

## Лабораторная работа №10

### Многотабличные запросы

Такая выборка данных предполагает **соединение** нескольких таблиц для получения единого набора результатов, включающих записи и поля каждой таблицы. Соединение позволяет собрать данные, разделенные в процессе нормализации.

#### 1. Неявное соединение

Неявное соединение соответствует более старому SQL-стандарту. Таблицы, участвующие в соединении, задаются списком с разделяющими запятыми в разделе **FROM** запроса **SELECT**. Условия для связи таблиц задаются среди условий поиска раздела **WHERE**. Не существует специального синтаксиса для указания, какие условия используются для поиска, а какие – для соединения.

Формат разделов **FROM** и **WHERE** при неявном соединении таблиц имеет вид:

**FROM** Таблица1, Таблица2, ...

**WHERE** условие\_соединения [AND условие\_поиска] ...

где условие\_соединения имеет вид: Таблица1.Поле1 = Таблица2.Поле2

В качестве операции сравнения можно использовать: =, <, >, <=, !>, >=, !<, <>, !=

**Задание 1.** Используя неявное соединение, вывести информацию о преподавателях (фамилия, имя, отчество) и учебных группах, в которых они проводят занятия.

**Задание 2.** Используя неявное соединение, вывести ФИО студентов, обучающихся на определенном направлении подготовки (например, «педагогическое образование»), с указанием их учебной группы, профиля и курса. Результат отсортировать сначала по профилю, а затем по ФИО.

#### 2. Явное соединение

В стандарте SQL:92 введен более универсальный синтаксис явного соединения, которое осуществляется с помощью конструкции **JOIN...ON**. Структура конструкции **JOIN...ON** дает возможность отличать условие соединения от условий поиска. Неявное соединение может быть всегда заменено эквивалентным явным, обратное возможно не всегда.

Существует **три вида явного соединения: внутреннее, внешнее и перекрестное.**

Для объединения трех и более таблиц можно применять последовательность соединений.

Для соединения таблиц необходимо раздел **FROM** дополнить ключевыми словами **JOIN**, которое определяет соединяемые таблицы и метод соединения, и **ON**, указывающее общие для таблиц поля.

FROM Таблица1 JOIN Таблица2 ON Таблица1.Поле\_связи1 = Таблица2.Поле\_связи2

##### 2.1. Внутреннее соединение

При таком виде соединения (ключевое слово **INNER JOIN** или просто **JOIN**) сравниваются значения общих полей двух таблиц, возвращаются только те записи, в которых значения общих полей совпадают. Записи, для которых не имеется пары в связанной таблице, в результат не включаются.

*Например*, в таблице DimProduct базы данных AdventureWorksDW2012 хранится только код субкатегории товара. Чтобы получить список товаров с указанием их субкатегорий, необходимо соединить таблицы DimProduct и DimProductSubcategory:

```
SELECT DimProduct.EnglishProductName,  
       DimProductSubcategory.EnglishProductSubcategoryName  
FROM DimProduct INNER JOIN DimProductSubcategory  
       ON DimProduct.ProductSubcategoryKey=  
          DimProductSubcategory.ProductSubcategoryKey
```

При таком соединении товары, для которых не указана их субкатегория, не включаются в набор результатов.

При соединении **трех** таблиц **A**, **B** и **C** конечный набор результатов получается посредством соединения третьей таблицы **C** с результатом соединения первых двух таблиц **A** и **B**:

**FROM (A INNER JOIN B ON ...) INNER JOIN C ON ...**

**Задание 3.** Создайте запрос, выводящий без повторов для всех учебных групп полную информацию об изучаемых ими дисциплинах (полное название, количество часов всех видов).

**Задание 4.** Создайте запрос, подсчитывающий количество студентов, обучающихся на каждой специальности / направлении подготовки.

**Задание 5.** Создайте запрос, подсчитывающий количество преподавателей, работающих на каждой кафедре. Выведите только самые малочисленные кафедры, например, численностью менее 5 человек.

**Задание 6.** Выведите список преподавателей (ФИО), совмещающих работу на двух и более кафедрах.

**Задание 7.** Создайте запрос, выводящий информацию о занятиях: шифр группы, название дисциплины, вид занятия, ФИО преподавателя, записанные в одном поле. Предусмотрите, чтобы в итоговый набор данных вошли только дисциплины, для которых предусмотрены лекционные часы.

**Задание 8.** Создайте запрос, который для каждого преподавателя подсчитывает количество уникальных студентов, у которых данный преподаватель ведет занятия. Предусмотрите, чтобы в итоговый набор данных вошли только преподаватели, обучающие не менее 10 разных студентов.

