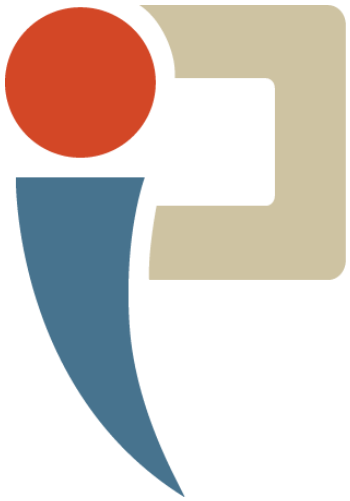
POLUDO INSTITUTE OF TECHNOLOGY & MEDIA



SQL TRAINING

Module 4

Inserting, deleting and modifying data on tables

Jesse Silva – jessetsilva@gmail.com

2018

Module 4: Inserting, deleting and modifying Data in Tables

The students will be able to modify the data in tables by using the INSERT, DELETE, and UPDATE statements. In addition, students will examine how transactions work in a database, the importance of transaction isolation levels, and how to manage transactions.

Lessons

- Inserting Data into Tables
    - The INSERT INTO statement
    - Inserting record in tables with auto increment
    - Inserting records in tables with foreign keys
- Updating Data from Tables
- Deleting Data from Tables
- Truncating a table

After completing this module, students will be able to:

- Insert data into tables.
- Delete data from tables.
- Update data in tables.
- Describe transactions.

INSERTING DATA INTO TABLES

- ## THE INSERT INTO STATEMENT

To insert new record in a table we must use the INSERT INTO statement. There are two possible syntax to use when inserting new records. The first one does not specify the columns names where the data will be inserted, but for it we need to give the data in the exact same order as the columns you want to use.

```
INSERT INTO table_name
VALUES (value1, value2, value3...);
```

You also can specify the column names and the values to be inserted, in the same order:

```
INSERT INTO table_name (column1, column2, column3...)
VALUES (value1, value2, value3...);
```

For example, let's insert a new record in our table address using both forms. In the first one we can use:

```
INSERT INTO `address` (`addressID`,`address`, `postalCode`, `city`, `state`, `country`)
VALUES (201,'789 Tuson', 'V4C 4C5', 'Vancouver', 'BC', 'Canada');
```

When we are inserting a new record in a table that contains the auto increment constraint configured for primary key, we don't need to insert the value for the column, so we need to exclude it from our query and leave it to the database manager.

```
INSERT INTO `address` (`address`, `postalCode`, `city`, `state`, `country`)
VALUES ('789 Tuson', 'V4C 4C5', 'Vancouver', 'BC', 'Canada');
```

If we want to insert values in specific columns, we need to use the second syntax option and describe the name of every field we are inserting. If we try

to insert a new record in the table without filling a not null column, we will see an error. As example, let's try to insert an address without the postal code and see the results.

```
INSERT INTO address (address ,  city, state, country)
VALUES ('789 Tuson', 'Vancouver', 'BC', 'Canada');
```

Result:

Error Code: 1364. Field 'postalCode' doesn't have a default value

- ## INSERTING RECORDS IN TABLES WITH FOREIGN KEYS

When inserting new records into a table that contains a foreign key, we need to make sure that the value being inserted in that column really exists in the parent table. For example, if you run the command below:

```
INSERT INTO `student` (`firstName`, `lastName`, `gender`, `email`, `birthDate`, `contactNumber`, `addressID`)
VALUES ('Jesse', 'Teixeira', 'male', 'jessetsilva@gmail.com', '1983-06-08', '6043546150', '530');
```

We will receive an error telling that the address 530 does not exist:

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (student`, CONSTRAINT `addressFK` FOREIGN KEY (`addressID`) REFERENCES `address` (`addressID`) ON DELETE NO ACTION ON UPDATE NO ACTION)

UPDATING DATA FROM TABLES

- ## THE UPDATE STATEMENT

To update an existing value in a table we need to use the update statement. The syntax is:

```
UPDATE table_name
SET column1=value1, column2 = value2 , ..., columnN = valueN, ...
WHERE some_column=some_value;
```

One thing that must be always kept in mind is the use of the WHERE clause. Although it is not mandatory, not using it can be a disaster since all the records in the table will be updated. You can update as many columns as you want in a single update statement based on how many conditions you want.

The update statement below will change the gender of all the students from "male" if their gender is defined only as "m"

```
update student set gender ='male' where gender ='m'
```

Do the same for the students that have the genders defined as "f". Change it to "female"

When we update a primary key that is referenced in another table as foreign key and configured as UPDATE ON CASCADE, the respective values will be updated in those tables as well.

DELETING DATA FROM TABLES

To delete a table record, we need to use the DELETE statement. The syntax is:

```
DELETE FROM table_name
WHERE some_column=some_value, … ,
      some_other_column = some_other_value;
```

Similarly with the UPDATE statement, the WHERE clause is very important, because if we do not use it all the records in the table will be deleted. When you delete all the table's record, it does not mean that the table itself will be dropped. The table structure, attributes, and indexes will be intact and even the auto increment values will continue to be incremented based on the last valid value and not start from 1 again.

As example, let's remove from the student table the student that has the firstName '"Jesse" and the lastName "Teixeira". The command will be:

```
delete from student where firstName='Jesse' and lastName ='Teixeira'
```

Now let's try to remove the course "Software Quality Assurance, Quality Control and Testing" that has the code '1':

```
delete from course where courseID = 1;
```

If we run this code, we will receive an error message because we have one table that uses the courseID as foreign key (table Course_Class) and it is filled with some references to the course. When we created it, we decided that it would not accept any record to be removed when referenced (no action).

When we delete a record having a primary key that is referenced in another table as foreign key and configured as DELETE ON CASCADE, the respective values will be deleted in those tables as well.

TRUNCATING TABLES

According to Microsoft: "The TRUNCATE TABLE statement is a fast, efficient method of deleting all rows in a table. TRUNCATE TABLE is similar to the DELETE statement without a WHERE clause. However, TRUNCATE TABLE is faster and uses fewer system and transaction log resources".

Truncating a table is different from dropping it. When we truncate our table all the data is removed but all the table's characteristics and constraints remain. The syntax is

```
truncate table table_name
```

For example, if we want to remove all our instructors' information from the database, but keep the table, the command would be:

```
truncate table instructor;
```

In the instructor table example, we have an auto increment primary key. When you truncate the table, automatically the newly inserted record will receive the first auto increment value, in this, the value will start on ONE again.

PRACTICING

- Update all the students so the emails and genders will be stored in lowercase.

- One of the instructors can't teach at the school for personal reasons. It's your job to remove this instructor from our database and make sure that all his classes will be taken by another Instructor (we are not hiring, so you need to use an existing one).

- Because of new regulations, we need to change our classes' schedules. All the classes that start or finish in November must be rescheduled to start and finish in October.

- Management asked us to always keep the Students and instructor records, even if they are not part of our school anymore so they can have the historical data if needed. It means that we can no longer just delete instructors and students.

    - We need a way to keep the record, but show if a student or instructor is active or inactive in our school
    - Discuss with your instructor a strategy to accomplish this request
        - This exercise will use knowledge we acquired in previous classes

REFERENCES

- **Books**
  - Fundamentals of database systems – 6th edition
  - Head First SQL - Lynn Beighley
- **Internet**
  - http://www.w3schools.com/sql/sql_delete.asp
  - http://www.webopedia.com/TERM/U/uppercase.html
  - http://www.sqlcourse.com/delete.html
  - https://technet.microsoft.com/en-us/library/ms188249(v=sql.105).aspx