

SQL : Structured Query Language

Il s'agit d'un langage de requête reprenant les concepts de l'algèbre relationnelle.
Deux modes de travail : mode interactif et mode intégré dans d'autres langages de programmation (C, COBOL, FORTRAN...)

1. Notations

Les mots réservés du langage sont en majuscule (en fait le logiciel ne fait pas la différence);

Les éléments terminaux sont en minuscule;

Les éléments non-terminaux sont encadrés par des <>;

Les parties facultatives sont encadrées par des [];

Les alternatives sont séparées par des \ et encadrées par des {};

Une répétition d'éléments est notée { }^o, dans l'instruction, le séparateur est la virgule;

Une instruction se termine par un ; et peut prendre plusieurs lignes.

2. Base utilisée pour les exemples

PIECE (NOP, DESIGNATION, COULEUR, POIDS)

SERVICE (NOS, INTITULE, LOCALISATION)

ORDRE (NOS, NOP, QUANTITE)

3. Définitions de données

▪ Définition des tables

CREATE TABLE nom-table ({<élément-table>}^o);

<élément-table> := nom-colonne <type> [contrainte-colonne] | <contrainte-table>

<type> : définit le type de la donnée :

NUMBER nombre (décimal ou entier)

CHAR (n) chaîne de longueur fixe

VARCHAR (n) chaîne de longueur variable, maxi n

DATE date

NOT NULL : à la saisie, le champ doit obligatoirement être renseigné

ex : CREATE TABLE SERVICE (nos CHAR(3) NOT NULL, intitulé CHAR(20), localisation CHAR(20)); PS : 3 est nombre de caractère dans une colonne

▪ Définition des contraintes colonnes

[[CONSTRAINT nom-contrainte][NOT] NULL]

[[CONSTRAINT nom-contrainte]{UNIQUE | PRIMARY KEY}]

[[CONSTRAINT nom-contrainte] REFERENCES <nom-table> [({nom-colonne})^o] [ON

DELETE CASCADE]]

[[CONSTRAINT nom-contrainte] CHECK (<condition>)]

- Définition des contraintes tables

```
[[CONSTRAINT nom-contrainte]{UNIQUE | PRIMARY KEY}({nom-colonne}°)  
[[CONSTRAINT nom-contrainte] FOREIGN KEY ({nom-colonne}°) REFERENCES <nom-  
tabla> [( {nom-colonne}°)] [ON DELETE CASCADE]]  
[[CONSTRAINT non-contraint] CHECK (<condition>)]
```

- Création des indexes

```
CREATE [UNIQUE] INDEX nom-indexe ON nom-table ( {nom-colonne [ {ASC \ DESC}]}°);
```

avec

UNIQUE : évite les doubles

ASC-DESC, sens du tri, ASC par défaut.

- Destruction

```
DROP TABLE nom-table
```

```
DROP INDEX nom-table
```

- Ajout d'éléments dans un table

```
ALTER TABLE nom-table ADD ({<élément-table>}°) (ps : ajouter une colonne)
```

Ex : ALTER TABLE PIECE ADD CONSTRAINT cle_piece primary key (nop);

- Modification de définition de colonnes

```
ALTER TABLE nom-table MODIFY ( {nom-colonne <type> [NOT NULL]}°);
```

4. Mises à jour

- Insertion d'une ligne

```
INSERT INTO nom-table [( {nom-colonne}°)] VALUES ( {<val>}°);
```

avec

<val> ::= {constante\variable\NULL}

la liste de val doit correspondre à celle des colonnes à insérer, tant pour le type que pour le nombre.

```
INSERT INTO nom-table [ {nom-colonne}°] <requête-pleine>;
```

avec

<requête-pleine> résultat d'une requête ensembliste (c.f. § requête)

- Suppression de lignes

```
DELETE FROM nom-table [WHERE <condition>];
```

avec

<condition> condition logique (c.f. § requête)

- Mise à jour de colonnes

```
UPDATE nom-table SET {nom-colonne = {<expression>\NULL}}
```

```
[WHERE <condition>];
```

Ex: UPDATE ORDRE SET quantité = quantité + 10 WHERE nop = 'P1';

5. Requêtes

▪ Sous-requête

```
SELECT [{ALL\DISTINCT}] {{{<expression>}}\*}
FROM {nom-table}°
[WHERE <condition>]           -sur le colonne ou expression
[GROUP BY {<expression>}°]    -il groupe les valeurs identiques
[HAVING <condition>]         -s'il y a une condition sur le group (toujours avec Group by)
```

Les clauses **GROUP BY** et **HAVING** servent à grouper les lignes résultantes de la requête en fonction de certaines conditions;

ALL ou **DISTINCT** : conserve ou élimine les lignes doubles, **ALL** par défaut.

