

Rapport des travaux pratiques de BDD

TP1 Création d'une base de données sous ORACLE :

B- Première création de table

1.

```
>CREATE TABLE etudiants(  
    NUMERO NUMBER(4) PRIMARY KEY,  
    NOM VARCHAR2(25) NOT NULL,  
    PRENOM VARCHAR2(25) NOT NULL,  
    SEXE CHAR(1) CHECK (SEX IN ('F', 'M')),  
    ATENAISANCE DATE NOT NULL,  
    POIDS NUMBER,  
    ANNEE NUMBER);
```

2.

```
> DESC etudiants;
```

<i>Nom</i>	<i>NULL ? Type</i>
NUMERO	NOT NULL NUMBER(4)
NOM	NOT NULL VARCHAR2(25)
PRENOM	NOT NULL VARCHAR2(25)
SEX	CHAR(1)
DATENAISANCE	NOT NULL DATE
POIDS	NUMBER
ANNEE	NUMBER

3. la clé primaire est numero

4. les contraintes definit sont : nom, prenom, datenaissance sont non nulle
sexe est un choix entre F et M

5.

```
>INSERT into etudiants values (71,'Traifor', 'Benoît', 'M', '10/12/1978', '77', '1');  
>INSERT into etudiants values (72,'Génial', 'Clément', 'M', '10/04/1978', '72', '1');  
>INSERT into etudiants values (73,'Paris', 'Adam', 'M', '28/06/1974', '72', '2');  
>INSERT into etudiants values (74,'Paris', 'Clémence', 'F', '20/09/1977', '72', NULL);  
>INSERT into etudiants values (69,'Saitout', 'Inès', 'F', '22/11/1969', '69', 2);
```

```
>INSERT into etudiants values (55,'Serafoub', 'Izouaf', 'M', '19/09/2013', '1', 0);
```

```
-- INSERT into etudiants values (55,NULL, 'Izouaf', 'M', '19/09/2013', '1', 0);
```

6.

```
-- ERREUR ♦ la ligne 1 :
```

```
-- ORA-01400: impossible d'ins♦rer NULL dans ("L3INFO_77"."ETUDIANTS"."NOM")
```

C- Base de données avec plusieurs tables

```
>CREATE TABLE ELEVES
```

```
(  
    NUM_ELEVE NUMBER(4) PRIMARY KEY,  
    NOM VARCHAR2(25) NOT NULL,  
    PRENOM VARCHAR2(25) NOT NULL,  
    DATE_NAISSANCE DATE NOT NULL,  
    POIDS NUMBER,  
    ANNEE NUMBER  
);
```

```
>CREATE TABLE PROFESSEURS
```

```
(  
    NUM_PROF NUMBER(4) PRIMARY KEY,  
    NOM VARCHAR2(25) NOT NULL,  
    SPECIALITE VARCHAR2(25) NOT NULL,  
    DATE_ENTREE DATE NOT NULL,  
    DER_PROM VARCHAR2(25),  
    SALAIRE_BASE NUMBER,  
    SALAIRE_ACTUEL NUMBER  
);
```

```
>CREATE TABLE ACTIVITES
```

```
(  
    NOM VARCHAR2(25) NOT NULL ,  
    NIVEAU NUMBER NOT NULL,  
    EQUIPE NUMBER,  
    PRIMARY KEY(NOM, NIVEAU)  
);
```

```
>CREATE TABLE COURS(
```

```
    NUM_COURS NUMBER(4) PRIMARY KEY,  
    NOM VARCHAR2(25) NOT NULL,  
    NBHEURES NUMBER NOT NULL,  
    ANNEE NUMBER(4) NOT NULL  
);
```

```

>CREATE TABLE RESULTATS
(
    NUM_ELEVE NUMBER(4) NOT NULL,
    NUM_COURS NUMBER(4) NOT NULL,
    POINTS NUMBER NOT NULL,
    PRIMARY KEY(NUM_ELEVE, NUM_COURS),
    CONSTRAINT fk_num_resultats_eleves FOREIGN KEY(NUM_ELEVE) REFERENCES
ELEVES(NUM_ELEVE),
    CONSTRAINT fk_num_resultats_cours FOREIGN KEY(NUM_COURS) REFERENCES
COURS(NUM_COURS)
);

```

```

>CREATE TABLE CHARGE
(
    NUM_PROF NUMBER(4) NOT NULL,
    NUM_COURS NUMBER(4) NOT NULL,
    PRIMARY KEY(NUM_PROF, NUM_COURS),
    CONSTRAINT fk_num_charge_professeurs FOREIGN KEY(NUM_PROF) REFERENCES
PROFESSEURS(NUM_PROF),
    CONSTRAINT fk_num_charge_cours FOREIGN KEY(NUM_COURS) REFERENCES
COURS(NUM_COURS)
);

```

```

>CREATE TABLE ACTIVITES_PRATIQUEES(
    NUM_ELEVE NUMBER(4) NOT NULL,
    NIVEAU NUMBER NOT NULL,
    NOM VARCHAR2(25) NOT NULL,
    PRIMARY KEY(NUM_ELEVE, NIVEAU, NOM),
    CONSTRAINT fk_num_activitesP_eleves FOREIGN KEY(NUM_ELEVE) REFERENCES
ELEVES(NUM_ELEVE),
    CONSTRAINT fk_num_activitesP_activites FOREIGN KEY(NOM,NIVEAU)
REFERENCES ACTIVITES(NOM,NIVEAU)
);

```

```

>DROP TABLE ELEVES CASCADE CONSTRAINT;
>DROP TABLE COURS CASCADE CONSTRAINT;
>DROP TABLE PROFESSEURS CASCADE CONSTRAINT;
>DROP TABLE ACTIVITES CASCADE CONSTRAINT;
>DROP TABLE RESULTATS CASCADE CONSTRAINT;
>DROP TABLE CHARGE CASCADE CONSTRAINT;
>DROP TABLE ACTIVITES_PRATIQUEES CASCADE CONSTRAINT;

```

TP2 Modification d'une base de données sous ORACLE :

A- Modification de contraintes

1.

>CREATE TABLE ETUDIANTS (

```
NUMERO NUMBER(4) CONSTRAINT NN_ETUDIANTS_NUMBER NOT NULL,  
NOM VARCHAR2(25) CONSTRAINT NN_ETUDIANTS_NOM NOT NULL,  
PRENOM VARCHAR2(25) CONSTRAINT NN_ETUDIANTS_PRENOM NOT NULL,  
SEXE CHAR(1) CONSTRAINT CK_ETUDIANTS_SEXE CHECK (SEXE IN ('F','M')),  
DATENAISSANCE DATE CONSTRAINT NN_ETUDIANTS_DATENAISSANCE NOT NULL,  
POIDS NUMBER,  
ANNEE NUMBER,  
CONSTRAINT PK_ETUDIANTS PRIMARY KEY(NUMERO));
```

2.

ttitle 'La liste des contraintes crees, sur la table ETUDIANTS, est :';

>select constraint_name from user_constraints where table_name='ETUDIANTS';

La liste des contraintes crees, sur la table ETUDIANTS, est :

CONSTRAINT_NAME

NN_ETUDIANTS_NUMBER
NN_ETUDIANTS_NOM
NN_ETUDIANTS_PRENOM
NN_ETUDIANTS_DATENAISSANCE
CK_ETUDIANTS_SEXE
PK_ETUDIANTS

3.

