S2 - TP 2 MySQL : requêtes, convertir un MCD en MLD

Soit le modèle relationnel suivant :

skieur(**idSkieur** , nomSkieur,*#specialite_id**,*#station_id**)
specialite(**idSpecialite**,libelleSpecialite)
station(**idStation**,nomStation,altitude,pays)
competition(**idCompetition**,libelleCompet,dateComp,*#station_id**)
classement(*#skieur_id**,*#competition_id**,classement)
comporte(#***competition_id*,#specialite_id****)

Un skieur appartient à une station. Une compétition se déroule dans une station.

Question 1

Écrire dans un fichier *script_tp2.sql* le script de création des tables correspondant au modèle ci dessus.

* Supprimer toutes les tables si elles existent.
* Créer toutes les tables si elles n'existent pas.
(Utiliser les mots clés **IF EXISTS** lors de la suppression de la table et **IF NOT EXISTS** lors de la création des tables.)
* Insérer les données des fichiers **csv** joints.

Question 2

Écrire les requêtes SQL suivantes :

1. Nombre de skieurs ayant participé à au moins une compétition.

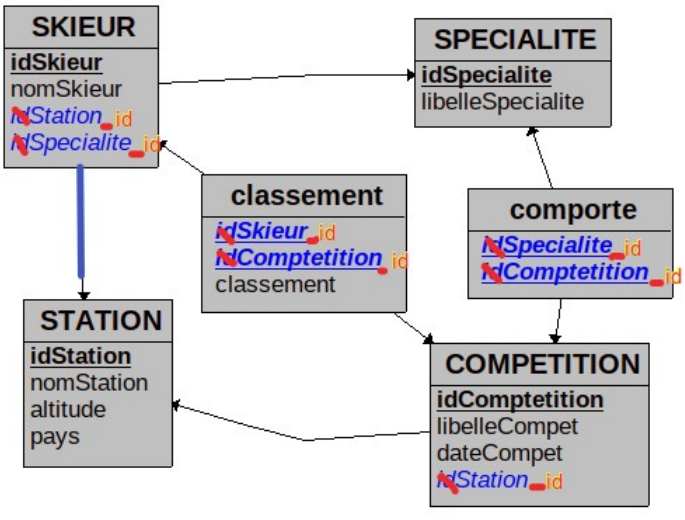
```
+-------------------------+
| NbreSkieurDansUneCompet |
+-------------------------+
|                       6 |
+-------------------------+
(exemple avec les données jointes)
```

```
SELECT COUNT(DISTINCT skieur_id) AS NbreSkieurDansUneCompet
FROM classement;
```

2. Nom de la station de chaque skieur (affichage : nom skieur + nom station, trier les lignes (tuples) par nom de station puis par nom de skieur)

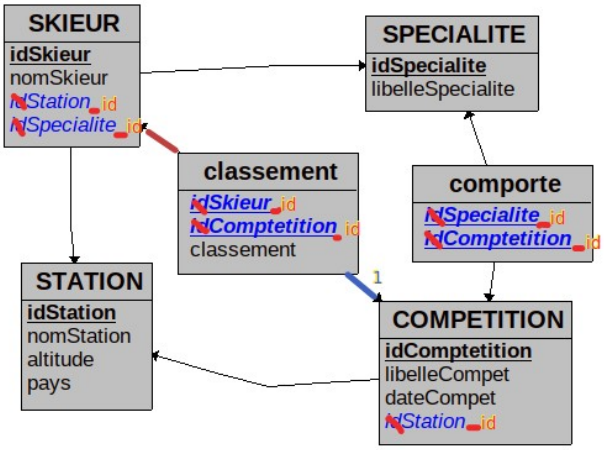
```
+-----------+--------------------+
| nomSkieur | nomStation         |
+-----------+--------------------+
| robert    | Chambery           |
| bernard   | Le Ballon d'alsace |
| alberto   | Metabief           |
| jacques   | Tignes             |
| paul      | Tignes             |
| pierre    | Tignes             |
| tom       | Tignes             |
| edouard   | Verbier            |
+-----------+--------------------+
(exemple avec les données jointes)
```

```
SELECT skieur.nomSkieur, station.nomStation
FROM skieur
INNER JOIN station
ON  skieur.station_id = station.idStation
ORDER BY  station.nomStation,skieur.nomSkieur;
```