<expression> : expression arithmétique dont les opérands sont des noms de colonnes; il peut s'agir aussi (souvent) d'une liste de colonne.

***** : toutes les colonnes.

FROM : les tables à manipuler.

WHERE : condition que doivent satisfaire les lignes à sélectionner.

▪ Requête pleine

```
{<sous-requête>\<requête-pleine>}
<opérateur ensembliste>
{<sous-requête>\<requête-plein>};
```

avec

<opérateur ensembliste> ::= {INTERSECT\MINUS\UNION}

▪ Requête

```
<requête-pleine> [ORDER BY] {{{nom-colonne [{ASC\DESC}]}°\{numéro-colonne [{ASC\DESC}]}°}
```

▪ Condition

Expression logique entre parenthèses, faisant intervenir les opérateurs logiques **AND**, **OR** et **NOT**.

<condition> ::= <forme-prédicative> {AND\OR} <forme-prédicative>

<forme-prédicative> ::= [NOT] {<prédicat>\<condition>}

six formes de prédicats :

- 1) comparaison scalaire entre deux valeurs
- 2) prédicat quantifié avec **ANY** et **ALL** pour comparer une valeur à un ensemble de valeurs
- 3) comparaison **BETWEEN** et **LIKE** -betwen And
- 4) test de valeur nulle
- 5) comparaison ensembliste **IN**
- 6) existence de ligne : **EXISTS**

▪ Fonctions d'agrégation sur un ensemble de valeurs

AVG : moyenne

SUM : somme

MIN : valeur minimum

MAX : valeur maximum

COUNT : nombre de valeur

6. Explications par des exemples (Création de la base Fabrication)

Requête simple

R1 Dans l'ordre croissant les numéros de services ayant en commande la pièce P1 avec une quantité supérieur à 10
SELECT nos FROM ORDRE
WHERE nop = 'P1' AND quantité > 10
ORDER BY nos; // sert à trier

R2 Le poids de la pièce emballée (+ 20%) pour chaque pièce
SELECT nop, poids+0,2*poids FROM PIECE
ORDER BY 2 **DESC**; // ordre décroissant sur le deuxième argument

R3 Donner le nombre de services
SELECT COUNT (*) FROM SERVICE;

R4 La quantité moyenne commandée pour la pièce P3
SELECT AVG(quantité) FROM ORDRE
WHERE nop = 'P3';

Jointure interne

R5 La liste des pièces commandées par le service 'S1' avec libellé et poids
SELECT nos, **ORDRE.nop**, désignation, poids
FROM ORDRE, PIECE
WHERE **ORDRE.nop** = **PIECE.nop** AND nos = 'S1'
ORDER BY **ORDRE.nop**;

On peut utiliser des **synonymes local** pour alléger l'écriture :

```
SELECT nos, R.nop,désignation, poids
FROM ORDRE R, PIECE P // ORDRE sera appelée R et PIECE P
WHERE R.nop = P.nop and nos = 'S1'
ORDER BY R.nop;
```

Le mécanisme de synonymie permet de faire des jointures sur une même table : **auto-jointure**

R6 Les numéros de service qui ont commandé P1 et P3 en même quantité.
SELECT A1.nos
FROM ORDRE A1, ORDRE A2
WHERE A1.nop='P1' AND A2.nop = 'P3' AND A1.nos = A2.nos AND A1.quantité =
A2.quantité;

Opérations ensemblistes

R7 Les numéros des pièces qui ont un poids inférieur à 50 ou qui ont été commandées par le service S2
SELECT nop FROM PIECE WHERE poids < 50
UNION
SELECT nop FROM ORDRE WHERE nos = 'S2';

Prédicat IS NULL (ou IS NOT NULL)

R8 Les numéros des services qui n'ont pas valué leur commande
SELECT nos FROM ORDRE
WHERE quantité IS NULL;

Prédicat IS LIKE (IS NOT LIKE)

si une forme particulière est présente dans une chaîne de caractères (% = joker)

R9 Le numéro des pièces dont la désignation contient 'ON'
SELECT nop FROM PIECE
WHERE désignation LIKE '*ON*';

Prédicat IN (NOT IN)

détermine si une valeur appartient (ou pas) à une liste du même type

R10 Désignation des pièces de couleur rouge ou noire
SELECT désignation FROM PIECE
WHERE couleur IN ('blue','noire');

SELECT imbriqué

Le résultat d'un SELECT est une table (ou une valeur), on va donc pouvoir refaire un SELECT sur ce retour d'information.

