

## Interrogation d'une base de données Oracle 10G

### 1 Requêtes LMD

1. Liste du contenu de chaque table de la base

```
SELECT * FROM tab;
SELECT * FROM [NOM_TABLE];
(où NOM_TABLE est le nom de la table à afficher)
```

2. Liste des serveurs

```
SELECT
NomServeur
      , VilleServeur
FROM
Serveur
ORDER BY
NomServeur ASC;
```

3. Liste des villes des serveurs

```
SELECT
VilleServeur
FROM
Serveur
ORDER BY
Villeserveur ASC;
```

4. Liste des serveurs classée par ville

```
SELECT
NomServeur, VilleServeur
FROM
Serveur
ORDER BY
Villeserveur ASC;
```

5. Nombre de place de la table 4

```
SELECT
NbPlace
FROM
Tables
WHERE
NumTable = 4
```

6. Liste des consommations dont le prix unitaire est supérieur à 1.5 Euros

```
SELECT
NumCons, LibCons, PrixCons
FROM
Consommation
WHERE
PrixCons > 1.5
```

7. Liste des serveurs de Belfort et de Delle

```
SELECT
Numserveur, NomServeur, VilleServeur
FROM
Serveur
WHERE
VilleServeur IN('BELFORT', 'DELLE');
```

8. Liste des factures du 22.02.06

```
SELECT
*
FROM
Facture
WHERE
DateFacture = '22/02/06';
```

9. Liste des serveurs dont le nom contient un i en deuxième position.

```
SELECT
NomServeur
FROM
```

```
Serveur
WHERE
NomServeur LIKE '_i%';
```

10. Liste des serveurs dont le nom commence par un P.

```
SELECT
NomServeur
FROM
Serveur
WHERE
NomServeur LIKE 'P%';
```

11. Liste des consommations classées par ordre alphabétique sur le libellé

```
SELECT
LibCons, NumCons, PrixCons
FROM
Consommation
ORDER BY
LibCons ASC;
```

12. Liste des serveurs nés en 1976

```
SELECT
NomServeur, DateNServeur
FROM
Serveur
WHERE
DateNServeur LIKE '__/__/76'
ORDER BY
NomServeur
ASC;
```

13. Liste des consommations de type bière

```
SELECT
LibCons, NumCons, PrixCons
FROM
Consommation
WHERE
LibCons LIKE '%Bière%'
```

```
ORDER BY  
LibCons ASC;
```

14. Liste des factures du 22/02/06 servies par Martin Cathy  
14.1

```
SELECT  
Facture.NumFacture  
  , Facture.NumTable  
  , Tables.NomTable  
FROM  
Facture  
  , Serveur  
  , Tables  
WHERE  
Facture.NumServeur = Serveur.NumServeur  
AND  
Facture.numTable = Tables.NumTable  
AND  
Serveur.NomServeur = 'Martin Cathy'  
AND  
Facture.DateFacture = '22/02/06';
```

14.2  
(Sous requete)

```
SELECT  
Facture.NumTable  
  , Tables.NomTable  
FROM  
Facture  
  , Serveur  
  , Tables  
WHERE  
Facture.NumServeur = Serveur.NumServeur  
AND  
Facture.numTable = Tables.NumTable  
AND  
Serveur.NomServeur = 'Martin Cathy'  
AND  
Facture.DateFacture = '22/02/06';
```

14.3

```
SELECT
Facture.NumFacture
, Tables.NomTable
FROM
Facture
NATURAL JOIN
Serveur
NATURAL JOIN
Tables
WHERE
Serveur.NomServeur = 'Martin Cathy'
AND
Facture.DateFacture = '22/02/06';
```

14.4

```
SELECT
Facture.NumFacture
, Tables.NomTable
FROM
Facture
INNER JOIN
Serveur
USING ("NumServeur")
INNER JOIN
Tables
USING ("NumTable")
WHERE
Serveur.NomServeur = 'Martin Cathy'
AND
Facture.DateFacture = '22/02/06'
;
```