3. Classement de la compétition de libellé ‘compet’ (affichage : nom skieur + classement, ordonnée les lignes(nuplets) par libellé de compétition puis par classement)

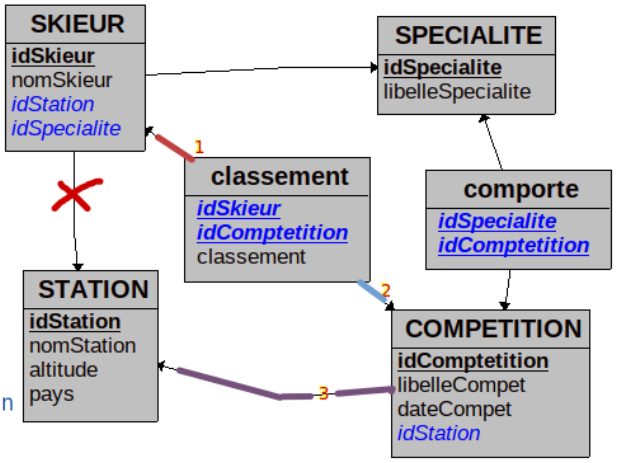
```
+-----------+------------+--------------------------+
| nomSkieur | classement | libelleCompet            |
+-----------+------------+--------------------------+
| pierre    |          1 | compet cadet france      |
| tom       |          2 | compet cadet france      |
| jacques   |          3 | compet cadet france      |
| robert    |          4 | compet cadet france      |
| edouard   |          5 | compet cadet france      |
| paul      |          1 | compet junior france     |
| pierre    |          2 | compet junior france     |
| robert    |          3 | compet junior france     |
| jacques   |          4 | compet junior france     |
| tom       |          5 | compet junior france     |
| edouard   |          6 | compet junior france     |
| paul      |          1 | competition cadet europe |
| jacques   |          2 | competition cadet europe |
| edouard   |          3 | competition cadet europe |
| robert    |          4 | competition cadet europe |
| jacques   |          1 | competition junior europe|
| pierre    |          2 | competition junior europe|
| edouard   |          3 | competition junior europe|
| robert    |          4 | competition junior europe|
+-----------+------------+--------------------------+
(exemple avec les données jointes)
```




```
SELECT sk.nomSkieur, cl.classement, ct.libelleCompet
FROM competition AS ct
INNER JOIN  classement AS cl ON ct.idCompetition = cl.competition_id
INNER JOIN  skieur AS sk ON cl.skieur_id = sk.idSkieur
WHERE
ct.libelleCompet like '%compet%'
ORDER BY ct.libelleCompet,cl.classement;
```

4. Liste des compétitions s’étant déroulées à ‘Tignes’ (nom de station) avec leur vainqueur (affichage : libellé Compétition, nom skieur vainqueur)

```
+--------------------------+-----------+
| libelleCompet            | nomSkieur |
+--------------------------+-----------+
| compet cadet france      | pierre    |
| compet junior france     | paul      |
| competition cadet europe | paul      |
+--------------------------+-----------+
(exemple avec les données jointes)
```




```
SELECT c.libelleCompet, sk.nomSkieur
FROM classement AS cl
INNER JOIN  skieur AS sk ON cl.skieur_id = sk.idSkieur
INNER JOIN competition AS c ON cl.competition_id = c.idCompetition
INNER JOIN station AS s ON c.station_id=s.idStation
WHERE  s.nomStation  like 'Tignes%'
AND cl.classement=1
[ORDER BY libelleCompet];
```

