Applied 11 SQL Part III - Advanced SQL

Applied 11 SQL Part III - Advanced SQL

After completing this activity, you should be able to:

- Consolidate your understanding of subqueries
- Use different SQL set operators (INTERSECT, DIFFERENCE, UNION)
- Outline the role of VIEWs in the database
- Code SQL OUTER join, recursive relationship joins and join between tables that have more than one relationship
- Describe the role of query optimisation and its relationship to relational algebra

This activity supports unit learning outcomes 1 and 4.

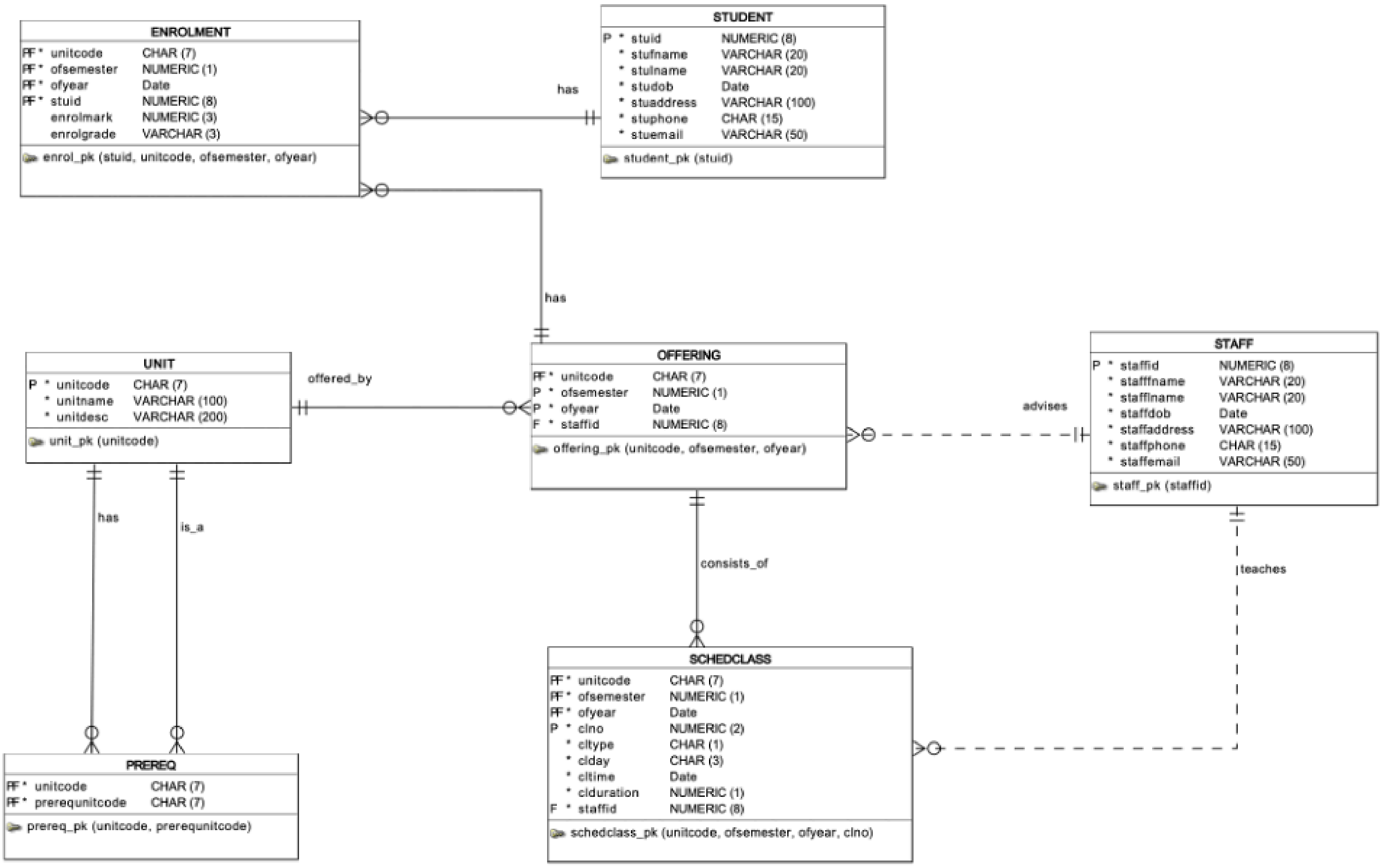
A11-1 SQL Part III (SQL Advanced)

Important: Use of ORDER BY CLAUSE ASC/DESC

When answering SQL queries for this unit please note the following:

- if descending order is required this will be explicitly stated in the form"Order by ... **descending**"
- if the questions simply states "Order by ..." ie. with no qualifier, then this must be treated as ordering by the default order ie. ascending

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



Download **applied11_sql_advanced.sql** from below:

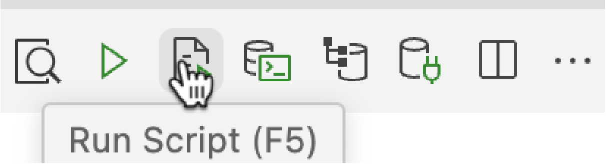
applied11_sql_advanced.sql

Place this file in your working directory in your App11 folder. Write your answers for question 1 - 10 in the provided area. Make sure that you include a semicolon ‘;’ at the end of each select statement.

