SQL- Intermediate (Part 2)

1.

The Report

https://www.hackerrank.com/challenges/the-report/problem?isFullScreen=true

Soln :

Column	Type
ID	Integer
Name	String
Marks	Integer

Student

Grade	Min_Mark	Max_Mark
1	0	9
2	10	19
3	20	29
4	30	39
5	40	49
6	50	59
7	60	69
8	70	79
9	80	89
10	90	100

Grades

>= =<

```
SELECT
CASE
  WHEN Grades.Grade > 7 THEN Students.Name
  ELSE NULL
END AS Student-Name,
Grades.grade,
Student.Marks
FROM Students
JOIN Grades ON Students.Marks >= Grades.Min-Mark
        AND Students.Marks <= Grades.Max-Mark
ORDER BY
Grades.Grade DESC
 CASE
     WHEN Grades.Grade > 7 THEN Students.Name
     ELSE Students.Marks
END
```

condition given in question

Join on Grades Table such that marks between min and max

Grades by descending

grade greater than 7 sort by Name

else with marks in ascending order

Ollivander's Journey

2.

https://www.hackerrank.com/challenges/harry-potter-and-wands/problem?isFullScreen=true

Requirements:- print id, age, coins_needed, power of wands sorted in descending order of power if same sort in descending age

Column	Type
id	Integer
code	Integer
coins_needed	Integer
power	Integer

Wands

Column	Type
code	Integer
age	Integer
is_evil	Integer

Wands_Property

Main condn:- minimum number of gold galleons needed to buy each non evil wand of high power and age.

Soln:

```
SELECT
    w1.id, wp.age, w1.coins_needed, w1.power
FROM
    Wands w1
JOIN
    Wands_Property wp on w1.code = wp.code
JOIN
    (SELECT w.code, MIN(w.coins_needed) as min_coins
    FROM
        Wands w
    JOIN
        Wands_Property wp ON w.code = wp.code
    WHERE
        wp.is_evil = 0
    GROUP BY
        w.code, wp.age, w.power) as min_wands
    ON w1.code = min_wands.code AND
    w1.coins_needed = min_wands.min_coins
WHERE
    wp.is_evil = 0
ORDER BY w1.power DESC, wp.age DESC;
```

minimum coins needed

ordering as in requirement

Contest Leaderboard

(3.)

https://www.hackerrank.com/challenges/contest-leaderboard/problem?isFullScreen=true

hacker_id, name and total score of hackers ordered in descending order

same total score ⇒ ascending hacker_id

Exclude all with 0 score

Total score is sum of all maximum scores for each challenge

Important Note before confusion arises :-

Difference bet^n WHERE and HAVING and when to use them.

WHERE	HAVING
* filter rows without grouping is applied to individual rows.	* 'HAVING' clause is used to filter groups in grouped queries ('GROUP BY')
* WHERE cannot have aggregate fn's because it filters rows before aggregation.	* Used in conjunction with aggregation fn's as it can contain aggregate fn because it filter the rows on basis of these fns.

Aggregation fn : SUM(), AVG(), MIN(), MAX(), COUNT()

SELECT

max_score_table.hacker_id,

Hackers.name,

sum(max_score.table.max_score) AS total_score ↳ totalling score

FROM (

```
SELECT Submissions.hacker_id,
       Submissions.challenge_id,
       MAX(Submissions.score) as max_score
FROM Submissions
GROUP BY Submissions.hacker_id, Submissions.challenge_id)
```

→ logic for creating table having maximum score for each challenge for each hacker

AS max_score_table

JOIN Hackers ON Hackers.hacker_id = max_score_table.hacker_id

GROUP BY Hackers.hacker_id, Hackers.name } → both since both are cndn distinct

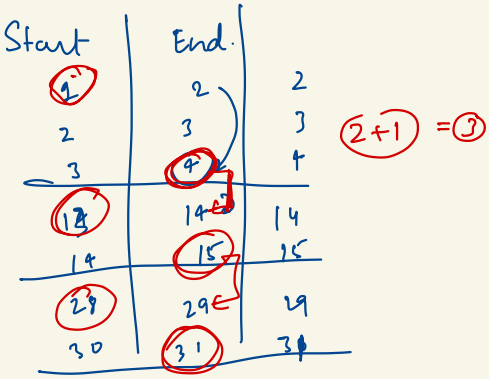
HAVING sum(max_score_table.max_score) > 0

ORDER BY total score DESC, hackers.hacker_id ASC

score descending => if same id ascending.

SQL - Project Planning

https://www.hackerrank.com/challenges/sql-projects/problem?isFullScreen=true



Some functionalities before moving towards to the problem:

Aggregate functions such as
(SUM, COUNT, AVG, MIN, MAX)
GROUP BY is needed here

* Use GROUP BY when you need summary data

Window Functions such as
(OVER, Partition, order, etc)

* Use PARTITION BY when you want to maintain and analyze individual raw details

For above question

- When start_date is not in end_date column that is project start date (1)
- When end_date is not in start_date column that is project end date (2)

```
SELECT  Start_Date, End_Date FROM
(Select  Start_date
        ROW_NUMBER() OVER( ORDER BY Start_Date) AS rowno

 FROM  Projects
 WHERE  Start_date  NOT IN
  (SELECT  End_date  FROM  Projects )) as Start_Date_Table.
JOIN
  (SELECT  End_Date,
          ROW_NUMBER( ) OVER( ORDER BY End_dates)
           AS rowno
   FROM  Project
   WHERE  End_date  NOT IN
   (SELECT Start_date FROM Projects))  AS  End_Date_Table
ON  Start_date_table.rowno =  End_Date_Table.rowno
ORDER BY  (End_date - Start_date)
```

Start_date_table ①

End_date_table ②

↳ Helps in joining with table

ordering by duration of dates

Placements

https://www.hackerrank.com/challenges/placements/problem?isFullScreen=true

Students		Friends		Packages	
ID	Name	ID	Friend_ID	ID	Salary

* Guaranty:- No two students got same salary.

Query best friends who got higher salary than them names in order of salary.

```
SELECT  Students.name   FROM  students
JOIN    Friends         ON    Students.ID = Friends.ID
JOIN    Packages P1     ON    Students.ID = P1.ID        → Student Salary
JOIN    Packages P2     ON    Friends.Friend_ID = P2.ID  → friend salary
WHERE   P2.salary > P1.salary   → friend salary > student salary
ORDER BY  P2.salary;
          friend salary
```