Applied 10 SQL Part II - Intermediate SQL

Applied 10 SQL Part II - Intermediate SQL

After completing this activity, you should be able to:

- use the SQL aggregate functions (SUM, AVG, COUNT)
- code SQL using GROUP BY Clause
- use the SQL clause HAVING
- use a subquery in SQL

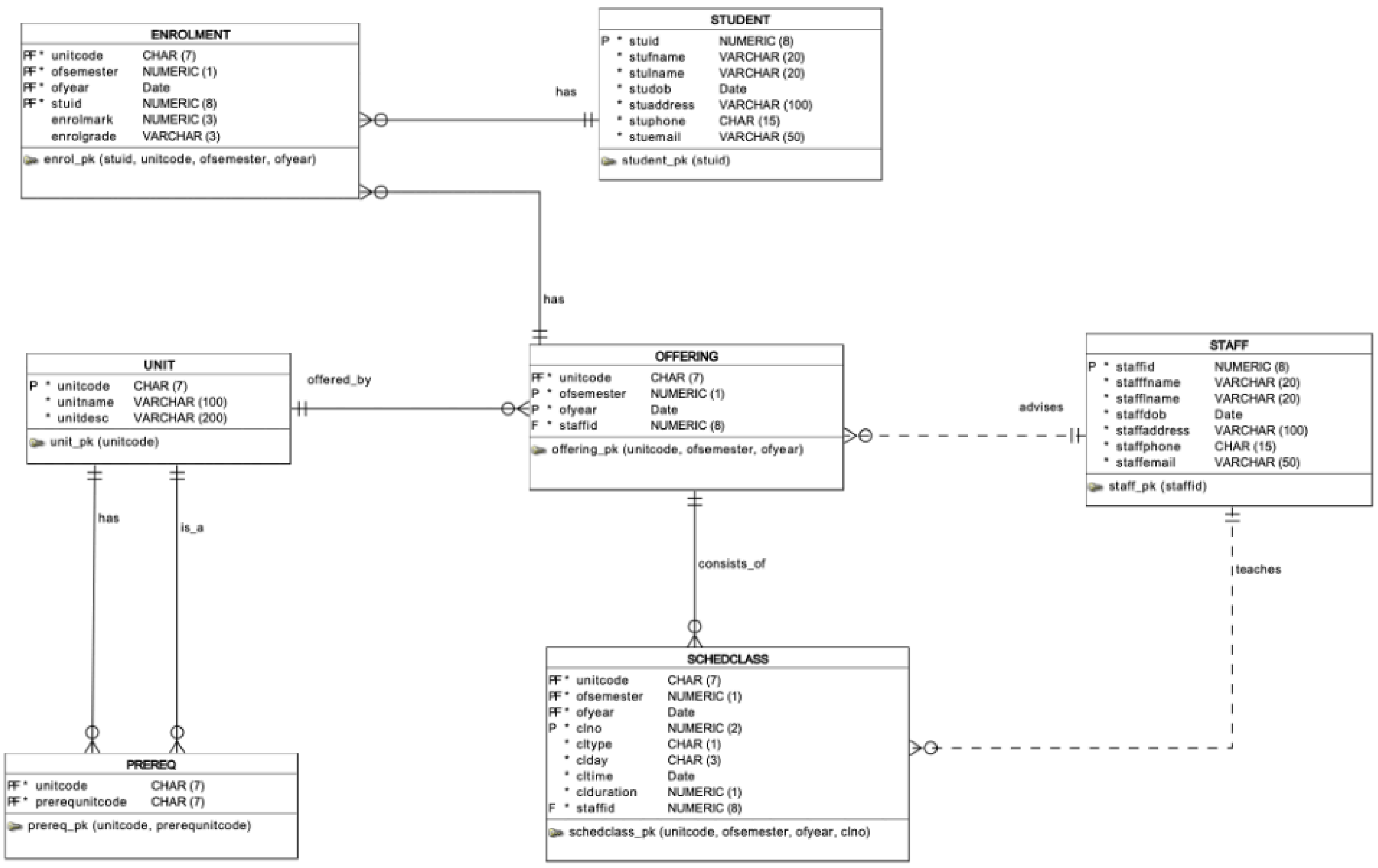
This activity supports unit learning outcomes 1 and 4.

