IS664 Database Programming
Fall 2022

Fundamentals

Database Programming

LECTURE 1: SCRIPT PROGRAMMING

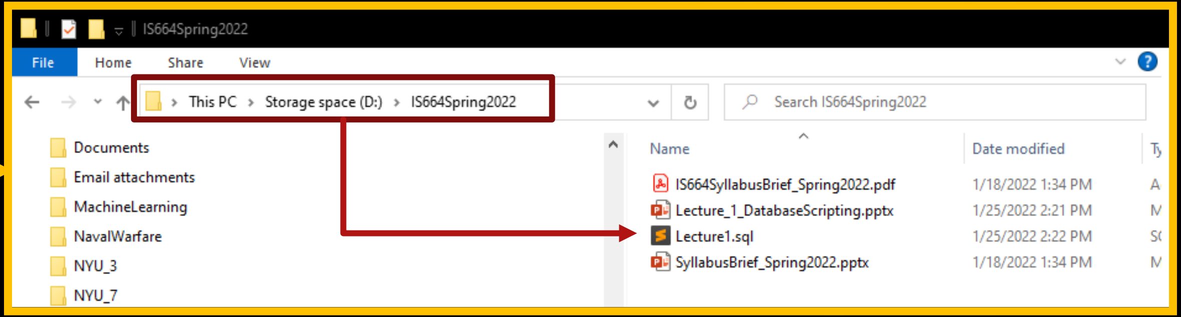
Professor HG Locklear
hlocklear@pace.edu

MySQL Scripts

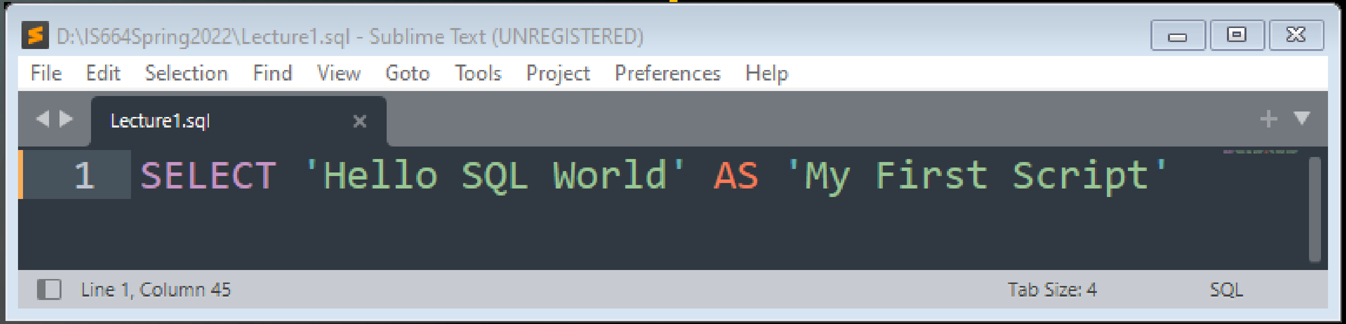
- A **Script** is simply a MySQL program (SQL Commands) than runs on the MySQL Community Server.
- A script can be created using any plain text editor such as Notepad, WordPad, Notepad++, etc.
- MySQL has a built-in graphical script development tool known as MySQL Workbench which can also be used.
  - **https://dev.mysql.com/downloads/workbench/**
- I recommend the use of Sublime Text 4… which is a very sophisticated plain text editor with features designed to assist with SQL script creation.
  - **https://www.sublimetext.com/**

Script Creation

- In order to create a script:
  - Open your plain text editor.
  - Type some SQL Commands.
  - Save the file as a .sql file.



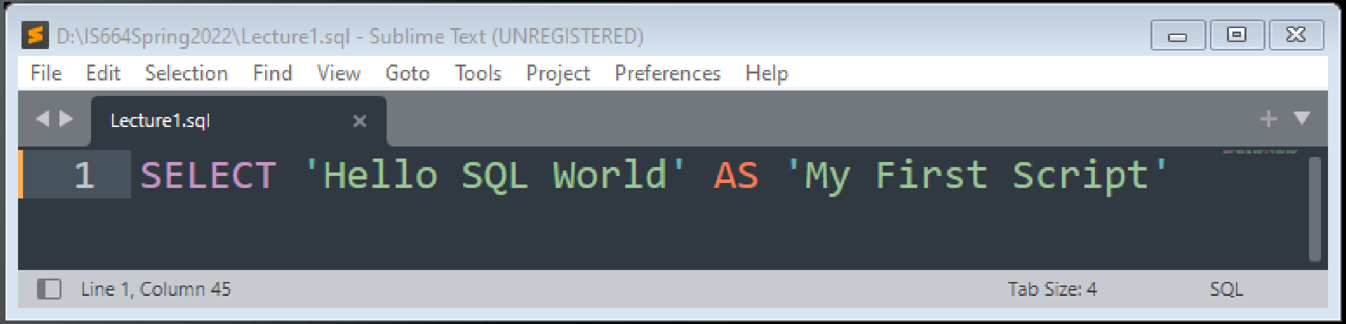
Name	Date modified
IS664SyllabusBrief_Spring2022.pdf	1/18/2022 1:34 PM
Lecture_1_DatabaseScripting.pptx	1/25/2022 2:21 PM
Lecture1.sql	1/25/2022 2:22 PM
SyllabusBrief_Spring2022.pptx	1/18/2022 1:34 PM



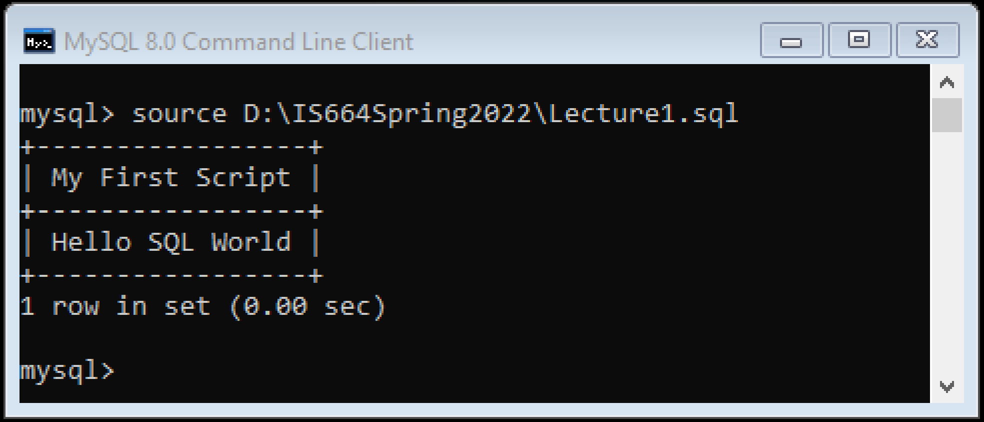
```sql
SELECT 'Hello SQL World' AS 'My First Script'
```

Script Execution

- In order to execute a script:
  - Open the MySQL 8.0 Command Line Client from your Start Menu.
  - At the MySQL Shell prompt, type source ***filepath to your script***.
  - The **source** command is used to execute all MySQL scripts.
  - If your filepath contains any spaces, surround the entire filepath in single quotes ‘***filepath to your script***’.
  - Script executes in the MySQL shell and displays any output.



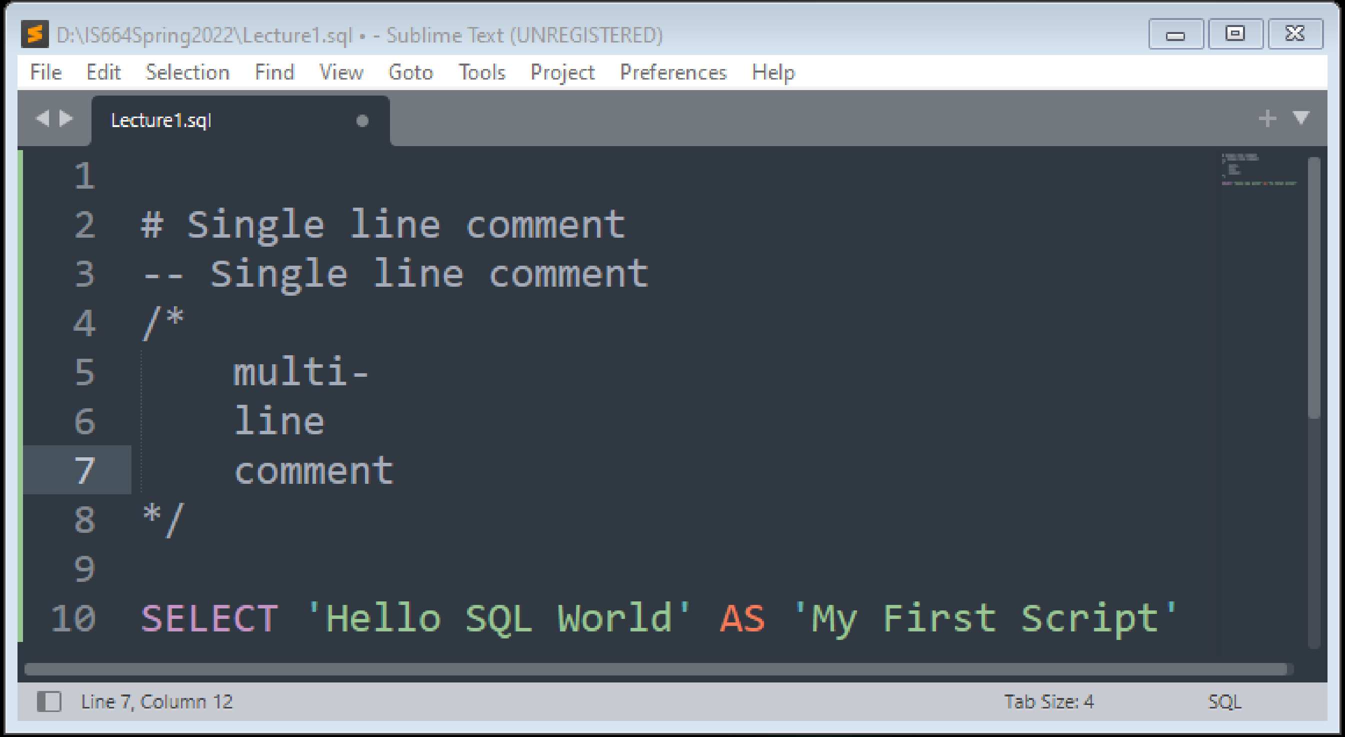
```sql
SELECT 'Hello SQL World' AS 'My First Script'
```



```
mysql> source D:\IS664Spring2022\Lecture1.sql
+-----------------+
| My First Script |
+-----------------+
| Hello SQL World |
+-----------------+
1 row in set (0.00 sec)

mysql>
```