R11 Les numéros des services qui ont commandé P3 avec une quantité inférieure à la quantité moyenne commandée pour cette pièce
SELECT nop, nos, quantité FROM ORDRE
WHERE nop = 'P3' AND quantité <
(SELECT AVG(quantité) FROM ORDRE
WHERE nop = 'P3');

Corrélation d'appel entre deux SELECT

R12 Les caractéristiques de chaque pièce ayant un poids inférieur à la moyenne des poids des pièces de leur couleur
SELECT * FROM PIECE P
WHERE poids <
(SELECT AVG (poids) FROM PIECE
WHERE couleur = P.couleur)
ORDER BY nop;

Emploi du IN

R13 Le numéro et la désignation des pièces commandées par 'S1'
SELECT nop, désignation FROM PIECE
WHERE nop IN
(SELECT nop FROM ORDRE
WHERE nos = 'S1')
ORDER BY nop;

R14 Le numéro et la désignation des pièces commandées par un service de diffusion
SELECT nop, désignation FROM PIECE
WHERE nop IN
(SELECT nop FROM ORDRE
WHERE nos IN
(SELECT nos FROM SERVICE
WHERE intitulé IS LIKE 'diffusion%'))
ORDER BY nop;

Prédicat ALL et ANY

ALL : vrai si

- la requête ne fournit aucune valeur

- si le prédicat de comparaison associé est vrai pour toutes les valeurs

ANY : vrai si le prédicat prend au moins une fois la valeur vraie

R15 Les numéros des services ayant recommandé au moins une pièce en quantité strictement supérieure à chacune des quantités de pièces déjà commandées par le service.

```
SELECT DISTINCT nos FROM ORDRE
```

```
WHERE quantité > ALL
```

```
(SELECT quantité FROM ORDRE WHERE nos = 'S1');
```

Prédicat EXISTS : faux si la requête a pour résultat l'ensemble vide

R16 Intitulés des services ayant commandé aucune pièce

```
SELECT intitulé FROM SERVICE
```

```
WHERE NOT EXISTS
```

```
(SELECT * FROM ORDRE C
```

```
WHERE C.nos = S.nos);
```

Subtilité : expression du prédicat "pour tout"

Pour tout x $P \Leftrightarrow \text{NOT}(\text{EXIST } x (\text{NOT } (P)))$

en français :

"Services tels que pour toute pièce il existe une commande"

\Leftrightarrow

"Services tels qu'il n'existe pas une pièce telle qu'il n'y ait pas de commande pour cette pièce"

R17 SELECT intitulé FROM SERVICE S

```
WHERE NOT EXISTS
```

```
(SELECT * FROM PIECE P
```

```
WHERE NOT EXISTS
```

```
(SELECT * FROM ORDRE C
```

```
WHERE S.nos=C.nos AND C.nop = P.nop
```

```
)
```

```
);
```

Jointure externe

Entre R et S : conserver toutes les lignes de R, Même celles qui n'ont pas de ligne en

correspondance dans S

R18 Pour chaque service, donner son numéro et son intitulé; et pour ceux qui ont des commandes, la liste des pièces commandées

```
SELECT S.nos, intitulé, nop, FROM SERVICE S, ORDRE R
```

```
WHERE S.nos = R.nos
```

```
UNION
```

```
SELECT nos, intitulé, " " FROM SERVICE
```

```
WHERE NOT EXISTS
```

```
(SELECT * FROM ORDRE
```

```
WHERE SERVICE.nos = ORDRE.nos);
```

Gestion centralisée des commandes et des livraisons

On veut mettre en place une base de données relative aux ventes d'une chaîne de magasins. Les magasins partagent un stock commun d'articles et bénéficient d'une gestion centralisée des commandes et des livraisons.

On dispose de :