5. Nombre de compétitions se déroulant dans chaque station (affichage : id station + nom station + nb de compétition)

```
+-----------+------------+-------------+
| idStation | nomStation | nbrDeCompet |
+-----------+------------+-------------+
|         1 | Tignes     |           5 |
|         3 | Chambery   |           2 |
|         2 | Verbier    |           1 |
+-----------+------------+-------------+
(exemple avec les données jointes)
```

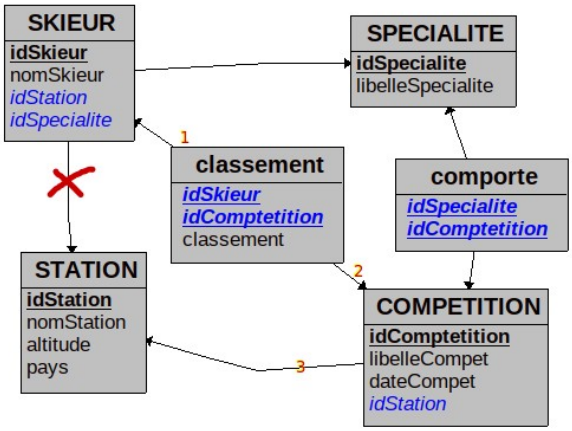
```
SELECT st.idStation,st.nomStation,
COUNT(st.idStation)as nbrDeCompet
FROM station st
INNER JOIN competition c
ON c.station_id = st.idStation
GROUP BY st.idStation,st.nomStation
[ORDER BY  nbrDeCompet DESC];
```

Ne pas confondre GROUP BY et ORDER BY
Remplacer COUNT(st.idStation) par COUNT(*)
Ajouter `ORDER BY  nbrDeCompet`

6. Nombre de victoires à ‘Tignes’ pour chaque skieur (affichage : id skieur + nom skieur + nb victoires)

```
+----------+-----------+-----------------+
| noSkieur | nomSkieur | NbreDeVictoires |
+----------+-----------+-----------------+
|        2 | pierre    |               1 |
|        3 | paul      |               2 |
+----------+-----------+-----------------+
(exemple avec les données jointes)
```

```
SELECT skieur.idSkieur, skieur.nomSkieur
, COUNT(*) as NbreDeVictoires
FROM classement
INNER JOIN  skieur
  ON classement.skieur_id = skieur.idSkieur
INNER JOIN competition
 ON classement.competition_id = competition.idCompetition
INNER JOIN station
 ON competition.station_id=station.idStation
WHERE
classement.classement=1
AND station.nomStation like 'Tignes'
GROUP BY skieur.idSkieur, skieur.nomSkieur
[ORDER BY  NbreDeVictoires];
```



7. Noms des skieurs ayant toujours terminé premier (nom skieur)

```
+-----------+
| nomSkieur |
+-----------+
| paul      |
+-----------+
(exemple avec les données jointes)
```

```
SELECT DISTINCT sk.nomSkieur
FROM classement cl
INNER JOIN skieur sk ON sk.idSkieur = cl.skieur_id
WHERE cl.classement = 1
AND sk.nomSkieur NOT IN (
    SELECT sk.nomSkieur
    FROM classement cl
    INNER JOIN skieur sk
        ON sk.idSkieur = cl.skieur_id
    WHERE cl.classement <>1
);
```

