Introduction to Microsoft SQL Server

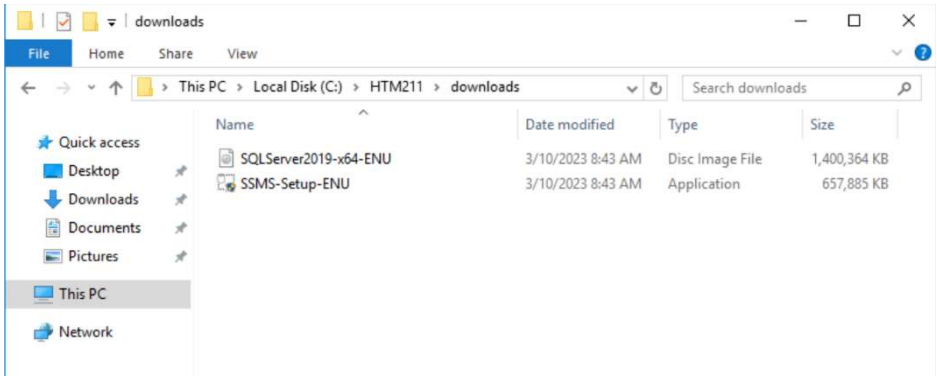
Instructions

1. Labs are presented in step-by-step format.
2. Please complete each step.
3. Text that you will type at the prompt or in the GUI is colored green.