14.5

```
SELECT
Facture.NumFacture
, Tables.NomTable
FROM
Facture
```

```
JOIN
Serveur
ON
Facture.NumServeur = Serveur.Numserveur
JOIN
Tables
ON
Facture.NumTable = Tables.NumTable
WHERE
Serveur.NomServeur = 'Martin Cathy'
AND
Facture.DateFacture = '22/02/06';
```

15.

```
Liste des consommations de la facture 1203
(Numcons, Libcons, Prixcons, Qte, Prixcons*Qte, Nomserveur)
-- Utiliser des alias de tables
```

```
SELECT
Consommation.NumCons
, Consommation.LibCons
, Consommation.PrixCons
, Comprend.Qte
, Consommation.PRixCons * Comprend.Qte AS Total
, Serveur.NomServeur
FROM
Consommation
INNER JOIN
Comprend
ON      Comprend.NumCons = Consommation.NumCons
INNER JOIN
Facture
ON      Facture.NumFacture = Comprend.NumFacture
INNER JOIN
Serveur
On      Serveur.NumServeur = Facture.NumServeur
WHERE
Facture.NumFacture = 1203
;
```

16. Liste des consommations plus chères que l'Orangina  
(Numcons, Libcons)

-- Utiliser une sous-requête puis une inéquijointure et ajointure ANSI

```
SELECT
NumCons
, LibCons
FROM
Consommation
WHERE
PrixCons > (SELECT PRixCons FROM Consommation WHERE LibCons LIKE '%Orangina%');
```

17. Noms des serveurs qui n'ont pas encore servi (Numserveur, Nomserveur)

-- 3 solutions : Not Exits, Not In et Minus

```
SELECT
NumServeur
, NomServeur
FROM
Serveur
WHERE
NumServeur NOT IN (SELECT DISTINCT NUmServeur FROM Facture)
;
```

18. Liste des serveurs qui n'ont pas servi la table n. 1 (Numserveur, Nomserveur)

-- 3 solutions : Not Exits, Not In et Minus

```
SELECT
NumServeur
, NomServeur
FROM
Serveur
WHERE
NumServeur NOT IN (SELECT DISTINCT NUmServeur FROM Facture WHERE NUMTABLE = 1)
;
```

19. Liste des consommations des factures 1200 et 1201 ( sans lignes en double)  
(Numcons, Libcons)

-- jointure prédicative et Jointure Naturelle ANSI

```
SELECT DISTINCT Consommation.NumCons
, Consommation.LibCons
FROM
Consommation
```

```
INNER JOIN
Comprend
ON
Comprend.NumCons = Consommation.NumCons
INNER JOIN
Facture
ON
Facture.Numfacture = Comprend.NumFacture
WHERE
Facture.NumFacture IN (1200, 1201)
;
```

20. Liste des serveurs qui ont servi la table fenetre3 le 21/02/06  
(Numserveur, Nomserveur, Numfacture)

```
SELECT
Serveur.Numserveur
, Serveur.Nomserveur
, Facture.Numfacture
FROM
Serveur
INNER JOIN
Facture
ON
Facture.NumServeur = Serveur.NumServeur
INNER JOIN
Tables
ON
Tables.NumTable = Facture.NumTable
WHERE
Tables.NomTable = 'fenetre3'
AND
Facture.DateFacture = '21/02/06'
;
```

21. Liste des serveurs qui habitent dans la même ville que Durant Pierre  
(Numserveur, Nomserveur, Villeserveur)

```
SELECT
Numserveur
, NomServeur
, VilleServeur
```

```
FROM
Serveur
WHERE
VilleServeur = (SELECT VilleServeur FROM Serveur WHERE NomServeur = 'Durant Pierre')
;
```

