Foreign Key Errors

by Sophia

WHAT'S COVERED

This lesson explores the foreign key errors that can occur with SQL statements, in two parts. Specifically, this lesson will cover:

1. Foreign Key Errors When Inserting Records

As we have seen in the prior lesson, when we have a set of tables with foreign keys, we must insert, update, and delete in a specific order. Data changes can result in an error if not done correctly.

When we insert data into a table, the most common issue with the foreign keys is the order in which we do it. However, another issue can arise if we're inserting data into a table that does not have the referenced value from the foreign key in the parent table.

Let's revisit our data set with the representatives and departments again:

```
CREATE TABLE representative ( representative_id INT PRIMARY KEY, first_name VARCHAR (30) NOT NULL, last_name VARCHAR (30) NOT NULL );
CREATE TABLE department (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(100) NOT NULL,
    manager_id INT,
    CONSTRAINT fk_manager FOREIGN KEY (manager_id) REFERENCES representative(representative_id)
);
);

INSERT INTO representative (representative_id, first_name, last_name)
VALUES (1, 'Bob','Evans'), (2, 'Tango','Rushmore'), (3, 'Danika','Arkane'), (4, 'Mac','Anderson');
INSERT INTO department (department_id, department_name,manager_id)
VALUES (1, 'Sales', 1), (2, 'Marketing', 3), (3, 'IT', 4), (4, 'Finance', null), (5, 'Support', null);
```

Suppose we then tried to insert a value for the manager_id that did not exist into the department table:

```
INSERT INTO department (department_id, department_name,manager_id)
VALUES (6, 'Test', 100);
```

We would get this error:

Query failed because of: error: insert or update on table "department" violates foreign key constraint "fk_manager"

This is because we attempted to pass 100 in the manager_id. But if we look at the representative table, we only have values 1–4 available. Likewise, if we attempted to update a row in the department table's manager_id to a value that didn't exist, we would get the same error:

```
UPDATE department
SET manager_id = 100
WHERE department_id = 5;
```

We indeed do get the same error message:

Query failed because of: error: insert or update on table "department" violates foreign key constraint "fk_manager"

Again, this is because we're trying to update the department table from a valid value (NULL, in this case) to 100, which does not exist in the representative table. Both the insert and update statements would be invalid in this case.

2. Foreign Key Errors When Deleting Records

Errors can also occur if we delete from a parent table when there are records associated with it in the child table. For example, the sales department has the manager_id set to 1 in the department table, referencing Bob Evans in the representative table. Suppose we tried to delete that record from the representative table:

```
DELETE FROM representative
WHERE representative_id = 1;
```

We will get this error:

Query failed because of: error: update or delete on table "representative" violates foreign key constraint "fk_manager" on table "department"

This is because we're trying to delete from the parent table representative when the department table still has that value as a foreign key. Consider if we tried to update the representative_id in the representative table to another value:

```
UPDATE representative
SET representative_id = 100
WHERE representative_id = 1;
```

We would get the same error. If we needed to delete that particular representative, we would either have to delete the records that reference the representative_id in the department table or update the representative_id to another value.

TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then, enter 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

In this lesson, you learned about some data insertion and deletion errors involving foreign keys. The **foreign key error occurs when a value is inserted** into a column that references a primary key in another table, but the corresponding value does not exist in the referenced table. This error ensures referential integrity by preventing inconsistent or disconnected data from being introduced. If you insert an order record without the associated customer ID in the customers' table, a foreign key error would occur, safeguarding database coherence.

Foreign key errors when deleting records occur when a record from a referenced table is deleted while dependent records still exist in the referencing table. As a result of this error, data loss is prevented, and relationships are maintained. If you try to delete a department from an organizational structure but there are employees assigned to that department, a foreign key error would occur, prompting you to handle the dependencies first. The foreign key error acts as a safeguard to maintain accurate and reliable data relationships in a database, regardless of whether the data is inserted or deleted.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND FAITHE WEMPEN (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR **TERMS OF USE**.