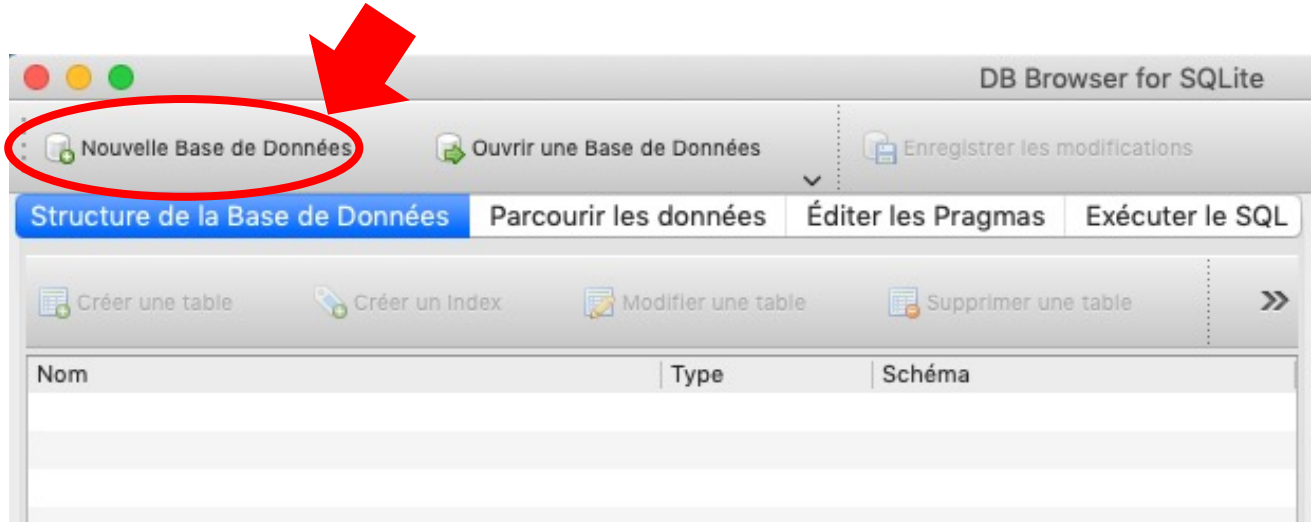
Chapitre 7 - Bases de données

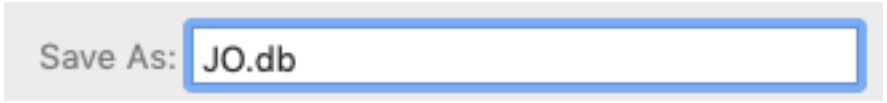
Séance 3 - TP Jeux olympiques de Londres 2012

Partie 1 - Création de la base de données

- Lancer DB Browser for SQLite
- Créer une nouvelle base de données



- Enregistrer sous : **JO.db**



- Lancer Visual Studio Code
- Ouvrir le fichier : **create_JO.sql**

create_JO.sql

```sql
BEGIN TRANSACTION;
DROP TABLE IF EXISTS Sport;
CREATE TABLE Sport
(
    id_sport INTEGER PRIMARY KEY,
    nom_sport VARCHAR(30)
);

DROP TABLE IF EXISTS Discipline;
CREATE TABLE Discipline
(
    id_disc   INTEGER PRIMARY KEY,
    nom_disc  VARCHAR(50),
    type_disc CHAR(1) CHECK (type_disc IN ('I', 'T')), -- u
    id_sport  INTEGER NOT NULL REFERENCES Sport (id_sport)
);

DROP TABLE IF EXISTS Pays;
CREATE TABLE Pays
(
    cio      CHAR(3) PRIMARY KEY, -- code à 3 lettres pour
```

```
389         (294, 'FREESTYLE 48 KG WOMEN', 'I', 40),
390         (295, 'FREESTYLE 63 KG WOMEN', 'I', 40),
391         (296, 'FREESTYLE 72 KG WOMEN', 'I', 40),
392         (297, 'GRECO-ROMAN 55 KG MEN', 'I', 40),
393         (298, 'GRECO-ROMAN 66 KG MEN', 'I', 40),
394         (299, 'GRECO-ROMAN 120 KG MEN', 'I', 40),
395         (300, 'FREESTYLE 66 KG MEN', 'I', 40),
```

```
438         ('CAN', 'CA', 'CAN', 'Canada', 'Ottawa', 'North America'),
439         ('CPV', 'CV', 'CPV', 'Cape Verde', 'Praia', 'Africa'),
440         ('CHI', 'CL', 'CHL', 'Chile', 'Santiago', 'South America'),
441         ('CHN', 'CN', 'CHN', 'China', 'Beijing', 'Asia'),
442         ('CYP', 'CY', 'CYP', 'Cyprus', 'Nicosie', 'Europe'),
443         ('COL', 'CO', 'COL', 'Colombia', 'Bogota', 'South America'),
```

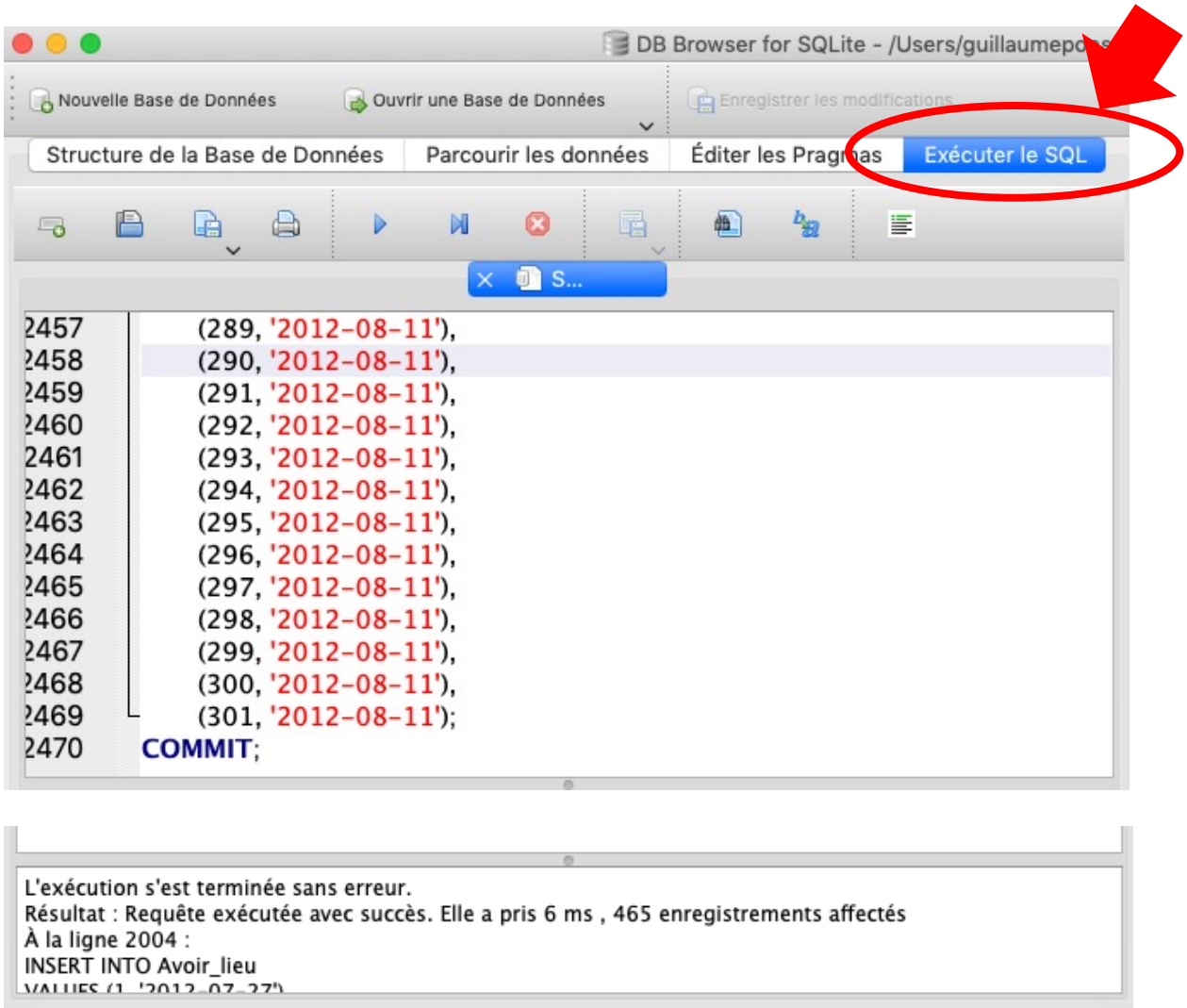
```
1288        (4, 269, 'G'),
1289        (4, 79, 'S'),
1290        (4, 161, 'B'),
1291        (5, 370, 'G'),
1292        (5, 464, 'S'),
1293        (5, 549, 'B'),
1294        (6, 371, 'G'),
1295        (6, 550, 'S'),
```

```
1226        (660, 'Egor', 'MEKHONTCEV', 'RUS', NULL),
1227        (661, 'Adilbek', 'NIYAZYMBETOV', 'KAZ', NULL),
1228        (662, 'Tony', 'ESTANGUET', 'FRA', NULL),
1229        (663, 'Maialen', 'CHOURRAUT', 'ESP', NULL),
1230        (665, 'Maris', 'STROMBERGS', 'LAT', NULL),
1231        (666, 'Chris', 'HOY', 'GBR', NULL),
1232        (667, 'Shuang', 'GUO', 'CHN', NULL),
1233        (668, 'Annette', 'EDMONDSON', 'AUS', NULL),
```

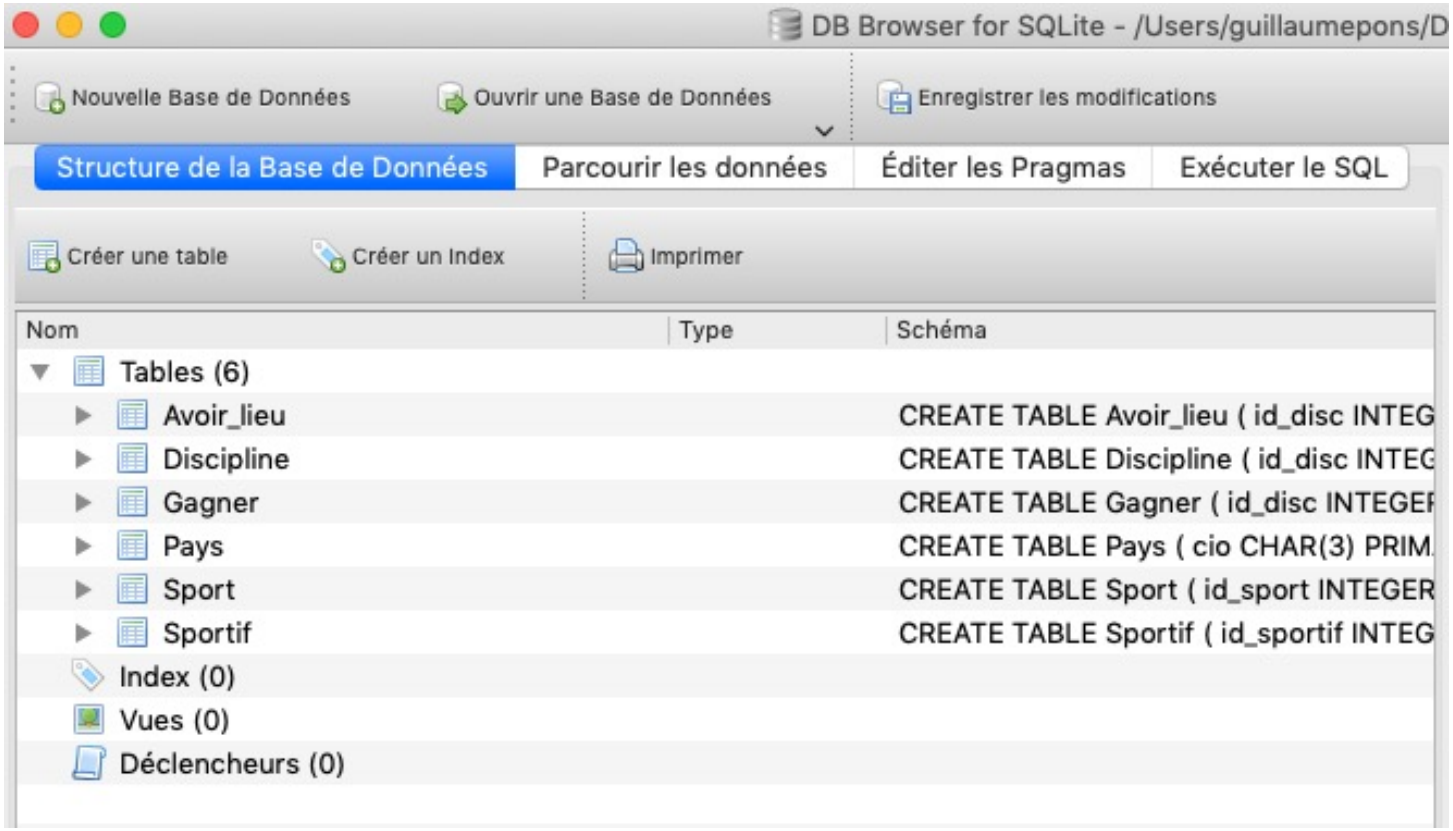
```
2460          (292, '2012-08-11'),
2461          (293, '2012-08-11'),
2462          (294, '2012-08-11'),
2463          (295, '2012-08-11'),
2464          (296, '2012-08-11'),
2465          (297, '2012-08-11'),
2466          (298, '2012-08-11'),
2467          (299, '2012-08-11'),
2468          (300, '2012-08-11'),
2469          (301, '2012-08-11');
2470  COMMIT;
```

- Copier les 2470 lignes

- Dans notre base de données **JO.db**, cliquer sur « Exécuter le SQL »
- Coller les 2470 lignes, puis exécuter avec 

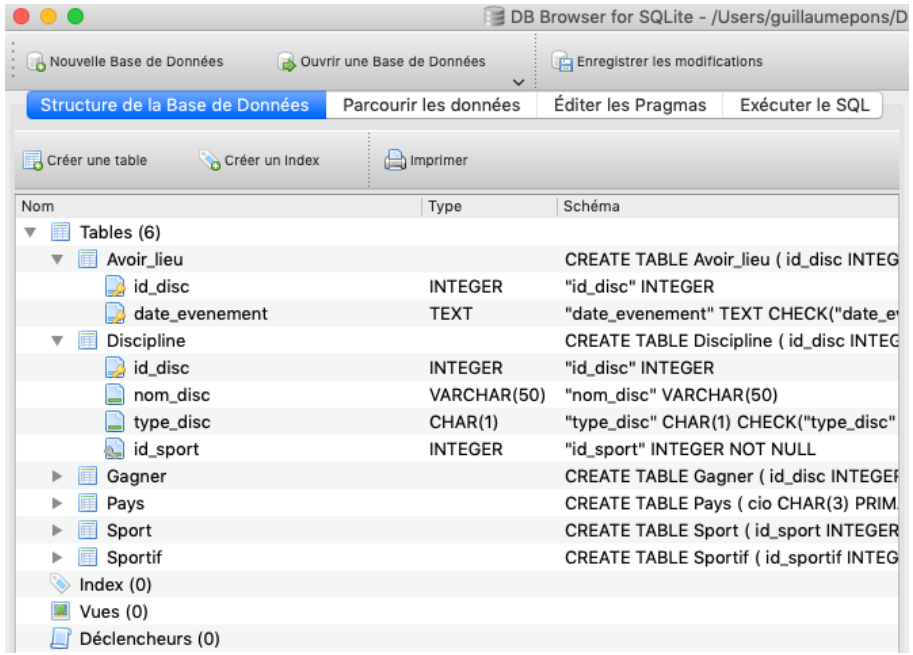


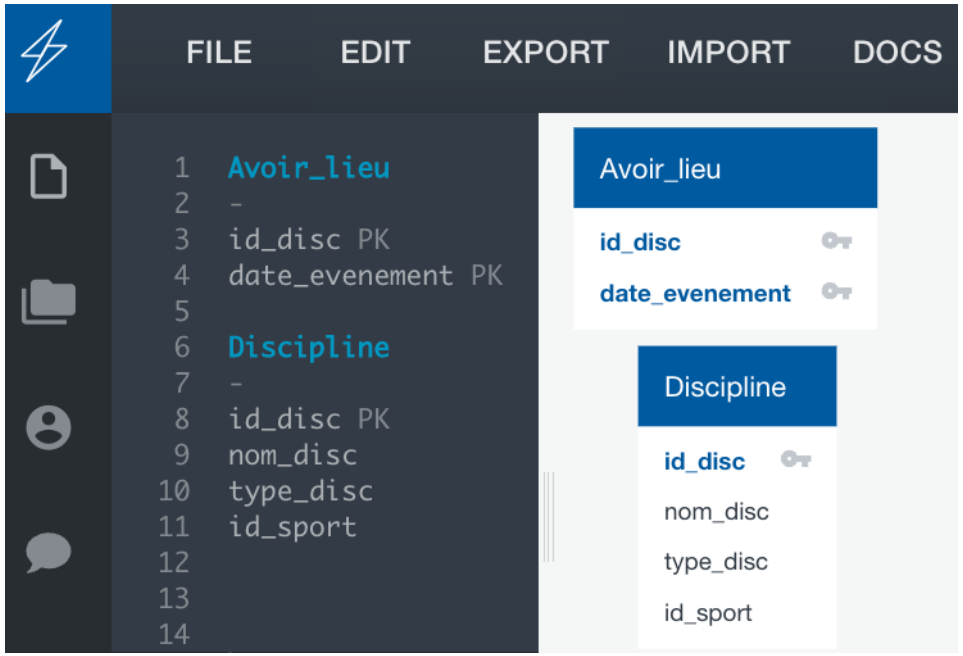
Notre base de données **JO.db** contient à présent **6 relations**



Partie 2 - Schéma et diagramme relationnels

- Aller sur le site https://www.quickdatabasediagrams.com/
- En observant la « Structure de la Base de Données » sur DB Browser for SQLite, écrire le schéma relationnel de la base de données **uniquement en précisant les clés primaires** comme dans l'exemple ci-dessous. Les diagrammes des relations se dessinent automatiquement sur la droite.