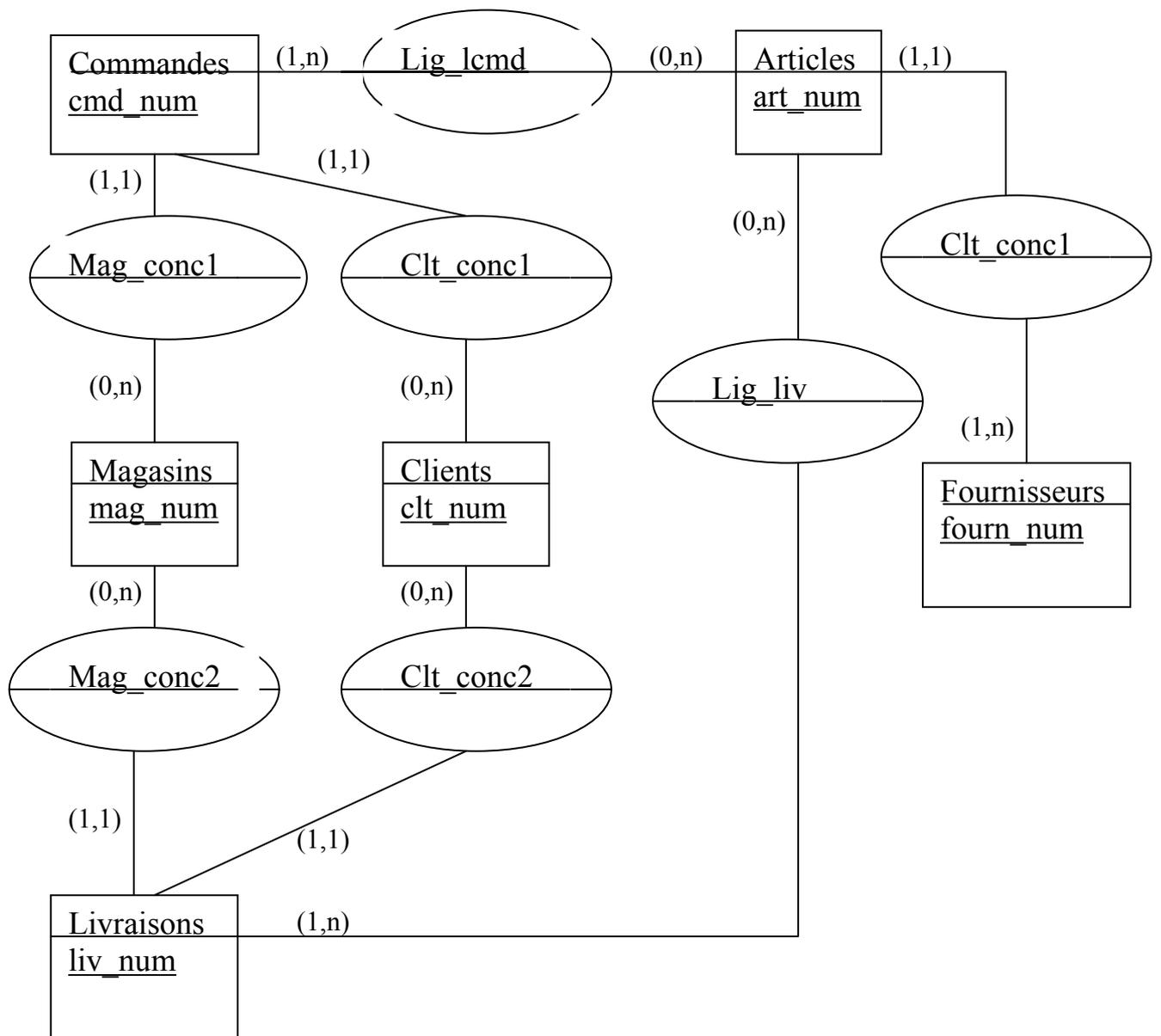
- Une liste des magasins de la chaîne avec pour chacun : un numéro, la localité où est situé le magasin, le nom du gérant.
- Une liste de clients avec un numéro, le nom, le prénom, la nationalité, la localité, le CA global et le type de client qui prend les valeurs 'Particulier', 'Administration', 'Grand compte', 'PME'.
- Une liste des articles commercialisés avec : un numéro, le nom de l'article, le poids(un décimal), la couleur, la quantité en stock, le prix d'achat, le prix de vente.
- Une liste des fournisseurs avec un numéro et le nom.

Une commande est caractérisée par un numéro et une date. Elle lie un client à un magasin et porte sur un ou plusieurs articles.

Une livraison, identifiée par un numéro, est semblable à une commande.

1. Elaborer le modèle E/R.
2. Déduire le modèle relationnel (tables avec noms de colonnes).
3. Créer les tables (mode SQL).
4. Contenu de la base
5. Ecrire les requêtes

Le modèle E/R



Les tables

Magasins(mag_num, mag_loc, mag_ger).

mag_num : numéro magasin, clé primaire

mag_loc : localité du magasin

mag_ger : nom du gérant

Clients(clt_num, clt_nom, clt_pnom, clt_pays, clt_loc, clt_ca, clt_type) .

clt_num : numéro client, clé primaire

clt_nom : nom client

clt_pnom : prénom client

clt_pays : nationalité client

clt_loc : localité du client

clt_ca : CA client

clt_type : type client

Articles(art_num, art_nom, art_poids, art_coul, art_stock, art_pa, art_pv,
art_four).

art_num : numéro article, clé primaire

art_nom : nom article

art_poids : poids article

art_coul : couleur article

art_stock : stock article

art_pa : prix d'achat article

art_pv : prix de vente article

art_four : numéro fournisseur, clé étrangère, correspond à four_num de
Fournisseurs.