Lab 1 - Basic SQL Server and SSMS Studio Installation

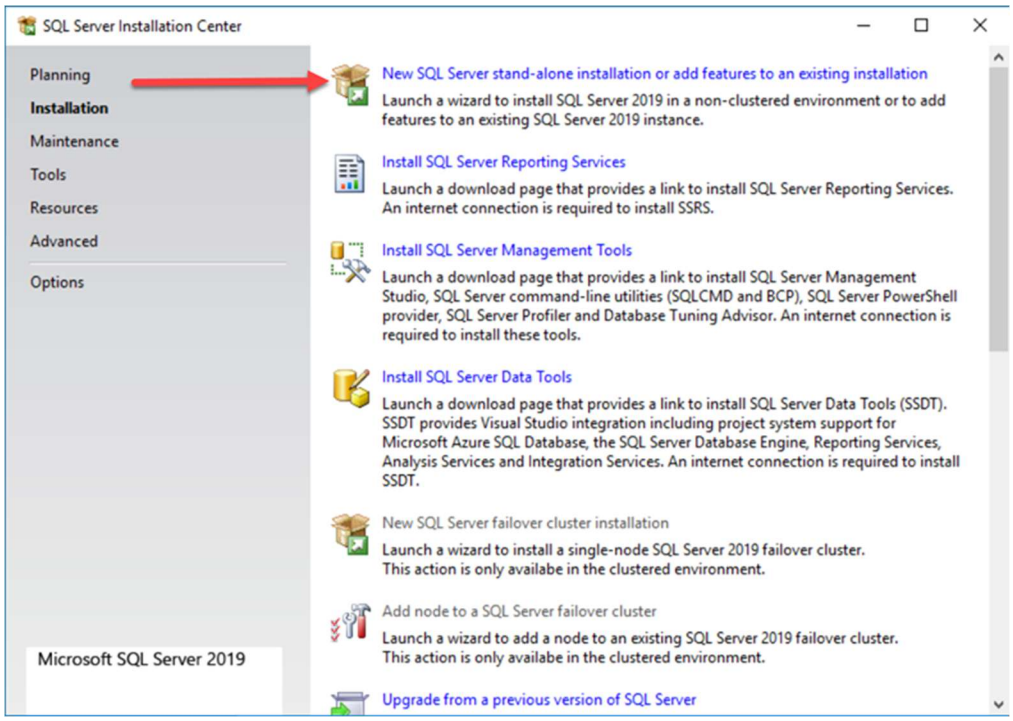
Install Microsoft SQL server

1. Obtain a copy of Microsoft SQL Server. There is a copy in your VM in c:\htm211\downloads\SQLServer2019-x64ENU.exe.



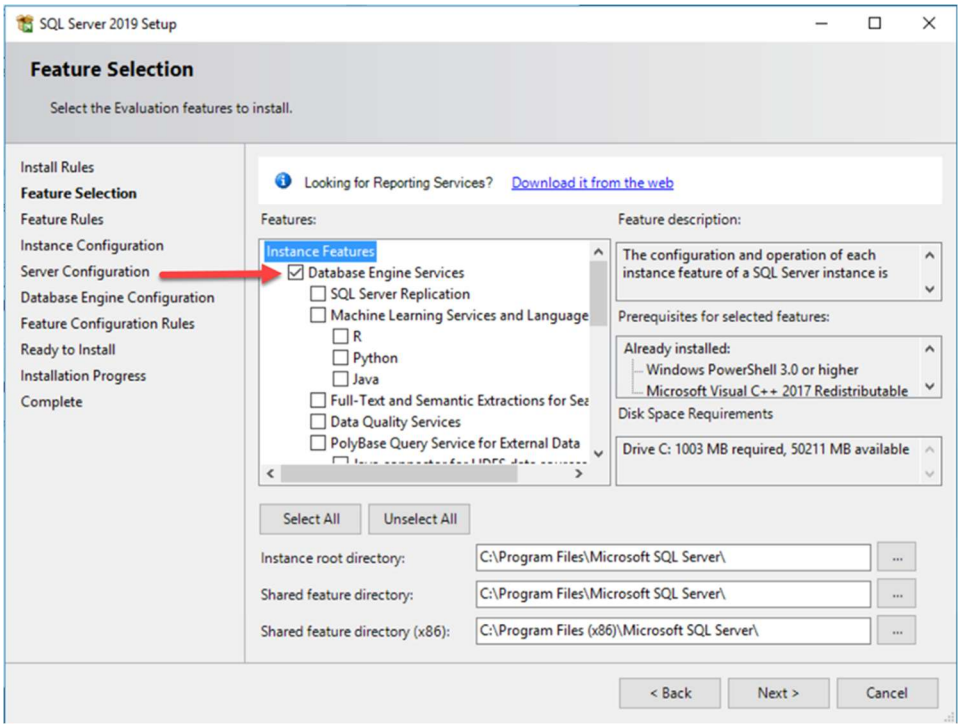
Open your File manager and type `c:` in the location bar. Then navigate to the htm211\downloads folder. Double click the file: SQLServer2019-x64-ENU.exe to open the ISO file. (To download from the web: https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2019?filetype=EXE)

2. Double click on `setup` to start the installation. It may take a few seconds for the installation screen to appear. (This is a huge program, so patience is a virtue when installing.)
3. Click on `Installation` then `New SQL Server stand-alone installation or add features to an existing installation`, then `Run`.
4. Follow the prompts to install a standalone SQL Server Instance.