Sport
id_sport
nom_sport
Avoir_lieu
id_disc
date_evenement
Gagner
id_disc
id_sportif
medaille
Discipline
id_disc
nom_disc
type_disc
id_sport
Pays
cio
iso2
iso3
nom
capitale
region
Sportif
id_sportif
prenom
nom
cio
sexe

- Compléter à présent le diagramme de droite **en traçant les liens entre les tables**, tout en parcourant les données sur DB Browser. Par exemple, en reliant la clé étrangère **id_disc** de la relation **Avoir_lieu** vers la clé primaire **id_disc** de la relation **Discipline**, le schéma se complète automatiquement à gauche.

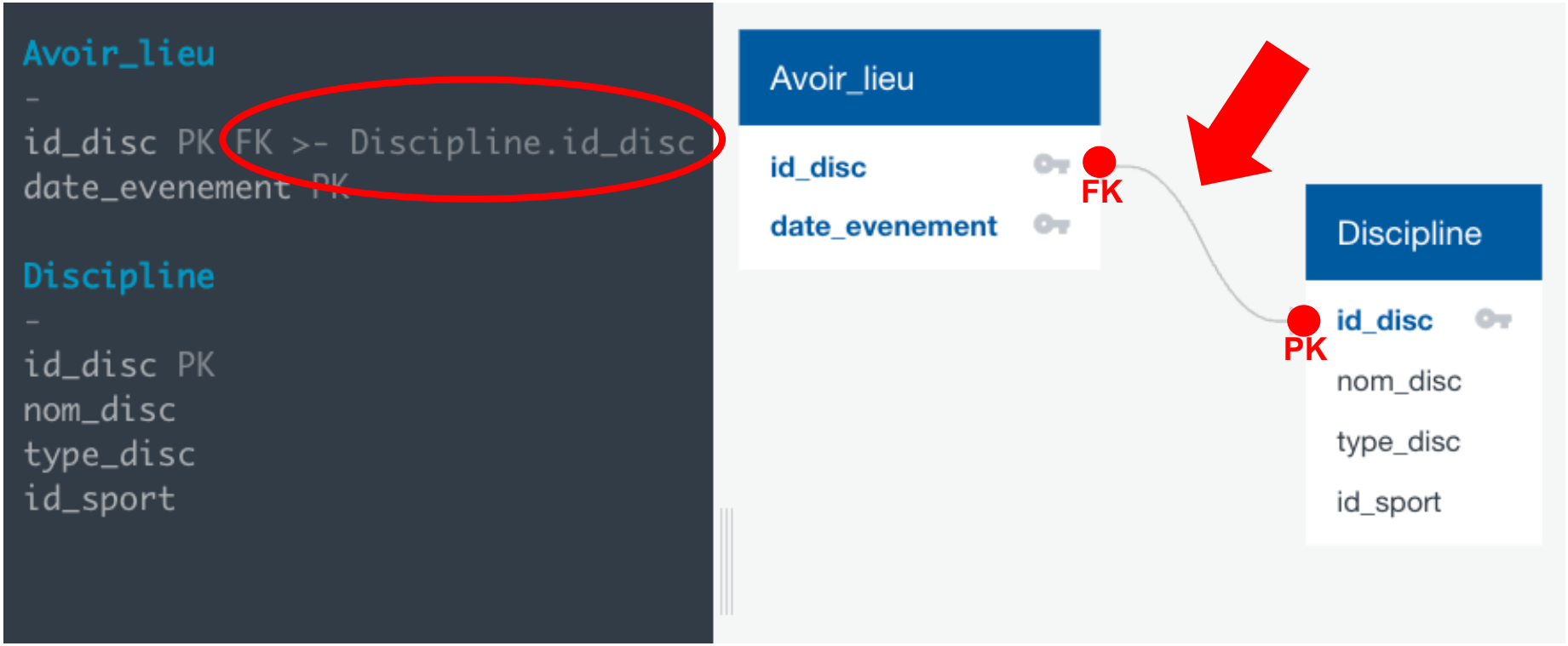


Diagramme relationnel de la base de données JO

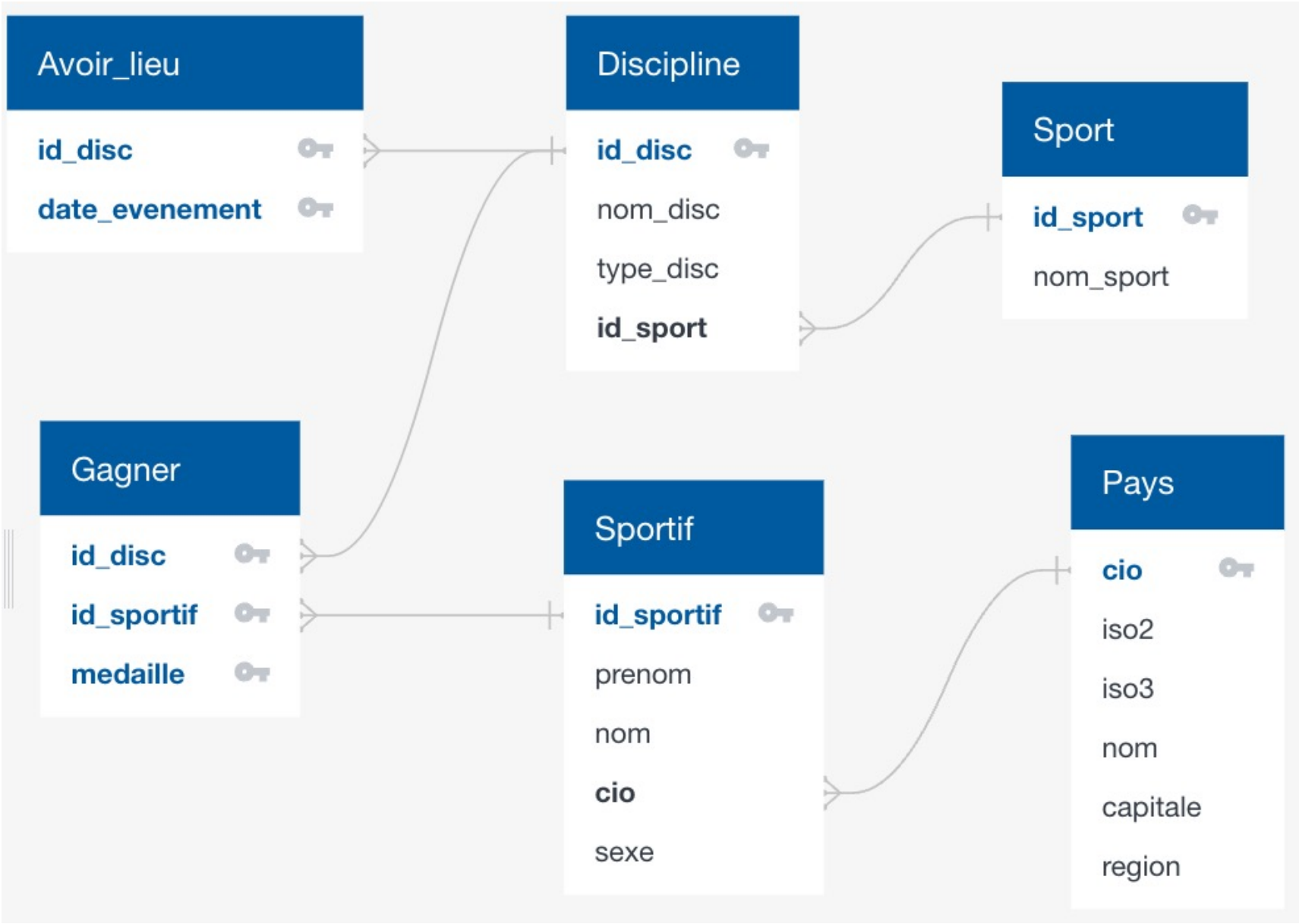


Diagramme relationnel de la base de données JO

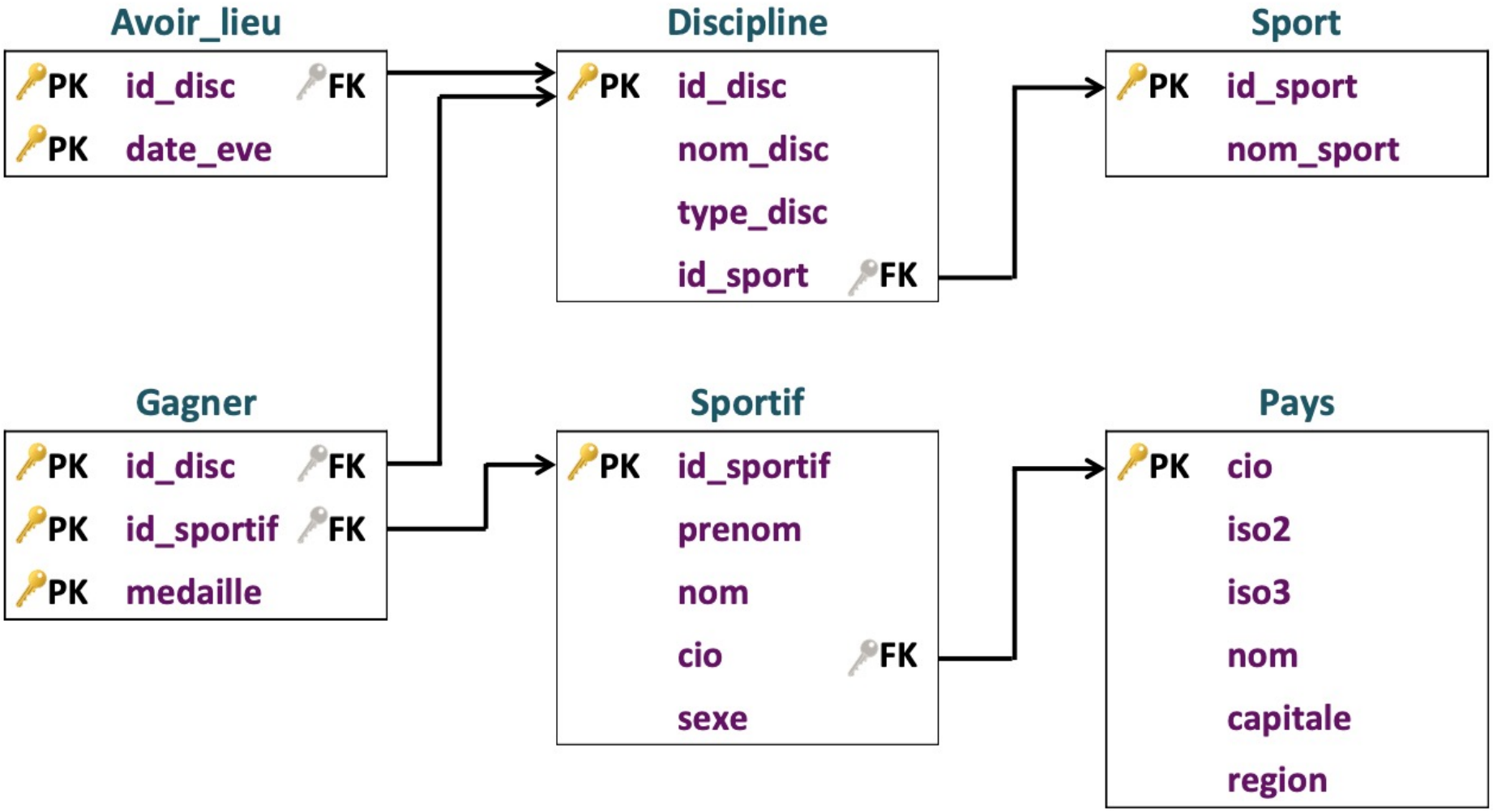
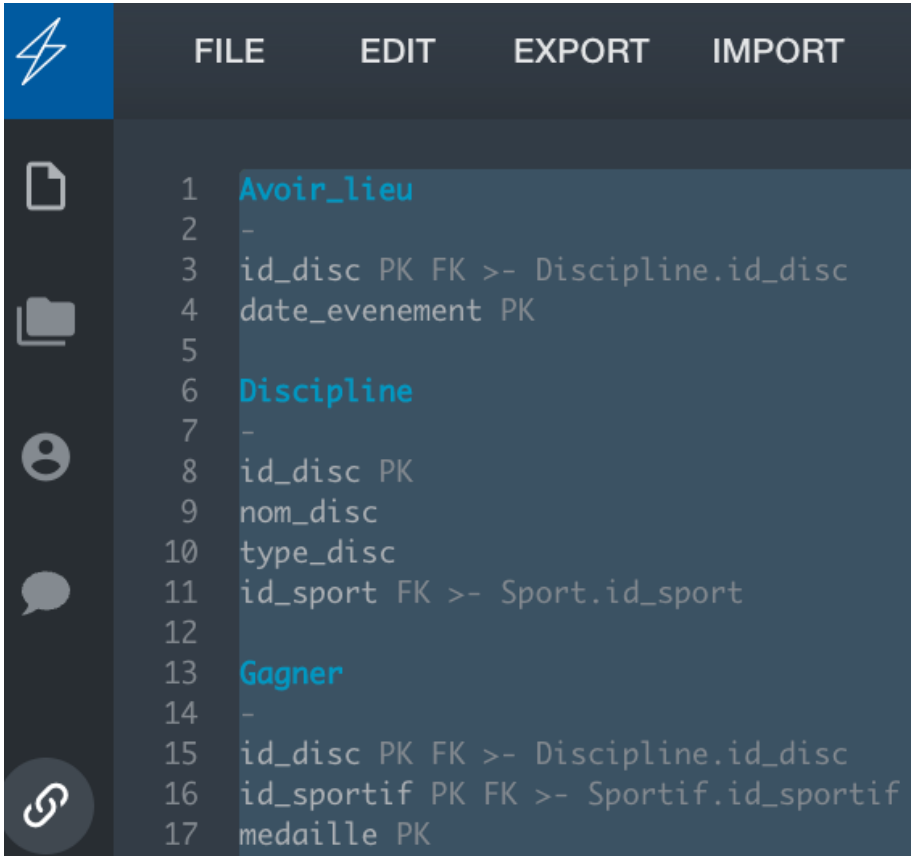
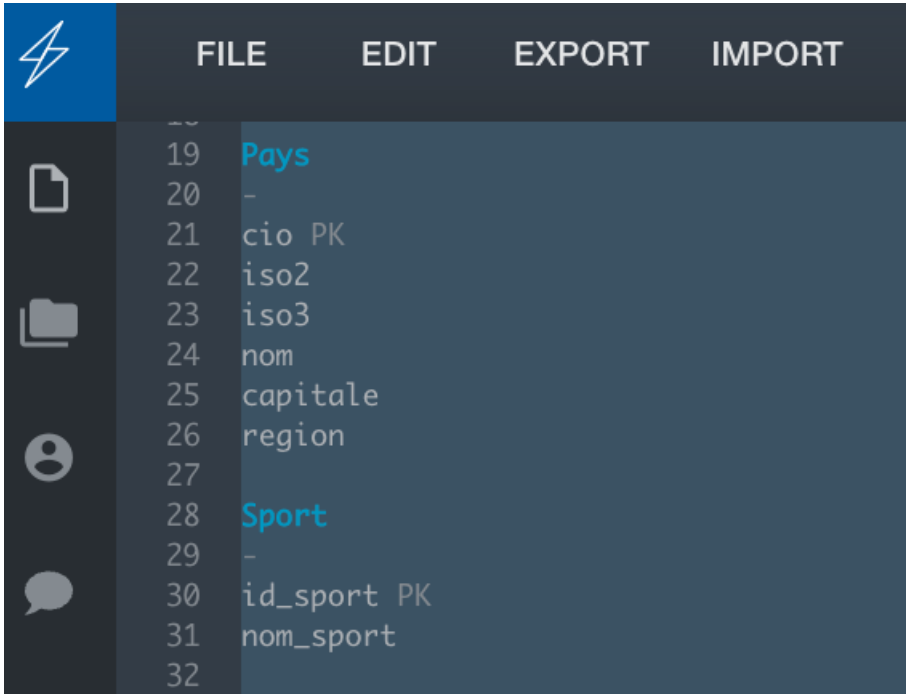


Schéma relationnel de la base de données JO





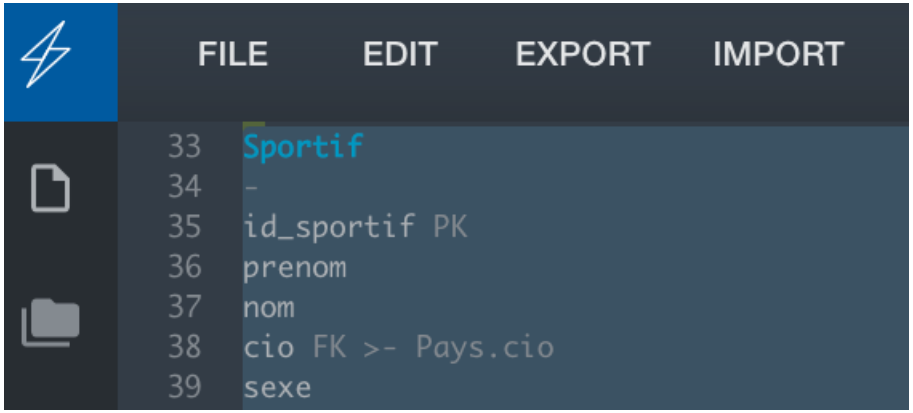


Schéma relationnel de la base de données JO

Avoir_lieu (**id_disc*** INTEGER , **date_evenement** TEXT)