A10-1 SQL Part II (SQL Intermediate)

The following exercises will allow you to be familiar with:

- All the SQL statements previously used.
- Aggregate functions: min, max, sum, avg, count
- The GROUP BY clause
- Subqueries

For this module we will continue to use the UNIVERSITY database model:



Download **applied10_sql_intermediate.sql** from below:

applied10_sql_intermediate.sql

Place this file in your working directory in your Applied 10 folder. Open the file in VSCode and add your details to the header. Save the file, stage (add), commit and push.

Then, open Oracle ORDS:

- https://ora-fit.ocio.monash.edu:8441/ords/sql-developer

Write your answers for questions 1 - 3 using Oracle ORDS - this is **practice for what you will be required to do in Class Test 2** (except in Class Test 2 you will paste into Moodle).

Code the select statements one by one, test the select statement by clicking the Run Statement icon and then, when working, copy the working code to the appropriate question in applied10_sql_intermediate.sql in VSCode. Clear the ORDS worksheet with the trash icon

and move to the next question - repeat for each of the three question. When finished disconnect and close ORDS.

In VSCode save the file, stage (add), commit and push.

Write your answers for questions 4-13 in the provided area on VSCode. Make sure that you include a semicolon ';' at the end of each select statement. Test the select statements one by one by clicking the Run Statement icon.

TASKS

1. Find the maximum mark for *FIT9136* in semester 2, 2019.
2. Find the average mark for *FIT2094* in semester 2, 2020. Show the average mark with two decimal places. Name the output column as average_mark.
3. List the average mark for each offering of *FIT9136*. A unit offering is an instance of a particular unit in a particular semester - for example FIT1045 offered in semester 1 of 2019 - is a unit offering. In the listing, include the year and semester number. Sort the result according to the year then the semester.
4. Find the number of students enrolled in *FIT1045* in the year 2019, under the following conditions (note two separate selects are required):
    - Repeat students are counted multiple times in each semester of 2019
    - Repeat students are only counted once across 2019
5. Find the total number of prerequisite units for *FIT5145*.
6. Find the total number of enrolments per semester for each unit in the year 2019. The list should include the unit code, semester and the total number of enrolments. Order the list in increasing order of enrolment numbers. For units with the same number of enrolments, display them by the unit code order then by the semester order.
7. Find the total number of prerequisite units for each unit which has prerequisites. In the list, include the unit code for which the count is applicable. Order the list by unit code.
8. Find the total number of students whose marks are being withheld (grade is recorded as *WH*) for each unit offered in semester 2 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.
9. 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 prerequisite unit code.

10. Display the unit code and unit name of units which had at least 2 students who were granted a deferred exam (grade is recorded as *DEF*) in semester 2 2021. Order the list by unit code.
11. Find the oldest student/s in *FIT9132.* Display the student's id, full name and the date of birth. Sort the list by student id.
12. A unit offering is an instance of a particular unit in a particular semester - for example FIT1045 offered in semester 1 of 2019 - is a unit offering. All unit offerings are listed in the OFFERING table. Find the unit offering/s with the highest number of enrolments for any unit offering which occurred in the year 2019. Display the unit code, offering semester and number of students enrolled in the offering. Sort the list by semester then by unit code.
13. Find all students enrolled in *FIT3157* in semester 1, 2020 who have scored more than the average mark for *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 workflow:

- before modifying any file/s, pull at the start of your working session, work on the activities you wish to/are able to complete during this session, add all(stage), commit changes and then push the changes back to the FIT GitLab server.