

СРСП 8. Реализация операций реляционной алгебры предложением SELECT

С помощью предложения SELECT можно реализовать любую операцию реляционной алгебры.

Объединение двух таблиц содержит те строки, которые есть либо в первой, либо во второй, либо в обеих таблицах. Но при этом исходные отношения или их объединяемые проекции должны быть совместимыми по объединению. Для SQL это означает, что две таблицы можно объединять тогда и только тогда, когда:

- a. они имеют одинаковое число столбцов, например, m;
- b. для всех i ($i = 1, 2, \dots, m$) i -й столбец первой таблицы и i -й столбец второй таблицы имеют в точности одинаковый тип данных.

Например, выдать названия продуктов, в которых нет жиров, либо входящих в состав блюда с кодом БЛ = 1:

Результат:

```
SELECT  Продукт
FROM    Продукты
WHERE   Жиры = 0
UNION
SELECT  Продукт
FROM    Состав
WHERE   БЛ = 1
```

Продукт

Майонез
Лук
Помидоры
Зелень
Яблоки
Сахар

Избыточные дубликаты всегда исключаются из результата UNION.

Предложением с UNION можно объединить любое число таблиц (проекций таблиц). Так, к предыдущему запросу можно добавить (перед точкой с запятой) конструкцию

```
UNION
SELECT  Продукт
FROM    Продукты
WHERE   Са < 250
```

Селекция (выборка, горизонтальное подмножество) таблицы создается из тех ее строк, которые удовлетворяют заданным условиям. Пример:

```
SELECT  *
FROM    Блюда
WHERE   Основа = 'Молоко' AND Выход > 200;
```

Проекция (вертикальное подмножество) таблицы создается из указанных ее столбцов (в заданном порядке) с последующим исключением избыточных дубликатов строк. Пример:

```
SELECT  DISTINCT Блюдо, Выход, Основа
FROM    Блюда;
```

Пересечение двух таблиц содержит только те строки, которые есть и в первой, и во второй. Пример:

```
SELECT  БЛ
FROM    Состав
WHERE   БЛ IN
(       SELECT БЛ
        FROM Меню);
```

Разность двух таблиц содержит только те строки, которые есть в первой, но отсутствуют во второй. Пример:

```
SELECT  БЛ
```

```
FROM      Состав
WHERE     БЛ NOT IN
(         SELECT БЛ
          FROM Меню);
```

Декартово произведение таблиц

Декартово произведение n таблиц - это таблица, содержащая все возможные строки r , такие, что r является сцеплением какой-либо строки из первой таблицы, строки из второй таблицы, ... и строки из n -й таблицы. Для получения декартова произведения нескольких таблиц надо указать во фразе FROM перечень перемножаемых таблиц, а во фразе SELECT – все их столбцы.

Так, для получения декартова произведения Товары и Поставщики надо выдать запрос

```
SELECT     Товары.*, Движение_товара.*
FROM       Товары, Движение_товара;
```

Получим таблицу, содержащую $4+6=10$ столбцов и $5 \times 12 = 60$ строк:

Эквисоединение таблиц

Если из декартова произведения убрать ненужные строки и столбцы, то можно получить актуальные таблицы, соответствующие любому из соединений.

Очевидно, что отбор актуальных строк обеспечивается вводом в запрос WHERE фразы, в которой устанавливается соответствие между:

- кодами поставщиков в таблицах Поставщики и Движения товара,
- кодами товаров в таблицах Товары и Движение_товаров,

Такой скорректированный запрос

```
SELECT     Товары.*, Поставщики.*, Движение_товаров.*
FROM       Товары, Поставщики, Движение_товара
WHERE      Товары.Код товара = Движение_товаров.Код товара
AND       Поставщики.Код поставщика = Движение_товаров.Код
```

поставщика;

позволит получить эквисоединение таблиц: Товары, Поставщики, Движение_товара.

Естественное соединение таблиц

В эквисоединение таблиц вошли дубликаты столбцов, по которым проводилось соединение. Для исключения этих дубликатов можно создать естественное соединение тех же таблиц:

```
SELECT     Дата, № накладной, Движение_товара.Код поставщика,
Поставщики.Наименование, Движение_товара.Код товара, Товары.Наименование,
Движение_товара.Количество, Движение_товара.Цена
FROM       Товары, Поставщики, Движение_товара
WHERE      Товары.Код товара = Движение_товаров.Код товара
AND       Поставщики.Код поставщика = Движение_товаров.Код
```

поставщика;

Тета-соединение таблиц

Трудно подобрать несложный пример, иллюстрирующий тета-соединение таблиц. Поэтому сконструируем такой надуманный запрос:

```
SELECT     Товары.*, Поставщики.*
FROM       Товары, Поставщики
WHERE      Категория товара > Банк;
```

позволяющий выбрать из полученного декартова произведения таблиц Товары и Поставщики лишь те строки, в которых значение банка "меньше" (по алфавиту) значения вида категории товара:

Соединение таблиц с дополнительным условием

При формировании соединения создается рабочая таблица, к которой применимы все операции: отбор нужных строк соединения (WHERE фраза), упорядочение получаемого результата (ORDER BY фраза) и агрегатирование данных (SQL-функции и GROUP BY фраза).

Например, для получения перечень накладных, поставивших товары по цене более 20 тенге, можно сформировать запрос на основе композиции:

```
SELECT Дата, № накладной, Движение_товара.Код поставщика,
Поставщики.Наименование, Движение_товара.Код товара, Товары.Наименование,
Движение_товара.Количество, Движение_товара.Цена
FROM Товары, Поставщики, Движение_товара
WHERE Товары.Код товара = Движение_товара.Код товара
AND Поставщики.Код поставщика = Движение_товара.Код
поставщика
AND Движение_товара.Цена > 20;
```

Соединение таблицы со своей копией

В ряде приложений возникает необходимость одновременной обработки данных какой-либо таблицы и одной или нескольких ее копий, создаваемых на время выполнения запроса.

Например, при создании списков товаров возможен повторный ввод данных о каком-либо товаре с присвоением ему второго кода товара. Для выявления таких ошибок можно соединить таблицу Товары с ее временной копией, установив в WHERE фразе равенство значений всех одноименных столбцов этих таблиц кроме столбцов с кодом товара (для последних надо установить условие неравенства значений).

Временную копию таблицы можно сформировать, указав имя псевдонима за именем таблицы во фразе FROM. Так, с помощью фразы

```
FROM Товары X, Товары Y, Товары Z
будут сформированы три копии таблицы Товары с именами X, Y и Z.
```

В качестве примера соединения таблицы с ней самой сформируем запрос на вывод таких пар товаров таблицы Товары, в которых совпадают единицы измерения, а название первого товара пары меньше (по алфавиту), чем номер второго названия товара пары. Для этого можно создать запрос с одной копией таблицы Товары (Копия):

```
SELECT Наименование, Копия.Наименование, Ед_изм
FROM Товары, Товары Копия
WHERE Ед_изм = Копия.Ед_изм
AND Наименование < Копия.Наименование;
```

Получим результат вида

Первая.Товары	Вторая.Товары	Ед_изм
Товар_1	Товар_2	шт
Товар_1	Товар_5	шт
Товар_2	Товар_5	шт

Задание к СРСИ 8:

1. Определить какие операции реляционной алгебры были реализованы при помощи предложения Select при создании запросов 1-5 согласно индивидуального варианта задания
2. Написать скрипты для реализации хранимых процедур (связанных с выборкой данных) согласно индивидуального варианта задания