Comments in MySQL

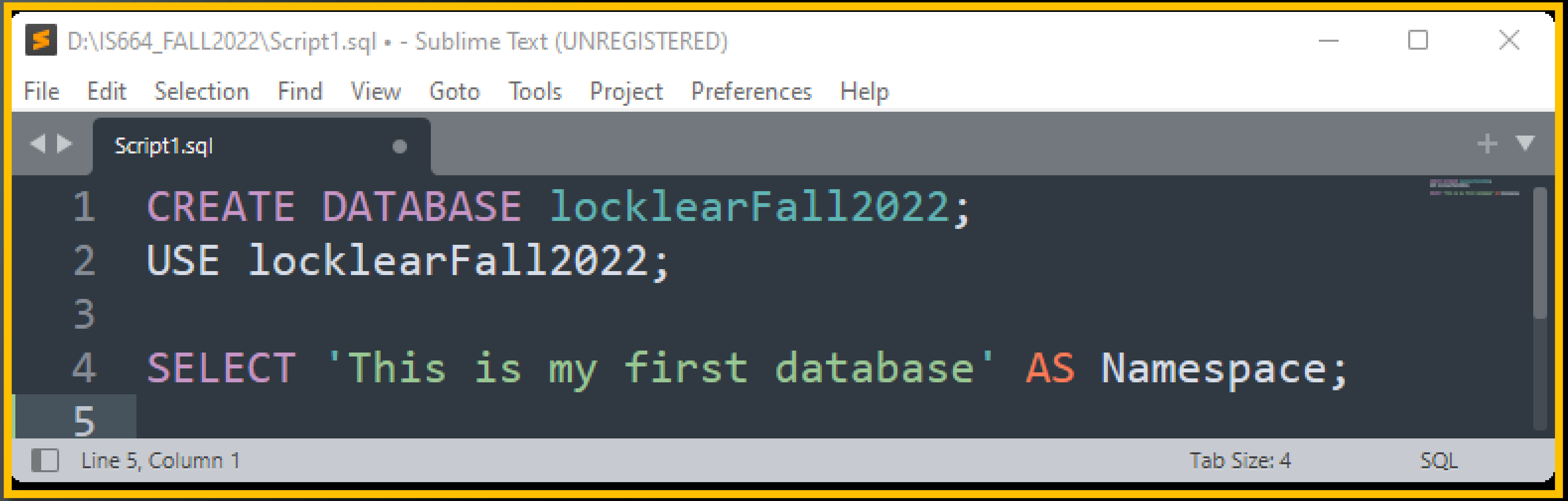
- **Comments** are a way of including text in our script that is not executed (ignored) by the MySQL compiler. Comments are very useful for documenting our program.
- MySQL supports three types of comments:
  - # Comments out a single line
  - -- Comments out a single line
  - /* */ Comments out multiple lines


D:\IS664Spring2022\Lecture1.sql • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Lecture1.sql
Line 7, Column 12
Tab Size: 4
SQL

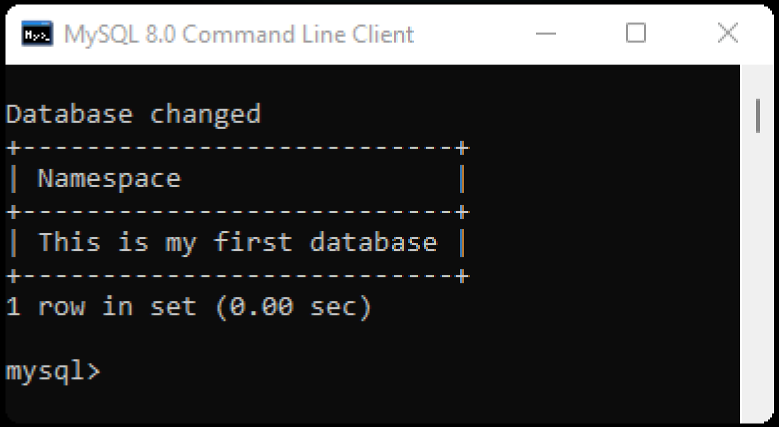
```

# Single line comment
-- Single line comment
/*
    multi-
    line
    comment
*/

SELECT 'Hello SQL World' AS 'My First Script'
```

Namespace in MySQL

- **Namespace** is the ‘box’ where our script executes on the MySQL Community Server.
- We can think of the server as a large block of memory and our namespace is the block of memory we are currently using.
- We define our namespace as a database, and we use the CREATE DATABASE command to create it.
- Anytime we want to utilize our namespace, we use the USE command to access the namespace.


D:\IS664_FALL2022\Script1.sql • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Script1.sql
Line 5, Column 1
Tab Size: 4
SQL

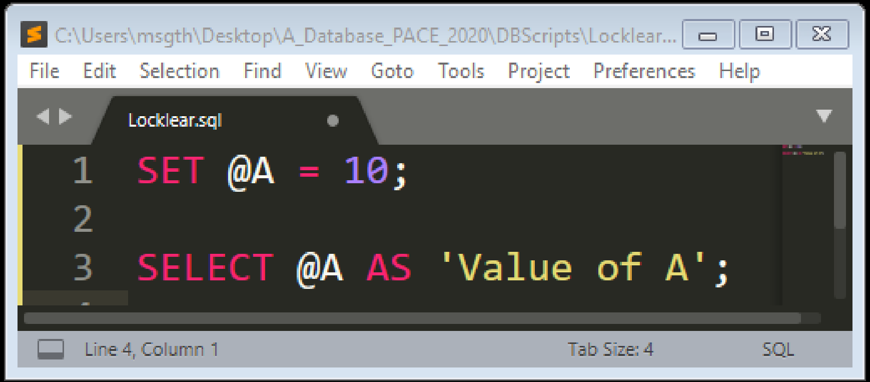
```sql
CREATE DATABASE locklearFall2022;
USE locklearFall2022;

SELECT 'This is my first database' AS Namespace;

```



```
Database changed
+----------------------------+
| Namespace                  |
+----------------------------+
| This is my first database |
+----------------------------+
1 row in set (0.00 sec)

mysql>
```

User-Defined Variables

- A value can be stored in a **user-defined variable** on the MySQL server.
  - We can think of this as a **Global Sessions variable** (available everywhere while we are logged in)
- Once stored the value, can be retrieved for some use in the program.
- This provides the ability to pass values from one statement to another.
- User-Defined variables are written as @variable_name where variable_name consist of alphanumeric characters, periods, underscores, and dollar signs (. _ $)
- We assign a value to a user-defined variable using the **SET** command.
  - **SET @A = 1;**
- User-Defined variable created by one client cannot be seen by another client.
- All variables for a given client session are automatically freed when the client exits the session.
- User variable names are not case sensitive.
- Names have a maximum length of 64 characters.



```sql
SET @A = 10;

SELECT @A AS 'Value of A';
```

