COLUMBIA UNIVERSITY | MAILMAN SCHOOL of PUBLIC HEALTH
BIOSTATISTICS

In-Class Assignment 17

Self-Joins, Full Joins, Cross Joins, and USING Syntax

Follow the instructions below and answer the questions that follow. Add your solutions to the SQL script called **In-Class Assignment 17.sql** and submit to Canvas by the deadline listed above. Save your file frequently to avoid losing work!

Instructions:

Complete the queries below using one of the 3 join types listed above. Apply the USING syntax to all joins.

Queries:

1. Using the **weight** schema, display all visit-to-visit weight changes and order the results from greatest to smallest weight change. The result set should look as follows:

participant_id	visit_type	weight	visit_type	weight	diff
1	0	68.04	4	65.77	-2.27
1	8	64.86	12	63.92	-0.94
1	4	65.77	8	64.86	-0.91
2	0	91.63	4	92.08	0.45

Hints:

- Be sure you've completed the data entry element from assignment 15 to ensure your results match the above exactly
- Only display records showing differences of exactly 1 visit (4 consecutive weeks apart)
- You will receive an error if you perform subtractions on visit type values that result in a negative value (due to the field being unsigned), so filter your results accordingly to prevent this error

2. Run the code given to create the schema called **clinics** and its corresponding tables. No primary or foreign keys have been defined, and no referential integrity has been enforced. Write a query involving a FULL JOIN that shows clinic ID, location category, and participant ID. The results should show
   (1) Clinics that have no associated participants
   (2) Participants recorded at clinics that don't exist in the clinics table
   (3) Participants with no clinic information at all

In-Class Assignment 17

3. Run the code given to create the schema called **class** and its corresponding tables. Help create a grading list for our teaching team that assigns students as follows:

    - Anjile: students with first names starting with A-H
    - Ifrah: students with first names starting with I-P
    - Charlene: students with first names starting with Q-Z

    Create a CTE that creates the grading pairs and call it **grading list**

    Hints:

    - Utilize the following syntax as an example

        var_name REGEXP '^[A-D]'

        This returns values of column var_name that begin with letters A-D.

    Then, write a query to determine whether the grading is evenly split between the teaching team (i.e., whether they are grading a similar number of students). Call the variable showing the number of students per teaching team member **student_count.**

    Answer the following questions in Canvas:

    1. Does it look like grading is evenly split? If not, how would you redistribute the grading?