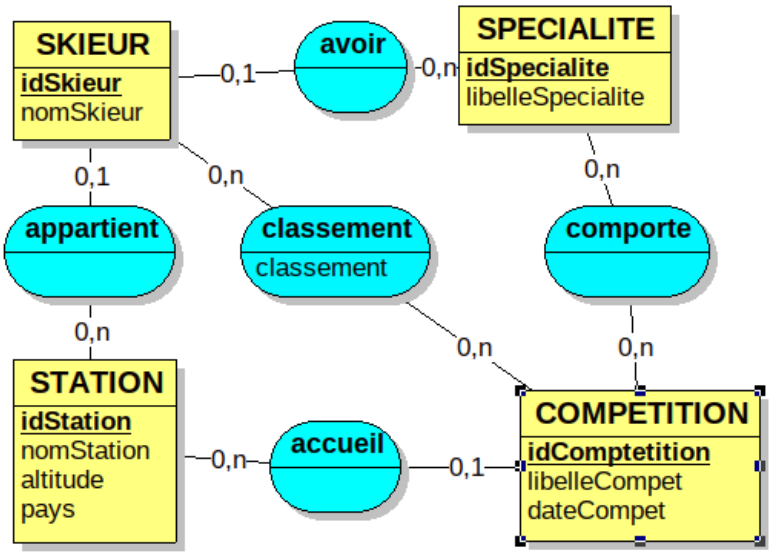
```
SELECT DISTINCT sk.nomSkieur
FROM classement cl
INNER JOIN skieur sk ON sk.idSkieur = cl.skieur_id
WHERE cl.classement = 1
AND sk.idSkieur NOT IN (
    SELECT cl.skieur_id
    FROM classement cl
    WHERE cl.classement <>1
);
```

Question 3

Télécharger un des 2 logiciels suivants :
* “Looping” **https://www.looping-mcd.fr/**
* “JMerise” **http://www.jfreesoft.com/JMerise/**

Utiliser un de ces 2 logiciels

(Démonstration en cours de ces 2 logiciels avec le MLD du TP précédent)
Établir le modèle conceptuel des données associé à ce modèle relationnel.



Pour les plus rapides :
Utiliser les nouveaux fichiers CSV :

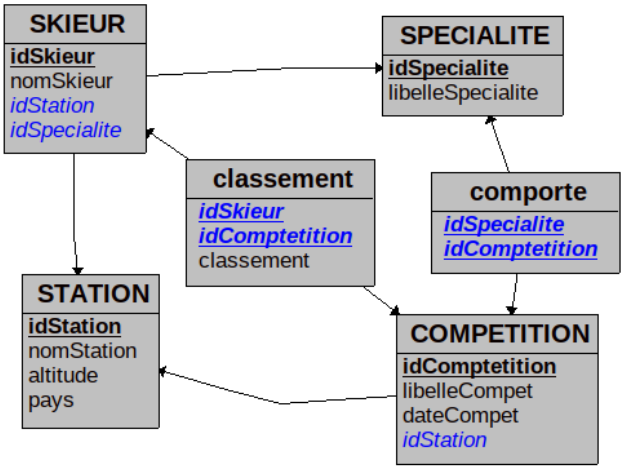
```
DELETE FROM classement;
DELETE FROM comporte;

-- DELETE FROM competition;    -- ne fonctionne pas
-- TRUNCATE competition;       -- ne fonctionne pas

SET FOREIGN_KEY_CHECKS = 0;
TRUNCATE competition;
SET FOREIGN_KEY_CHECKS = 1;

LOAD DATA LOCAL INFILE '/home/login/rep/COMPETITIONv2.csv' INTO TABLE competition
FIELDS TERMINATED BY ',';
LOAD DATA LOCAL INFILE '/home/login/rep/COMPORTEv2.csv' INTO TABLE comporte
FIELDS TERMINATED BY ',';
LOAD DATA LOCAL INFILE '/home/login/rep/CLASSEMENTv2.csv' INTO TABLE classement
FIELDS TERMINATED BY ',';
```