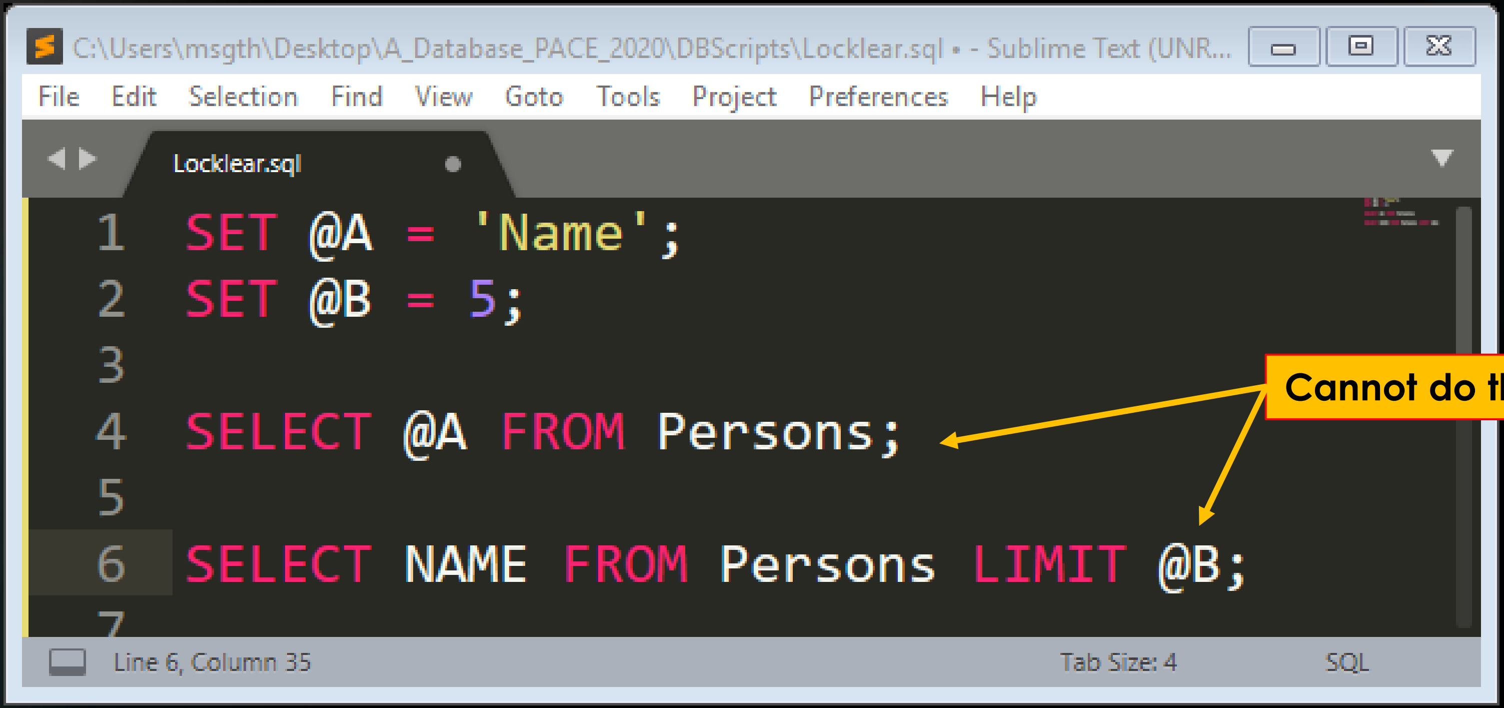
User-Defined Variables

- User-Defined variables can be assigned a value from a limited set of data types.
  - **Integer** (1,2,3..)
  - **Decimal** (1.1,1.2,1.3…)
  - **Floating Point** (1.1,1.2,1.3…)
  - **Binary String** (1010101)
  - **String** ('Gene')
  - **Null** (empty value)

- Precision or scale is not preserved (MySQL determines precision)
- A value other than the permissible types is converted to a permissible type.
- Variables that have not been initialized have a value of **Null** and a type of **String**.
- User-Defined variables cannot be used in a context that require a literal value such as in the **LIMIT** clause of a **SELECT** statement.

User-Defined Variables

- User-Defined variables are intended to provide data values and cannot be used directly in an SQL statement as an identifier or as part of an identifier.


C:\Users\msgth\Desktop\A_Database_PACE_2020\DBScripts\Locklear.sql • - Sublime Text (UNR...
File Edit Selection Find View Goto Tools Project Preferences Help
Locklear.sql
Line 6, Column 35
Tab Size: 4
SQL

```sql
SET @A = 'Name';
SET @B = 5;

SELECT @A FROM Persons;

SELECT NAME FROM Persons LIMIT @B;

```

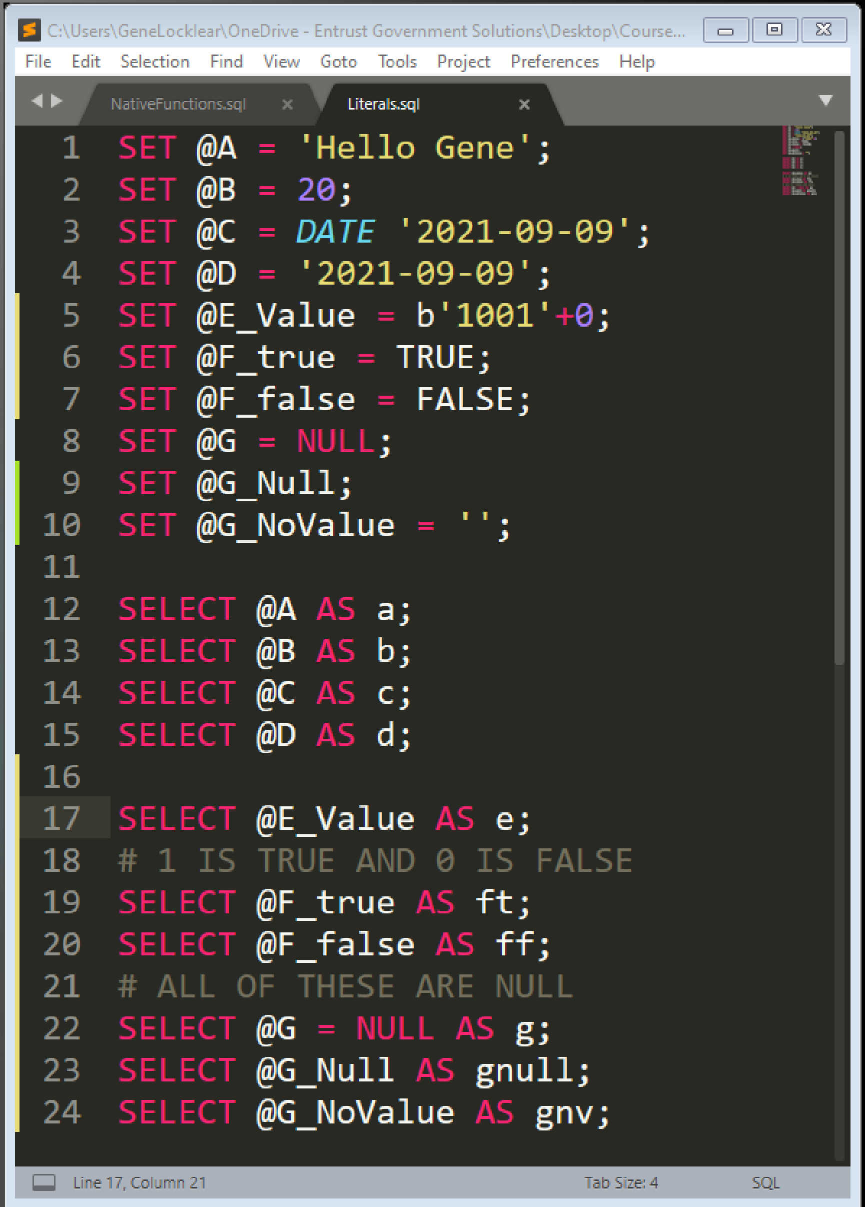
Cannot do this

Literals

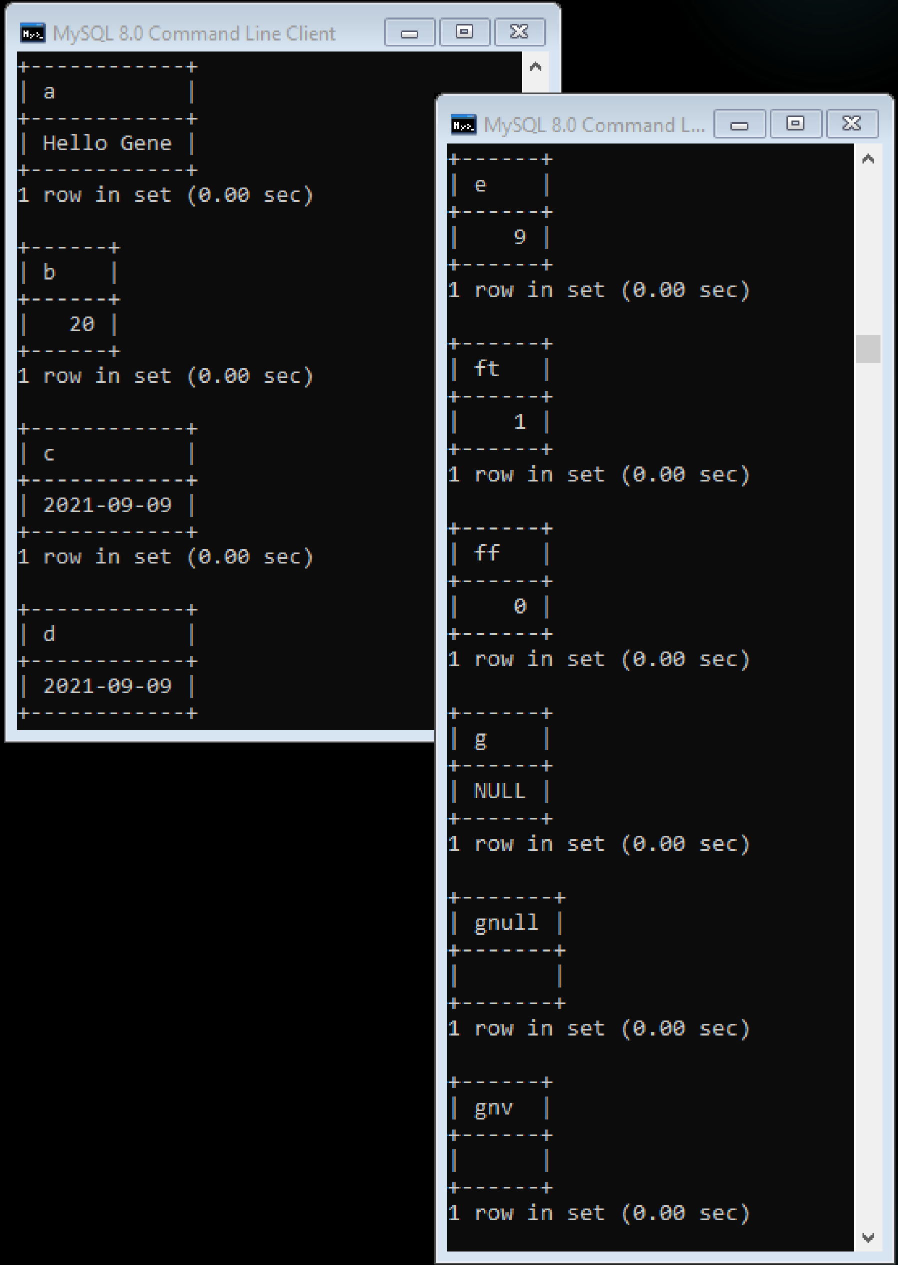
- A **literal** is the direct expression of a data type.
- In MySQL we have:
  - **String Literals** (**‘Hello Gene’**)
  - **Numeric Literals** (**123**)
  - **Date and Time Literals** (DATE **2020-01-30**) (DATETIME **2020-01-30 11:20:30**)
  - **Hexadecimal Literals** (**X’41’**) …interpreted as ASCII
  - **Bit Value Literals** (**b’1000001’ + 0** …interpreted as NUMERIC VALUE
  - **Boolean Literals** (**TRUE**) …represented as 1 …FALSE is 0
  - **NULL**

