TP 15

SQL
Sources :

Proposition de corrigé

Q 1 :

Commencez par récupérer le rapport de scène de crime correspondant dans la base de données du service de police.

```
SELECT description FROM crime_scene_report
WHERE date = '20180115' AND type = 'murder' AND city = 'SQL City'
```

qui renvoie :

```
"Security footage shows that there were 2 witnesses. The first witness lives at the last
house on ""Northwestern Dr"". The second witness, named Annabel, lives somewhere on ""Franklin
Ave""."
```

On creuse alors la piste du témoin 1 :

```
SELECT * FROM person
WHERE address_street_name = 'Northwestern Dr'
ORDER BY address_number DESC
```

qui renvoie :

id	name	license_id	address_number	address_street_name	ssn
14887	Morty Schapiro	118009	4919	Northwestern Dr	111564949
17729	Lasonya Wildey	439686	3824	Northwestern Dr	917817122
53890	Sophie Tiberio	957671	3755	Northwestern Dr	442830147
...	...	...	...	...	...

Si on ne veut que le premier on peut indiquer une limite (bonus) :

```
SELECT * FROM person
WHERE address_street_name = 'Northwestern Dr'
ORDER BY address_number DESC LIMIT 1
```

qui renvoie :

id	name	license_id	address_number	address_street_name	ssn
14887	Morty Schapiro	118009	4919	Northwestern Dr	111564949

En ce qui concerne le deuxième témoin :

```
SELECT * FROM person
WHERE name like '%Annabel%' AND address_street_name = 'Franklin Ave'
```

qui renvoie :

Le témoignage de ces deux témoins donne avec le mot-clé `in` (mais c'est un exemple), à l'aide de la requête :

id	name	license_id	address_number	address_street_name	ssn
16371	Annabel Miller	490173	103	Franklin Ave	318771143

```
SELECT * FROM interview where person_id in (14887, 16371)
```

donne :

- 14887 - I heard a gunshot and then saw a man run out. He had a "Get Fit Now Gym" bag. The membership number on the bag started with "48Z". Only gold members have those bags. The man got into a car with a plate that included "H42W".
- 16371 - I saw the murder happen, and I recognized the killer from my gym when I was working out last week on January the 9th.

On s'attaque aux personnes du côté de la gym, avec le bon sac, le statut est présent le 09 janvier.

```
SELECT m.person_id, m.name FROM get_fit_now_member AS m
JOIN get_fit_now_check_in AS c ON m.id=c.membership_id
WHERE m.id like '48Z%' and m.membership_status = 'gold'
AND c.check_in_date = '20180109'
```

ce qui renvoie deux candidats :

person_id	name
28819	Joe Germuska
67318	Jeremy Bowers

Il faut donc croiser avec les plaques minéralogiques.

```
SELECT p.id, p.name FROM person as p JOIN drivers_license as d
ON d.id = p.license_id
WHERE d.plate_number like '%H42W%'
```

qui nous renvoie 3 possibilités :

id	name
51739	Tushar Chandra
67318	Jeremy Bowers
78193	Maxine Whitely

Remarquons la possibilité de conclure l'histoire sur ce personnage :

```
INSERT INTO solution VALUES (1, "Jeremy Bowers");
SELECT value FROM solution;
```

qui renvoie :

```
Congrats, you found the murderer! But wait, there's more... If you think you're up for
a challenge, try querying the interview transcript of the murderer to find the real villian
behind this crime. If you feel especially confident in your SQL skills, try to complete
this final step with no more than 2 queries.
```

Le tueur est donc : `Jeremy Bowers`. Regardons alors ce qu'il a à dire pour sa défense :

```
SELECT transcript FROM interview WHERE person_id = 67318
```

ce qui donne :

```
I was hired by a woman with a lot of money. I don't know her name but I know she's around
5'5" (65") or 5'7" (67"). She has red hair and she drives a Tesla Model S. I know that she
attended the SQL Symphony Concert 3 times in December 2017.
```

Du côté de la voiture, du genre et de la couleur de cheveux :

```
SELECT * FROM drivers_license
WHERE gender = 'female' AND hair_color = 'red'
AND car_make = 'Tesla' AND car_model = 'Model S'
AND height BETWEEN 65 AND 67
```

id	age	height	eye_color	hair_color	gender	plate_number	car_make	car_model
202298	68	66	green	red	female	500123	Tesla	Model S
291182	65	66	blue	red	female	08CM64	Tesla	Model S
918773	48	65	black	red	female	917UU3	Tesla	Model S

nous permet d'obtenir 3 numéros :

Est-ce que l'une d'entre elles est riche?

```
SELECT i.annual_income, p.id, p.name, d.age FROM income AS i
JOIN person AS p
ON i.ssn = p.ssn
JOIN drivers_license AS d
ON p.license_id = d.id
WHERE d.id in (202298,291182,918773)
ORDER BY i.annual_income DESC
```

Oui!

annual_income	id	name	age
310000	99716	Miranda Priestly	68
278000	78881	Red Korb	48

Mais qui est allé au concert de musique 3 fois au mois de décembre?

```
SELECT person_id, COUNT(*) AS nbFois
FROM facebook_event_checkin
WHERE event_name = 'SQL Symphony Concert'
AND date between 20171201 and 20171231
GROUP BY person_id
HAVING nbFois = 3
```

person_id	nbFois
24556	3
99716	3

Je crois que Miranda est coincée!

```
INSERT INTO solution VALUES (1, "Miranda Priestly");
SELECT value FROM solution;
```

Et félicitations, vous avez trouvé le cerveau du meurtre! Tout le monde à SQL City vous salue comme le plus grand détective SQL de tous les temps. Il est temps de sabrer le champagne!