Test the select statements one by one, by highlighting the required code and then using the run option in VS Code. The run statement uses proportional fonts which makes it difficult to size output, for example when testing LPAD.

If the query is concerned with placement of the column data e.g., right aligning, we need non-proportional fonts. In such a situation you can test/run the code by highlighting the code you wish to tun and using the Run Script button:

(a) In VS Code:



(b) As an alternative, with Oracle SQL Developer for VS Code, you add a SPOOL command to the top of your script, such as:

```
SPOOL output.txt
SET PAGESIZE 100
SET LINESIZE 300
```

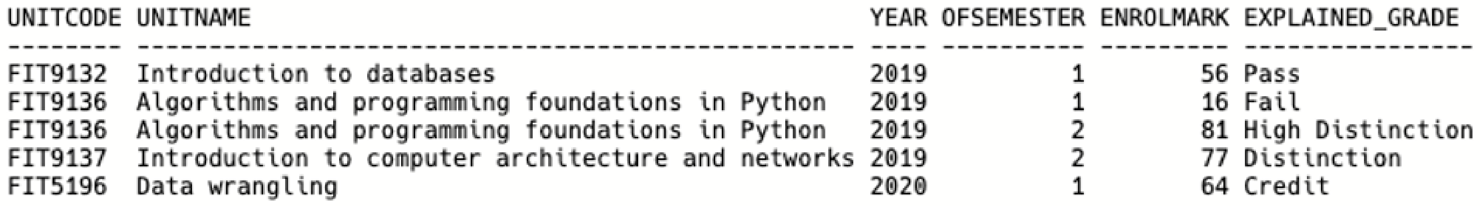
- pagesize = how many lines of output before a page feed (a new set of column headings is output)
- linesize = how many characters on a line

When run as a script, this will spool the output into a test file using non-proportional fonts. You can then open the output file and view the data placement.

NOTE: The sample output is shown for your guidance as to the displayed columns and formats. The data in the database is constantly changing and as a result your actual data displayed (the values in the rows of output) may be different. Therefore, it is important to:

- check whether your query is correct by manually checking the data in the table/s, and
- ensuring your query will work for any future possible cases/changes to the data.

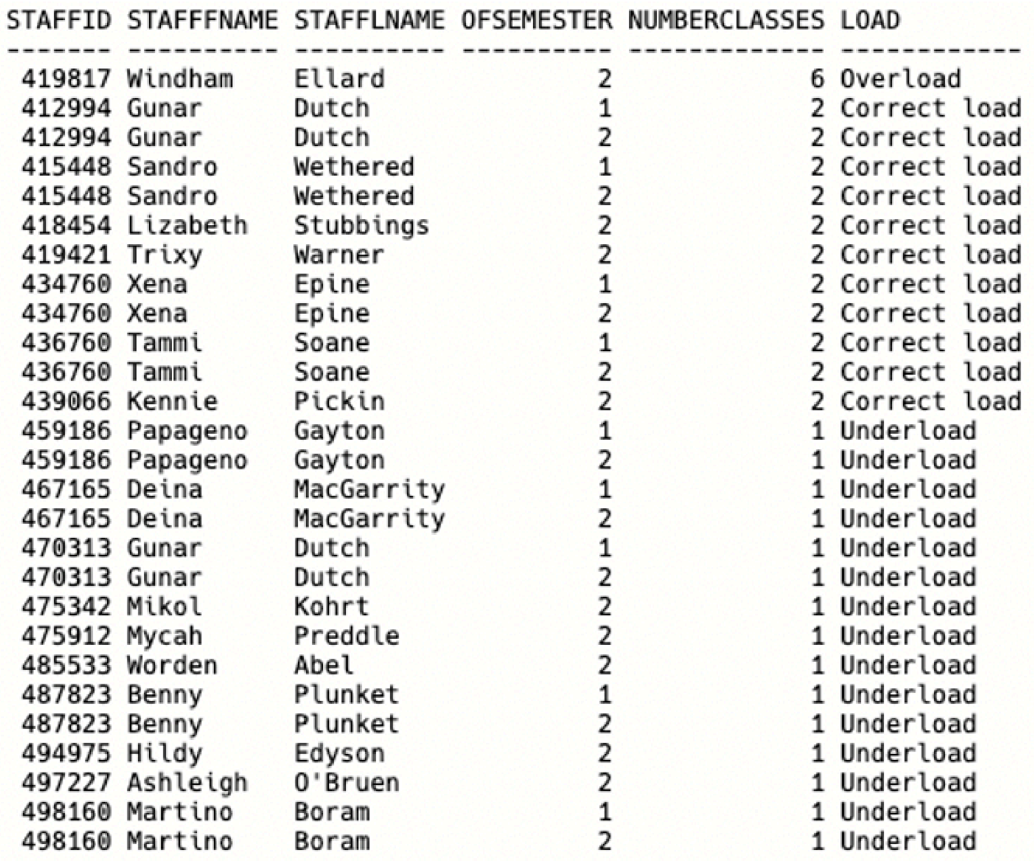
1. Assuming that the student's name is unique, display *Claudette Serman*'s academic record. Include the unit code, unit name, year, semester, mark and explained_grade in the listing. The Explained Grade column must show Fail for N, Pass for P, Credit for C, Distinction for D and High Distinction for HD. Order by year, within the same year order the list by semester, and within the same semester order the list by the unit code.



```
UNITCODE UNITNAME                                           YEAR OFSEMESTER ENROLMARK EXPLAINED_GRADE
-------- -------------------------------------------------- ---- ---------- --------- ----------------
FIT9132  Introduction to databases                          2019          1        56 Pass
FIT9136  Algorithms and programming foundations in Python   2019          1        16 Fail
FIT9136  Algorithms and programming foundations in Python   2019          2        81 High Distinction
FIT9137  Introduction to computer architecture and networks 2019          2        77 Distinction
FIT5196  Data wrangling                                     2020          1        64 Credit
```