  - There are a **lot of nuances** when dealing with literals… you can get more information about it here.
  - **https://dev.mysql.com/doc/refman/8.0/en/literals.html**

Literals


C:\Users\GeneLocklear\OneDrive - Entrust Government Solutions\Desktop\Course...
File Edit Selection Find View Goto Tools Project Preferences Help
NativeFunctions.sql
Literals.sql
Line 17, Column 21
Tab Size: 4
SQL

```sql
SET @A = 'Hello Gene';
SET @B = 20;
SET @C = DATE '2021-09-09';
SET @D = '2021-09-09';
SET @E_Value = b'1001'+0;
SET @F_true = TRUE;
SET @F_false = FALSE;
SET @G = NULL;
SET @G_Null;
SET @G_NoValue = '';

SELECT @A AS a;
SELECT @B AS b;
SELECT @C AS c;
SELECT @D AS d;

SELECT @E_Value AS e;
# 1 IS TRUE AND 0 IS FALSE
SELECT @F_true AS ft;
SELECT @F_false AS ff;
# ALL OF THESE ARE NULL
SELECT @G = NULL AS g;
SELECT @G_Null AS gnull;
SELECT @G_NoValue AS gnv;
```

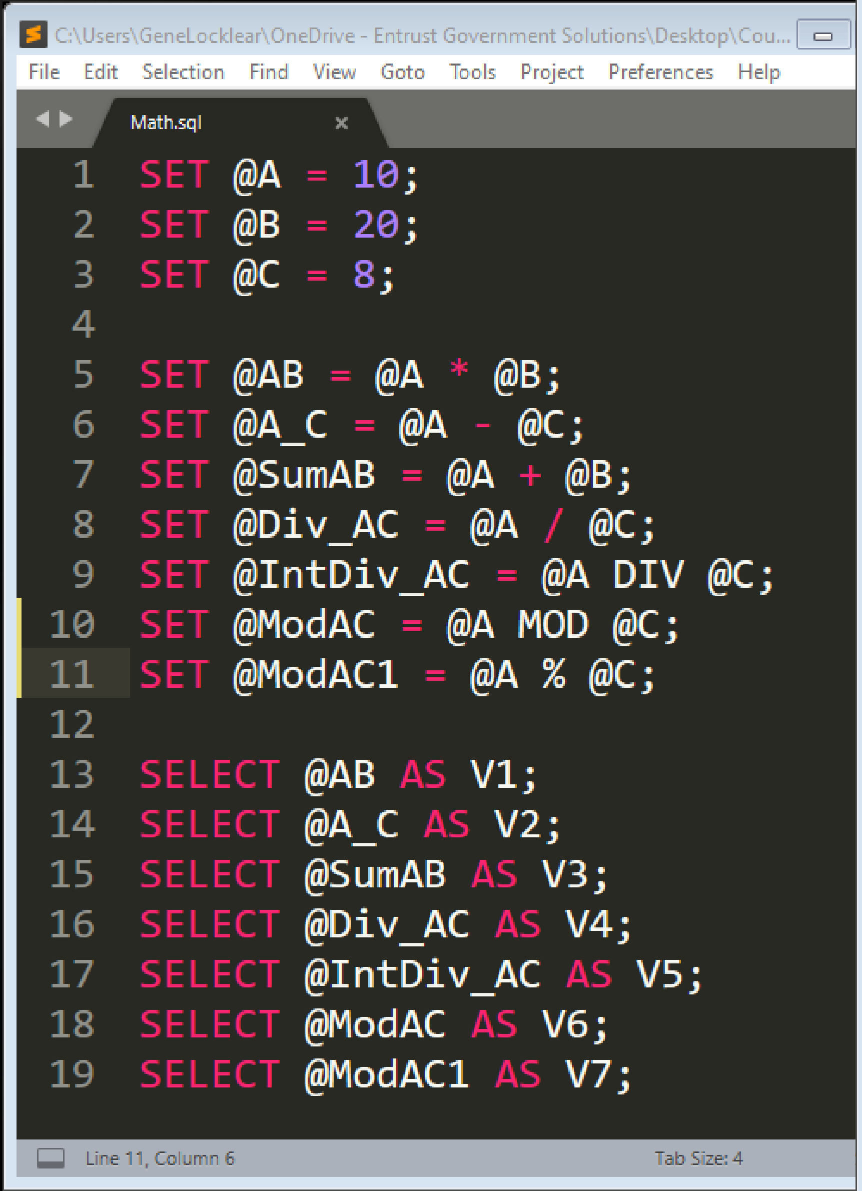


```
+------------+
| a          |
+------------+
| Hello Gene |
+------------+
1 row in set (0.00 sec)

+------+
| b    |
+------+
|   20 |
+------+
1 row in set (0.00 sec)

+------------+
| c          |
+------------+
| 2021-09-09 |
+------------+
1 row in set (0.00 sec)

+------------+
| d          |
+------------+
| 2021-09-09 |
+------------+
```

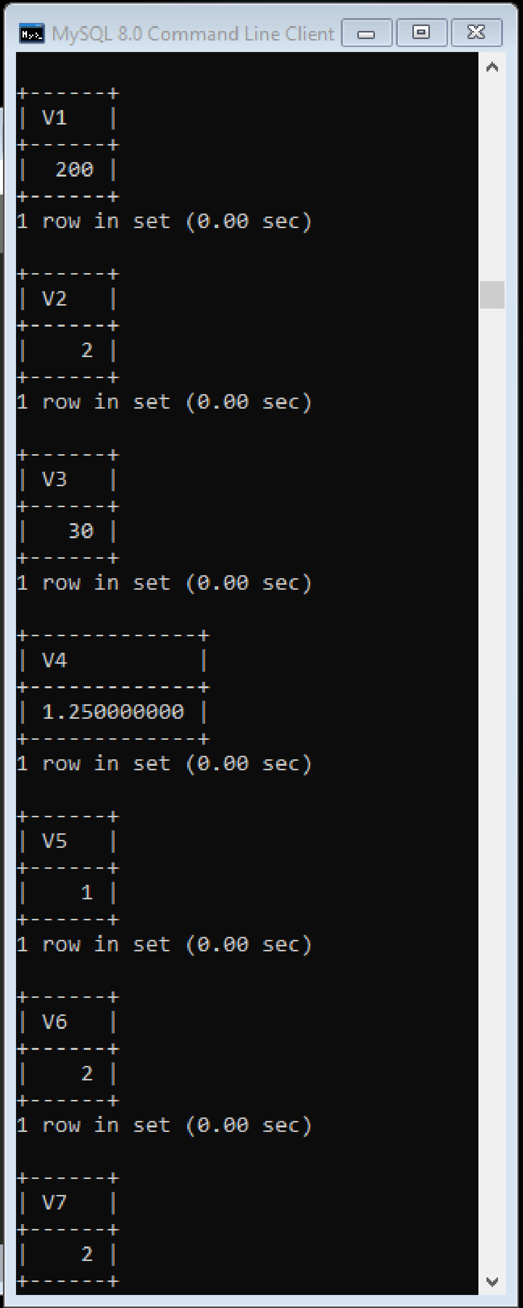
```
+------+
| e    |
+------+
|    9 |
+------+
1 row in set (0.00 sec)

+------+
| ft   |
+------+
|    1 |
+------+
1 row in set (0.00 sec)

+------+
| ff   |
+------+
|    0 |
+------+
1 row in set (0.00 sec)

+------+
| g    |
+------+
| NULL |
+------+
1 row in set (0.00 sec)

+-------+
| gnull |
+-------+
|       |
+-------+
1 row in set (0.00 sec)

+------+
| gnv  |
+------+
|      |
+------+
1 row in set (0.00 sec)
```

Mathematical Operators

Operator	Purpose
%	Modulus Division
MOD	Modulus Division
+	Addition
-	Subtraction
*	Multiplication
/	Division
DIV	Integer Division


C:\Users\GeneLocklear\OneDrive - Entrust Government Solutions\Desktop\Cou...
File Edit Selection Find View Goto Tools Project Preferences Help
Math.sql
Line 11, Column 6
Tab Size: 4

