FIT9132 Introduction to Databases

Week 2 Tutorial Activities

FIT Database Teaching Team

Complete the week 2 tutorial activities:

2.1. Software Setup Recap

2.2. Git Setup, Pull and Push

2.3. Using the SQL Developer GUI to Manage Data

2.4. Using SQL Developer Git Client to Revert and Reset Commit

2.5. Using the SQL Command to Retrieve Data

2.6. Using the SQL Developer GUI to Drop a Table

2.7. Relational vs NoSQL Databases Comparison

FIT9132 2021 S2

FIT9132 Introduction to Databases

Author: FIT Database Teaching Team

Learning Objectives

At the completion of the tutorial questions and tasks, you should be able to:

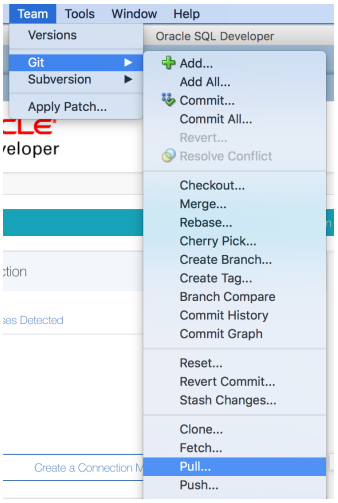
- make use of GitLab and SQL Developer git version control for file management,
- transfer SQL scripts and other files to/from your local hard disk to your local repository,
- load SQL scripts and execute them within SQL Developer,
- use the SQL Developer Graphical User Interface to perform INSERT, UPDATE and DELETE operations, and
- understand that a range of different database types exists.

2.1. Software Setup Recap

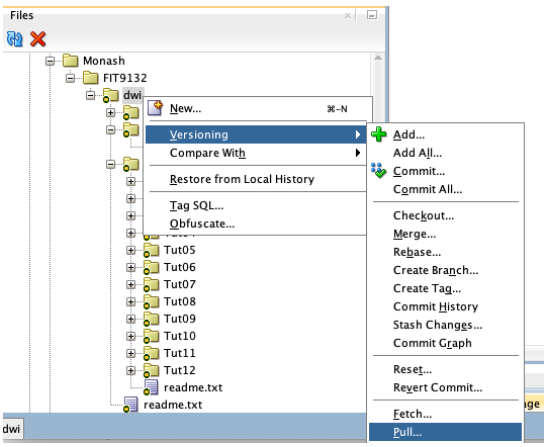
Before starting this week's activities ensure you have installed SQL Developer locally (or know how to open SQL Developer on MoVE), have created a working folder and have cloned your remote repo to this folder. The instructions are provided in the Week 1 tutorial handout. During this tutorial, your tutor may ask you to share your screen and check your SQL Developer set up (including your Oracle connection) and your local repo.

2.2. Git Setup, Pull and Push

Before starting ANY tutorial activity or any working session, first use SQL Developer to pull from the FIT GitLab server to ensure your files on your local repo are in sync with files in the server. On SQL Developer click Team → Git → Pull.

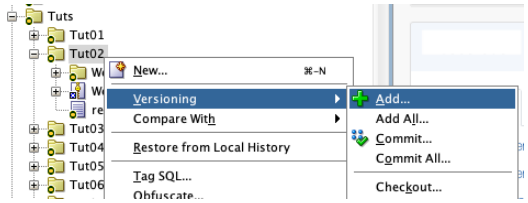


or right click on your working directory folder on the Files window → versioning → pull

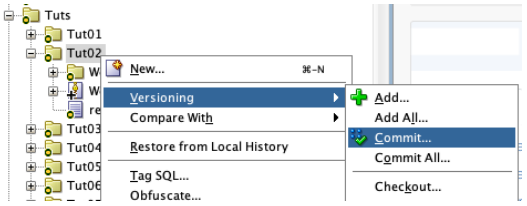


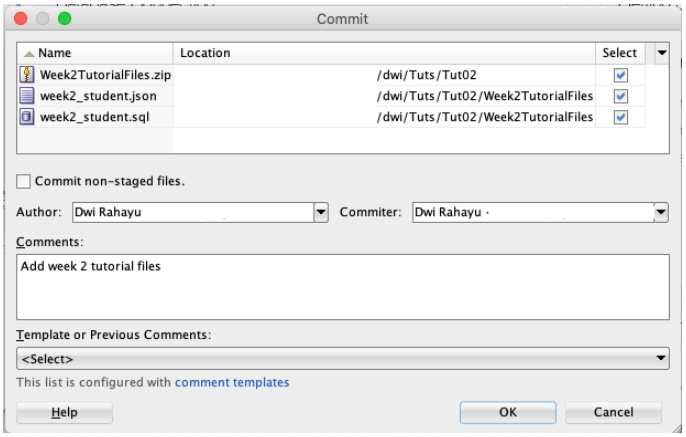
Once all files are in sync. download the Week2TutorialFiles.zip from Moodle, save the file into your local repo (under the Tut02 folder) and **extract** the contents. Before starting the activities below ensure that the two files for this week's activities (week2_student.sql, and week2_student.json) are located in the Tut02 folder of your local repo. If you use MoVE, please follow the video "Copying Files Local to Server" for detailed instructions.

