

42. Напишите команду, которая выдает список фамилий студентов, с комментарием ‘успевает’ у студентов, имеющих все положительные оценки, комментарием ‘не успевает’ для сдававших экзамены, но имеющих хотя бы одну неудовлетворительную оценку, и комментарием ‘не сдавал’ – для всех остальных. В выводимом результате фамилии студентов упорядочить по алфавиту.
43. Выведите объединенный список студентов и преподавателей, живущих в Москве, с соответствующими комментариями ‘студент’ или ‘преподаватель’.
44. Выведите объединенный список студентов и преподавателей Воронежского государственного университета с соответствующими комментариями ‘студент’ или ‘преподаватель’.

2.19. Соединение таблиц с использованием оператора JOIN

Если в операторе **SELECT** после ключевого слова **FROM** указывается не одна, а две таблицы, то в результате выполнения запроса, в котором отсутствует предложение **WHERE**, каждая строка одной таблицы будет соединена с каждой строкой второй таблицы. Такая операция называется *декартовым произведением* или *полным (CROSS) соединением* таблиц базы данных. Сама по себе эта операция не имеет практического значения, более того, при ошибочном использовании она может привести к неожиданным нештатным ситуациям, так как в этом случае в ответе на запрос количество записей будет равно произведению числа записей в соединяемых таблицах, то есть может оказаться чрезвычайно большим. Соединение таблиц имеет смысл тогда, когда соединяются *не все* строки исходных таблиц, а только те, которые интересуют пользователя. Такое ограничение может быть осуществлено с помощью использования в запросе соответствующего условия в предложении **WHERE**. Таким образом, SQL позволяет выводить информацию из нескольких таблиц, связывая их по значениям определенных полей.

Например, если необходимо получить фамилии студентов (таблица STUDENT) и для каждого студента – названия университетов (таблица

UNIVERSITY), расположенных в городе, где живет студент, то необходимо получить все комбинации записей о студентах и университетах в обеих таблицах, в которых значение поля CITY совпадает. Это можно сделать с помощью следующего запроса.

```
SELECT STUDENT.SURNAME, UNIVERSITY.UNIV_NAME, STUDENT.CITY
FROM STUDENT, UNIVERSITY
WHERE STUDENT.CITY = UNIVERSITY.CITY;
```

Соединение, использующее предикаты, основанные на равенствах, называется *эквисоединением*. Рассмотренный пример соединения таблиц относится к виду так называемого *внутреннего (INNER) соединения*. При таком типе соединения соединяются только те строки таблиц, для которых является истинным предикат, задаваемый в предложении **ON** выполняемого запроса.

Приведенный выше запрос может быть записан иначе, с использованием ключевого слова **JOIN**.

```
SELECT STUDENT.SURNAME, UNIVERSITY.UNIV_NAME, STUDENT.CITY
FROM STUDENT INNER JOIN UNIVERSITY
ON STUDENT.CITY = UNIVERSITY.CITY;
```

Ключевое слово **INNER** в запросе может быть опущено, так как эта опция в операторе **JOIN** действует по умолчанию.

Рассмотренный выше случай полного соединения (декартова произведения таблиц) с использованием ключевого слова **JOIN** будет выглядеть следующим образом

```
SELECT * FROM STUDENT JOIN UNIVERSITY;
```

что эквивалентно

```
SELECT * FROM STUDENT, UNIVERSITY;
```

Заметим, что в СУБД Oracle задаваемый стандартом языка SQL оператор **JOIN** не поддерживается.

2.19.1. Операции соединения таблиц посредством ссылочной целостности

Информация в таблицах STUDENT и EXAM_MARKS уже связана посредством поля STUDENT_ID. В таблице STUDENT поле STUDENT_ID является первичным ключом, а в таблице EXAM_MARKS, ссылающимся на него внешним ключом. Состояние связанных таким образом таблиц называется состоянием ссылочной целостности. В данном случае ссылочная целостность этих таблиц подразумевает, что *каждому* значению поля STUDENT_ID в таблице EXAM_MARKS *обязательно* соответствует *такое же значение* поля STUDENT_ID в таблице STUDENT. Другими словами, в таблице EXAM_MARKS не может быть записей, имеющих идентификаторы студентов, которых нет в таблице STUDENT. Стандартное применение операции соединения состоит в извлечении данных в терминах этой связи.

Чтобы получить список фамилий студентов с полученными ими оценками и идентификаторами предметов можно использовать следующий запрос:

```
SELECT SURNAME, MARK, SUBJ_ID
FROM STUDENT, EXAM_MARKS
WHERE STUDENT.STUDENT_ID = EXAM_MARKS.STUDENT_ID;
```

Тот же самый результат может быть получен при использовании в запросе для задания операции соединения таблиц ключевого слова **JOIN**. Запрос с оператором **JOIN** выглядит следующим образом

```
SELECT SURNAME, MARK
FROM STUDENT JOIN EXAM_MARKS
ON STUDENT.STUDENT_ID = EXAM_MARKS.STUDENT_ID;
```

Хотя выше речь шла о соединении двух таблиц, можно сформировать запросы путем соединения более чем двух таблиц.