>ALTER TABLE ETUDIANTS ADD CONSTRAINT CK_ETUDIANTS_ANNEE CHECK
(ANNEE IN (1,2));

>DROP TABLE ETUDIANTS;

TP3 Fonctions Oracle

A) Exploration de quelques fonctions ORACLE

1.

```
>select rpad('Soleil',17,'bla') "rpad exemple" from dual;  
--rpad('chaine1',nombre,'chaine2') :  
--remplis à droite de chaine 1 avec chaine2 jusqu'à obtenir le nombre de caractère au total
```

RPAD exemple

Soleilblablabl

```
>select lpad('Master 2 EID',15,'*.*') "lpad exemple" from dual;  
--lpad('chaine1',nombre,'chaine2') :  
--remplis à gauche de chaine 1 avec chaine2 jusqu'à obtenir le nombre de caractère au total
```

LPAD exemple

**.*Master 2 EID*

```
>select substr('DESS EID',6,3) "substr exemple" from dual;  
--substr('chaîne',nb,nb2)  
--A partir de nb éme caractère de la chaine on écrit nb2 caractère
```

SUB

EID

```
>select substr('abcdefghij',-5,4)"substr exemple"from dual;
--substr('chaîne',nb,nb2)
--C'est la même fonction que la précédente mais ici nb est négatif cela permet de lire la
chaîne de droite à gauche
```

SUBS

FGHI

```
>select to_char(sysdate,'mm-dd-yyyy hh24:mi:ss')"now" from dual;
--affiche la date actuelle du système to_char transforme sysdate en caractère et la deuxième
variable correspond au format voulu
```

Now

12-22-2019 20:42:40

```
>select length('web warehouse')"longueur en caractere" from dual;
--affiche la longueur de la chaîne en comptant le nombre de caractère
```

Longueur en caractères

13

```
>select round(17.0958,1)"round exemple" from dual;
--round(nb,nb2)
--round arrondit le nb avec nb2 chiffre après la virgule
```

ROUND exemple

17,1

```
>select round(17.58,2)"round exemple" from dual;
--même chose que le >select précédent
```

ROUND exemple

17,58

```
>select trunc(1958.0917,1)"trunc exemple" from dual;
--trunc(nb,nb2)
--coupe nb sans arrondir juste devant le nb2 ème chiffre après la virgule
```

TRUNC exemple

1958

```
>select trunc(1958.0917,2)"trunc exemple" from dual;  
--comme au dessus
```

TRUNC exemple

1958,09

```
>select round(TO_DATE('17-SEP-2009'),'YEAR')"New Year" from dual;  
--ERREUR caractère non numérique trouvé à la place d'un caractère numérique
```

```
>select round(TO_DATE('17-09-2009'),'YEAR')"New Year" from dual;
```

```
>select sysdate from dual;  
--affiche la date actuelle
```

SYSDATE

22/12/19

```
>select extract(year from sysdate) from dual;  
--récupère year de sysdate si la variable existe alors elle l'affiche
```

EXTRACT(YEARFROMSYSDATE)

2019

```
>select add_months(sysdate,7) from dual;  
--affiche la date avec 7 mois en plus
```

ADD_MONT

22/07/20

```
>select TRUNC(MONTHS_BETWEEN(SYSDATE,TO_DATE('19-JUN-2001'))))AS AGE  
FROM DUAL;  
--ERREUR caractère non numérique trouvé à la place d'un caractère numérique
```

```
>select TRUNC(MONTHS_BETWEEN(SYSDATE,TO_DATE('19-06-2001'))))AS AGEBB  
FROM DUAL;
```

--affiche le nombre de mois qui sépare la date actuelle à celle qu'on a entrer

```
>select to_number(to_char(sysdate,'YYYY')) from dual;
```

--to_number transforme une chaîne de caractère en nombre

```
TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))
```

2019

2.

```
ALTER session set NLS_DATE_FORMAT = 'DD-MM-YYYY';
```

--modifie le format de la date dans cette session

B)Exemple sur une vraie table

1.

```
drop table etudiants cascade constraints;
```

```
>create table etudiants(  
  numero number(4) NOT NULL,  
  nom varchar(25) NOT NULL,  
  prenom varchar(25) NOT NULL,  
  sexe char(1) check(sexe in('F','M')),  
  datenaissance date not null,  
  poids number,  
  annee number,  
  constraint pk_etudiants primary key(numero));
```

2.

```
>insert into etudiants values(71,'traifor','benoit','M','10/12/1978',77,1);  
>insert into etudiants values(72,'genial','clement','M','10/04/1978',72,1);  
>insert into etudiants values(73,'paris','adam','M','28/06/1974',72,2);  
>insert into etudiants values(74,'parees','clemence','F','20/09/1977',72,null);  
>insert into etudiants values(69,'saitout','ines','F','22/11/1969',69,2);  
>insert into etudiants values(55,'serafoub','izouaf','M','19/09/2013',81,1);
```

3.

```
>select decode(annee,1,'Premiere',2,'Seconde','Valeur differente de 1 et 2 !!!') as anetude  
from etudiants;
```

--affiche la colonne année en remplaçant 1 par premiere, 2 seconde et ainsi de suite selon ce qu'on a entrer

ANETUDE

Premiere

Premiere

Seconde

Valeur differente de 1 et 2 !!

Seconde

Premiere

>select upper(nom) from etudiants;

--affiche tous les noms de la base de données en majuscule

UPPER(NOM)

TRAIFOR

GENIAL

PARIS

PAREES

SAITOUT

SERAFOUB

>select lower(nom) from etudiants;

--affiche les noms en minuscule

LOWER(NOM)

traifor

genial

paris

parees

saitout

serafoub

>select NVL(annee,'Valeur non renseignée') from etudiants;

>select NVL(annee,'Valeur non renseignée') as an_etude from etudiants;

--ERREUR Nombre non valide

>select NVL(annee,1) from etudiants;

--Le deuxième argument est doit être un nombre nvl permet d'afficher le premier argument si celui ci est vide alors il affiche le second

>select NVL(annee,2) as an_etude from etudiants;

NVL(ANNEE,1)

1
1
2
1
2
1

4.

>select * from etudiants;

NUMERO NOM	PRENOM	S DATENAIS
71 traifor	benoit	M 10/12/78
72 genial	clement	M 10/04/78
73 paris	adam	M 28/06/74
74 parees	clemence	F 20/09/77
69 saitout	ines	F 22/11/69
55 seraifoub	izouaf	M 19/09/13

5.

TP4 SQL Simple, Tri et regroupements

A. Table employés

1.

>CREATE TABLE EMPLOYE

(

NumEmp NUMERIC NOT NULL PRIMARY KEY,

NomEmp VARCHAR2(20) NOT NULL CHECK (UPPER(NomEmp) = NomEmp),

DateEmb DATE NOT NULL,

DateSortie DATE DEFAULT NULL

);

```

>INSERT INTO EMPLOYE VALUES(9007, 'CHEVALIER', '01/01/96', NULL);
>INSERT INTO EMPLOYE VALUES(9701, 'LEROY', '17/09/97', NULL);
>INSERT INTO EMPLOYE VALUES(9703, 'LAMI', '17/09/97', NULL);
>INSERT INTO EMPLOYE VALUES(9801, 'SULTAN', '20/03/98', NULL);
>INSERT INTO EMPLOYE VALUES(9802, 'CLEMENCE', '16/10/98', NULL);
>INSERT INTO EMPLOYE VALUES(9803, 'CAVALIER', '22/11/98', NULL);
>INSERT INTO EMPLOYE VALUES(9901, 'ALEXANDRE', '21/02/99', NULL);

```

2.

