**3 separate tables** ->

* Projects
* Tasks
* Teams

Each table **except Teams** must contain **5X5** rows and columns.

1) Declare a **CTE (Common Table Expression)** -> Each project must have tasks\_assigned and a subset of tasks\_completed. Therefore, SELECT FROM **project\_name, tasks\_assigned** AS **total\_tasks, tasks\_completed**

2) Top LIMIT = 2,top entries **(member\_name)** that has the **max(tasks\_assigned)** independent of **project\_name** but display it

3) Correlated subquery -> tasks\_assigned EXISTS WHERE due\_date < AVG (due\_date)

4) Subquery -> SELECT FROM project\_name HAVING max(budget)

5) Query -> %age of each **project\_name** completed = (**tasks\_completed/tasks\_assigned**)\*100

6) SELECT FROM **member\_name, tasks\_assigned** AS **task\_name, COUNT(tasks\_assigned)** and SELECT FROM **tasks\_assigned** ORDER BY **member\_name**

7) due date -> **today() + 15** && **tasks\_assigned** =TRUE&& **tasks\_completed** =FALSE**, REPLACE member\_name WITH** 'Team Lead'

8) SELECT **project\_ids, project\_name, task\_name** WHERE task record does not exist

===============================================================================

**Table\_name:** Model\_Training

**Column\_name:** training\_id, project\_id, model\_name, accuracy, training\_date

**Query ->** SELECT **model\_name, accuracy** WHERE **project\_id** WITH **max(accuracy)** /\*for each project\*/

==============================================================================

**Table\_name:** Data\_Sets

**Column\_name:** dataset\_id, project\_id, dataset\_name, size\_gb, last\_updated

**Query ->** SELECT **project\_id** WHERE **size\_gb > 10** && **last\_updated < today() - 30**