

Использование оператора EXISTS

Используемое в SQL ключевое слово **EXISTS** (СУЩЕСТВУЕТ) представляет собой предикат, принимающий значение *истина* или *ложь*. 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 *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 при выполнении подзапроса находится строка, в которой значение в столбце MARK удовлетворяет условию, его выполнение прекращается и возвращается 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 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.

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);
```

УПРАЖНЕНИЕ

1. Напишите запрос с EXISTS, позволяющий вывести данные обо всех студентах обучающихся в вузах; имеющих рейтинг выше 300.
2. Напишите предыдущий запрос, используя соединения.
3. Напишите запрос с EXISTS, выбирающий сведения о каждом студенте, для которого в городе его проживания имеется хотя бы один университет, в котором он не учится.
4. Напишите запрос, выбирающий: из таблицы: SUBJECT данные о названиях предметов обучения, экзамены по которым сданы более чем одним студентом.
5. Напишите запрос для получения списка городов проживания студентов, в которых есть хотя бы один университет.
6. Напишите запрос для получения списка городов проживания студентов, в которых нет ни одного университета.
7. Напишите запрос для получения списка предметов, для которых не назначено ни одного преподавателя.
8. Напишите запрос для получения списка предметов, изучаемых в течение одного семестра.
9. Напишите запрос для получения списка предметов, изучаемых более чем в одном семестре.
10. Напишите запрос для получения списка университетов, в которых не работает ни один преподаватель.
11. Напишите запрос для получения списка университетов, в которых не учится ни один студент.
12. Напишите запрос для получения списка предметов, по которым на экзаменах не получено ни одной неудовлетворительной оценки.

13. Напишите запрос для получения списка предметов, по которым на экзаменах получена хотя бы одна неудовлетворительная оценка.
14. Напишите запрос для, получения списка студентов, не получивших на экзаменах ни одной неудовлетворительной оценки.
15. Напишите запрос для получения списка студентов, получивших на экзаменах хотя бы одну неудовлетворительную оценку.
16. Напишите запрос, выполняющий вывод имен и фамилий студентов, получивших хотя бы одну отличную оценку.
17. Напишите запрос, выполняющий вывод имен и фамилий студентов, не получивших ни одной отличной оценки.
18. Напишите запрос, выполняющий вывод количества студентов, не имеющих ни одной оценки.
19. Напишите запрос, выполняющий вывод количества студентов, имеющих только отличные оценки.
20. Напишите запрос, выполняющий вывод количества студентов, имеющих хотя бы одну неудовлетворительную оценку и проживающих в городе, не совпадающем с городом их университета.

Операторы сравнения с множеством значений IN, ANY, ALL

Операторы сравнения с множеством значений имеют следующий смысл.

IN *Равно* хотя бы одному из значений, полученных во внутреннем запросе.

NOT IN *Не равно НИ* одному из значений, полученных во внутреннем запросе.

=ANY То же, что и IN. Соответствует логическому оператору OR.

$>ANY, \geq ANY$ Больше (либо больше или равно) хотя бы одного из полученных значений. Эквивалентно $>$ (или \geq) для наименьшего из полученных значений.

$<ANY, \leq ANY$ Меньше (либо меньше или равно) хотя бы одного из полученных значений. Эквивалент $<$ (или \leq) для наибольшего из полученных значений.

$=ALL$ Равно каждому из полученных значений.

Эквивалентно логическому оператору AND.

$>ALL, \geq ALL$ Больше (либо больше или равно) каждого из полученных значений. Эквивалент $>$ (или \geq) для наибольшего из полученных значений.

$<ALL, \leq ALL$ Меньше (либо меньше или равно) каждого из полученных значений. Эквивалентно $<$ (или \leq) наименьшего из полученных значений.

Пример 1.

Выбрать сведения о студентах, обучающихся в университете, расположенном в городе их проживания.

```
SELECT *  
FROM STUDENT S  
WHERE CITY=ANY  
(SELECT CITY  
FROM UNIVERSITY U  
WHERE U.UNIV_ID=S.UNIV_ID);
```

Пример 2.

Другой вариант запроса из примера 1

```
SELECT *  
FROM STUDENT S
```

```
WHERE CITY IN  
(SELECT CITY  
FROM UNIVERSITY U  
WHERE U.UNIV_ID=S.UNIV_ID);
```

Пример 3.

Выборка данных об идентификаторах студентов, у которых оценки превосходят величину по крайней мере одной из оценок, полученных 6 октября 2005 г.:

```
SELECT DISTINCT STUDENT_ID  
FROM EXAM_MARKS  
WHERE MAKK > ANY  
(SELECT MARK  
FROM EXAM_MARKS  
WHERE MAKK_DATE='06/10/2005');
```

Оператор ALL, как правило, эффективно используется с неравенствами, а не с равенствами, поскольку значение *равно всем; которое* должно получиться в этом случае в результате выполнения подзапроса, может иметь место, только если все результаты идентичны. Такая ситуация практически не может быть реализована, так как если подзапрос генерирует множество различных значений, то никакое отдельное значение не может быть равно сразу всем значениям в обычном смысле. В SQL выражение **<>ALL** реально означает. не *равно ни одному* из результатов подзапроса.

Пример 4.

Подзапрос, выбирающий данные о названиях всех университетов с рейтингом более высоким, чем рейтинг любого университета в Воронеже

```
SELECT  
FROM UNIVERSITY  
WHERE RATING > ALL  
(SELECT RATING  
FROM UNIVERSITY  
WHERE CITY = 'Воронеж');
```

Пример 5.

В предыдущем примере. вместо ALL можно также использовать ANY (проанализируйте, как в этом случае изменится смысл приведенного запроса?):

```
SELECT *  
FROM UNIVERSITY  
WHERE NOT RATING > ANY  
(SELECT RATING  
FROM UNIVERSITY  
WHERE CITY = 'Воронеж');
```

УПРАЖНЕНИЕ

1. Напишите запрос для получения списка названий предметов изучаемых в нескольких семестрах.
2. Напишите запрос, выполняющий вывод имен и фамилий студентов, имеющих весь набор положительных (тройки, четверки, пятерки) оценок.
3. Напишите запрос, выполняющий выборку значений идентификаторов студентов, имеющих такие же оценки, что: и .студент с идентификатором 12.
4. Напишите запрос, выполняющий вывод количества студентов, не имеющих ни одной оценки.
5. Напишите запрос, выполняющий вывод данных о предметах обучения, которые преподает Колесников.

6. Напишите запрос, выполняющий вывод имен и фамилий преподавателей, проводящих занятия на первом курсе.

7. Напишите запрос, выполняющий вывод данных о преподавателях, ведущих обучение хотя бы по одному из предметов обучения, которые преподает Сорокин.

8. Напишите запрос, выполняющий вывод списка фамилий студентов университета, расположенного в городе, название которого стоит первым в алфавитном списке городов.

9. Напишите запрос, выполняющий вывод списка фамилий студентов, имеющих только отличные оценки и проживающих в городе, не совпадающем с городом их университета.ё

ПРЕДСТАВЛЕНИЯ

1. Создайте представление для получения сведений обо всех студентах, имеющих только отличные оценки.

2. Создайте представление для получения сведений о количестве студентов в каждом городе.

3. Создайте представление для получения следующих сведений по каждому студенту: его идентификатор, фамилия, имя, средний и общий баллы.

4. Создайте представление для получения сведений о количестве экзаменов, которые сдавал каждый студент.