# 1

SELECT

l.P\_ID, l.Dev\_ID, p.PName, l.Difficulty

FROM

level\_details2 AS l

JOIN

player\_details AS p ON l.P\_ID = p.P\_ID

WHERE

l.Level = 0;

# 2

SELECT

p.L1\_Code, AVG(l.Kill\_Count)

FROM

player\_details AS p

JOIN

level\_details2 AS l ON p.P\_ID = l.P\_ID

WHERE

l.lives\_earned = 2 AND l.stages\_crossed > 3

GROUP BY

p.L1\_Code;

# 3

SELECT

Difficulty, SUM(Stages\_crossed) AS total\_stages\_crossed

FROM

level\_details2

WHERE

Level = 2 AND Dev\_ID IN ('zm\_013', 'zm\_015', 'zm\_017')

GROUP BY

Difficulty

ORDER BY

total\_stages\_crossed DESC;

# 4

SELECT

P\_ID, COUNT(DISTINCT DATE(TimeStamp)) AS total\_unique\_dates

FROM

level\_details2

GROUP BY

P\_ID

HAVING

COUNT(DISTINCT DATE(TimeStamp)) > 1;

# 5

SELECT

P\_ID, SUM(Kill\_Count) AS total\_kill\_count, Level

FROM

level\_details2

WHERE

Kill\_Count > (

SELECT

AVG(Kill\_Count)

FROM

level\_details2

WHERE

Difficulty = 'Medium'

)

GROUP BY

P\_ID, Level;

# 6

SELECT

l.Level,

p.L1\_Code,

p.L2\_Code,

SUM(l.Lives\_Earned) AS Total\_Lives\_Earned

FROM

level\_details2 l

JOIN

player\_details p ON l.P\_ID = p.P\_ID

WHERE

l.Level <> 0

GROUP BY

l.Level, p.L1\_Code, p.L2\_Code

ORDER BY

l.Level ASC;

# 7

WITH Ranked\_Scores AS (

SELECT

Dev\_ID,

Score,

Difficulty,

ROW\_NUMBER() OVER (PARTITION BY Dev\_ID ORDER BY Score DESC) AS Score\_Rank

FROM

level\_details2

)

SELECT

Dev\_ID,

Score,

Difficulty,

Score\_Rank

FROM

Ranked\_Scores

WHERE

Score\_Rank <= 3;

# 8

SELECT

Dev\_ID,

MIN(TimeStamp) AS first\_login

FROM

level\_details2

GROUP BY

Dev\_ID;

# 9

WITH Ranked\_Scores AS (

SELECT

Difficulty,

Dev\_ID,

RANK() OVER (PARTITION BY Difficulty ORDER BY Score DESC) AS Score\_Rank

FROM

level\_details2

)

SELECT

Difficulty,

Dev\_ID,

Score\_Rank

FROM

Ranked\_Scores

WHERE

Score\_Rank <= 5;

# 10

WITH First\_Login\_Per\_Player AS (

SELECT

P\_ID,

Dev\_ID,

MIN(TimeStamp) AS first\_login\_datetime

FROM

level\_details2

GROUP BY

P\_ID, Dev\_ID

)

SELECT

P\_ID,

Dev\_ID,

first\_login\_datetime

FROM

First\_Login\_Per\_Player

WHERE

first\_login\_datetime = (

SELECT MIN(first\_login\_datetime)

FROM First\_Login\_Per\_Player AS InnerTable

WHERE InnerTable.P\_ID = First\_Login\_Per\_Player.P\_ID

);

# 11

# Using Window function:

SELECT

P\_ID,

TimeStamp,

SUM(Kill\_Count) OVER (PARTITION BY P\_ID ORDER BY TimeStamp) AS total\_kills\_so\_far

FROM

level\_details2;

# Without using Window function;

SELECT

ld.P\_ID,

ld.TimeStamp,

SUM(ld2.kill\_count) AS total\_kills\_so\_far

FROM

level\_details2 ld

JOIN

level\_details2 ld2 ON ld.P\_ID = ld2.P\_ID AND ld.TimeStamp >= ld2.TimeStamp

GROUP BY

ld.P\_ID, ld.TimeStamp

ORDER BY

ld.P\_ID, ld.TimeStamp;

# 12

WITH Ranked\_Stages AS (

SELECT

P\_ID,

TimeStamp,

Stages\_crossed,

ROW\_NUMBER() OVER (PARTITION BY P\_ID ORDER BY TimeStamp DESC) AS row\_num

FROM

level\_details2

)

SELECT

P\_ID,

TimeStamp,

SUM(stages\_crossed) OVER (PARTITION BY P\_ID ORDER BY TimeStamp ASC) AS cumulative\_sum

FROM

Ranked\_Stages

WHERE

row\_num > 1;

# 13

WITH Ranked\_Scores AS (

SELECT

Dev\_ID,

P\_ID,

SUM(Score) AS total\_scores,

RANK() OVER (PARTITION BY Dev\_ID ORDER BY SUM(Score) DESC) AS Score\_Rank

FROM

level\_details2

GROUP BY

Dev\_ID, P\_ID

)

SELECT

Dev\_ID,

P\_ID,

total\_scores

FROM

Ranked\_Scores

WHERE

Score\_Rank <= 3

ORDER BY

Dev\_ID, Score\_Rank;

# 14

WITH sum\_of\_scores AS (

SELECT

P\_ID,

SUM(Score) AS total\_score

FROM

level\_details2

GROUP BY

P\_ID

)

SELECT

P\_ID,

total\_score

FROM

sum\_of\_scores

WHERE

total\_score > 0.5 \* (SELECT AVG(total\_score) FROM sum\_of\_scores);

# 15

call Get\_Top\_Headshot\_Count(3);