Discipline (**id_disc** INTEGER , **nom_disc** VARCHAR(50) ,
type_disc CHAR(1) , **id_sport*** INTEGER)

Gagner (**id_disc*** INTEGER , **id_sportif*** INTEGER , **medaille** CHAR(1))

Pays (**cio** CHAR(3) , **iso2** CHAR(2) , **iso3** CHAR(3) , **nom** VARCHAR(50) ,
capitale VARCHAR(30) , **region** VARCHAR(30))

Sport (**id_id_sport** INTEGER , **nom_sport** VARCHAR(30))

Sportif (**id_sportif** INTEGER , **prenom** VARCHAR(30) , **nom** VARCHAR(30) ,
cio* CHAR(3) , **sexe** CHAR(1))

Partie 3 - Requêtes

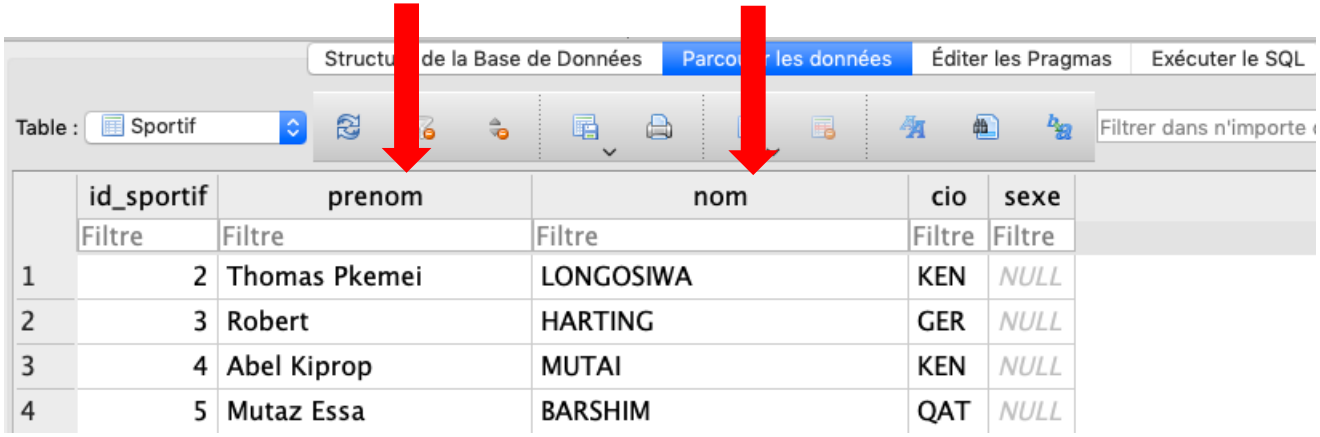
Ouvrir le fichier **JO-requetes.sql** avec Visual Studio Code et répondre aux questions en exécutant les bonnes requêtes SQL avec DB Browser for SQLite. Penser à enregistrer vos réponses sur le fichier.

```
1   -- Le double tiret permet de faire un commentaire jusqu'à la fin de la ligne.
2   /*
3   Le commentaire multi-ligne à l'avantage de pouvoir
4   indiquer où commence et où se termine le commentaire.
5   Il est donc possible de l'utiliser en plein milieu
6   d'une requête SQL sans problème.
7   */
8
9   -- TP JO 2012
10  -- Partie 4
11
12  -- Requêtes sans jointure
13  -- 1) Afficher le nom et prénom des sportifs. Combien y en a-t-il ?
14
15
16  -- 2) Afficher la liste des sportifs français (utiliser cio = 'France').
17
18
19  -- 3) Afficher la liste des 301 disciplines triées par l'identifiant du sport
20
21
```

Question 1	**Afficher le nom et prénom des sportifs.** Combien y en a-t-il ?

Question 1

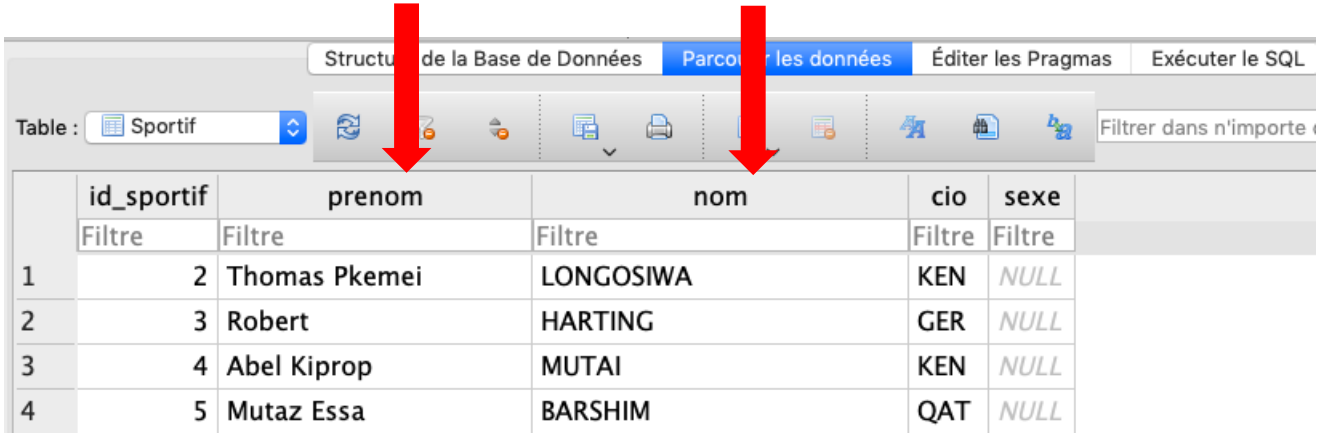
Afficher le nom et prénom des sportifs.
Combien y en a-t-il ?

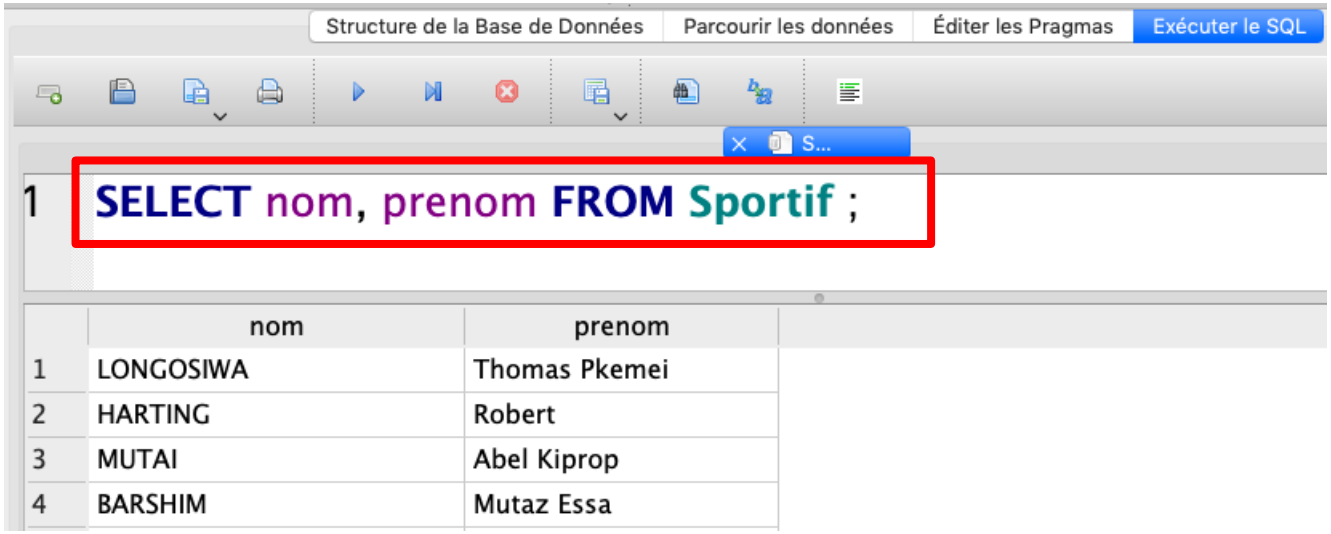


	id_sportif	prenom	nom	cio	sexe
	Filtre	Filtre	Filtre	Filtre	Filtre
1	2	Thomas Pkemei	LONGOSIWA	KEN	NULL
2	3	Robert	HARTING	GER	NULL
3	4	Abel Kiprop	MUTAI	KEN	NULL
4	5	Mutaz Essa	BARSHIM	QAT	NULL

Question 1

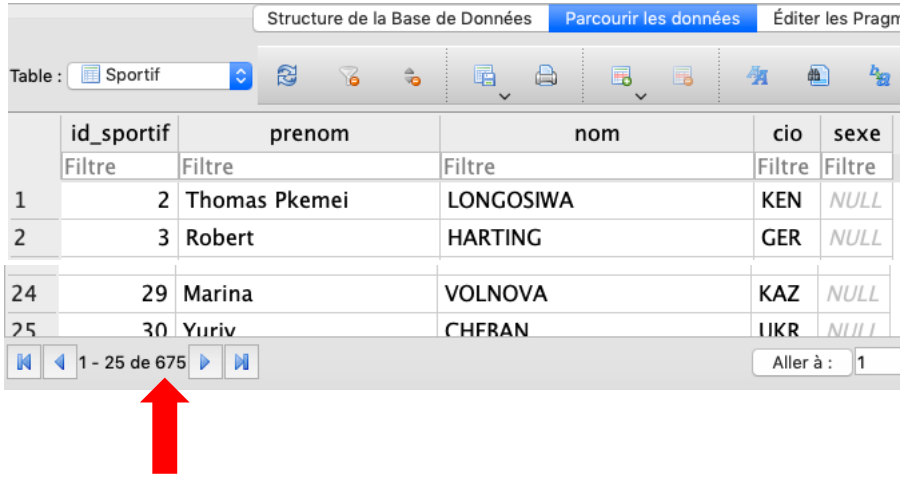
Afficher le nom et prénom des sportifs.
Combien y en a-t-il ?

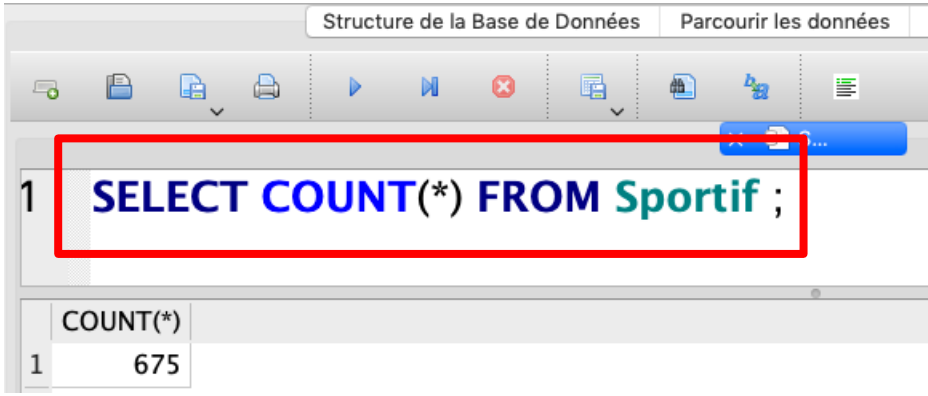


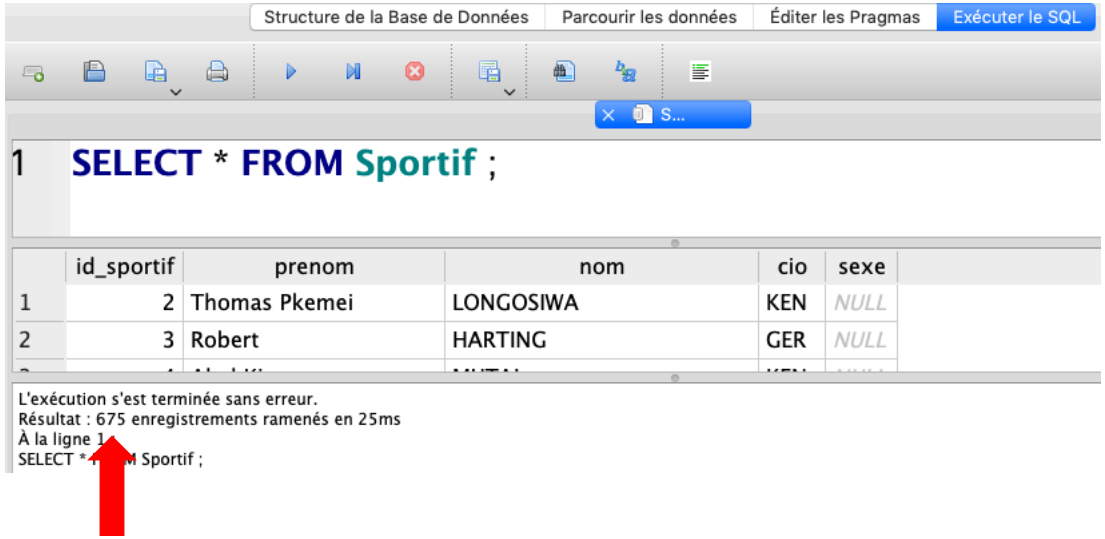


Question 1

Afficher le nom et prénom des sportifs.
Combien y en a-t-il ?





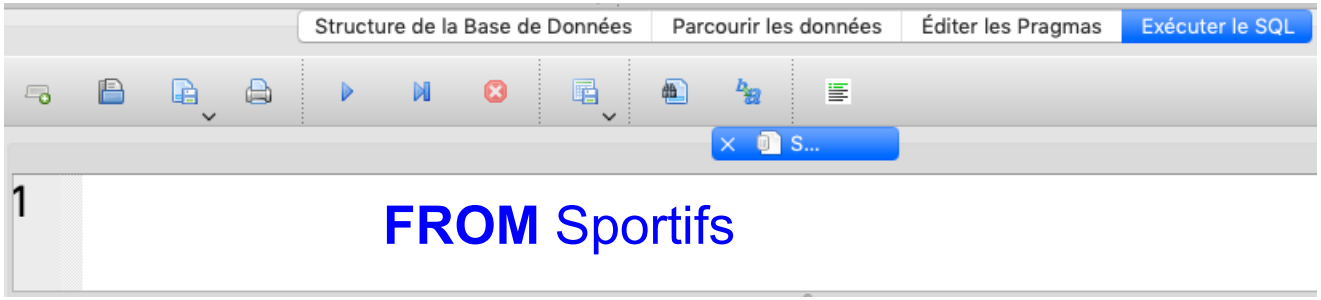


Question 2 Afficher la liste des sportifs français (utiliser cio = 'FRA').

Question 2 Afficher la liste des sportifs français (utiliser cio = 'FRA').