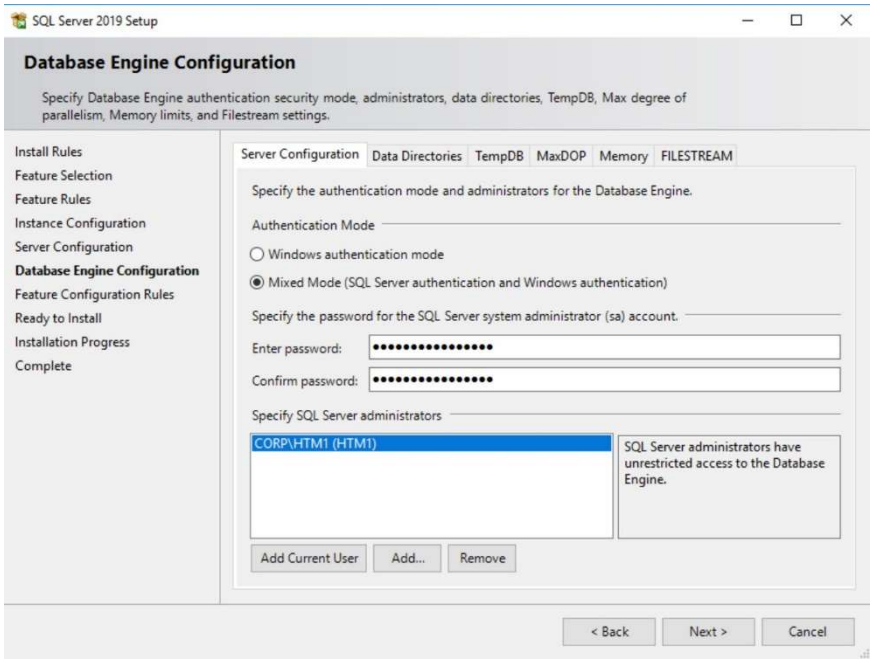


5. Choose Evaluation as the free edition to install. Click Next.
6. Accept the license terms. Click Next.
7. Do not check for updates (box remains unchecked). Click Next.
8. The installation will start, and it will take some time to install setup files.
9. At the Install Rules dialog just ignore any warnings and click Next.

10. At the Feature Selection dialog choose Database Engine Services and click Next.



11. This will be the default instance. Leave the default name at MSSQLSERVER. Click Next.
12. Nothing to change at the Server Configuration dialog. Click Next to continue.
13. Choose Mixed Mode from the Database Engine Configuration dialog. Use `Thel1ioninwint3r` as the password. Also click Add Current User as a SQL Server

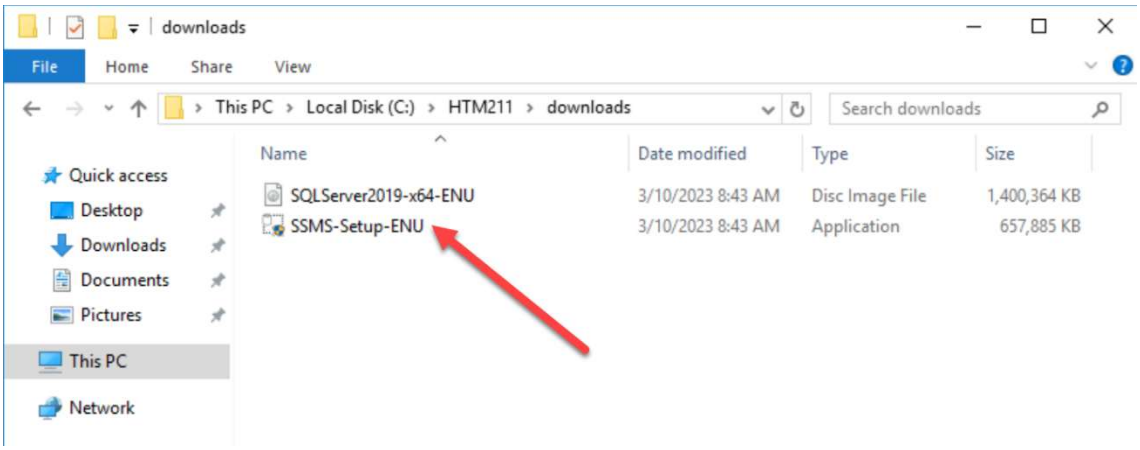


administrator.

14. Choose Install. The installation process will take a few minutes.
15. Choose Close and the installation is complete. Also close the installation dialog.

To install SQL Server Management Studio (SSMS)

1. Install Microsoft SQL Server Management Studio by double clicking SSMS-Setup-ENU.exe from the File Manager.



2. It will take a minute or two for the installation to start, then click Yes in the User Account Control dialog box.
3. Click Install.
4. Click Close to complete the installation.

