

Interrogation et création d'une base de données : Gestion des courses

1 Requetes LMD

1- Liste des coureurs entraînés par WEBER.
(num Coureur, nom coureur, ville)

```
SELECT
NumCoureur
, NomCoureur
, VilleCoureur
FROM
Coureur, Entraîneur
WHERE
Coureur.NumEntr = Entraîneur.NumEntr
AND
Entraîneur.NomEntr LIKE 'WEBER'
;
```

```
SELECT
NumCoureur
, NomCoureur
, VilleCoureur
FROM
Coureur
INNER JOIN
Entraîneur
ON
Entraîneur.NumEntr = Coureur.NumEntr
WHERE
Entraîneur.NomEntr LIKE 'WEBER'
;
```

2- Qui a gagné la "course du Lion" du 12/09/04 ?
(nom coureur, ville)

```
SELECT
```

```
Coureur.NomCoureur
, Coureur.VilleCoureur
FROM
Coureur
, Participe
, Course
WHERE
Coureur.NumCoureur = Participe.NumCoureur
AND
Participe.NumCourse = Course.NumCourse
AND
Course.DateCourse = '12/09/04'
AND
Participe.Ordre = 1
;
```

```
SELECT
Coureur.NomCoureur
, Coureur.VilleCoureur
FROM
Coureur
INNER JOIN
Participe
ON
Participe.NumCoureur = Coureur.NumCoureur
INNER JOIN
Course
ON
Course.NumCourse = Participe.NumCourse
WHERE
Course.DateCourse = '12/09/04'
AND
Participe.Ordre = 1

;
```

3- Liste des participants à la "course du Lion" du 12/09/04, classée par ordre croissant d'ordre d'arrivée. (nom coureur, ordre).

```
SELECT
Coureur.NomCoureur
```

```
, Participe.Ordre
FROM
Coureur
, Participe
, Course
WHERE
Coureur.NumCoureur = Participe.NumCoureur
AND
Participe.NumCourse = Course.NumCourse
AND
Course.DateCourse = '12/09/04'
ORDER BY
Participe.Ordre
;
```

```
SELECT
Coureur.NomCoureur
, Participe.Ordre
FROM
Coureur
INNER JOIN
Participe
ON
Participe.NumCoureur = Coureur.NumCoureur
INNER JOIN
Course
ON
Course.NumCourse = Participe.NumCourse
WHERE
Course.DateCourse = '12/09/04'
ORDER BY
Participe.Ordre
;
```

4- A quelles courses ont participé les coureurs entraînés par WEBER ?
(libellé course, date, ville, n. coureur)

```
SELECT
Course.LibCourse
, Course.DateCourse
, Course.VilleCourse
```

```
, Participe.NumCoureur
FROM
Course
, Participe
, Entraîneur
, Coureur
WHERE
Course.NumCourse = Participe.NumCourse
AND
Participe.NumCoureur = Coureur.NumCoureur
AND
Coureur.NumEntr = Entraîneur.NumEntr
AND
Entraîneur.NomEntr LIKE 'WEBER'
;
```

```
SELECT
Course.LibCourse
, Course.DateCourse
, Course.VilleCourse
, Participe.NumCoureur
FROM
Course
INNER JOIN
Participe
ON
Participe.NumCourse = Course.NumCourse
INNER JOIN
Coureur
ON
Coureur.NumCoureur = Participe.NumCoureur
INNER JOIN
Entraîneur
ON
Entraîneur.NumEntr = Coureur.NumEntr
WHERE
Entraîneur.NomEntr LIKE 'WEBER';
```

5- Liste des coureurs qui ont déjà gagné une course et nom de leur entraîneur.

(nom coureur, libellé course, date course, nom entraîneur).

```
SELECT
Coureur.NomCoureur
, Course.LibCourse
, Course.DateCourse
, Entraîneur.NomEntr
FROM
Coureur
, Course
, Entraîneur
, Participe
WHERE
Coureur.NumEntr = Entraîneur.NumEntr
AND
Coureur.NumCoureur = Participe.NumCoureur
AND
Course.NumCourse = Participe.NumCourse
AND
Participe.Ordre = 1
;
```

```
SELECT
Coureur.NomCoureur
, Course.LibCourse
, Course.DateCourse
, Entraîneur.NomEntr
FROM
Coureur
INNER JOIN
Entraîneur
ON
Entraîneur.NumEntr = Coureur.NumEntr
INNER JOIN
Participe
ON
Participe.NumCoureur = Coureur.NumCoureur
INNER JOIN
Course
ON
Course.NumCourse = Participe.NumCourse
WHERE
Participe.Ordre = 1
```

;

6- Pour chaque coureur donnez son adresse et pour ceux qui ont couru, la liste des courses auxquelles ils ont participé.
(numcoureur, ville, numcourse, libcourse)

```
SELECT
Coureur.NumCoureur
, Coureur.VilleCoureur
, Participe.NumCourse
, Course.LibCourse
FROM
Coureur
FULL OUTER JOIN
Participe
ON
Participe.NumCoureur = Coureur.NumCoureur
FULL OUTER JOIN
Course
ON
Participe.NumCourse = Course.NumCourse
;
```