-- Liste des employes

```
>SELECT * FROM EMPLOYE;
```

<i>NUMEMP</i>	<i>NOMEMP</i>	<i>DATEEMB</i>	<i>DATESORT</i>
9007	CHEVALIER	01/01/96	
9701	LEROY	17/09/97	
9703	LAMI	17/09/97	
9801	SULTAN	20/03/98	
9802	CLEMENCE	16/10/98	
9803	CAVALIER	22/11/98	
9901	ALEXANDRE	21/02/99	

-- Liste des noms de tous les employes

```
>SELECT NomEmp FROM EMPLOYE;
```

NOMEMP

```

-----
CHEVALIER
LEROY
LAMI
SULTAN
CLEMENCE
CAVALIER
ALEXANDRE

```

-- Nom des employes embauches a partir du 1er janvier 1999

```
>SELECT NomEmp FROM EMPLOYE WHERE DateEmb > '01/01/1999';
```

NOMEMP

```

-----
ALEXANDRE

```

-- Liste des employés (Num et Nom) dont le nom commence par la lettre C

```
>SELECT NumEmp, NomEmp FROM EMPLOYE WHERE SUBSTR(NomEmp, 0, 1) = 'C';
```

NUMEMP NOMEMP

9007 CHEVALIER
9802 CLEMENCE
9803 CAVALIER

-- Liste des employés triés par ordre décroissant sur les noms
>SELECT * FROM EMPLOYE ORDER BY NomEmp DESC;

NUMEMP NOMEMP	DATEEMB DATESORT
-----	-----
9801 SULTAN	20/03/98
9701 LEROY	17/09/97
9703 LAMI	17/09/97
9802 CLEMENCE	16/10/98
9007 CHEVALIER	01/01/96
9803 CAVALIER	22/11/98
9901 ALEXANDRE	21/02/99

-- Nombre d'employés embauchés chaque année
>SELECT EXTRACT(YEAR FROM DateEmb), count(NumEmp) FROM EMPLOYE GROUP BY EXTRACT(YEAR FROM DateEmb);

EXTRACT(YEARFROMDATEEMB) COUNT(NUMEMP)

-----	-----
1999	1
1996	1
1998	3
1997	2

-- Nombre d'employés embauchés chaque année ayant un nom de plus de 5 lettres
>SELECT EXTRACT(YEAR FROM DateEmb), count(NumEmp) FROM EMPLOYE WHERE LENGTH(NomEmp) > 5 GROUP BY EXTRACT(YEAR FROM DateEmb);

EXTRACT(YEARFROMDATEEMB) COUNT(NUMEMP)

-----	-----
1999	1
1996	1
1998	3

-- Nombre d'employés embauchés chaque année ayant un nom commençant par L ou C, en ne gardant que les années avec au moins deux employés

```
>SELECT EXTRACT(YEAR FROM DateEmb), count(NumEmp) FROM EMPLOYE WHERE
SUBSTR(NomEmp, 0, 1) = 'L' or SUBSTR(NomEmp, 0, 1) = 'C' GROUP BY
EXTRACT(YEAR FROM DateEmb) HAVING count(*) > 1;
```

```
EXTRACT(YEARFROMDATEEMB) COUNT(NUMEMP)
```

```
-----
1998                2
1997                2
```

```
>DROP TABLE EMPLOYE;
```

B. Table postes

```
cl scr --clear the screen
```

1.

```
>CREATE TABLE POSTES
(
NumEmp NUMERIC NOT NULL,
Poste VARCHAR2(20) NOT NULL CHECK (Poste IN ('PRESIDENT', 'SECRETAIRE',
'DIRECTEUR')),
Salaire NUMERIC NOT NULL,
NumServ VARCHAR2(2),
DateDeb DATE NOT NULL,
DateFin DATE DEFAULT NULL,
CONSTRAINT pk_postes PRIMARY KEY (NumEmp, Poste, DateDeb)
);
```

```
>INSERT INTO POSTES VALUES(9701,'PRESIDENT', 5800,'S2','17/09/97',NULL);
>INSERT INTO POSTES VALUES(9703,'SECRETAIRE', 950,'S1','17/09/97','31/12/98');
>INSERT INTO POSTES VALUES(9703,'SECRETAIRE', 1200,'S1','01/01/99',NULL);
>INSERT INTO POSTES VALUES(9801,'DIRECTEUR', 5300,'S1','07/07/97','31/12/98');
>INSERT INTO POSTES VALUES(9801,'DIRECTEUR', 3200,'S5','20/03/98',NULL);
>INSERT INTO POSTES VALUES(9802,'DIRECTEUR', 3500,'S2','16/10/98',NULL);
>INSERT INTO POSTES VALUES(9803,'INGENIEUR', 2600,'S4','22/11/98',NULL);
>INSERT INTO POSTES VALUES(9901,'DIRECTEUR', 3000,'S3','21/02/99',NULL);
```

2.

-- Liste de tous les noms de postes
>SELECT DISTINCT Poste FROM POSTES;

POSTE

PRESIDENT
SECRETAIRE
DIRECTEUR

-- Postes occupés dont le salaire de l'employé est supérieur ou égal à 3000
>SELECT DISTINCT Poste FROM POSTES WHERE Salaire >= 3000 AND DateFin IS NULL;

POSTE

PRESIDENT
DIRECTEUR

-- Postes occupés, triés par ordre décroissant et salaires par ordre croissant
>SELECT DISTINCT NumEmp, Poste, Salaire FROM POSTES WHERE DateFin IS NULL
ORDER BY Salaire ASC, Poste DESC;

<i>NUMEMP</i>	<i>POSTE</i>	<i>SALAIRE</i>
9703	SECRETAIRE	1200
9901	DIRECTEUR	3000
9801	DIRECTEUR	3200
9802	DIRECTEUR	3500
9701	PRESIDENT	5800

-- Salaire le plus bas
>SELECT min(Salaire) FROM POSTES;

MIN(SALAIRE)

950

-- Moyenne des salaires
>SELECT avg(Salaire) FROM POSTES;

AVG(SALAIRE)

3278,57143

```
-- Moyenne des salaires pour les postes actuellement occupé
>SELECT avg(Salaire) FROM POSTES WHERE DateFin IS NULL;
```

AVG(SALAIRE)

```
-----
3340
```

```
-- Nombre de salariés avec un salaire > 3000
>SELECT DISTINCT count(NumEmp) FROM POSTES WHERE Salaire > 3000;
```

COUNT(NUMEMP)

```
-----
4
```

```
-- Moyenne des salaires actuels pour chaque service
>SELECT NumServ, avg(Salaire) FROM POSTES GROUP BY NumServ;
```

NU AVG(SALAIRE)

```
-----
S2    4650
S1 2483,33333
S5    3200
S3    3000
```

```
-- Moyenne des salaires pour chaque poste avec au moins 2 employés
>SELECT Poste, avg(Salaire) FROM POSTES GROUP BY Poste HAVING count(NumEmp)
> 1;
```

POSTE *AVG(SALAIRE)*

```
-----
SECRETAIRE                      1075
DIRECTEUR                        3750
```

```
>DROP TABLE POSTES;
```

C.Table Etudiants

```
cl scr -- clear screen
```

```
>CREATE TABLE ETUDIANTS(
  NUMERO NUMBER(4) NOT NULL,
  NOM VARCHAR2(25) NOT NULL,
  PRENOM VARCHAR2(25) NOT NULL,
  SEXE CHAR(1) CHECK(SEXE IN ('F', 'M')),
  DATENAISSANCE DATE NOT NULL,
  POIDS NUMBER,
```

```
ANNEE NUMBER,  
CONSTRAINT PK_ETUDIANTS PRIMARY KEY(NUMERO));  
-- Moyenne des poids par sexe  
>SELECT avg(POIDS) FROM ETUDIANTS GROUP BY SEXE;  
  
-- Moyenne des poids par sexe et par tranche d'âge  
  
-- Moyenne des poids par année, par sexe et par tranche d'âge  
  
-- Moyenne des poids par sexe, par année et par tranche d'âge
```


TP5 Jointures

A. Gestion d'un café

-- Liste du contenu de chaque table de la base.

