Practical 6: Hands on SQL

Activity1: Create Database and Table using SQL

1. Start the MySQL connection you have created in Practical 4.

2. Create database called “egt206_prac6DB” and select it.

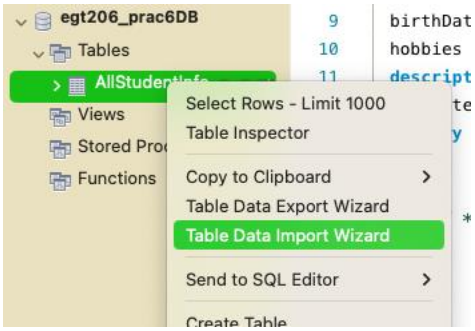
3. Create a table that will hold information of many students from different schools.

```
CREATE TABLE AllStudentInfo
(
        school varchar(4) NOT NULL,
        name varchar(100) NOT NULL,
        gender varchar(1) NOT NULL,
        birthDate date NOT NULL,
        hobbies varchar(100) NULL,
        description varchar(100) NULL,
        collectedBy varchar(10) NOT NULL,
        primary key(school, name, birthDate)
);
```

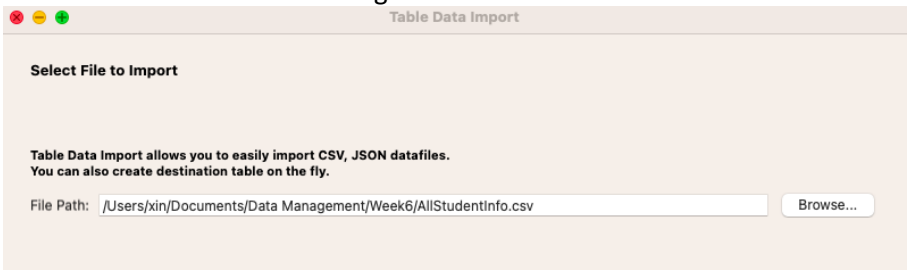
Activity2: Insertion of the data from csv file

1. Go to Brightspace to download the data file AllStudentInfo.csv in a folder of your pc.

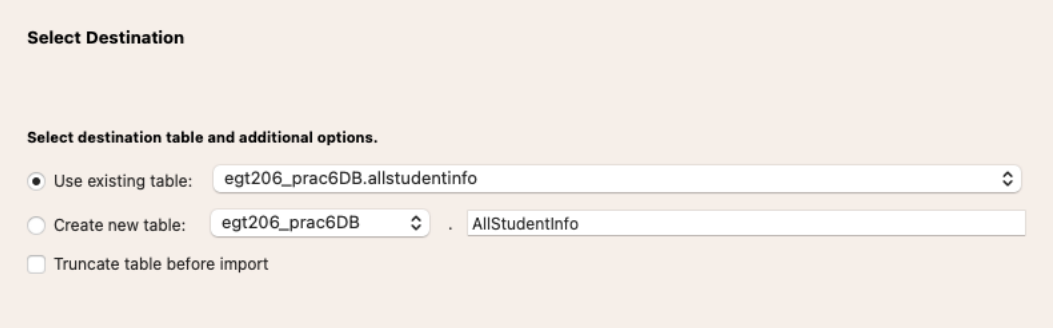
2. Follow the following steps to import the csv file to the table
   a. In schema tab, right click on the table “AllStudentInfo” and select “Table Data Import Wizard”.



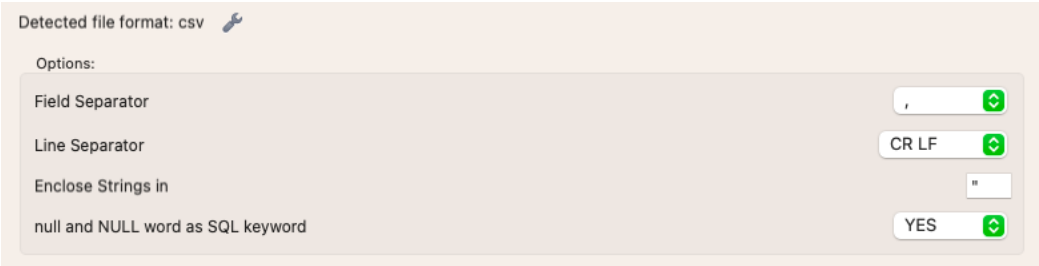
b. Select the file using “Browse” button and click ‘Next’



