

Базы данных

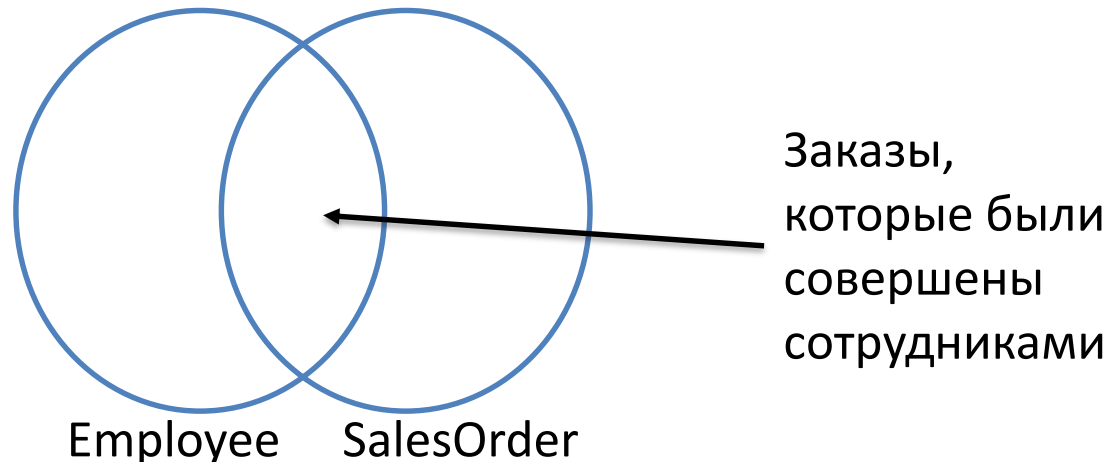
03 | Запросы к нескольким таблицам с соединениями

План

- Основные концепции соединений
- Синтаксис соединений
- Внутренние соединения
- Внешние соединения
- Перекрестные соединения
- Самосоединения

Соединения (JOIN). Основные концепции

- Соединяет строки из нескольких таблиц с помощью указываемого критерия соответствия:
 - Обычно базируется на парах первичный ключ – внешний ключ
 - Например, вернуть строки, которые соединяют данные из таблиц **Employee** и **SalesOrder** на основе равенства значений первичного ключа **Employee.EmployeeID** и внешнего **SalesOrder.EmployeeID**
- Можно представлять таблицы в виде множеств на диаграммах Венна



Синтаксис соединений

- **ANSI SQL-92**

- Таблицы соединяются при помощи оператора JOIN в предложении FROM

- Предпочитаемый вариант

```
SELECT ...  
FROM   Table1 JOIN Table2  
       ON <on_predicate>;
```

- **ANSI SQL-89**

- Таблицы соединяются при помощи перечисления через запятую в предложении FROM

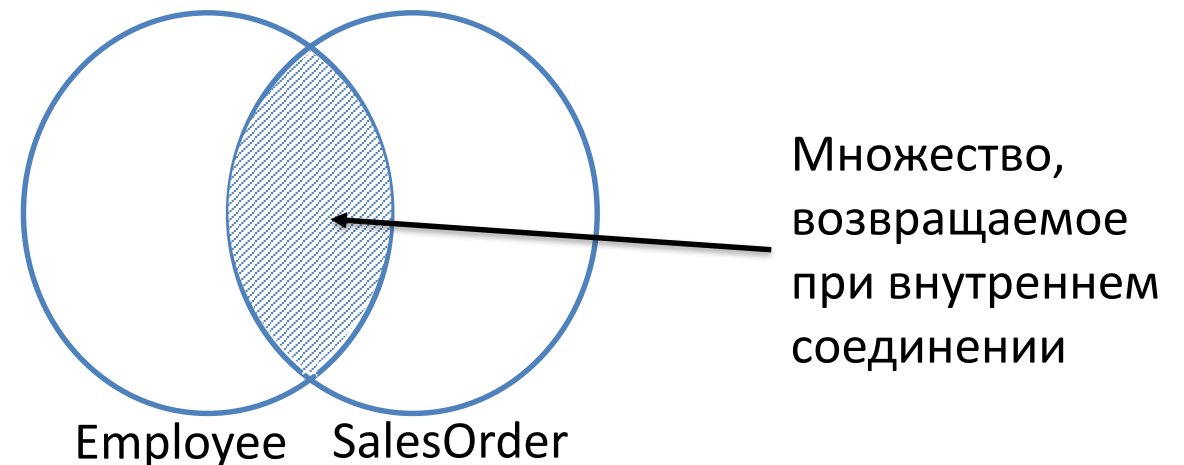
- Не рекомендуется: при ошибке возможно декартово произведение!

```
SELECT ...  
FROM   Table1, Table2  
WHERE  <where_predicate>;
```