22. Liste des tables servies après le 21/02/06  
(NomTable, Numserveur, Nomservreur, Numfacture)

```
SELECT DISTINCT
NomTable
, NumServeur
, NomServeur
, NumFacture
FROM
Tables
INNER JOIN
Facture
ON
Facture.NumTable = Tables.NumTable
INNER JOIN
Serveur
ON
Serveur.NumServeur = Facture.NumServeur
WHERE
DateFacture > '21/02/06'
ORDER BY
NomTable;
```

23. Liste des consommations qui n'ont encore jamais été commandées  
(Numcons, Libcons)

-- Not Exists

```
SELECT
NumCons
, LibCons
FROM
Consommation
WHERE
NumCons NOT IN (SELECT DISTINCT NumCons FROM Comprend)
;
```

24. Liste des consommations et pour celles qui ont été consommées, liste des factures dans lesquelles elles sont intervenues.

(Numcons, Libcons, Numfacture)

```
SELECT
Consommation.NumCons
, Consommation.LibCons
, Facture.NumFacture
FROM
Consommation
FULL OUTER JOIN
Comprend
ON
Comprend.NumCons = Consommation.NumCons
FULL OUTER JOIN
Facture
ON
Facture.NumFacture = Comprend.NumFacture
ORDER BY
Consommation.NumCons
;
```

25. Nombre de tables dans le café.

(nb tables)

```
SELECT
COUNT(NumTable) AS NB_TABLES
FROM
TABLES
;
```

26. Nombre de serveurs habitant Belfort. (Nbserveur)

```
SELECT
COUNT(NumServeur) AS NbServeur
FROM
Serveur
WHERE
VilleServeur = 'BELFORT'
;
```

27. Nombre de factures effectués par le serveur n.53 depuis le début de l'année

2006. (Nombrefacture)

```
SELECT
COUNT(NumFacture) AS NombreFacture
FROM
Facture
WHERE
NumServeur = 53
AND
DateFacture >= '01/01/06'
;
```

28. Nombre de clients servi par Martin depuis son embauche. (Nombrefacture)

```
SELECT
COUNT(NumFacture) AS NombreFacture
FROM
Facture
INNER JOIN
Serveur
ON
Serveur.NumServeur = Facture.NumServeur
WHERE
Serveur.NomServeur = 'Martin Cathy'
;
```

29. Nombre de factures établies dans les journées du 21/02/06 et 22/02/06.  
(Nbfactures)

```
SELECT
COUNT(NumFacture) AS NombreFacture
FROM
Facture
WHERE
DateFacture BETWEEN '21/02/06' AND '22/02/06'
;
```

30. Montant de la facture N. 1204.  
(Numfacture, Montant)

```
SELECT
SUM(Consommation.PrixCons * Comprend.QTE) AS Montant
```

```
FROM
Comprend
INNER JOIN
Consommation
ON
Consommation.NumCons = Comprend.NumCons
WHERE
Comprend.NumFacture = 1204
;

SELECT
Comprend.NumFacture
, SUM(Consommation.PrixCons * Comprend.QTE) AS Montant
FROM
Comprend
INNER JOIN
Consommation
ON
Consommation.NumCons = Comprend.NumCons
WHERE
Comprend.NumFacture = 1204
GROUP BY
Comprend.NumFacture
;
```

31. Chiffre d'affaires du mois de février 2006  
(Montanttotalfacture)

```
SELECT
SUM(Comprend.Qte * Consommation.PrixCons) AS MontantTotalFacture
FROM
Comprend
INNER JOIN
Consommation
ON
Consommation.NumCons = Comprend.NumCons
INNER JOIN
Facture
ON
Facture.NumFacture = Comprend.NumFacture
WHERE
Facture.DateFacture LIKE '__/02/06'
```

;

32. Nombre de consommations par facture.

(Numfacture, nbconso)

-- nbconso = somme des quantités

```
SELECT
Facture.NumFacture
, COUNT(Comprend.NumCons) AS NbConso
FROM
Facture
INNER JOIN
Comprend
ON
Comprend.NumFacture = Facture.NumFacture
GROUP BY
Facture.NumFacture
;
```

