SQL Level 3

Bonus for PostgreSQL

noble desktop

Inserting Data

Adding Data to Tables

INSERT INTO Statement

INSERT INTO is used to add new records into a database table.

There are basically two ways to INSERT data into a table:

- One row at a time
- Multiple rows at a time

Record Insertion Methods

Inserting *all column* values:

- The column names do not need be specified in the INSERT clause. However, the order of the values specified in the VALUES clause must match the order in which the columns are organized in the table.

Inserting *specific column* values:

- The required columns names must be specified immediately after the table name in the **INSERT** clause. In addition, the order in which the values are written in the **VALUES** clause must match the order of column names specified in the **INSERT** clause.

INSERT: All Columns

```
INSERT INTO products
VALUES( DEFAULT, 'Comfy Shoes', 79.95, NOW() );
```

- DEFAULT is for an auto-incrementing primary key
- In Microsoft SQL Server replace **`NOW()`** with **`GETDATE()`**

INSERT: Specific Columns

```
INSERT INTO products (title, price)
VALUES( 'Comfy Shoes', 79.95 );
```

INSERT: Multiple Rows

```
INSERT INTO products
VALUES
   ( DEFAULT, 'Comfy Shoes', 79.95, NOW() ),
   ( DEFAULT, 'Sweatshirt', 25.95, NOW() )
;
```

Exercise

Open the file “B1 Inserting Data into Tables.sql”
(in SQL Level 3 > Bonus for PostgreSQL folder)

Create Tables/Columns/Keys & Add/Import/Export Data

Working with Tables & Data

Database Fundamentals

- Typically, only a Database Administrator (DBA) has permission to change the database.
- You might not need to do this, but it’s good to be familiar with the basics.
- We won’t go in depth in database design (that’s beyond the scope of this class), but you’ll become familiar with the most important concepts.

Using Dbeaver’s Interface

You can use Dbeaver’s interface to create things without having to code SQL. Here’s some of the things you can do:

- Create & Delete Tables
- Create Columns (with appropriate data types)
- Define Primary & Foreign Keys
- Add, Update, Import, and Export Data

And if you want to see the SQL code, DBeaver will show it to you!

Exercise

Exercise B1 in the eBook

Transactions

Database Transactions

Transactions

- A transaction groups one or more operations into a single execution.
- A transaction has only 2 results: success or failure.
- If any task in the group fails, the entire transaction fails.
- Transactions let you preview changes before committing to them (you can do a rollback if needed).
- "PostgreSQL actually treats every SQL statement as being executed within a transaction. If you do not issue a BEGIN command, then each individual statement has an implicit BEGIN and (if successful) COMMIT wrapped around it." — postgresql.org

Exercise

Open the file “B2 Transactions.sql"

(in SQL Level 3 > Bonus for PostgreSQL folder)

Updating Data

Modifying Data in Tables

UPDATE Statement

The UPDATE statement modifies data in a column. It consists of two parts:

- An UPDATE clause with the name of the table whose column values need to be updated.
- And a SET clause that specifies the column name and the new value assigned to the column.

Targets rows in the WHERE clause, and if WHERE is not stated the entire table is updated.

USE WITH CAUTION!

UPDATE Syntax

ONE COLUMN:

```
UPDATE table_name
SET column_name=value
WHERE condition;
```

MULTIPLE COLUMNS:

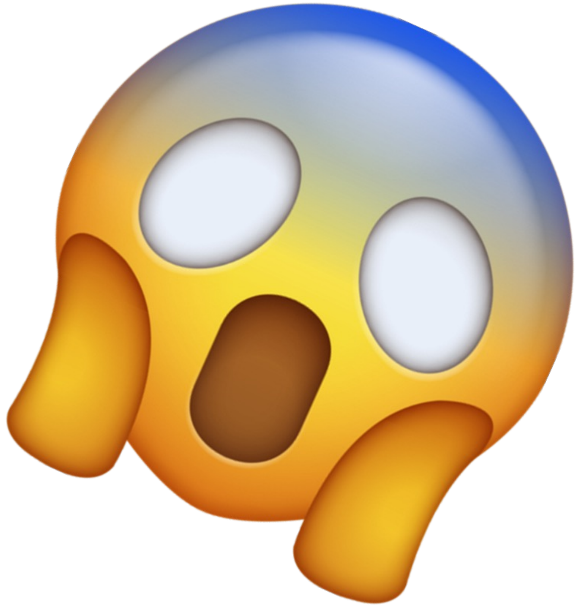
```
UPDATE table_name
SET col1=value1, col2=value2
WHERE condition;
```

UPDATE Example

```
UPDATE products
SET tags = 'green'
WHERE tags = 'emerald';
```

UPDATE Bad Example

```
UPDATE products
SET tags = 'green';
```



Oh No!
Without a WHERE clause, every product is changed!

Exercise

Open the file B3 Modifying Tables.sql

(in SQL Level 3 > Bonus for PostgreSQL folder)