

INFO20003 Week 11 Lab

Database Administration

Introduction

In this lab, you will use MySQL Workbench to backup and restore schemas and tables in the MySQL database. This lab has three sections, in the first you will perform a logical backup of a schema and an individual table. In the second section you will import a table from an external data source. In the third section you will do a backup and then simulate user error and partial logical recovery of a table.

Goals

By the end of this lab, you should be able to:

- Use MySQL Workbench Data Export and Table Data Export Wizards
- Import tables from a csv file using Table Data Import Wizard
- Restore data after simulating user error using Table Data Import Wizard

Section 1 Logical Exports

There are two types of backup we can perform: physical and logical. A physical backup is an exact copy of the physical file that stores the table and index information including the data dictionary. The exact row location in the physical file does not change.

A logical backup takes the data, structure and meta data information which will recreate the objects in our database including tables, indexes, constraints and views. However, the physical location within the file will be different as the object (table, index) is effectively dropped and recreated. During the life of a database rows are updated and deleted. These changes can result in row storage and index storage becoming fragmented.

TASK 1.1 Open and connect to your Engineering IT MySQL Server using MySQL Workbench

In MySQL Workbench inspect the Management section of the Navigator Window. It provides information about the Server performance and configuration parameters. We will be focussing on the Data Export management tasks.

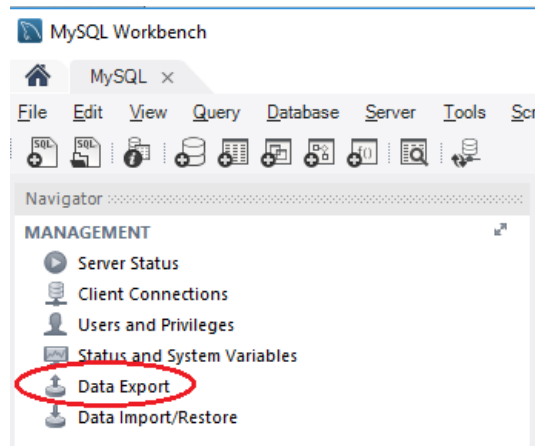


Figure 1: The MySQL Management Navigator. Data Export is circled

TASK 1.2 Click on the Data Export Icon

This will open the Data Export Wizard as shown below:

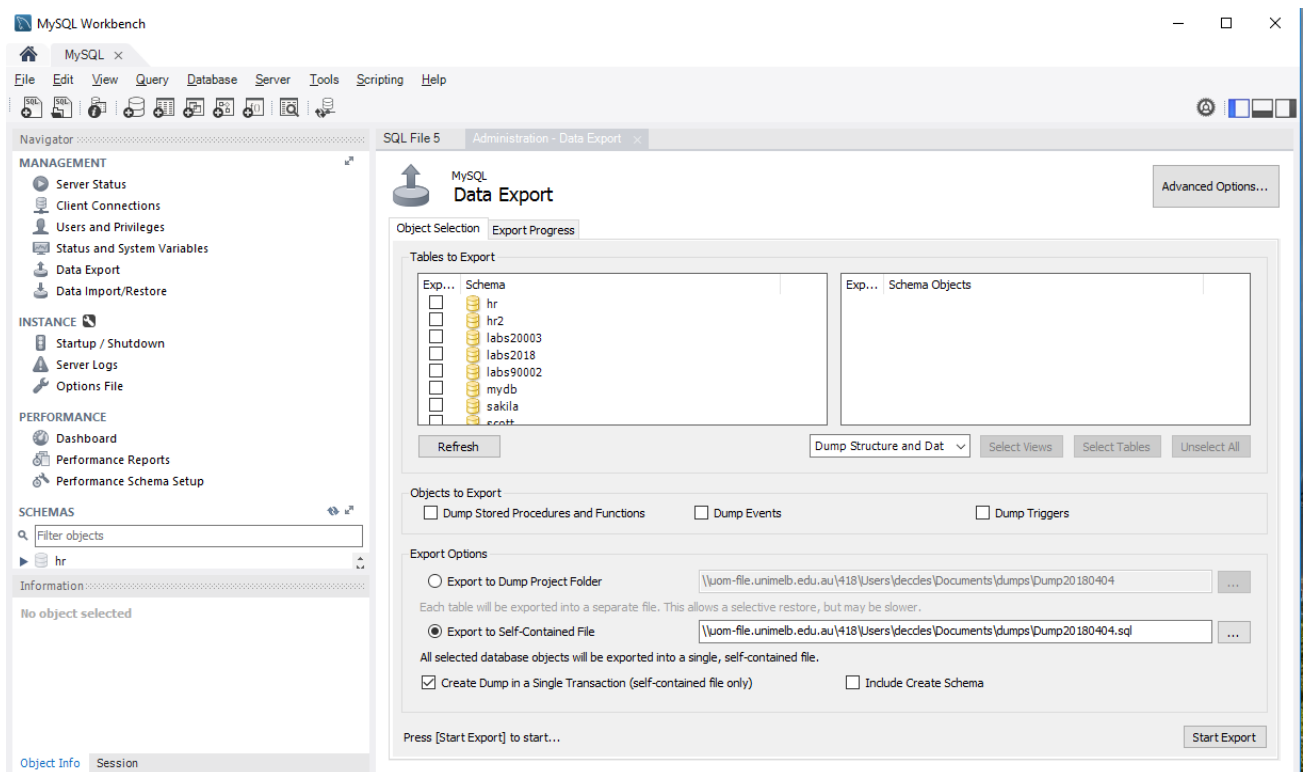


Figure 2: The Schema Data Export Wizard

TASK 2.3 Select your schema for export.

Your schema name should be the same as your username on the Engineering IT server. For the purposes of this lab the screenshots will be selecting labs2018 as the schema to be exported.

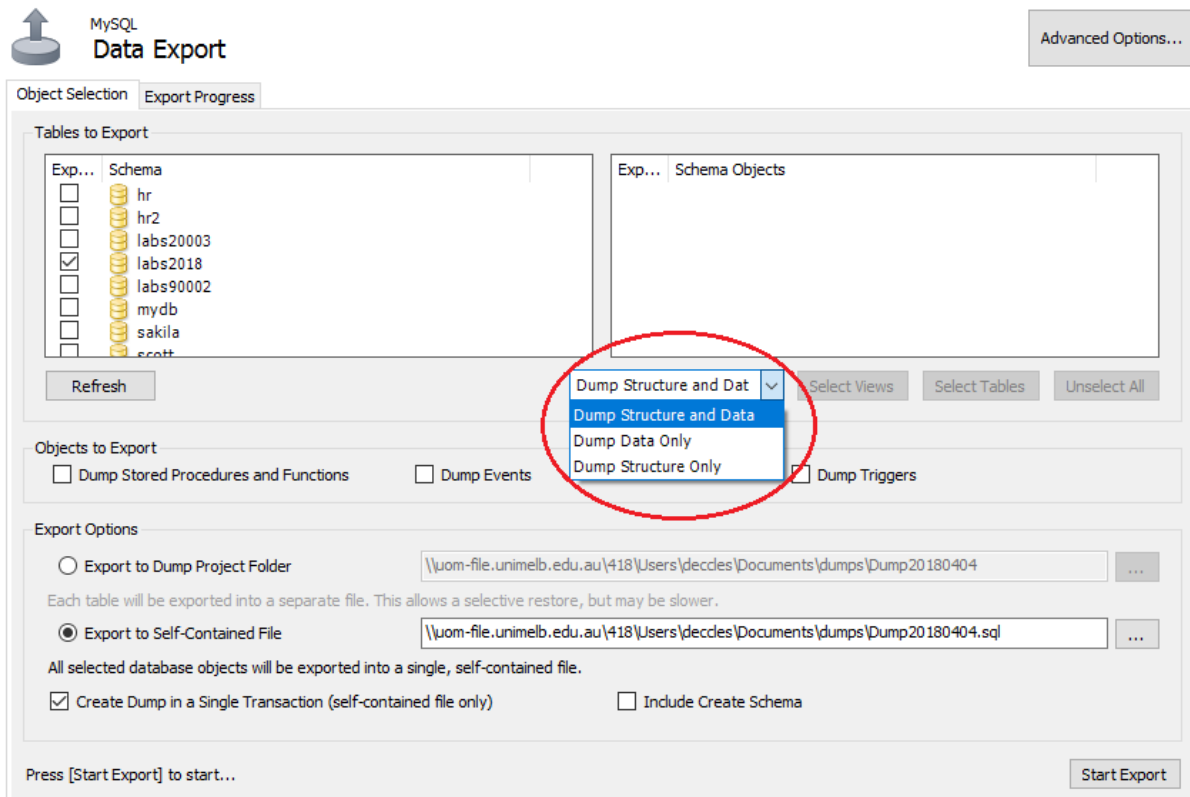


Figure 3 The Data Export Wizard window.