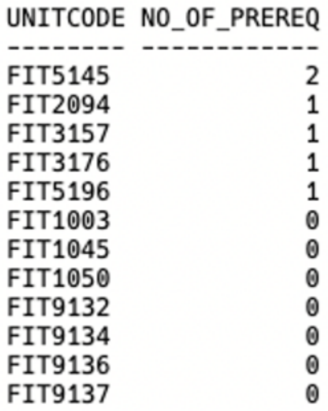
2. Find the number of scheduled classes assigned to each staff member for each semester in 2019. If the number of classes is 2 then this should be labelled as a correct load, more than 2 as an overload

and less than 2 as an underload. Include the staff id, staff first name, staff last name, semester, number of scheduled classes and load in the listing. Sort the list by decreasing order of the number of scheduled classes and when the number of classes is the same, sort by the staff id then by the semester.



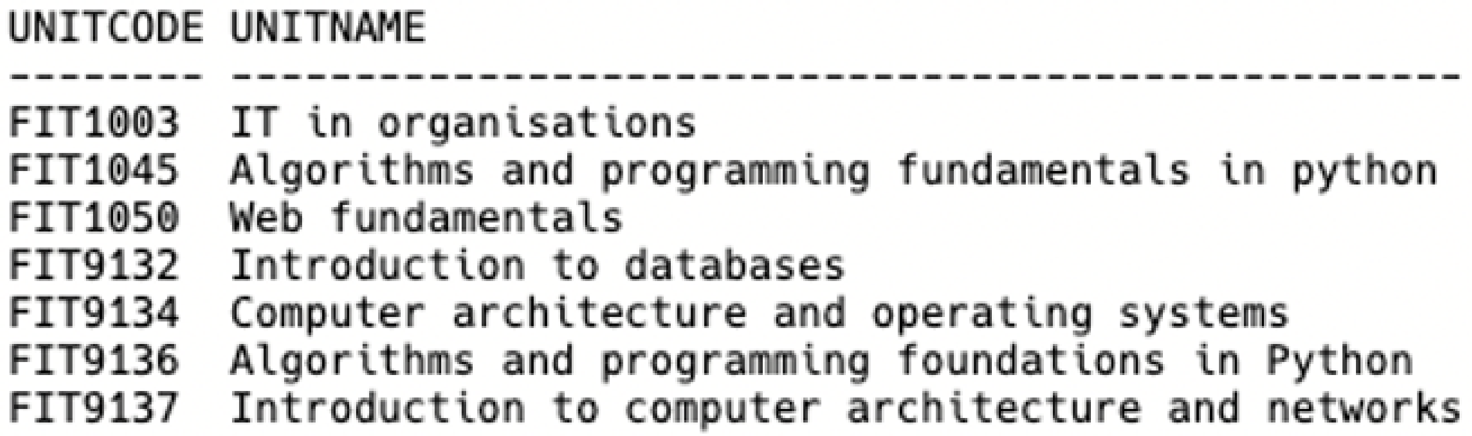
```
STAFFID STAFFFNAME STAFFLNAME OFSEMESTER NUMBERCLASSES LOAD
------- ---------- ---------- ---------- ------------- ------------
 419817 Windham    Ellard              2             6 Overload
 412994 Gunar      Dutch               1             2 Correct load
 412994 Gunar      Dutch               2             2 Correct load
 415448 Sandro     Wethered            1             2 Correct load
 415448 Sandro     Wethered            2             2 Correct load
 418454 Lizabeth   Stubbings           2             2 Correct load
 419421 Trixy      Warner              2             2 Correct load
 434760 Xena       Epine               1             2 Correct load
 434760 Xena       Epine               2             2 Correct load
 436760 Tammi      Soane               1             2 Correct load
 436760 Tammi      Soane               2             2 Correct load
 439066 Kennie     Pickin              2             2 Correct load
 459186 Papageno   Gayton              1             1 Underload
 459186 Papageno   Gayton              2             1 Underload
 467165 Deina      MacGarrity          1             1 Underload
 467165 Deina      MacGarrity          2             1 Underload
 470313 Gunar      Dutch               1             1 Underload
 470313 Gunar      Dutch               2             1 Underload
 475342 Mikol      Kohrt               2             1 Underload
 475912 Mycah      Preddle             2             1 Underload
 485533 Worden     Abel                2             1 Underload
 487823 Benny      Plunket             1             1 Underload
 487823 Benny      Plunket             2             1 Underload
 494975 Hildy      Edyson              2             1 Underload
 497227 Ashleigh   O'Bruen             2             1 Underload
 498160 Martino    Boram               1             1 Underload
 498160 Martino    Boram               2             1 Underload
```

3. Find the total number of prerequisite units for all units. Include in the list the unit code of units that do not have a prerequisite. Order the list in descending order of the number of prerequisite units and where several units have the same number of prerequisites order then by unit code.