7- Quels sont les courses qui se sont déroulées dans la ville où habite DUPONT (Libellé course, date course)

```
SELECT
Course.LibCourse
, Course.DateCourse
FROM
Course
WHERE
VilleCourse LIKE (SELECT VilleCoureur
FROM Coureur
WHERE NomCoureur LIKE 'DUPONT')
```

;

8- Liste des coureurs qui sont entraînés par FANNIER et qui n'ont jamais couru.
(nomcoureur) -- Utilisation du not exists

```
SELECT
```

```
Coureur.NumCoureur
, Coureur.NomCoureur
FROM
Coureur
WHERE
NOT EXISTS
(SELECT
Participe.NumCoureur
FROM
Participe
INNER JOIN
Coureur
ON
Participe.NumCoureur = Coureur.NumCoureur
INNER JOIN
Entraîneur
ON
Coureur.NumEntr = Entraîneur.NumEntr
WHERE
Entraîneur.NomEntr LIKE 'FANNIER')
;
```

Note : Utilisation de NOT IN ici / Echec des utilisations de NOT EXISTS

```
SELECT
NomCoureur
FROM
ENTRAINEUR
INNER JOIN
COUREUR
USING (NUMENTR)
WHERE
NOMENTR='FANNIER'
AND NOT EXISTS (
SELECT
NUMCOUREUR
FROM
PARTICIPE
WHERE
PARTICIPE.NUMCOUREUR=COUREUR.NUMCOUREUR
);
```

9-

```
SELECT
NOMCOUREUR
FROM
COUREUR
INNER JOIN
PARTICIPE
USING (NUMCOUREUR)
INNER JOIN
COURSE
USING (NUMCOURSE)
WHERE
NUMCOURSE='18'
AND
ORDRE < (SELECT
ORDRE
FROM
PARTICIPE
INNER JOIN
COUREUR
USING (NUMCOUREUR)
WHERE
NUMCOURSE = '18' AND
NOMCOUREUR = 'ALBERT');
```

10-

```
SELECT
NOMCOUREUR
, NOMENTR
FROM
ENTRAINEUR
INNER JOIN
COUREUR
USING (NUMENTR)
WHERE
VILLECOUREUR = 'BELFORT'
OR
NOMENTR = 'WEBER';
```

11-

```
SELECT
NOMCOUREUR
```

```
FROM
COUREUR
WHERE
NOMCOUREUR NOT IN (
SELECT
NOMCOUREUR
FROM
COUREUR
INNER JOIN
PARTICIPE
USING (NUMCOUREUR)
INNER JOIN
COURSE
USING (NUMCOURSE)
WHERE
LIBCOURSE='Course du Lion'
AND
DATECOURSE LIKE '%05');
```

```
12-
SELECT
NOMCOUREUR
FROM
ENTRAINEUR
INNER JOIN
COUREUR
USING (NUMENTR)
WHERE
VILLECOUREUR='BELFORT'
AND
NOMENTR='DUBROCK'
UNION
SELECT
NOMCOUREUR
FROM
ENTRAINEUR
INNER JOIN
COUREUR
USING (NUMENTR)
WHERE
VILLECOUREUR='MONTBELIARD'
AND
```

```
NOMENTR='DUBROCK' ;
```

```
13-
```

```
SELECT
NOMCOUREUR
, LIBCOURSE
FROM
COUREUR
INNER JOIN
PARTICIPE
USING (NUMCOUREUR)
INNER JOIN
COURSE
USING (NUMCOURSE)
WHERE
DATECOURSE='17/01/05'
UNION
SELECT
NOMCOUREUR
, LIBCOURSE
FROM
COUREUR
INNER JOIN
PARTICIPE
USING (NUMCOUREUR)
INNER JOIN
COURSE
USING (NUMCOURSE)
WHERE
DATECOURSE='25/01/05' ;
```

```
14-
```

```
SELECT
NOMCOUREUR
FROM
COUREUR
MINUS
SELECT
NOMCOUREUR
FROM
COUREUR
INNER JOIN
```

```
PARTICIPE
USING (NUMCOUREUR)
INNER JOIN
COURSE
USING (NUMCOURSE)
WHERE
DATECOURSE='17/01/05';
```

```
15-
SELECT
NOMCOUREUR
FROM
COUREUR
INNER JOIN
PARTICIPE
USING (NUMCOUREUR)
INNER JOIN
COURSE
USING (NUMCOURSE)
WHERE
DATECOURSE='17/01/05';
INTERSECT
SELECT
NOMCOUREUR
FROM
COUREUR
INNER JOIN
PARTICIPE
USING (NUMCOUREUR)
INNER JOIN
COURSE
USING (NUMCOURSE)
WHERE
DATECOURSE='25/01/05';
```

```
16-
SELECT
NUMCOUREUR
, NOMCOUREUR
FROM
ENTRAINEUR
INNER JOIN
```

```
COUREUR
USING (NUMENTR)
WHERE
VILLEENTR='BELFORT'
AND
(NUMCOUREUR='2'
OR NUMCOUREUR='7'
OR NUMCOUREUR='8');
```