Пусть требуется найти фамилии всех студентов, получивших неудовлетворительную оценку, вместе с названиями предметов обучения, по которым получена эта оценка.

```
SELECT SUBJ_NAME, SURNAME, MARK
FROM STUDENT, SUBJECT, EXAM_MARKS
WHERE STUDENT.STUDENT_ID = EXAM_MARKS.STUDENT_ID
AND SUBJECT.SUBJ_ID = EXAM_MARKS.SUBJ_ID
```

AND EXAM_MARKS.MARK = 2;

То же самое с использованием оператора **JOIN**

```
SELECT SUBJ_NAME, SURNAME, MARK
FROM STUDENT JOIN SUBJECT JOIN EXAM_MARKS
ON STUDENT.STUDENT_ID = EXAM_MARKS.STUDENT_ID
AND SUBJECT.SUBJ_ID = EXAM_MARKS.SUBJ_ID
AND EXAM_MARKS.MARK = 2;
```

2.19.2. Внешнее соединение таблиц

Как отмечалось ранее, при использовании *внутреннего* (**INNER**) соединения таблиц соединяются только те их строки, в которых совпадают значения полей, задаваемые в предложении **WHERE** запроса. Однако во многих случаях это может привести к нежелательной потере информации. Рассмотрим еще раз приведенный выше пример запроса на выборку списка фамилий студентов с полученными ими оценками и идентификаторами предметов. При использовании, как это было сделано в рассматриваемом примере, внутреннего соединения в результат запроса не попадут студенты, которые еще не сдавали экзамены и которые, следовательно, отсутствуют в таблице EXAM_MARKS. Если же необходимо иметь записи об этих студентах в выдаваемом запросом списке, то можно присоединить сведения о студентах, не сдававших экзамен, путем использования оператора **UNION** с соответствующим запросом. Например, следующим образом:

```
SELECT SURNAME, CAST MARK AS CHAR(1), CAST SUBJ_ID AS CHAR(10)
FROM STUDENT, EXAM_MARKS
WHERE STUDENT.STUDENT_ID = EXAM_MARKS.STUDENT_ID
UNION
SELECT SURNAME, CAST NULL AS CHAR(1), CAST NULL AS CHAR(10)
FROM STUDENT
WHERE NOT EXIST
  (SELECT *
   FROM EXAM_MARKS
   WHERE STUDENT.STUDENT_ID = EXAM_MARKS.STUDENT_ID);
```

(здесь функция преобразования типов **CAST** используется для обеспечения совместимости типов полей объединяемых запросов).

Нужный результат, однако, может быть получен и путем использования *внешнего соединения*, точнее одной из его разновидностей – *левого внешнего соединения*, с использованием которого запрос будет выглядеть следующим образом:

```
SELECT SURNAME, MARK
FROM STUDENT LEFT OUTER JOIN EXAM_MARKS
ON STUDENT.STUDENT_ID = EXAM_MARKS.STUDENT_ID;
```

При использовании *левого* соединения расширение выводимой таблицы осуществляется за счет записей входной таблицы, имя которой указано *слева* от оператора **JOIN**.

Следует заметить, что нотация запросов с внешним соединением в СУБД ORACLE отличается от приведенной нотации, задаваемой стандартом языка SQL. В нотации, используемой в Oracle, этот же запрос будет иметь вид

```
SELECT SURNAME, MARK, SUBJ_ID
FROM STUDENT, EXAM_MARKS
WHERE STUDENT.STUDENT_ID = EXAM_MARKS.STUDENT_ID(+);
```

Знак (+) ставится у той таблицы, которая дополняется записями с **NULL**-значениями, чтобы при соединении таблиц в выходное отношение попали и те записи другой таблицы, для которых в таблице со знаком (+) не находится строк с соответствующими значениями атрибутов, используемых для соединения. То есть для *левого* внешнего соединения (по нотации стандарта SQL) в запросе ORACLE-SQL указатель (+) ставится у *правой* таблицы.

Приведенный выше запрос может быть реализован и с применением *правого внешнего соединения*. Он будет иметь следующий вид

```
SELECT SURNAME, MARK
FROM EXAM_MARKS RIGHT OUTER JOIN STUDENT
ON EXAM_MARKS.STUDENT_ID = STUDENT.STUDENT_ID;
```

Здесь таблица STUDENT, за счет записей которой осуществляется расширение выводимой таблицы, стоит справа от оператора **JOIN**.

В нотации Oracle этот запрос будет выглядеть следующим образом.

```
SELECT SURNAME, MARK, SUBJ_ID
FROM STUDENT, EXAM_MARKS
WHERE EXAM_MARKS.STUDENT_ID(+) = STUDENT.STUDENT_ID;
```

Видно, что использование внешнего правого или левого соединения

позволяет существенно упростить запрос, сделать его запись более компактной.

Иногда возникает необходимость включения в результат запроса записей из обеих (правой и левой) соединяемых таблиц, для которых не удовлетворяется условие соединения. Такое соединение называется *полным внешним соединением* и осуществляется указанием в запросе ключевых слов **FULL OUTER JOIN** или **UNION JOIN**.

УПРАЖНЕНИЯ

45. Напишите запрос, который выполняет вывод данных о фамилиях, *сдававших* экзамены студентов, вместе с идентификаторами каждого сданного ими предмета обучения.
46. Напишите запрос, который выполняет выборку значений фамилии *всех* студентов с указанием для студентов, сдававших экзамены, идентификаторов сданных ими предметов обучения.
47. Напишите запрос, который выполняет вывод данных о фамилиях студентов, *сдававших* экзамены, вместе с наименованиями каждого сданного ими предмета обучения.
48. Напишите запрос на выдачу для каждого студента названий всех предметов обучения, по которым этот студент получил оценку 4 или 5.
49. Напишите запрос на выдачу данных о названиях всех предметов, по которым студенты получили только хорошие (4 и 5) оценки. В выходных данных должны быть приведены фамилии студентов, названия предметов и оценка.
50. Напишите запрос, который выполняет вывод списка университетов с рейтингом, превышающим 300, вместе со значением максимального размера стипендии, получаемой студентами в этих университетах.
51. Напишите запрос на выдачу списка фамилий студентов (в алфавитном порядке) вместе со значением рейтинга университета, где каждый из них учится, включив в список и тех студентов, для которых в базе данных не указано место их учебы.