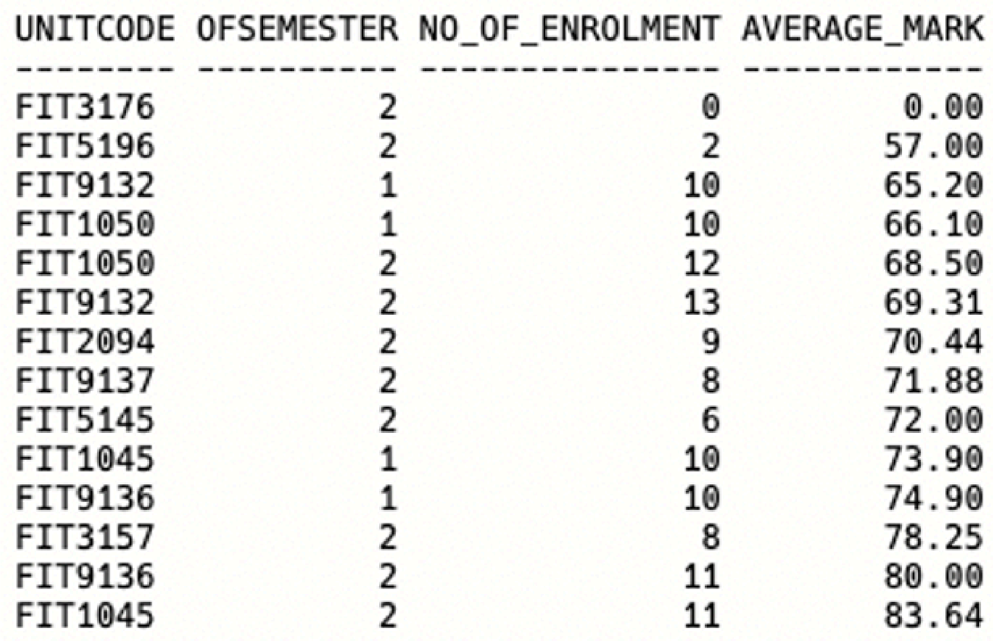
```
UNITCODE NO_OF_PREREQ
-------- ------------
FIT5145             2
FIT2094             1
FIT3157             1
FIT3176             1
FIT5196             1
FIT1003             0
FIT1045             0
FIT1050             0
FIT9132             0
FIT9134             0
FIT9136             0
FIT9137             0
```

4. Display the unit code and unit name for units that do not have a prerequisite. Order the list by unit code. There are many approaches that you can take in writing an SQL statement to answer this query. You can use the SET OPERATORS, OUTER JOIN and a SUBQUERY. Write SQL statements based on ***all*** three approaches.



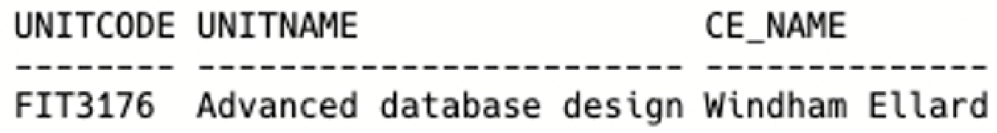
```
UNITCODE UNITNAME
-------- --------------------------------------------------------
FIT1003  IT in organisations
FIT1045  Algorithms and programming fundamentals in python
FIT1050  Web fundamentals
FIT9132  Introduction to databases
FIT9134  Computer architecture and operating systems
FIT9136  Algorithms and programming foundations in Python
FIT9137  Introduction to computer architecture and networks
```

5. List the unit code, semester, number of enrolments and the average mark for each unit offering in 2019. A unit offering is a particular unit in a particular semester for a particular year - for example the offering of FIT3176 in semester 2 of 2019 is one offering. Include offerings without any enrolment in the list. Round the average to 2 digits after the decimal point. If the average result is 'null', display the average as 0.00. The average must be shown with two decimal digits and right aligned. Order the list by the average mark, and when the average mark for several offerings is the same, sort by the semester then by the unit code.



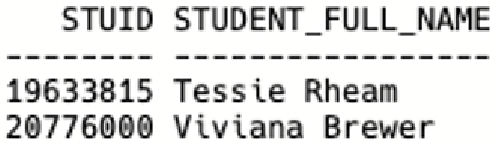
```
UNITCODE OFSEMESTER NO_OF_ENROLMENT AVERAGE_MARK
-------- ---------- --------------- ------------
FIT3176           2               0         0.00
FIT5196           2               2        57.00
FIT9132           1              10        65.20
FIT1050           1              10        66.10
FIT1050           2              12        68.50
FIT9132           2              13        69.31
FIT2094           2               9        70.44
FIT9137           2               8        71.88
FIT5145           2               6        72.00
FIT1045           1              10        73.90
FIT9136           1              10        74.90
FIT3157           2               8        78.25
FIT9136           2              11        80.00
FIT1045           2              11        83.64
```

6. List all units offered in semester 2 2019 which do not have any students enrolled. Include the unit code, unit name, and the chief examiner's name in a single column titled ce_name. Order the list based on the unit code.