The next step is to ensure that the files are securely backed up in the FIT GitLab server by pushing these files across to the server. Right click on Tut02 folder → Versioning → Add to stage the folder/files.

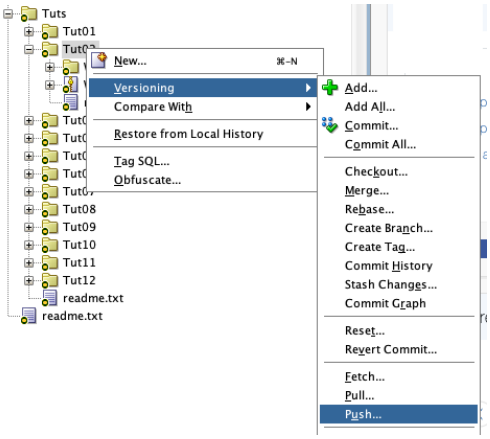


Then, commit the files. Right click on Tut02 folder → Versioning → Commit. Add a meaningful commit comment for future reference.



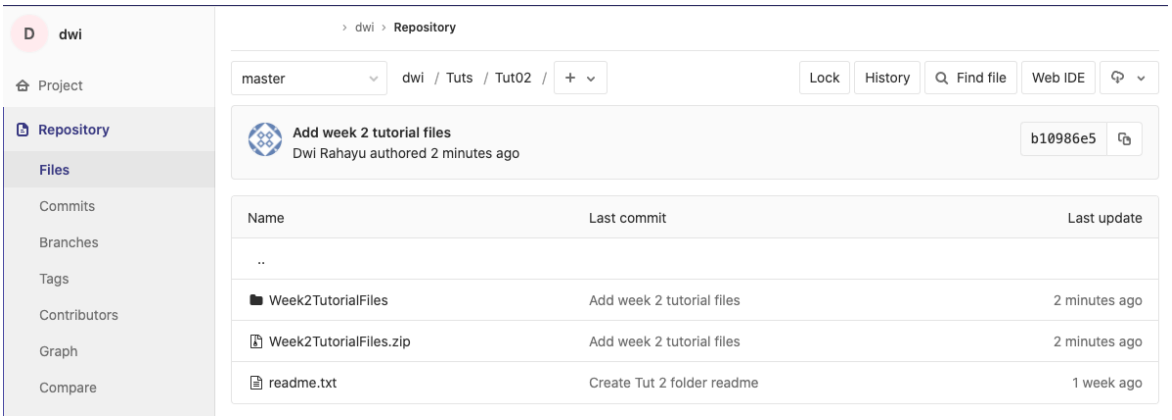


Lastly, push the files across to the server. Right click on Tut02 folder → Versioning → Push.

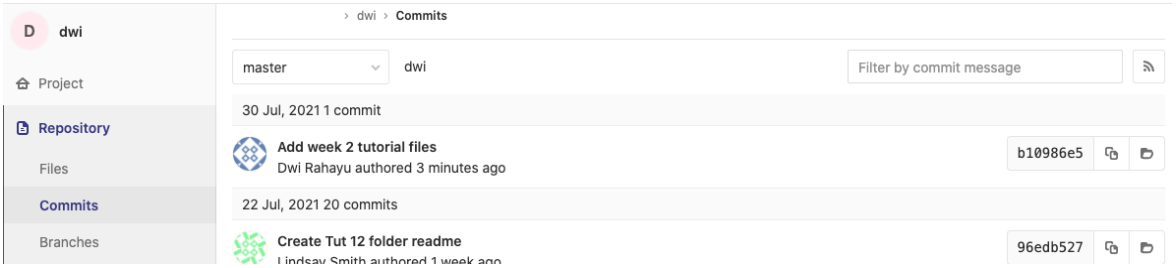


You can also follow the video "AddCommitPush with Git-Part 1" for detailed instructions.

It is crucial to check whether the files have been correctly pushed to the FIT GitLab server. Login to your account on the FIT GitLab server in a web browser, and *view* your files to ensure the files are present and that you have pushed correctly. You should see something like the following:



As well as checking the files in the appropriate folder, you should also check your commits and the commit message you added. Select "Commits" in the left panel:



Please ensure the commit message you add is meaningful and describes the actions you have taken.

A new folder in your local repository can be created using File Explorer or Finder. A new file can also be created using SQL Developer. It is important to note that **file or folder deletion, within your local repo, must be done using SQL Developer** so that the GIT client is aware of the change. Detailed instructions on how to modify/maintain your local repository are provided in "AddCommitPush with Git-Part 2" video.

