

## Лабораторна робота №8. Вибірка даних з кількох таблиць

### 4. Выборка данных из нескольких таблиц

#### 4.1 Определение соединений

Соединение используется для выборки данных из более чем одной таблицы.

```
SELECT Table1.column, table2.column
FROM table1, table2
WHERE table1.column1 = table2.column2;
```

Условие соединения указывается в предложении WHERE.

- Если одно и то же имя столбца присутствует более чем в одной таблице, к имени столбца добавляется имя таблицы в виде префикса.
- Для выборки данных из нескольких таблиц требуется условие соединения. Строки одной таблицы соединяются со строками другой с помощью общих значений в соответствующих столбцах — обычно в столбцах первичных и внешних ключей.
- Для вывода данных из двух или более взаимосвязанных таблиц можно задать простое условие соединения в предложении WHERE.

Синтаксис (см. команду выше):

Table.column	таблица и столбец, из которых производится выборка.
table1.column1 = table2.column2	условие, соединяющее таблицы (или устанавливающее связь между ними).

Отметим некоторые моменты, связанные с применением соединений:

- Для упрощения чтения команд SELECT, соединяющих таблицы, и расширения возможностей доступа к БД указывайте имена таблиц перед именами столбцов.
- Если в нескольких таблицах есть столбцы с одинаковыми именами, указание имени таблицы перед именем столбца обязательно.
- Для соединения  $n$  таблиц требуется, по крайней мере,  $(n-1)$  условий соединения. Следовательно, для соединения четырех таблиц требуются, по крайней мере, три условия. Это правило может не относиться к таблицам с составным первичным ключом — в этом случае для однозначной идентификации строки требуется более одного столбца.

#### 4.2 Декартово произведение (перекрестное соединение, кросс-джойн)

Декартово произведение образуется, если:

- Опущено условие соединения.
- Условие соединения недействительно.
- Все строки первой таблицы соединяются со всеми строками второй таблицы.

Во избежание получения декартова произведения, предложение WHERE всегда должно включать допустимое условие соединения.

Если условие соединения недействительно или опущено, результатом запроса будет декартово произведение двух таблиц, включающее все комбинации строк. Все строки первой таблицы соединяются со всеми строками второй таблицы.

Как правило, декартово произведение содержит большое количество строк, и результат получается не очень полезным. Поэтому, если вывод всех комбинаций строк не требуется, следует использовать предложение WHERE с допустимым условием соединения.

Декартово произведение образуется при отсутствии условия соединения таблиц. Приведенный пример показывает вывод фамилии служащего и оклада из таблиц inspector и oklad. Т.к. предложение WHERE отсутствует, все строки таблицы inspector (8 строк) соединяются со всеми строками таблицы oklad (8 строк), что дает 64 выходных строк.

*Пример*

```
SELECT      fio, oklad      FROM      inspector, oklad;
```

### 4.3 Виды соединений

Имеется два основных типа условий соединения:

1. Эквисоединения
2. Не-эквисоединения

Дополнительные способы соединения:

1. Внешние соединения
2. Соединения таблицы с собой

#### 4.3.1 Эквисоединения

Для определения оклада, кот. получает служащий, значение столбца KODINSP в таблице INSPECTOR сравнивается со значениями KODINSP в таблице OKLAD. Отношение между таблицами INSPECTOR и OKLAD представляет собой эквисоединение (equijoin), т.е. значения столбца KODINSP в обеих таблицах должны быть равны. Часто эти столбцы являются компонентами главного и внешнего ключа.

Примечание: эквисоединения называются также простыми (simple) или внутренними (inner) соединениями.

*Пример*

Выборка записей с помощью эквисоединений

```
SELECT      fio, oklad      FROM      inspector i, oklad o WHERE  
i.kodinsp=o.kodinsp;
```

В вышеприведенном примере:

Предложение SELECT задает имена столбцов для выборки:

- фамилия служащего (столбец таблицы INSPECTOR)
- оклад (столбец таблицы OKLAD)

Предложение FROM задает две таблицы, к которым должна обращаться база данных:

- Таблица INSPECTOR
- Таблица OKLAD

Предложение WHERE указывает, как соединяются таблицы:

- i.kodinsp=o.kodinsp

Т.к. столбец KODINSP присутствует в обеих таблицах, имени столбца должно предшествовать имя таблицы (или псевдоним) во избежание путаницы.

