FIT2094 Databases

Week 10 Tutorial Activities

SQL Part II (SQL Intermediate)

FIT Database Teaching Team

Complete the week 10 activities listed below:

FIT2094-FIT3171 2021 S2

FIT2094-FIT3171 Introduction to Databases

Author: FIT Database Teaching Team

License: Copyright © Monash University, unless otherwise stated. All Rights Reserved.


Learning Objectives:

- be able to use SQL aggregate functions (SUM, AVG, COUNT)
- be able to write SQL using GROUP BY Clause
- be able to use SQL clause HAVING
- be able to use a subquery in SQL

10.1 Using the SQL Developer Query Builder

First, set the preference to open the table using a double click instead of a single click. Open SQL Developer Preferences → Database → uncheck Open Object on Single Click.
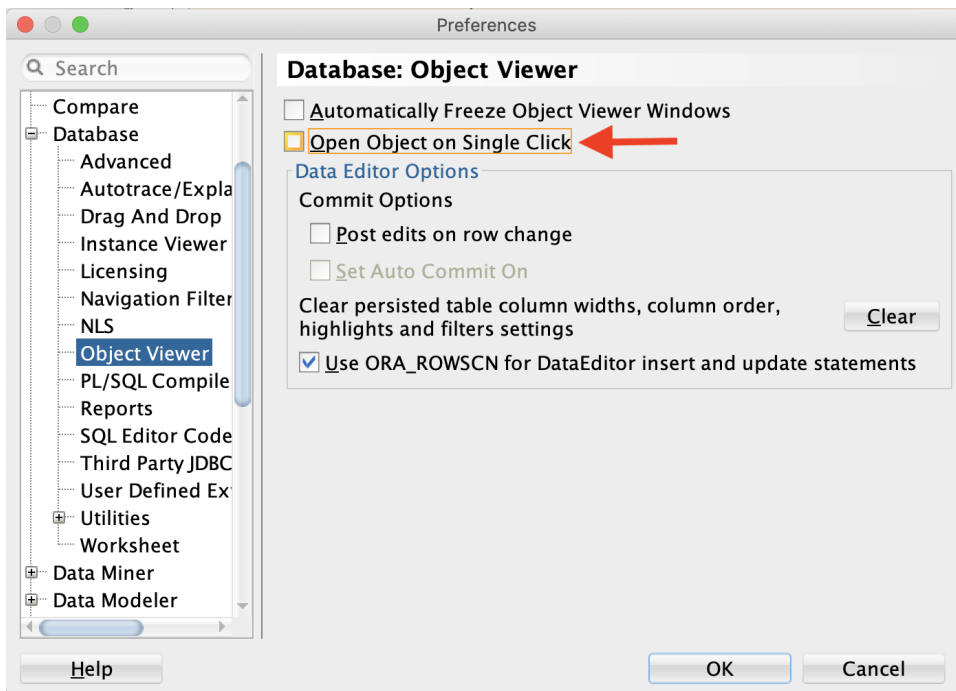
The Query Builder tab is located alongside the Worksheet tab:
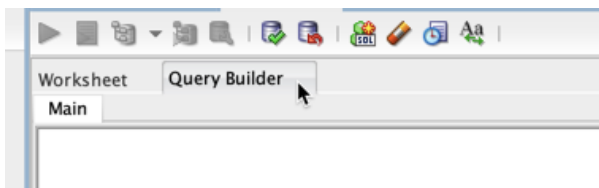
Using the Query Builder you can quickly and efficiently create SQL queries or beginning templates.

As an example take the query "*List the student name of all students who have marks in the range of 60 to 70*".