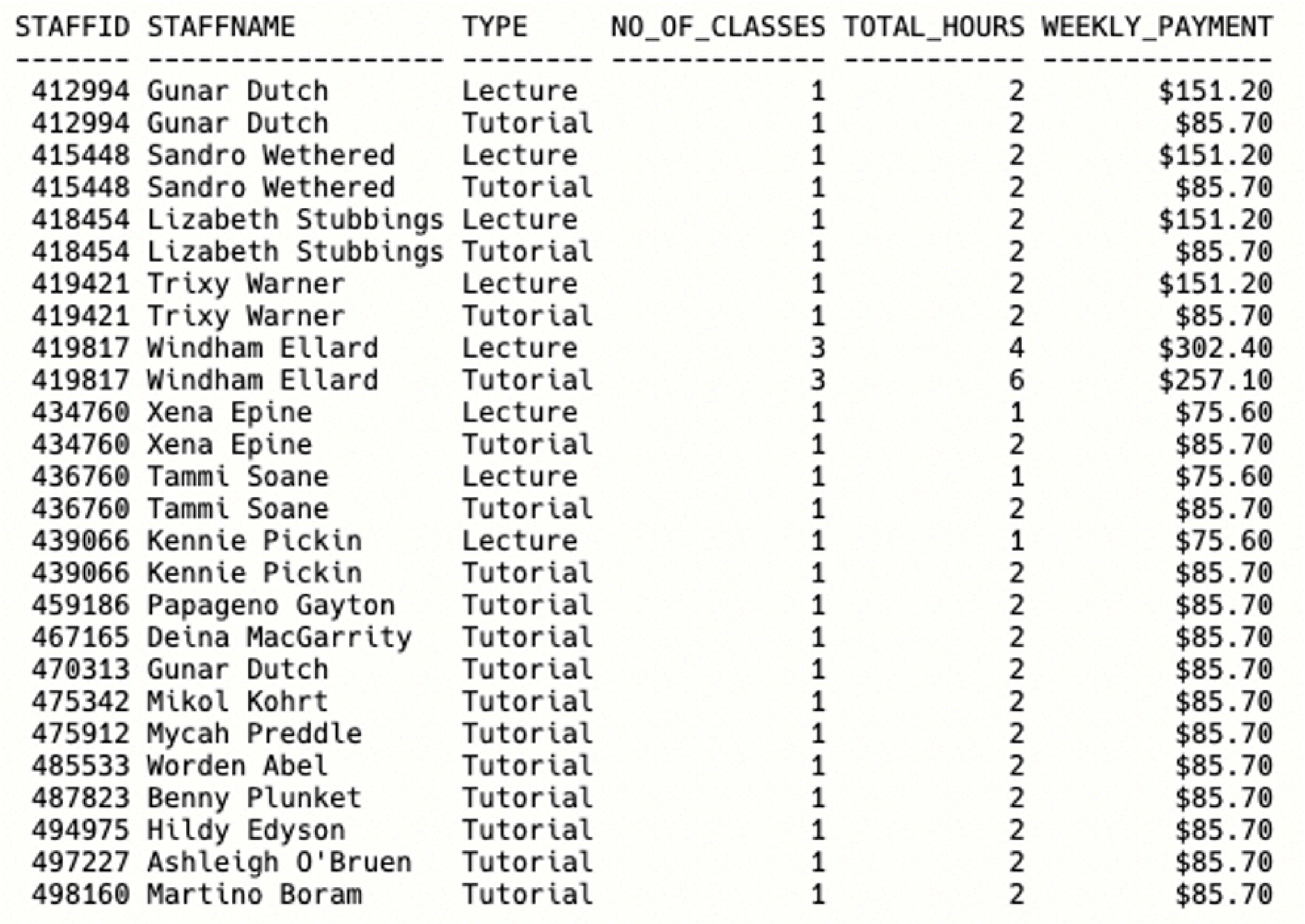
```
UNITCODE UNITNAME                  CE_NAME
-------- ------------------------- --------------
FIT3176  Advanced database design Windham Ellard
```

7. List the id and full name (in a single column titled student_full_name) of students who are enrolled in both *Introduction to databases* and *Introduction to computer architecture and networks* (note: both unit names are unique) in semester 1 2020. You should note that the case provided for these unit names does not necessarily match the case in the database. Order the list by the student id.