The Data Export Wizard provides many options to export data from the MySQL database server. In Figure 3 we are choosing to take both the structure and data but have options to take the table and index information (including constraint information) without the rows, or only the row data. The export will be to a self contained file which will create the structure and the data.

TASK 1.4 Choose 'Dump Structure and Data' as the export format

TASK 1.5 Change the Export to Self-Contained File, file_name to the schema name (e.g. *labs2018.sql*), and the directory location to your Downloads directory.

TASK 1.6 Click the Start Export button

You may see a warning dialogue window about mysqldump version mismatch. This can be safely ignored. Click 'Continue Anyway'.

The export will run and produce a file in the downloads directory in your schema name.

TASK 1.7 Open the file

When you open the file you will notice more than the table structure and row data. The first is that there are some settings which will enable rows to be imported by disabling the foreign key checks. This is so that the table and data can be entered without checking foreign key constraints. The constraints are enabled after the rows are imported. This is in case there are PK / FK references in the one table such as the relationship of EmployeeID to BossID in the Employee table. Further to this, we create the structures at the end so as to speed up the import.

Table Data Export Wizard

In this section of the lab we will use the MySQL Workbench Table Data Export Wizard to take a logical backup of the Sale table.

TASK 1.8 Open and connect to your Engineering server MySQL Workbench

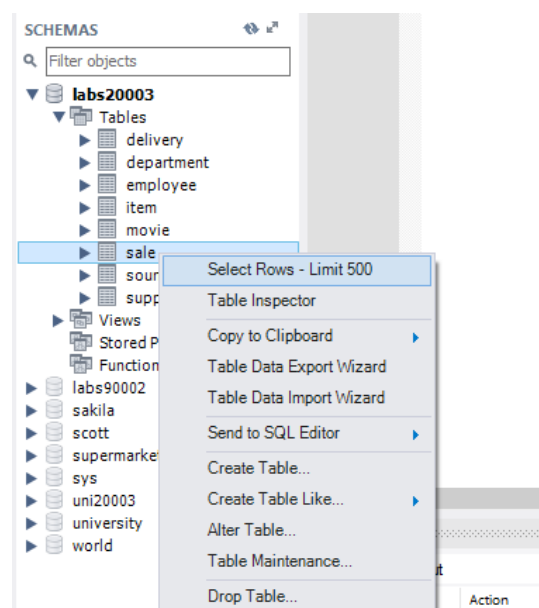
TASK 1.9. Connect to your MySQL Server and execute the following query:

```
SELECT *  
FROM sale;
```

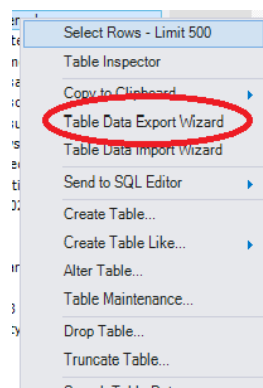
This table, from the department store lab schema, is the table data we will be exporting today. Familiarise yourself with this table.

TASK 1.10. Use the Table Data Export Wizard

In the Navigation Panel on the left hand side of the MySQL Window under the Schema section Right Click on the Sale table.



TASK 1.11 Select Data Export Wizard



TASK 1.12 The Table Data Export Wizard window opens

Table Data Export

Select data for export

Select source table for export:

Select columns you'd like to export

Export	Column name
<input checked="" type="checkbox"/>	SaleID
<input checked="" type="checkbox"/>	DepartmentID
<input checked="" type="checkbox"/>	SaleDate

☒ Select / Deselect all entries

Row Offset: Count:

Advanced >> < Back Next > Cancel

TASK 1.13 Select 'Next' and the file location window appears

Table Data Export

Select output file location

Table Data Export allows you to easily export data into CSV, JSON datafiles.

File Path: Browse...

