Natural Joins

by Sophia Tutorial

WHAT'S COVERED

This tutorial explores using a natural join between tables in two parts:

1. Getting Started
2. Problems with Natural Joins

1. Getting Started

The ability to do a natural join is dependent on whether two tables have a column with the same name. A natural join links tables together by choosing the rows with common values using their common attribute. Typically, a natural join is a result of three stages:

- The first stage creates the product of the two tables together.
- The next stage takes the output from the prior stage and only displays the rows in which the common attributes are equal to one another.
- In the last stage, a PROJECT is performed on the results to have a single copy of each attribute, which removes the duplicate column.

The final result based on the natural join provides a result set that only has the matches between the two tables. It does not include any unmatched pairs from either table.
The syntax of a natural join looks like the following:

```
SELECT <column_list>
FROM <table1>
NATURAL [INNER, LEFT, RIGHT] JOIN <table2>
```

Note that with a natural join, the default type is an inner join, but it can also perform a left or right join. Instead of a column list, we can also use the asterisk (*). This will include all columns from both tables that have the same name.

Let's build a simple set of tables of product and categories:

```
CREATE TABLE category ( category_id serial PRIMARY KEY, category_name VARCHAR (100) NOT NULL );

CREATE TABLE product ( product_id serial PRIMARY KEY, product_name VARCHAR (100) NOT NULL, category_id INT NOT NULL, FOREIGN KEY (category_id) REFERENCES category (cate

INSERT INTO category (category_name)
VALUES ('Game'), ('Movie'), ('CD');

INSERT INTO product (product_name, category_id)
VALUES ('Call of Duty', 1), ('Final Fantasy', 1), ('Wizard of Oz', 2), ('Jaws', 2), ('Great Hits', 3), ('Journey', 3);
```

To join them using a natural join, we can set it as:

```
SELECT *
FROM category
NATURAL JOIN product;
```

Query Results

Row count: 6

category_id	category_name	product_id	product_name
1	Game	1	Call of Duty
1	Game	2	Final Fantasy
2	Movie	3	Wizard of Oz
2	Movie	4	Jaws
3	CD	5	Great Hits
3	CD	6	Journey

Note that we could also swap the two tables in the query:

```
SELECT *
FROM product
NATURAL JOIN category;
```

© 2022 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

Query Results

Row count: 6

category_id	product_id	product_name	category_name
1	1	Call of Duty	Game
1	2	Final Fantasy	Game
2	3	Wizard of Oz	Movie
2	4	Jaws	Movie
3	5	Great Hits	CD
3	6	Journey	CD

You should notice that in both cases, the result set will display the common column first, which in this example is the category_id. Then it will display all of the columns from the first table (other than the common column) and then all of the columns from the second table (other than the common column).

2. Problems with Natural Joins

There are potential issues to be aware of with a natural join, and they can sometimes create unexpected results. Let's recreate these tables, but instead of having product_name and category_name, let's just call these columns "name" in both tables:

```
CREATE TABLE category ( category_id serial PRIMARY KEY, name VARCHAR (100) NOT NULL );

CREATE TABLE product ( product_id serial PRIMARY KEY, name VARCHAR (100) NOT NULL, category_id INT NOT NULL, FOREIGN KEY (category_id) REFERENCES category (category_id) )

INSERT INTO category (name)
VALUES ('Game'), ('Movie'), ('CD');

INSERT INTO product (name, category_id)
VALUES ('Call of Duty', 1), ('Final Fantasy', 1), ('Wizard of Oz', 2), ('Jaws', 2), ('Great Hits', 3), ('Journey', 3);
```

If we now tried to run the natural join:

```
SELECT *
FROM product
NATURAL JOIN category;
```

We get the following result:

Query Results

Query ran successfully. 0 rows to display.

So although we have the category_id in both tables, we also have names in both tables. This common column has different meanings in each table, and since the values in the two do not match, no rows are returned.

Video Transcription

[MUSIC PLAYING] A natural join is a type of [INAUDIBLE] join that allows you to link data together between two tables on the columns that have the exact same name. So in this case here, we have a category table and a product table. Both of them have this category_id. That's inserted into them.

So as part of that, when we're actually doing a natural join, the query looks like the following. SELECT, then the column names from the first table, then the keywords NATURAL JOIN, and then the second table, product.

In this case here, it tries to link all those different items together based on the category_id and first lists the category_id being the adjoining item, then the rows-- in this case here, columns from the category table, then the link between that with the product table, and then the product name.

If there had been multiple different columns within the two tables that match up together, it would also try to join on those different columns. Typically, the natural join is not recommended if you have multiple different columns associated with it. Not a big deal, though, if there is just a single column based on that foreign key.

[MUSIC PLAYING]

TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.

SUMMARY

© 2022 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

The natural join allows you to combine data between two or more tables that have common columns.

Source: Authored by Vincent Tran

© 2022 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.