33. Consommation la moins chère.

(Numcons, Libcons, Prixcons)

```
SELECT
NumCons
, LibCons
, PrixCons
FROM
Consommation
WHERE
ROWNUM <= 1
ORDER BY
PRixCons
ASC
;
```

34. Consommation la plus chère servie par Pillot.

(Numcons, Libcons, Prixcons)

```
SELECT
Consommation.NumCons
, Consommation.LibCons
```

```
, Consommation.PrixCons
, ROWNUM AS numLigne
FROM
Consommation
INNER JOIN
Comprend
ON
Comprend.NumCons = Consommation.NumCons
INNER JOIN
Facture
ON
Facture.NumFacture = Comprend.NumFacture
INNER JOIN
Serveur
ON
Serveur.NumServeur = Facture.NumServeur
WHERE
Consommation.PrixCons = MAX
(SELECT
Consommation.PrixCons
FROM
Consommation
INNER JOIN
Comprend
ON
Comprend.NumCons = Consommation.NumCons
INNER JOIN
Facture
ON
Facture.NumFacture = Comprend.NumFacture
INNER JOIN
Serveur
ON
Serveur.NumServeur = Facture.NumServeur
WHERE
Serveur.NomServeur LIKE 'Pilotot %')
;
```

Solution : Récupérer la consommation la plus chère servie par le serveur, et "égaliser" avec un

35. Nombre de factures établies chaque jour.  
(Datefacture, Nbfactures)

```
SELECT
DateFacture
, COUNT(NumFacture) AS NbFacture
FROM
Facture
GROUP BY
DateFacture
ORDER BY
DateFacture
;
```

36. Nombre de factures établies par chaque serveur.  
(Numserveur, Nbfacture)  
-- Tous les serveurs doivent être dans le résultat

```
SELECT
Serveur.NumServeur
, COUNT(Facture.NumFacture) As Nbfacture
FROM
Serveur
LEFT JOIN
Facture
ON
Facture.NumServeur = Serveur.NumServeur
GROUP BY
Serveur.NumServeur
;
```

37. Nombre de factures établies par chaque serveur.  
(Numserveur, Nomserveur, Nbfacture)

```
SELECT
Serveur.NumServeur
, Serveur.Nomserveur
, COUNT(Facture.NumFacture) AS Nbfacture
FROM
Serveur
LEFT JOIN
Facture
ON
```

```
Facture.NumServeur = Serveur.NumServeur
GROUP BY
Serveur.NumServeur
, Serveur.NomServeur
;
```

38. Liste des serveurs qui ont établi plus de 3 factures.  
(Numserveur, Nomserveur, Nbfacture)

```
SELECT
Serveur.NumServeur
, Serveur.Nomserveur
, COUNT(Facture.NumFacture) AS Nbfacture
FROM
Serveur
LEFT JOIN
Facture
ON
Facture.NumServeur = Serveur.NumServeur
GROUP BY
Serveur.NumServeur
, Serveur.NomServeur
HAVING COUNT(Facture.NumFacture) >= 3
;
```

39. Chiffre d'affaire de chaque serveur  
(Numserveur, montanttotalfacture)

```
SELECT
Serveur.NumServeur
,SUM(Comprend.Qte * Consommation.PrixCons) AS montanttotalfacture
FROM
Serveur
INNER JOIN
Facture
ON
Facture.NumServeur = Serveur.NumServeur
INNER JOIN
Comprend
ON
Comprend.NumFacture = Facture.NumFacture
```

```
INNER JOIN
Consommation
ON
Consommation.NumCons = Comprend.NumCons
GROUP BY
Serveur.NumServeur
;
```

40. Prix moyen des consommations.  
(prixmoyen)

```
SELECT AVG(PrixCons)
FROM
Consommation
;
```

41. Prix moyen du café.  
(prixmoyen)

```
SELECT
AVG(PrixCons) AS PrixMoyen
FROM
Consommation
WHERE
LibCONS LIKE '%Café%'
;
```