Your tutor will check your pull, add files (stage), commit and push during this tutorial as you continue with the activities listed below. *As we progress through the semester, you need to regularly login to the FIT GitLab server and make sure that your files are being correctly pushed to the server*. Often students commit and add but forget the vital step of push.

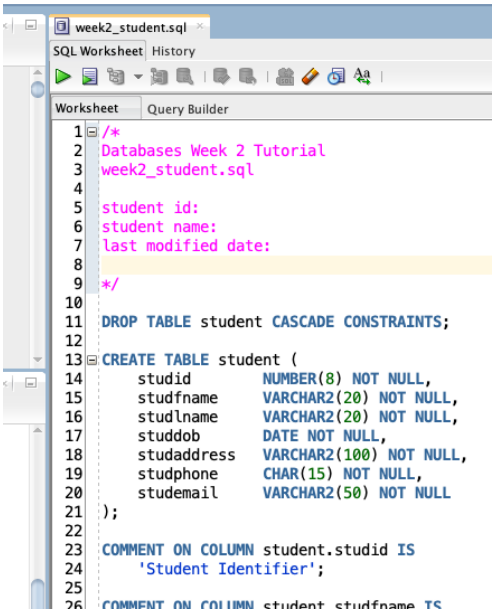
2.3. Using the SQL Developer GUI to Manage Data

There are two main approaches to interact with the Oracle database using SQL Developer, through the Graphical User Interface (GUI) and the SQL Worksheet. In this section, you will learn to use the SQL Worksheet to create a collection of tables and how to use the GUI to add, update and delete data from the tables.

1. Opening an SQL file in the SQL Worksheet.

We will use the SQL Worksheet to create the database by running an SQL script. Follow these steps:

1. Open SQL Developer on your machine.
2. Open the connection to an Oracle server.
3. Click File → Open → .../Tut02/week2_student.sql. You will see the script in the SQL Worksheet area.



4. Add your details at the top of the script:

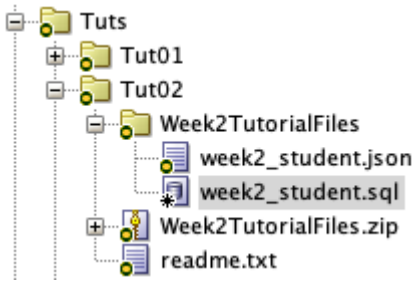
```
/*
Databases Week 2 Tutorial
week2_student.sql

student id: yourStudentID
student name: yourName
last modified date: today'sDate
*/
```

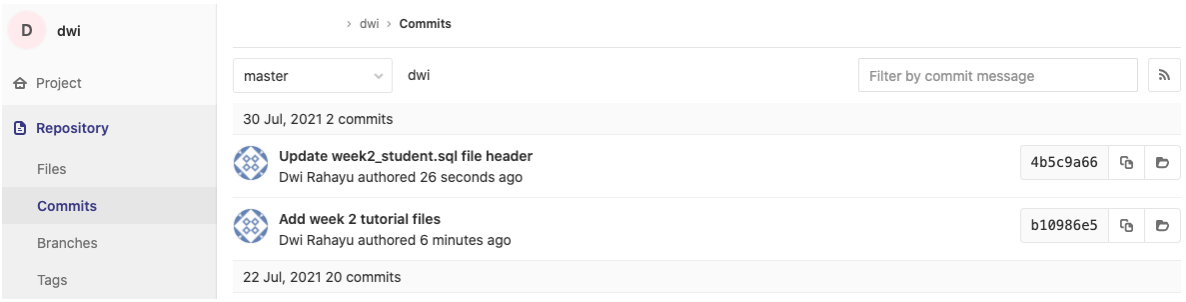
5. Save the script file by clicking the save icon  or pressing ctrl + s buttons. Any file which you modify should be pushed across to Git to record the changes and ensure the file is safely stored.

2. Push the SQL file to the FIT GitLab Server.

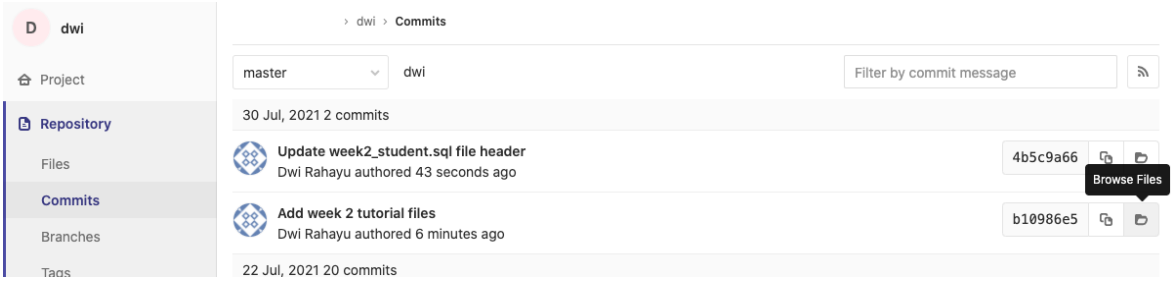
Observe the File window on SQL Developer, especially the Tut02 folder.