Ajouter des enregistrements pour vérifier les requêtes ci-sessous

R8 : Noms des skieurs ayant toujours terminé dans les 3 premiers à “Tigne”

```
+-----------+------------+-------------+----------+
| nomSkieur | classement | lieu_compet | club     |
+-----------+------------+-------------+----------+
| pierre    |          1 | Tignes      | Tignes   |
| pierre    |          2 | Tignes      | Tignes   |
| paul      |          1 | Tignes      | Tignes   |
| alberto   |          1 | Tignes      | Metabief |
+-----------+------------+-------------+----------+
```

```
SELECT DISTINCT sk.nomSkieur, cl.classement, st.nomStation ,  st2.nomStation
FROM classement cl
  INNER JOIN skieur sk ON sk.idSkieur = cl.skieur_id
  INNER JOIN competition ct ON ct.idCompetition = cl.competition_id
  INNER JOIN station st ON ct.station_id = st.idStation
  INNER JOIN station st2 ON sk.station_id = st2.idStation
WHERE   st.nomStation LIKE 'tignes'
AND   cl.classement <= 3
AND sk.idSkieur NOT IN (
    SELECT classement.skieur_id
    FROM classement
     INNER JOIN competition ON classement.competition_id = competition.idCompetition
     INNER JOIN station ON competition.station_id = station.idStation
    WHERE classement.classement > 3
       AND station.nomStation LIKE 'tignes'
);
```



R9 : Noms des skieurs ayant toujours terminé dans les 3 premiers “à la maison” (dans leur station)

```
SELECT DISTINCT sk.nomSkieur
FROM classement cl
INNER JOIN skieur sk ON sk.idSkieur = cl.skieur_id
INNER JOIN competition ct ON ct. idCompetition = cl.competition_id
WHERE
 sk.station_id = ct.station_id
AND cl.classement <= 3
AND sk.idSkieur NOT IN (
       SELECT cl2.skieur_id
       FROM classement cl2
       INNER JOIN competition ct2 ON ct2. idCompetition = cl2.competition_id
       INNER JOIN skieur sk2 ON sk2.idSkieur = cl2.skieur_id
       WHERE cl2.classement >3
      AND sk2.station_id = ct2.station_id
);
```

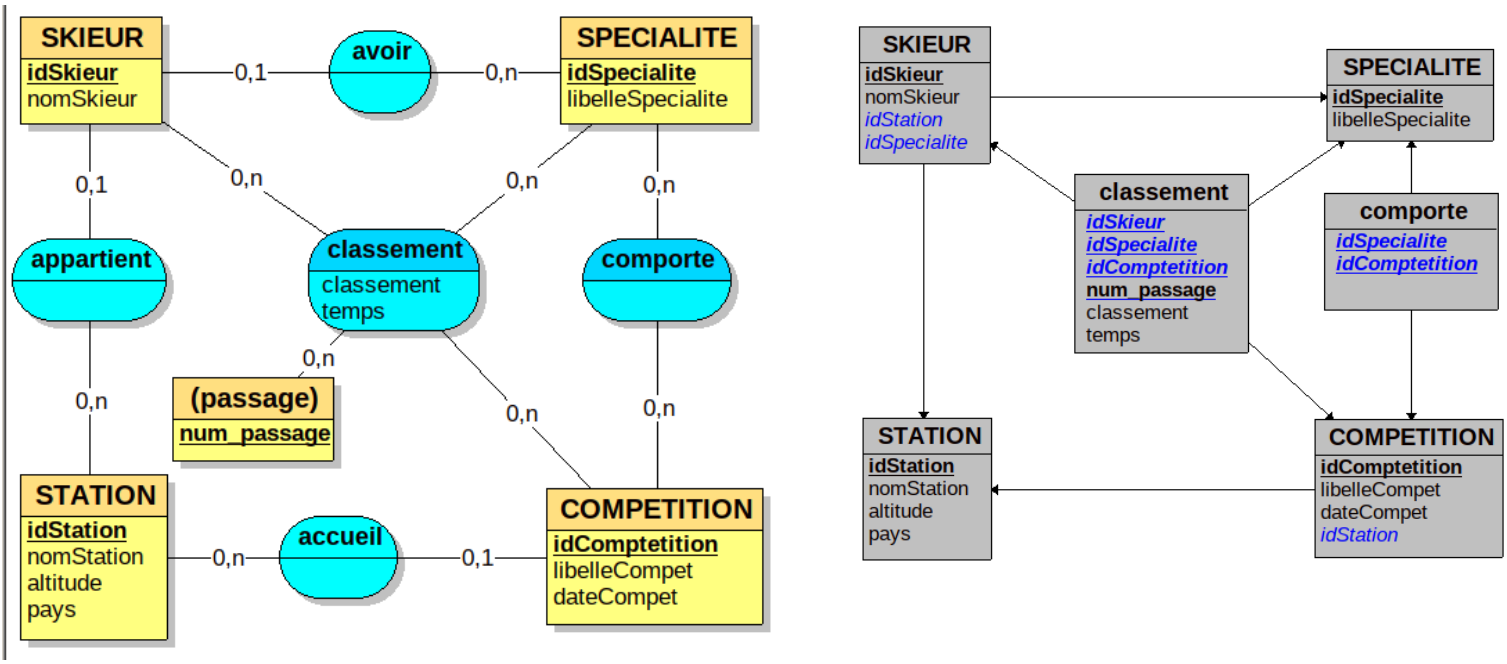
```
+-----------+
| nomSkieur |
+-----------+
| pierre    |
| paul      |
+-----------+
```