## **Лабораторная работа 4.1 Создание простого соединения**

### **Задание:**

Создайте запрос, который бы возвращал информацию из таблиц employees и jobs схемы hr. При этом должна возвращаться информация:

- об имени сотрудника (таблица employees);
- о фамилии сотрудника (таблица employees);
- о разнице между максимальной заработной платой для данной должности (из таблицы jobs) и реальной заработной платой (из таблицы employees).

При этом сотрудники, для которых такая разница максимальна, должны выводиться первыми.

Результат выполнения запроса должен выглядеть так, как представлено на рис. 4.1-1.

Имя	Фамилия	Должность	Разница с максимальной
1 Steven	King	President	16000
2 Lex	De Haan	Administration Vice President	13000
3 Neena	Kochhar	Administration Vice President	13000
4 Eleni	Zlotkey	Sales Manager	9500
5 Gerald	Cambraut	Sales Manager	9000
6 Alberto	Errazuriz	Sales Manager	8000
7 Karen	Partners	Sales Manager	6500
8 John	Russell	Sales Manager	6000
9 Sundita	Kumar	Sales Representative	5900

Рис. 4.1-1

### 4.3.2 Различение одноименных столбцов

Для различения одноименных столбцов из разных таблиц используются префиксы в виде имен таблиц.

- Использование префиксов в виде имен таблиц увеличивает производительность.
- Одноименные столбцы из разных таблиц можно различать по их псевдонимам.

Во избежание путаницы в столбцах именам столбцов в предложении WHERE должны предшествовать имена таблиц.

Если одноименных столбцов в таблицах нет, то для различения столбцов указание имен таблиц не требуется. Но благодаря префиксам увеличивается производительность, потому что сервер Oracle точно знает, где искать столбцы.

Необходимость различения одноименных столбцов может возникнуть и в других предложениях – например, SELECT или ORDER BY.

### 4.3.3 Дополнительные условия поиска с оператором AND

Предложение WHERE позволяет задавать не только условия соединения таблиц, но и другие критерии. Например, чтобы получить оклад для служащего по фамилии Савилов, требуется дополнительное условие в предложении WHERE.

```
SELECT fio, oklad FROM inspector i, oklad o WHERE i.kodinsp=o.kodinsp and i.fio='Савилов Ю.Н.';
```

### 4.3.4 Псевдонимы таблиц

Использование псевдонимов таблиц упрощает запросы.

```
SELECT fio, oklad FROM inspector i, oklad o WHERE i.kodinsp=o.kodinsp and i.fio='Савилов Ю.Н.';
```

### РАВНОСИЛЬНО

```
SELECT inspector .fio, oklad.oklad FROM inspector , oklad WHERE inspector.kodinsp = oklad..kodinsp and inspector.fio='Савилов Ю.Н.';
```

Различение столбцов с помощью имен таблиц требует времени - особенно, если имена у таблиц длинные. Поэтому вместо имен таблиц можно использовать их псевдонимы (aliases). Как и псевдонимы столбцов, которые дают альтернативное имя столбцу, псевдонимы таблиц дают альтернативное имя таблице. Использование псевдонимов таблиц уменьшает объем кода SQL и, следовательно, экономит память.

Некоторые рекомендации по использованию псевдонимов:

1. Псевдонимы таблиц могут быть длиной до тридцати символов, но чем они короче, тем лучше.

2. Если в предложении FROM для указания таблицы используется псевдоним, этот псевдоним таблицы должен использоваться вместо имени таблицы во всем предложении SELECT.
3. Следует выбирать осмысленные псевдонимы.
4. Действие псевдонима таблицы распространяется лишь на текущую команду SELECT.