☒ csv ☐ json

Options:

Line Separator:

Endose Strings in:

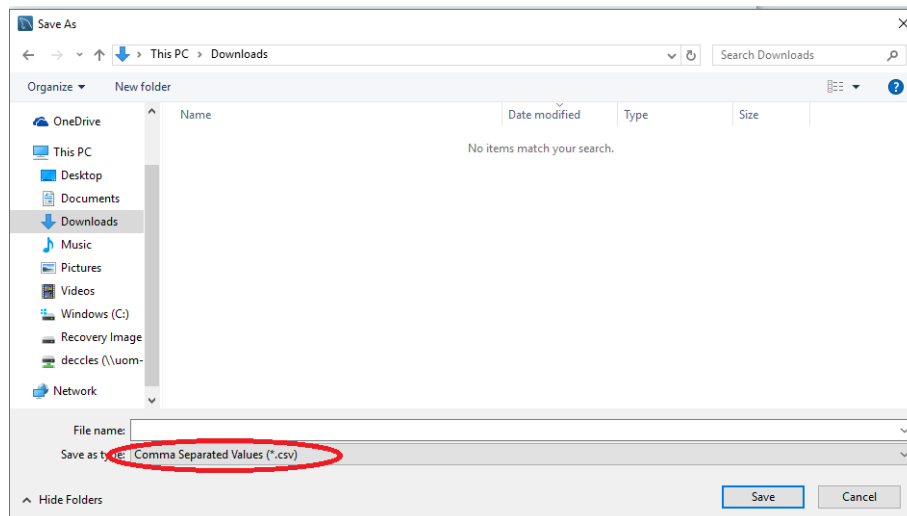
Field Separator:

☒ Export to local machine

If checked, rows will be exported on the location that started Workbench.
 If not checked, rows will be exported on the server.
 If server and computer that started Workbench are different machines, import of that file can be done manual way only.

< Back Next > Cancel

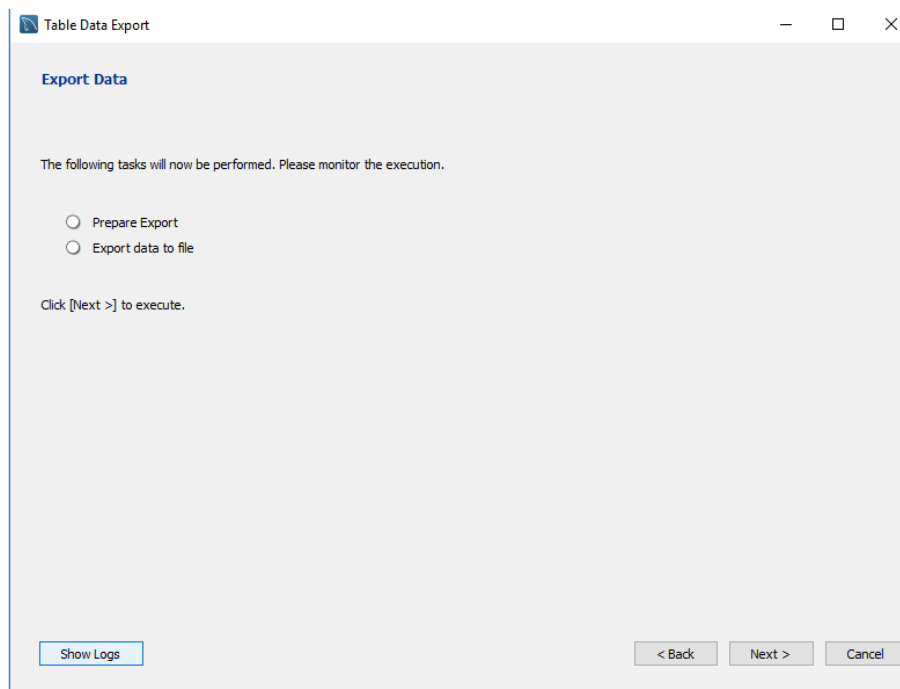
TASK 1.14 Select 'Browse' and nominate the 'Downloads' folder as the location to save the table export



TASK 1.15 Save the file as 'save.csv' in the Downloads folder.

TASK 1.16 The Table Data Exports option window appears and asks you to confirm Line Separator, Enclose Strings and Field Separator. Then select 'Next'.

TASK 1.17 The following dialog window is the data export window listing what will occur when you select the next button. It will prepare the database for export and then export the table data to the nominated file.



Select 'Next'. The table data export will now occur.

Notice that both tasks have now been checked (ticked).