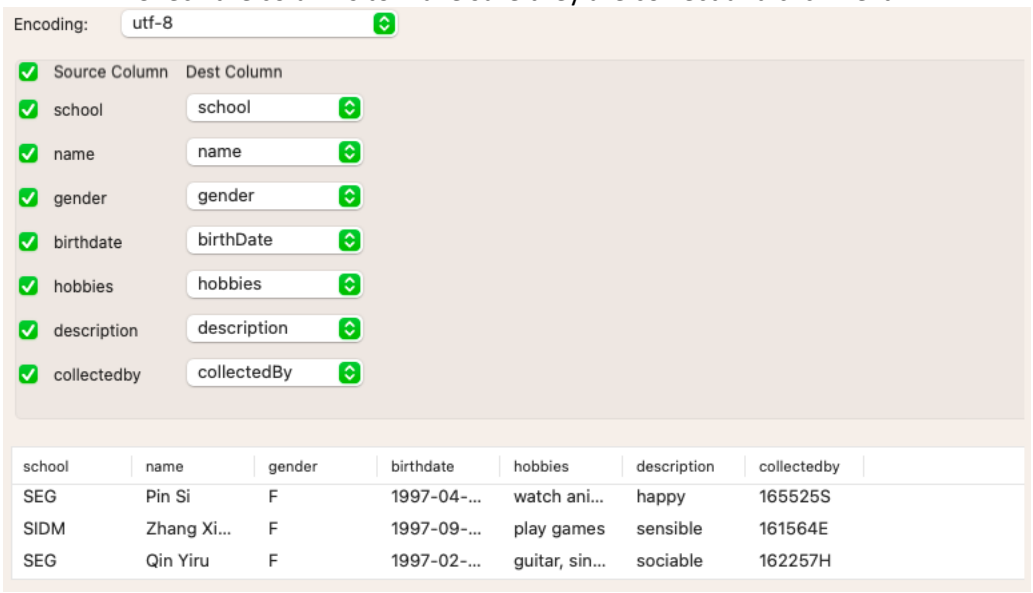
c. Select “Use existing table”



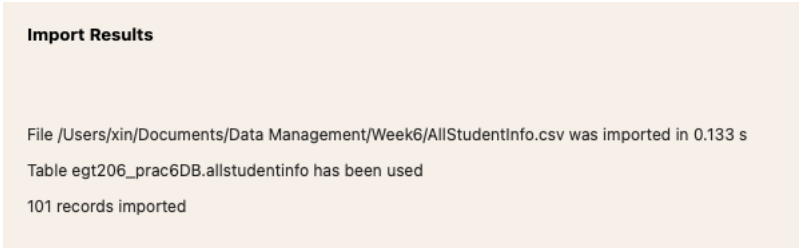
d. Click on the tools icon and change Field Separator and Line Separator to following:



Check the columns to make sure they are correct and click Next.



e. Click Next twice. You should see the following, showing “101 records imported”. Click “Finish” to close the window.

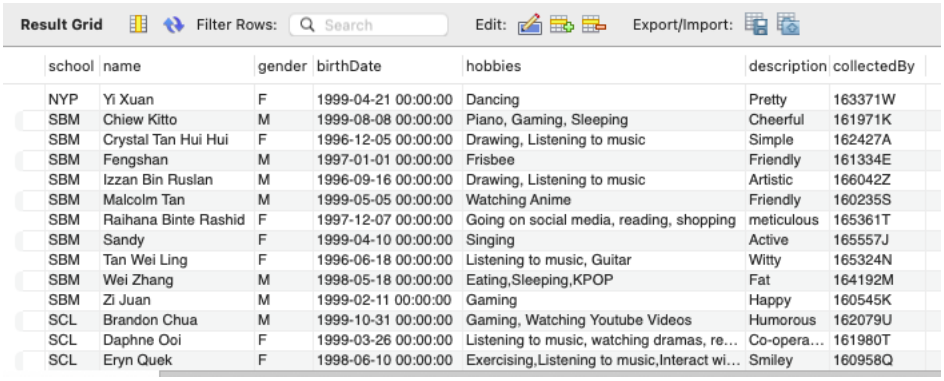


If you see 0 record imported, it means that nothing has been imported. Please go through steps (a to e) to make sure you have done it correctly.

3. Perform a “SELECT” command.

```
SELECT * FROM AllStudentInfo;
```

You should see the table showing up in the Result Grid.



Result Grid | Filter Rows: Search | Edit: | Export/Import:

school	name	gender	birthDate	hobbies	description	collectedBy
NYP	Yi Xuan	F	1999-04-21 00:00:00	Dancing	Pretty	163371W
SBM	Chiew Kitto	M	1999-08-08 00:00:00	Piano, Gaming, Sleeping	Cheerful	161971K
SBM	Crystal Tan Hui Hui	F	1996-12-05 00:00:00	Drawing, Listening to music	Simple	162427A
SBM	Fengshan	M	1997-01-01 00:00:00	Frisbee	Friendly	161334E
SBM	Izzan Bin Ruslan	M	1996-09-16 00:00:00	Drawing, Listening to music	Artistic	166042Z
SBM	Malcolm Tan	M	1999-05-05 00:00:00	Watching Anime	Friendly	160235S
SBM	Raihana Binte Rashid	F	1997-12-07 00:00:00	Going on social media, reading, shopping	meticulous	165361T
SBM	Sandy	F	1999-04-10 00:00:00	Singing	Active	165557J
SBM	Tan Wei Ling	F	1996-06-18 00:00:00	Listening to music, Guitar	Witty	165324N
SBM	Wei Zhang	M	1998-05-18 00:00:00	Eating,Sleeping,KPOP	Fat	164192M
SBM	Zi Juan	M	1999-02-11 00:00:00	Gaming	Happy	160545K
SCL	Brandon Chua	M	1999-10-31 00:00:00	Gaming, Watching Youtube Videos	Humorous	162079U
SCL	Daphne Ooi	F	1999-03-26 00:00:00	Listening to music, watching dramas, re…	Co-opera…	161980T
SCL	Eryn Quek	F	1998-06-10 00:00:00	Exercising,Listening to music,Interact wi…	Smiley	160958Q