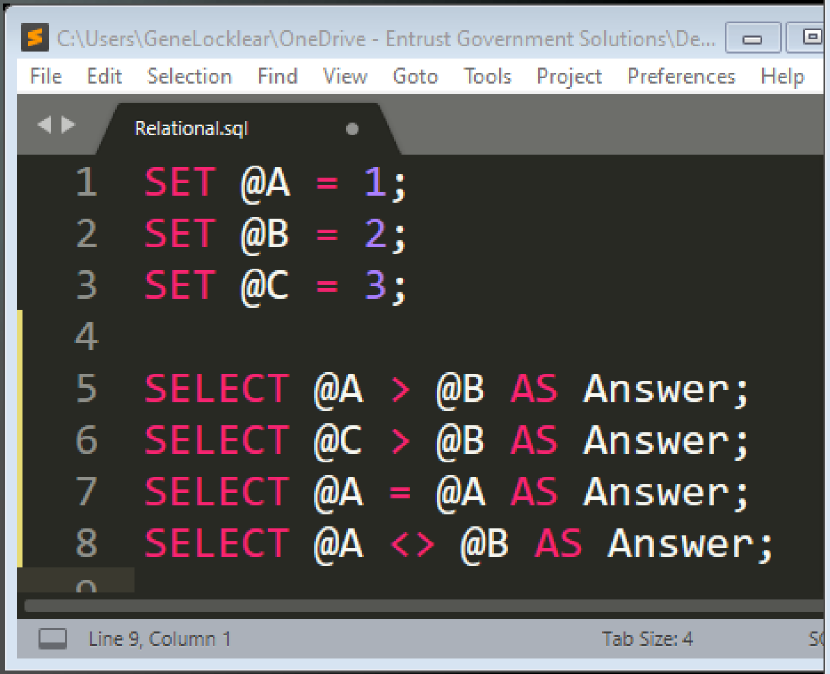
```sql
SET @A = 10;
SET @B = 20;
SET @C = 8;

SET @AB = @A * @B;
SET @A_C = @A - @C;
SET @SumAB = @A + @B;
SET @Div_AC = @A / @C;
SET @IntDiv_AC = @A DIV @C;
SET @ModAC = @A MOD @C;
SET @ModAC1 = @A % @C;

SELECT @AB AS V1;
SELECT @A_C AS V2;
SELECT @SumAB AS V3;
SELECT @Div_AC AS V4;
SELECT @IntDiv_AC AS V5;
SELECT @ModAC AS V6;
SELECT @ModAC1 AS V7;
```



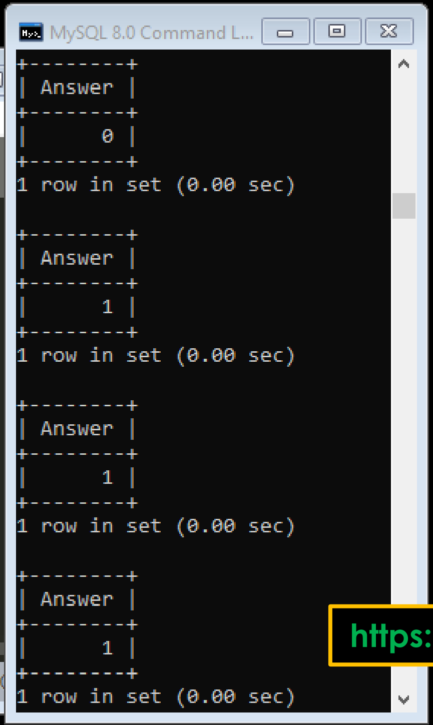
```
+------+
| V1   |
+------+
|  200 |
+------+
1 row in set (0.00 sec)

+------+
| V2   |
+------+
|    2 |
+------+
1 row in set (0.00 sec)

+------+
| V3   |
+------+
|   30 |
+------+
1 row in set (0.00 sec)

+-------------+
| V4          |
+-------------+
| 1.250000000 |
+-------------+
1 row in set (0.00 sec)

+------+
| V5   |
+------+
|    1 |
+------+
1 row in set (0.00 sec)

+------+
| V6   |
+------+
|    2 |
+------+
1 row in set (0.00 sec)

+------+
| V7   |
+------+
|    2 |
+------+
```

Comparison Operators

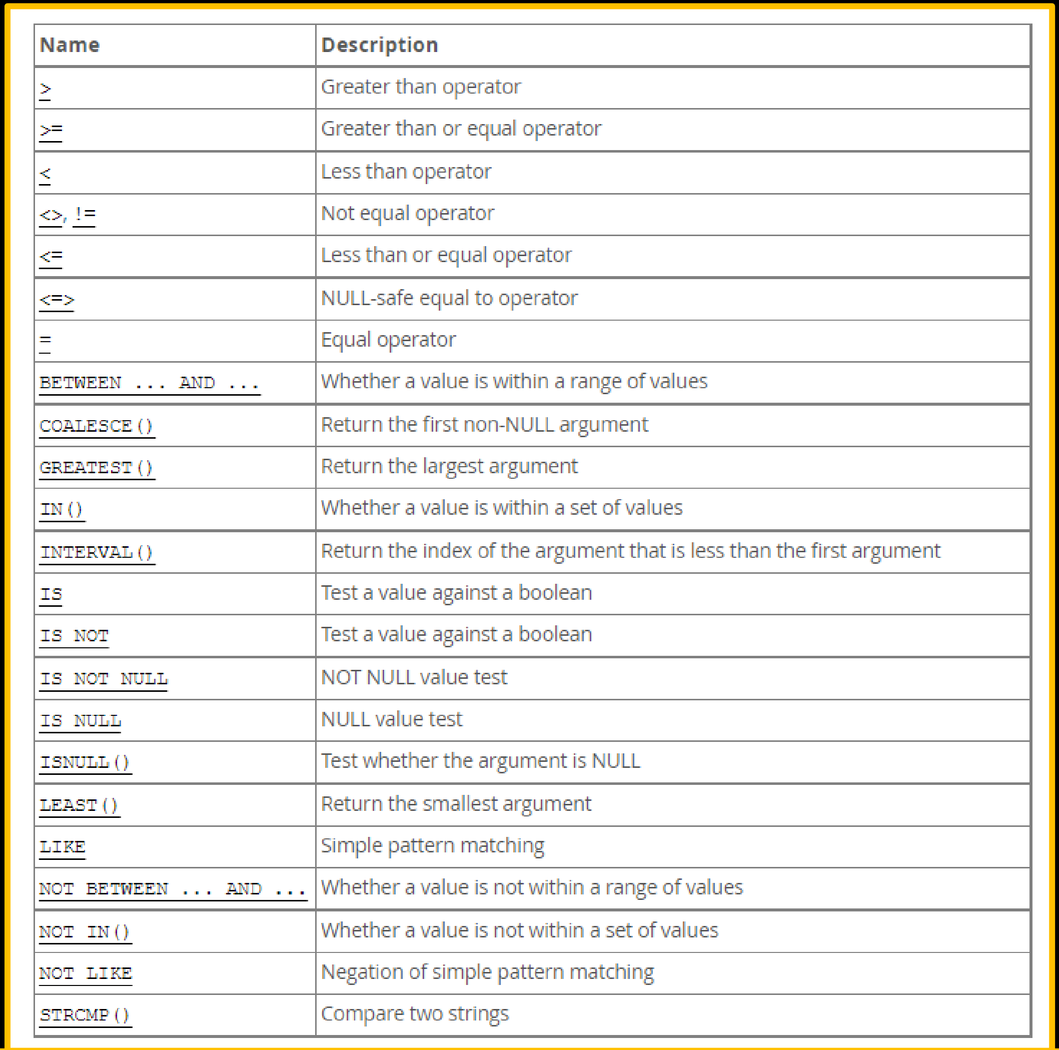
- **Comparison operators** result in a value of **1** (TRUE), **0** (FALSE), or **NULL**.
- Comparison operators work for both numbers and strings.
- Strings are automatically converted to numbers and vice versa.


C:\Users\GeneLocklear\OneDrive - Entrust Government Solutions\De...
File Edit Selection Find View Goto Tools Project Preferences Help
Relational.sql
Line 9, Column 1
Tab Size: 4

```sql
SET @A = 1;
SET @B = 2;
SET @C = 3;

SELECT @A > @B AS Answer;
SELECT @C > @B AS Answer;
SELECT @A = @A AS Answer;
SELECT @A <> @B AS Answer;
```



```
+--------+
| Answer |
+--------+
|      0 |
+--------+
1 row in set (0.00 sec)

+--------+
| Answer |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)

+--------+
| Answer |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)

+--------+
| Answer |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)
```



Name	Description
>	Greater than operator
>=	Greater than or equal operator
<	Less than operator
<>, !=	Not equal operator
<=	Less than or equal operator
<=>	NULL-safe equal to operator
=	Equal operator
BETWEEN ... AND ...	Whether a value is within a range of values
COALESCE()	Return the first non-NULL argument
GREATEST()	Return the largest argument
IN()	Whether a value is within a set of values
INTERVAL()	Return the index of the argument that is less than the first argument
IS	Test a value against a boolean
IS NOT	Test a value against a boolean
IS NOT NULL	NOT NULL value test
IS NULL	NULL value test
ISNULL()	Test whether the argument is NULL
LEAST()	Return the smallest argument
LIKE	Simple pattern matching
NOT BETWEEN ... AND ...	Whether a value is not within a range of values
NOT IN()	Whether a value is not within a set of values
NOT LIKE	Negation of simple pattern matching
STRCMP()	Compare two strings

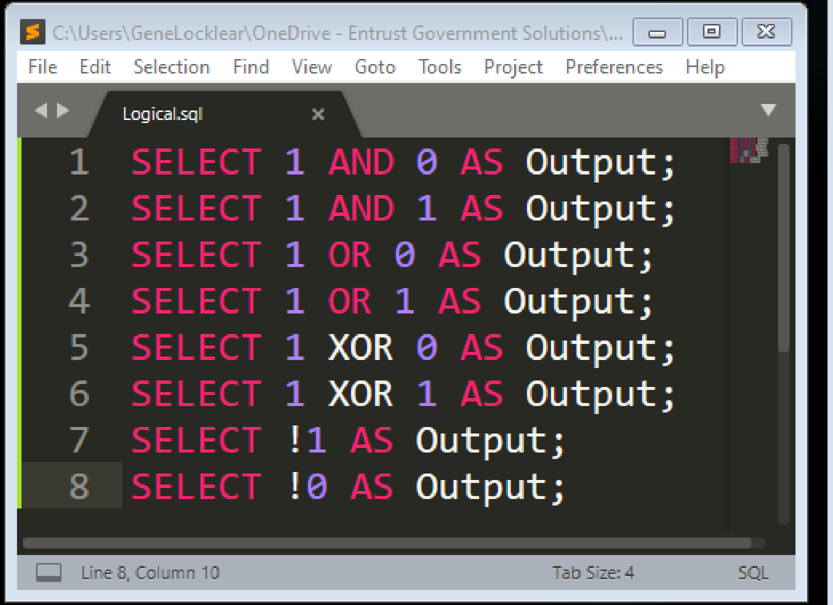
https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html#operator_is