1. First, we need to identify the tables which will be needed - here uni.student and uni.enrolment

2. Drag and drop the tables from the uni account into query builder:
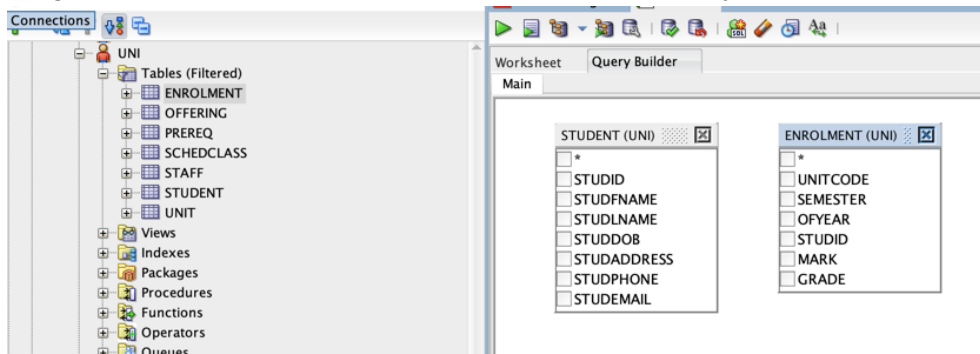
3. Indicate the join condition between the two tables, by dragging the STUDID attribute from STUDENT onto the STUDID attribute of ENROLMENT:
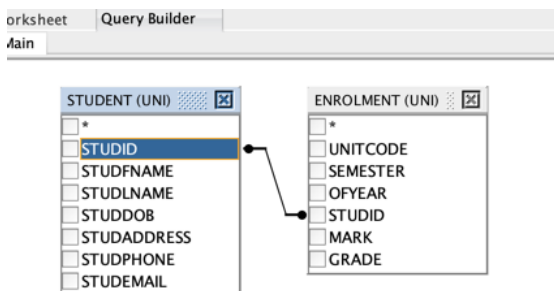
If required the join can be further specified by double-clicking on the join line, for example, to specify this should be an outer join (Select all from the appropriate side)
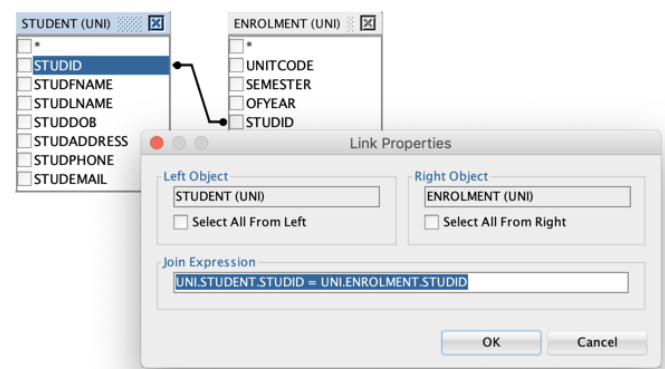
4. Indicate which attributes you wish to be displayed in the select list by ticking the appropriate boxes:
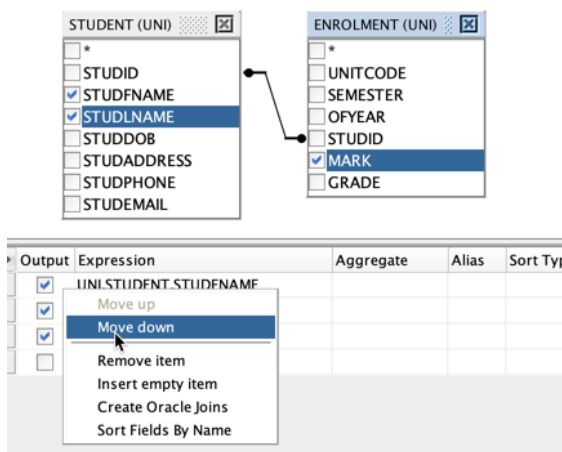
Right-clicking on an output row offers a number of actions including move up/down to position the item

5. Use the Criteria column to set the WHERE clause:

	Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria	Or
	✔	UNI.STUDENT.STUDLNAME					☐		
	✔	UNI.STUDENT.STUDFNAME					☐		
	✔	UNI.ENROLMENT.MARK					☐	Between 60 And 70	
	☐						☐		

6. Lastly, specify the sort order:

	Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria	Or
	✔	UNI.STUDENT.STUDLNAME			Ascending	2	☐		
	✔	UNI.STUDENT.STUDFNAME			Ascending	1	☐		
	✔	UNI.ENROLMENT.MARK			Ascending	3	☐	Between 60 And 70	
	☐						☐		

The query can be run directly from within the Query Builder by selecting the normal Run Statement (the green arrow) as you did in the Worksheet.

	Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria	Or
	✔	UNI.STUDENT.STUDLNAME			Ascending	2	☐		
	✔	UNI.STUDENT.STUDFNAME			Ascending	1	☐		
	✔	UNI.ENROLMENT.MARK			Ascending	3	☐	Between 60 And 70	

Query Result ×

SQL | All Rows Fetched: 43 in 0.01 seconds

	STUDLNAME	STUDFNAME	MARK
1	MacGinney	Aldwin	65
2	MacGinney	Aldwin	65
3	Swiffen	Artus	69
4	Corp	Averell	65
5	Corp	Averell	69
6	Friedank	Billie	65
7	Friedank	Billie	70
8	Kilgour	Brior	69

Alternatively, if you swap back to the Worksheet, you will see your full SQL query built:

```
Worksheet    Query Builder
SELECT
    uni.student.studlname,
    uni.student.studfname,
    uni.enrolment.mark
FROM
    uni.student
    INNER JOIN uni.enrolment ON uni.student.studid = uni.enrolment.studid
WHERE
    uni.enrolment.mark BETWEEN 60 AND 70
ORDER BY
    uni.student.studfname,
    uni.student.studlname,
    uni.enrolment.mark
```

The query builder can help you quickly and efficiently build queries. **However, it is very important that you are able to code your own queries in the worksheet and do not depend on the query builder to do the task for you.** It is simply a tool you may use in building queries.

10.2 SQL Intermediate

Important

Remember before starting any lab activity which involves working with files, first use SQL Developer to pull from FIT GitLab server to ensure your local files and the FIT GitLab server files are in sync. *During this activity, you will be creating a set of SQL scripts, these **MUST** be sent to the FIT GitLab server.*

The following exercises will allow you to be familiar with:

- All the SQL statements previously used
- Aggregate functions such as min, max, avg
- The GROUP BY clause
- Subqueries

This week we will continue to use the UNIVERSITY database model:
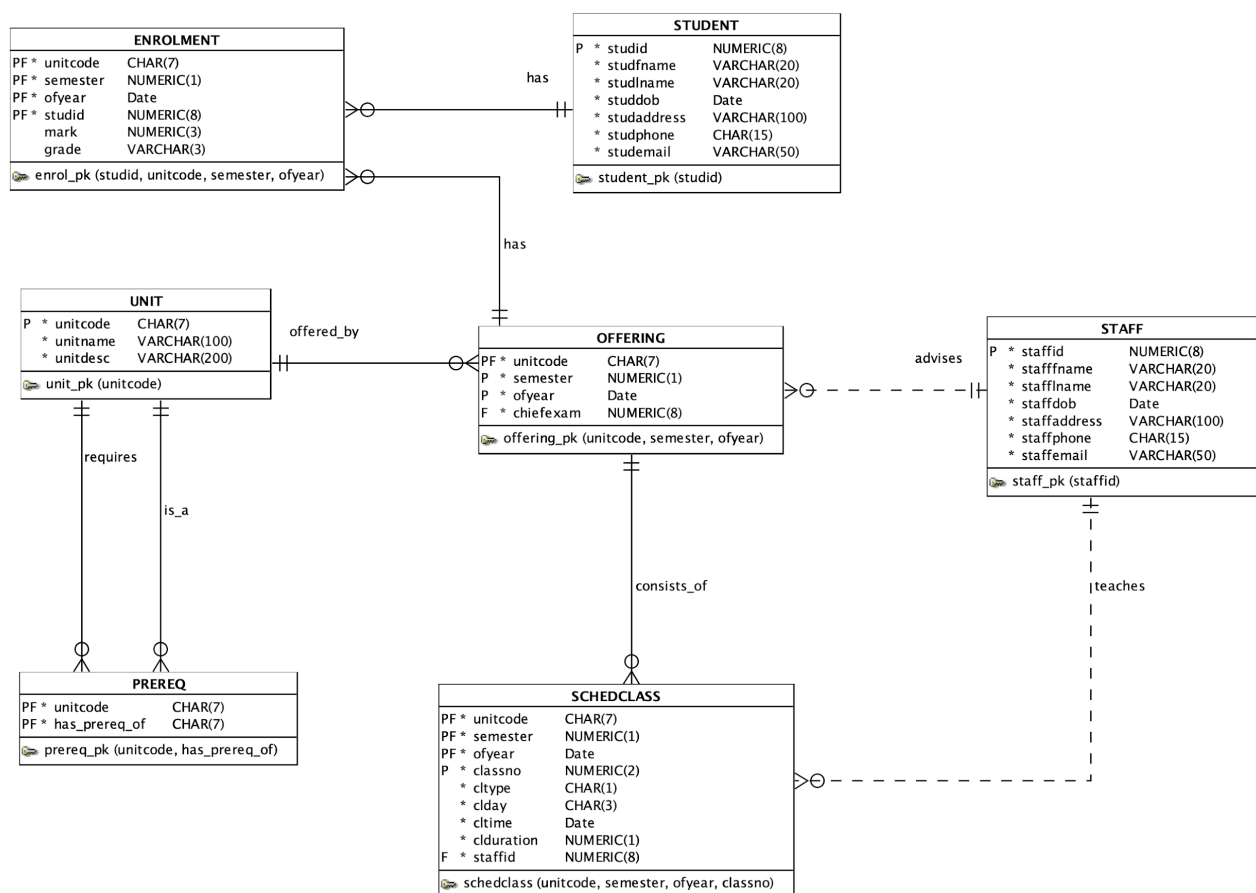
University Data model