Remarque :

Il est demandé de le faire en deux requêtes ce qui n'est pas très pédagogique, mais bien sûr c'est possible. On peut proposer cela [1] :

```
WITH red_haired_tesla_drivers AS (
    SELECT id AS license_id
    FROM drivers_license
    WHERE gender = 'female' AND hair_color = 'red'
    AND car_make = 'Tesla' AND car_model = 'Model S'
    AND height BETWEEN 65 AND 67
), rich_suspects AS (
    SELECT person.id AS person_id, name, annual_income
    FROM red_haired_tesla_drivers AS rhtd
    LEFT JOIN person ON rhtd.license_id = person.license_id
    LEFT JOIN income ON person.ssn = income.ssn
), symphony_attenders AS (
    SELECT person_id, COUNT(*) AS nbFois
    FROM facebook_event_checkin
```

1. https://gist.github.com/bearloga/cfc8099223d1dace2604c8737dcbb4c3

```
    WHERE event_name = 'SQL Symphony Concert'
AND date between 20171201 and 20171231
    GROUP BY person_id
    HAVING nbFois = 3
)
SELECT name, annual_income
FROM rich_suspects
JOIN symphony_attenders ON rich_suspects.person_id = symphony_attenders.person_id
```

1 Solution 2

Q 2 :
La requête

```
SELECT count(*) FROM person;
```

donne :

count(*)
10011

Q 3 :
La requête

```
SELECT * FROM person LIMIT 10;
```

donne

id	name	license_id	address_number	address_street_name	ssn
10000	Christoper Peteuil	993845	624	Bankhall Ave	747714076
10007	Kourtney Calderwood	861794	2791	Gustavus Blvd	477972044
10010	Muoi Cary	385336	741	Northwestern Dr	828638512
10016	Era Moselle	431897	1987	Wood Glade St	614621061
10025	Trena Hornby	550890	276	Daws Hill Way	223877684
10027	Antione Godbolt	439509	2431	Zelham Dr	491650087
10034	Kyra Buen	920494	1873	Sleigh Dr	332497972
10039	Francesco Agundez	278151	736	Buswell Dr	861079251
10095	Leslie Thate	729987	2772	Camellia Park Circle	127944356
10122	Alva Conkel	779002	116	Diversey Circle	148521773

Q 4 :
La requête

```
SELECT DISTINCT type FROM crime_scene_report;
```

donne

type
robbery
murder
theft
fraud
arson
bribery
assault
smuggling
blackmail

Q 5 :
La requête

```
SELECT * FROM person WHERE name = 'Kinsey Erickson'
```

id	name	license_id	address_number	address_street_name	ssn
89906	Kinsey Erickson	510019	309	Northwestern Dr	635287661

donne

Q 6 :
La requête

```
SELECT * FROM crime_scene_report
WHERE type = 'theft'
AND city = 'Chicago';
```

donne

date	type	description	city
20180115	theft	Big Bully stole my lunch money!	Chicago
20170101	theft	‘Yes,’ said Alice, ‘we learned French and music.’	Chicago
20171227	theft	silence, and then another confusion of voices-‘Hold up his head-Brandy	Chicago

Q 7 :
La requête

```
SELECT DISTINCT city
FROM crime_scene_report
WHERE city LIKE 'I%';
```

donne

city
Irving
Indianapolis
Irvine
Inglewood
Independence

Q 8 :
La requête

```
SELECT DISTINCT city
FROM crime_scene_report
WHERE city BETWEEN 'W%' AND 'Z%';
```

donne

city
Wilmington
Waterbury
West Valley City
Winter Haven
Youngstown
Wichita
West Covina
Yakima
Washington
Winston
Westminster
Waco
Yonkers
Warren
Worcester
Waterloo
York

Q 9 :
La requête

```
SELECT max(age) FROM drivers_license;
```

donne

max(age)
89

Q 10 :
La requête

```
SELECT * FROM drivers_license ORDER BY age ASC LIMIT 10
```

donne

id	age	height	eye_color	hair_color	gender	plate_number	car_make	car_model
101255	18	79	blue	grey	female	5162Z1	Lexus	GS
108374	18	63	brown	red	male	X2KE6N	Ford	Escape
112201	18	57	green	green	male	HW66XJ	BMW	325
115674	18	74	blue	blue	female	2OGQIP	Mitsubishi	Diamante
122161	18	73	black	black	male	2H3Y1S	BMW	M5
127288	18	59	black	black	female	A2OYSP	Ford	Freestar
131246	18	57	brown	white	female	VOU6R8	Suzuki	Grand Vitara
141220	18	70	green	blue	male	215AK2	Lexus	LX
152848	18	58	brown	blonde	male	4YOOIK	Mazda	Millenia
160151	18	67	blue	grey	female	O1G724	Mitsubishi	Montero

Q 11 :
La requête

```
SELECT person.name, income.annual_income
FROM income
JOIN person
ON income.ssn = person.ssn
WHERE annual_income > 450000
```

donne

name	annual_income
Claudio Carlan	473100
Felice Prudden	486600
Buena Cosimini	475700
Dianna Eyster	476300
Numbers Cranker	498500
Truman Haaker	489800

Q 12 :
La requête

```
SELECT name, annual_income as income,
gender, eye_color as eyes, hair_color as hair
FROM income i
JOIN person p
ON i.ssn = p.ssn
JOIN drivers_license dl
ON p.license_id = dl.id
WHERE annual_income > 450000
```

donne

name	income	gender	eyes	hair
Claudio Carlan	473100	male	black	brown
Felice Prudden	486600	female	green	green
Buena Cosimini	475700	female	brown	blonde
Dianna Eyster	476300	female	brown	black
Numbers Cranker	498500	male	brown	green
Truman Haaker	489800	male	brown	grey