

TD 7

1. Schema:

games (**gamescore** integer)

Each record contains a game score achieved in Tetris.

Chaque ligne contient un score de jeu obtenu dans Tetris.

Write a query to list the rank and the gamescore at that rank. Assume all game scores are distinct.
Écrivez une requête pour lister le rang et le score de jeu à ce rang. Supposons que tous les scores de jeu sont distincts.

Example:

<i>gamescore</i>
53
11
99
45
59
46

(a) input

<i>games</i>	<i>rank</i>
53	4
11	1
99	6
45	2
59	5
46	3

(b) output

Solution.

```
SELECT A.gamescore, count(*)
FROM games as A CROSS JOIN games as B
WHERE B.gamescore <= A.gamescore
GROUP BY A.gamescore
```

2. Schema:

exam (**time** integer, **questions** integer)

Each record contains an hour, and the number of questions done during that hour.

Chaque ligne contient une heure et le nombre de questions complétées pendant cette heure.

Write a query to list, for each hour, the **total** number of questions completed after that hour has passed.
Écrivez une requête pour lister, pour chaque heure, le nombre total de questions complétées après l'expiration de cette heure.

Example:

<i>time</i>	<i>questions</i>
1	5
2	2
3	8
4	14
5	0
6	1

(c) input

<i>time</i>	<i>questions</i>
1	5
2	7
3	15
4	29
5	29
6	30

(d) output

Solution.

```
SELECT E1.time, sum(E2.questions)
FROM exam as E1, exam as E2
WHERE E2.time <= E1.time
GROUP BY E1.time
```

3. Schema:

company (name text)

Each record contains the name of a person.

Chaque ligne contient le nom d'une personne.

Write a query to list the names that appear more than once.

Écrivez une requête pour lister les noms qui apparaissent plus d'une fois.

Example:

<i>name</i>
btasdf
asdfasdf
asdfsad
jfhghd
jjkhasdf
zXcvm
asdfasdf

(e) input

<i>name</i>
asdfasdf

(f) output

Solution.

```
SELECT name
FROM company
GROUP BY name
HAVING count(*) > 1
```

4. Schema:

customer (**id** integer, **name** text)

Each record contains the name of a person, and their unique **id**.
Chaque ligne contient le nom d'une personne et son **id** unique.

orders (**id** integer, **menu_item** integer)

Each record contains the **id** and the menu item that that person has ordered.
Chaque ligne contient le **id** et l'élément de menu que cette personne a commandé.

Write a query to list the names of the people that have not ordered anything.
Écrivez une requête pour lister les noms des personnes qui n'ont rien commandé.

Example:

<i>id</i>	<i>name</i>
1	jack
5	jill
32	john
62	james
3	jack
55	asdf
10	asdfasdf

(g) customer

<i>id</i>	<i>menu_item</i>
62	4
55	22
3	7
10	7

(h) orders

<i>name</i>
jack
jill
john

(i) output

Solution.

```
SELECT name
FROM customer
WHERE id NOT IN (SELECT id
                  FROM orders);
```

Second solution:

```
(SELECT name
 FROM customer)

EXCEPT

(SELECT distinct name
 FROM customer INNER JOIN orders USING (id))
```

5. Schema:

temperature (**day_no** integer, **temp** integer)

Each record contains the day of the month (1 – 30) and its temperature.

Chaque ligne contient le jour du mois (1 – 30) et sa température.

Write a query to list the days where the temperature is higher than that of the *previous* day.

Écrivez une requête pour lister les jours où la température est supérieure à celle du jour *précédent*.

Example:

<i>day_no</i>	<i>temp</i>
1	27
2	30
3	31
4	30
5	28
6	29
7	27
8	32

(j) temperature

<i>day_no</i>
2
3
6
8

(k) output

Solution.

```
SELECT A.day_no
FROM temperature AS A CROSS JOIN temperature AS B
WHERE B.temp < A.temp AND B.day_no = A.day_no-1;
```

Solution 2:

```
SELECT A.day_no
FROM temperature AS A
WHERE A.temp >= (SELECT MAX(B.temp)
                  FROM temperature AS B
                  WHERE B.temp < A.temp
                  AND B.day_no = A.day_no-1);
```

6. Schema:

server (*id* integer, *device* integer, *login_date* date)

Each record contains login student *id*, laptop number they used, and the date they logged in.

Chaque ligne contient l'identifiant de connexion de l'étudiant, le numéro de l'ordinateur portable qu'il a utilisé et la date à laquelle il s'est connecté.

Write a query to list, for each student, the first date they logged in.

Écrivez une requête pour lister, pour chaque élève, la première date de sa connexion.