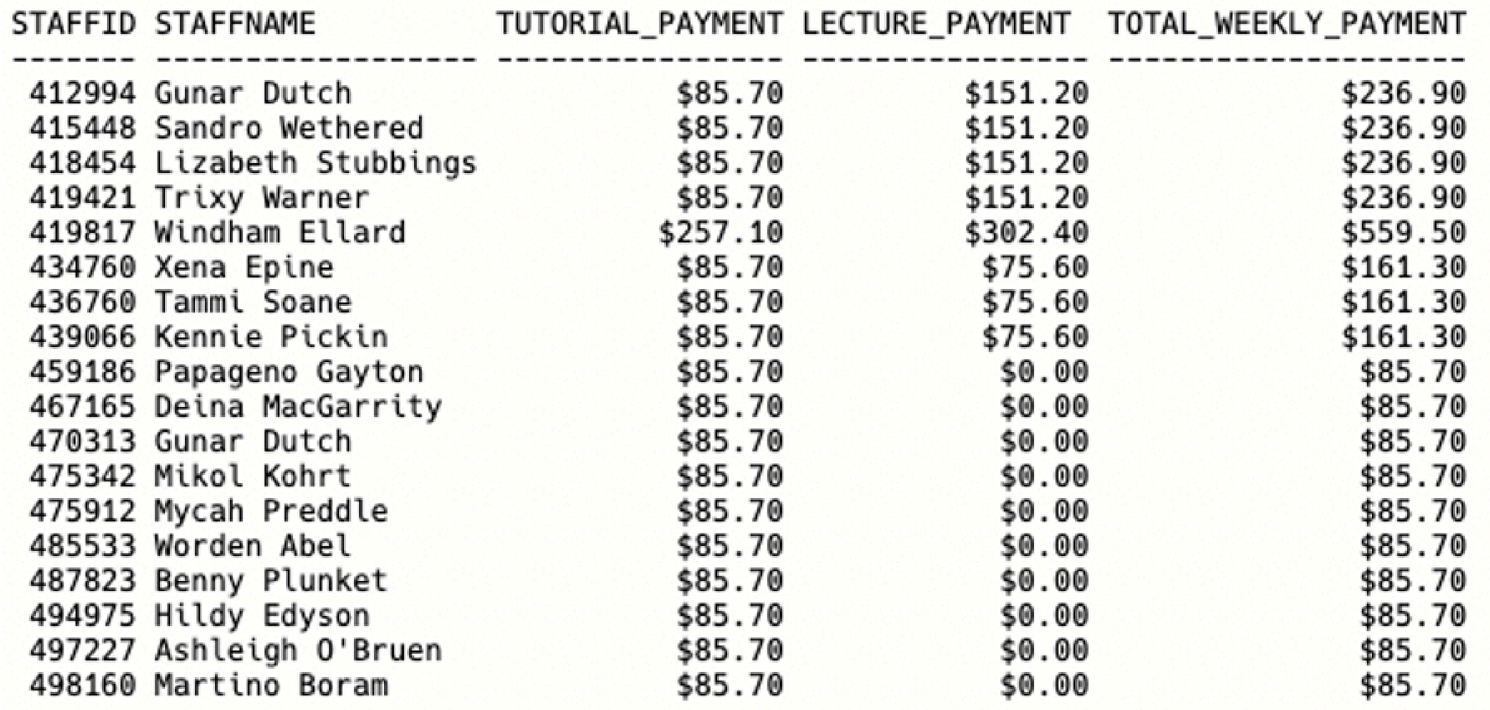
```
   STUID STUDENT_FULL_NAME
-------- -----------------
19633815 Tessie Rheam
20776000 Viviana Brewer
```

8. Given that the payment rate for a tutorial is $42.85 per hour and the payment rate for a lecture is $75.60 per hour, calculate the weekly payment per type of class for each staff member in semester 1 2020. In the display, include staff id, staff name, type of class (lecture - *L* or tutorial - *T*), number of classes, number of hours (total duration), and weekly payment (number of hours * payment rate). The weekly payment must be displayed to two decimal points and right aligned. Order the list by the staff id and for a given staff id by the type of class.



```
STAFFID STAFFNAME          TYPE     NO_OF_CLASSES TOTAL_HOURS WEEKLY_PAYMENT
------- ------------------ -------- ------------- ----------- --------------
 412994 Gunar Dutch        Lecture              1           2        $151.20
 412994 Gunar Dutch        Tutorial             1           2         $85.70
 415448 Sandro Wethered    Lecture              1           2        $151.20
 415448 Sandro Wethered    Tutorial             1           2         $85.70
 418454 Lizabeth Stubbings Lecture              1           2        $151.20
 418454 Lizabeth Stubbings Tutorial             1           2         $85.70
 419421 Trixy Warner       Lecture              1           2        $151.20
 419421 Trixy Warner       Tutorial             1           2         $85.70
 419817 Windham Ellard     Lecture              3           4        $302.40
 419817 Windham Ellard     Tutorial             3           6        $257.10
 434760 Xena Epine         Lecture              1           1         $75.60
 434760 Xena Epine         Tutorial             1           2         $85.70
 436760 Tammi Soane        Lecture              1           1         $75.60
 436760 Tammi Soane        Tutorial             1           2         $85.70
 439066 Kennie Pickin      Lecture              1           1         $75.60
 439066 Kennie Pickin      Tutorial             1           2         $85.70
 459186 Papageno Gayton    Tutorial             1           2         $85.70
 467165 Deina MacGarrity   Tutorial             1           2         $85.70
 470313 Gunar Dutch        Tutorial             1           2         $85.70
 475342 Mikol Kohrt        Tutorial             1           2         $85.70
 475912 Mycah Preddle      Tutorial             1           2         $85.70
 485533 Worden Abel        Tutorial             1           2         $85.70
 487823 Benny Plunket      Tutorial             1           2         $85.70
 494975 Hildy Edyson       Tutorial             1           2         $85.70
 497227 Ashleigh O'Bruen   Tutorial             1           2         $85.70
 498160 Martino Boram      Tutorial             1           2         $85.70
```