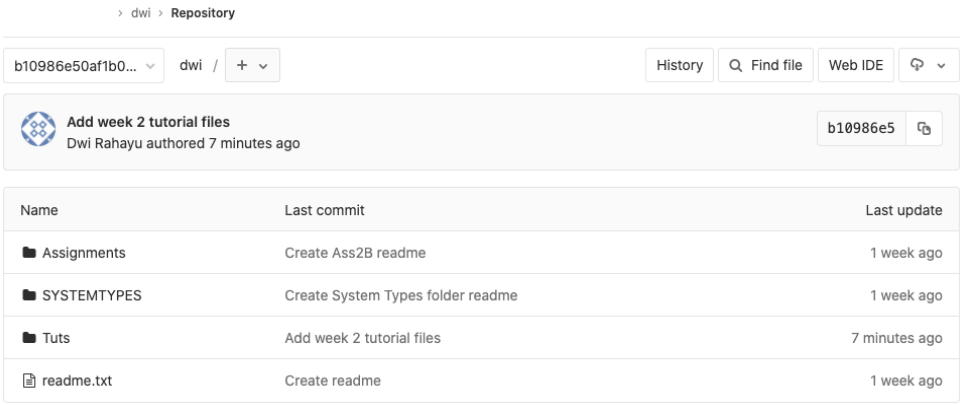
The icon indicates that the file has been modified and has not been staged on Git. Now add, commit and push the file into the FIT GitLab server. Follow the steps discussed in 2.2. Git Setup, Pull and Push. Check if the changes have been pushed to FIT GitLab server.



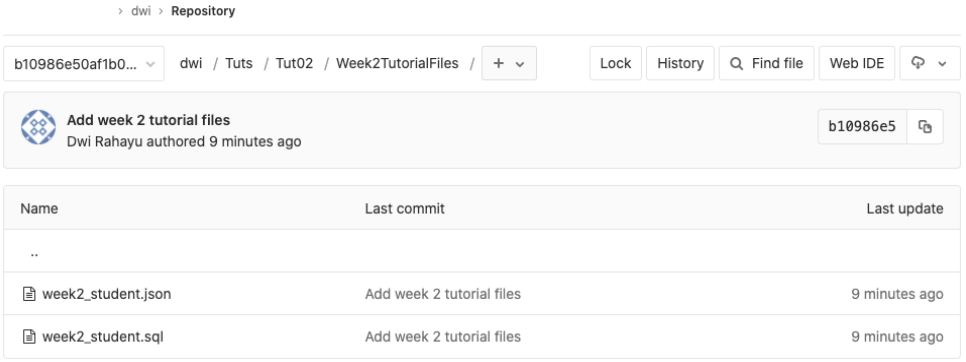
To view the files from a particular commit, click on the folder icon on the right hand side of the commit list:



You will then be presented with the files from this commit:



Not the commit SHA (or "hash") in the top right, this is the commit identifier - here b10986e50af1b03101b13b6eaddfc340ab21ec56 (this is used by the Git system to identify a particular commit). If you navigate to the Tuts folder and Tut02 you can access the files as they were committed:



If the file is a plain text file, as these files are, clicking on the file will show the contents of the file. The previous version represented here can be downloaded via the download icon in the top right of the display.

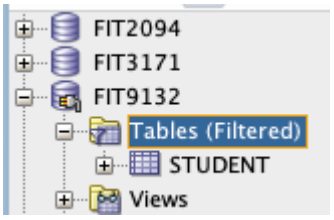
You can clearly see that Git manages files (creates versions) which means ***you do not, and should not, give different names such as week2_student_v1.sql, week2_student_v2.sql etc.*** Doing so in a commercial situation makes it very difficult for your fellow developers to track changes to the files since they will use Git to check versions.

Check your file versions in Git

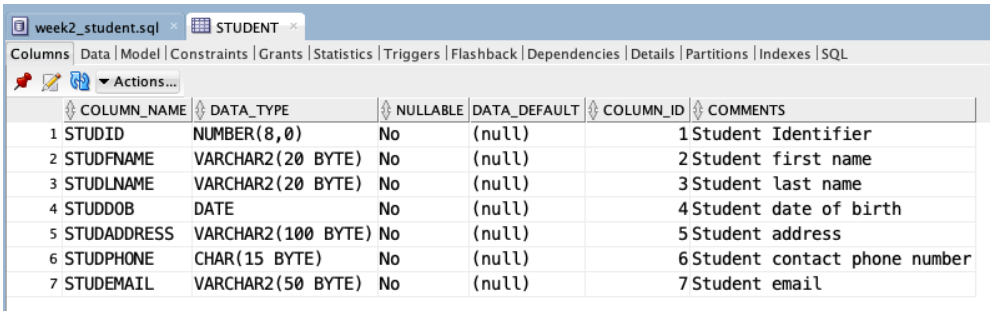
Use the web interface of Git to check the two versions of week2_student.sql that you have pushed to Git - the original version with no header, and the updated version with a completed header.