### 4.3.5 Соединение более чем двух таблиц

Иногда требуется соединить более двух таблиц. Например, чтобы получить фамилию, оклад, и номера предприятий, зарегистрированных на каждом инспекторе, необходимо соединить таблицы inspector, oklad и kart.

```
SELECT inspector .fio, oklad.oklad, kart.n_pred FROM inspector , oklad , kart
WHERE inspector.kodinsp = oklad..kodinsp and kart.kodinsp=inspector.kodinsp;
```

## Лабораторная работа 4.2 Соединение более чем двух таблиц

### Задание:

Напишите запрос, который возвращал бы информацию из таблиц employees и countries схемы hr. При этом для каждого сотрудника должна возвращаться информация:

- об имени сотрудника (таблица employees);
- о фамилии сотрудника (таблица employees);
- о стране, в которой он находится (таблица countries).

Убедитесь, что в результате выполнения вашего запроса не получается декартово произведение (должно вернуться всего 106 записей). Для написания этого запроса необходимо самостоятельно проанализировать связи между таблицами в схеме hr, используя, например, средства Oracle SQL Developer. Результат выполнения запроса должен выглядеть так, как представлено на рис. 4.2-1.

	Имя	Фамилия	Страна
1	Donald	OConnell	United States of America
2	Douglas	Grant	United States of America
3	Jennifer	Whalen	United States of America
4	Michael	Hartstein	Canada
5	Pat	Fay	Canada
6	Susan	Mavris	United Kingdom
7	Hermann	Baer	Germany
8	Shelley	Higgins	United States of America
9	William	Gietz	United States of America

Рис. 4.2-1

## 4.4 Не-эквисоединения

Допустим, существует таблица sqlgrade с диапазонами окладов (поля hisal, losal). Отношение между таблицами OKLAD и SALGRADE не является эквисоединением, поскольку ни один столбец таблицы OKLAD не соответствует точно столбцу в таблице SALGRADE. Связь между двумя таблицами такова, что значение столбца OKLAD в таблице OKLAD находится между значениями столбцов LOSAL и HISAL таблицы SALGRADE. Для такого соединения оператор "равно" (=) использоваться не может.

*Пример.* Выборка записей с помощью не-эквисоединений

SQL>	SELECT	oklad, losal, hisal
2	FROM	oklad, salgrade
3	WHERE	oklad
4	BETWEEN	losal AND hisal;

В примере не-эквисоединение создается для оценки категории заработной платы служащего. Заработная плата должна быть между любой парой нижней и верхней границ диапазона. Важно отметить, что при выполнении запроса каждый служащий появляется только один раз. Ни один служащий в списке не повторяется. Для этого существует две причины:

- Ни одна строка в таблице SALGRADE не содержит категорий зарплат, перекрывающих друга. Т.е. заработная плата служащего может быть между нижним и верхним значениями зарплат только на одной строке таблицы SALGRADE.
- Заработная плата всех служащих находится в пределах, обозначенных таблицей SALGRADE. Т.е. ни один служащий не зарабатывает меньше, чем самое малое значение в столбце LOSAL или больше, чем самое большое значение в столбце HISAL.

Примечание: можно было бы использовать и другие операторы (например,  $\leq$  и  $\geq$ ), но оператор BETWEEN является самым простым. Не забудьте, что в операторе BETWEEN сначала указывается нижняя граница, а затем верхняя.

## Лабораторная работа 4.3 Не экви-соединения

### Задание:

Напишите запрос на основе информации из таблиц employees и jobs схемы hr. При этом должна возвращаться информация:

- об имени сотрудника (таблица employees);
- о фамилии сотрудника (таблица employees);
- о его заработной плате (столбец salary в таблице employees);
- о минимальной заработной плате для его должности (столбец min\_salary в таблице jobs).

При этом информация должна возвращаться только для тех сотрудников, чья заработная плата отличается от минимальной для данной должности не более чем на 20 процентов (для проверки: должно вернуться 38 записей). Результат выполнения запроса должен выглядеть так, как представлено на рис. 4.3-1.