Внутренние соединения (Inner Joins)

- Возвращает только те строки, для которых найдено соответствие в обоих входных таблицах
- Соответствие между строками таблиц через указанные в предикате атрибуты
- Если используется операция =, то получается эквисоединение (equi-join)

```
SELECT emp.FirstName, ord.Amount  
FROM HR.Employee AS emp  
[INNER] JOIN Sales.SalesOrder AS ord  
ON emp.EmployeeID = ord.EmployeeID
```



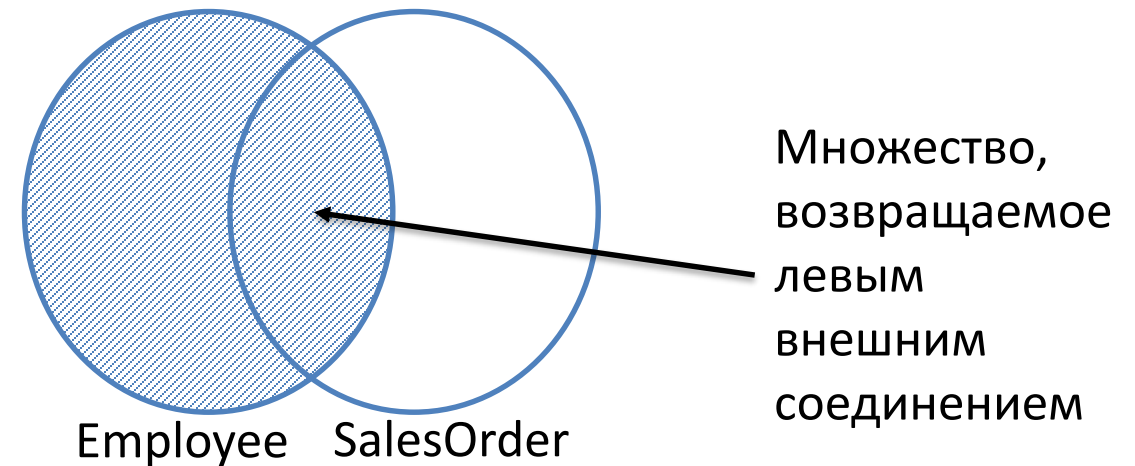
DEMO

Использование внутренних соединений

Внешние соединения(Outer Joins)

- Возвращает все строки из одной таблицы и найденные соответствия из другой
- Все строки одной из таблиц сохраняются
 - Обозначается ключ. словами: LEFT, RIGHT, FULL
 - Все строки из “сохраненной” таблицы попадают в результат
- Извлекаются соответствия из другой таблицы
- Дополнительные строки добавляются к результату, если соответствие не найдено
 - Используются значения NULL для столбцов, где не найдено соответствие

```
SELECT emp.FirstName, ord.Amount  
FROM HR.Employee AS emp  
LEFT [OUTER] JOIN Sales.SalesOrder AS ord  
ON emp.EmployeeID = ord.EmployeeID;
```



Пример: Найти всех сотрудников. Для тех, кто делал заказы - вернуть сумму заказа, для тех у кого нет заказов – вернуть NULL в столбце для суммы заказа.

