



Ten Minute Design 5 Stored Procedures 1

Elementary School Library

Software Engineering
CSCI-3321

Dr. Tom Hicks
Computer Science Department

Grab Documents

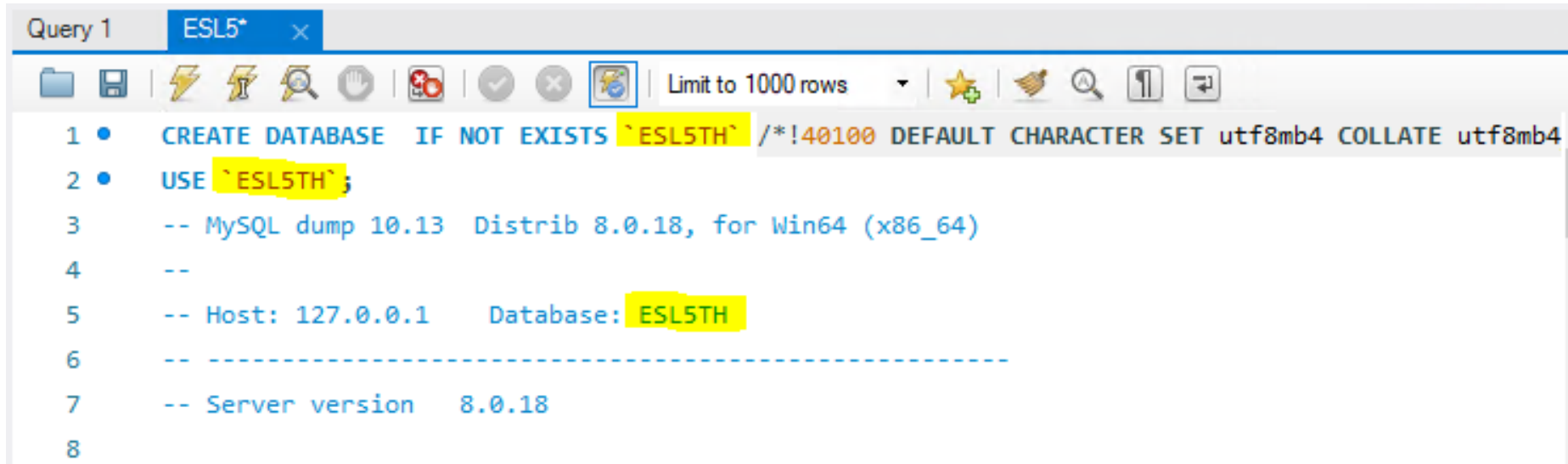
Grab The Database

- Copy **ESL5.sql** from The Schedule Page
OR From Mars

Import Database ESL5

Import ESL5.sql

1 - Load MySQL Workbench - Import **ESL5.sql** → Call The Database **ELS5TH** ← Using Your Initials.



The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for file operations, execution, and search. The main area displays a SQL query in a text editor. The query is as follows:

```
1 • CREATE DATABASE IF NOT EXISTS `ESL5TH` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4
2 • USE `ESL5TH`;
3 -- MySQL dump 10.13  Distrib 8.0.18, for Win64 (x86_64)
4 --
5 -- Host: 127.0.0.1    Database: ESL5TH
6 -----
7 -- Server version  8.0.18
8
```

Note That Table User Has No Records

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
◇ ID	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◇ First	VARCHAR(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◇ MI	CHAR(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◇ Last	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◇ FullName	VARCHAR(40)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◇ Email	VARCHAR(40)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◇ Zipcode	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◇ DateDeleted	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
◇ Deleted	CHAR(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

User		UserType	
ID	AUTO INT	ID	AUTO INT
First	VarChar(15)	Description	VarChar(100)
MI	Char(1)	Deleted	Char(1)
Last	VarChar(25)	Examples: Student, Faculty, Librarian, Guest	
UserTypeID	Int		
FullName	VarChar(40)		
Email	VarChar(40)		
Zipcode	Char(5)		
DateDeleted	Date		
Deleted	Char(1)		

1 - Table User Layout Matches Our CD Design.