SELECT

*

FROM

LESTABLES;

<i>NUMTABLE</i>	<i>NOMTABLE</i>	<i>NBPLACE</i>
1	entree-gche	6
2	entree-dte	10
3	fenetre1	3
4	fenetre2	8
5	fenetre3	4
6	fond-gche	4
7	fond-dte	2

SELECT

*

FROM

SERVEUR;

<i>NUMSERVEUR</i>	<i>NOMSERVEUR</i>	<i>RUESERVEUR</i>	<i>CPSERVEUR</i>
50	Pizzi	3 rue des lilas	90000 BELFORT
51	Cathy	25 av Roosevelt	90100 DELLE
52	Totof	46 grande rue	90500 BAVILLIERS
53	Pilou	5 impasse Martin	90000 BELFORT
54	Alice	15 rue de la barre	95880 ENGHEN

SELECT

*

FROM

CONSOMMATION;

NUMCONS	LIBCONS	PRIXCONS
---------	---------	----------

100	Cafe	,9
101	Cafe double	1,3
102	Cafe creme	1
105	Chocolat	1,5
106	Biere pression	1,8
107	Biere 25cl	2
108	Biere 33cl	2,2
110	Biere 50cl	2,5
120	Jus de fruits	1,7
121	Jus de fruits presse	2,6
122	Perrier	1,6

NUMCONS	LIBCONS	PRIXCONS
---------	---------	----------

124	Orangina	1,4
130	Coca Cola	1,7

SELECT

*

FROM

FACTURE;

NUMFACTURE	NUMTABLE	NUMSERVEUR	DATEFACT
------------	----------	------------	----------

1200	1	53	01/02/10
1201	5	53	01/02/10
1202	3	52	01/02/10
1203	5	50	01/02/10
1204	4	52	02/02/10
1205	1	53	02/02/10
1206	3	52	02/02/10
1207	5	53	02/02/10
1208	7	54	02/02/10

SELECT

*

FROM

COMPRED;

<i>NUMFACTURE</i>	<i>NUMCONS</i>	<i>QTE</i>
1200	101	3
1200	106	1
1200	120	1
1201	105	2
1201	106	2
1202	100	2
1202	122	1
1203	102	1
1203	108	1
1203	121	1
1203	130	1

<i>NUMFACTURE</i>	<i>NUMCONS</i>	<i>QTE</i>
1204	122	4
1204	124	2
1205	100	2
1206	108	3
1207	108	1
1207	110	2
1208	108	2

-- Nombre de places de la table N°4 (Nbplace).

```

SELECT
    Nbplace
FROM
    LESTABLES
WHERE
    NumTable = 4;

```

<i>NBPLACE</i>
8

-- Liste des consommations dont le prix unitaire est supérieur à 1 euro (Numcons, Libcons, Prixcons).

```

SELECT
    Numcons,
    Libcons
FROM
    CONSOMMATION
WHERE

```

Prixcons > 1;

NUMCONS LIBCONS

101 Cafe double
105 Chocolat
106 Biere pression
107 Biere 25cl
108 Biere 33cl
110 Biere 50cl
120 Jus de fruits
121 Jus de fruits presse
122 Perrier
124 Orangina
130 Coca Cola

-- Liste des serveurs de Belfort et de Delle (Numserv, Nomserveur, Villeserveur)

```
SELECT
    Numserveur,
    Nomserveur
FROM
    SERVEUR
WHERE
    Villeserveur = 'BELFORT'
    OR Villeserveur = 'DELLE';
```

NUMSERVEUR NOMSERVEUR

50 Pizzi
51 Cathy
53 Pilou

-- Liste des factures du 2 février servies par le serveur 52 (Numfact, Numtable).

```
SELECT
    Numfacture,
    NumTable
FROM
    FACTURE
WHERE
    Datefacture = '02/02/10'
    AND Numserveur = 52;
```

NUMFACTURE NUMTABLE

1204	4
1206	3

-- Liste des consommations de la facture 1203 (Numcons, Qte)

```
SELECT
    Numcons,
    Qte
FROM
    COMPREND
WHERE
    Numfactory = 1203;
```

NUMCONS	QTE

102	1
108	1
121	1
130	1

-- Liste des consommations des factures 1200 et 1201 (sans lignes en double)
(Numcons).

```
SELECT
    DISTINCT Numcons,
    Qte
FROM
    COMPREND
WHERE
    Numfactory = 1200
    OR Numfactory = 1201;
```

NUMCONS	QTE

106	1
120	1
106	2
101	3
105	2

-- Liste des serveurs qui sont nés en 1976 (Nomserv, Datenserveur)

```
SELECT
```

```

        Nomserveur,
        Datenserveur
FROM
    SERVEUR
WHERE
    EXTRACT(
        YEAR
        FROM
        Datenserveur
    ) = 1976;

```

```

NOMSERVEUR    DATENSER
-----
Pizzi         12/01/76

```

-- Liste des consommations de type bière (Numcons,Libcons, Prixcons).

```

SELECT
    Numcons,
    Libcons,
    Prixcons
FROM
    CONSOMMATION
WHERE
    Libcons LIKE '%Biere%';

```

```

NUMCONS LIBCONS    PRIXCONS
-----
106 Biere pression    1,8
107 Biere 25cl        2
108 Biere 33cl        2,2
110 Biere 50cl        2,5

```

-- Liste des tables servies après le 1 février.

```

SELECT
    NumTable,
    Datefacture
FROM
    FACTURE
WHERE
    Datefacture > '01/02/10';

```

NUMTABLE DATEFACT

```
-----  
4 02/02/10  
1 02/02/10  
3 02/02/10  
5 02/02/10  
7 02/02/10
```

-- Liste des serveurs dont le nom contient i en deuxième position (Nomserv).

```
SELECT  
    Nomserveur  
FROM  
    SERVEUR  
WHERE  
    SUBSTR(Nomserveur, 2, 1) = 'i';
```

NOMSERVEUR

```
-----  
Pizzi  
Pilou
```

-- Liste des serveurs dont le nom commence par un P (Nomserv)

```
SELECT  
    Nomserveur  
FROM  
    SERVEUR  
WHERE  
    SUBSTR(Nomserveur, 1, 1) = 'P';
```

NOMSERVEUR

```
-----  
Pizzi  
Pilou
```

-- Liste des serveurs par ville (Nomserv, Villeserveur).

```
SELECT  
    Nomserveur,  
    Villeserveur  
FROM  
    SERVEUR  
ORDER BY  
    Villeserveur;
```

NOMSERVEUR	VILLESERVEUR
------------	--------------

Totof	BAVILLIERS
Pilou	BELFORT
Pizzi	BELFORT
Cathy	DELLE
Alice	ENGHIEN

-- Liste des consommations classées par ordre alphabétique sur le libellé
(Libcons, Numcons, Prixcons)

```

SELECT
    Numcons,
    Libcons,
    Prixcons
FROM
    CONSOMMATION
ORDER BY
    Libcons;