① **FROM** Sportifs

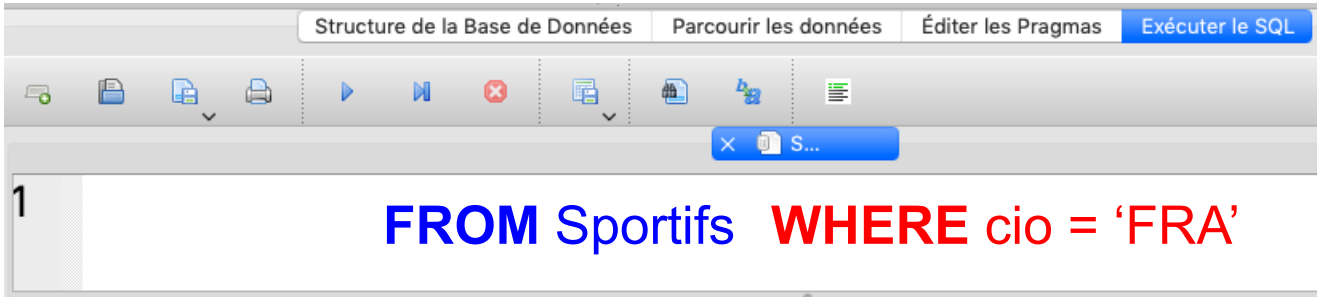
id_sportif	prenom	nom	cio	sexe
Filtre	Filtre	Filtre	Filtre	Filtre
506	Yibing	CHEN	CHN	*NULL*
507	Ugo	LEGRAND	FRA	*NULL*
508	Alexander	MIKHAYLIN	RUS	*NULL*
509	Mika	SUGIMOTO	JPN	*NULL*
510	Wen	TONG	CHN	*NULL*
511	Corina	CAPRIORIU	ROU	*NULL*
512	Automne	PAVIA	FRA	*NULL*
513	Adam	MAROSI	HUN	*NULL*
514	MahÈ	DRYSDALE	NZL	*NULL*
515	Alan	CAMPBELL	GBR	*NULL*
516	Rasmus	MYRGREN	SWE	*NULL*
518	Jonas	HOGH-CHRISTENSEN	DEN	*NULL*
519	Jonathan	LOBERT	FRA	*NULL*



Question 2 Afficher la liste des sportifs français (utiliser cio = 'FRA').

id_sportif	prenom	nom	cio	sexe
Filtre	Filtre	Filtre	Filtre	Filtre
506	Yibing	CHEN	CHN	NULL
507	Ugo	LEGRAND	FRA	NULL
508	Alexander	MIKHAYLIN	RUS	NULL
509	Mika	SUGIMOTO	JPN	NULL
510	Wen	TONG	CHN	NULL
511	Corina	CAPRIORIU	ROU	NULL
512	Automne	PAVIA	FRA	NULL
513	Adam	MAROSI	HUN	NULL
514	MahÈ	DRYSDALE	NZL	NULL
515	Alan	CAMPBELL	GBR	NULL
516	Rasmus	MYRGREN	SWE	NULL
518	Jonas	HOGH-CHRISTENSEN	DEN	NULL
519	Jonathan	LOBERT	FRA	NULL

② **WHERE** cio = 'FRA'

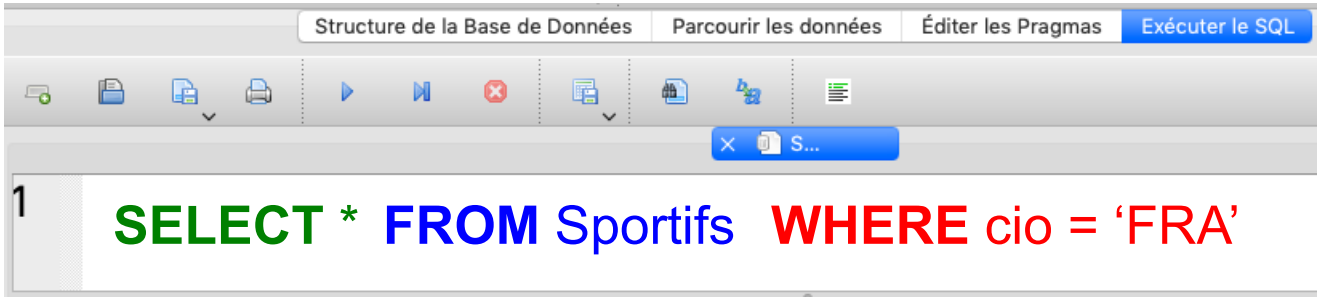


Question 2

Afficher la liste des sportifs français (utiliser cio = 'FRA').

id_sportif	prenom	nom	cio	sexe
Filtre	Filtre	Filtre	Filtre	Filtre
506	Yibing	CHEN	CHN	NULL
507	Ugo	LEGRAND	FRA	NULL
508	Alexander	MIKHAYLIN	RUS	NULL
509	Mika	SUGIMOTO	JPN	NULL
510	Wen	TONG	CHN	NULL
511	Corina	CAPRIORIU	ROU	NULL
512	Automne	PAVIA	FRA	NULL
513	Adam	MAROSI	HUN	NULL
514	MahÈ	DRYSDALE	NZL	NULL
515	Alan	CAMPBELL	GBR	NULL
516	Rasmus	MYRGREN	SWE	NULL
518	Jonas	HOGH-CHRISTENSEN	DEN	NULL
519	Jonathan	LOBERT	FRA	NULL

③ **SELECT ***

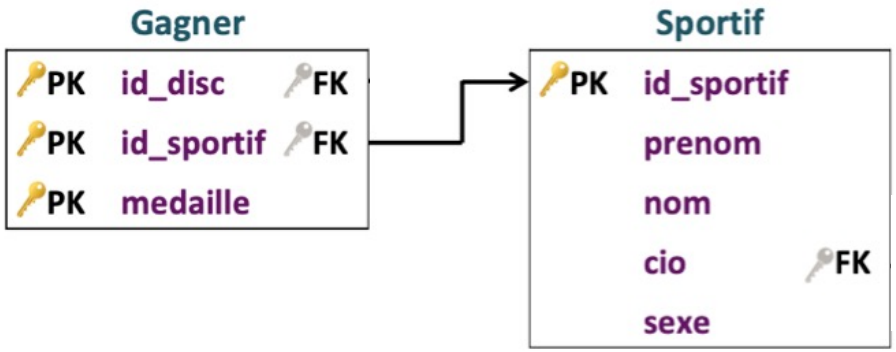


	id_sportif	prenom	nom	cio	s
1	47	Priscilla	GNETO	FRA	
2	78	Steeve	GUENOT	FRA	
3	110	Julie	BRESSET	FRA	
4	120	Hamilton	SABOT	FRA	
5	131	Delphine	RACINET	FRA	

L'exécution s'est terminée sans erreur.
Résultat : 24 enregistrements ramenés en 28ms
À la ligne 1 :
SELECT * FROM Sportif WHERE cio = 'FRA' ;

Question 3

Afficher la liste des médaillés français.



Question 3

Afficher la liste des médaillés français.

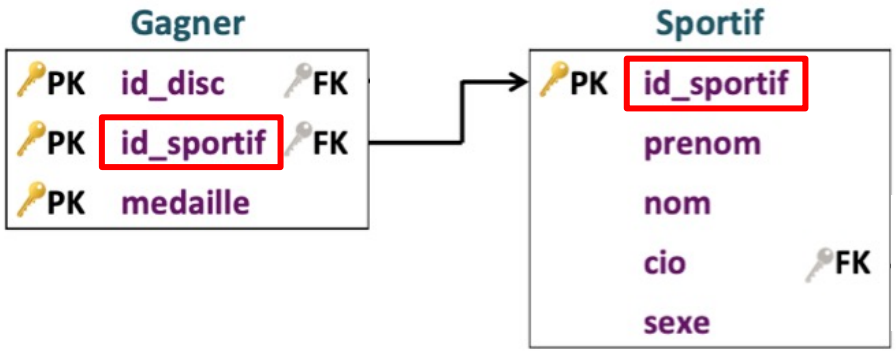


Table : Gagner

	id_disc	id_sportif	medaille
	Filtre	Filtre	Filtre
1	1	547	G
2	1	265	S
3	1	266	B
4	4	269	G
5	4	79	S
6	4	161	B

: Sportif

id_sportif	prenom	nom	cio	sexe
Filtre	Filtre	Filtre	Filtre	Filtre
264	Denis	TSARGUSH	RUS	NULL
265	Takaharu	FURUKAWA	JPN	NULL
266	Xiaoxiang	DAI	CHN	NULL
269	Bo Bae	KI	KOR	NULL
270	Timothy	KITUM	KEN	NULL
271	Keshorn	WALCOTT	TRI	NULL

Question 3

Afficher la liste des médaillés français.

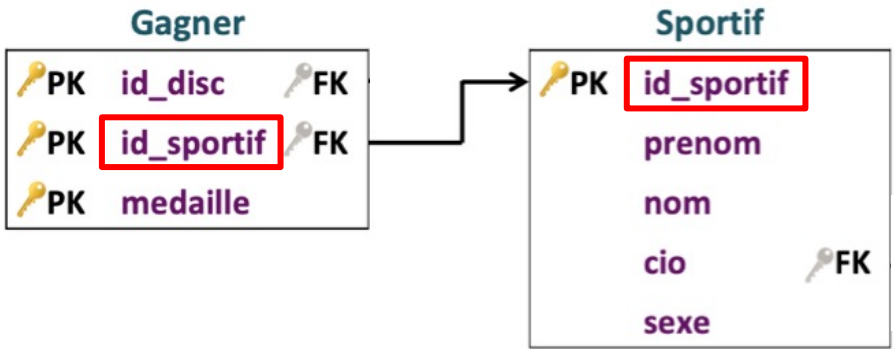


Table : Gagner

	id_disc	id_sportif	medaille
	Filtre	Filtre	Filtre
1	1	547	G
2	1	265	S
3	1	266	B
4	4	269	G
5	4	79	S
6	4	161	B

Sportif

id_sportif	prenom	nom	cio	sexe
Filtre	Filtre	Filtre	Filtre	Filtre
264	Denis	TSARGUSH	RUS	NULL
265	Takaharu	FURUKAWA	JPN	NULL
266	Xiaoxiang	DAI	CHN	NULL
269	Bo Bae	KI	KOR	NULL
270	Timothy	KITUM	KEN	NULL
271	Keshorn	WALCOTT	TRI	NULL

Structure de la Base de Données | Parcourir les données | Éditer les Pragmas | Exécuter le SQL

S...

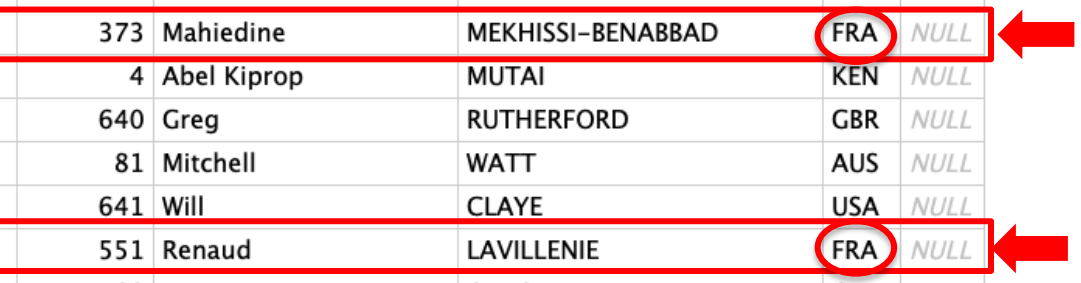
```
1  FROM Gagner JOIN Sportif ON Gagner.id_sportif = Sportif.id_sportif
```

Question 3

Afficher la liste des médaillés français.

```
1  SELECT *
2  FROM Gagner JOIN Sportif On Gagner.id_sportif = Sportif.id_sportif ;
```

	id_disc	id_sportif	medaille	id_sportif	prenom	nom	cio	sexe
25	11	272	G	272	Ezekiel	KEMBOI	KEN	NULL
26	11	373	S	373	Mahiedine	MEKHISSI-BENABBAD	FRA	NULL
27	11	4	B	4	Abel Kiprop	MUTAI	KEN	NULL
28	12	640	G	640	Greg	RUTHERFORD	GBR	NULL
29	12	81	S	81	Mitchell	WATT	AUS	NULL
30	12	641	B	641	Will	CLAYE	USA	NULL
31	13	551	G	551	Renaud	LAVILLENIE	FRA	NULL
32	13	468	S	468	Bjorn	OTTO	GER	NULL



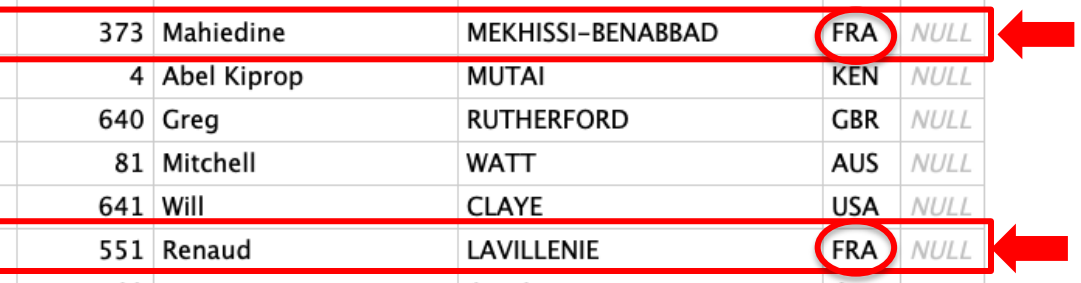
L'exécution s'est terminée sans erreur.
Résultat : 719 enregistrements ramenés en 7ms
À la ligne 1 :
SELECT *
FROM Gagner JOIN Sportif On Gagner.id_sportif = Sportif.id_sportif ;

Question 3

Afficher la liste des médaillés français.

```
1  SELECT *
2  FROM Gagner JOIN Sportif On Gagner.id_sportif = Sportif.id_sportif ;
```

	id_disc	id_sportif	medaille	id_sportif	prenom	nom	cio	sexe
25	11	272	G	272	Ezekiel	KEMBOI	KEN	NULL
26	11	373	S	373	Mahiedine	MEKHISSI-BENABBAD	FRA	NULL
27	11	4	B	4	Abel Kiprop	MUTAI	KEN	NULL
28	12	640	G	640	Greg	RUTHERFORD	GBR	NULL
29	12	81	S	81	Mitchell	WATT	AUS	NULL
30	12	641	B	641	Will	CLAYE	USA	NULL
31	13	551	G	551	Renaud	LAVILLENIE	FRA	NULL
32	13	468	S	468	Bjorn	OTTO	GER	NULL



L'exécution s'est terminée sans erreur.
Résultat : 719 enregistrements ramenés en 7ms
À la ligne 1 :
SELECT *
FROM Gagner JOIN Sportif On Gagner.id_sportif = Sportif.id_sportif ;

```
1  SELECT *
2  FROM Gagner JOIN Sportif On Gagner.id_sportif = Sportif.id_sportif
3  WHERE Sportif.cio = 'FRA'  ;
```

	id_disc	id_sportif	medaille	id_sportif	prenom	nom	cio	sexe
1	11	373	S	373	Mahiedine	MEKHISSI-BENABBAD	FRA	NULL
2	13	551	G	551	Renaud	LAVILLENIE	FRA	NULL
3	74	662	G	662	Tony	ESTANGUET	FRA	NULL

L'exécution s'est terminée sans erreur.
Résultat : 25 enregistrements ramenés en 24ms

Question 4

Afficher les noms des 82 pays situés après la France et avant la Portugal par ordre alphabétique. Utiliser les opérateurs inférieur et supérieur. Remarquer que l'opérateur BETWEEN ne produit pas le résultat attendu (84 pays).

cio	iso2	iso3	nom	capitale	region
Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
CAY	KY	CYM	Cayman Islands	George Town	Caribbean
IVB	VG	VGB	Virgin Islands UK	Road Town	Caribbean
ISV	VI	VIR	Virgin Islands US	Charlotte Amalie	Caribbean
AFG	AF	AFG	Afghanistan	Kabul	Asia
RSA	ZA	ZAF	South Africa	Pretoria	Africa
ALB	AL	ALB	Albania	Tirana	Europe
ALG	DZ	DZA	Algeria	Alger	Africa
GER	DE	DEU	Germany	Berlin	Europe
AND	AD	AND	Andorra	Andorr-la-Vieille	Europe

Question 4	Afficher les noms des 82 pays situés après la France et avant la Portugal par ordre alphabétique. Utiliser les opérateurs inférieur et supérieur. Remarquer que l'opérateur BETWEEN ne produit pas le résultat attendu (84 pays).

```
1  SELECT nom FROM Pays ORDER BY nom ;
```

	nom
1	Afghanistan
2	Albania
3	Algeria
4	American Samoa
5	Andorra
6	Angola
7	Antigua and Barbuda

```
L'exécution s'est terminée sans erreur.
Résultat : 207 enregistrements ramenés en 22ms
À la ligne 1 :
SELECT nom FROM Pays ORDER BY nom ;
```

Question 4

Afficher les noms des 82 pays situés après la France et avant la Portugal par ordre alphabétique. Utiliser les opérateurs inférieur et supérieur. Remarquer que l'opérateur BETWEEN ne produit pas le résultat attendu (84 pays).

```
1  SELECT nom FROM Pays
2  WHERE nom > 'France' AND nom < 'Portugal'
3  ORDER BY nom ;
```

	nom
1	Gabon
2	Gambia
3	Georgia
4	Germany
5	Ghana
6	Greece
7	Grenad

```
L'exécution s'est terminée sans erreur.
Résultat : 82 enregistrements ramenés en 25ms
À la ligne 1 :
SELECT nom FROM Pays
WHERE nom > 'France' AND nom < 'Portugal'
ORDER BY nom ;
```

Question 4

Afficher les noms des 82 pays situés après la France et avant la Portugal par ordre alphabétique. Utiliser les opérateurs inférieur et supérieur. Remarquer que l'opérateur BETWEEN ne produit pas le résultat attendu (84 pays).

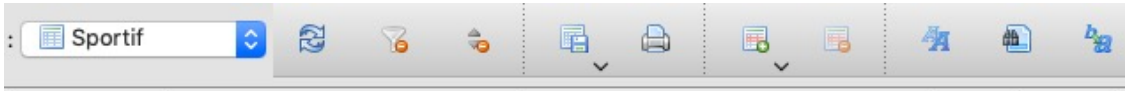
```
1  SELECT nom FROM Pays
2  WHERE nom BETWEEN 'France' AND 'Portugal' ORDER BY nom ;
```

	nom
1	France
2	Gabon
3	Gambia

```
L'exécution s'est terminée sans erreur.
Résultat : 84 enregistrements ramenés en 19ms
À la ligne 1 :
SELECT nom FROM Pays
WHERE nom BETWEEN 'France' AND 'Portugal' ORDER BY nom ;
```

Question 5

Afficher les noms des 61 sportifs qui sont soit français (FRA) soit britanniques (GBR).



id_sportif	prenom	nom	cio	sexe
Filtre	Filtre	Filtre	Filtre	Filtre
264	Denis	TSARGUSH	RUS	NULL
265	Takaharu	FURUKAWA	JPN	NULL
266	Xiaoxiang	DAI	CHN	NULL
269	Bo Bae	KI	KOR	NULL
270	Timothy	KITUM	KEN	NULL
271	Keshorn	WALCOTT	TRI	NULL

Question 5

Afficher les noms des 61 sportifs qui sont soit français (FRA) soit britanniques (GBR).