We Need
100,000 User Records
For A Final Application
Check-Out

1] Execute 100,000 Queries → One At A Time

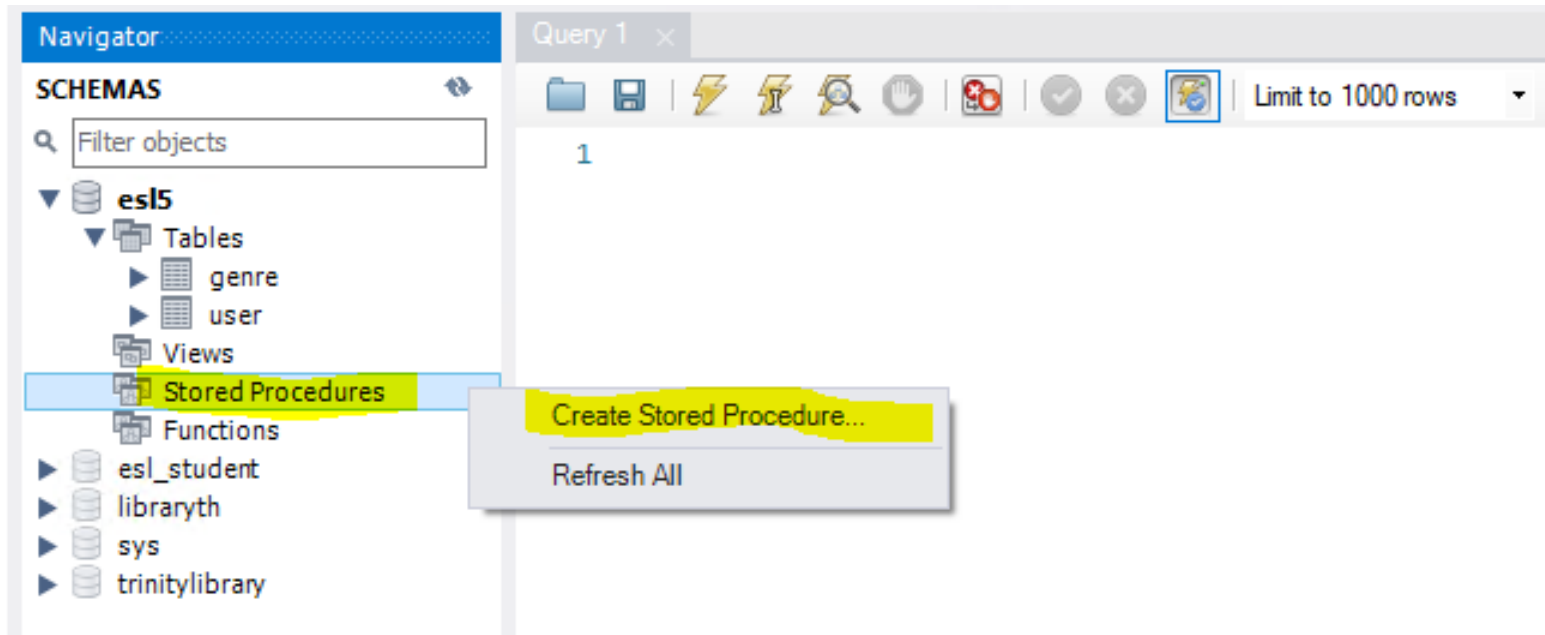
```
INSERT INTO User (ID, First, MI, Last, UserTypeID, Email,  
                FullName, Zipcode, FullName,  
                Deleted, Dated)  
VALUES ( ...
```

**Maybe Do It With
Program Code?**

Stored
Program
DisplayGenre()

Create Stored Program DisplayGenre

1 - Right-Mouse Click On **Stored Procedure** → **Select Create**



2 - Name The Procedure → **'DisplayGenre'**

```
1 ● CREATE PROCEDURE `new_procedure` ()  
2 ○ BEGIN  
3 |  
4 |  
4 | END
```

```
1 ● CREATE PROCEDURE `DisplayGenre` ()  
2 ○ BEGIN  
3 |  
4 |  
4 | END
```

Add The Code To Display All Of Genre In Alpha Order

```
1 ● CREATE PROCEDURE `DisplayGenre` ()  
2 ○ BEGIN  
3     SELECT *  
4     FROM Genre  
5     ORDER BY Genre;  
6 END
```

1] Push Apply Button

2] Note The Wrapper That Workbench Puts Around Our Procedure.

```
1 USE `esl5th` ;  
2 DROP procedure IF EXISTS `DisplayGenre` ;  
3  
4 DELIMITER $$  
5 USE `esl5th` $$  
6 CREATE PROCEDURE `DisplayGenre` ()  
7 ○ BEGIN  
8     SELECT *  
9     FROM Genre  
10    ORDER BY Genre;  
11 END $$  
12  
13 DELIMITER ;
```

Execute The Procedure

```
1 USE ESL5TH;
2 • Call DisplayGenre();
```

2] You Can Pass Arguments To The Procedure.

	id	genre	deleted
▶	2	-- Select Genre --	F
	1	--- All Genre ---	F
	3	Adventure	F
	4	African American Interest	F
	5	Animals	F
	6	Anthology	F
	7	Art	F
	8	Art History	F
	183	Artificial intelligence	F
	9	Autobiography	F
	10	Bible	F
	11	Bible Studies	F
	12	Biography	F
	13	Careers	F
	14	Childcare	F
	15	Children's	F
	17	Children's 10-12	F

Command Line Execution

```
mysql> use es15th;
Database changed
mysql> call DisplayGenre();
```

id	genre	deleted
2	-- Select Genre --	F
1	--- All Genre ---	F
3	Adventure	F
4	African American Interest	F
5	Animals	F
6	Anthology	F
7	Art	F
8	Art History	F
183	Artificial intelligence	F
9	Autobiography	F
10	Bible	F
11	Bible Studies	F
12	Biography	F
13	Careers	F
14	Childcare	F
15	Children's	F
17	Children's 10-12	F