17-

Liste des coureurs qui ont gagné toutes les courses auxquelles ils ont participé

```
SELECT DISTINCT
NomCoureur
FROM
Coureur
INNER JOIN
Participe
USING(NumCoureur)
WHERE
NumCoureur
NOT IN
(SELECT
NumCoureur
FROM
Participe
WHERE
Ordre != 1);
```

18-

Noms des coureurs qui ont participé exactement aux memes courses que martin

selection de tous les coureurs ayant couru aux cotés de Martin

```
SELECT DISTINCT
NomCoureur
FROM
Coureur
INNER JOIN
```

```
Participe
USING (NumCoureur)
WHERE
NumCourse IN (SELECT
NumCourse
FROM
Participe
INNER JOIN
Coureur
USING (NumCoureur)
WHERE
NomCoureur = 'MARTIN')
AND
NomCoureur != 'MARTIN'
ORDER BY
NomCoureur;
```

```
19-
SELECT
NOMENTR
, COUNT(NUMENTR)
FROM
ENTRAINEUR
INNER JOIN
COUREUR
USING (NUMENTR)
GROUP BY
NONENTR
HAVING
COUNT(NUMENTR) > 2;
```

```
20-
SELECT
NONENTR
, COUNT(*)
FROM
ENTRAINEUR
INNER JOIN
COUREUR
USING (NUMENTR)
GROUP BY
NONENTR
```

```
UNION
SELECT
VILLECOUREUR
, COUNT(*)
FROM
ENTRAINEUR
INNER JOIN
COUREUR
USING (NUMENTR)
GROUP BY
VILLECOUREUR
```

```
21-
SELECT
VILLEENTR
, AVG(SALAIREENTR)
FROM
ENTRAINEUR
GROUP BY
VILLEENTR
ORDER BY
VILLEENTR
```

```
22-
SELECT
NOMCOUREUR
, COUNT(*)
FROM
COUREUR
INNER JOIN
PARTICIPE
USING (NUMCOUREUR)
INNER JOIN
COURSE
USING (NUMCOURSE)
WHERE
ORDRE < 4
GROUP BY
NOMCOUREUR
```

23- Liste des coureurs qui ont déjà participé à plus de 2 courses.

(nom coureur, nbcourses)

```
SELECT
NomCoureur
, COUNT(NumCourse) AS NbCourse
FROM
Coureur
INNER JOIN
Participe
USING (NumCoureur)
HAVING
COUNT(NumCourse) >= 2
GROUP BY
NomCoureur
```

24- Nombre de participants par course disputée à Belfort après le 18/01/05 (numérocourse, libellé course, nbparticipants)

```
SELECT
Numcourse
, LibCourse
, COUNT(NumCoureur) AS NbParticipant
FROM
Course
INNER JOIN
Participe
USING (NumCourse)
WHERE
VilleCourse = 'BELFORT'
AND
DateCourse > '18/01/05'
GROUP BY
NumCourse
, LibCourse;
```

25- Courses dans lesquelles ont participé plus de 5 coureurs.
(libellé course, nbcoureurs)

```
SELECT
LibCourse
, COUNT(NumCoureur) AS NbCoureurs
FROM
```

```
Course
INNER JOIN
Participe
USING (NumCourse)
HAVING COUNT(NumCoureur) >= 5
GROUP BY
NumCourse
, LibCourse;
```

26- Liste des coureurs qui habitent Belfort et qui ont participé à plus de 2 courses dans cette ville.
(numcoureur, nomcoureur)

27- Quels sont les entraîneurs qui gagnent plus de 7000F et qui entraînent 3 coureurs ? (nomentraîneur, salaire, nbcoueurs)

```
SELECT
NomEntr
, SalaireEntr
, COUNT(NumCoureur) AS NbCoureur
FROM
Entraîneur
INNER JOIN
Coureur
ON
Coureur.NumEntr = Entraîneur.NumEntr
WHERE
SalaireEntr >= 7000
HAVING COUNT(NumCoureur) = 3
GROUP BY
NomEntr
, SalaireEntr
```

28- Nombre de coureurs qui n'ont pas encore couru.

```
SELECT
COUNT(NumCoureur) AS NbCoureur
FROM
Coureur
WHERE
```

```
NumCoureur NOT IN (SELECT NumCoureur FROM Participe)
;
```

29- Quel est le nombre d'entraîneurs qui habitent Belfort et qui entraînent plus de 2 coureurs différents.
(nbentraîneur) Rédiger une requête avec un group by
Rédiger une requête sans group by

```
SELECT
COUNT(NumEntr) AS NbEntraîneur
FROM
Entraîneur
WHERE
VilleEntr = 'BELFORT'
AND
NumEntr IN ( SELECT
NumEntr
FROM
Entraîneur
INNER JOIN
Coureur
USING(NumEntr)
HAVING COUNT(NumCoureur) > 2
GROUP BY
NumEntr
)
```

30- Liste des coureurs, avec pour ceux qui ont couru, le nombre de courses auxquelles ils ont participé.
(NomCoureur, Nbcourses)

```
SELECT
NomCoureur
, COUNT(NumCourse) AS NbCourses
FROM
Coureur
FULL OUTER JOIN
Participe
ON
Participe.NumCoureur = Coureur.NumCoureur
GROUP BY
```