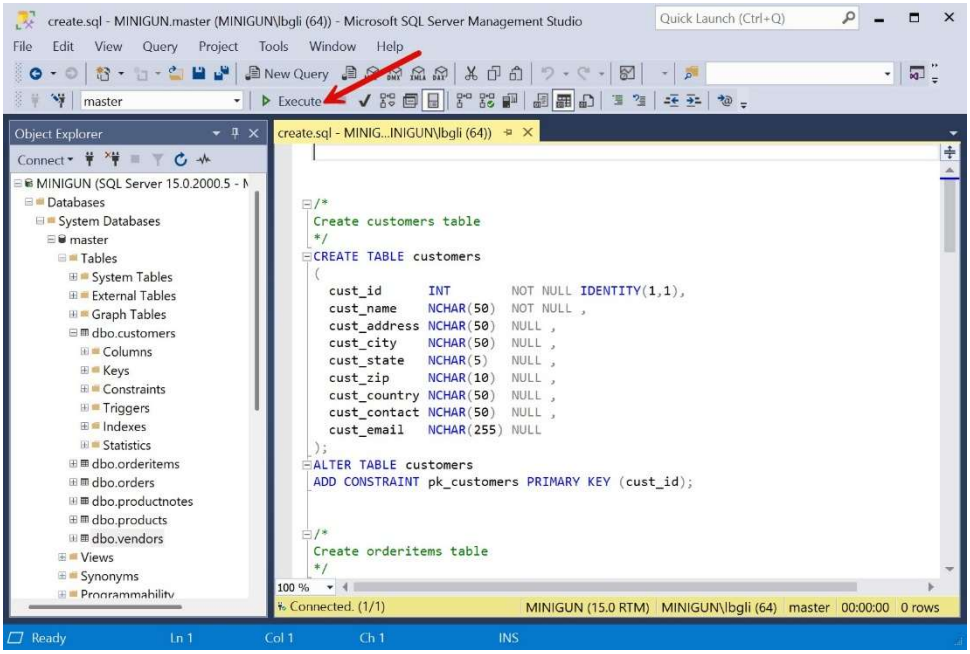
Lab 1.1 – Download SQL Server Database Setup Scripts

1. Open the Edge or Firefox browser in your Amazon Workspace and go to https://htm.wpdevdaemon.com.
2. Click on the link for SQL Setup Scripts Zip File.
3. Download the file.
4. Open the file folder in the downloads dialog.
5. Unzip the file you downloaded. It is named `sql-scripts.zip`.
6. The file will be extracted to the `downloads\sql-scripts\scripts` folder.

Lab 1.2 - Install and Populate the Sample Database

1. Navigate to downloads\sql-scripts\scripts folder.
2. Double click the `create.sql` file.
3. A dialog box may pop up saying “How do you want to open this file?” Choose SSMS18 and click OK.
4. Click Connect.
5. SSMS will open with the script ready to run in the window.

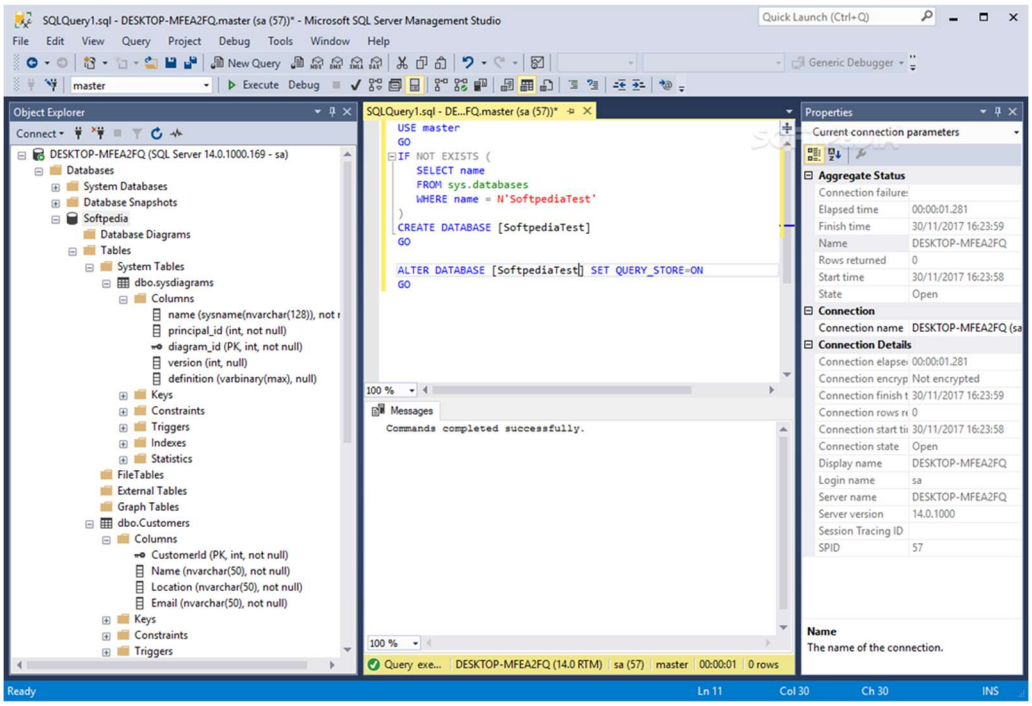
6. Click Execute in the Toolbar. This will create the sample database.