Logical Operators

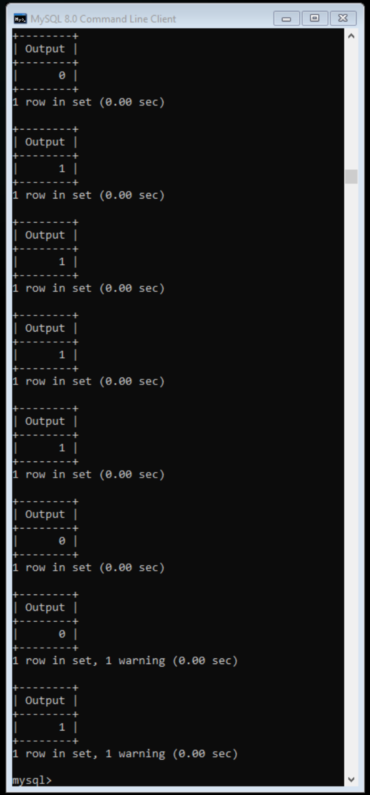
- **Logical operators** result in a value of **1** (TRUE), **0** (FALSE), or **NULL**.

- AND (&&)
  - Evaluates to 1 of all operands are nonzero and not NULL.
  - Evaluates to 0 if one or more operands are 0.
  - Evaluates otherwise to NULL.
- NOT(!)
  - Evaluates to 1 if the operand is 0.
  - Evaluates to 0 if the operand is nonzero.
  - Evaluates to NULL if operand is NULL.
- OR (| |)
  - Evaluates to 1 if both operands are non-NULL.
  - Evaluates to 1 if either operand is non-NULL .
  - Evaluates to 0 if both operands are 0.
  - Evaluates to NULL if both operands are NULL.

- XOR
  - Think of as 0 only if both operands are the same.
  - Evaluates to NULL if either operand is NULL.



```sql
SELECT 1 AND 0 AS Output;
SELECT 1 AND 1 AS Output;
SELECT 1 OR 0 AS Output;
SELECT 1 OR 1 AS Output;
SELECT 1 XOR 0 AS Output;
SELECT 1 XOR 1 AS Output;
SELECT !1 AS Output;
SELECT !0 AS Output;
```



```
+--------+
| Output |
+--------+
|      0 |
+--------+
1 row in set (0.00 sec)

+--------+
| Output |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)

+--------+
| Output |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)

+--------+
| Output |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)

+--------+
| Output |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)

+--------+
| Output |
+--------+
|      0 |
+--------+
1 row in set (0.00 sec)

+--------+
| Output |
+--------+
|      0 |
+--------+
1 row in set, 1 warning (0.00 sec)

+--------+
| Output |
+--------+
|      1 |
+--------+
1 row in set, 1 warning (0.00 sec)

mysql>
```

Control Flow Functions

- **Control Flow functions** allow our scripts to make decisions.
- MySQL has four control flow functions.

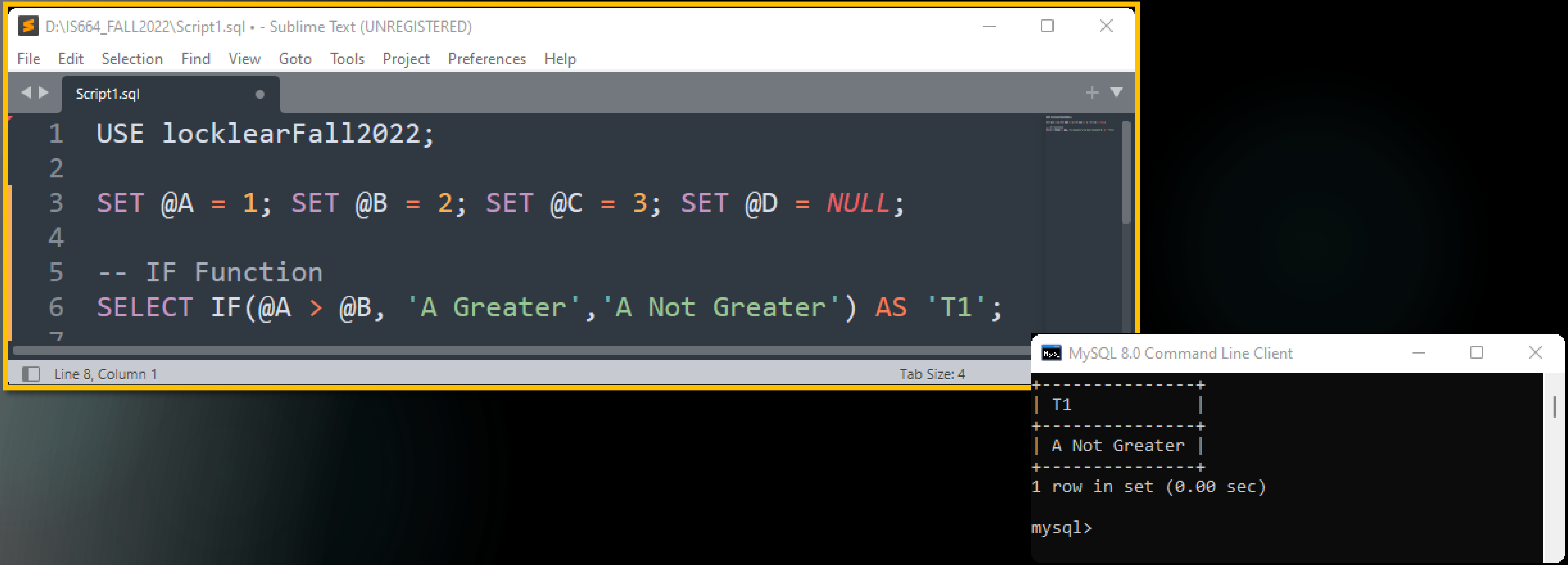
IF()
If expression_1 is TRUE, Then return expression_2 else return expression_3

IFNULL()
If expression_1 is NOT NULL, Then return expression_1 else return expression_2

NULLIF()
If expression_1 is equal to expression_2 return NULL

CASE
CASE value WHEN [value] THEN result [WHEN [value] THEN result ...] [ELSE result] END

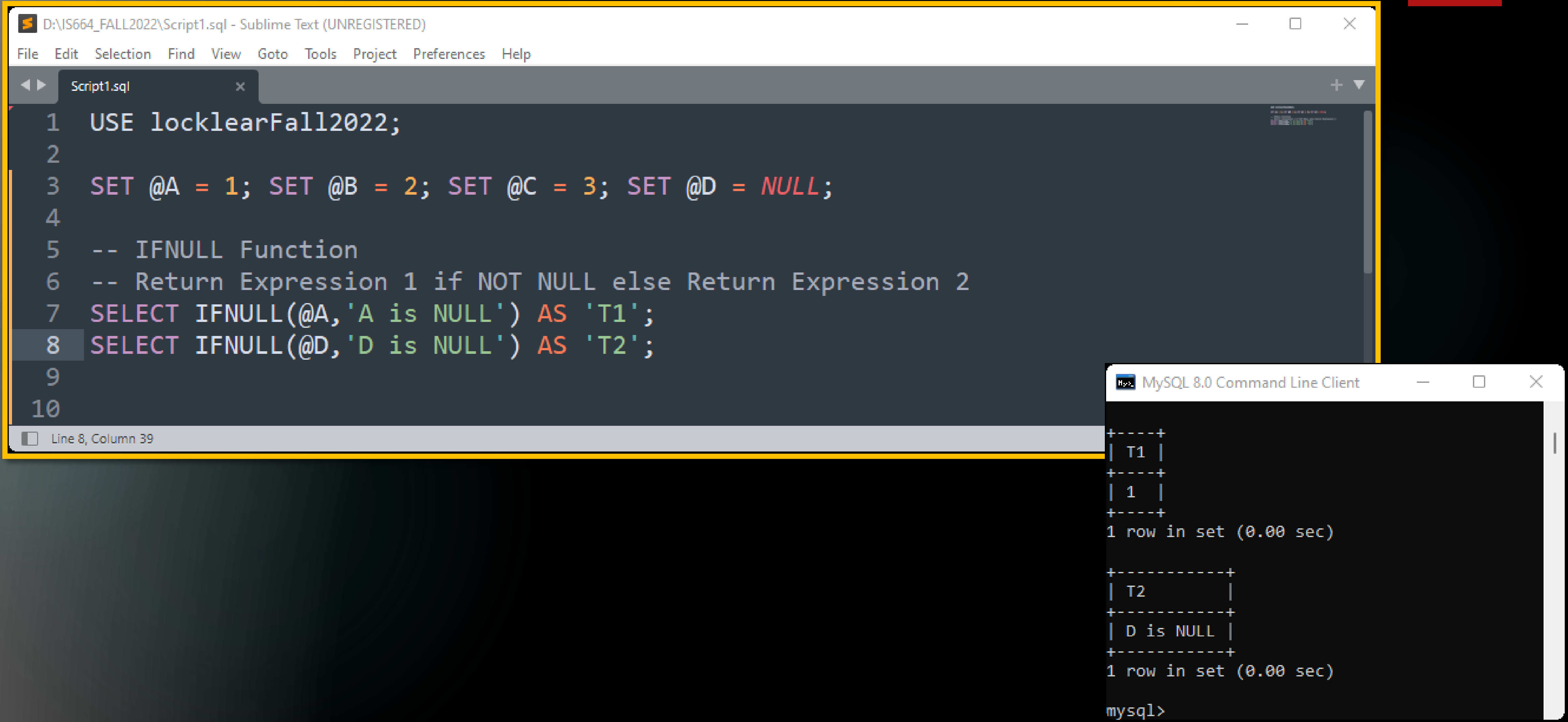
Control Flow Functions


D:\IS664_FALL2022\Script1.sql • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Script1.sql
Line 8, Column 1
Tab Size: 4
MySQL 8.0 Command Line Client

```sql
USE locklearFall2022;

SET @A = 1; SET @B = 2; SET @C = 3; SET @D = NULL;

-- IF Function
SELECT IF(@A > @B, 'A Greater','A Not Greater') AS 'T1';
```

```
+---------------+
| T1            |
+---------------+
| A Not Greater |
+---------------+
1 row in set (0.00 sec)

mysql>
```

