Hands-on 7: Import data into the Yum Juice table

Overview

In this exercise you will import data from a text file into the Juice table of your Yum database.

Before starting you need to have created the Juice table (Hands-on 6) and checked that you have the fields and data types specified in that exercise.

If you have not already created the Juice table you can create it using the sample script. If you do this you should still review Hands-on 6 at a later time to ensure you understand the material it covers.

When you have completed this exercise you will be able to:

- Identify the absolute path for a file stored in your work area.

- Import data from a text file into a database table.

- Enter records into a table using AUTO_INCREMENT and ENUM data types with default values

- Describe how MySQL uses AUTO_INCREMENT to generate new values.

Step-by-step

1: Download the Yum Juice Data File

> **Notes:**
>
> You used the INSERT INTO statement to enter the data into the Customer table in Hands-on 4. As you have details of all the Juices in the sample data, you could take the same approach to enter the juice data. However, with details of 20 items to enter this would be a rather time-consuming and error-prone process.
>
> MySQL offers the facility to import data into a table from a text file. This means that data already stored in another electronic format can easily be imported into a MySQL database

Create a folder named YumData in your work folder for this module. Download the juiceData.zip file. Unzip and save the juice.txt text file in the folder you have just created.

Once you have saved the file you need to make a note of its absolute path as you will need to use this for the data import command in the next section.

The path for the file shown in Fig. 1 is: . Z:/Teaching/IDT/YumData/juiceData.txt. Note this is the path that will be used in the command in this exercise but *the path for the file you have stored in your workspace will be different*.

Open Windows Explorer and make a note of the path for your copy of the juice data file.

It will probably be something like: I:/DT/YumData/juiceData.txt but it will depend on your folder structure.

> **Note:**
>
> You need to use forward slashes / to separate the parts of the path or it will not be recognised by the MySQL processor.
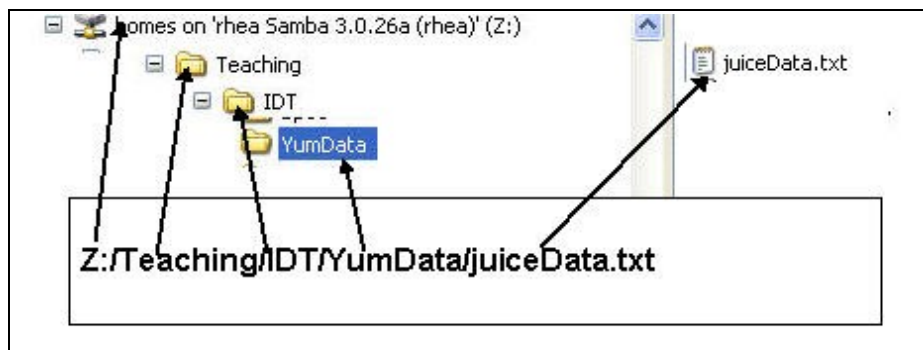

Fig. 1: Absolute path for juice data file

2: View the Yum Juice data file

MySQL can import data in a number of different formats. In this case we are importing from a *tab-separated* text file. Each record is stored on a new line of the file and within each record the fields are separated by a tab character.

To view the structure of the file, open it in Microsoft Word. Click the Show/Hide button on the toolbar to show the non-printing characters. The file will appear as shown in Fig.2 with each record ending in a paragraph marker ¶ and each field within the record separated by a tab marker →.
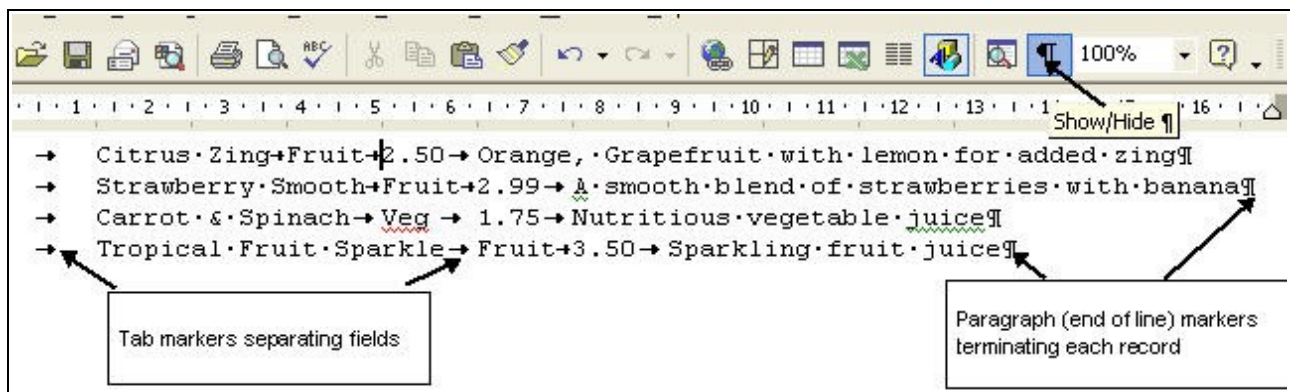

Fig. 2: Viewing Juice data file in Microsoft Word

The data import process will load the data on each line from left to right into the fields of a new record in a specified table. The order of the data on each line corresponds to the order of the fields in each record. Each line of the Juice data file starts with a tab (empty field) as MySQL will automatically assign the next value in the sequence to the auto-increment juice_id primary key field.