Fournisseurs(four_num, four_nom).

four_num : numéro fournisseur, clé primaire

four_nom : nom fournisseur

Commandes(cmd_num, cmd_date, cmd_clt, cmd_mag).

cmd_num : numéro commande, clé primaire

cmd_date : date de la commande

cmd_clt : numéro client, clé étrangère, correspond à clt_num de Clients

cmd_mag : numéro magasin, clé étrangère, correspond à mag_num de
magasin

Lig_cmd(lcd_cmd, lcd_art, lcd_qte, lcd_liv, lcd_pu, lcd_datliv).

lcd_cmd : numéro commande, correspond à cmd_num

lcd_art : numéro article, correspond à art_num

lcd_cmd, lcd_art : clé primaire

lcd_qte : quantité commandée

lcd_liv : quantité livrée

lcd_pu : prix unitaire de vente

lcd_datliv : date de livraison prévue

Livraison(liv_num, liv_date, liv_clt, liv_mag).

liv_num : numéro livraison, clé primaire

liv_date : date de la livraison

liv_clt : numéro client

liv_mag : numéro magasin

Lig_liv(llv_liv, llv_art, llv_qte, llv_cmd)

llv_liv : numéro livraison

llv_art : numéro article

llv_liv, llv_art : clé primaire

llv_qte : quantité livrée

llv_cmd : numéro commande

Création des tables et Contenu de la base

Table Magasins

mag_num	mag_loc	mag_ger
M01	Paris 5	BERTON Louis
M02	Paris 10	JANNEAU Luc
M03	Lyon	MOUILLARD Marcel
M04	Marseille	CAMUS Marius
M05	Montpellier	BAIJOT Marc
M06	Bordeaux	DETIENNE Nicole
M07	Nantes	DUMONT Henri
M08	Tours	DEMARTEAU Renée
M09	Rouen	NOSENT Daniel
M10	Lille	NIZET Jean_Luc
M11	Bruxelles	VANDERSMISSEN Fernand
M12	Liège	HANNEAU Vincent

```
create table magasins
(mag_num char(8) primary key,
mag_loc varchar(25) not null,
mag_ger varchar(25) not null);
```

create table clients
 (clt_num char(8) primary key,
 clt_nom varchar(25) not null,
 clt_pnom varchar(25),
 clt_pays varchar(2) not null,
 clt_loc varchar(20) not null,
 clt_ca integer,
 clt_type varchar(16));

Table Clients

clt_num	clt_nom	clt_pnom	clt_pays	clt_loc	clt_ca	clt_type
C01	DEFRERE	Marc	F	Paris	806	Particulier
C02	DECERF	Jean	F	Paris	0	Particulier
C03	DEFAWE	Léon	B	Liège	0	Particulier
C04	NOSSENT	Serge	B	Bruxelles	149	Particulier
C05	JACOB	Marthe	F	Marseille	31	Administration
C06	JAMAR	Pierre	B	Liège	21	Administration
C07	DECKERS	William	F	Lyon	140	Grand compte
C08	DECLERCQ	Lucien	B	Bruxelles	349	Grand compte
C09	DEFYZ	Maurice	F	Bordeaux	0	Particulier
C10	DEFOOZ	Francis	F	Lille	60	Particulier
C11	RAMJOIE	Victor	F	Nantes	0	Particulier
C12	RENARDY	Jean	F	Marseille	275	Grand compte
C13	MANTEAU	Yvan	F	Paris	320	Grand compte
C14	JONAS	Henri	F	Paris	0	PME
C15	DELVIENNE	Christian	F	Lyon	0	Grand compte
C16	DEFPEZ	André	F	Lyon	0	Particulier

Table Fournisseurs

four_num	four_nom
F01	CATIO ELECTRONIC
F02	LES STYLOS REUNIS
F03	MECANIQUE DE PRECISION
F04	SARL ROULAND
F05	ELECTROLAMP

create table fournisseurs
 (four_num char(8) primary key,
 four_nom varchar(25) not null unique);

Table Articles

create table articles
 (art_num char(8) primary key,
 art_nom varchar(25) not null,
 art_poids numeric,
 art_coul varchar(20),
 art_stock integer,
 art_pa integer not null,
 art_pv integer not null,
 art_four char(8) **references fournisseurs**);
 PS : il faut construire des liens entre les tables

art_num	art_nom	art_poids	art_coul	art_stock	art_pa	art_pv	art_four
A01	AGRAFEUSE	150	ROUGE	10	20	29	F04
A02	CALCULATRICE	150	NOIR	5	200	235	F01
A03	CACHET-DATEUR	100	BLANC	3	21	30	F04
A04	LAMPE	550	ROUGE	3	105	149	F05
A05	LAMPE	550	BLANC	3	105	149	F05
A06	LAMPE	550	BLEU	3	105	149	F05
A07	LAMPE	550	VERT	3	105	149	F05
A08	PESE-LETTRE- 1-500			2	120	200	F03
A09	PESE-LETTRE- 1-1000			2	150	250	F03
A10	CRAYON	20	ROUGE	210	1	2	F02
A11	CRAYON	20	BLEU	190	1	2	F02
A12	CRAYON LUXE	20	ROUGE	95	3	5	F02
A13	CRAYON LUXE	20	VERT	90	3	5	F02
A14	CRAYON LUXE	20	BLEU	80	3	5	F02
A15	CRAYON LUXE	20	NOIR	450	3	5	F02