Sportif

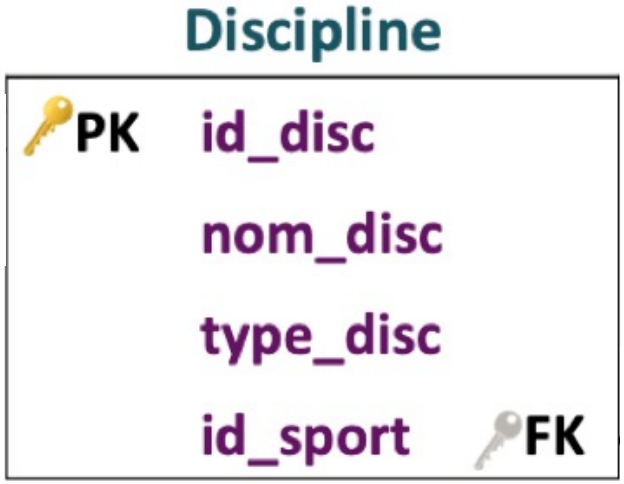
id_sportif	prenom	nom	cio	sexe
Filtre	Filtre	Filtre	Filtre	Filtre
264	Denis	TSARGUSH	RUS	NULL
265	Takaharu	FURUKAWA	JPN	NULL

```
1  SELECT nom
2  FROM Sportif
3  WHERE cio = 'FRA' OR cio = 'GBR' ;
4
```

	nom
1	OHURUOGU
2	ENNIS
3	TROTT

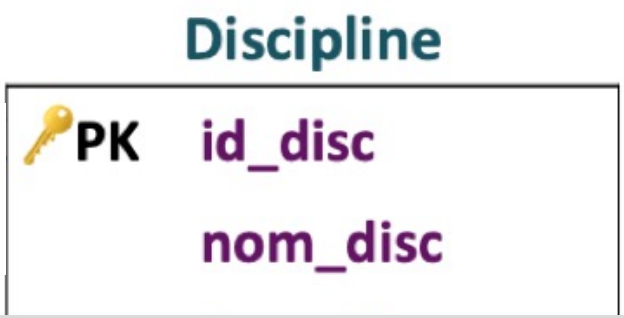
L'exécution s'est terminée sans erreur.
Résultat : 61 enregistrements ramenés en 388ms
À la ligne 1 :
SELECT nom
FROM Sportif
WHERE cio = 'FRA' OR cio = 'GBR' ;

Question 6	Afficher les intitulés des 131 disciplines contenant la chaîne de caractères «WOMEN».



Question 6

Afficher les intitulés des 131 disciplines contenant la chaîne de caractères «WOMEN».

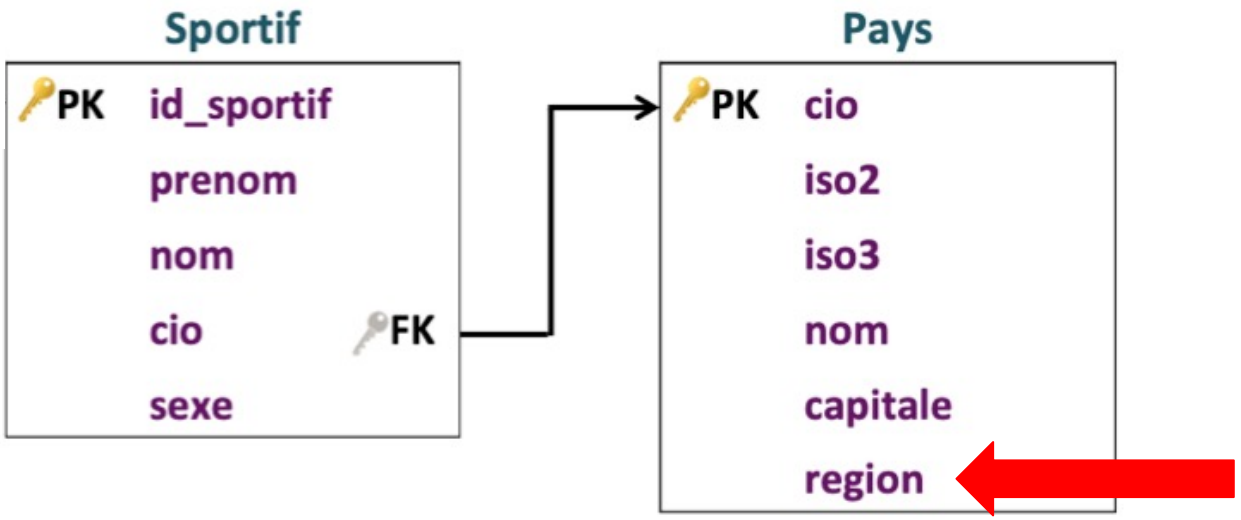



```
1  SELECT nom_disc
2  FROM Discipline
3  WHERE nom_disc LIKE '%WOMEN%' ;
4
```

	nom_disc
1	TEAM FITA OLYMPIC ROUND 70M ...
2	INDIVIDUAL FITA OLYMPIC ROUND ...
3	MARATHON WOMEN
4	4X400M RELAY WOMEN

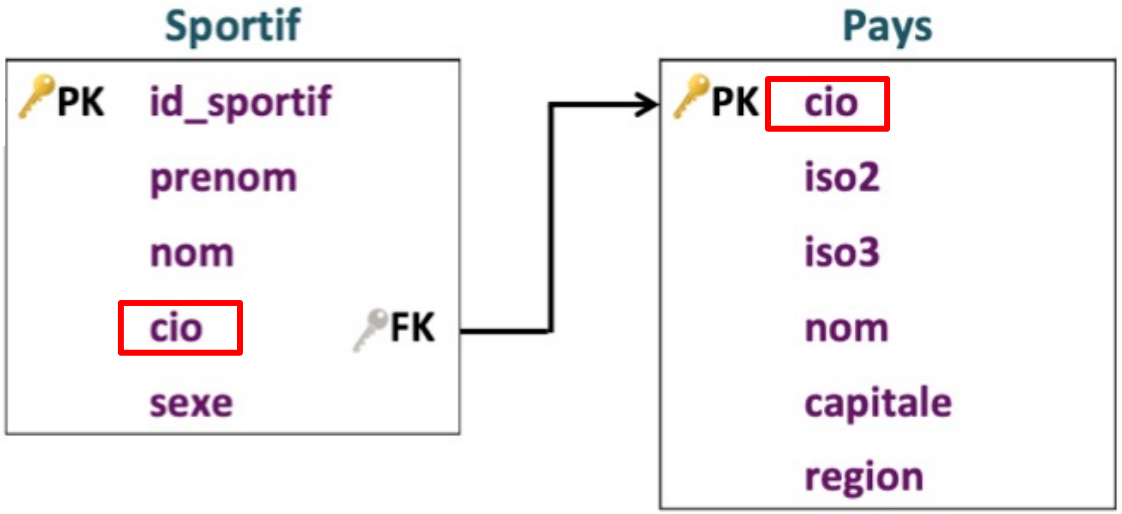
```
L'exécution s'est terminée sans erreur.
Résultat : 131 enregistrements ramenés en 19ms
À la ligne 1 :
SELECT nom_disc
FROM Discipline
WHERE nom_disc LIKE '%WOMEN%' ;
```

Question 7	Afficher la liste des noms et prénoms des sportifs européens.



Question 7

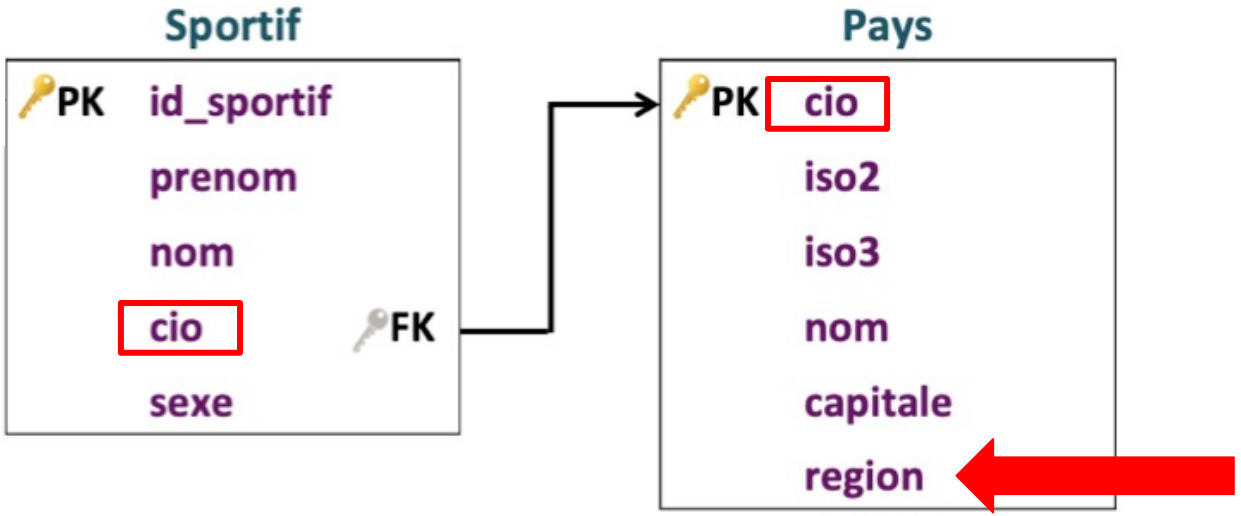
Afficher la liste des noms et prénoms des sportifs européens.




```
1  SELECT ...
2  FROM Sportif
3  JOIN Pays ON Sportif.cio = Pays.cio
4  WHERE ...
```

Question 7

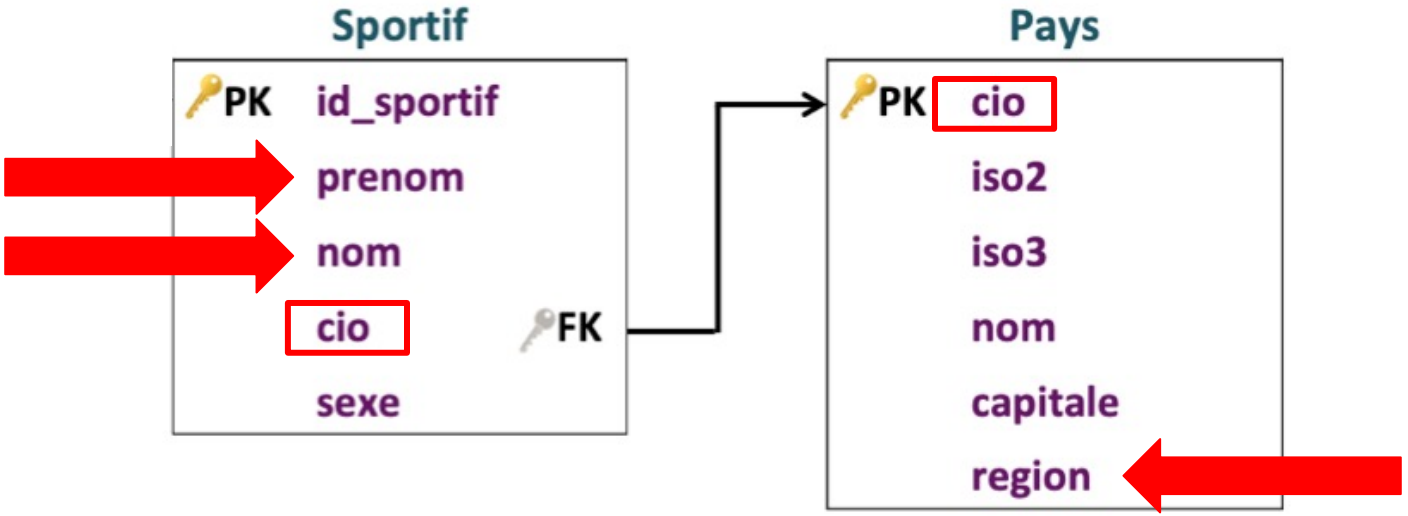
Afficher la liste des noms et prénoms des sportifs européens.




```
1   SELECT ...
2    FROM Sportif
3    JOIN Pays ON Sportif.cio = Pays.cio
4    WHERE Pays.region = 'Europe'  ;
```

Question 7

Afficher la liste des noms et prénoms des sportifs européens.

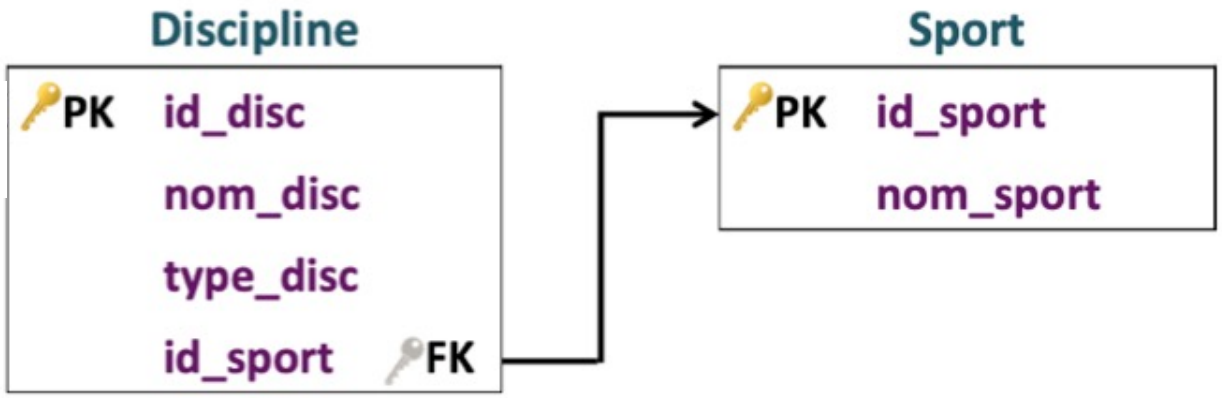



```
1  SELECT Sportif.nom, Sportif.prenom
2  FROM Sportif
3  JOIN Pays ON Sportif.cio = Pays.cio
4  WHERE Pays.region = 'Europe' ;
```

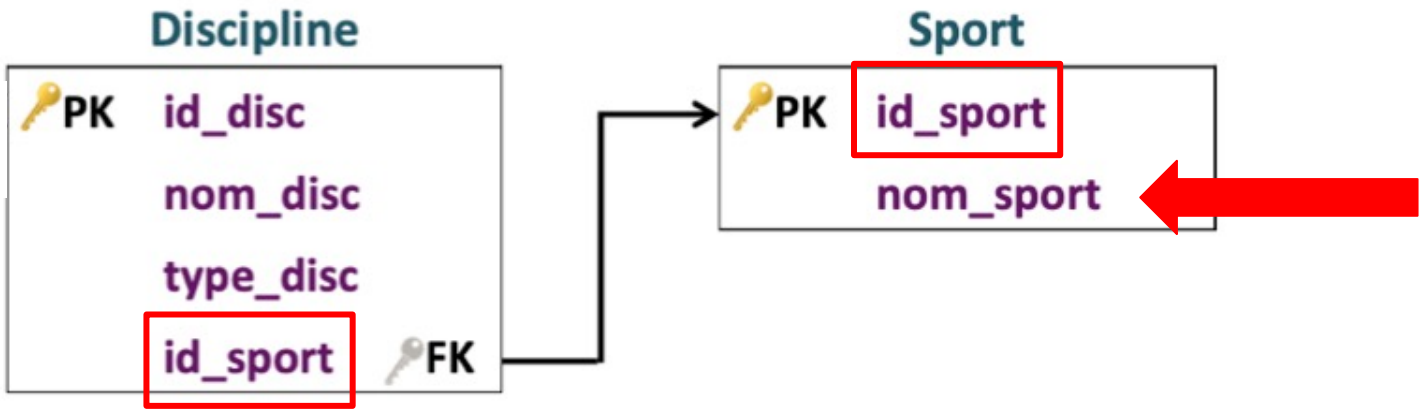
	nom	prenom
1	HARTING	Robert
2	MAJEWSKI	Tomasz

L'exécution s'est terminée sans erreur.
Résultat : 306 enregistrements ramenés en 22ms

Question 8 Afficher la liste des disciplines dépendant de l'athlétisme.



Question 8 Afficher la liste des disciplines dépendant de l'athlétisme.