42. Quantité moyenne consommée pour chaque consommation.  
(Numcons, Qtémoyenne)

```
SELECT
NumCons
, AVG(Qte) As QteMoyenne
FROM
Comprend
GROUP BY
NumCons
;
```

43. Quantité moyenne consommée pour chaque consommation.  
(Numcons, Libcons, Qtémoyenne)

```
SELECT
Consommation.NumCons
, AVG(Comprend.Qte) As QteMoyenne
, Consommation.LibCons
FROM
Comprend
INNER JOIN
Consommation
ON
Consommation.NumCons = Comprend.NumCons
GROUP BY
Consommation.NumCons
, Consommation.LibCons
;
```

44. Nombre de serveurs par ville  
(Villeserveur, nbserveur)

```
SELECT
COUNT(NumServeur) AS Nbserveur
, VilleServeur
FROM
Serveur
GROUP BY
VilleServeur
;
```

45. Factures établies avant le 22/02/06 et qui comportent plus de 3 consommations différentes.  
(Numfacture, DateFacture, Nbconso)

```
SELECT
Facture.Numfacture
, Facture.Datefacture
, COUNT(Comprend.NumCons) AS NbConso
FROM
Facture
INNER JOIN
Comprend
ON
Comprend.NumFacture = Facture.Numfacture
```

```
GROUP BY
Facture.NumFacture
, Facture.DateFacture
HAVING COUNT(Comprend.NumCons) > 3
;
```

46. Liste des villes dans lesquelles habitent plus d'un serveur.  
(villeserveur, nbserveur)

```
SELECT
Villeserveur
, COUNT(NumServeur) AS Nbserveur
FROM
Serveur
GROUP BY
VilleServeur
HAVING
Count(NumServeur) > 1
;
```

47. Consommations qui interviennent dans plus de deux factures.  
(Numcons, Libcons, Nbfactures)

```
SELECT
Consommation.NumCons
, Consommation.LibCons
, COUNT(Comprend.NumFacture) AS Nbfactures
FROM
Consommation
INNER JOIN
Comprend
ON
Comprend.NumCons = Consommation.NumCons
GROUP BY
Consommation.NumCons
, Consommation.LibCons
HAVING
COUNT(Comprend.NumFacture) > 2
;
```

48. Tables pour lesquelles le chiffre d'affaires réalisé est le plus élevé.

(NumTable, montanttotalfacture)

```
SELECT
Tables.NumTable
, SUM(Comprend.Qte * Consommation.PrixCons) AS montanttotalfacture
FROM
Tables
INNER JOIN
Facture
ON
Facture.NumTable = Tables.NumTable
INNER JOIN
Comprend
ON
Comprend.NumFacture = Facture.NumFacture
INNER JOIN
Consommation
ON
Consommation.NumCons = Comprend.NumCons
GROUP BY
Tables.NumTable
ORDER BY
SUM(Comprend.Qte * Consommation.PrixCons) DESC
```

(Le résultat attendu semble être le bon (La "meilleure" table est la 5), mais je n'ai pas réussi à faire les "sous" requêtes (R1 et R2) comme demandé dans le poly)

49. Calculer le chiffre d'affaires par Année et Mois  
(Année, Mois, MontantTotalFactures)

50. Calculer le chiffre d'affaires par Année et Mois (inclure le total par année et général) (Année, Mois, MontantTotalFactures)

51. Calculer le chiffre d'affaires par Ville et Année (inclure le total par ville, année et général) (Villeserveur, Année, MontantTotalFactures)

```
SELECT
EXTRACT(YEAR FROM Facture.DateFacture) AS Annee
, EXTRACT(MONTH FROM Facture.DateFacture) AS Mois
, SUM(Comprend.NumCons * Consommation.PrixCons) AS montanttotalfacture
FROM
Facture
```

```
INNER JOIN
Comprend
ON
comprend.NumFacture = Facture.NumFacture
INNER JOIN
Consommation
ON
Consommation.NumCons = Comprend.NumCons
GROUP BY
ROLLUP(EXTRACT(YEAR FROM Facture.DateFacture)
, EXTRACT(MONTH FROM Facture.DateFacture))
;
```

Note : le EXTRACT n'est pas censé servir à ça, mais plus pour gérer du XML, selon M. Fischer.