	Имя	Фамилия	Окл...	Минимальный оклад
1	Donald	OConnell	2600	2500
2	Douglas	Grant	2600	2500
3	Steven	King	24000	20000
4	Neena	Kochhar	17000	15000
5	Lex	De Haan	17000	15000
6	David	Austin	4800	4000
7	Valli	Pataballa	4800	4000
8	Diana	Lorentz	4200	4000
9	Shelli	Baida	2900	2500
10	Sigal	Tobias	2800	2500

Рис. 4.3.1

## 4.5. Внешние соединения

### 4.5.1 Выборка записей, не имеющих прямых совпадений, с помощью внешних соединений

Если строка не удовлетворяет условию соединения, она не включается в результат запроса. Например, при эквисоединении для вывода фамилии и номера служащих вместе с фамилиями и номерами их начальников, в список не попадут служащие, не имеющие начальников.

```
select i.fio ,i.kodinsp ,K.fio ,K.kodinsp from inspector i,inspector k where i.mgr=k.kodinsp
```

Чтобы вывести всех служащих, нужно написать такой select:

```
select i.fio ,i.kodinsp , K.fio , K.kodinsp from inspector i, inspector k
where i.mgr=k.kodinsp(+)
```

Внешнее соединение используется для выборки строк, не удовлетворяющих обычным условиям соединения.

Оператором внешнего соединения является знак плюс (+).

```
SELECT table.column, table.column
FROM table1, table2
WHERE table1.column (+)= table2.column;
```

```
SELECT table.column, table.column
FROM table1, table2
WHERE table1.column = table2.column(+);
```

Недостающие строки можно получить, если использовать в условии соединения оператор внешнего соединения (outer join). Этим оператором является символ "плюс" в скобках (+), и помещается он на "стороне" соединения с недостающей информацией. В результате применения оператора создается одна или более строк с неопределенными значениями, к которым можно присоединить одну или более строк из таблицы, где имеется вся необходимая информация.

Синтаксис:

*table1.column* = условие соединения таблиц (или установления связи между ними).

*table2.column* (+) символ внешнего соединения; он может находиться на любой стороне условия в предложении WHERE, но не на обеих сторонах одновременно. Символ внешнего соединения помещается за именем столбца таблицы, где не оказывается соответствующих строк.

Ограничения для внешних соединений:

- Оператор внешнего соединения может использоваться только на одной стороне выражения — на стороне, где недостаточно информации. Он возвращает строки таблицы, для которых нет соответствующей строки в другой таблице.
- Условия, предполагающие внешнее соединение, не могут включать оператор IN или быть связанными с другими условиями с помощью оператора OR

## **Лабораторная работа 4.4 Внешние джойны**

### **Задание:**

Напишите запрос на основе информации из таблиц employees и departments схемы hr. Запрос должен возвращать информацию:

- о названии отдела (столбец department\_name в таблице departments);
- о имени сотрудника;
- о фамилии сотрудника.

При этом должна выводиться также информация о тех отделах, для которых сотрудников в таблице employees нет. Информация должна быть упорядочена по названиям отделов.

Результат выполнения запроса должен выглядеть так, как представлено на рис. Лаб 4.4-1.

	Отдел	Имя	Фамилия
1	Accounting	Shelley	Higgins
2	Accounting	William	Gietz
3	Administration	Jennifer	Whalen
4	Benefits	(null)	(null)
5	Construction	(null)	(null)
6	Contracting	(null)	(null)
7	Control And Credit	(null)	(null)
8	Corporate Tax	(null)	(null)
9	Executive	Lex	De Haan
10	Executive	Steven	King

Рис. Лаб. 4.4-1.

## 4.6 Соединение таблицы с собой

Иногда требуется соединить таблицу с собой.

Для имитации двух таблиц в предложении FROM используется два псевдонима одной и той же таблицы

На этом занятии мы увидели, что существует много способов соединения таблиц. Общим, однако, является то, что условие соединения задается в предложении WHERE. Выбор того или иного метода зависит от желаемого результата и используемых структур данных.

```
SELECT    table1.column, table2.column
FROM      table1, table2
WHERE     table1.column1 = table2.column2;
```