NOT EXISTS

Last modified: Jan 28, 2020

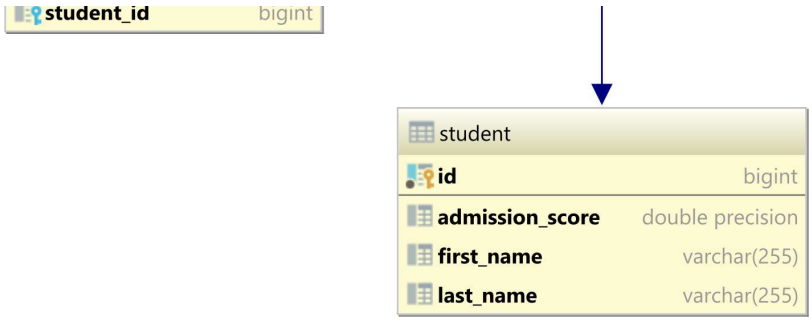
Introduction

In this article, we are going to see how the SQL EXISTS operator works and when you should use it.

Although the EXISTS operator has been available since SQL:86, the very first edition of the SQL Standard, I found that there are still many application developers who don't realize how powerful SQL subquery expressions really are when it comes to filtering a given table based on a condition evaluated on a different table.

Database table model

Let's assume we have the following two tables in our database, that form a one-to-many table relationship. The `student` table is the parent, and the `student_grade` is the child table since it has a



The `student` table contains the following two records:

id	first_name	last_name	admission_scor
1	Alice	Smith	8.95
2	Bob	Johnson	8.75

And, the `student_grade` table stores the grades the students received:

id	class_name	grade	student_id
1	Math	10	1
2	Math	9.5	1
3	Math	9.75	1
4	Science	9.5	1
5	Science	9	1
6	Science	9.25	1
7	Math	8.5	2
8	Math	9.5	2
9	Math	9	2
10	Science	10	2
11	Science	9.4	2

SQL EXISTS

Let's say we want to get all students that have received a 10 grade in Math class.

```
6       student_grade.grade = 10 AND
7       student_grade.class_name = 'Math'
8   ORDER BY
9       student_grade.student_id
```

But, the application is interested in displaying the full name of a student, not just the identifier, so we need info from the `student` table as well.

In order to filter the `student` records that have a 10 grade in Math, we can use the EXISTS SQL operator, like this:

```
1   SELECT
2       id, first_name, last_name
3   FROM
4       student
5   WHERE EXISTS (
6       SELECT 1
7       FROM
8           student_grade
9       WHERE
10          student_grade.student_id = stud
11          student_grade.grade = 10 AND
12          student_grade.class_name = 'Mat
13  )
14  ORDER BY id
```

When running the query above, we can see that only the Alice row is selected:

```
| id | first_name | last_name |
|----|------------|-----------|
| 1  | Alice      | Smith     |
```

The outer query selects the `student` row columns we are interested in returning to the client. However, the WHERE clause is using the EXISTS operator with an associated inner subquery.

The inner subquery is correlated because the `student_id` column of the `student_grade` table is matched against the `id` column of the outer `student` table.

SQL NOT EXISTS

Let's consider we want to select all students that have no grade lower than 9. For this, we can use NOT EXISTS, which negates the logic of the EXISTS operator.

Therefore, the NOT EXISTS operator returns `true` if the underlying subquery returns no record. However, if a single record is matched by the inner subquery, the NOT EXISTS operator will return `false`, and the subquery execution can be stopped.

To match all `student` records that have no associated `student_grade` with a value lower than 9, we can run the following SQL query:

```
SELECT
    id, first_name, last_name
FROM
    student
WHERE NOT EXISTS (
    SELECT 1
    FROM
        student_grade
    WHERE
        student_grade.student_id = stud
        student_grade.grade < 9
)
ORDER BY id
```

Cool, right?

Conclusion

The advantage of using the SQL EXISTS and NOT EXISTS operators is that the inner subquery execution can be stopped as long as a matching record is found.