7. From the `downloads\sql-scripts\scripts` folder double click the `populate.sql` script.
8. This will open a tab for the script.
9. Click the Execute item in the Toolbar. This will populate the databases with sample data.
10. Close the Tabs for the create.sql and populate.sql scripts.
11. Leave SSMS open as we will be using it for subsequent labs.

Congratulations! You are now ready to do the rest of the labs.

Lab 2 – Retrieving, Sorting, and Filtering Data



Retrieve a Column from a Table

1. From the Toolbar, Select New Query.
2. In the tab that opens type: `SELECT prod_name FROM products;` (be sure to include the semicolon)
3. Click Execute. The product names will appear as a result.

Retrieve Multiple Columns from a Table

1. Click New Query.
2. In the tab that opens type: `SELECT prod_id, prod_name, prod_price FROM products;`
3. Notice that multiple colums are displayed.

Retrieve all Columns from a Table

1. Click New Query.
2. In the tab that opens type: `SELECT * FROM products;`
3. Click Execute.
4. Notice that multiple colums are displayed.

Notice each query opens a new tab. We don't have to open new tabs, just erase the command in the current tab.

Retrieve all Rows from a Table

1. Click New Query.
2. In the tab that opens type: `SELECT vend_id FROM products;`
3. Click Execute.
4. Notice that there are duplicate vendor numbers.

Retrieve Distinct Rows from a Table

1. Click New Query.
2. In the tab that opens type: `SELECT DISTINCT vend_id FROM products;`
3. Click Execute.

Retrieve Limited Results from a Table

1. Click New Query.
2. In the tab that opens type: `SELECT TOP(5) prod_name FROM products;`
3. Click Execute.

Retrieve Limited Results from a Table

1. Click New Query.
2. In the tab that opens type: `SELECT TOP(25) PERCENT prod_name FROM products;`
3. Click Execute.

Retrieving and Sorting Results

1. Click New Query.
2. In the tab that opens type: `SELECT prod_name FROM products ORDER BY prod_name;`
3. Click Execute.

Sorting by Multiple Columns

1. Click New Query.
2. In the tab that opens type: `SELECT prod_id, prod_price, prod_name FROM products ORDER BY prod_price, prod_name;`
3. Click Execute.

Specifying Sort Direction

1. Click New Query.
2. In the tab that opens type: `SELECT prod_id, prod_price, prod_name FROM products ORDER BY prod_price DESC;`
3. Click Execute.

Specify Sort Direction and Display Multiple Columns

1. Click New Query.
2. In the tab that opens type: `SELECT prod_id, prod_price, prod_name FROM products ORDER BY prod_price DESC, prod_name;`

3. Click Execute.

Filtering Data using Where

1. Click New Query.
2. In the tab that opens type: `SELECT prod_name, prod_price FROM products WHERE prod_price = 2.50;`
3. Click Execute.