Example:

<i>id</i>	<i>device</i>	<i>login_date</i>
463	4	30-01-2023
422	1	20-01-2023
463	2	11-01-2023
834	6	16-01-2023
585	8	22-01-2023
936	1	23-01-2023
834	4	08-01-2023
463	3	21-01-2023

(l) server

<i>id</i>	<i>login_date</i>
463	11-01-2023
422	20-01-2023
834	08-01-2023
936	23-01-2023
585	22-01-2023

(m) output

Solution.

```
SELECT id, min(login_date)
FROM server
GROUP BY id
```

```
SELECT A.id, A.login_date
FROM server AS A
WHERE A.login_date <= ALL (SELECT B.login_date
                           FROM server AS B
                           WHERE B.id = A.id)
```

7. Schema:

server (*id* integer, *device* integer, *login_date* date)

Each record contains login student *id*, laptop number they used, and the date they logged in.

Chaque ligne contient l'identifiant de connexion de l'étudiant, le numéro de l'ordinateur portable qu'il a utilisé et la date à laquelle il s'est connecté.

Write a query to list, for each student, the *device* they used the first time they logged in.

Écrivez une requête pour lister, pour chaque élève, l'appareil qu'il a utilisé la première fois qu'il s'est connecté.

Example:

<i>id</i>	<i>device</i>	<i>login_date</i>
463	4	30-01-2023
422	1	20-01-2023
463	2	11-01-2023
834	6	16-01-2023
585	8	22-01-2023
936	1	23-01-2023
834	4	08-01-2023
463	3	21-01-2023

(n) server

<i>id</i>	<i>device</i>
463	2
422	1
834	4
936	1
585	8

(o) output

Solution.

```
SELECT A.id, A.device
FROM server AS A
WHERE A.login_date <= ALL (SELECT B.login_date
                           FROM server AS B
                           WHERE B.id = A.id)
```

Solution 2:

```
SELECT A.id, A.device
FROM server AS A
WHERE A.login_date IN (SELECT min(B.login_date)
                      FROM server AS B
                      WHERE B.id = A.id)
```

8. Schema:

employee (*id* integer, *name* text, *manager_id* integer)

Each record contains employee *id*, *name* and the *id* of the manager.

Chaque ligne contient l'identifiant de l'employé, le nom et l'identifiant du responsable.

bonus (*id* integer, *amount* integer)

Each record contains employee *id*, and the bonus they received this year.

Chaque ligne contient l'identifiant de l'employé et la prime qu'il a reçue cette année.

Write a query to list the name and bonus of all employees who received a bonus of at most 1000.

Écrivez une requête pour lister le nom et le bonus de tous les employés qui ont reçu un bonus d'au plus 1000.

Example:

<i>id</i>	<i>name</i>	<i>manager_id</i>
2	john	4
6	james	9
4	jamil	9
9	jean	null
7	joe	2

(p) employee

<i>id</i>	<i>amount</i>
4	500
9	2000

(q) bonus

<i>name</i>	<i>prime</i>
john	null
jamil	500
james	null
joe	null

(r) output

Solution.

```
SELECT employee.name, amount AS prime
FROM employee LEFT OUTER JOIN bonus ON employee.id = bonus.id
WHERE bonus.amount <= 1000 OR amount IS null;
```


9. Schema:

tree (node_id integer, parent_id integer)

Each record contains a binary tree node **id**, and the **id** of its parent.

Chaque ligne contient un nœud d'arbre binaire **id** et le **id** de son parent.

Write a query to list, for each node, if it is a 'Root' node, or an 'Inner' node or a 'Leaf' node.

Écrivez une requête pour lister, pour chaque nœud, s'il s'agit d'un nœud « Racine », d'un nœud « Interne » ou d'un nœud « Feuille ».

Example:

<i>node_id</i>	<i>parent_id</i>
1	null
5	1
7	1
2	5
3	7

(s) tree

<i>node_id</i>	<i>type</i>
1	Root
5	Inner
7	Inner
2	Leaf
3	Leaf

(t) output

Solution.

```
SELECT node_id, 'Root' AS type
FROM tree
WHERE parent_id IS null

UNION

SELECT node_id, 'Inner' AS type
FROM tree
WHERE node_id IN (SELECT parent_id FROM tree)
AND parent_id IS NOT null

UNION

SELECT node_id, 'Leaf' AS type
FROM tree
WHERE node_id NOT IN (SELECT parent_id
FROM tree
WHERE parent_id IS NOT null)
```

Solution 2:

```
SELECT node_id,
CASE
WHEN parent_id IS null THEN 'Root'
WHEN parent_id IS NOT null
AND node_id IN (SELECT parent_id FROM tree) THEN 'Inner'
ELSE 'Leaf'
END AS type
FROM tree;
```