Control Flow Functions

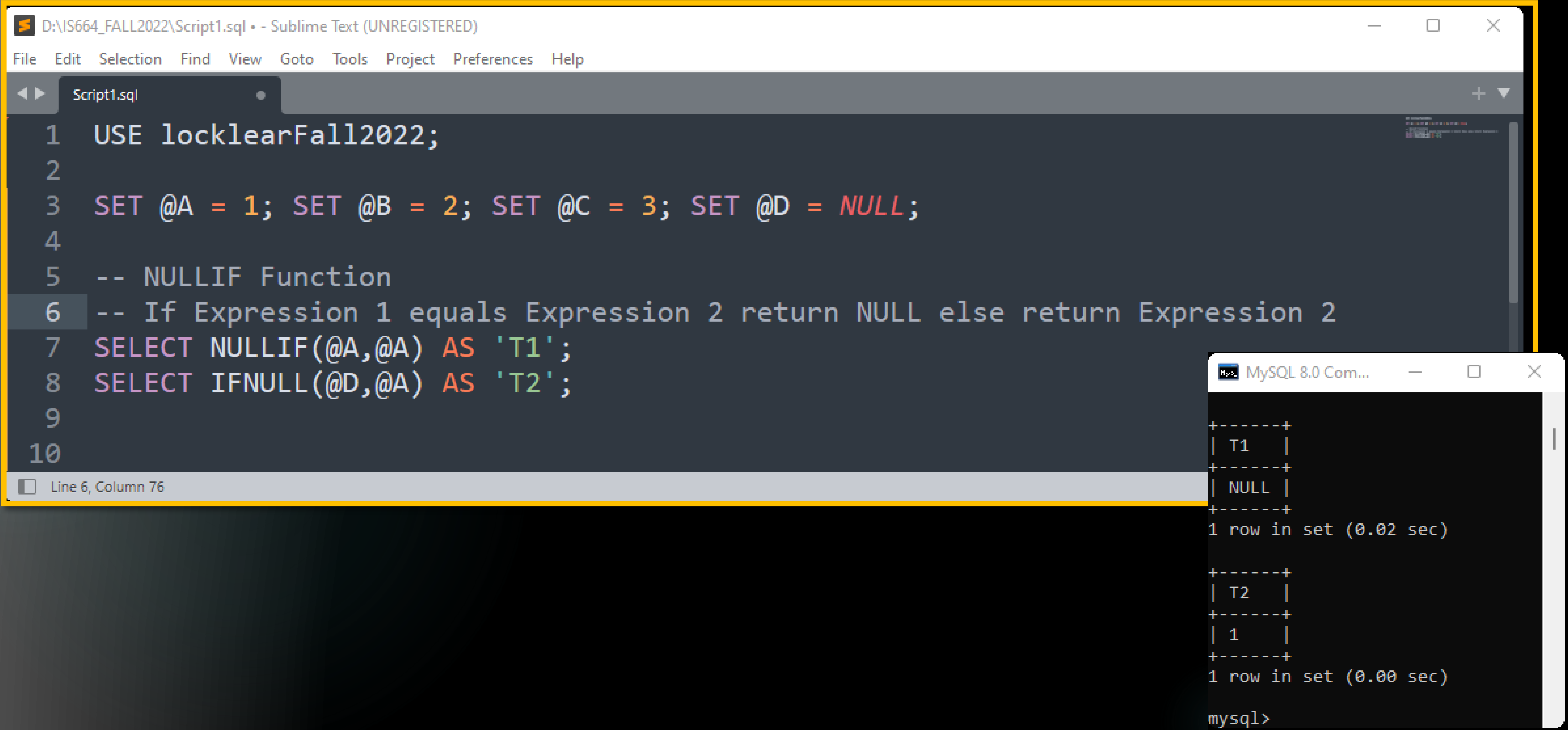


D:\IS664_FALL2022\Script1.sql - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Script1.sql
Line 8, Column 39
MySQL 8.0 Command Line Client

```sql
USE locklearFall2022;

SET @A = 1; SET @B = 2; SET @C = 3; SET @D = NULL;

-- IFNULL Function
-- Return Expression 1 if NOT NULL else Return Expression 2
SELECT IFNULL(@A,'A is NULL') AS 'T1';
SELECT IFNULL(@D,'D is NULL') AS 'T2';
```

```
+----+
| T1 |
+----+
| 1  |
+----+
1 row in set (0.00 sec)

+-----------+
| T2        |
+-----------+
| D is NULL |
+-----------+
1 row in set (0.00 sec)

mysql>
```

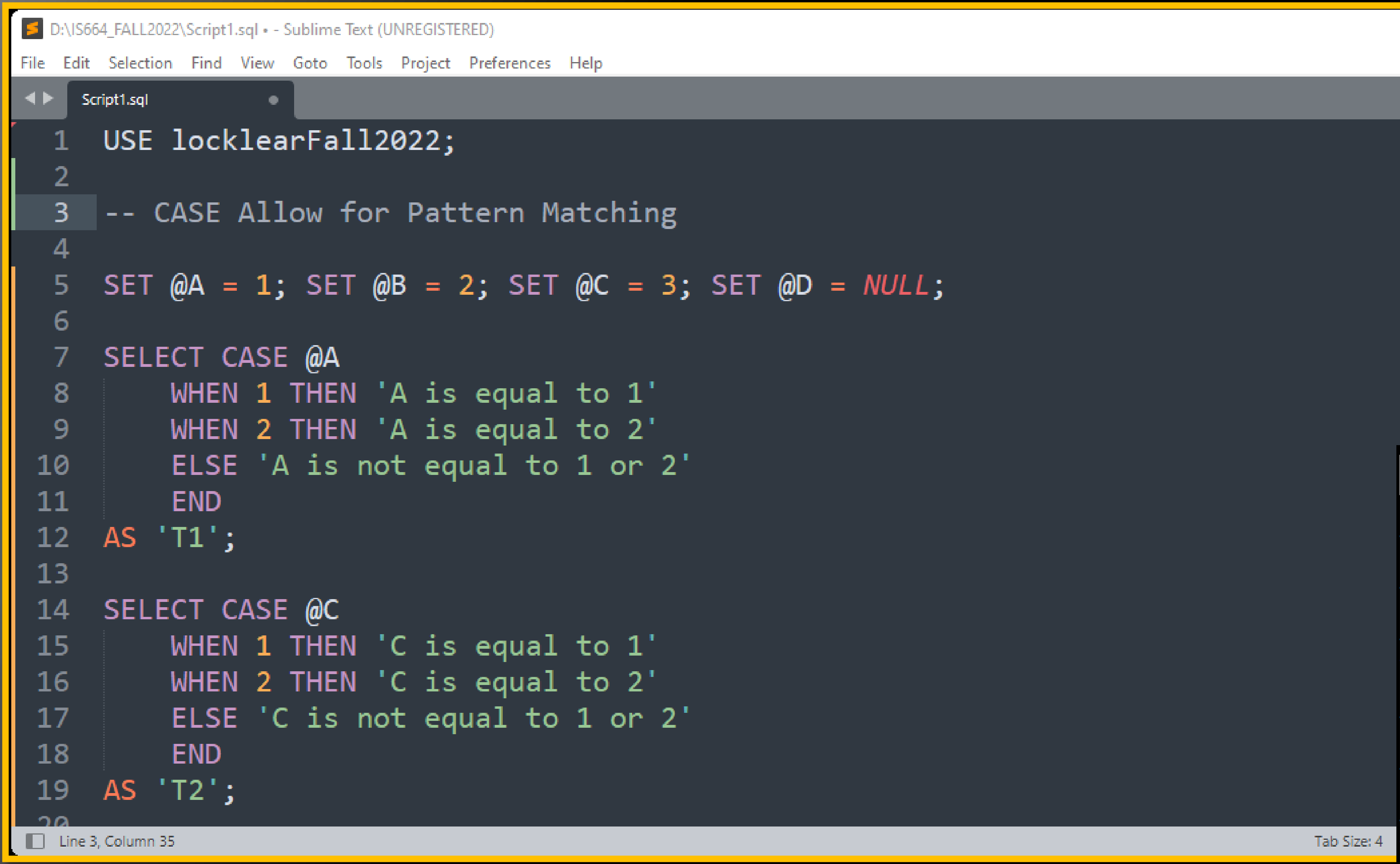
Control Flow Functions


D:\IS664_FALL2022\Script1.sql • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Script1.sql
Line 6, Column 76
MySQL 8.0 Com...

