

подзапроса.

2.11. Использование оператора EXISTS

Используемый в SQL оператор **EXISTS** (СУЩЕСТВУЕТ) генерирует значение **ИСТИНА** или **ЛОЖЬ**, подобно булеву выражению. Используя подзапросы в качестве аргумента, этот оператор оценивает результат выполнения подзапроса как истинный, если этот подзапрос генерирует выходные данные, то есть в случае *существования* (возврата) хотя бы одного найденного значения. В противном случае результат подзапроса – ложный. Оператор **EXISTS** не может принимать значение **unknown** (неизвестно).

Пусть, например, нужно извлечь из таблицы EXAM_MARKS данные о студентах, получивших хотя бы одну неудовлетворительную оценку.

```
SELECT DISTINCT STUDENT_ID
FROM EXAM_MARKS A
WHERE EXISTS
  (SELECT *
   FROM EXAM_MARKS B
   WHERE MARK < 3
   AND B.STUDENT_ID=A.STUDENT_ID);
```

При использовании связанных подзапросов предложение **EXISTS** анализирует каждую строку таблицы, на которую имеется ссылка во внешнем запросе. Главный запрос получает строки-кандидаты на проверку условия. Для каждой строки-кандидата выполняется подзапрос. Как только подзапрос находит строку, где в столбце MARK значение удовлетворяет условию, он прекращает выполнение и возвращает значение **ИСТИНА** внешнему запросу, который затем анализирует свою строку-кандидата.

Например, требуется получить идентификаторы предметов обучения, экзамены по которым сдавались не одним, а несколькими студентами:

```

SELECT DISTINCT SUBJ_ID
FROM EXAM_MARKS A
WHERE EXISTS
( SELECT *
  FROM EXAM_MARKS B
  WHERE A.SUBJ_ID = B.SUBJ_ID
    AND A.STUDENT_ID <> B.STUDENT_ID);

```

Часто **EXISTS** применяется с оператором **NOT** (по-русски **NOT EXISTS** интерпретируется, как “*не существует ...*”). Если предыдущий запрос сформулировать следующим образом – найти идентификаторы предметов обучения, которые сдавались одним и только одним студентом (другими словами, для которых не существует другого сдававшего студента), то достаточно просто поставить **NOT** перед **EXISTS**.

Следует иметь в виду, что в подзапросе, указываемом в операторе **EXISTS**, *нельзя использовать агрегирующие функции*.

Возможности применения вложенных запросов весьма разнообразны. Например, пусть из таблицы **STUDENT** требуется извлечь строки для каждого студента, сдавшего более одного предмета.

```

SELECT *
FROM STUDENT FIRST
WHERE EXISTS
( SELECT SUBJ_ID
  FROM EXAM_MARKS SECOND
  GROUP BY SUBJ_ID
  HAVING COUNT(SUBJ_ID) > 1
  WHERE FIRST.STUDENT_ID = SECOND.STUDENT_ID);

```

УПРАЖНЕНИЯ

33. Напишите запрос с **EXISTS**, позволяющий вывести данные обо всех студентах обучающихся в вузах, имеющих рейтинг выше 300.
34. Напишите предыдущий запрос, используя соединения.
35. Напишите запрос с **EXISTS**, выбирающий сведения обо всех студентах, для которых в том же городе, где живет студент, существуют

университеты, в которых он не учится.

36. Напишите запрос, выбирающий из таблицы SUBJECT данные о названиях предметов обучения, экзамены по которым сданы более чем одним студентом.

2.12. Операторы сравнения с множеством значений IN, ANY, ALL

Операторы сравнения с множеством значений имеют следующий смысл.

IN	<i>Равно</i> любому из значений, полученных во внутреннем запросе.
NOT IN	<i>Не равно</i> ни одному из значений, полученных во внутреннем запросе.
= ANY	То же, что и IN . Соответствует логическому оператору OR .
> ANY, > = ANY	<i>Больше, чем</i> (либо <i>больше</i> или <i>равно</i>) любое полученное число. Эквивалентно > или > = для самого меньшего полученного числа.
< ANY, < = ANY	<i>Меньше, чем</i> (либо <i>меньше</i> или <i>равно</i>) любое полученное число. Эквивалентно < или < = для самого большего полученного числа.
= ALL	Равно всем полученным значениям. Эквивалентно логическому оператору AND .
> ALL, > = ALL	<i>Больше, чем</i> (либо <i>больше</i> или <i>равно</i>) все полученные числа. Эквивалентно > или > = для самого большего полученного числа.
< ALL, < = ALL	<i>Меньше, чем</i> (либо <i>меньше</i> или <i>равно</i>) все полученные числа. Эквивалентно < или < = самого меньшего полученного числа.

Следует иметь в виду, что в некоторых СУБД поддерживаются не все из этих операторов.

Примеры запросов с использованием приведенных операторов.

Выбрать сведения о студентах, проживающих в городе, где расположен университет, в котором они учатся.