10.2.1 Class Discussion

1. Find the maximum mark for FIT9136 in semester 2, 2019.

2. Find the total number of enrolments per semester for each unit in the year 2019. The list should include the unitcode, semester and year. Order the list in increasing order of enrolment numbers.

3. Find the oldest student/s in FIT9132? Display the student's full name and the date of birth. Sort the list by student id.

10.2.2 SQL Intermediate Questions

Download **week10_sql_intermediate.sql** from week 10 block in Moodle, place this file in your working directory in the Tut10 folder. Write your answers for question 1 - 10 in the provided area. Test the select statement one by one.

1. Find the average mark of FIT2094 in semester 2, 2019. Show the average mark with two decimal places. Name the output column as "Average Mark".

2. List the average mark for each offering of FIT9136. In the listing, include the year and semester number. Sort the result according to the year then the semester.

3. Find the number of students enrolled in FIT1045 in the year 2019, under the following conditions (note two separate selects are required):

   a. Repeat students are counted each time

   b. Repeat students are only counted once

4. Find the total number of prerequisite units for FIT5145.

5. Find the total number of prerequisite units for each unit. In the list, include the unit code for which the count is applicable. Order the list by unit code.

6. Find the total number of students whose marks are being withheld (grade is recorded as 'WH') for each unit offered in semester 1 2020. In the listing include the unit code for which the count is applicable. Sort the list by descending order of the total number of students whose marks are being withheld, then by the unit code.

7. For each prerequisite unit, calculate how many times it has been used as a prerequisite (number of times used). In the listing include the prerequisite unit code, the prerequisite unit name and the number of times used. Sort the output by unit name.

8. Display unit code and unit name of units which had at least 1 student who was granted deferred exam (grade is recorded as 'DEF') in semester 1 2020. Order the list by unit code.

9. Find the unit/s with the highest number of enrolments for each offering in the year 2019. Sort the list by semester then by unit code.

10. Find all students enrolled in FIT3157 in semester 1, 2020 who have scored more than the average mark of FIT3157 in the same offering? Display the students' name and the mark. Sort the list in the order of the mark from the highest to the lowest then in increasing order of student name.

Important

You need to get into the habit of establishing this as a standard FIT2094-FIT3171 workflow - pull at the start of your working session, work on the activities you wish to/are able to complete during this session, save the files, add all (stage), commit and then push the changes back to the FIT GitLab server