```

NUMCONS	LIBCONS	PRIXCONS
106	Biere pression	1,8
107	Biere 25cl	2
108	Biere 33cl	2,2
110	Biere 50cl	2,5
100	Cafe	,9
102	Cafe creme	1
101	Cafe double	1,3
105	Chocolat	1,5
130	Coca Cola	1,7
120	Jus de fruits	1,7
121	Jus de fruits presse	2,6

NUMCONS	LIBCONS	PRIXCONS
124	Orangina	1,4
122	Perrier	1,6

-- Liste des villes où habitent des serveurs (sans lignes en double)(Villeserveur).

```

SELECT
    DISTINCT Villeserveur
FROM
    SERVEUR;

```


VILLESERVEUR

BELFORT

BAVILLIERS

ENGHIEN

DELLE

-- Le nombre de tables du restaurant

```
SELECT
    count(NumTable)
FROM
    LESTABLES;
```

COUNT(NUMTABLE)

7

-- Le nombre de places disponibles sur l'ensemble des tables

```
SELECT
    sum(Nbplace)
FROM
    LESTABLES;
```

SUM(NBPLACE)

37

-- Nombre de factures établies par chaque serveur (Numserv, Nbfacture).

```
SELECT
    Numserveur,
    count(Numfacture)
FROM
    FACTURE
GROUP BY
    Numserveur;
```

NUMSERVEUR COUNT(NUMFACTURE)

```
-----  
52          3  
54          1  
50          1  
53          4
```

-- Nombre de factures établies chaque jour (Datefacture, Nbfacture).

```
SELECT  
    Datefacture,  
    count(Numfacture)  
FROM  
    FACTURE  
GROUP BY  
    Datefacture;
```

DATEFACT COUNT(NUMFACTURE)

```
-----  
01/02/10    4  
02/02/10    5
```

-- Liste des serveurs qui ont établi plus de 3 factures (Numserv, Nbfacture).

```
SELECT  
    Numserveur,  
    count(Numfacture)  
FROM  
    FACTURE  
GROUP BY  
    Numserveur  
HAVING  
    count(Numfacture) > 3;
```

NUMSERVEUR COUNT(NUMFACTURE)

```
-----  
53          4
```

-- Prix moyen des consommations (Prixmoyen)

```
SELECT  
    avg(Prixcons)  
FROM  
    CONSOMMATION;
```

AVG(PRIXCONS)

1,70769231

-- Prix moyen du café (Prixmoyen)

```
SELECT
    avg(Prixcons)
FROM
    CONSOMMATION
WHERE
    Libcons LIKE '%Cafe%';
```

AVG(PRIXCONS)

1,06666667

-- Quantité moyenne consommée pour chaque consommation (Numcons, Qtemoyenne)

```
SELECT
    Numcons,
    avg(Qte)
FROM
    COMPREND
GROUP BY
    Numcons;
```

NUMCONS AVG(QTE)

101	3
122	2,5
110	2
120	1
106	1,5
130	1
102	1
121	1
105	2
124	2
100	2

-- Nombre de serveurs par ville (Villeserveur, Nbserveur).

```
SELECT
    Villeserveur,
    count(Numserveur)
```

```

FROM
    SERVEUR
GROUP BY
    Villeserveur;
VILLESERVEUR  COUNT(NUMSERVEUR)

```

```

-----
BELFORT                2
BAVILLIERS              1
ENGHIEN                 1
DELLE                   1

```

-- Liste des villes dans lesquelles habitent plus d'un serveur (Villeserveur, Nbserveur).

```

SELECT
    Villeserveur
FROM
    SERVEUR
GROUP BY
    Villeserveur
HAVING
    count(Numserveur) > 1;

```

```

VILLESERVEUR

```

```

-----
BELFORT

```

-- Nombre de types de consommations par factures (Numfacture, Nbcons).

```

SELECT
    Numfacture,
    count(Numcons)
FROM
    COMPREND
GROUP BY
    Numfacture;

```

NUMFACTURE COUNT(NUMCONS)

```
-----  
1200      3  
1201      2  
1202      2  
1203      4  
1204      2  
1205      1  
1206      1  
1207      2  
1208      1
```

-- Nombre total de consommations (en comptant la quantité) par
facture(Numfacture, Qtecons).

```
SELECT  
    Numfacture,  
    sum(Qte)  
FROM  
    COMPREND  
GROUP BY  
    Numfacture;
```

NUMFACTURE SUM(QTE)

```
-----  
1204      6  
1200      5  
1201      4  
1207      3  
1202      3  
1205      2  
1208      2  
1206      3  
1203      4
```

-- Nombre de factures par consommation (Numcons, Nbfactures).

```
SELECT  
    Numcons,  
    count(Numfacture)  
FROM  
    COMPREND  
GROUP BY  
    Numcons;
```

NUMCONS COUNT(NUMFACTURE)

101	1
122	2
110	1
120	1
106	2
130	1
102	1
121	1
105	1
124	1
100	2

-- Consommations qui interviennent dans plus de 2 factures (Numcons, Nbfactures)

```
SELECT
    Numcons,
    count(Numfacture)
FROM
    COMPREND
GROUP BY
    Numcons
HAVING
    count(Numfacture) > 2;
```

NUMCONS COUNT(NUMFACTURE)

108	4

-- Liste des serveurs, triés par nom de ville croissant, puis nom de serveur croissant

```
SELECT
    Numserveur,
    Nomserveur,
    Villeserveur
FROM
    serveur
ORDER BY
    Villeserveur ASC,
    Nomserveur ASC;
```

NUMSERVEUR	NOMSERVEUR	VILLESERVEUR
52	Totof	BAVILLIERS
53	Pilou	BELFORT
50	Pizzi	BELFORT
51	Cathy	DELLE
54	Alice	ENGHIEN

-- Liste des serveurs, triés par nom de ville décroissant, puis nom de serveur croissant.

```

SELECT
    Numserveur,
    Nomserveur,
    Villeserveur
FROM
    serveur
ORDER BY
    Villeserveur DESC,
    Nomserveur ASC;

```

NUMSERVEUR	NOMSERVEUR	VILLESERVEUR
54	Alice	ENGHIEN
51	Cathy	DELLE
53	Pilou	BELFORT
50	Pizzi	BELFORT
52	Totof	BAVILLIERS

-- Liste des factures avec leur numéro de table et le nom du serveur (Numfacture, Numtable, Nomserveur).

```

SELECT
    Numfacture,
    Numtable,
    Nomserveur
FROM
    FACTURE NATURAL
    JOIN SERVEUR;

```

NUMFACTURE NUMTABLE NOMSERVEUR

```
-----  
1200      1 Pilou  
1201      5 Pilou  
1202      3 Totof  
1203      5 Pizzi  
1204      4 Totof  
1205      1 Pilou  
1206      3 Totof  
1207      5 Pilou  
1208      7 Alice
```

-- Liste des factures de la table 5 avec le nom du serveur (Numfacture, Nomserveur).

```
SELECT  
    Numfacture,  
    Nomserveur  
FROM  
    FACTURE NATURAL  
    JOIN SERVEUR  
WHERE  
    Numtable = 5;
```

NUMFACTURE NOMSERVEUR

```
-----  
1203 Pizzi  
1201 Pilou  
1207 Pilou
```

-- Liste des factures avec leur nom de table et le nom du serveur (Numfacture, Numtable, Nomserveur).

```
SELECT  
    Numfacture,  
    Numtable,  
    Nomserveur  
FROM  
    SERVEUR NATURAL  
    JOIN FACTURE NATURAL  
    JOIN LESTABLES;
```


