OPTIONAL BONUS LAB CHALLENGE – **window functions**

You can use [these slides](https://docs.google.com/presentation/d/1ZUWC263bu1QfS3yu82HdlpjShmtK45-qHQFWuVIXJJg/edit#slide=id.ge0b9ad2165_0_0) for reference material, as well as the [window functions cheat sheet](https://github.com/student-IH-labs-and-stuff/BCNDATA0122/blob/main/ClassMaterials/Week3/Window_Functions_Cheat_Sheet.pdf)

SETUP

Open [sql query](https://github.com/student-IH-labs-and-stuff/BCNDATA0122/blob/main/ClassMaterials/Week3/ranking%20student.sql) in mysqlworkbench, run the DDL commands to create the ExamResult table in the Exams database.

#Challenges :

#A : use Row\_Number() function get a row number for students and subjects as per their marks- highest to lowest

Table

Description automatically generated

#B : Use RANK() function to specify rank for each row in the result set. We have student results for multiple subjects. We want to rank the result for each student as per their marks in the subjects. For example, student Isabella got her highest marks in French and her lowest marks in German

Table

Description automatically generated

#C : Use RANK() function to specify rank for each row in the result set by the marks alone. sort by the rank column

Table

Description automatically generated

#D : Use DENSE\_RANK() function to specify rank for each row in the result set by the marks alone - so that no rank numbers are skipped in the resulting column. sort by the rank column

Table

Description automatically generated

#E : Use DENSE\_RANK() function to specify the rank each student has achieved in each subject versus the other students, eg Isabella came first in French, 2nd in German, Science etc

Table

Description automatically generated

#F: Use Row\_Number() and a subquery to return the top position 1-3 students, a) first overall by grades, and then b) for just the european languages

Example for (b)

Table

Description automatically generated

NOTE you can also combine row\_number with LIMIT to return the top 3 in either scenario and would not require a sub query. Give it a try!