1] You Can Do All Of This
From The Command
Line

Stored
Program
DisplayNGenre(15)

Add The Code To Display N Genre In Alpha Order

1] Let's Make Sure We Can Pass Information Into A Stored Procedure.

```
1 • CREATE PROCEDURE `DisplayNGenre` (IN NoToDisplay INT )
2 BEGIN
3     SELECT NoToDisplay;
4 END
```

2] Call DisplayNGenre(15);

```
mysql> call DisplayNGenre(15);
+-----+
| NoToDisplay |
+-----+
|          15 |
+-----+
1 row in set (0.00 sec)
```

3] If **NoToDisplay = 15** → This Stored Procedure Is To **Display All Of The Information About The First 15 Undeleted Genre In Alpha Order.**

WRITE THE CODE!

What New Thing Do We Need To Know How To Do?

1] MySQL Equivalent Of Form Loop

```
1 • CREATE PROCEDURE `DisplayNGenre` (IN NoToDisplay INT )
2 BEGIN
3     DECLARE Counter INT Default 1 ;
4     loop1: LOOP
5         SELECT Counter;
6         SET Counter = Counter + 1;
7         IF (Counter > NoToDisplay) THEN
8             LEAVE loop1;
9         END IF;
10    END LOOP loop1;
11    SELECT NoToDisplay;
12 END
```

```
mysql> call DisplayNGenre(4);
+-----+
| Counter |
+-----+
|      2 |
+-----+
1 row in set (0.00 sec)

+-----+
| Counter |
+-----+
|      3 |
+-----+
1 row in set (0.01 sec)

+-----+
| Counter |
+-----+
|      4 |
+-----+
1 row in set (0.01 sec)

+-----+
| NoToDisplay |
+-----+
|          4 |
+-----+
1 row in set (0.01 sec)
```

2] Call DisplayNGenre(4);

Stored
Program
CreateUsers(15)

1] Set The IDs = 1, 2, 3, ... NoUsers

```
1 • CREATE PROCEDURE `CreateUsers` (IN NoUsers INT)
2 BEGIN
3 DECLARE Counter INT Default 1 ;
4     loop1: LOOP
5         INSERT INTO User(ID)
6         VALUES (Counter);
7         SET Counter = Counter + 1;
8         IF (Counter > NoUsers) THEN
9             LEAVE loop1;
10        END IF;
11    END LOOP loop1;
12 END
```

2] Lots Of Solutions

3] Call CreateUsers (10);

1] Call CreateUsers (10);

```
mysql> call CreateUsers(10);  
Query OK, 1 row affected (0.10 sec)
```

```
mysql> Select * From User;
```

ID	First	MI	Last	FullName	Email	Zipcode	DateDeleted	Deleted
1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
10 rows in set (0.00 sec)
```

2] We Will Handle First, MI, Last, etc Next Time. We Will Create Our DataFile One Field At A Time.

WHAT DO YOU THINK OF THIS?

Run Procedure CreateUsers Again

1] Call CreateUsers (10);

```
mysql> Select * From User;
```

ID	First	MI	Last	FullName	Email	Zipcode	DateDeleted	Deleted
1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

20 rows in set (0.00 sec)

2] WHAT DO YOU THINK OF THIS?

How Can We Avoid The Duplicates?

1] Automate → Delete From User

```
1 • CREATE PROCEDURE `CreateUsers` (IN NoUsers INT)
2 BEGIN
3 DECLARE Counter INT Default 1 ;
4 DELETE FROM User;
5     loop1: LOOP
6         INSERT INTO User(ID)
7         VALUES (Counter);
8         SET Counter = Counter + 1;
9         IF (Counter > NoUsers) THEN
10             LEAVE loop1;
11         END IF;
12     END LOOP loop1;
13 END
```

2] Lots Of Solutions

3] Call CreateUsers (10);

Backup Your Database ESL5TH

INCLUDE YOUR
STORED PROCEDURES!

Workbench → Server → Data Export

Query 1 Administration - Data Export x

Localhost
Data Export Advanced Options...

Object Selection Export Progress

Tables to Export

Exp...	Schema
<input checked="" type="checkbox"/>	esl5th
<input type="checkbox"/>	esl_student
<input type="checkbox"/>	libraryth
<input type="checkbox"/>	sys
<input type="checkbox"/>	trinitylibrary

Exp... Schema Objects

Refresh Dump Structure and Dat v Select Views Select Tables Unselect All

Objects to Export

Dump Stored Procedures and Functions Dump Events Dump Triggers

Export Options

Export to Dump Project Folder C:\Users\thicks\Documents\dumps\Dump20191229 ...

Each table will be exported into a separate file. This allows a selective restore, but may be slower.

Export to Self-Contained File C:\Temp\Ten Minute Design\ESL5TH.sql ...

All selected database objects will be exported into a single, self-contained file.

Create Dump in a Single Transaction (self-contained file only) Include Create Schema

Press [Start Export] to start... Start Export

**TO BE
CONTINUED...→**