NUMFACTURE	NOMTABLE	NOMSERVEUR
------------	----------	------------

1200	entree-gche	Pilou
1205	entree-gche	Pilou
1202	fenetre1	Totof
1206	fenetre1	Totof
1204	fenetre2	Totof
1201	fenetre3	Pilou
1203	fenetre3	Pizzi
1207	fenetre3	Pilou
1208	fond-dte	Alice

-- Liste des serveurs et des tables qu'ils ont servies ordonnés selon le nom du serveur (pas de ligne double) (Nomserveur, Nomtable).

```

SELECT
    DISTINCT Nomserveur,
    Nomtable
FROM
    SERVEUR NATURAL
    JOIN FACTURE NATURAL
    JOIN LESTABLES
ORDER BY
    Nomserveur;

```

NOMSERVEUR	NOMTABLE
------------	----------

Alice	fond-dte
Pilou	fenetre3
Pilou	entree-gche
Pizzi	fenetre3
Totof	fenetre1
Totof	fenetre2

-- Liste des consommations de la facture 1203 avec leur nom, leur prix et leur quantité (Numcons, Libcons, Prixcons, Qte)

```

SELECT
    Numcons,
    Libcons,
    Prixcons,
    Qte
FROM
    FACTURE NATURAL
    JOIN COMPREND NATURAL
    JOIN CONSOMMATION

```

WHERE

Numfacture = 1203;

NUMCONS	LIBCONS	PRIXCONS	QTE
102	Cafe creme	1	1
108	Biere 33cl	2,2	1
121	Jus de fruits presse	2,6	1
130	Coca Cola	1,7	1

-- Liste des consommations du premier février de la table 5 avec leur nom, leur prix et leur quantité (Numcons, Libcons, Prixcons, Qte).

SELECT

Numcons,
Libcons,
Prixcons,
Qte

FROM

FACTURE NATURAL
JOIN COMPREND NATURAL
JOIN CONSOMMATION

WHERE

Numtable = 5
AND Datefacture = '01/02/10';

NUMCONS	LIBCONS	PRIXCONS	QTE
102	Cafe creme	1	1
108	Biere 33cl	2,2	1
121	Jus de fruits presse	2,6	1
130	Coca Cola	1,7	1
NUMCONS	LIBCONS	PRIXCONS	QTE
105	Chocolat	1,5	2
106	Biere pression	1,8	2
102	Cafe creme	1	1
108	Biere 33cl	2,2	1
121	Jus de fruits presse	2,6	1
130	Coca Cola	1,7	1

-- Liste des tables et des numéros de factures qui leur sont associées.

Attention, on veut voir toutes les tables même si elles n'ont pas de factures. La table de départ (celle du FROM) sera LESTABLES (Nomtable, Numfacture).

SELECT

```

        Nomtable,
        Numfacture
FROM
    LESTABLES
    LEFT JOIN FACTURE ON LESTABLES.Numtable = FACTURE.Numtable;

```

<i>NOMTABLE</i>	<i>NUMFACTURE</i>
entree-gche	1200
fenetre3	1201
fenetre1	1202
fenetre3	1203
fenetre2	1204
entree-gche	1205
fenetre1	1206
fenetre3	1207
fond-dte	1208
fond-gche	
entree-dte	

-- Même question que précédemment,mais avec FACTURE comme table de départ.

```

SELECT
    Nomtable,
    Numfacture
FROM
    FACTURE
    RIGHT JOIN LESTABLES ON LESTABLES.Numtable =
FACTURE.Numtable;

```

<i>NOMTABLE</i>	<i>NUMFACTURE</i>
<i>entree-gche</i>	1200
<i>fenetre3</i>	1201
<i>fenetre1</i>	1202
<i>fenetre3</i>	1203
<i>fenetre2</i>	1204
<i>entree-gche</i>	1205
<i>fenetre1</i>	1206
<i>fenetre3</i>	1207
<i>fond-dte</i>	1208
<i>fond-gche</i>	
<i>entree-dte</i>	

-- Liste des tables qui n'ont eu aucune factures (Numtable,Nomtable).

```

SELECT
    LESTABLES.Numtable,
    Nomtable
FROM
    LESTABLES
    LEFT JOIN FACTURE ON LESTABLES.Numtable = FACTURE.Numtable
WHERE
    Numfacture IS NULL;

```

NUMTABLE NOMTABLE

6 *fond-gche*
2 *entree-dte*

-- Liste des consommations qui ont déjà été servies par le serveur 52
(Numcons, Libcons)

```

SELECT DISTINCT
    Numcons,
    Libcons
FROM
    CONSOMMATION NATURAL
    JOIN FACTURE
WHERE
    Numserveur = 52;

```

NUMCONS LIBCONS

100 Cafe
101 Cafe double
102 Cafe creme
105 Chocolat
106 Biere pression
107 Biere 25cl
108 Biere 33cl
110 Biere 50cl
120 Jus de fruits
121 Jus de fruits presse
122 Perrier

-- Liste des consommations qui n'ont jamais été servies (Numcons, Libcons)

```
SELECT
    CONSOMMATION.Numcons,
    Libcons
FROM
    CONSOMMATION
    LEFT JOIN COMPREND ON CONSOMMATION.Numcons =
COMPREND.Numcons
WHERE
    COMPREND.Numcons IS NULL;
```

NUMCONS LIBCONS

107 Biere 25cl

-- La liste des factures avec leur date et leur nombre de consommations
(prendre en compte la quantité) (Numfacture, Datefacture, Nbcons)

```
SELECT
    Numfacture,
    Datefacture,
    sum(Qte)
FROM
    FACTURE NATURAL
    JOIN COMPREND
GROUP BY
    Numfacture,
    Datefacture;
```

NUMFACTURE DATEFACT SUM(QTE)

```
-----
1200 01/02/10      5
1201 01/02/10      4
1207 02/02/10      3
1205 02/02/10      2
1208 02/02/10      2
1203 01/02/10      4
1204 02/02/10      6
1202 01/02/10      3
1206 02/02/10      3
```

-- La liste des factures et le montant de leur addition (Numfacture, Prixfacture)

```
SELECT
    Numfacture,
    sum(Qte * Prixcons)
FROM
    FACTURE NATURAL
    JOIN COMPREND NATURAL
    JOIN CONSOMMATION
GROUP BY
    Numfacture;
```

NUMFACTURE SUM(QTE*PRIXCONS)

```
-----
1204      9,2
1200      7,4
1201      6,6
1207      7,2
1202      3,4
1205      1,8
1208      4,4
1206      6,6
1203      7,5
```

-- Nombre de consommations servies par jour (Datefacture, Nbcons)

```
SELECT
    DateFacture,
    sum(Qte)
FROM
    COMPREND NATURAL
    JOIN FACTURE
GROUP BY
    DateFacture;
```

DATEFACT SUM(QTE)

```
-----  
01/02/10      16  
02/02/10      16
```

-- Montant global du chiffre d'affaire par jour (Datefacture, ca)

```
SELECT  
    DateFacture,  
    sum(Qte * Prixcons)  
FROM  
    CONSOMMATION NATURAL  
    JOIN COMPREND NATURAL  
    JOIN FACTURE  
GROUP BY  
    DateFacture;
```