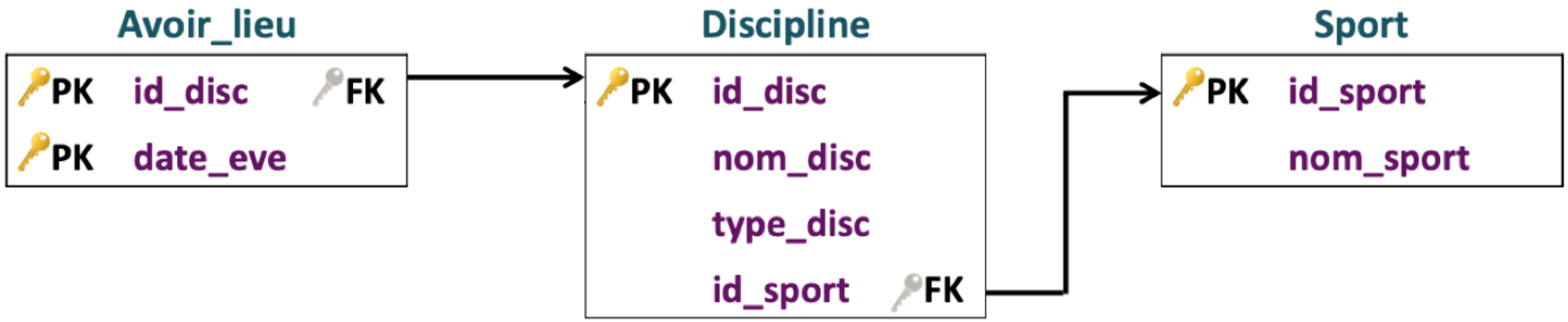

```
1  SELECT *
2  FROM Discipline
3  JOIN Sport ON Discipline.id_sport = Sport.id_sport
4  WHERE Sport.nom_sport = 'ATHLETICS' ;
```

	id_disc	nom_disc	type_disc	id_sport	id_sport	nom_sport
1	5	10000M MEN	I	2	2	ATHLETICS
2	6	800M MEN	I	2	2	ATHLETICS
3	7	5000M MEN	I	2	2	ATHLETICS

L'exécution s'est terminée sans erreur.
Résultat : 47 enregistrements ramenés en 23ms

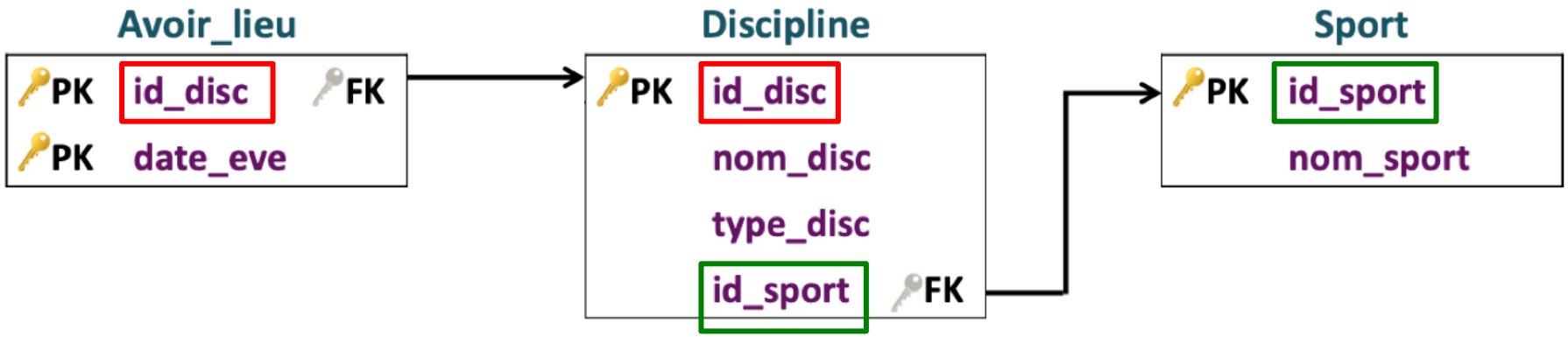
Question 9

Afficher tous les jours pendant lesquels un évènement lié à l'athlétisme a eu lieu.



Question 9

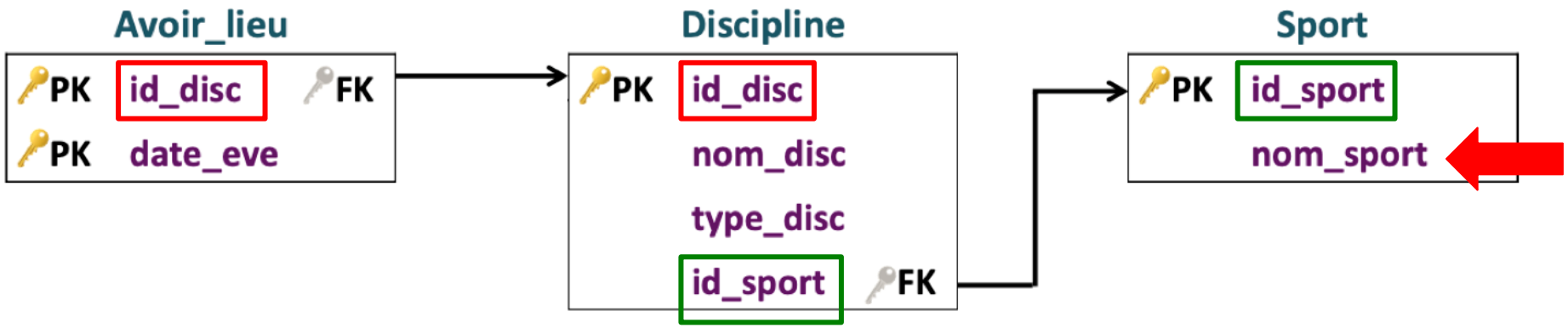
Afficher tous les jours pendant lesquels un évènement lié à l'athlétisme a eu lieu.




```
1  SELECT ...
2  FROM Avoir_lieu
3  JOIN Discipline ON Avoir_lieu.id_disc = Discipline.id_disc
4  JOIN Sport ON Discipline.id_sport = Sport.id_sport
5  WHERE ...
```

Question 9

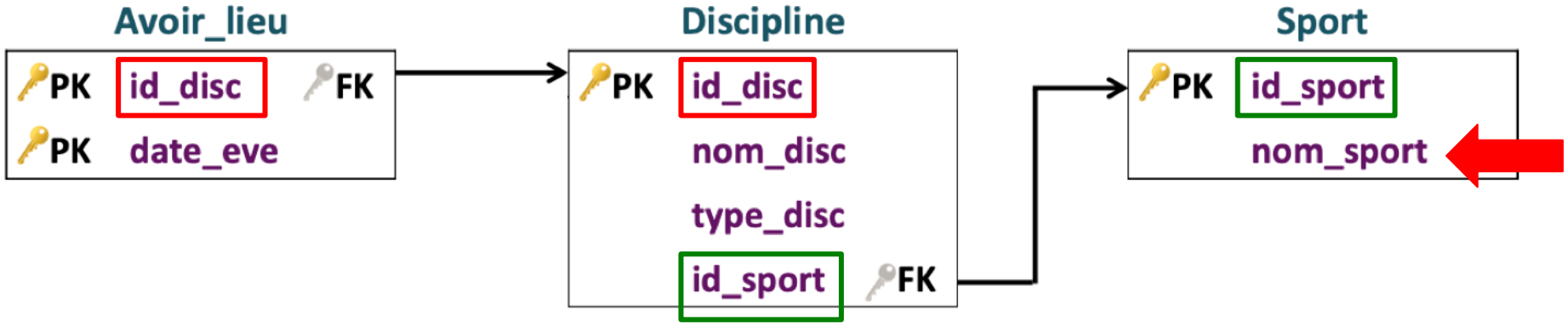
Afficher tous les jours pendant lesquels un évènement lié à l'athlétisme a eu lieu.




```sql
SELECT ...
FROM Avoir_lieu
JOIN Discipline ON Avoir_lieu.id_disc = Discipline.id_disc
JOIN Sport ON Discipline.id_sport = Sport.id_sport
WHERE Sport.nom_sport = 'ATHLETICS' ;
```

Question 9

Afficher tous les jours pendant lesquels un évènement lié à l'athlétisme a eu lieu.



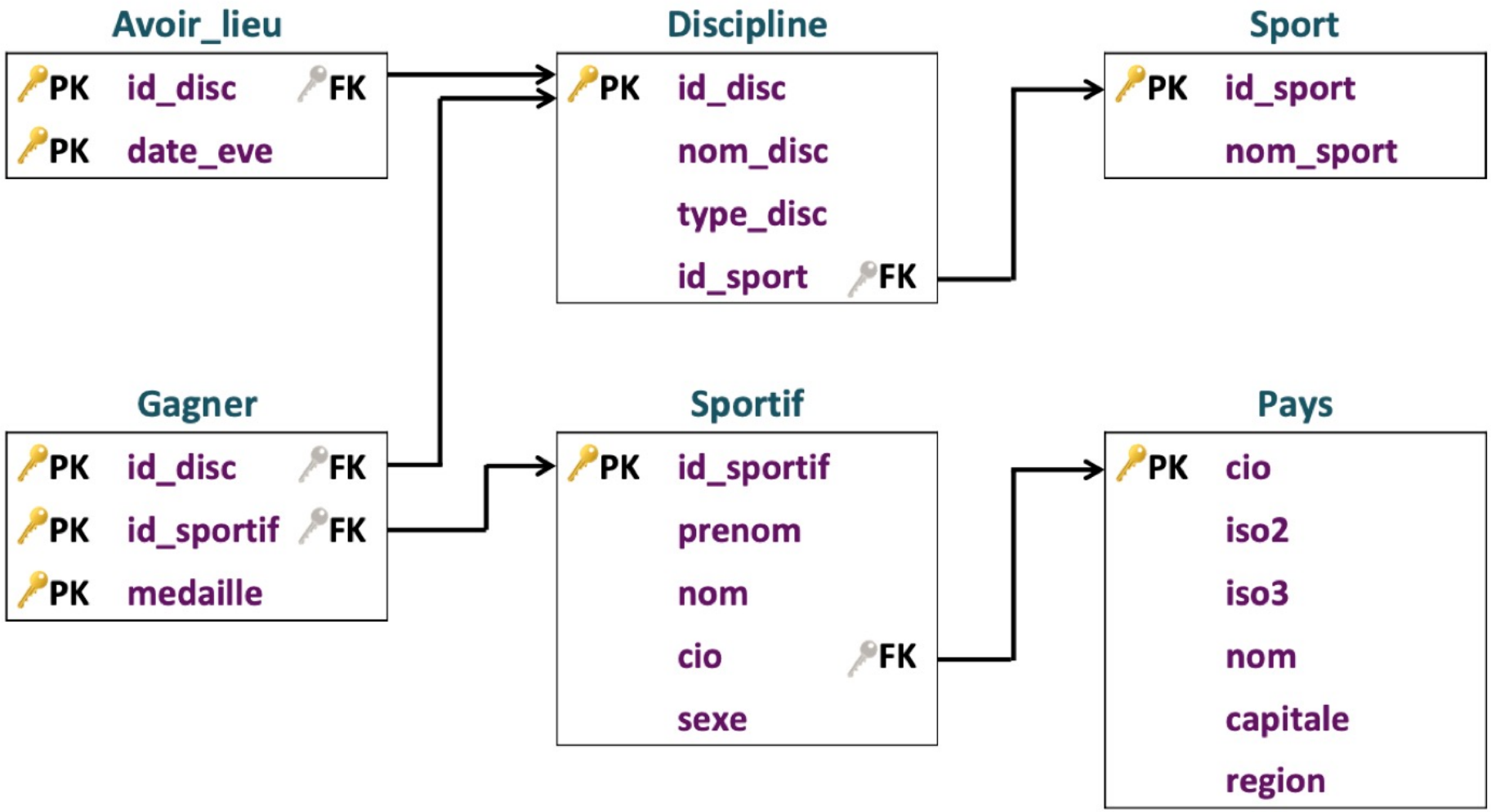

```
1  SELECT DISTINCT date_evenement
2  FROM Avoir_lieu
3  JOIN Discipline ON Avoir_lieu.id_disc = Discipline.id_disc
4  JOIN Sport ON Discipline.id_sport = Sport.id_sport
5  WHERE Sport.nom_sport = 'ATHLETICS' ;
```

	date_evenement
1	2012-08-01
2	2012-08-02
3	2012-08-03

L'exécution s'est terminée sans erreur.
Résultat : 5 enregistrements ramenés en 20ms

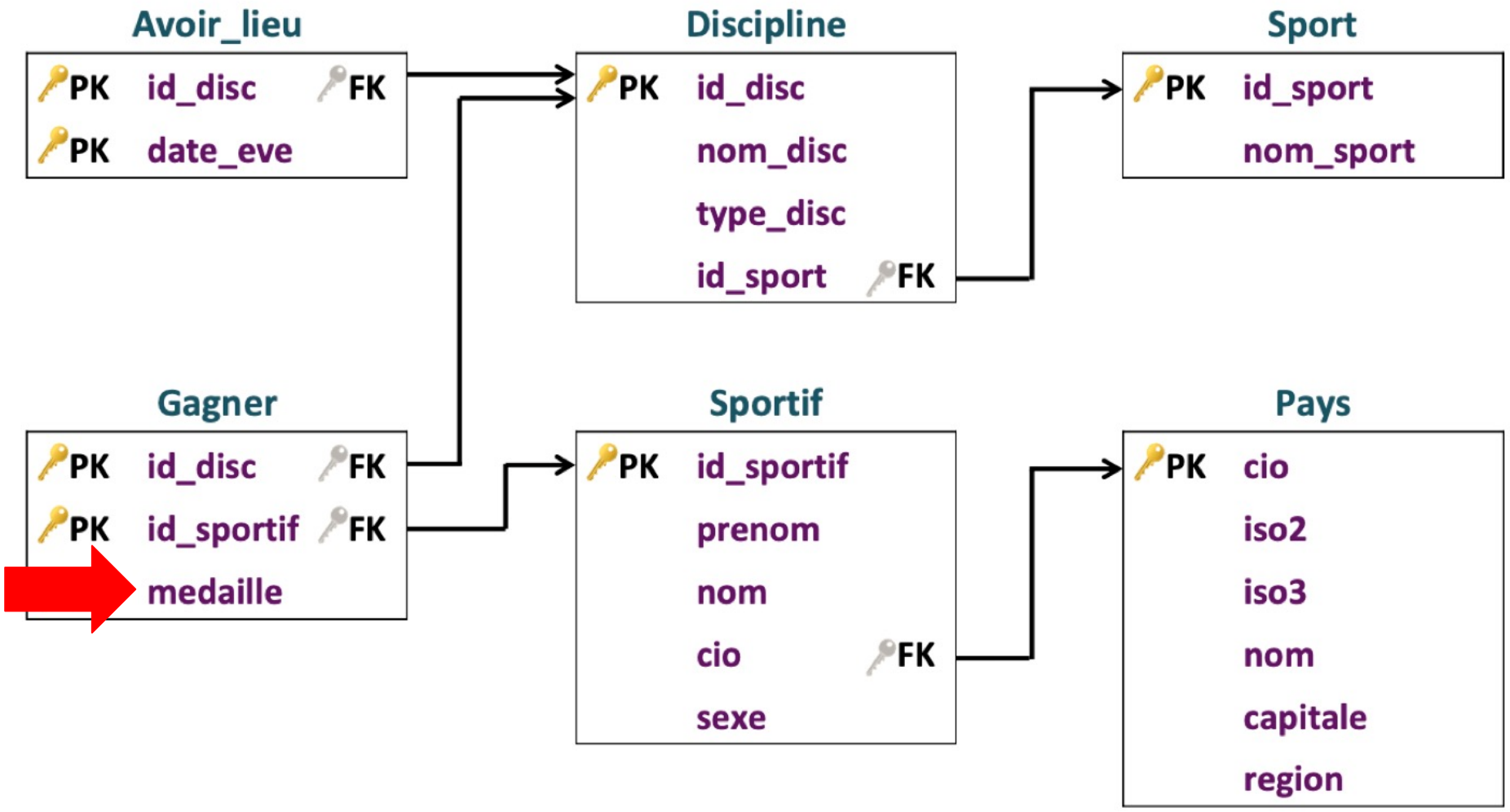
Question 10

Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.



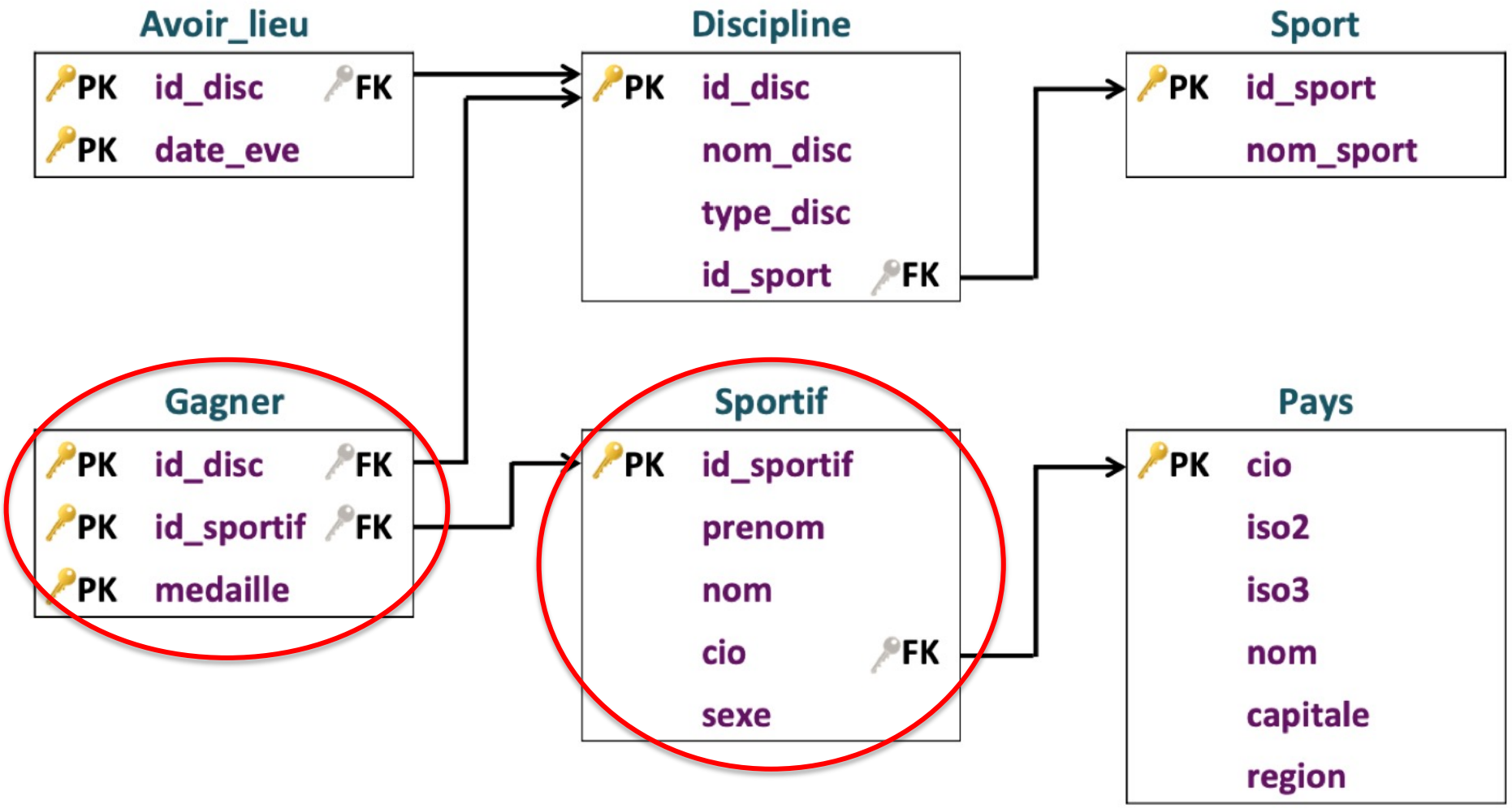
Question 10

Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.