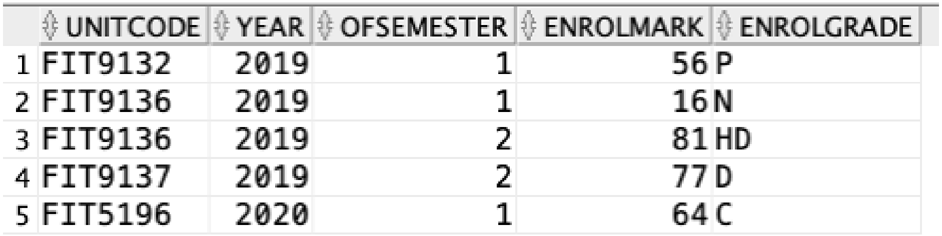
9. Given that the payment rate for a tutorial is $42.85 per hour and the payment rate for a lecture is $75.60 per hour, calculate the total weekly payment (the sum of both tutorial and lecture payments) for each staff member in semester 1 2020. In the display, include staff id, staff name, total weekly payment for tutorials, total weekly payment for lectures and the total weekly payment as a single line of output. If the payment is null, show it as $0.00. The tutorial payment, lecture payment and total weekly payment must be displayed to two decimal points and right aligned. Order the list by the staff id.



```
STAFFID STAFFNAME          TUTORIAL_PAYMENT LECTURE_PAYMENT  TOTAL_WEEKLY_PAYMENT
------- ------------------ ---------------- ---------------- --------------------
 412994 Gunar Dutch                  $85.70          $151.20              $236.90
 415448 Sandro Wethered              $85.70          $151.20              $236.90
 418454 Lizabeth Stubbings           $85.70          $151.20              $236.90
 419421 Trixy Warner                 $85.70          $151.20              $236.90
 419817 Windham Ellard              $257.10          $302.40              $559.50
 434760 Xena Epine                   $85.70           $75.60              $161.30
 436760 Tammi Soane                  $85.70           $75.60              $161.30
 439066 Kennie Pickin                $85.70           $75.60              $161.30
 459186 Papageno Gayton              $85.70            $0.00               $85.70
 467165 Deina MacGarrity             $85.70            $0.00               $85.70
 470313 Gunar Dutch                  $85.70            $0.00               $85.70
 475342 Mikol Kohrt                  $85.70            $0.00               $85.70
 475912 Mycah Preddle                $85.70            $0.00               $85.70
 485533 Worden Abel                  $85.70            $0.00               $85.70
 487823 Benny Plunket                $85.70            $0.00               $85.70
 494975 Hildy Edyson                 $85.70            $0.00               $85.70
 497227 Ashleigh O'Bruen             $85.70            $0.00               $85.70
 498160 Martino Boram                $85.70            $0.00               $85.70
```

10. Assume that all units are worth 6 credit points each, calculate each student’s Weighted Average Mark (WAM) and GPA. Please refer to these Monash websites: https://www.monash.edu/exams/results/wam and https://www.monash.edu/exams/results/gpa for more information about WAM and GPA respectively. Do not include NULL, WH or DEF grade in the calculation.

Calculation example for student 14374036 (Claudette Serman):

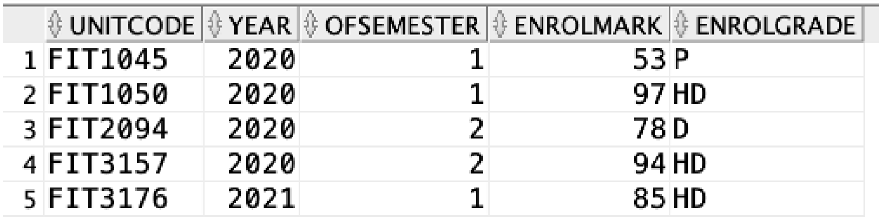


	UNITCODE	YEAR	OFSEMESTER	ENROLMARK	ENROLGRADE
1	FIT9132	2019	1	56	P
2	FIT9136	2019	1	16	N
3	FIT9136	2019	2	81	HD
4	FIT9137	2019	2	77	D
5	FIT5196	2020	1	64	C

WAM = (56x6 + 16x6 + 81x6 + 77x6 + 64x6)/(6+6+6+6+6) = 58.80

GPA = (1x6+ 0.3x6 + 4x6 + 3x6 + 2x6)/(6+6+6+6+6) = 2.06

Calculation example for student 23545528 (Benny Plunket):



