

- Для БД HR составить запрос, включающий все поля таблицы JOBS и все поля таблицы JOB_HISTORY. Результирующая таблица является результатом соединения двух упомянутых таблиц. Пусть таблица JOBS располагается слева от оператора JOIN, а таблица JOB_HISTORY – справа от этого оператора. Требуется составить три варианта запроса с использованием операторов LEFT OUTER JOIN, RIGHT OUTER JOIN и FULL OUTER JOIN.
- Составить запрос для БД HR, который состоит из двух колонок. Первая колонка – название страны (таблица COUNTRIES). Вторая колонка название должности (из таблицы JOBS). Каждая строка содержит название страны и название должности. Выводятся те должности, которые не встречаются у работников в этой стране. Если в стране вообще нет работников, то название этой страны также требуется вывести, при этом в названии должности должно быть значение NULL.

Содержание отчета:

1. Тема, цель лабораторной работы.
2. Примеры выполнения запросов к базе данных.
3. Составленные согласно заданию запросы и скриншоты полученных результатов.
5. Выводы.

Контрольные вопросы:

1. Опишите разновидности операции JOIN. Внешние и внутренние соединения.
2. Объясните диаграммы Венна для различных видов соединений. Сколько разновидностей диаграмм Вы знаете? Опишите диаграммы Венна каждого из указанных Вами типов.

3. Назовите общие черты и отличия левого и правого внешнего соединения таблиц.

4. Дайте характеристику операции FULL OUTER JOIN.

5. Нужно ли в операциях внешнего соединения использовать ключевое слово ON?

6. Для каких разновидностей операции внешнего соединения требуется использовать раздел WHERE оператора SQL?

7. Укажите способы уточнения принадлежности полей таблицам для случая, когда имена полей в обеих соединяемых таблицах совпадают.

8. Составьте запрос, который определяет для работников каких профессий (JOBS) не предусматривается процент вознаграждения (COMMISSION_PCT).

9. Составьте три варианта запроса соединяющего две таблицы (LOCATIONS и COUNTRIES). Требуется левую, правую разновидность внешних соединений, а также левое и правое внешнее соединение одновременно. Объясните полученные результаты.

10. Охарактеризуйте операцию ... JOIN ... ON ...

Файл: Лабораторная работа 7
Каталог: C:\Windows\system32
Шаблон: C:\Users\sss\AppData\Roaming\Microsoft\Шаблоны\Normal.dotm
Заголовок:
Содержание:
Автор: sss
Ключевые слова:
Заметки:
Дата создания: 12.11.2018 22:31:00
Число сохранений: 72
Дата сохранения: 14.11.2018 22:43:00
Сохранил: sss
Полное время правки: 293 мин.
Дата печати: 14.11.2018 22:46:00
При последней печати
 страниц: 13
 слов: 2 407 (прибл.)
 знаков: 13 724 (прибл.)