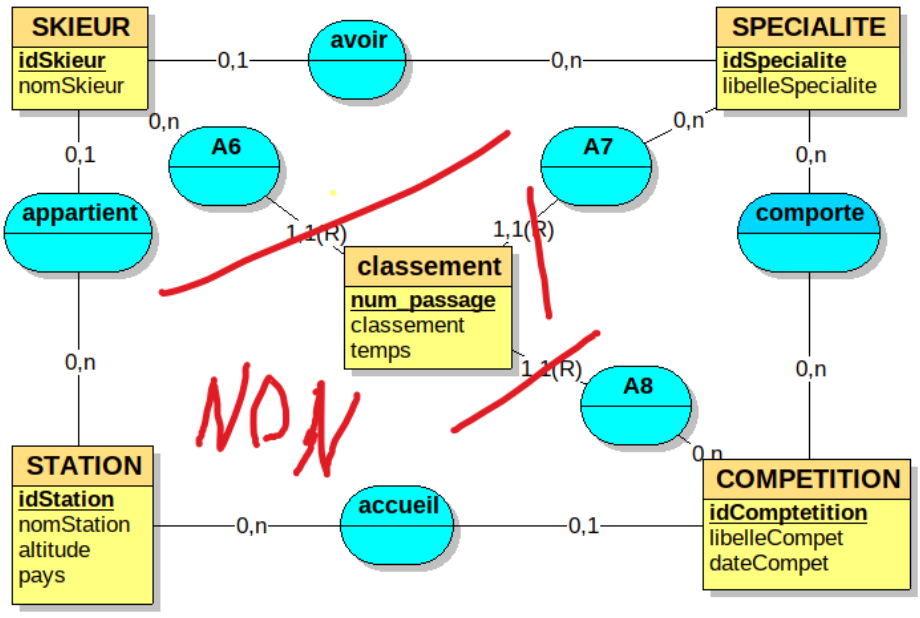
En vous aidant du document

http://perso.modulonet.fr/placurie/Ressources/BTS1-ALSI/Evaluation-5%20(correction).pdf (http://perso.modulonet.fr/placurie/Alsi1.htm)

R10 : Modifier le MLD : sans supprimer la table, ajouter une colonne dans la table « classement » de nom « spécialite_id », créer une clé primaire sur cette table, ajouter une colonne « temps ». Un skieur ne peut pas se classer plusieurs fois dans la même spécialité. Mais si on supposait que un skieur peut se classer plusieurs fois dans la même spécialité lors de la même compétition, il faudrait ajouter une colonne « passage » ou « heureDepart ». (2.3)