NomCoureur

31- Noms des coureurs qui ont participé à au moins une course à laquelle a participé 'Martin'. Autojointure ou sous-requête

```
SELECT DISTINCT
NomCoureur
FROM
Coureur
INNER JOIN
Participe
ON
Participe.NumCoureur = Coureur.NumCoureur
WHERE
Participe.NumCourse IN ( SELECT
NumCourse
FROM
Participe
INNER JOIN
Coureur
USING (NumCoureur)
WHERE NomCoureur = 'MARTIN')
;
```

32- Liste des courses qui se sont déroulés il y plus de 2 ans.
(libelle course, datecourse) Deux solutions demandées :
l'une utilisera la fonction add_months

```
SELECT
LibCourse
, DateCourse
FROM
Course
WHERE
DateCourse < '%/04/05';
```

```
SELECT
LibCourse
, DateCourse
FROM
Course
WHERE
```

```
ADD_MONTHS(DateCourse, 24) < SYSDATE;
```

33- Entraîneurs dont le salaire est supérieur au salaire moyen de tous les entraîneurs. Solution 1 : (nomentraîneur, salaire)
Solution 2 : (nomentraîneur, salaire, salmoyen)

```
SELECT
NomEntr
, SalaireEntr
FROM
Entraîneur
WHERE SalaireEntr > ( SELECT
AVG(SalaireEntr)
FROM
Entraîneur)
```

```
SELECT
NomEntr
, SalaireEntr
, AVG (SalaireEntr) AS SalMoyen
FROM
Entraîneur
WHERE SalaireEntr > ( SELECT
AVG(SalaireEntr)
FROM
Entraîneur)
GROUP BY
NomEntr
, SalaireEntr
```

Dans la deuxième version de la requête, le calcul du salaire moyen est incohérent.

34- Liste des coureurs qui ont gagné plus de courses que 'Martin'.
(nomcoureur, nbcourses)

```
SELECT
NomCoureur
, COUNT(NumCourse) AS NbCourse
FROM
Coureur
INNER JOIN
```

```
Participe
ON
Participe.NumCoureur = Participe.NumCoureur
WHERE
Ordre = 1
HAVING
COUNT(NumCourse) > (SELECT
COUNT(NumCourse)
FROM
Participe
INNER JOIN
Coureur
ON
Coureur.NumCoureur = Participe.NumCoureur
WHERE
NomCOureur = 'MARTIN'
AND
Ordre = 1)
GROUP BY
NomCoureur;
```

Résultat incohérent ...

35- Course(s) dans la(les)quelle(s) il y a eu le plus de participants.
(libellé course, nbcoureurs)

```
SELECT
LibCourse
, COUNT(NumCoureur) AS NbCoureurs
FROM
Course
INNER JOIN
Participe
ON
Participe.NumCourse = Course.NumCourse
HAVING COUNT(NumCoureur) >= MAX(SELECT
COUNT(NumCoureur) AS NbCoureurs
FROM
Participe);
```

36- Liste des coureurs qui ont participé à toutes les courses

qui se sont déroulées à Belfort durant l'année 2005.
 (numcoureur) Deux solutions : avec comptage et not exists

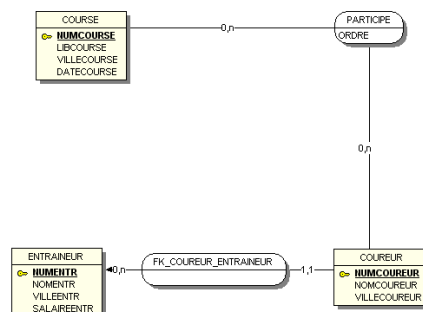
```

SELECT
Coureur.NumCoureur
FROM
Coureur
INNER JOIN
Participe
ON
participe.NumCoureur = Coureur.NumCoureur
WHERE
Numcourse IN ( SELECT
NumCourse
FROM
Course
WHERE
VilleCourse = 'BELFORT'
AND
DateCourse < '01/01/06'
AND
DateCourse >= '01/01/05'
);

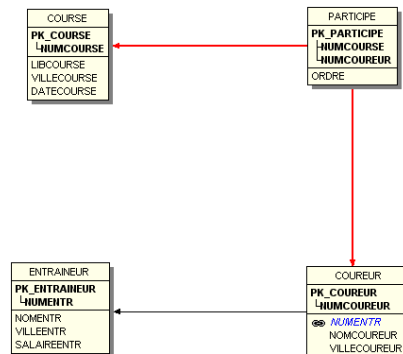
```

2 Génération du script SQL du modèle physique et chargement des tables

- MEA



- MLR



- Script SQL pour serveur Oracle

```

-----
--          Génération d'une base de données pour
--          Oracle Version 9i
--          (11/4/2007 14:23:58)
-----
--  Nom de la base : partieIIregen
--  Projet : tp4
--  Auteur : Génie Informatique
--  Date de dernière modification : 11/4/2007 14:23:34
-----

```

```
DROP TABLE COUREUR CASCADE CONSTRAINTS;
```

```
DROP TABLE ENTRAINEUR CASCADE CONSTRAINTS;
```

```
DROP TABLE COURSE CASCADE CONSTRAINTS;
```

```
DROP TABLE PARTICIPE CASCADE CONSTRAINTS;
```

```

-----
--          TABLE : COUREUR
-----

