

CS2209 INFORMATION STORAGE AND MANAGEMENT II

Lab 01: Indexing and Trees

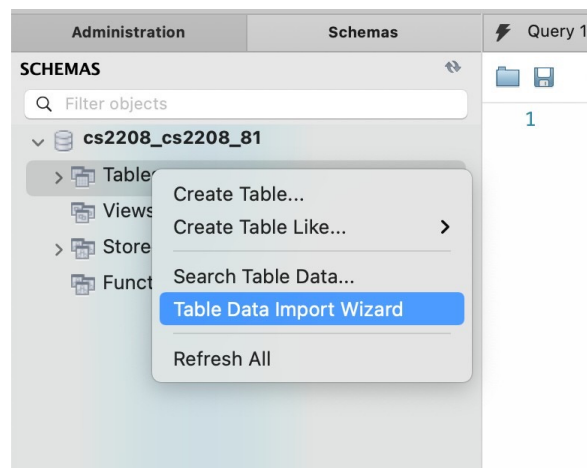
1. Indexing

In this section, we will work with a relatively large file to see how we can use the MySQL Workbench application to load the file. We will then use the EXPLAIN statement to understand the benefits of creating an index.

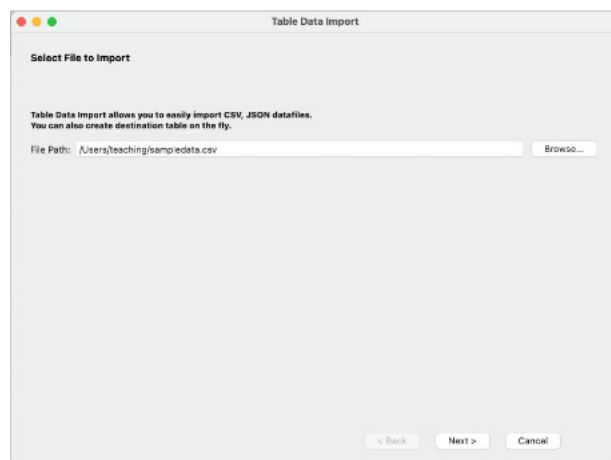
1.a. Importing data

Use MySQL Workbench and perform the following steps:

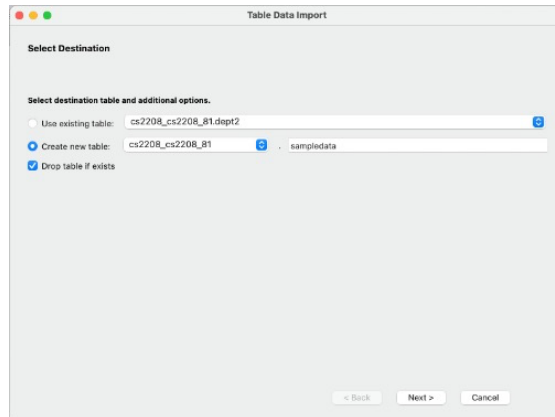
- Download the sampledata.csv file from canvas
- Log onto MySQL Workbench using your CS MySQL credentials.
- Create a table using the sampledata.csv file and the 'Table Data Import Wizard'. You'll find this by right-clicking on Tables under the Schema tab.



- Select the file and click next



- The next window determines where the file's data will upload to. We are going to create a new table - it will be part of the current schema and the default name is the name of the data file. We also choose the drop the table if it already exists. The other option is to select an existing table - this would have to have the same schema as the file.



Select Destination

Select destination table and additional options.

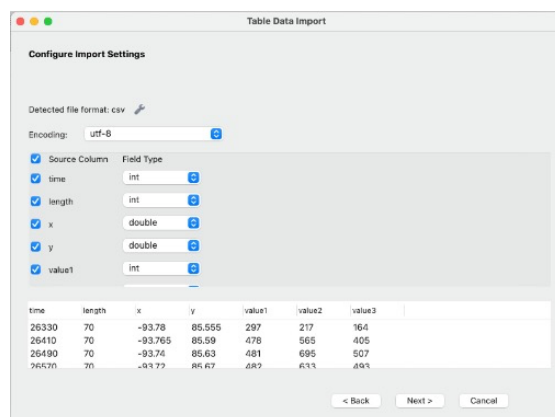
☐ Use existing table: cs2208_cs2208_R1.dept2

☒ Create new table: cs2208_cs2208_R1 . sampledata

☒ Drop table if exists

< Back Next > Cancel

- Since we are creating a new file from the sample data file, we can configure the table create as we wish. As it is, default values will be chosen from the file.



Configure Import Settings

Detected file format: csv

Encoding: utf-8

Source Column	Field Type
<input checked="" type="checkbox"/> time	int
<input checked="" type="checkbox"/> length	int
<input checked="" type="checkbox"/> x	double
<input checked="" type="checkbox"/> y	double
<input checked="" type="checkbox"/> value1	int

time	length	x	y	value1	value2	value3
26330	70	-93.78	85.555	297	217	164
26410	70	-93.765	85.59	478	565	405
26490	70	-93.74	85.63	481	695	507
26570	70	-93.72	85.67	482	811	591

< Back Next > Cancel

- Finally, we set to create the new table and import the data. This can take a little while to execute. You may have to refresh the table list (right click on tables and select from drop down).



Import Data

The following tasks will now be performed. Please monitor the execution.

- Prepare Import
- Import data file

Click [Next >] to execute.

Show Logs < Back Next > Cancel

1.b. Explain statements

Once we have imported the file, we can run a query against this. Run the following:

```
SELECT * FROM sampledata;
```

Note the execution time of the query.