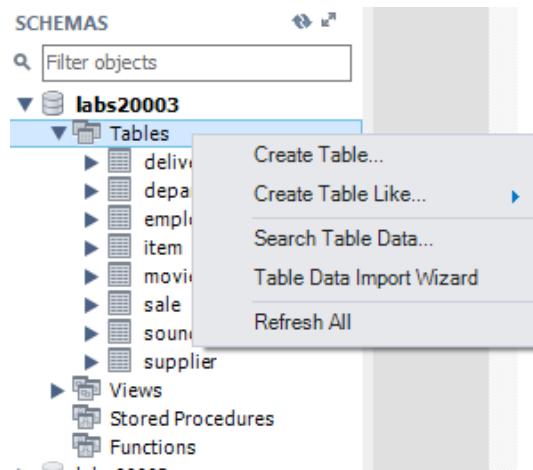
Select 'Next' then 'Finish' to close the Table Data Export Window

TASK 1.18 Using a spreadsheet program, open the *sale.csv* file in the 'downloads' folder. Confirm it has the same details as the Sale table in the Department store schema in the MySQL database. Close the csv file.

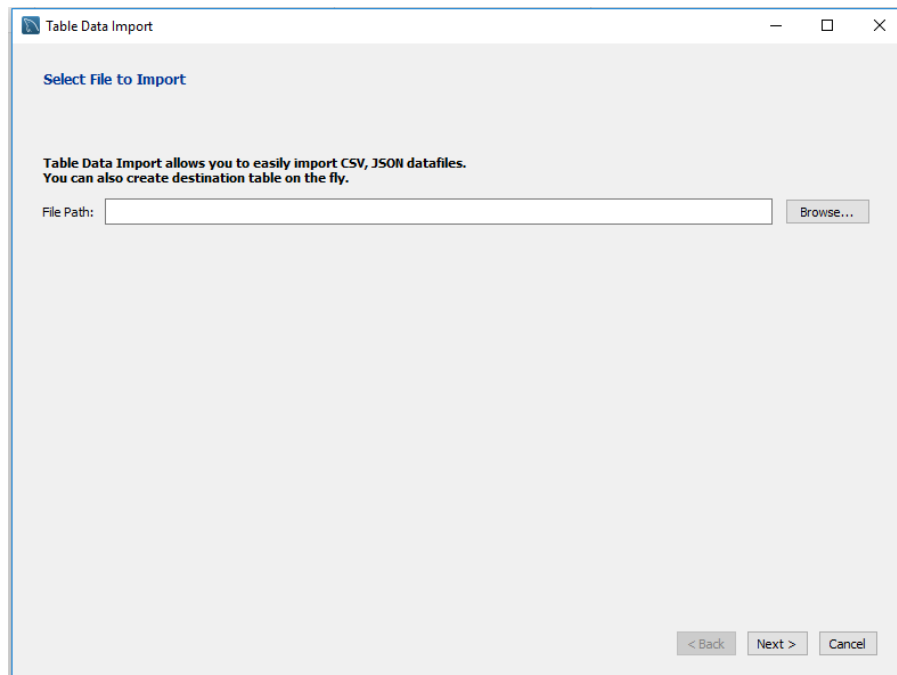
Section 2 Import Tables

TASK 2.1 Download the *PrintMusic.csv* file from the LMS Labs Tab. Save this file to the Downloads folder.

TASK 2.2 In MySQL Workbench Navigation window right click in the word "Tables" in the Schema panel and select "Table Data Import Wizard".



TASK 2.3 Table Data Import Wizard window appears



Click 'Browse' and select the 'PrintMusic.csv' file from the Downloads folder.

Table Data Import

Select File to Import

Table Data Import allows you to easily import CSV, JSON datafiles. You can also create destination table on the fly.

File Path: C:\Users\decdes\Downloads\PrintMusic.csv Browse...

< Back Next > Cancel

Select 'Next'

Table Data Import

Select Destination

Select destination table and additional options.

☐ Use existing table: labs20003.department

☒ Create new table: labs20003 . PrintMusic

☐ Drop table if exists

< Back Next > Cancel

TASK 2.4 Confirm that the Create new table radio button is selected and the new table is in the format <schemaname>.printmusic. Where <schemaname> is your schema name on the Engineering IT server.

TASK 2.5 Confirm the data types and character set encoding in the Confirm Configuration window.