3. Viewing the table structure and the data of a relational database table.

1. In the SQL Developer worksheet area, run the SQL script (week2_student.sql) by pressing the "run script" icon. 
2. To view the table using the graphical user interface, expand the Table option in the Connection tab and find "STUDENT" from the list, then double click on the "STUDENT".

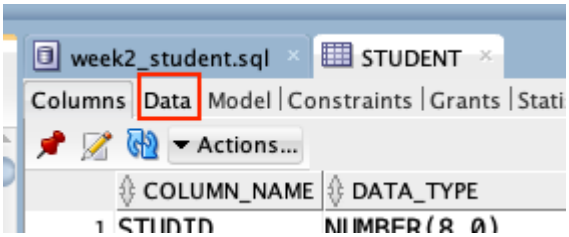


3. You will see the listing of the database structure of the "STUDENT" table.

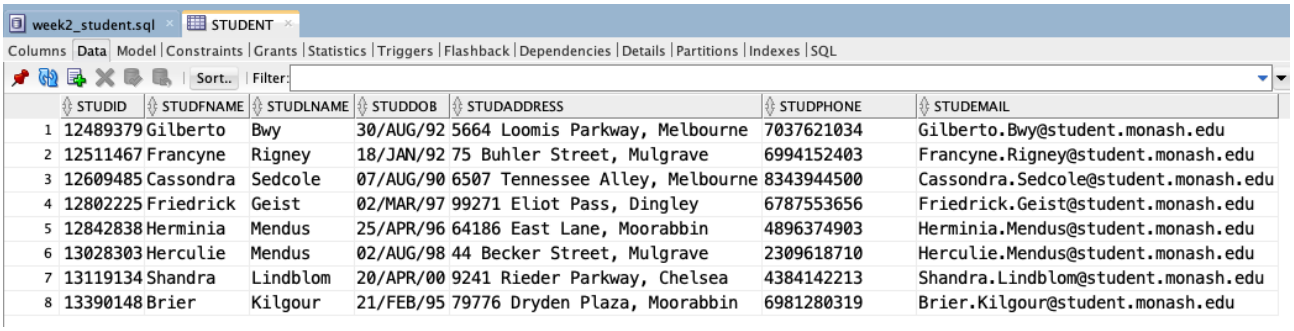


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	STUDID	NUMBER(8,0)	No	(null)	1	Student Identifier
2	STUDFNAME	VARCHAR2(20 BYTE)	No	(null)	2	Student first name
3	STUDLNAME	VARCHAR2(20 BYTE)	No	(null)	3	Student last name
4	STUDDOB	DATE	No	(null)	4	Student date of birth
5	STUDADDRESS	VARCHAR2(100 BYTE)	No	(null)	5	Student address
6	STUDPHONE	CHAR(15 BYTE)	No	(null)	6	Student contact phone number
7	STUDEMAIL	VARCHAR2(50 BYTE)	No	(null)	7	Student email

4. To view and change the data inside the STUDENT, click on the "Data" tab on the right-hand panel:

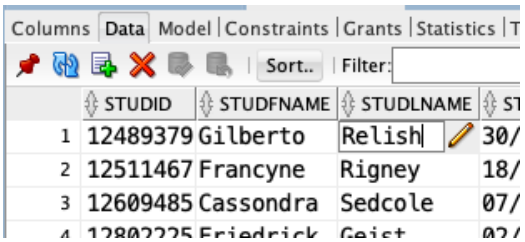


5. You will see the data listing inside the "STUDENT":

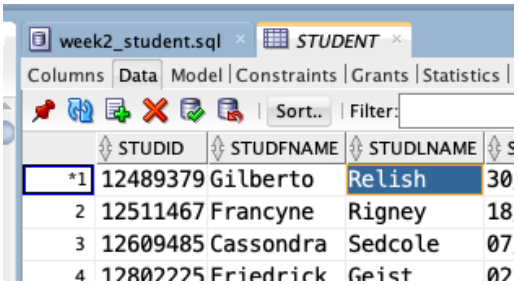


4. Updating the data inside a table.

1. Double click on the cell where the data needs to be changed. For example, let's change Gilberto's (row 1) last name (studlname) from "Bwy" to "Relish" and press enter.