Tables Commandes

cmd_num	cmd_date	cmd_clt	cmd_mag
C9401	06/06/94	C07	M03
C9402	06/06/94	C06	M12
C9403	06/06/94	C13	M01
C9404	06/06/94	C01	M02
C9405	06/06/94	C08	M11
C9406	07/06/94	C05	M04
C9407	07/06/94	C04	M11
C9408	08/06/94	C03	M12
C9409	08/06/94	C10	M11
C9410	08/06/94	C01	M11
C9411	09/06/94	C12	M01
C9412	10/06/94	C01	M11

create table commandes
 (cmd_num char(8) primary key,
 cmd_date date not null,
 cmd_clt char(8) not null references clients,
 cmd_mag char(8) not null references magasins);

Table Lig_cmd

lcd_cmd	lcd_art	lcd_qte	lcd_liv	lcd_pu	lcd_datliv
C9401	A04	1	1	140	06/06/94
C9402	A10	1	1	2	06/06/94
C9402	A11	2	2	2	06/06/94
C9402	A14	3	3	5	06/06/94
C9403	A02	1	1	230	06/06/94
C9403	A03	2	2	30	06/06/94
C9403	A15	5	5	4	06/06/94
C9403	A14	1	1	5	06/06/94
C9403	A13	1	1	5	06/06/94
C9404	A02	2	0	235	10/06/94
C9404	A12	1	0	5	10/06/94
C9404	A13	10	0	4	10/06/94
C9404	A15	8	0	4	10/06/94
C9404	A05	1	0	149	10/06/94
C9405	A06	1	1	149	06/06/94
C9405	A08	1	1	200	06/06/94
C9406	A10	1	1	2	07/06/94
C9407	A07	1	0	149	10/06/94
C9408	A01	1	1	29	08/06/94
C9409	A12	3	1	5	08/06/94
C9409	A13	3	1	5	08/06/94
C9409	A14	3	1	5	08/06/94
C9409	A15	3	1	5	08/06/94
C9410	A12	8	8	5	08/06/94
C9410	A14	8	8	5	08/06/94
C9411	A09	1	0	250	14/06/94
C9411	A15	5	5	5	09/06/94
C9412	A03	1	0	30	15/06/94

create table lig_cmd
(lcd_cmd char(8) not null,
lcd_art char(8) not null,
lcd_qte integer not null,
lcd_liv integer,
lcd_pu integer not null,
lcd_datliv date,
primary key (lcd_cmd, lcd_art),
foreign key (lcd_cmd) references commandes,
foreign key (lcd_art) references articles);

Table Livraisons

liv_num	liv_date	liv_clt	liv_mag
L9401	06/06/94	C07	M03
L9402	06/06/94	C06	M12
L9403	06/06/94	C13	M01
L9404	06/06/94	C08	M11
L9405	06/06/94	C05	M04
L9406	06/06/94	C03	M12
L9407	08/06/94	C10	M11
L9408	08/06/94	C01	M11
L9409	09/06/94	C12	M01

create table livraisons
(liv_num char(8) primary key,
liv_date date not null,
liv_clt char(8) not null references clients,
liv_mag char(8) not null references magasins);

Table Lig_liv

llv_liv	llv_art	llv_qte	llv_cmd
L9401	A04	1	C9401
L9402	A10	1	C9402
L9402	A11	2	C9402
L9402	A14	3	C9402
L9403	A02	1	C9403
L9403	A03	2	C9403
L9403	A15	5	C9403
L9403	A14	1	C9403
L9403	A13	1	C9403
L9404	A06	1	C9405
L9404	A08	1	C9405
L9405	A10	1	C9406
L9406	A01	1	C9408
L9407	A12	1	C9409
L9407	A13	1	C9409
L9407	A14	1	C9409
L9407	A15	1	C9409
L9408	A12	8	C9410
L9408	A14	8	C9410
L9409	A15	5	C9411

```

create table lig_liv
(llv_liv char(8) not null,
llv_art char(8) not null,
llv_qte integer not null,
llv_cmd char(8) not null,
primary key (llv_liv, llv_art),
foreign key (llv_liv) references livraisons,
foreign key (llv_art) references articles,
foreign key (llv_cmd, llv_art) references lig_cmd);

```

Requêtes

R1 - Tous les noms des fournisseurs.

four_nom
CATIO ELECTRONIC
LES STYLOS REUNIS
MECANIQUE DE PRECISION
SARL ROULAND
ELECTROLAMP

```

SELECT four_nom
FROM fournisseurs;

```

R2 - Liste des localités où habitent les clients.

localité
Bordeaux
Bruxelles
Liège
Lille
Lyon
Marseille
Nantes
Paris

```
SELECT DISTINCT clt_loc AS localité
FROM clients;
```

R3 - Tous les clients habitants à Paris.

clt_num	clt_nom	clt_pnom
C01	DEFRERE	Marc
C02	DECERF	Jean
C13	MANTEAU	Yvan
C14	JONAS	Henri

```
SELECT clt_num, clt_nom, clt_pnom
FROM clients
WHERE clt_loc='Paris';
```

R4 - Tous les articles dont le poids est inférieur à 500.