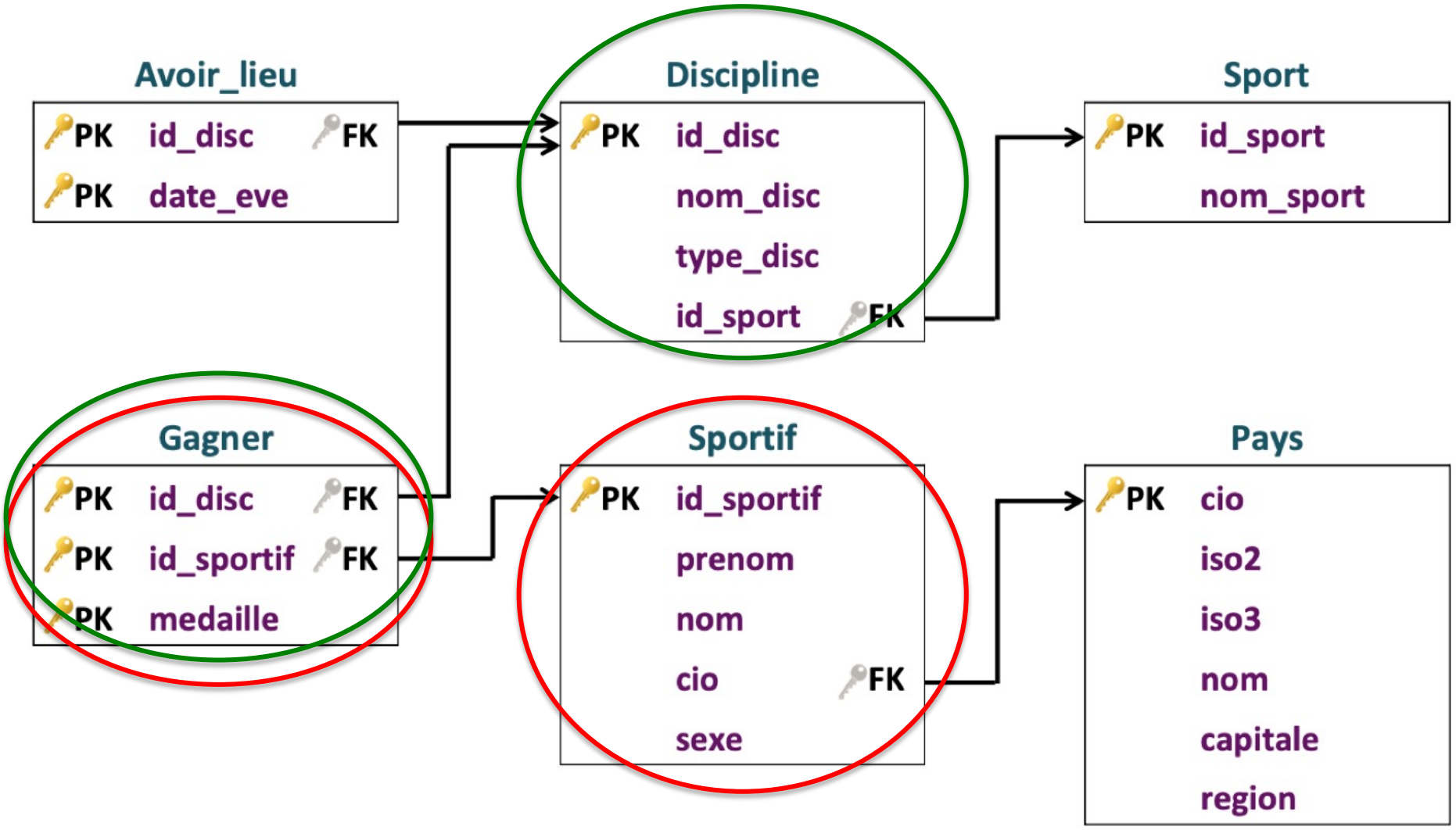
Question 10

Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.



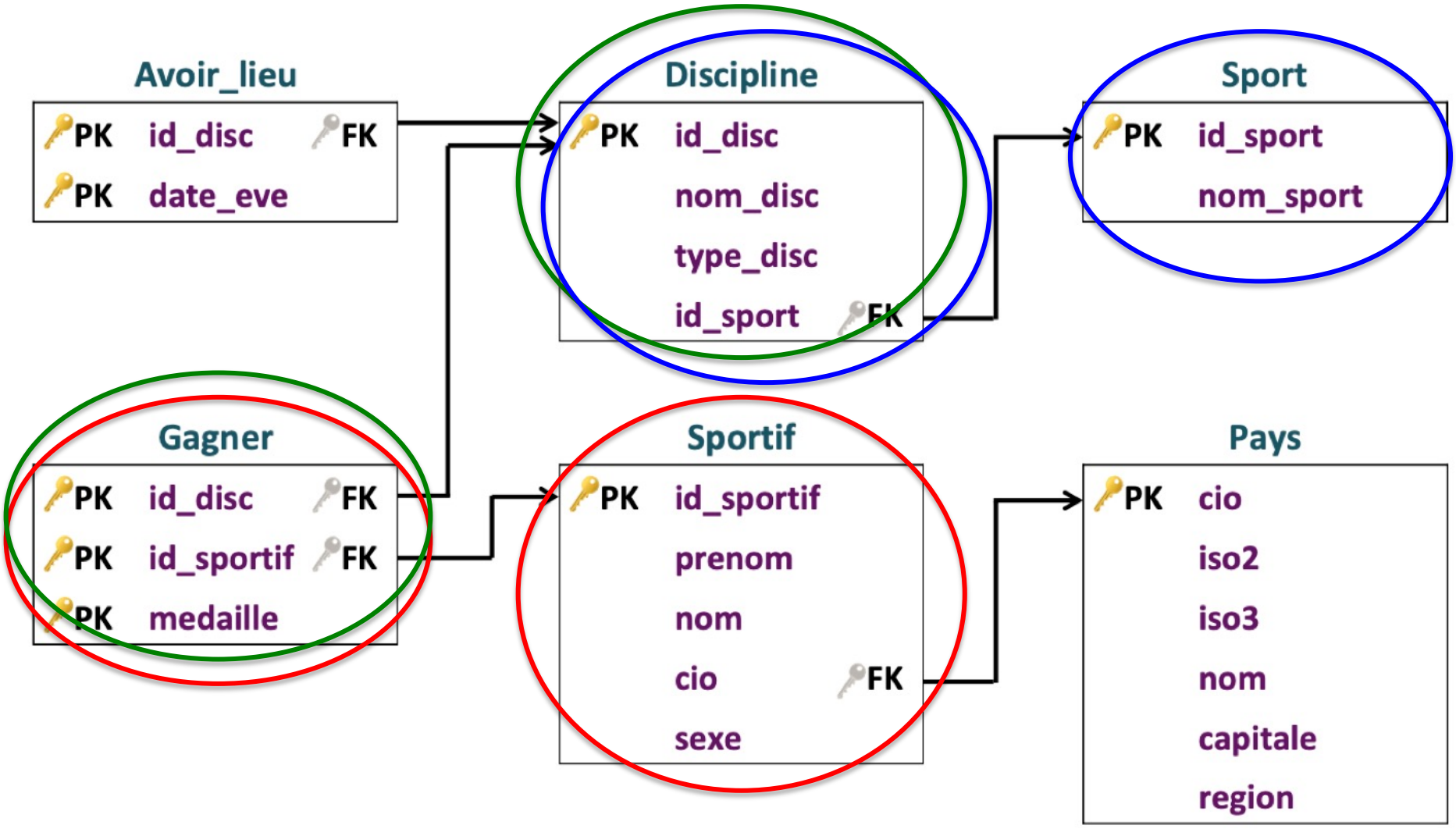
Question 10

Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.



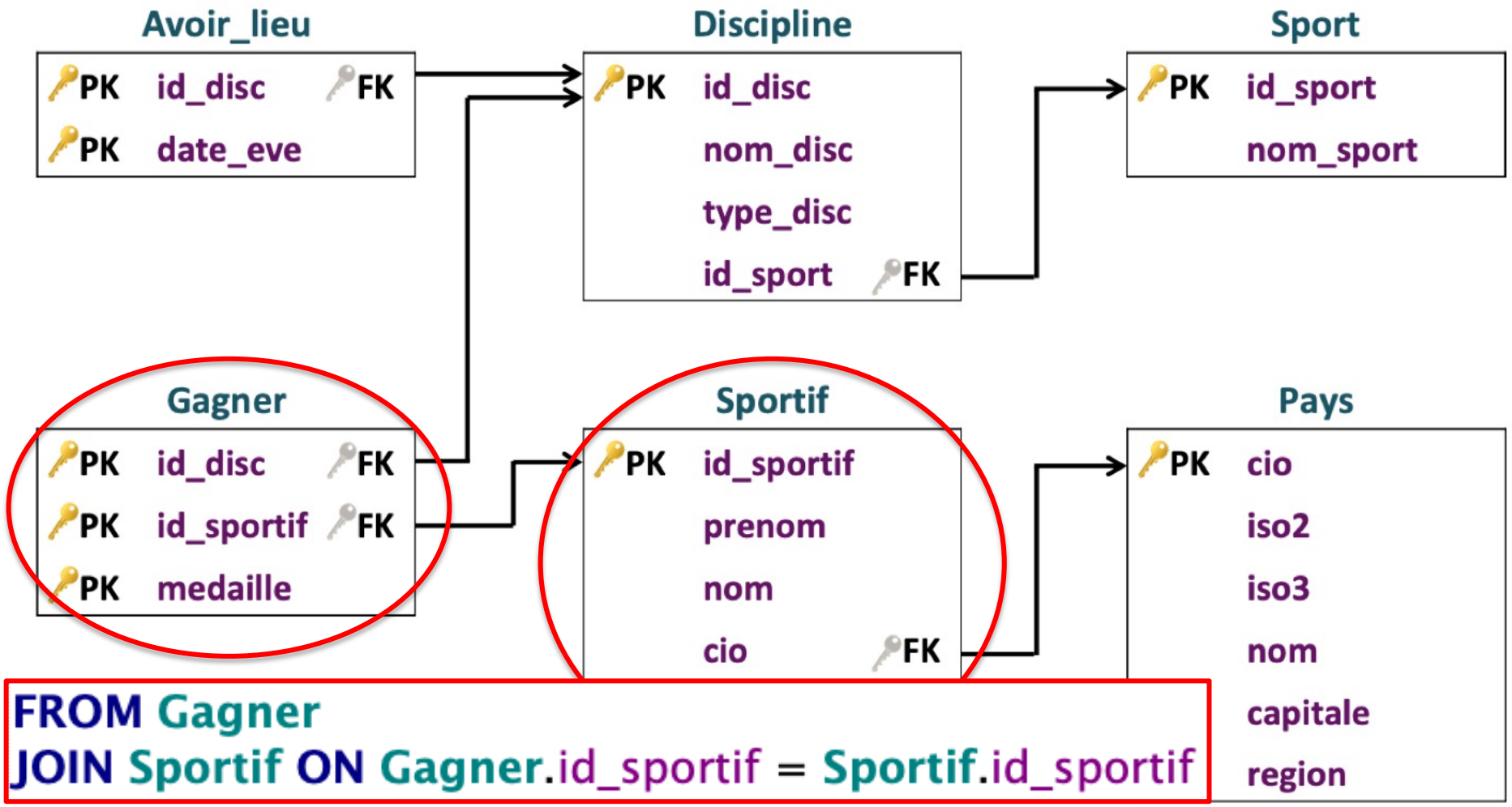
Question 10

Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.



Question 10

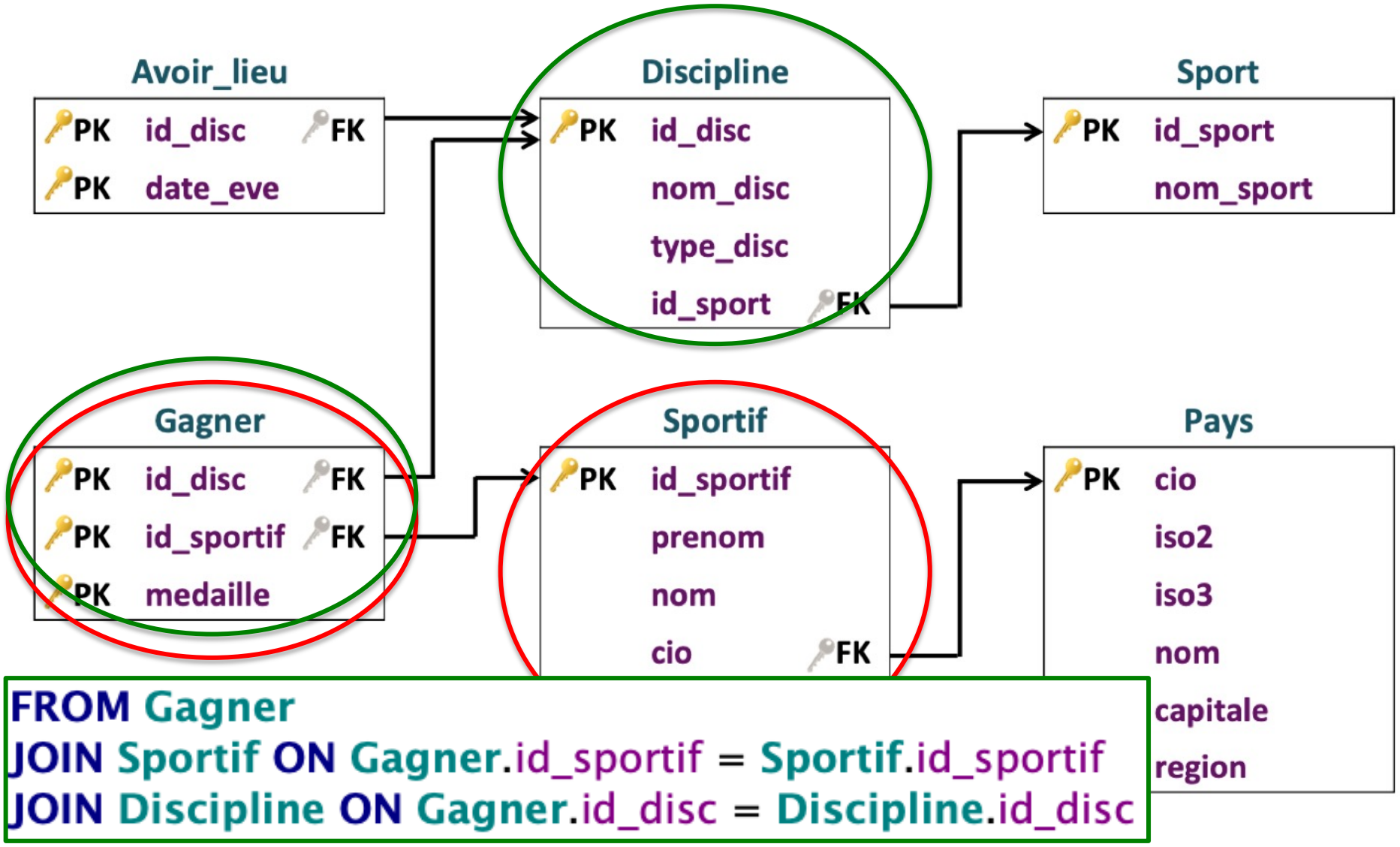
Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.




```
FROM Gagner
JOIN Sportif ON Gagner.id_sportif = Sportif.id_sportif
```