*DATEFACT SUM(QTE*PRIXCONS)*

```
-----  
01/02/10      24,9  
02/02/10      29,2
```

-- La liste des serveurs par nom et leur nombre de factures. Attention, les serveurs n'ayant fait aucune facture doivent apparaître dans le résultat (Nomserveur, Nbfactures)

```
SELECT  
    Nomserveur,  
    count(Numfacture)  
FROM  
    SERVEUR  
    LEFT JOIN FACTURE ON SERVEUR.Numserveur = FACTURE.Numserveur  
GROUP BY  
    SERVEUR.Numserveur;
```

NOMSERVEUR COUNT(NUMFACTURE)

```
-----  
Cathy          0  
Totof          3  
Pilou          4  
Pizzi          1  
Alice          1
```

-- La liste des serveurs par nom et le nombre de consommations qu'ils ont servies (NomServeur, Nbcons).

```
SELECT
    Nomserveur,
    sum(Qte)
FROM
    SERVEUR NATURAL
    JOIN FACTURE NATURAL
    JOIN COMPREND
GROUP BY
    Nomserveur;
```

NOMSERVEUR	SUM(QTE)
Totof	12
Pilou	14
Pizzi	4
Alice	2

-- La liste des serveurs par nom et leur chiffre d'affaire (somme des additions encaissées) (Nomserveur, Ca)

```
SELECT
    Nomserveur,
    sum(Qte * Prixcons)
FROM
    SERVEUR NATURAL
    JOIN FACTURE NATURAL
    JOIN COMPREND NATURAL
    JOIN CONSOMMATION
GROUP BY
    Nomserveur;
```

NOMSERVEUR	SUM(QTE*PRIXCONS)
Totof	19,2
Pilou	23
Pizzi	7,5
Alice	4,4

-- Le nom des tables qui ont eu au moins deux factures (Nomtable, Nbfactures).

```
SELECT
    Nomtable
```



```
FROM
    LESTABLES NATURAL
    JOIN FACTURE
GROUP BY
    Nomtable
HAVING
    count(Numfacture) > 1;
```

NOMTABLE

fenetre1
entree-gche
fenetre3

-- La liste complète des consommations et le nombre de factures dans
lesquels elles apparaissent (Libcons, Nbfactures)

```
SELECT
    Libcons,
    count(Numfacture)
FROM
    CONSOMMATION
    LEFT JOIN COMPREND ON CONSOMMATION.Numcons =
COMPREND.Numcons
GROUP BY
    Libcons;
```

LIBCONS	COUNT(NUMFACTURE)
---------	-------------------

Coca Cola	1
Jus de fruits	1
Cafe creme	1
Orangina	1
Perrier	2
Biere 33cl	4
Biere pression	2
Cafe	2
Biere 25cl	0
Jus de fruits presse	1
Cafe double	1

LIBCONS	COUNT(NUMFACTURE)
---------	-------------------

Chocolat	1
Biere 50cl	1

-- La liste complète des tables et leur chiffre d'affaire (Nomtable, Ca)

SELECT

Nomtable,
sum(Prixcons * Qte)

FROM

LESTABLES
LEFT JOIN FACTURE NATURAL
JOIN COMPREND NATURAL

JOIN CONSOMMATION ON LESTABLES.Numtable = FACTURE.Numtable

GROUP BY

Nomtable;

NOMTABLE	SUM(PRIXCONS*QTE)
----------	-------------------

fenetre1	10
fond-gche	
fond-dte	4,4
fenetre2	9,2
entree-dte	
entree-gche	9,2
fenetre3	21,3

genealogie

-- Liste des enfants d'Elizabeth II (Nom, DateNaissance)

SELECT Nom, DateNaissance FROM genealogie WHERE Mere = 3;

NOM	DATENAIS
-----	----------

Prince Charles	14/11/48
Princesse Anne	15/08/50
Prince Andrew	19/02/60
Prince Edward	10/03/64

-- La mère du prince William(Nom, DateNaissance)

SELECT Nom, DateNaissance FROM genealogie WHERE numPer = (SELECT Mere FROM genealogie WHERE numPer = 11);

NOM	DATENAIS
-----	----------

Diana Spencer	01/07/61
---------------	----------

-- Les parents d'Elizabeth II (Nom, DateNaissance).

SELECT Nom,DateNaissance FROM genealogie WHERE numPer IN (SELECT Mere FROM genealogie WHERE numPer = 3 UNION SELECT Pere FROM genealogie WHERE numPer = 3);

NOM	DATENAIS
-----	----------

George VI	14/12/95
Elizabeth Bowes-Lyon	04/08/00

-- Les frères et sœurs du Prince Charles(Nom, DateNaissance)

SELECT Nom, DateNaissance FROM genealogie WHERE numPer != 6 AND (Pere = (SELECT Pere FROM genealogie WHERE numPer = 6) OR Mere = (SELECT Mere FROM genealogie WHERE numPer = 6));

NOM	DATENAIS
-----	----------

Princesse Anne	15/08/50
Prince Andrew	19/02/60
Prince Edward	10/03/64

-- Le nom des individus, le nom de leur père (ou NULL) et le nom de leur mère (ou NULL) (Nom, Nompere, Nommere).

SELECT t1.nom, t2.nom as Nompere, t3.nom as Nommere FROM (genealogie t1 LEFT JOIN genealogie t2 ON t1.Pere = t2.numPer) LEFT JOIN genealogie t3 ON t1.Mere = t3.numPer;

NOM	NOMPERE	NOMMERE
Elizabeth II	George VI	Elizabet
Margaret du Royaume-Uni	George VI	Elizabet
Prince Charles	Philip Mountbatten	Elizabet
Princesse Anne	Philip Mountbatten	Elizabet
Prince Andrew	Philip Mountbatten	Elizabet
Prince Edward	Philip Mountbatten	Elizabet
Prince William	Prince Charles	Diana Sp
George VI		
Elizabeth Bowes-Lyon		
Philip Mountbatten		
Diana Spencer		

-- La liste des individus et le nombre de leurs enfants étant dans la base de données (Nom,NbEnfants).

SELECT t1.nom, count(t2.numPer) from genealogie t1 LEFT JOIN genealogie t2 ON t1.numPer = t2.Pere OR t1.numPer = t2.Mere GROUP BY t1.nom;

NOM	COUNT(T2.NUMPER)
George VI	2
Prince William	0
Prince Andrew	0
Prince Edward	0
Diana Spencer	1
Elizabeth II	4
Prince Charles	1
Princesse Anne	0
Margaret du Royaume-Uni	0
Philip Mountbatten	4
Elizabeth Bowes-Lyon	2

TP6

--1

select nom,prenom,date_naissance from eleves;

-- On selectionne 3 colonnes de la table eleves

NOM	PRENOM	DATE_NAISS
Brisefer	Benoit	10-12-1978
G ®nial	Olivier	10-04-1978
Jourdan	Gil	28-06-1974
Spring	Jerry	16-02-1974
Tsuno	Yoko	29-10-1977
Lebut	Marc	29-04-1974
Lagaffe	Gaston	08-04-1975
Dubois	Robin	20-04-1976
Walth ®ry	Natacha	07-09-1977
Danny	Buck	15-02-1973

--2

select * from activites;

--affiche toutes les données de la table activites

NIVEAU	NOM	EQUIPE
1	Mini foot	Amc Indus
1	Surf	Les planchistes ...
2	Tennis	Ace Club
3	Tennis	Ace Club
1	Volley ball	Avs80
2	Mini foot	Les as du ballon
2	Volley ball	smash

--3

select nom,specialite from professeurs;

--affiche les specialite des professeurs

NOM	SPECIALITE
Bottle	po ®sie
Bolenov	r ®seau
Tonilaclasse	poo
Pastecnov	sql
Selector	sql
Vilplusplus	poo
Francesca	
Pucette	sql