Checking Against a Single Value

1. Click New Query.
2. In the tab that opens type: `SELECT prod_name, prod_price FROM products WHERE prod_name ='fuses';`
3. Click Execute.

Checking Against a Range of Values

1. Click New Query.
2. In the tab that opens type: SELECT prod_name, prod_price FROM products WHERE prod_price BETWEEN 5 AND 10;
3. Click Execute.

Advanced Filtering Using AND

1. Click New Query.
2. In the tab that opens type: `SELECT prod_id, prod_price, prod_name FROM products WHERE vend_id = 1003 AND prod_price <= 10;`
3. Click Execute.

Advance Filtering Using OR

1. Click New Query.
2. In the tab that opens type: `SELECT prod_name, prod_price FROM products WHERE vend_id = 1002 OR vend_id = 1003;`
3. Click Execute.

Advance Filtering Using Parentheses

1. Click New Query.
2. In the tab that opens type: `SELECT prod_name, prod_price FROM products WHERE (vend_id = 1002 OR vend_id = 1003) AND prod_price >= 10;`
3. Click Execute.

Advanced Filtering Using IN

1. Click New Query.
2. In the tab that opens type: `SELECT prod_name, prod_price FROM products WHERE vend_id IN (1002,1003) ORDER BY prod_name;`

3. Click Execute.

Advanced Filtering Using NOT

1. Click New Query.
2. In the tab that opens type: `SELECT prod_name, prod_price FROM products WHERE vend_id NOT IN (1002,1003) ORDER BY prod_name;`
3. Click Execute.

Wildcard Filtering Using %

1. Click New Query.
2. In the tab that opens type: `SELECT prod_id, prod_name FROM products WHERE prod_name LIKE '%anvil%';`
3. Click Execute.

Wildcard Filtering Using _

1. Click New Query.
2. In the tab that opens type: `SELECT prod_id, prod_name FROM products WHERE prod_name LIKE '_ ton anvil%';`
3. Click Execute.

Wildcard Filtering Using []

1. Click New Query.
2. In the tab that opens type: `SELECT cust_contact FROM customers WHERE cust_contact LIKE '[EJ]%' ORDER BY cust_contact;`
3. Click Execute.

Bonus:

Write queries that check for less than, greater than, not equal to values. Write queries that check for less than or equal to, greater than or equal to values or text.

Write queries that use each of the wildcards.

Congratulations! You have finished this lab.

Lab 3 – Grouping and Summarizing Data

Aggregating Data Using Avg()

1. Click New Query.
2. In the tab that opens type: `SELECT Avg(prod_price) AS avg_price FROM products;`
3. Click Execute.

Aggregating Data Using Count()

1. Click New Query.
2. In the tab that opens type: `SELECT Count(cust_email) AS num_cust FROM customers;`
3. Click Execute.

Aggregating Data Using Max()

1. Click New Query.
2. In the tab that opens type: `SELECT Max(prod_price) AS max_price FROM products;`
3. Click Execute.

Aggregating Data Using Min()

1. Click New Query.
2. In the tab that opens type: `SELECT Min(prod_price) AS min_price FROM products;`
3. Click Execute.

Aggregating Data Using Sum()

1. Click New Query.
2. In the tab that opens type: `SELECT Sum(quantity) AS items_ordered FROM orderitems WHERE order_num = 20005;`
3. Click Execute.

Aggregating Data on Distinct Values

1. Click New Query.
2. In the tab that opens type: `SELECT Avg(DISTINCT prod_price) AS avg_price FROM products WHERE vend_id = 1003;`
3. Click Execute.

Combining Aggregate Functions

1. Click New Query.
2. In the tab that opens type: `SELECT Count(*) AS num_items, Min(prod_price) AS price_min, Max(prod_price) AS price_max, Avg(prod_price) AS price_avg FROM products;`

3. Click Execute.

Grouping Data

1. Click New Query.
2. In the tab that opens type: `SELECT vend_id, Count(*) AS num_prods FROM products GROUP BY vend_id;`
3. Click Execute.