DEMO

Использование внешних соединений

Перекрестные соединения (Cross Joins)

- Комбинирует каждую строку из первой таблицы с каждой строкой из второй табл.
- Выводятся все возможные комбинации
- Логическая основа для внутренних и внешних соединений
 - Внутреннее соединение начинает с декартова произведения, потом накладывается фильтр
 - Внешнее соединение после фильтрации результатов декартова произведения добавляет NULL для строк без соответствия
- Редко применяется

| Employee | | Product | |
|------------|-----------|-----------|--------|
| EmployeeID | FirstName | ProductID | Name |
| 1 | Dan | 1 | Widget |
| 2 | Aisha | 2 | Gizmo |

```
SELECT emp.FirstName, prd.Name  
FROM HR.Employee AS emp  
CROSS JOIN Production.Product AS prd;
```

| Result | |
|-----------|--------|
| FirstName | Name |
| Dan | Widget |
| Dan | Gizmo |
| Aisha | Widget |
| Aisha | Gizmo |

Исключение: Генерирование данных для тестирования

DEMO

Использование перекрестных соединений

Самосоединение (Self Joins)

- Сравнивает строки в той же таблице
- Создает два экземпляра той же таблицы в предложении FROM
 - Требуется, как минимум, один псевдоним
- Пример: Вернуть всех сотрудников и имена их начальников (менеджеров)

| Employee | | |
|------------|-----------|-----------|
| EmployeeID | FirstName | ManagerID |
| 1 | Dan | NULL |
| 2 | Aisha | 1 |
| 3 | Rosie | 1 |
| 4 | Naomi | 3 |

```
SELECT emp.FirstName AS Employee,  
       man.FirstName AS Manager  
FROM HR.Employee AS emp  
LEFT JOIN HR.Employee AS man  
ON emp.ManagerID = man.EmployeeID;
```

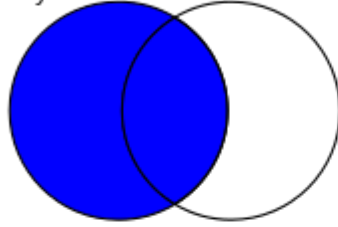
| Result | |
|----------|-------------|
| Employee | Manager |
| Dan | <i>NULL</i> |
| Aisha | Dan |
| Rosie | Dan |
| Naomi | Aisha |

DEMO

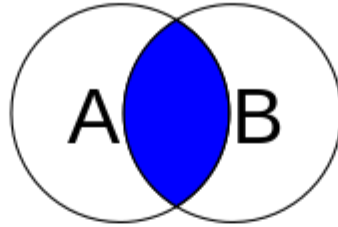
Использование самосоединений

Для запоминания

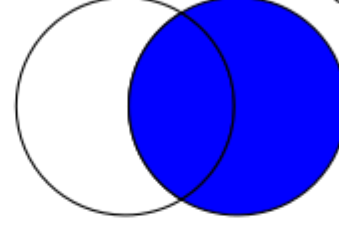
```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
```



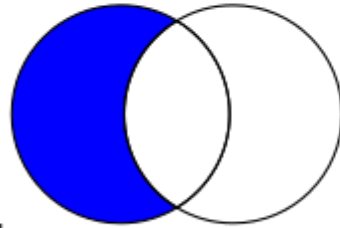
```
SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key
```



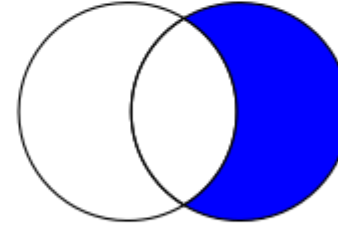
```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
```



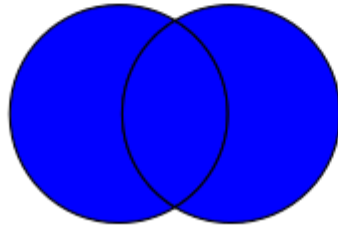
```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL
```



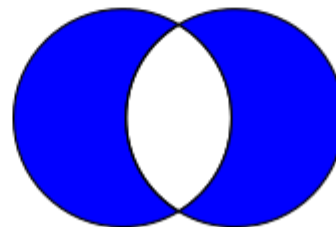
```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE a.key IS NULL
```



```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
```



```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```



SQL JOINS

Изучено

- Основные концепции соединений
- Синтаксис соединений
- Внутренние соединения
- Внешние соединения
- Перекрестные соединения
- Самосоединения