--4

select nom,prenom,poids from eleves where(poids<45 and annee=1) or annee=2;

--on affiche le nom et prenom des eleves de premiere annee avec un poids inferieur à 45 et tous les etudiants de 2ème annee

NOM	PRENOM	POIDS
Brisefer	Benoît	35
Gagnial	Olivier	42
Jourdan	Gil	72
Spring	Jerry	78
Lebut	Marc	75
Dubois	Robin	60
Danny	Buck	82

--5

select nom,poids from eleves where poids between 60 and 80;

--affiche le nom des eleves dont le poids est entre 60 et 80

NOM	POIDS
Jourdan	72
Spring	78
Lebut	75
Lagaffe	61
Dubois	60

--6

select nom,specialite from professeurs where specialite in('poésie','sql');

-- affiche le nom des professeurs spécialisé en poésie ou sql

NOM	SPECIALITE
Bottle	poésie
Pastecnov	sql
Selector	sql
Pucette	sql

--7

select nom from eleves where nom like 'L%';

--affiche le nom des eleves qui commence par L

NOM
Lebut
Lagaffe

--8

```
select nom from professeurs where specialite is null;
--les noms des professeurs qui n'ont pas de spécialité
```

NOM

Francesca

--9

```
select nom,prenom,poids,annee from eleves where poids<45 and annee=1;
--affiche tous les eleves de premiere avec un poids inferieur a 45
```

NOM	PRENOM	POIDS	ANNEE
Brisefer	Benoit	35	1
G ®nial	Olivier	42	1

--10

```
select nom,NVL(specialite,'****') as specialite from professeurs;
--affiche les noms et specialite des professeurs est remplace une specialite vide par
'****'
```

NOM	SPECIALITE
Bottle	po ®sie
Bolenov	r ®seau
Tonilaclasse	poo
Pastecnov	sql
Selector	sql
Vilplusplus	poo
Francesca	****
Pucette	sql

--11

--*****

--12

```
select e.nom from (ACTIVITES_PRATIQUEES ap join ACTIVITES a on (ap.NOM =
a.nom and ap.NIVEAU = a.NIVEAU)) join ELEVES e on ap.num_eleve=e.num_eleve
where a.equipe like'Amc Indus';
```

NOM

Brisefer

G_l®nial

Tsuno

Dubois

Walth_l®ry

Danny

--13

--*****

--14

select nom, SALAIRE_ACTUEL, (SALAIRE_ACTUEL-SALAIRE_BASE)as
augmentation_mensuelle from professeurs where specialite like'sql';

NOM

SALAIRE_ACTUEL AUGMENTATION_MENSUELLE

Pastecnov 2500000 0

Selector 1900000 0

Pucette 2500000 500000

--15

select nom from professeurs where salaire_actuel>((Salaire_base*125)/100);

NOM

Bottle

Bolenov

Francesca

--16

select e.nom as nom_eleve, c.nom as nom_cours,(r.points*5) as points from (cours c
join resultats r on c.num_cours=r.num_cours) join eleves e on
r.num_eleve=e.num_eleve where e.nom like'Tsuno';

NOM_ELEVE

NOM_COURS

POINTS

Tsuno R_l®seau 25

Tsuno Sgbd 32,5

Tsuno Sgbd 65

Tsuno Analyse 65

--17

select AVG(poids) as poid_moyen from eleves where annee=1;

POID_MOYEN

48,4

--18

select sum(points) as total_point from resultats where num_eleve=3;

TOTAL_POINT

65

--19

select min(r.points) as min_point,max(r.points) as max_point from resultats r join
eleves e on r.num_eleve=e.num_eleve where e.nom like 'Brisefer';

MIN_POINT MAX_POINT

8 20

--20

select count(num_eleve) as nb_annee1_eleve from eleves where annee=1;

NB_ANNEE1_ELEVE

5

--21

select avg(SALAIRE_BASE) as moy_base, avg(salaire_actuel-salaire_base) as
moy_augmentation,avg(salaire_actuel) as moy_actuel from professeurs where
specialite like 'sql';

MOY_BASE MOY_AUGMENTATION MOY_ACTUEL

2133333,33 166666,667 2300000

--22

select extract(year from DER_PROM) as derniere_promo from professeurs where
nom like 'Pucette';

DERNIERE_PROMO

1996

--23

```
select          Num_prof,Date_entree,DER_PROM,(extract(year          from
DER_PROM)-extract(year  from  DATE_ENTREE)) as  annee_ecoule  from
professeurs;
```

NUM_PROF DATE_ENTRE DER_PROM ANNEE_ECOULE

1	01-10-1970	01-10-1988	18
2	15-11-1968	01-10-1998	30
3	01-10-1979	01-01-1989	10
4	01-10-1975		
5	15-10-1982	01-10-1988	6
6	25-04-1990	05-06-1994	4
7	01-10-1975	11-01-1998	23
8	06-12-1988	29-02-1996	8

--24

```
select  AVG(extract(year  from  sysdate)-extract(year  from  date_naissance)) as
age_moy from  eleves;
```

AGE_MOY

43,4

--25

```
select  nom  ,(Der_prom-Date_entree)/30  as  nb_mois  from  professeurs  where
((DER_PROM-Date_entree)/30)>50;
```

NOM	NB_MOIS
-----	-----
Bottle	219,166667
Bolenov	363,733333
Tonilaclasse	112,666667
Selector	72,6
Vilplusplus	50,066667
Francesca	271,266667
Pucette	88,033333

--26

```
select nom,date_naissance from eleves where*****
```

```
--27
```

```
select * from eleves order by annee,nom;
```

```
--28
```

```
select e.nom as nom_eleve, c.nom as nom_cours,(r.points*5) as points from (cours c
join resultats r on c.num_cours=r.num_cours) join eleves e on
r.num_eleve=e.num_eleve where e.nom like'Tsuno' order by r.points desc;
```

NOM_ELEVE	NOM_COURS	POINTS
Tsuno	Sgbd	65
Tsuno	Analyse	65
Tsuno	Sgbd	32,5
Tsuno	R ®seau	25

```
--29
```

```
select e.nom, avg(r.points) as moyenne from eleves e join resultats r on
e.num_eleve=r.num_eleve where annee=1 group by nom;
```

NOM	MOYENNE
Tsuno	9,375
Brisefer	13,375
G ®nial	9,5
Lagaffe	10,375
Walth ®ry	15,875

```
--30
```

```
--select*****
```

```
--31
```

```
--*****
```

```
--32
```

```
select e.nom from (ACTIVITES_PRATIQUEES ap join ACTIVITES a on (ap.NOM =
a.nom and ap.NIVEAU = a.NIVEAU)) join ELEVES e on ap.num_eleve=e.num_eleve
where a.equipe like'Amc Indus';
```

NOM

Brisefer
G |[®]nial
Tsunō
Dubois
Walth |[®]ry
Danny

--33

--*****

--34

select nom,poids from eleves where annee=1 and poids>(select max(poids) from
eleves where annee=2);

--35

select nom,poids from eleves where annee=1 and poids>(select min(poids) from
eleves where annee=2);

NOM POIDS

Lagaffe 61

--36

select e.nom,e.poids,e.annee from eleves e where e.poids>(select avg(e2.poids)
from eleves e2 where e.annee=e2.annee);

NOM POIDS ANNEE

Spring 78 2
Lebut 75 2
Lagaffe 61 1
Walth |[®]ry 59 1
Danny 82 2