> **Note:**

DO NOT MODIFY THIS FILE IN ANY WAY OR THE DATA IMPORT PROCESS MAY NOT WORK CORRECTLY.

(If you do inadvertently modify the file you should download another copy of the original.)

Close the file before importing the data in the next step.

3: Import the data into the Yum Juice table

Note:

Before importing the data ensure that you check that your table structure has been created correctly (see Hands-on 6). If the table structure is incorrect you may encounter errors importing the data.

Open the Query Browser with your database as the default. (Review Hands-on 5 if you're not sure how to do this.)

Create a new script to import the data containing the following command (replacing *filename* with the absolute path for your data file enclosed in double quotes):

```
LOAD DATA LOCAL INFILE "filename"

INTO TABLE Juice

FIELDS TERMINATED BY '\t'

LINES TERMINATED BY '\r\n';
```

Note:

You can enter the command as a query or as a script but if you use a script you can easily save the command to re-use later.

Understand the command

Line 1 is the command to instruct MySQL to import data from the specified file. It includes the word LOCAL as you are importing data from a file stored on your drive not on the MySQL server.

Line 2 defines the table to be inserted into.

Line 3 tells MySQL what character separates the fields in each line of the file in this case a tab '\t'.

Line 4 tells MySQL how the lines in the file are terminated. The characters inserted to mark the end of line vary depending on the operating system used to create the file. In this case, as the file was created in Windows format, you need to specify two characters – carriage-return ('\r') and line-feed ('\n').

Once you have entered the command, press the Execute button to run it and import the data into your table.

4: View the data in the Juice table

Once you have successfully executed the import command, view the data in the Query

Browser to check that it has been entered correctly. (You should know how to do this by now.)

Your data should appear as shown in Fig. 3.

juice_id	juice_name	juice_type	juice_price	juice_description
1	Citrus Zing	Fruit	2.50	Orange, Grapefruit with lemon for added...
2	Strawberry Smooth	Fruit	2.99	A smooth blend of strawberries with bana...
3	Carrot & Spinach	Veg	1.75	Nutritious vegetable juice
4	Tropical Fruit Sparkle	Fruit	3.50	Sparkling fruit juice
5	Fruit Punch	Fruit	3.50	Refreshing fruit mix with strawberry chunks
6	Summer Shake	Fruit	3.25	Apricot, pineapple & strawberry shake
7	Nothing but Apple	Fruit	1.95	Clear apple juice
8	Orchard Fruits	Fruit	2.50	Thick blended apple and pear juice
9	Smoothie Surprise	Fruit	3.25	Four-fruit smoothie
10	Apple & Kiwi	Fruit	2.75	Apples with kiwis
11	Beet Route	Veg	2.75	Liquid beauty treatment
12	Green Giant	Veg	2.50	Super vegetable combo
13	Very Berry Sparkler	Fruit	2.50	Sparkling summer fruit blend
14	Citrus Classic	Fruit	3.25	Orange, lemon, lime, grapefruit plus kumq...
15	Tang	Fruit	2.75	Blended tropical fruits
16	Saprkling Sunshine	Fruit	3.25	A blend of grape and pomegranate juice
17	Bright eyes	Veg	1.75	Pure carrot juice
18	Tomato plus	Veg	1.75	Tomatoes with celery
19	Berry Blue	Fruit	2.50	Pure blueberry juice
20	Summer Cooler	Fruit	2.50	Summer fruits with mint

Fig. 3: Viewing Juice data in Query Browser

If the data has not been imported correctly, you need to check the reason for this and make any necessary corrections. If you are not sure what has caused the problem, check with your tutor before proceeding further.

5: Explore data entry

Now enter data for more juices to familiarise yourself with the data entry process.

> **Note:**
>
> You can either enter the juice details given in the examples or be creative and make up your own!

5.1: Enter data for a new fruit juice

Assuming that the new juice to be entered is a fruit juice, you only need to supply data for the juice_name and juice_price fields.