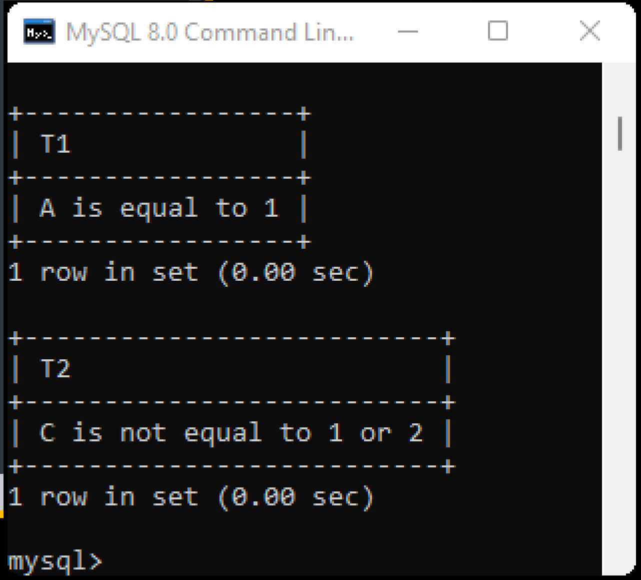
```sql
USE locklearFall2022;

SET @A = 1; SET @B = 2; SET @C = 3; SET @D = NULL;

-- NULLIF Function
-- If Expression 1 equals Expression 2 return NULL else return Expression 2
SELECT NULLIF(@A,@A) AS 'T1';
SELECT IFNULL(@D,@A) AS 'T2';
```

```
+------+
| T1   |
+------+
| NULL |
+------+
1 row in set (0.02 sec)

+------+
| T2   |
+------+
| 1    |
+------+
1 row in set (0.00 sec)

mysql>
```

Control Flow Functions


D:\IS664_FALL2022\Script1.sql • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Script1.sql
Line 3, Column 35
Tab Size: 4

```
USE locklearFall2022;

-- CASE Allow for Pattern Matching

SET @A = 1; SET @B = 2; SET @C = 3; SET @D = NULL;

SELECT CASE @A
    WHEN 1 THEN 'A is equal to 1'
    WHEN 2 THEN 'A is equal to 2'
    ELSE 'A is not equal to 1 or 2'
    END
AS 'T1';

SELECT CASE @C
    WHEN 1 THEN 'C is equal to 1'
    WHEN 2 THEN 'C is equal to 2'
    ELSE 'C is not equal to 1 or 2'
    END
AS 'T2';
```



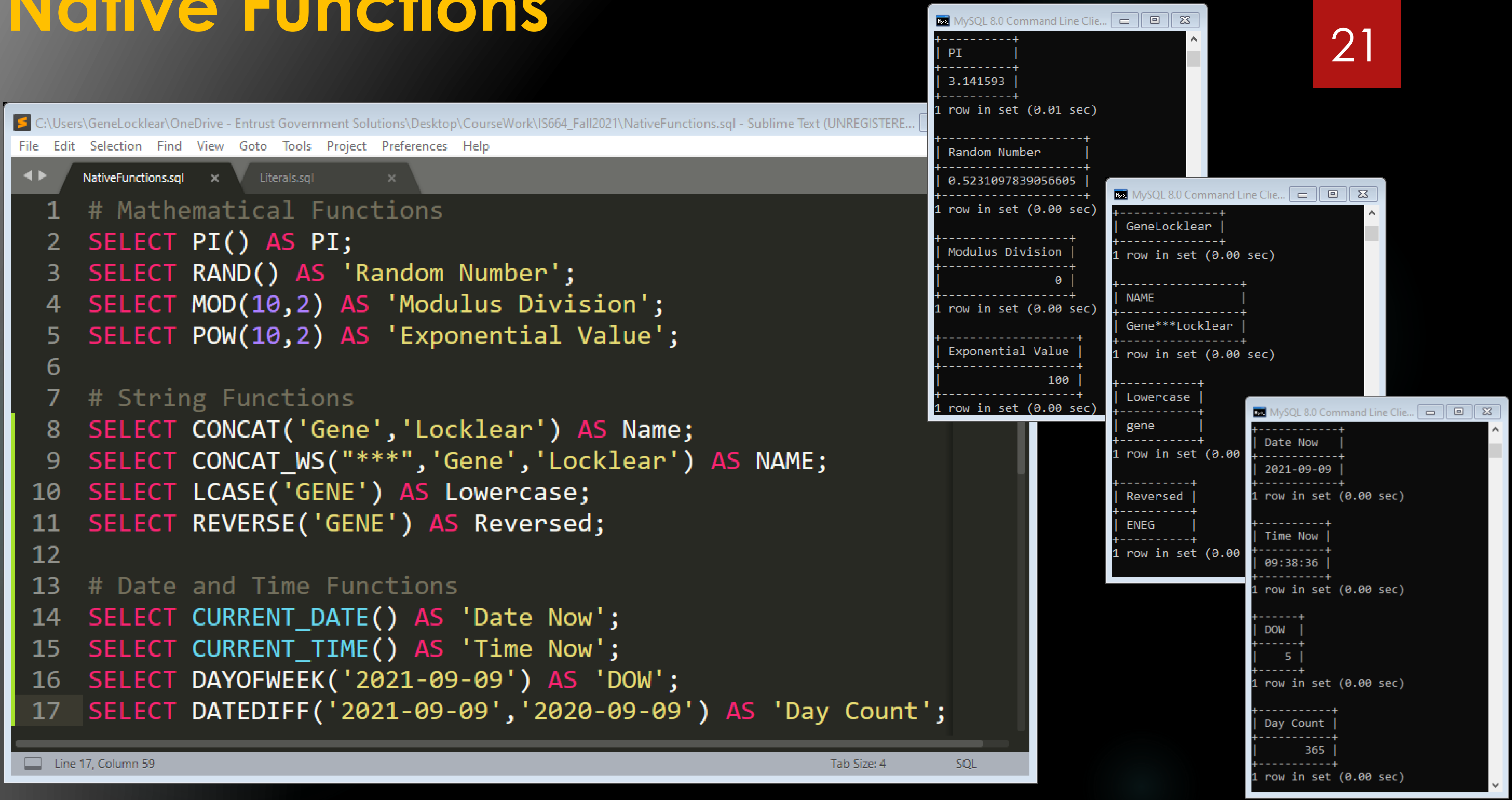
```
+-----------------+
| T1              |
+-----------------+
| A is equal to 1 |
+-----------------+
1 row in set (0.00 sec)

+--------------------------+
| T2                       |
+--------------------------+
| C is not equal to 1 or 2 |
+--------------------------+
1 row in set (0.00 sec)

mysql>
```

Native Functions

- **Native Functions** are built in self-contained blocks of code that can be used to perform operations.
- MySQL has Mathematical functions
  - **https://dev.mysql.com/doc/refman/5.7/en/mathematical-functions.html**
- MySQL has String functions
  - **https://dev.mysql.com/doc/refman/5.7/en/string-functions.html**
- MySQL has Date and Time functions
  - **https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html**

- Native function will assist in performing operations in which writing our own code to perform the operation would be extremely difficult or impossible.

Native Functions



C:\Users\GeneLocklear\OneDrive - Entrust Government Solutions\Desktop\CourseWork\IS664_Fall2021\NativeFunctions.sql - Sublime Text (UNREGISTERE...
File Edit Selection Find View Goto Tools Project Preferences Help
NativeFunctions.sql
Literals.sql
Line 17, Column 59
Tab Size: 4
SQL
MySQL 8.0 Command Line Clie...
MySQL 8.0 Command Line Clie...
MySQL 8.0 Command Line Clie...

```sql
# Mathematical Functions
SELECT PI() AS PI;
SELECT RAND() AS 'Random Number';
SELECT MOD(10,2) AS 'Modulus Division';
SELECT POW(10,2) AS 'Exponential Value';

# String Functions
SELECT CONCAT('Gene','Locklear') AS Name;
SELECT CONCAT_WS("***",'Gene','Locklear') AS NAME;
SELECT LCASE('GENE') AS Lowercase;
SELECT REVERSE('GENE') AS Reversed;

# Date and Time Functions
SELECT CURRENT_DATE() AS 'Date Now';
SELECT CURRENT_TIME() AS 'Time Now';
SELECT DAYOFWEEK('2021-09-09') AS 'DOW';
SELECT DATEDIFF('2021-09-09','2020-09-09') AS 'Day Count';
```

```
+----------+
| PI       |
+----------+
| 3.141593 |
+----------+
1 row in set (0.01 sec)

+--------------------+
| Random Number      |
+--------------------+
| 0.5231097839056605 |
+--------------------+
1 row in set (0.00 sec)

+------------------+
| Modulus Division |
+------------------+
|                0 |
+------------------+
1 row in set (0.00 sec)

+-------------------+
| Exponential Value |
+-------------------+
|               100 |
+-------------------+
1 row in set (0.00 sec)
```

```
+--------------+
| GeneLocklear |
+--------------+
1 row in set (0.00 sec)

+----------------+
| NAME           |
+----------------+
| Gene***Locklear |
+----------------+
1 row in set (0.00 sec)

+-----------+
| Lowercase |
+-----------+
| gene      |
+-----------+
1 row in set (0.00

+----------+
| Reversed |
+----------+
| ENEG     |
+----------+
1 row in set (0.00
```

```
+------------+
| Date Now   |
+------------+
| 2021-09-09 |
+------------+
1 row in set (0.00 sec)

+----------+
| Time Now |
+----------+
| 09:38:36 |
+----------+
1 row in set (0.00 sec)

+-----+
| DOW |
+-----+
|   5 |
+-----+
1 row in set (0.00 sec)

+-----------+
| Day Count |
+-----------+
|       365 |
+-----------+
1 row in set (0.00 sec)
```