2. You will see an asterisk for the first row.

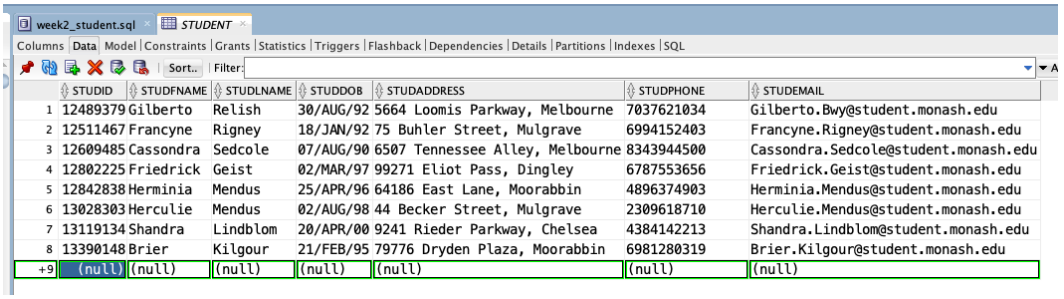


3. To accept the changes made on the row(s) with an asterisk, you will need to issue a COMMIT by pressing the "tick"  icon.

4. When the database accepts the COMMIT instruction you have made, the asterisk should disappear from row 1.

5. Adding a new row to the table.

1. Click on the insert new row icon. 
2. You will see a new row is being added to the table with all values listed as (null).



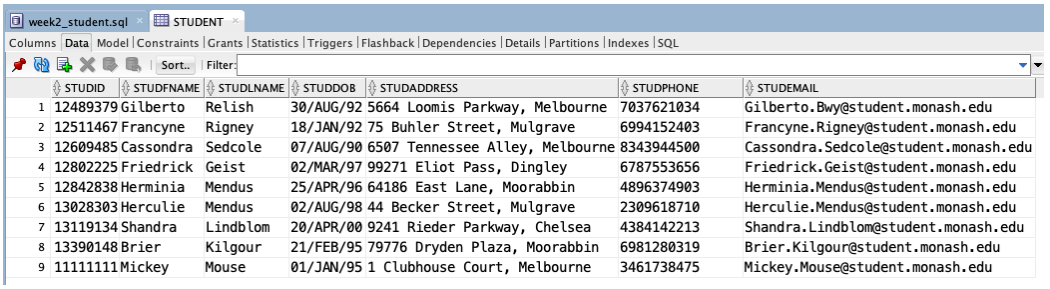
week2_student.sql | STUDENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	STUDID	STUDFNAME	STUDLNAME	STUDDOB	STUDADDRESS	STUDPHONE	STUDEMAIL
1	12489379	Gilberto	Relish	30/AUG/92	5664 Loomis Parkway, Melbourne	7037621034	Gilberto.Bwy@student.monash.edu
2	12511467	Francyne	Rigney	18/JAN/92	75 Buhler Street, Mulgrave	6994152403	Francyne.Rigney@student.monash.edu
3	12609485	Cassondra	Sedcole	07/AUG/90	6507 Tennessee Alley, Melbourne	8343944500	Cassondra.Sedcole@student.monash.edu
4	12802225	Friedrick	Geist	02/MAR/97	99271 Eliot Pass, Dingley	6787553656	Friedrick.Geist@student.monash.edu
5	12842838	Herminia	Mendus	25/APR/96	64186 East Lane, Moorabbin	4896374903	Herminia.Mendus@student.monash.edu
6	13028303	Herculie	Mendus	02/AUG/98	44 Becker Street, Mulgrave	2309618710	Herculie.Mendus@student.monash.edu
7	13119134	Shandra	Lindblom	20/APR/00	9241 Rieder Parkway, Chelsea	4384142213	Shandra.Lindblom@student.monash.edu
8	13390148	Brier	Kilgour	21/FEB/95	79776 Dryden Plaza, Moorabbin	6981280319	Brier.Kilgour@student.monash.edu
+9	(null)	(null)	(null)	(null)	(null)	(null)	(null)

3. Replace the (null) value with 11111111, Mickey, Mouse, 01/JAN/95, 1 Clubhouse Court, Melbourne, 3461738475 and Mickey.Mouse@student.monash.edu.
4. Click on the COMMIT icon  to save the changes.
5. You will see the new row has been added to the table.