52. Liste des consommations qui apparaissent dans toutes les factures du mois de février 2006.

(Numcons, Libcons)

```
SELECT NUMCONS,LIBCONS
FROM CONSOMMATION C1
WHERE NOT EXISTS
(
  SELECT NUMFACTURE
  FROM FACTURE
  WHERE NOT EXISTS
  (
    SELECT NUMFACTURE
    FROM COMPREND
    WHERE COMPREND.NUMFACTURE=FACTURE.NUMFACTURE
    AND COMPREND.NUMCONS=C1.NUMCONS
  )
  AND DATEFACTURE LIKE '%/02/06'
);
```

53. A l'aide de l'opérateur REGEXP\_LIKE afficher tous les serveurs dont l'adresse EMAIL est conforme au modèle p.nom @ .fr

```
SELECT
Serveur.Email
FROM
```

```

Serveur
WHERE
REGEXP_LIKE (Serveur.Email, 'p+.[\w][\w].fr');

```

54. Affichez les adresses contenant un L ou R en lettre majuscule

```

SELECT
Serveur.Email
FROM
Serveur
WHERE
REGEXP_LIKE (Serveur.Email, '*[L,R]*');

```

55. Extraire le prénom du serveur.

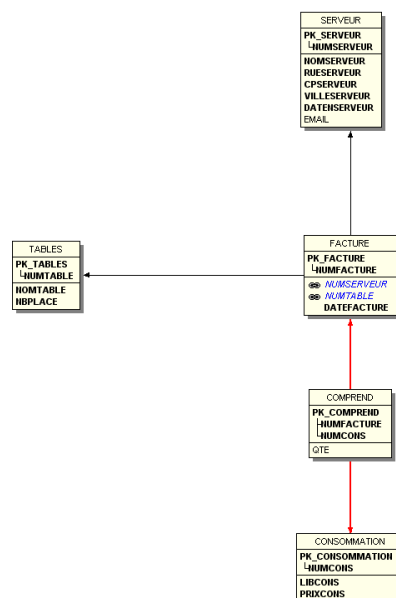
```

SELECT
REGEXP_REPLACE(SERVEUR.NomServeur, '[A-Za-z]+[ ]', '') AS PrenomServeur
FROM
Serveur;