## 2.2. Внешнее соединение

Такое соединение также возвращает записи, соответствующие условию соединения. Однако левые и правые соединения возвращают записи и не соответствующие указанному условию соединения. Такие соединения применяются для получения полного набора записей одной из таблиц.

При **левом соединении** (ключевое слово **LEFT JOIN**) в результат будут включены все записи левой таблицы (имя которой расположено слева от **LEFT JOIN**), независимо от того, есть для них соответствующая запись в правой таблице (имя которой расположено справа от **LEFT JOIN**) или нет.

*Например*, если в примере из предыдущего пункта вместо **INNER JOIN** поставить **LEFT JOIN**, то для товаров, у которых нет субкатегории, в поле EnglishProductSubcategoryName будет содержаться значение NULL.

При **правом соединении** (ключевое слово **RIGHT JOIN**) в результат включаются все записи правой таблицы, независимо от того, есть для них соответствующая строка в левой таблице.

**Задание 9.** Выполните соединение таблиц Группа (Sgroup) и Студент (Student) с помощью внутреннего, левого и правого соединений. Сравните результаты запросов. Сделайте вывод: во всех ли группах есть студенты; у всех ли студентов указана учебная группа?

**Задание 10.** Создайте запрос, выводящий список аудиторий с проектором, в которых не проводилось ни одного занятия.

**Задание 11.** Создайте запрос, выводящий список студентов, не получивших ни одной оценки.

**Задание 12.** С помощью запроса выясните, все ли преподаватели проводят занятия. Выведите ФИО тех преподавателей, которые не проводят занятий.

### 2.3. Произведение (Перекрестное соединение)

Операция произведение может быть реализована оператором перекрестного соединения. При таком соединении (ключевое слово **CROSS JOIN**) выводятся **все комбинации** записей таблиц, при этом не требуется указание совпадающих значений полей, поэтому условие **ON** опускается.

В базах данных перекрестные соединения используются для получения списка всех возможных комбинаций записей двух таблиц, т. е. число записей итогового набора равняется произведению числа записей первой таблицы на число записей второй.

*Например*, с помощью перекрестного соединения можно перечислить все возможные виды валюты для оплаты всех наименований товаров:

```
SELECT EnglishProductName, CurrencyName  
FROM DimProduct CROSS JOIN DimCurrency
```

**Задание 13.** Создайте запрос, который для каждого студента (Фамилия, Имя, Отчество) выводит названия всех дисциплин, которые он мог бы изучать. Предусмотрите, чтобы результирующий список был отсортирован по фамилиям и не содержал повторяющиеся записи. Проверьте, равно ли количество строк результата произведению количества строк таблиц Студент (Student) и Дисциплина (Subject). Обоснуйте ответ.

### 3. Подзапросы

**Подзапрос** – это оператор **SELECT**, включенный в другие запросы. Подзапросы применяются для разбивки сложного запроса на серию логических этапов. Их применение эффективно, если запрос использует записи, возвращенные другим запросом.

Существует два вида подзапросов: **вложенные** и **связанные**.

#### 3.1. Вложенные (некоррелированные) подзапросы

Вложенные подзапросы возвращают единственное значение либо список значений. Вложенный запрос выполняется один раз, а затем результирующее значение используется во внешнем запросе.

*Например*, чтобы определить коды заказчиков, купивших товары с максимальной ценой, сначала можно воспользоваться следующим запросом:

```
SELECT CustomerKey
FROM FactInternetSales
WHERE UnitPrice = (SELECT Max(ListPrice) FROM DimProduct)
```

Выполнение этого запроса осуществляется в два этапа: на первом этапе выполняется подзапрос – он выдает максимальную цену товара, эта цена используется на втором этапе при выполнении внешнего запроса – тогда получаем коды заказчиков.

Полученные коды заказчиков могут быть использованы далее в подзапросе для получения имен заказчиков:

```
SELECT FirstName, LastName
FROM DimCustomer
WHERE CustomerKey IN
    (SELECT CustomerKey
     FROM FactInternetSales
     WHERE UnitPrice =(SELECT Max(ListPrice) FROM DimProduct))
```

Внешняя часть этого запроса может быть оформлена и в виде соединения таблиц. Запросы с соединением таблиц обрабатываются значительно быстрее, поэтому если можно обойтись без подзапросов – лучше так и поступить.

**Задание 14.** Выведите список всех преподавателей с максимальным стажем.

**Задание 15.** Выведите список всех студентов, возраст которых ниже среднего по факультету.

#### Задание 16

А) Выведите список студентов (номера зачетных книжек), получивших по определенной дисциплине (например, «Базы данных») оценку выше средней.