Now execute:

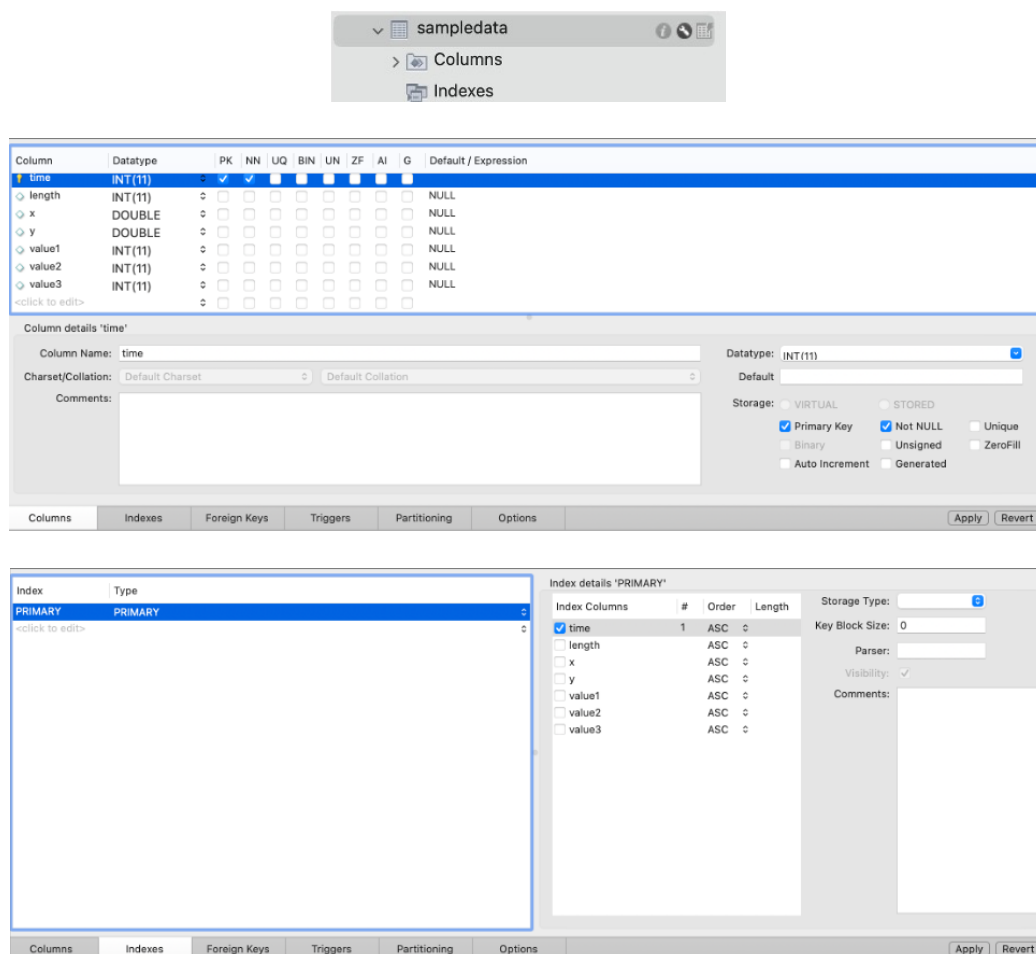
```
SELECT * FROM sampledata WHERE time=26490;
```

Again, note the execution time of the query.

Now rerun the last query but with 'explain'. This will produce a set of outputs that includes type and rows inspected by the query:

```
EXPLAIN SELECT * FROM sampledata WHERE time=26490;
```

Now add an index on time. This can be added by editing the table to make time a primary key or by explicitly creating an index. We can edit the table structure by clicking on the spanner icon and select PK next to the time column name (remember to press apply). If we now go into indexes tab in the same window we can see a new primary index added.



The screenshot shows the 'sampledata' table structure in a database management tool. The table has columns: time (INT(11)), length (INT(11)), x (DOUBLE), y (DOUBLE), value1 (INT(11)), value2 (INT(11)), and value3 (INT(11)). The 'time' column is marked as the Primary Key (PK).

Below the table structure, the 'Column details 'time'' section shows the column name 'time', datatype 'INT(11)', and various options like 'Primary Key', 'Not NULL', 'Unique', 'Unsigned', 'ZeroFill', 'Auto Increment', and 'Generated'. The 'Primary Key' checkbox is checked.

The 'Indexes' tab is selected, showing a list of indexes. A new primary index named 'PRIMARY' is listed with the column 'time' as the index column.

The 'Index details 'PRIMARY'' section shows the index columns, their order, and length. The 'time' column is the first and only column in the index, with order 1 and length ASC.

We could also create an index using SQL:

```
CREATE INDEX sampleindex ON sampledata(time);
```

Where *sampleindex* is the name of the index, *sampledata* is the table and *time* is the column in the table that we index on.

Rerun our query and note the execution time. Then run explain on the same query - note the change to the number of rows examined.

2. Trees

2.a. B-Tree

Construct and draw a B-tree of order 3 ($m=3$) for the following set of key values:

40 23 65 35 55

Assume that the tree is initially empty and values are added in the given order. Draw the tree (either as an image or sketch on paper and submit a photo) and submit via canvas.

2.b. B+-Tree

Construct and draw a B+-tree of order 3 ($m=3$) for the following set of key values:

40 23 65 35 55

Assume that the tree is initially empty and values are added in the given order. Draw the tree (either as an image or sketch on paper and submit photo) and submit via canvas.