```

```

CREATE TABLE COUREUR
(
  NUMCOUREUR NUMBER(2) NOT NULL,
  NUMENTR NUMBER(2) NOT NULL,
  NOMCOUREUR VARCHAR2(128) NULL,

```

```
VILLECOUREUR VARCHAR2(128) NULL
, CONSTRAINT PK_COUREUR PRIMARY KEY (NUMCOUREUR)
) ;
```

```
-----
-- INDEX DE LA TABLE COUREUR
-----
```

```
CREATE INDEX I_FK_COUREUR_ENTRAINEUR
ON COUREUR (NUMENTR ASC)
;
```

```
-----
-- TABLE : ENTRAINEUR
-----
```

```
CREATE TABLE ENTRAINEUR
(
  NUMENTR NUMBER(2) NOT NULL,
  NOMENTR VARCHAR2(128) NULL,
  VILLEENTR VARCHAR2(128) NULL,
  SALAIREENTR NUMBER(5) NULL
, CONSTRAINT PK_ENTRAINEUR PRIMARY KEY (NUMENTR)
) ;
```

```
-----
-- TABLE : COURSE
-----
```

```
CREATE TABLE COURSE
(
  NUMCOURSE NUMBER(2) NOT NULL,
  LIBCOURSE VARCHAR2(128) NULL,
  VILLECOURSE VARCHAR2(128) NULL,
  DATECOURSE DATE NULL
, CONSTRAINT PK_COURSE PRIMARY KEY (NUMCOURSE)
) ;
```

```
-----
-- TABLE : PARTICIPE
-----
```

```
CREATE TABLE PARTICIPE
(
  NUMCOURSE NUMBER(2) NOT NULL,
  NUMCOUREUR NUMBER(2) NOT NULL,
  ORDRE NUMBER(2) NULL
, CONSTRAINT PK_PARTICIPE PRIMARY KEY (NUMCOURSE, NUMCOUREUR)
) ;
```

```
-----
--          INDEX DE LA TABLE PARTICIPE
-----
```

```
CREATE INDEX I_FK_PARTICIPE_COURSE
ON PARTICIPE (NUMCOURSE ASC)
;
```

```
CREATE INDEX I_FK_PARTICIPE_COUREUR
ON PARTICIPE (NUMCOUREUR ASC)
;
```

```
-----
--          CREATION DES REFERENCES DE TABLE
-----
```

```
ALTER TABLE COUREUR ADD (
  CONSTRAINT FK_COUREUR_ENTRAINEUR
    FOREIGN KEY (NUMENTR)
      REFERENCES ENTRAINEUR (NUMENTR)) ;
```

```
ALTER TABLE PARTICIPE ADD (
  CONSTRAINT FK_PARTICIPE_COURSE
    FOREIGN KEY (NUMCOURSE)
      REFERENCES COURSE (NUMCOURSE)) ;
```

```
ALTER TABLE PARTICIPE ADD (
  CONSTRAINT FK_PARTICIPE_COUREUR
    FOREIGN KEY (NUMCOUREUR)
      REFERENCES COUREUR (NUMCOUREUR)) ;
```

```
-----  
--                FIN DE GENERATION  
-----
```

Une fois le script de création des tables généré, on peut se connecter sur le serveur Oracle en utilisant nos identifiants propres.

La requête :

```
SELECT * FROM tabs;
```

Permettra de s'assurer que notre base de données propre est bien vide.