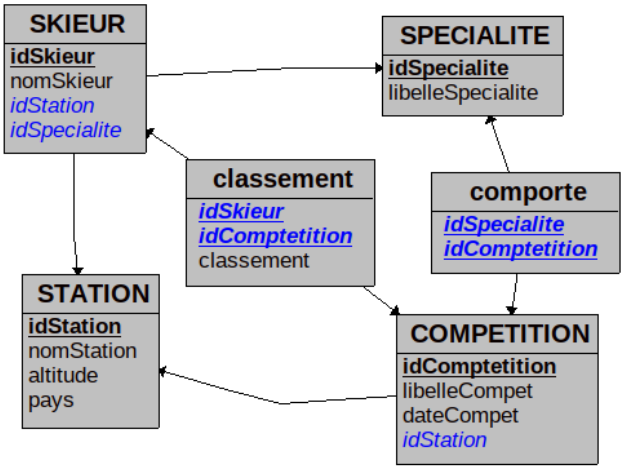
```
--  ALTER TABLE classement DROP PRIMARY KEY;
--  Inutile pas de clé primaire
ALTER TABLE  classement ADD specialite_id int AFTER classement;
ALTER TABLE  classement ADD temps time AFTER specialite_id;
ALTER TABLE  classement ADD passage int AFTER temps;
-- il faut vider la table DELETE FROM classement;
ALTER TABLE classement ADD CONSTRAINT PK_classement PRIMARY KEY
(skieur_id,competition_id,specialite_id,passage);
```





R11 : Donner la liste des stations dont sont originaires au moins 2 skieurs

```
SELECT station.nomStation
FROM station
INNER JOIN skieur
    ON station.idStation = skieur.station_id
GROUP BY station.nomStation
HAVING COUNT(skieur.idSkieur) >= 2;
```



R12 : Donner la liste des skieurs qui pratiquent la même spécialité que le skieur « paul »

```
SELECT skieur.nomSkieur
FROM skieur
WHERE skieur.specialite_id = (
    SELECT skieur.specialite_id
    FROM skieur
    WHERE skieur.nomSkieur LIKE 'paul'
    )
  AND skieur.nomSkieur NOT LIKE 'paul';
```

R13 : Insérer dans la table SKIEUR le skieur «alphand» qui a la même spécialité que le skieur «paul» et la même station que le skieur «pierre» (4.2)

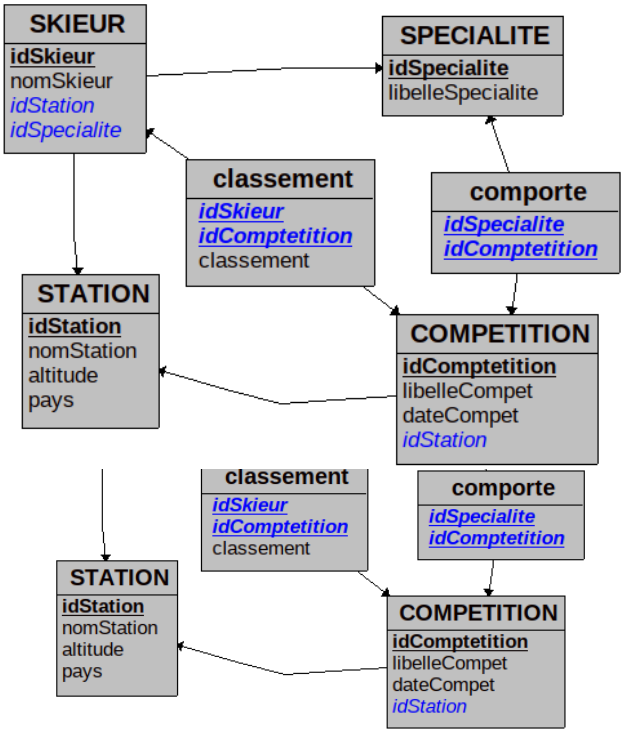
```
INSERT INTO skieur
SELECT NULL, 'alphand', sk1.specialite_id, sk2.station_id FROM skieur sk1, skieur sk2 WHERE
sk1.nomSkieur = 'paul' AND sk2.nomSkieur = 'pierre';


SELECT * FROM skieur;
```