week2_student.sql | STUDENT

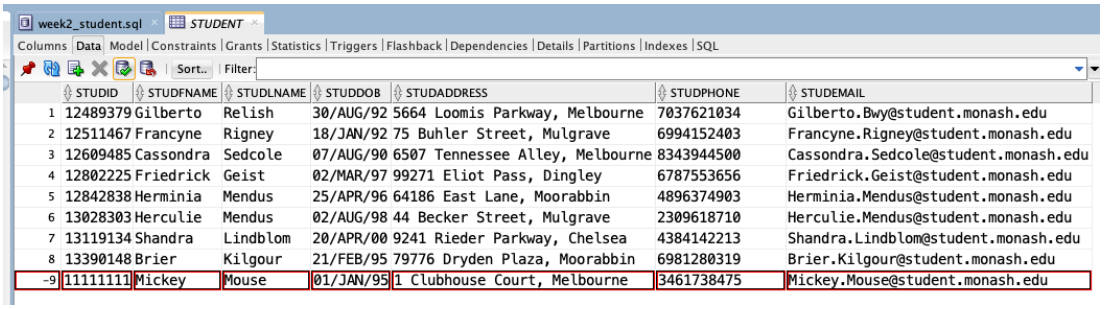
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	STUDID	STUDFNAME	STUDLNAME	STUDDOB	STUDADDRESS	STUDPHONE	STUDEMAIL
1	12489379	Gilberto	Relish	30/AUG/92	5664 Loomis Parkway, Melbourne	7037621034	Gilberto.Bwy@student.monash.edu
2	12511467	Francyne	Rigney	18/JAN/92	75 Buhler Street, Mulgrave	6994152403	Francyne.Rigney@student.monash.edu
3	12609485	Cassondra	Sedcole	07/AUG/90	6507 Tennessee Alley, Melbourne	8343944500	Cassondra.Sedcole@student.monash.edu
4	12802225	Friedrick	Geist	02/MAR/97	99271 Eliot Pass, Dingley	6787553656	Friedrick.Geist@student.monash.edu
5	12842838	Herminia	Mendus	25/APR/96	64186 East Lane, Moorabbin	4896374903	Herminia.Mendus@student.monash.edu
6	13028303	Herculie	Mendus	02/AUG/98	44 Becker Street, Mulgrave	2309618710	Herculie.Mendus@student.monash.edu
7	13119134	Shandra	Lindblom	20/APR/00	9241 Rieder Parkway, Chelsea	4384142213	Shandra.Lindblom@student.monash.edu
8	13390148	Brier	Kilgour	21/FEB/95	79776 Dryden Plaza, Moorabbin	6981280319	Brier.Kilgour@student.monash.edu
9	11111111	Mickey	Mouse	01/JAN/95	1 Clubhouse Court, Melbourne	3461738475	Mickey.Mouse@student.monash.edu

6. Deleting a row from the table.

1. Click on the row to be deleted, for example, choose row 9.
2. Once the row is highlighted, choose the delete icon from the menu. 
3. You will see the row to be marked with a negative sign at the front of the row number.



week2_student.sql | STUDENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	STUDID	STUDFNAME	STUDLNAME	STUDDOB	STUDADDRESS	STUDPHONE	STUDEMAIL
1	12489379	Gilberto	Relish	30/AUG/92	5664 Loomis Parkway, Melbourne	7037621034	Gilberto.Bwy@student.monash.edu
2	12511467	Francyne	Rigney	18/JAN/92	75 Buhler Street, Mulgrave	6994152403	Francyne.Rigney@student.monash.edu
3	12609485	Cassondra	Sedcole	07/AUG/90	6507 Tennessee Alley, Melbourne	8343944500	Cassondra.Sedcole@student.monash.edu
4	12802225	Friedrick	Geist	02/MAR/97	99271 Eliot Pass, Dingley	6787553656	Friedrick.Geist@student.monash.edu
5	12842838	Herminia	Mendus	25/APR/96	64186 East Lane, Moorabbin	4896374903	Herminia.Mendus@student.monash.edu
6	13028303	Herculie	Mendus	02/AUG/98	44 Becker Street, Mulgrave	2309618710	Herculie.Mendus@student.monash.edu
7	13119134	Shandra	Lindblom	20/APR/00	9241 Rieder Parkway, Chelsea	4384142213	Shandra.Lindblom@student.monash.edu
8	13390148	Brier	Kilgour	21/FEB/95	79776 Dryden Plaza, Moorabbin	6981280319	Brier.Kilgour@student.monash.edu
-9	11111111	Mickey	Mouse	01/JAN/95	1 Clubhouse Court, Melbourne	3461738475	Mickey.Mouse@student.monash.edu

4. Click on the COMMIT icon  to save the deletion.
5. You should now see that the row has been deleted.

2.4. Using SQL Developer Git Client to Revert and Reset Commit

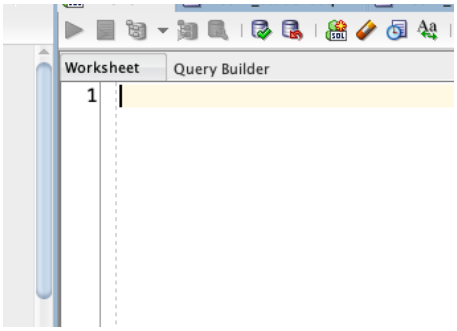
Since Git maintains all versions of your work, you may use this to go back to a previous version of a particular file by removing a commit. Commits which you have added to Git may be removed in one of two ways:

- using REVERT which adds a compensating commit to cancel out a previous commit, or
- using RESET which resets the head of the Git branch

REVERT is used in the scenario where you have pushed a commit to the remote Git Server (the FITGit Lab server) ie. to *undo a commit on the remote server*. RESET (HARD) is used to undo a commit *on your local repo before it is pushed to the remote server*. A video is available which explains each of these processes and shows some simple examples of revert and reset.