Б) Используя только вложенные запросы (т.е. не используйте соединение таблиц) измените ранее созданный запрос (16 А) таким образом, чтобы вместо номеров зачетных книжек выводились Фамилия, Имя, Отчество студентов, получивших по определенной дисциплине (например, «Базы данных») оценку выше средней.

В) Решите задачу 16 Б) с использованием соединения таблиц для внешнего запроса.

### 3.2. Связанные (коррелированные) подзапросы

Связанные подзапросы используют данные внешнего запроса, причем связанный запрос выполняется один раз для каждой записи внешнего запроса.

*Например*, для определения списка заказчиков, заказавших в общей сложности более 20 единиц товара, необходимо:

```
SELECT FirstName, LastName
FROM DimCustomer C
WHERE 20 < (SELECT sum(OrderQuantity)
            FROM FactInternetSales S WHERE C.CustomerKey=S.CustomerKey)
```

В связанных подзапросах для краткости часто применяют псевдонимы таблиц. В примере для таблицы DimCustomer определен псевдоним C, а для таблицы FactInternetSales – S.

Однако, использование связанных подзапросов неэффективно, лучше преобразовывать их в соединения таблиц, что позволяет оптимизатору запросов выбирать наилучший способ обработки данных.

*Например*, для товаров отобразить сведения о самом последнем их заказе: наименование товара, количество заказанного товара и номер заказа.

```
SELECT EnglishProductName, OrderQuantity, SalesOrderNumber
FROM DimProduct P INNER JOIN FactInternetSales S1
                    ON P.ProductKey=S1.ProductKey
WHERE OrderDateKey = (SELECT max(OrderDateKey)
                      FROM FactInternetSales S2
                      WHERE S2.ProductKey=P.ProductKey)
```

**Задание 17.** Для каждого студента отобразите полную информацию о самом первом его зачете/экзамене: № зачетки, ФИО студента, Дата, Название дисциплины, Фамилия И.О. преподавателя, Оценка.

**Задание 18.** Вывести список самых «строгих» преподавателей по каждой дисциплине, т.е. у кого средний балл ниже среднего балла по дисциплине в целом.

### 4. Объединение, пересечение, разность

Оператор **UNION** объединяет результаты двух и более операторов **SELECT** и применяется в случае, когда данные нельзя получить при помощи одного запроса.

Для получения единого итогового набора данных необходимо написать отдельные операторы **SELECT** и объединить их при помощи оператора **UNION**, при этом, в отличие от соединения, записи в итоговый набор добавляются друг за другом:

```
SELECT СписокПолейПервойТаблицы
FROM ИмяПервойТаблицы
UNION [ALL]
SELECT СписокПолейВторойТаблицы
FROM ИмяВторойТаблицы
[,...]
```

По умолчанию повторяющиеся записи удаляются, для получения всех записей необходимо указать ключевое слово **ALL**. Необходимо так же учитывать, что в списках полей всех объединяемых запросов количество полей, порядок их следования и их типы должны быть соответственно одинаковы (совместимость по типу).

Имена полей итогового набора берутся из первого запроса, поэтому создание псевдонимов полей выполняется в нем. Для получения отсортированного набора данных раздел **ORDER BY** указывается после последнего оператора **SELECT**.

Пересечение запросов, выполняется с помощью оператора **INTERSECT**, который выполняется аналогично оператору **UNION**.

Вычитание выполняется с помощью оператора **EXCEPT**, который выполняется аналогично оператору **UNION**.

**Задание 19.** Выведите единый набор данных содержащий ФИО (в одном поле) студентов и преподавателей, а также их структурные подразделения (учебная группа или кафедра + факультет/институт), даты рождения и телефоны. Данные отсортируйте по столбцам Структурное подразделение и ФИО.

**Задание 20.** Создайте запрос, выводящий список преподавателей, которые одновременно являются и студентами (например, учатся в аспирантуре или магистратуре). Совпадение определяем по ФИО и дате рождения.

**Задание 21.** Вывести преподавателей, не поставивших на экзамене ни одной «двойки».

**Задание 22.** Вывести список названий дисциплин, которые изучают и группы МКб, и группы ФИб и группы ПМИб. Совпадение определяем по названию дисциплины.

**Задание 23.** Создайте запрос, выводящий студентов, которые не явились на все зачеты и экзамены.

**Задание 24.** Создайте запрос, выводящий наименования дисциплин, которые не изучает ни один студент (нет занятий по таким дисциплинам и нет ни одной оценки).