R14 : Afficher les compétitions avec plus de 2 spécialités

```
SELECT competition.libelleCompet    -- ,COUNT(comporte.specialite_id)
FROM  comporte
INNER JOIN competition ON comporte.competition_id = competition.idCompet
GROUP BY competition.idCompet
HAVING COUNT(comporte.specialite_id) > 2;
```

R15 : Afficher les skieurs dans le même club que “tom”

```
SELECT skieur.nomSkieur
FROM skieur
WHERE skieur.nomSkieur not like 'tom'
and skieur.station_id = (
   SELECT skieur.station_id
    FROM skieur
    WHERE skieur.nomSkieur like 'tom'
    );
```

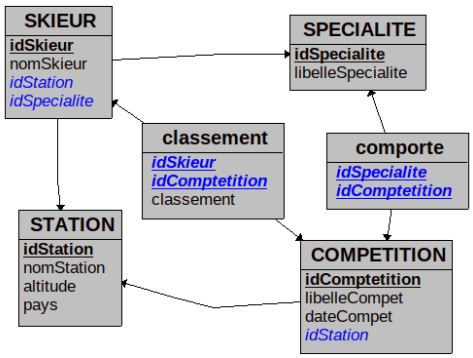


R16 : Afficher les skieurs avec un classement égal ou meilleur que celui de “tom”

```
# R16
SELECT DISTINCT skieur.nomSkieur, MIN(classement.classement)
FROM classement
INNER JOIN skieur ON skieur.idSkieur = classement.skieur_id
GROUP BY skieur.nomSkieur;


SELECT DISTINCT skieur.nomSkieur
FROM classement
INNER JOIN skieur ON skieur.idSkieur = classement.skieur_id
WHERE
classement.classement <= (
        SELECT min(classement.classement)
        FROM classement
        INNER JOIN skieur ON skieur.idSkieur = classement.skieur_id
        WHERE skieur.nomSkieur like 'tom'
    )
AND skieur.nomSkieur not like 'tom'
;
```

R17 : Afficher les skieurs qui ont plus ou au moins autant de victoires que “pierre”

```
 # R17
 SELECT DISTINCT skieur.nomSkieur, COUNT(classement.classement)
 FROM classement
 INNER JOIN skieur ON skieur.idSkieur = classement.skieur_id
 WHERE classement.classement=1
 GROUP BY skieur.nomSkieur;

 SELECT skieur.nomSkieur  -- ,COUNT(classement.classement)
 FROM classement
 INNER JOIN skieur ON skieur.idSkieur = classement.skieur_id
 WHERE
 classement.classement=1
 AND skieur.nomSkieur not like 'pierre'
 GROUP BY skieur.nomSkieur
 HAVING COUNT(classement.classement) >=
  (
         SELECT count(classement.classement)
         FROM classement
                  INNER JOIN skieur ON skieur.idSkieur = classement.skieur_id
         WHERE skieur.nomSkieur like 'pierre' AND classement.classement=1
  )
 ;
```



R18 : Créer une transaction qui supprime la station «chambery» et déplace toutes les compétions dans la station «valoire» (4.4)

```
START TRANSACTION;
--  SELECT competition.idCompet, station.nomStation
--  FROM station
--  INNER JOIN competition ON competition.station_id = station.idStation
--  WHERE station.nomStation LIKE 'chambery';

UPDATE competition SET
station_id =(select idStation from station where nomStation like 'valoire')
WHERE station_id in (SELECT station.idStation
FROM station
WHERE station.nomStation LIKE 'chambery');

UPDATE skieur SET
station_id =(select idStation from station where nomStation like 'valoire')
WHERE station_id in (SELECT station.idStation
FROM station
WHERE station.nomStation LIKE 'chambery');

DELETE FROM station WHERE station.nomStation LIKE 'chambery';

--  SELECT competition.idCompet, station.nomStation
--  FROM station
--  INNER JOIN competition ON competition.station_id = station.idStation
--  WHERE station.nomStation LIKE 'chambery';

COMMIT
```