- Peuplement des tables

En se connectant sous le compte CAFE, et en exécutant un listing des différentes tables, on génère rapidement un fichier de script de peuplement des tables :

```
INSERT INTO ENTRAINEUR (NUMENTR, NOMENTR, VILLEENTR, SALAIREENTR)
VALUES
(1, 'WEBER', 'BELFORT', 8000);
INSERT INTO ENTRAINEUR (NUMENTR, NOMENTR, VILLEENTR, SALAIREENTR)
VALUES
(2, 'HARRY', 'SOCHAUX', 5000);
INSERT INTO ENTRAINEUR (NUMENTR, NOMENTR, VILLEENTR, SALAIREENTR)
VALUES
(3, 'FANNIER', 'BELFORT', 7000);
INSERT INTO ENTRAINEUR (NUMENTR, NOMENTR, VILLEENTR, SALAIREENTR)
VALUES
(4, 'DUBROCK', 'MONTBELIARD', 9000);
INSERT INTO ENTRAINEUR (NUMENTR, NOMENTR, VILLEENTR, SALAIREENTR)
VALUES
(5, 'JUSTIN', 'PARIS', 6000);

INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(1, 'ALBERT', 'BELFORT', 4);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
```

```
(2, 'DUPONT', 'MULHOUSE', 1);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(3, 'MARTIN', 'BELFORT', 1);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(4, 'DURANT', 'MULHOUSE', 2);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(5, 'BARREAU', 'MONTBELIARD', 3);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(6, 'PARRET', 'SEVENANS', 2);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(7, 'PARISSE', 'BELFORT', 1);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(8, 'PARADIS', 'MONTBELIARD', 4);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(9, 'MANILLE', 'PARIS', 3);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(10, 'DALO', 'BELFORT', 4);
INSERT INTO COUREUR (NUMCOUREUR, NOMCOUREUR, VILLECOUREUR, NUMENTR)
VALUES
(11, 'FILOU', 'MULHOUSE', 3);

INSERT INTO COURSE (NUMCOURSE, LIBCOURSE, VILLECOURSE, DATECOURSE)
VALUES
(15, 'Course du Lion', 'MONTBELIARD', '15/09/03');
INSERT INTO COURSE (NUMCOURSE, LIBCOURSE, VILLECOURSE, DATECOURSE)
VALUES
(16, 'Course du Lion', 'BELFORT', '12/09/04');

INSERT INTO COURSE (NUMCOURSE, LIBCOURSE, VILLECOURSE, DATECOURSE)
VALUES
(21, 'Course du Lion', 'MONTBELIARD', '12/09/05');
INSERT INTO COURSE (NUMCOURSE, LIBCOURSE, VILLECOURSE, DATECOURSE)
VALUES
(17, 'Marathon', 'BELFORT', '15/10/04');
```

```
INSERT INTO COURSE (NUMCOURSE, LIBCOURSE, VILLECOURSE, DATECOURSE)
VALUES
(18,'Course du tour','PARIS','17/01/05');
INSERT INTO COURSE (NUMCOURSE, LIBCOURSE, VILLECOURSE, DATECOURSE)
VALUES
(19,'Course du Territoire','BELFORT','25/01/05');
INSERT INTO COURSE (NUMCOURSE, LIBCOURSE, VILLECOURSE, DATECOURSE)
VALUES
(20,'L''alsacienne','MULHOUSE','30/07/05');
```

```
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(15,2,2);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(15,7,1);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(16,2,3);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(16,4,2);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(16,7,1);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(17,1,5);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(17,3,1);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(17,4,4);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(17,5,2);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(17,6,3);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
```

```
(17,9,6);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(17,10,6);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(18,1,4);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(18,2,3);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(18,3,2);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(18,7,1);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(18,8,5);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(18,9,7);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(18,10,6);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(19,2,3);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(19,3,4);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(19,6,2);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(19,4,8);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(19,7,1);

INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
```

```
VALUES
(19,9,5);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(20,6,1);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(20,8,2);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(21,9,2);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(21,10,1);
INSERT INTO PARTICIPE (NUMCOURSE, NUMCOUREUR, ORDRE)
VALUES
(21,3,3);
```

Notons que l'ordre de peuplement des tables est important, du fait de l'existence de contraintes sur des clés étrangères.

3 Mise à jour des données de la base de données

1. Déménagement du coureur à Bordeaux :

```
UPDATE
COUREUR
SET
VILLECOUREUR = 'BORDEAUX'
WHERE
NOMCOUREUR = 'MARTIN';
```

2.

```
INSERT INTO
ENTRAINEUR (NUMENTR, NONENTR, VILLEENTR, SALAIREENTR)
VALUES
(6, 'PACHINKO', 'BELFORT', 6500);
INSERT INTO
ENTRAINEUR (NUMENTR, NONENTR, VILLEENTR, SALAIREENTR)
```

```
VALUES
(7,'BOURGEAULT', 'BESANCON', 4500);
INSERT INTO
ENTRAINEUR (NUMENTR,NOMENTR, VILLEENTR, SALAIREENTR)
VALUES
(8,'CORBET', 'PARIS', 8500);
```

Note : ici, on a mis 'en dur' les identifiants d'entraîneurs, il aurait été intéressant de faire appel à un trigger et une sequence, comme expliqué ici :

<http://webxadmin.free.fr/article/oracle-autoincrement-columns-134.php>

3- Supprimer de la table entraîneur tous les entraîneurs qui habitent Sochaux. Quelle est la condition pour que cette suppression soit possible.

```
DELETE FROM
Entraîneur
WHERE
VilleEntr = 'SOCHAUX';
```

La condition pour que cela fonctionne est qu'aucun coureur ne soit suivi par les entraîneurs concernés par la suppression. Il faut soit supprimer les coureurs concernés par le suivi de l'entraîneur, soit leur affecter un nouvel entraîneur, le système obligeant les coureurs à être suivi par un entraîneur.

(UPDATE COUREUR SET NumEntr = 6 WHERE NUMCOUREUR IN (4,6); par exemple)

4- Donner à tous les entraîneurs de Belfort et de Montbéliard une augmentation de 5%.

```
UPDATE
Entraîneur
SET
SalaireEntr = SalaireEntr + SalaireEntr * 5 / 100;
```

5- Donner à l'entraîneur du vainqueur de la course « le lion 2005 » une augmentation de 20%.

```
UPDATE
```

```
Entreneur
SET
SalaireEntr = SalaireEntr + SalaireEntr * 20 / 100
WHERE
NumEntr = ( SELECT
Coureur.NumEntr
FROM
Coureur
JOIN
Participe
ON
Participe.NumCoureur = Coureur.NumCoureur
INNER JOIN
Course
ON
Course.NumCourse = Participe.NumCourse
WHERE
Participe.Ordre = 1
AND
Course.dateCourse LIKE '%/05'
AND
Course.LibCourse = 'Course du Lion');
```

6- Ajouter une prime de 200 F à l'entraîneur qui gagne le moins.

```
UPDATE
Entreneur
SET
SalaireEntr = SalaireEntr + 200
WHERE
NumEntr = ( SELECT
NumEntr
FROM
Entreneur
WHERE
SalaireEntr = (SELECT MIN(SalaireEntr) FROM Entreneur));
```

7- Donner une valeur « null » à tous les salaires des entraîneurs de Paris.

```
UPDATE
```

```
Entraîneur
SET
SalaireEntr = NULL
WHERE
VilleEntr = 'PARIS'
```

8- Supprimer de la table courses toutes les courses qui se sont déroulées en 2004.