Art num	art nom	art poids
A01	AGRAFEUSE	150
A02	CALCULATRICE	150
A03	CACHET-DATEUR	100
A10	CRAYON	20
A11	CRAYON	20
A12	CRAYON LUXE	20
A13	CRAYON LUXE	20
A14	CRAYON LUXE	20
A15	CRAYON LUXE	20

```
SELECT art_num, art_nom, art_poids
FROM articles
WHERE art_poids<500;
```

R5 - Tous les articles ayant un prix de vente supérieur ou égal au double du prix d'achat.

Art_num	art_nom	art_pa	art_pv
A10	CRAYON	1	2
A11	CRAYON	1	2

```
SELECT art_num, art_nom, art_pa, art_pv
FROM articles
WHERE art_pv>=2*art_pa;
```

R6 - Tous les articles rouges de poids supérieur à 100.

Art_num	art_nom	art_poids	art_coul
A01	AGRAFEUSE	150	ROUGE
A04	LAMPE	550	ROUGE

```
SELECT art_num, art_nom, art_poids, art_coul
FROM articles
WHERE art_coul='ROUGE' and art_poids>100;
```

R7 - Tous les articles rouges et ceux de poids supérieur à 500.

Art_num	art_nom	art_poids	art_coul
A01	AGRAFEUSE	150	ROUGE
A04	LAMPE	550	ROUGE
A05	LAMPE	550	BLANC
A06	LAMPE	550	BLEU
A07	LAMPE	550	VERT
A10	CRAYON	20	ROUGE
A12	CRAYON LUXE	20	ROUGE

```
SELECT art_num, art_nom, art_poids, art_coul
FROM articles
WHERE art_coul='ROUGE' or art_poids>500;
```

R8 - Tous les articles qui ne sont pas rouges et ceux dont le poids est inférieur ou égal à 500.

Art_num	art_nom	art_poids	art_coul
A02	CALCULATRICE	150	NOIR
A03	CACHET-DATEUR	100	BLANC
A11	CRAYON	20	BLEU
A13	CRAYON LUXE	20	VERT
A14	CRAYON LUXE	20	BLEU
A15	CRAYON LUXE	20	NOIR

```
SELECT art_num, art_nom, art_poids, art_coul
FROM articles
WHERE not(art_coul='ROUGE' or art_poids>500);
```

R9 – Tous les articles dont, soit la couleur est 'ROUGE' et le poids est supérieur à 100, soit la couleur est 'VERT'.

```
SELECT art_num, art_nom, art_poids, art_coul
FROM articles
WHERE (art_coul='ROUGE' and art_poids>100) or art_coul='VERT';
```

Art_num	art_nom	art_poids	art_coul
A01	AGRAFEUSE	150	ROUGE
A04	LAMPE	550	ROUGE
A07	LAMPE	550	VERT
A13	CRAYON LUXE	20	VERT

R10 – Liste des articles dont le prix d’achat est compris entre 100 et 150 francs.

art_num	art_nom	art_pa
A04	LAMPE	105
A05	LAMPE	105
A06	LAMPE	105
A07	LAMPE	105
A08	PESE-LETTRE- 1-500	120
A09	PESE-LETTRE- 1-1000	150

```
SELECT [art_num], [art_nom], [art_pa]
FROM articles
WHERE art_pa between 100 and 150;
```

R11 - Liste des articles de couleur soit ‘ROUGE’ soit ‘NOIR’.

art_num	art_nom	art_coul
A15	CRAYON LUXE	NOIR
A02	CALCULATRICE	NOIR
A12	CRAYON LUXE	ROUGE
A10	CRAYON	ROUGE
A04	LAMPE	ROUGE
A01	AGRAFEUSE	ROUGE

```
SELECT art_num, art_nom, art_coul
FROM articles
WHERE art_coul In ("rouge","noir")
ORDER BY art_coul;
```

Ou

```
SELECT art_num, art_nom, art_coul
FROM articles
WHERE art_coul="rouge"
UNION SELECT art_num, art_nom, art_coul
FROM articles where art_coul="noir"
ORDER BY art_coul;
```

R12 - Clients dont le nom commence par JA.

clt_num	clt_nom	clt_pnom
C05	JACOB	Marthe
C06	JAMAR	Pierre

```
SELECT clt_num, clt_nom, clt_pnom
FROM clients
WHERE clt_nom Like 'JA*';
```