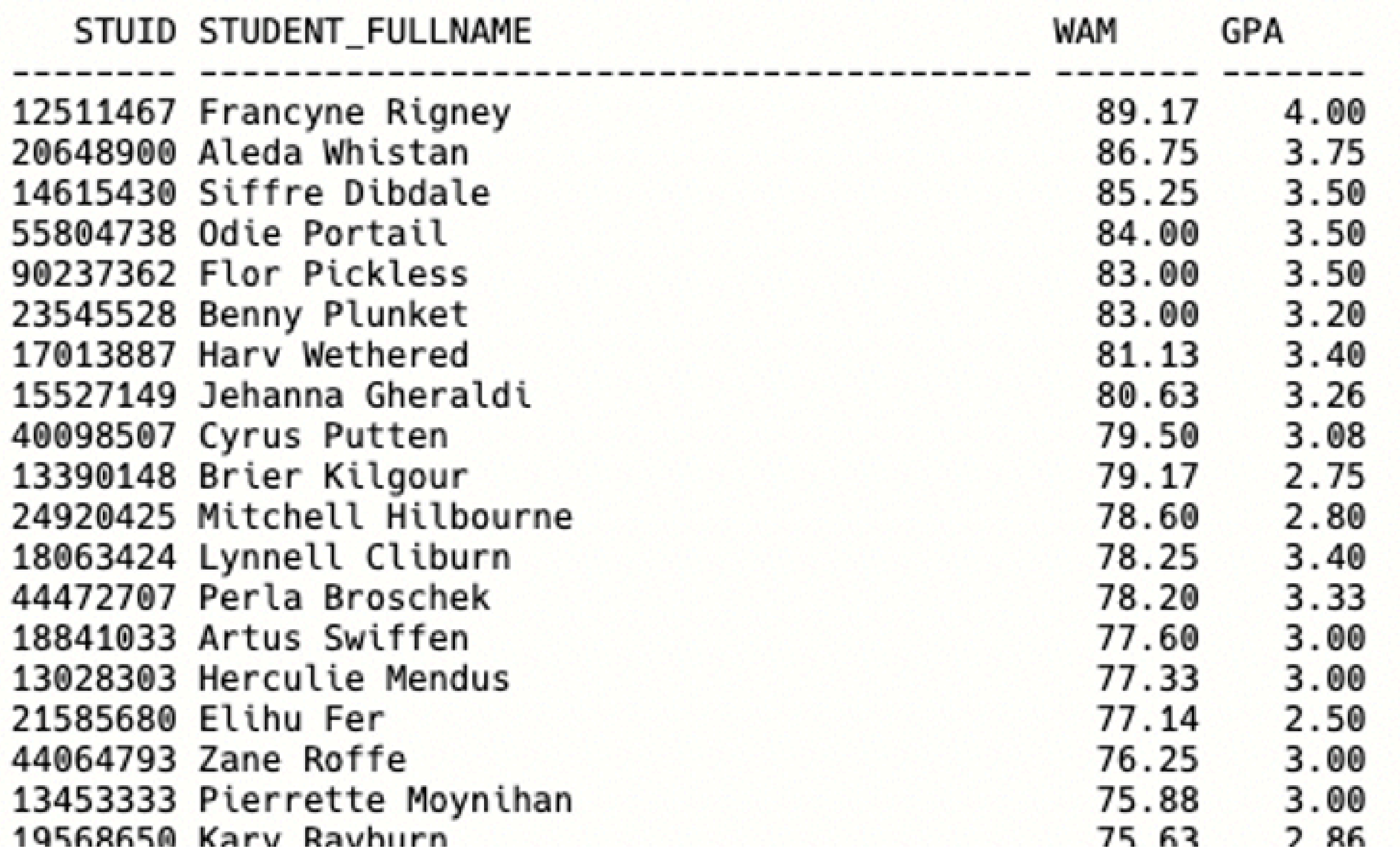
	UNITCODE	YEAR	OFSEMESTER	ENROLMARK	ENROLGRADE
1	FIT1045	2020	1	53	P
2	FIT1050	2020	1	97	HD
3	FIT2094	2020	2	78	D
4	FIT3157	2020	2	94	HD
5	FIT3176	2021	1	85	HD

WAM = (53x3 + 97x3 + 78x6 + 94x6 + 85 x 6)/(3+3+6+6+6) = 83.00

GPA = (1x6 + 4x6+ 3x6 + 4x6 + 4x6)/(6+6+6+6+6) = 3.20

Include student id, student full name (in a 40 characters wide column headed student_fullname), WAM and GPA in the display. Order the list by descending order of WAM then descending order of GPA. If two students have the same WAM and GPA, order them by their respective id.

Only some data shown:



```
   STUID STUDENT_FULLNAME                                  WAM     GPA
-------- ---------------------------------------- ------- -------
12511467 Francyne Rigney                                 89.17    4.00
20648900 Aleda Whistan                                   86.75    3.75
14615430 Siffre Dibdale                                  85.25    3.50
55804738 Odie Portail                                    84.00    3.50
90237362 Flor Pickless                                   83.00    3.50
23545528 Benny Plunket                                   83.00    3.20
17013887 Harv Wethered                                   81.13    3.40
15527149 Jehanna Gheraldi                                80.63    3.26
40098507 Cyrus Putten                                    79.50    3.08
13390148 Brier Kilgour                                   79.17    2.75
24920425 Mitchell Hilbourne                              78.60    2.80
18063424 Lynnell Cliburn                                 78.25    3.40
44472707 Perla Broschek                                  78.20    3.33
18841033 Artus Swiffen                                   77.60    3.00
13028303 Herculie Mendus                                 77.33    3.00
21585680 Elihu Fer                                       77.14    2.50
44064793 Zane Roffe                                      76.25    3.00
13453333 Pierrette Moynihan                              75.88    3.00
19568650 Kary Rayburn                                    75.63    2.86
```

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.

Oracle SQL Common Functions