The juice_id field will automatically be assigned the next value in the sequence as it has been set to auto_increment. The default value of 'Fruit' will be entered into the juice_type field unless the user specifies the alternative value of 'Veg'. The juice_description field allows NULL values so it is not necessary to enter data for this field.

To see how this works, enter details for a new juice using the following INSERT command:

```
INSERT INTO Juice (juice_name, juice_price)
VALUES
('Fruit Fizz', 2.75)
```

View the data in the table again and you should see the new record as shown in Fig. 4.

Fig. 4: Data entry using auto_increment, default and NULL values

5.2: Enter data for a new vegetable juice

Modify the INSERT command you used above to enter data for a new vegetable juice with juice_name 'Vegetable Surprise' and juice_price £1.99.

View the data in the table to check that the new juice has been entered correctly (Fig. 5).

Fig. 5: New vegetable juice entered into table

5.3: Experiment with auto_increment

Modify the previous INSERT command to include the juice_id field and enter data for a new fruit juice with an id that already exists in the table. For example you could enter 'Fruit Heaven', price £2.50 , id 5.

When you run this command you get an error message "Duplicate entry '5' for key 1" as you are trying to enter a duplicate value for the primary key of the table which, as you should know, is not allowed.

Change the id value to 0 (zero) and run the command again.

This time the command will run successfully. You will notice that the new record has been added with the juice_id set to the next value in the sequence. (Fig. 6).

Fig. 6: Entering 0 as value for auto_increment field assigns next value in the sequence

Now try explicitly entering NULL as the value for the auto_increment juice_id field by entering a record for another fruit juice, 'Fruit Bliss'. Again you will see that this has the effect of assigning the next value in the sequence.

Next delete the record that you have just entered by running the following command:

```
DELETE
FROM Juice
WHERE juice_id = 24
```

View the data again to see that the record has been removed from the table.

Now re-insert the record for 'Fruit Bliss' juice. (You can either enter this without a value for the juice_id or give the juice_id field a value of 0 or NULL as you did above.)

View the data again and note that the juice_id value for the record you deleted has not been re-used but the new record has been assigned the next value after that in the sequence (Fig. 7).

| 23 | Fruit Heaven | Fruit | 2.50 | NULL |
| 25 | Fruit Bliss | Fruit | 2.50 | NULL |

Gap in sequence shows that auto_increment value for the deleted field has not been re-used

Fig. 7: Auto_increment values are not re-used

Enter another juice record but this time give it a juice_id that is higher than the maximum juice_id in the table - for example, 'Fruit Fool', £2.50, juice_id 30.

View the data again and note that it is possible to assign a specific value to an auto_increment field if it is higher than the maximum value already assigned. The auto_increment value will then continue with the next value after this. To see that this is the case enter another record – for example 'Go Bananas', £2.50 – with no value specified for the juice_id. (See Fig. 8.)

25	Fruit Bliss	Fruit	2.50	NULL
30	Fruit Fool	Fruit	2.50	NULL
31	Go Bananas	Fruit	2.50	NULL

Auto_increment sequence continues after highest value already in table

Fig. 8: Restarting the auto_increment numbering at a higher value

Finally, try entering juices with ids corresponding to the missing numbers in the sequence. For example, 'Peach Delight', juice_id 26 (this would have been the next number generated if you had not specifically entered the 'Fruit Fool' record with a juice_id of 30) and 'Mango Surprise', juice_id 25 (the same juice_id as previously existed for a record you deleted).

View the data again and note that both these records have been successfully entered into the table (Fig. 9).

24	Mango Surprise	Fruit	2.50	NULL
25	Fruit Bliss	Fruit	2.50	NULL
26	Peach Delight	Fruit	2.50	NULL
30	Fruit Fool	Fruit	2.50	NULL
31	Go Bananas	Fruit	2.50	NULL

Record entered with same id as one previously deleted

Record entered with id lower than maximum value already in the table

Fig. 9: Entering records with auto_increment values lower than maximum value already in table

> **Note:**
>
> You need to think carefully about the way in which MySQL assigns numbering when you use an auto_increment field. These examples should have helped you to understand this. To find out more, read Section 3.6.9 of the MySQL User Manual.

Summary

Review

Check your understanding of the material covered in this exercise by answering the following questions:

What is the meaning of the keyword LOCAL in the LOAD DATA INFILE command?

What type of data needs to be enclosed in single quote marks when it is inserted?

If you do not specifically supply a value for an ENUM field that has a default value associated with it, what value will be entered into the field?

What happens when you enter a record including a value that already exists in the table for an auto_increment field?

What happens when you enter a record with a value of 0 or NULL for an auto_increment field?

What happens when you enter a record with a value higher than the maximum value already in the table for an auto_increment field?

What happens if you enter a record with a value for an auto_increment field that does not already exist in the table but is lower than the maximum value already stored?