2.5. Using the SQL Command to Retrieve Data

In this section, you will learn to use SQL commands to retrieve data from the tables. You can write SQL commands in the Worksheet area on SQL Developer. Please note at this stage you do not need to be concerned with the syntax of the SQL select command, we will return to this later in the semester, just type the text in as given in the examples below.



1. Retrieve all student details

Write:

```
select *
from student;
```

Click the run icon or press ctrl + enter button. Observe the output in the Query Result window.

2. Display all students who live in Moorabbin

Write:

```
select *
from student
where studaddress like '%Moorabbin%';
```

Click the run icon or press ctrl + enter button. Observe the output in the Query Result window.

3. Display the first name and last name of all students who live in Moorabbin

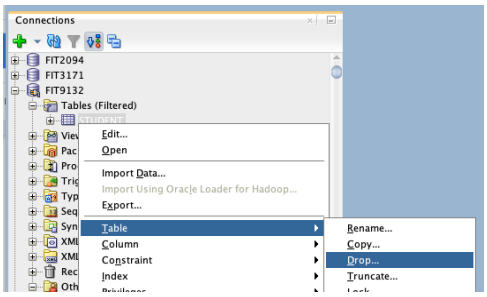
Write:

```
select studfname, studlname
from student
where studaddress like '%Moorabbin%';
```

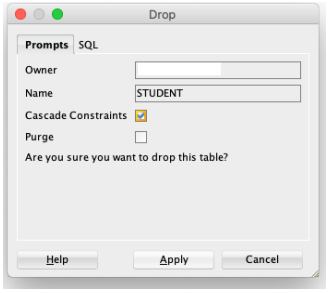
Click the run icon or press ctrl + enter button. Observe the output in the Query Result window.

2.6. Using the SQL Developer GUI to Drop a Table

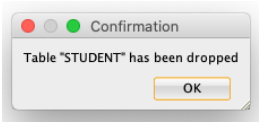
Each account has limited storage in our Oracle server, therefore it is important to only keep tables in your account which you are *currently* using. When a table is not required any longer, you must drop the table. At this stage of the semester, to drop the table, right click on the table name → Table → Drop.



Check “Cascade Constraints” box, Then click “Apply”



Finally, click “OK”



2.7. Relational vs NoSQL Databases Comparison

Relational databases are currently the most popular database type used in the industry. There are other database types, however, which are more suited to specific types of data. For example, relational databases are not the best database type for storing data with complex and varied structure. MongoDB is a noSQL database which is well suited to storing such data. The following is a "quick" look at MongDB, we will return to MongoDB in more depth in week 11.

Open the MongoDB shell https://docs.mongodb.com/manual/tutorial/getting-started/, then click the working window to connect to the database.

A MongoDB database is a set of collections, and a collection is a set of documents. Create a collection by inserting documents into the collection. Use the following syntax to create the collection. Paste the json data provided in week2_student.json within the [] (square brackets).

```
db.student.insertMany([<paste the json text here>]);
```

1. Retrieve all student details

Type:

```
db.student.find().pretty();
```

Press enter, discuss the output and compare the MongoDB query process with using SQL in Oracle.

2. Display all students who live in Moorabbin

Type:

```
db.student.find({"address":/.*Moorabbin*./}).pretty();
```

Press enter, discuss the output and compare the MongoDB query process with using SQL in Oracle.

3. Display the id, first name and last name of all students who live in Moorabbin

Type:

```
db.student.find({"address":/.*Moorabbin*./},{"_id":0,"firstName":1,"lastName":1});
```

Press enter, discuss the output and compare the MongoDB query process with using SQL in Oracle.

Summary

Designing what data and how it is stored in an optimal form within an organisation is an important task. A poorly designed/incorrect type of data store may lead to serious problems/performance issues. In this unit, you will learn how to ensure that you will build a database which suits a company's data requirements. In the week 2 workshop and week 3 tutorial, we will take the first design step where you will learn how to create a conceptual design for a database.

Important

You need to get into the habit of establishing this as a standard FIT9132 workflow - Open SQL Developer, pull at the start of your working session, work on the activities you wish to/are able to complete during this session, add files (stage) and commit changes, then push the changes back to the FIT GitLab server. We suggest you add, commit and push regularly as you change and save files - do not leave it to just once at the end of the tutorial, do ensure at the end all work you have completed is added, committed and pushed.

When you have finished with SQL Developer it is very important that you disconnect from your database connection, close all open SQL Developer files/projects and then close SQL Developer before closing your laptop.