Note : Il est nécessaire de vider avant la table participe de toutes les courses concernées

```
DELETE FROM
Participe
WHERE
Participe.NumCourse IN (SELECT NumCourse FROM Course WHERE DateCourse LIKE '%/04');
```

```
DELETE FROM
Course
WHERE
DateCourse LIKE '%/04';
```

9- Supprimer le coureur ALBERT. A quelle condition cela est-il possible ?

```
DELETE FROM
Coureur
WHERE
NomCoureur = 'ALBERT';
```

A condition qu'il n'ait pas participé à des courses :

```
DELETE FROM
Participe
WHERE
NumCoureur = (SELECT NumCoureur FROM Coureur WHERE NomCoureur = 'ALBERT');
```

10- Ajouter une colonne code postal (numérique, 5 positions) à la table entraîneur.

```
ALTER TABLE
    Entraîneur
```

```
ADD (CodePostal NUMBER(5));
```

11- Ajoutez une contrainte d'intégrité de domaine sur le salaire de l'entraîneur. (salaire positif et < 10000F)

```
ALTER TABLE Entraîneur
    ADD Constraint salary Check (SalaireEntr between 0 and 10000);
```

12- Supprimer de la table entraîneur la colonne salaire. Est-ce possible ?

```
ALTER TABLE Entraîneur
    DROP COLUMN SalaireEntr
```

4 Modification du schéma relationnel de la base de données

1- Créer un index sur la colonne ville de la table entraîneur.

```
CREATE INDEX
ville_idx
ON
Entraîneur( VilleEntr )
```

2- Créer une vue VBNC contenant le nombre de coureurs entraînés par chaque entraîneur (Nomentraîneur, nbcoureurs)

```
CREATE VIEW VBNC AS
SELECT
NomEntr
, COUNT(NumCoureur) AS NbCoureurs
FROM
Entraîneur
INNER JOIN
Coureur
ON
Coureur.NumEntr = Entraîneur.NumEntr
GROUP BY
NomEntr
```

3- Créer une vue VLION contenant tous les coureurs ayant participé à la course «le lion 2005 » (numcoureur, nomcoureur, ordre)

```
CREATE VIEW VLION AS
SELECT
Participe.NumCoureur
, Coureur.NomCoureur
, Participe.Ordre
FROM
Participe
JOIN
Coureur
ON
Coureur.NumCoureur = Participe.NumCoureur
WHERE
NumCourse = ( SELECT
NumCourse
FROM
Course
WHERE
Datecourse LIKE '%/05'
AND
LibCourse = 'Course du Lion' );
```

4- Ajouter un coureur dans la table coureur. Effet sur la vue VNBC

```
INSERT INTO Coureur
VALUES (12, 4, 'PAPET', 'LYON');
```

Incrémentation du nombre de coureurs pour le l'entraîneur num. 4 (dans ce cas ici)

5- A l'aide d'une requête, créer une table TCOURBEL contenant tous les coureurs de Belfort.

```
CREATE TABLE TCOURBEL (NomCoureur VARCHAR2(128));
INSERT INTO TCOURBEL VALUES
('PARISSE');
INSERT INTO TCOURBEL VALUES
('DALO');
INSERT INTO TCOURBEL VALUES
```

```
('MARTIN')
```

```
;
```

(Cela ne répond pas réellement à ce qui était demandé, mais je n'ai pas trouvé de requête simple permettant un remplissage automatique)

6- Créer une vue VCOURBEL contenant tous les coureurs de Belfort.

```
CREATE VIEW VCOURBEL AS
SELECT
NomCoureur
FROM
Coureur
WHERE
VilleCoureur = 'BELFORT';
```

7- Ajouter un coureur habitant Belfort dans la table coureur.
Effet sur la table TCOURBEL et sur la vue VCOURBEL ?

Directement mis à jour dans la vue, alors que cela nécessite une mise à jour via un ajout dans la table TCOURBEL.

8- Supprimez un coureur dans la vue VCOURBEL.
Effet sur la table COUREUR et sur la vue VCOURBEL ?

```
DELETE FROM
Participe
WHERE
NumCoureur = (SELECT NumCoureur FROM Coureur WHERE NomCoureur = 'MARTIN');
```

```
DELETE FROM
VCOURBEL
WHERE
NomCoureur = 'MARTIN';
```

Ceci provoque évidemment une suppression dans la vue, mais aussi dans la table Coureur.

9- Supprimer un coureur dans la table TCOURBEL.
Effet sur la table COUREUR et sur la vue VCOURBEL ?