Activity3: SELECT only certain columns of the rows

1. To display only a few columns in your query, you can do the following:

```
SELECT school, name, gender FROM AllStudentInfo;
```

2. You can specify the order of the columns, differentfly from that of the order of the columns during table creation.

```
SELECT gender, name, school FROM AllStudentInfo;
```

3. You can give an identifier/alias to the column name to be displayed in the output.

```
SELECT name as StudentName FROM AllStudentInfo;
```

4. You can output unique column values by using the keyword DISTINCT.

```
SELECT DISTINCT(school) FROM AllStudentInfo;
```

How many rows will the above query return, when compared to the following query?

```
SELECT school FROM AllStudentInfo;
```

Activity4: Using the WHERE condition

We can obtain a subset of rows by using the WHERE clause in our SQL statements.

```
SELECT * FROM AllStudentInfo WHERE <conditions>;
```

Operator	Description
=	equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
<> or != or ^=	not equal to

Exercises

1. Display the names, gender, birthdates, hobbies and descriptions of the students from SBM.
2. Display the names, birthdates, hobbies and descriptions of all girls.
3. Display the names, birthdates, hobbies and descriptions of the girls from SIT.
4. Display the names, birthdates, hobbies and descriptions of the guys who are NOT from SEG.
5. Display all information of students who are born in year 1995.
6. Display all information of students who specify 'Gaming' as one of their hobbies.
7. Display all students who are being described as "Friendly', "Cheerful" or "Brave".

```
SELECT name, gender, birthDate, hobbies, description
FROM AllStudentInfo
WHERE school='SBM';
```

```
SELECT name, birthDate, hobbies, description
FROM AllStudentInfo
WHERE gender='F';
```

```
SELECT name, birthDate, hobbies, description
FROM AllStudentInfo
WHERE gender='F' AND school='SIT';
```

```
SELECT name, birthDate, hobbies, description
FROM AllStudentInfo
WHERE gender='M' AND school!='SEG';
```

```
SELECT * FROM AllStudentInfo
WHERE birthDate
BETWEEN '1995-01-01' AND '1995-12-31';
```

```
SELECT * FROM AllStudentInfo
WHERE hobbies
LIKE ('%Gaming%');
```

```
SELECT * FROM AllStudentInfo
WHERE description
IN ('Friendly', 'Cheerful', 'Brave');
```

Activity5: Deletion of records using WHERE

Sometimes, you need to delete existing records. The syntax is as follows:

```
DELETE FROM AllStudentInfo WHERE <conditions>;
```

Exercises

1. Delete all students from SIDM.
2. Delete all girls who are from SHS.

```
DELETE FROM AllStudentInfo
WHERE school = 'SIDM';
```

```
DELETE FROM AllStudentInfo
WHERE gender='F' AND school='SHS';
```

Activity6: Ordering of output

Now we are back to SELECT statements.

Frequently, for display purposes, we would like an ordering of the data based on certain columns. We achieve this by the following syntax:

```
SELECT <column> FROM <Tablename> WHERE <condition>
ORDER BY <columnName>;
```

Exercises

1. SELECT all girls from SEG with their name in alphabetical ascending order.

2. SELECT all students with their school in descending order first, and then name in ascending order.

```
SELECT * FROM AllStudentInfo
WHERE gender='F' AND school='SEG'
ORDER BY name;
```

```
SELECT * FROM AllStudentInfo
ORDER BY school DESC, name;
```

Activity7: GROUP BY

In SELECT queries, a subset of rows of the entire data is returned. Sometimes, we would like to return a value that is aggregated from different groups within the subset of the data returned. Each group would have the same value of the column name specified after the GROUP BY clause.

```
SELECT school, COUNT(*) FROM AllStudentInfo
WHERE gender = 'F'
GROUP BY school;
```

1. Firstly, a subset of data containing all girls are returned (as a result of the condition specified by WHERE gender=’F’).

2. We would like to consider different groups within this subset of data by the column ‘school’. One group would be “SEG”, another group “SIT” and so on.

3. We then perform the COUNT(*) function on each group.