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

Source Column	Field Type
<input checked="" type="checkbox"/> ScoreID	int
<input checked="" type="checkbox"/> Category	text
<input checked="" type="checkbox"/> Composer	text
<input checked="" type="checkbox"/> Title	text
<input checked="" type="checkbox"/> Ochestration	text

ScoreID	Category	Composer	Title	Ochestration	PublisherID
1	Classical	Mozart	Ave Verum...	Chorus and...	3
2	Romantic	Beethoven	Symphony...	Orchestra	3
3	Baroque	Handel	The Messiah	Chorus and...	3
4	20th Century	Britten	Serenade f...	Tenor Voic...	1
5	Second Vie...	Berg	Five Pieces...	Orchestra	1

< Back Next > Cancel

When importing tables MySQL makes assumptions about the text encoding (utf8 in this example) and the data types.

TASK 2.6 Accept the defaults to complete the table import:

Select 'Next' to confirm this configuration.

Select 'Next' to perform the import.

Select 'Next' to confirm the import has succeeded.

Select 'Finish' to close the Table Data Import Wizard.

TASK 2.7 Confirm the data and table are now in your schema by entering the following SQL:

```
SELECT *
FROM printmusic;
```

	ScoreID	Category	Composer	Title	Ochestration	PublisherID
	1	Classical	Mozart	Ave Verum Chorus	Chorus and Oqaan	3
	2	Romantic	Beethoven	Symphony No 3 Eroica	Orchestra	3
	3	Baroque	Handel	The Messiah	Chorus and Orchestra	3
	4	20th Centurv	Britten	Serenade for Tenor Horn and Strinas	Tenor Voice French Horn and Strina Orch...	1
	5	Second Viennese School	Berg	Five Pieces for Orchestra	Orchestra	1
	6	Classical	Haydn	Op72 No2 Emperor	String Quartet	3
	7	20th Centurv	Stravinsky	The Firebird	Orchestra	1
	8	20th Centurv	Strauss Richard	Four Last Songs	Soprano and Orchestra	4

Note that the SQL Output window (the bottom frame of MySQL Workbench) reports that MySQLTable Import wizard did many things as part of this table import:

#	Time	Action	Message
27	14:20:26	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
28	14:20:26	SHOW DATABASES	OK
29	14:20:32	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
30	14:20:32	SHOW COLUMNS FROM 'abs20003'. 'printmusic'	OK
31	14:20:40	PREPARE stmt FROM 'INSERT INTO 'abs20003'. 'printmusic' ('Category','ScoreID','Ochestration','Title','Composer','PublisherID') VALUES(?,?,?,?,,?)'	OK
32	14:20:40	DEALLOCATE PREPARE stmt	OK

Section 3. Restore table data

TASK 3.1 Take a table export of the printmusic table. Save the file as *printmusic2.csv* in the Downloads directory.

TASK 3.2 Simulate user error by using the DDL command TRUNCATE TABLE and remove the data from the table printmusic:

```
TRUNCATE TABLE printmusic;
```

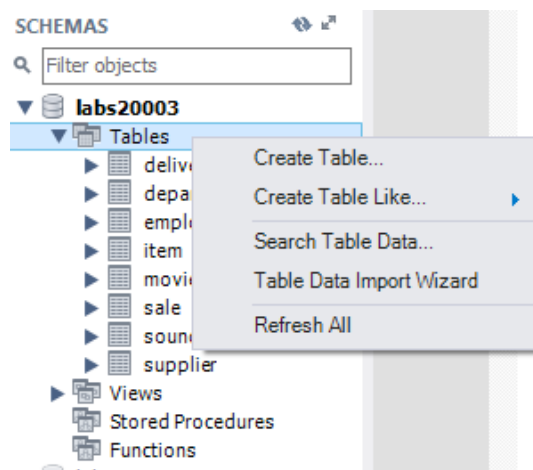
This SQL statement will remove all rows from the PrintMusic table but keep the table structure.