```

## 2 Génération du script SQL du modèle physique

- MLR



---

- Script SQL du modèle physique pour base de données Oracle

```
-- -----  
--           Génération d'une base de données pour  
--           Oracle Version 9i  
--           (28/3/2007 16:18:00)  
-- -----  
--     Nom de la base : MLR1  
--     Projet : Espace de travail  
--     Auteur : Génie Informatique  
--     Date de dernière modification : 28/3/2007 16:17:18  
-- -----  
  
DROP TABLE FACTURE CASCADE CONSTRAINTS;  
  
DROP TABLE TABLES CASCADE CONSTRAINTS;  
  
DROP TABLE SERVEUR CASCADE CONSTRAINTS;  
  
DROP TABLE CONSOMMATION CASCADE CONSTRAINTS;  
  
DROP TABLE COMPREND CASCADE CONSTRAINTS;  
  
-- -----  
--           TABLE : FACTURE  
-- -----  
  
CREATE TABLE FACTURE  
  (  
    NUMFACTURE NUMBER(4) NOT NULL,  
    NUMSERVEUR NUMBER(2) NOT NULL,  
    NUMTABLE NUMBER(2) NOT NULL,  
    DATEFACTURE DATE NOT NULL  
  ,   CONSTRAINT PK_FACTURE PRIMARY KEY (NUMFACTURE)  
  ) ;  
  
-- -----  
--           INDEX DE LA TABLE FACTURE  
-- -----  
  
CREATE INDEX I_FK_FACTURE_SERVEUR
```

```
    ON FACTURE (NUMSERVEUR ASC)
;
```

```
CREATE INDEX I_FK_FACTURE_TABLES
    ON FACTURE (NUMTABLE ASC)
;
```

```
-----
--          TABLE : TABLES
-----
```

```
CREATE TABLE TABLES
(
    NUMTABLE NUMBER(2) NOT NULL,
    NOMTABLE VARCHAR2(15) NOT NULL,
    NBPLACE NUMBER(2) NOT NULL
,   CONSTRAINT PK_TABLES PRIMARY KEY (NUMTABLE)
) ;
```

```
-----
--          TABLE : SERVEUR
-----
```

```
CREATE TABLE SERVEUR
(
    NUMSERVEUR NUMBER(2) NOT NULL,
    NOMSERVEUR VARCHAR2(20) NOT NULL,
    RUESERVEUR VARCHAR2(20) NOT NULL,
    CPSERVEUR NUMBER(5) NOT NULL,
    VILLESERVEUR VARCHAR2(20) NOT NULL,
    DATENSERVEUR DATE NOT NULL,
    EMAIL VARCHAR2(30) NULL
,   CONSTRAINT PK_SERVEUR PRIMARY KEY (NUMSERVEUR)
) ;
```

```
-----
--          TABLE : CONSOMMATION
-----
```

```
CREATE TABLE CONSOMMATION
(
    NUMCONS NUMBER(3) NOT NULL,
```

```
LIBCONS VARCHAR2(20) NOT NULL,  
PRIXCONS NUMBER(8,2) NOT NULL  
, CONSTRAINT PK_CONSOMMATION PRIMARY KEY (NUMCONS)  
);
```

```
-----  
--          TABLE : COMPREND  
-----
```

```
CREATE TABLE COMPREND  
(  
    NUMFACTURE NUMBER(4) NOT NULL,  
    NUMCONS NUMBER(3) NOT NULL,  
    QTE NUMBER(2) NULL  
, CONSTRAINT PK_COMPREND PRIMARY KEY (NUMFACTURE, NUMCONS)  
);
```

```
-----  
--          INDEX DE LA TABLE COMPREND  
-----
```

```
CREATE INDEX I_FK_COMPREND_FACTURE  
ON COMPREND (NUMFACTURE ASC)  
;
```

```
CREATE INDEX I_FK_COMPREND_CONSOMMATION  
ON COMPREND (NUMCONS ASC)  
;
```

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

```
ALTER TABLE FACTURE ADD (  
    CONSTRAINT FK_FACTURE_SERVEUR  
        FOREIGN KEY (NUMSERVEUR)  
            REFERENCES SERVEUR (NUMSERVEUR)) ;
```

```
ALTER TABLE FACTURE ADD (  
    CONSTRAINT FK_FACTURE_TABLES
```

```
FOREIGN KEY (NUMTABLE)
REFERENCES TABLES (NUMTABLE)) ;
```

```
ALTER TABLE COMPREND ADD (
CONSTRAINT FK_COMPREND_FACTURE
FOREIGN KEY (NUMFACTURE)
REFERENCES FACTURE (NUMFACTURE)) ;
```

```
ALTER TABLE COMPREND ADD (
CONSTRAINT FK_COMPREND_CONSOMMATION
FOREIGN KEY (NUMCONS)
REFERENCES CONSOMMATION (NUMCONS)) ;
```