Aucun.

10- Ajoutez un coureur dans la vue VCOURBEL. Effet sur la table COUREUR ?

Impossible, car les contraintes sur le système empêchent l'ajout de valeur nulles dans la table coureur.

11- Afficher le texte SQL définissant la vue VCOURBEL

```
DESCRIBE VCOURBEL;
```

12- Augmentez la taille de la colonne nomcoureur dans la table COUREUR.

```
ALTER TABLE  
Coureur  
MODIFY (NomCoureur VARCHAR2(256));
```

13- Ajoutez les colonnes Date de naissance et Email dans la table COUREUR. Effet sur la vue VCOURBEL ?
SI nécessaire modifier la vue pour afficher ces deux colonnes

```
ALTER TABLE  
Coureur  
ADD (DateNaissCoureur DATE);
```

```
ALTER TABLE  
Coureur  
ADD (MailCoureur VARCHAR2(512));
```

Aucun effet sur VCOURBEL (qui ne s'intéresse qu'à la ville).

```
DROP VIEW VCOURBEL;
```

```
CREATE VIEW VCOURBEL AS  
SELECT  
NomCoureur  
, DateNaissCoureur  
, MailCoureur  
FROM
```

```
Coureur
WHERE
VilleCoureur = 'BELFORT';
```

14- Modifiez l'email de tous les coureurs en affectant comme adresse de mél : nom@utbm.fr

```
UPDATE
Coureur
SET
MailCoureur = CONCAT(NomCoureur, '@utbm.fr');
```

15- Supprimer la colonne Email dans la table Coureur

```
ALTER TABLE
Coureur
DROP COLUMN
MailCoureur;
```

16- Supprimez la table TCOURBEL. Quelle est la conséquence de cette suppression ?

```
DROP TABLE TCOURBEL;
```

Aucune conséquence (Si ce n'est la disparition de la table TCOURBEL).

5 Mise à disposition des données et gestion de la sécurité

1- Créer un nouvel utilisateur avec le privilège CONNECT.

Nom utilisateur : votrenomC01
Mot de passe : votremotpasseC01

```
CREATE USER
mauditC01
IDENTIFIED BY
mauditC01;
GRANT
CONNECT
```

```
TO
mauditC01;
```

2- Donnez à l'utilisateur votrenomC01 le droit de consulter et d'insérer des lignes dans la table COUREUR.

```
GRANT
SELECT
        , INSERT
ON
COUREUR
TO
mauditC01;
```

3- Donnez à l'utilisateur votrenomC01 le droit de consulter et de mettre à jour la vue VCOURBEL.

```
GRANT
SELECT
        , UPDATE
ON
VCOURBEL
TO
mauditC01;
```

4- Déconnectez-vous, puis reconnectez-vous en tant que
Nom utilisateur : votrenomC01
Mot de passe : votremotpasseC01

5- Visualisez le contenu de la table COUREUR dans le compte votrenom.

```
(SELECT table_name FROM all_tables WHERE owner='maudit';)
SELECT * FROM MAUDUIT.COUREUR;
```

6- Insérez un nouveau coureur dans la table COUREUR du compte votrenom.

```
INSERT INTO
        MAUDUIT.COUREUR
VALUES (12, 4, 'BREWER', 'PARIS');
```

7- Le coureur FILOU déménage à BELFORT. Modifiez la table coureur du compte votrenom. Est-ce possible ? Pourquoi ?

```
UPDATE
Coureur
SET
VilleCoureur = 'BELFORT'
WHERE
NomCoureur = 'FILOU'
;
```

Les privilèges sont insuffisants.

8- A chaque fois que vous utilisez la table d'un autre propriétaire, vous devez spécifier le nom de son propriétaire. (expl : select * from votrenom.COUREUR). Par quelle requête l'administrateur de la base pourrait-il rendre accessible à tous les utilisateurs votre table coureur, sans qu'ils aient à spécifier le nom du propriétaire ?

En modifiant la propriété TABLESPACE des utilisateurs (et en donnant les droits suffisants);

9- Créer un synonyme privé dans votre compte pour chaque table du compte votrenom.

```
CREATE SYNONYM COURSE FOR MAUDUIT.COURSE;
```

10- Déconnectez-vous, puis reconnectez-vous en tant que
Nom utilisateur : votrenom
Mot de passe : votremotpasse

11- Supprimez les privilèges d'insertion pour l'utilisateur votrenomC01 sur votre table COUREUR.

```
REVOKE
INSERT
ON
COUREUR
FROM
mauditC01;
```

12- Supprimez tous les privilèges accordés sur la vue VCOURBEL

à l'utilisateur votrenomC01.

```
REVOKE
ALL
ON
VCOURBEL
FROM
mauditC01
;
```

13- Modifiez votre mot de passe de connexion. Reconnectez-vous et vérifiez la prise en compte du nouveau mot de passe.

```
ALTER USER
maudit
IDENTIFIED BY
newpw
;
```

14- Vérifiez la prise en compte de la modification des privilèges dans le compte votrenomC01.

La connexion avec le nouveau mot de passe fonctionne, on peut retourner à l'ancien mot de passe :

```
ALTER USER
maudit
IDENTIFIED BY
maudit
;
```