TASK 3.2 Confirm there are no rows in the printmusic table:

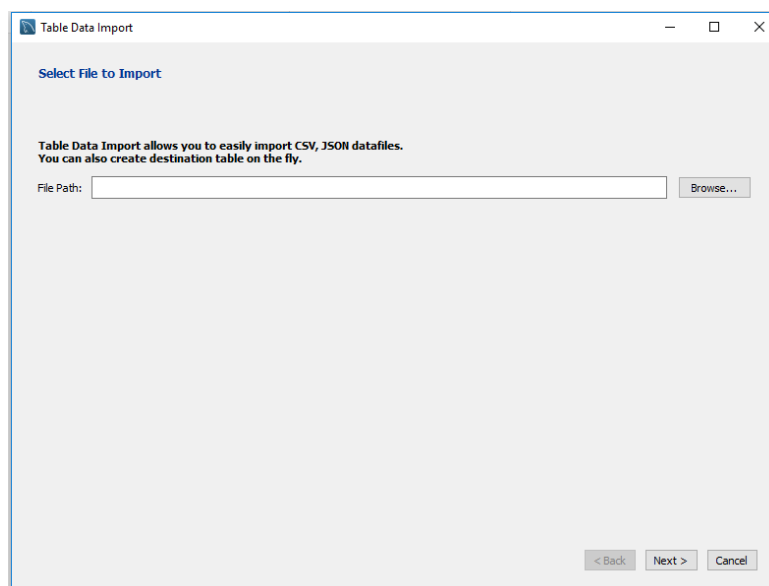
```
SELECT *  
FROM printmusic;
```

Because we have the *printmusic2.csv* file containing the rows of the printmusic table we can restore this table using the Table Data Import Wizard.

TASK 3.3 In MySQL Workbench Navigation Panel Right click the word "Tables" in the Schema panel and select "Table Data Import Wizard".



The Table Data Import Wizard window appears



Click 'Browse' and select the '*PrintMusic2.csv*' file from the Downloads folder.

The screenshot shows the 'Table Data Import' dialog box with the 'Select File to Import' step. The text reads: 'Table Data Import allows you to easily import CSV, JSON datafiles. You can also create destination table on the fly.' Below this, the 'File Path' is set to 'C:\Users\decodes\Downloads\PrintMusic.csv'. A 'Browse...' button is to the right of the path. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Select 'Next'

The screenshot shows the 'Table Data Import' dialog box with the 'Select Destination' step. The text reads: 'Select destination table and additional options.' There are two radio buttons: 'Use existing table:' (which is selected and circled in red) and 'Create new table:'. The 'Use existing table:' option has a dropdown menu showing 'labs20003.printmusic'. The 'Create new table:' option has a dropdown menu showing 'labs20003' and a text field showing 'PrintMusic'. There is also a checkbox for 'Truncate table before import' which is unchecked. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Confirm that the 'Use existing table' radio button is selected and the table is <schemaname>.printmusic.

Confirm the data types and character set encoding in the Confirm Configuration window.

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

Source Column	Field Type
<input checked="" type="checkbox"/> ScoreID	int
<input checked="" type="checkbox"/> Category	text
<input checked="" type="checkbox"/> Composer	text
<input checked="" type="checkbox"/> Title	text
<input checked="" type="checkbox"/> Ochestration	text

ScoreID	Category	Composer	Title	Ochestration	PublisherID
1	Classical	Mozart	Ave Verum...	Chorus and...	3
2	Romantic	Beethoven	Symphony...	Orchestra	3
3	Baroque	Handel	The Messiah	Chorus and...	3
4	20th Century	Britten	Serenade f...	Tenor Voic...	1
5	Second Vie...	Berg	Five Pieces...	Orchestra	1

< Back Next > Cancel

TASK 3.4 Accept the defaults and complete the restore of the row data:

Select 'Next' to confirm this configuration.

Select 'Next' to perform the restore of the truncated rows

Select 'Next' to confirm the import has succeeded.

Select 'Finish' to close the Table Data Import Wizard.

TASK 3.5 Confirm the rows have been restored to the printmusic table:

SELECT *
FROM printmusic;

	ScoreID	Category	Composer	Title	Ochestration	PublisherID
	1	Classical	Mozart	Ave Verum Chorus	Chorus and Oorgan	3
	2	Romantic	Beethoven	Symphony No 3 Eroica	Orchestra	3
	3	Baroque	Handel	The Messiah	Chorus and Orchestra	3
	4	20th Centurv	Britten	Serenade for Tenor Horn and Strinas	Tenor Voice French Horn and Strinda Orch...	1
	5	Second Viennese School	Berg	Five Pieces for Orchestra	Orchestra	1
	6	Classical	Haydn	Op72 No2 Emperor	Strinda Ouartet	3
	7	20th Centurv	Stravinsky	The Firebird	Orchestra	1
	8	20th Centurv	Strauss Richard	Four Last Songs	Soprano and Orchestra	4

END OF LAB 10