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

#### - Script SQL pour serveur MySQL

```
CREATE DATABASE IF NOT EXISTS CAFE;
```

```
USE CAFE;
```

```
DROP TABLE IF EXISTS FACTURE;
```

```
DROP TABLE IF EXISTS TABLES;
```

```
DROP TABLE IF EXISTS SERVEUR;
```

```
DROP TABLE IF EXISTS CONSOMMATION;
```

```
DROP TABLE IF EXISTS COMPREND;
```

```
# -----
# TABLE : FACTURE
# -----
```

```
CREATE TABLE IF NOT EXISTS FACTURE
(
    NUMFACTURE NUMBER(4) NOT NULL ,
    NUMSERVEUR NUMBER(2) NOT NULL ,
    NUMTABLE NUMBER(2) NOT NULL ,
    DATEFACTURE DATE NOT NULL
    , PRIMARY KEY (NUMFACTURE)
)
comment = "";
```

```
# -----
#      INDEX DE LA TABLE FACTURE
# -----
```

```
CREATE INDEX I_FK_FACTURE_SERVEUR
ON FACTURE (NUMSERVEUR ASC);
```

```
CREATE INDEX I_FK_FACTURE_TABLES
ON FACTURE (NUMTABLE ASC);
```

```
# -----
#      TABLE : TABLES
# -----
```

```
CREATE TABLE IF NOT EXISTS TABLES
(
    NUMTABLE NUMBER(2) NOT NULL ,
    NOMTABLE VARCHAR2(15) NOT NULL ,
    NBPLACE NUMBER(2) NOT NULL
    , PRIMARY KEY (NUMTABLE)
)
comment = "";
```

```
# -----
#      TABLE : SERVEUR
# -----
```

```
CREATE TABLE IF NOT EXISTS SERVEUR
(
    NUMSERVEUR NUMBER(2) NOT NULL ,
```

```
NOMSERVEUR VARCHAR2(20) NOT NULL ,
RUESERVEUR VARCHAR2(20) NOT NULL ,
CPSERVEUR NUMBER(5) NOT NULL ,
VILLESERVEUR VARCHAR2(20) NOT NULL ,
DATENSERVEUR DATE NOT NULL ,
EMAIL VARCHAR2(30) NULL
, PRIMARY KEY (NUMSERVEUR)
)
comment = "";

# -----
#      TABLE : CONSOMMATION
# -----

CREATE TABLE IF NOT EXISTS CONSOMMATION
(
  NUMCONS NUMBER(3) NOT NULL ,
  LIBCONS VARCHAR2(20) NOT NULL ,
  PRIXCONS NUMBER(8,2) NOT NULL
  , PRIMARY KEY (NUMCONS)
)
comment = "";

# -----
#      TABLE : COMPREND
# -----

CREATE TABLE IF NOT EXISTS COMPREND
(
  NUMFACTURE NUMBER(4) NOT NULL ,
  NUMCONS NUMBER(3) NOT NULL ,
  QTE NUMBER(2) NULL
  , PRIMARY KEY (NUMFACTURE,NUMCONS)
)
comment = "";

# -----
#      INDEX DE LA TABLE COMPREND
# -----

CREATE INDEX I_FK_COMPREND_FACTURE
```

```
ON COMPREND (NUMFACTURE ASC);
```

```
CREATE INDEX I_FK_COMPREND_CONSOMMATION  
ON COMPREND (NUMCONS ASC);
```

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

```
ALTER TABLE FACTURE  
ADD FOREIGN KEY FK_FACTURE_SERVEUR (NUMSERVEUR)  
REFERENCES SERVEUR (NUMSERVEUR) ;
```

```
ALTER TABLE FACTURE  
ADD FOREIGN KEY FK_FACTURE_TABLES (NUMTABLE)  
REFERENCES TABLES (NUMTABLE) ;
```

```
ALTER TABLE COMPREND  
ADD FOREIGN KEY FK_COMPREND_FACTURE (NUMFACTURE)  
REFERENCES FACTURE (NUMFACTURE) ;
```

```
ALTER TABLE COMPREND  
ADD FOREIGN KEY FK_COMPREND_CONSOMMATION (NUMCONS)  
REFERENCES CONSOMMATION (NUMCONS) ;
```

### - Remarques

On peut remarquer que les commentaires ne se font pas de la même manière sous les deux environnements, que le nom des types de données varie, et que la déclaration des clés étrangères n'est pas tout à fait la même.