Question 10

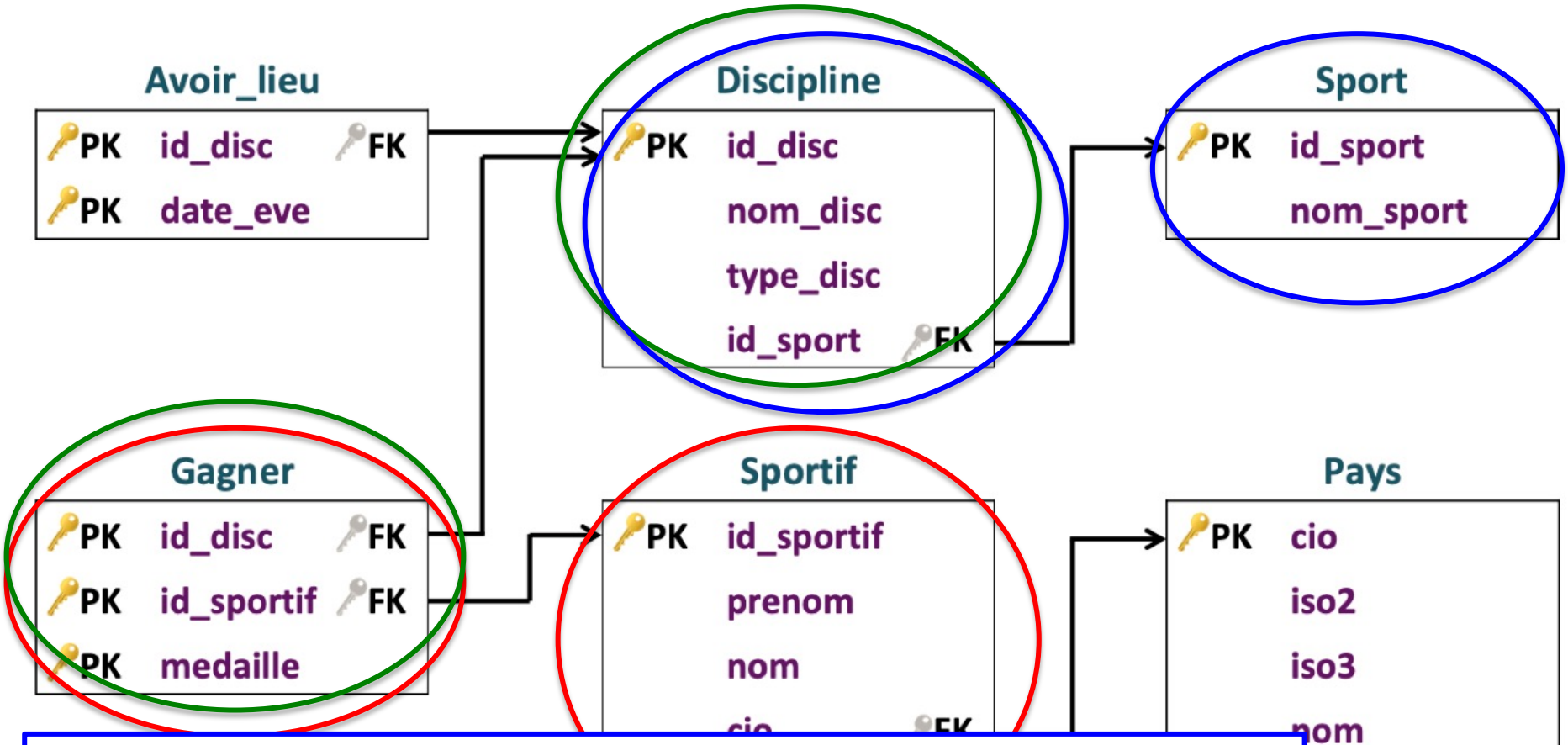
Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.




```
FROM Gagner
JOIN Sportif ON Gagner.id_sportif = Sportif.id_sportif
JOIN Discipline ON Gagner.id_disc = Discipline.id_disc
```

Question 10

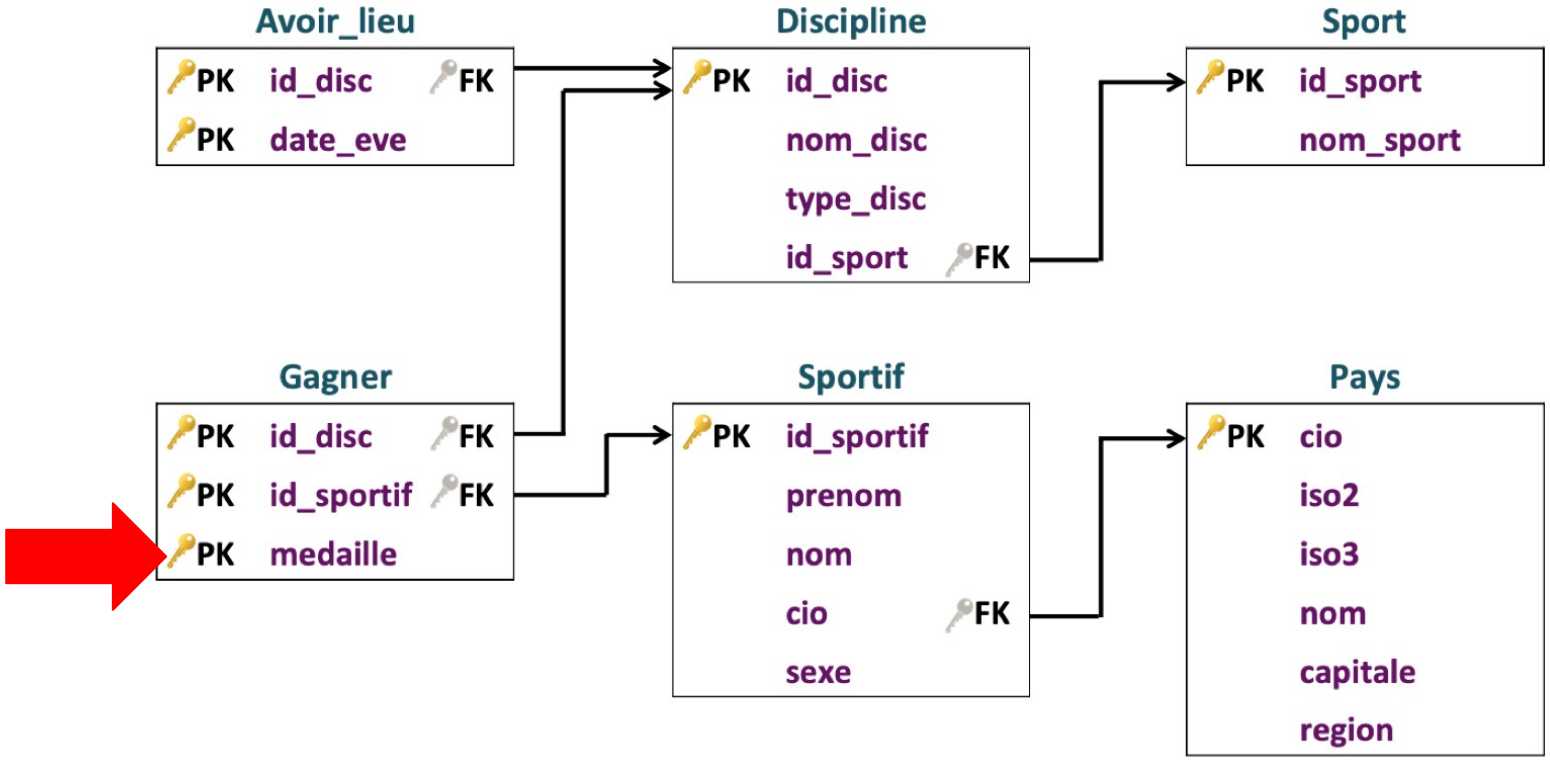
Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.




```
FROM Gagner
JOIN Sportif ON Gagner.id_sportif = Sportif.id_sportif
JOIN Discipline ON Gagner.id_disc = Discipline.id_disc
JOIN Sport ON Discipline.id_sport= Sport.id_sport
```

Question 10

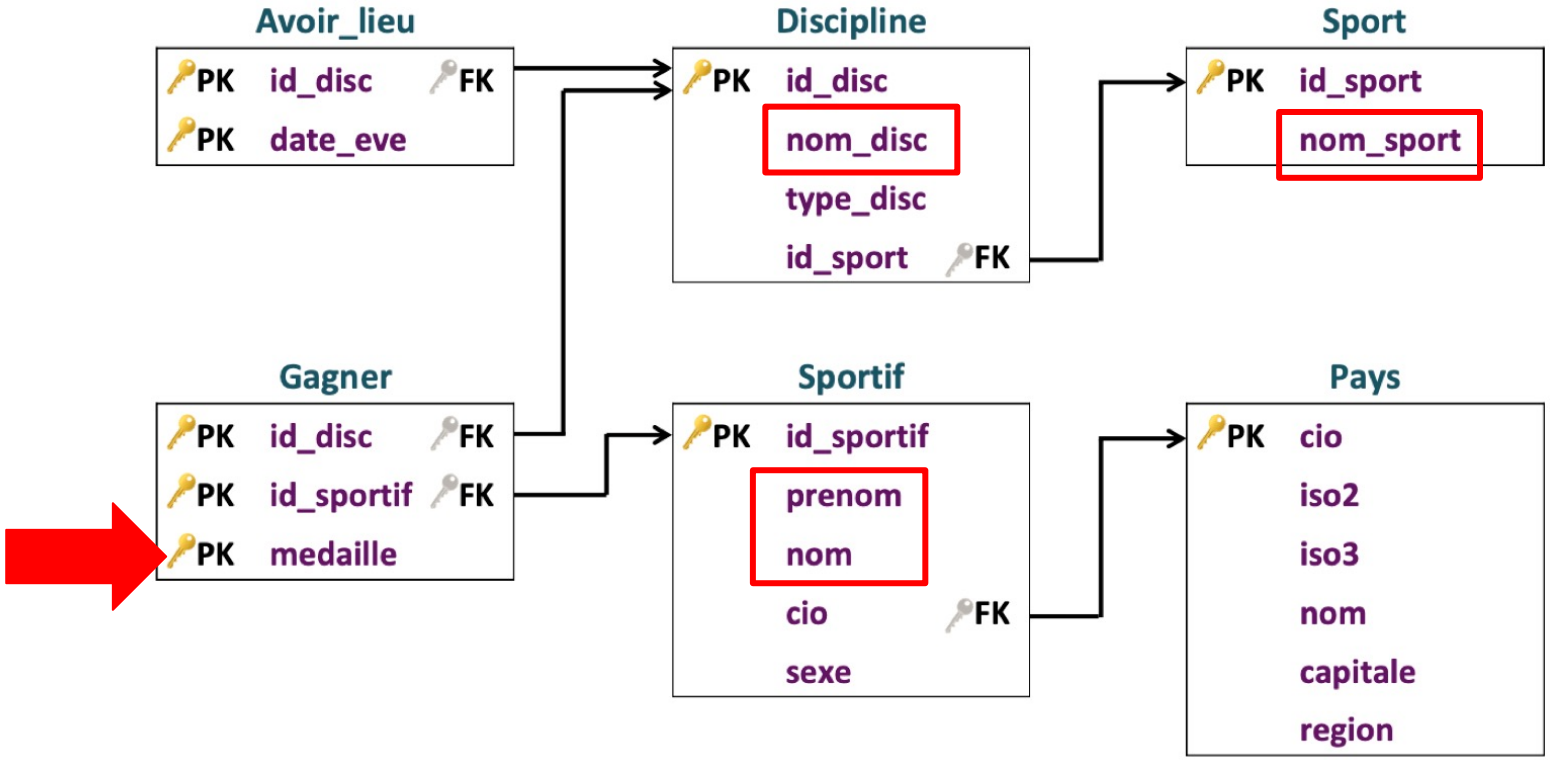
Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.




```
FROM Gagner
JOIN Sportif ON Gagner.id_sportif = Sportif.id_sportif
JOIN Discipline ON Gagner.id_disc = Discipline.id_disc
JOIN Sport ON Discipline.id_sport= Sport.id_sport
WHERE Gagner.medaille = 'G' ;
```

Question 10

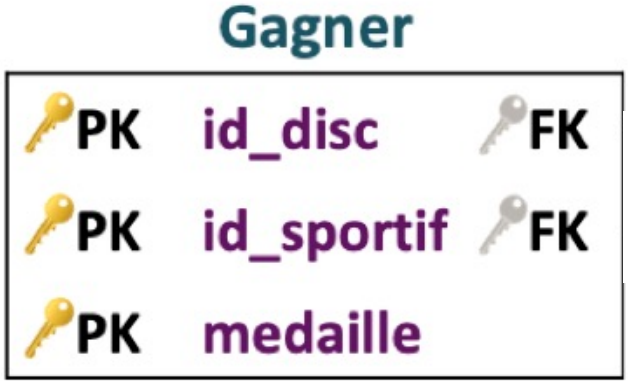
Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.




```sql
SELECT Sportif.nom, Sportif.prenom, Sport.nom_sport, Discipline.nom_disc
FROM Gagner
JOIN Sportif ON Gagner.id_sportif = Sportif.id_sportif
JOIN Discipline ON Gagner.id_disc = Discipline.id_disc
JOIN Sport ON Discipline.id_sport= Sport.id_sport
WHERE Gagner.medaille = 'G' ;
```

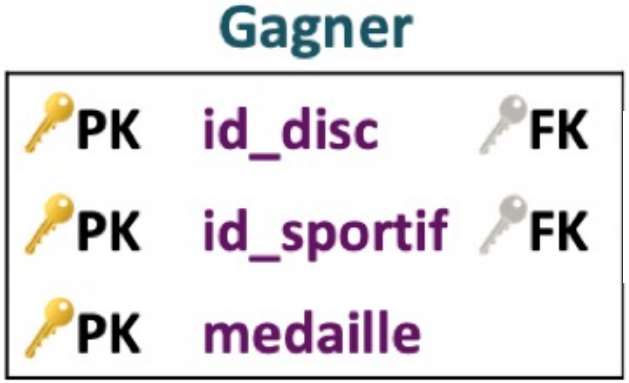
Question 11

Donner le nombre de médailles d'or attribuées lors de ces JO.



Question 11

Donner le nombre de médailles d'or attribuées lors de ces JO.




```
1   SELECT count(*) FROM Gagner WHERE medaille = 'G' ;
```

	count(*)
1	221

L'exécution s'est terminée sans erreur.
Résultat : 1 enregistrements ramenés en 482ms
À la ligne 1 :
SELECT count(*) FROM Gagner WHERE medaille = 'G' ;

Question 12	Combien de noms de familles différents sont portés par les sportifs ?

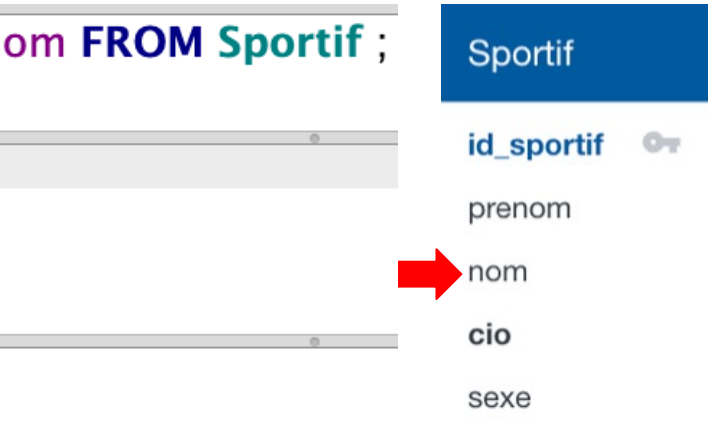
Question 12

Combien de noms de familles différents sont portés par les sportifs ?

```
1  SELECT DISTINCT nom FROM Sportif ;
```

	nom
1	LONGOSIWA
2	HARTING
3	MUTAI

```
L'exécution s'est terminée sans erreur.
Résultat : 626 enregistrements ramenés en 34ms
À la ligne 1 :
SELECT DISTINCT nom FROM Sportif ;
```



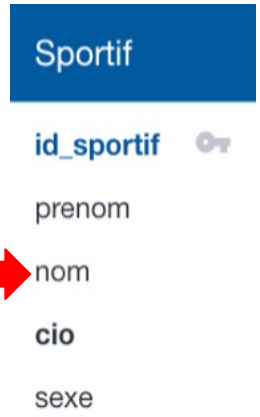
Question 12

Combien de noms de familles différents sont portés par les sportifs ?

```
1  SELECT DISTINCT nom FROM Sportif ;
```

	nom
1	LONGOSIWA
2	HARTING
3	MUTAI

L'exécution s'est terminée sans erreur.
Résultat : 626 enregistrements ramenés en 34ms
À la ligne 1 :
SELECT DISTINCT nom FROM Sportif ;



La requête renvoie une table, on peut la placer directement à la suite d'un FROM

Question 12

Combien de noms de familles différents sont portés par les sportifs ?

sous-requête

FROM

```
1  SELECT DISTINCT nom FROM Sportif ;
```

	nom
1	LONGOSIWA
2	HARTING
3	MUTAI

L'exécution s'est terminée sans erreur.
Résultat : 626 enregistrements ramenés en 34ms
À la ligne 1 :
SELECT DISTINCT nom FROM Sportif ;

Question 12 Combien de noms de familles différents sont portés par les sportifs ?

SELECT COUNT(*) FROM

sous-requête

```
1  SELECT DISTINCT nom FROM Sportif ;
```

	nom
1	LONGOSIWA
2	HARTING
3	MUTAI

L'exécution s'est terminée sans erreur.
Résultat : 626 enregistrements ramenés en 34ms
À la ligne 1 :
SELECT DISTINCT nom FROM Sportif ;

```
1  SELECT count(*) FROM (SELECT DISTINCT nom FROM Sportif) ;
```

	count(*)
1	626

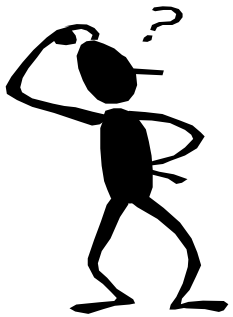
Partie 3 – Requêtes - **Bilan**

Requêtes basiques

SELECT… FROM… WHERE…

Requêtes avec fonctions d'agrégation

COUNT(*) AVG MIN MAX SUM



Requêtes imbriquées

SELECT …

FROM (SELECT … FROM …)

Requêtes avec jointures

JOIN … ON …

JOIN … ON …

Chapitre 7 - Bases de données

Séance 4 - Exercices

Exercice 1 CDI

Exercice 2 CinéTop

Exercice 3 Enquête dans SQL City

Exercice 4 Entraînement