Grouping and Sorting Data

1. Click New Query.
2. In the tab that opens type: `SELECT order_num, Sum(quantity*item_price) AS ordertotal FROM orderitems GROUP BY order_num HAVING Sum(quantity*item_price) >= 50 ORDER BY ordertotal;`
3. Click Execute.

Congratulations! You have finished this lab.

Lab 4 – Working with Subqueries

Combining Queries

1. Click New Query.
2. In the tab that opens type:
```
SELECT cust_name, cust_contact
FROM customers WHERE cust_id IN (SELECT cust_id FROM orders WHERE
order_num IN (SELECT order_num
FROM orderitems WHERE prod_id = 'TNT2'));
```
3. Click Execute.

Calculated Fields in Subqueries

1. Click New Query.
2. In the tab that opens type:
```
SELECT cust_name, cust_state,  (SELECT Count(*)
FROM orders WHERE orders.cust_id = customers.cust_id) AS orders FROM
customers ORDER BY cust_name;
```
3. Click Execute.

Correlated Subqueries

1. Click New Query.
2. In the tab that opens type:
```
SELECT cust_id, cust_name FROM customers WHERE
cust_id IN (SELECT cust_id FROM orders WHERE DateDiff(month,
order_date, '2016-09-01') = 0 AND customers.cust_id =
orders.cust_id);
```
3. Click Execute.

Congratulations! You have finished this lab.

Lab 5 – Joining Tables

Joining Tables

1. Click New Query.
2. In the tab that opens type: `SELECT vend_name, prod_name, prod_price FROM vendors, products WHERE vendors.vend_id = products.vend_id ORDER BY vend_name, prod_name;`
3. Click Execute.

Creating an Inner Join

1. Click New Query.
2. In the tab that opens type: `SELECT customers.cust_id, customers.cust_name FROM customers INNER JOIN orders ON customers.cust_id = orders.cust_id WHERE DateDiff(month, order_date, '2016-09-01') = 0 AND customers.cust_id = orders.cust_id;`
3. Click Execute.

Joining Multiple Tables

1. Click New Query.
2. In the tab that opens type: `SELECT prod_name, vend_name, prod_price, quantity FROM orderitems, products, vendors WHERE products.vend_id = vendors.vend_id AND orderitems.prod_id = products.prod_id AND order_num = 20005;`
3. Click Execute.

Congratulations! You have finished this lab.

Lab 6 - Updating and Deleting Data

Inserting a Row

1. Click New Query.
2. In the tab that opens type: `INSERT INTO customers(cust_name, cust_contact, cust_email, cust_address, cust_city, cust_state, cust_zip, cust_country) VALUES('Pep E. LaPew', NULL, NULL,'100 Main Street','Los Angeles','CA','90046','USA');`
3. Click Execute.

Updating Data

1. Click New Query.
2. In the tab that opens type: `UPDATE customers SET cust_email = 'elmer@fudd.com' WHERE cust_id = 10005;`
3. Click Execute.

Deleting Data

1. Click New Query.
2. In the tab that opens type: `DELETE FROM customers WHERE cust_id = 10006;` Click Execute.

Congratulations! You have finished this lab.

Lab 7 – Backing Up a Database

1. Open SSMS and connect to your SQL Server instance.
2. Expand the Databases node in Object Explorer.
3. Right-click the model database, hover over Tasks, and select Back up….
4. In the Back Up Database dialog box, ensure that Backup type is set to Full and Backup component is set to Database.
5. Under Destination, confirm that the path for your backup is correct. The path should be d:\. If you need to change the path, select Remove to remove the existing path, and then select Add to specify a new path.
6. Click OK to start the backup.

Alternatively you can use an SQL Statement to backup the database.

```
BACKUP DATABASE model
TO DISK = 'd:\BackupFile.bak'
WITH FORMAT;
```

Congratulations! You have finished this lab.