R13 - Clients dont le nom commence par DE et se termine par Z.

clt_num	clt_nom	clt_pnom
C09	DEFYZ	Maurice
C10	DEFOOZ	Francis
C16	DEFPEZ	André

```
SELECT clt_num, clt_nom, clt_pnom
FROM clients
WHERE clt_nom Like 'DE*Z';
```

R14 - Articles dont la couleur n’a pas été saisie

art_num	art_nom	art_coul
A08	PESE-LETTRE- 1-500	
A09	PESE-LETTRE- 1-1000	

```
SELECT art_num, art_nom, art_coul
FROM articles
WHERE art_coul is null;
```

R15 - Articles triés selon l'ordre croissant de leur poids.

art_num	art_nom	art_poids
A09	PESE-LETTRE- 1-1000	
A08	PESE-LETTRE- 1-500	
A15	CRAYON LUXE	20
A14	CRAYON LUXE	20
A13	CRAYON LUXE	20
A12	CRAYON LUXE	20
A11	CRAYON	20
A10	CRAYON	20
A03	CACHET-DATEUR	100
A02	CALCULATRICE	150
A01	AGRAFEUSE	150
A07	LAMPE	550
A06	LAMPE	550
A05	LAMPE	550
A04	LAMPE	550

```
SELECT art_num, art_nom, art_poids
FROM articles
ORDER BY art_poids;
```

R16 - Articles de poids inférieur ou égal à 100, triés selon l'ordre croissant de leur poids, et à poids égal par prix d'achat décroissant.

art_num	art_nom	art_poids	art_pa
A15	CRAYON LUXE	20	3
A14	CRAYON LUXE	20	3
A13	CRAYON LUXE	20	3
A12	CRAYON LUXE	20	3
A11	CRAYON	20	1
A10	CRAYON	20	1
A03	CACHET-DATEUR	100	21

```
SELECT art_num, art_nom, art_poids, art_pa
FROM articles
WHERE art_poids<=100
ORDER BY art_poids, art_pa DESC;
```

R17 – Marge bénéficiaire des articles par ordre croissant de celle-ci.

```
SELECT art_num, art_nom, art_pa, art_pv, art_pv-art_pa AS marge
FROM articles
ORDER BY 5; // 5 veut dire 5em champ ou colonne)
```

art_num	art_nom	art_pa	art_pv	marge
A11	CRAYON	1	2	1
A10	CRAYON	1	2	1
A15	CRAYON LUXE	3	5	2
A14	CRAYON LUXE	3	5	2
A13	CRAYON LUXE	3	5	2
A12	CRAYON LUXE	3	5	2
A03	CACHET-DATEUR	21	30	9
A01	AGRAFEUSE	20	29	9
A02	CALCULATRICE	200	235	35
A07	LAMPE	105	149	44
A06	LAMPE	105	149	44
A05	LAMPE	105	149	44
A04	LAMPE	105	149	44
A08	PESE-LETTRE- 1-500	120	200	80
A09	PESE-LETTRE- 1-1000	150	250	100

R18 – Calculer le poids moyen des articles.

poids_moyen
209,230769230769

```
SELECT avg(art_poids) AS poids_moyen
FROM articles;
```

R19 – Rechercher le prix de l'article le plus cher.

prix_max
250

```
SELECT max(art_pv) AS prix_max
FROM articles;
```

R20 – Calculer le poids moyen, la marge maximum, pour les articles dont la couleur est définie

```
SELECT avg(art_poids) AS poids_moyen, max(art_pv-art_pa) AS marge_max
FROM articles
WHERE art_coul is not null;
```

poids_moyen	marge_max
209,230769230769	44

R21 – Calculer le prix de vente moyen de chaque couleur d'articles.

art_coul	prix_moyen
	225
BLANC	89,5
BLEU	52
NOIR	120
ROUGE	46,25
VERT	77

```

SELECT art_coul, avg(art_pv) AS prix_moyen
FROM articles
GROUP BY art_coul
ORDER BY art_coul; // AS est un alias pour créer un champ
// virtuelle pour pouvoir afficher les
// resultants voulues

```

R21_bis – Exclure les articles dont le prix d'achat est inférieur à 10.

art_coul	prix_moyen
	225
BLANC	89,5
BLEU	149
NOIR	235
ROUGE	89
VERT	149

```

SELECT art_coul, avg(art_pv) AS prix_moyen
FROM articles
WHERE art_pa >= 10
GROUP BY art_coul
ORDER BY art_coul;

```

R21_bisbis

art_coul	prix_moyen
	225
BLEU	149
NOIR	235
VERT	149

```

SELECT art_coul, avg(art_pv) AS prix_moyen
FROM articles
WHERE art_pa >= 10
GROUP BY art_coul
HAVING avg(art_pv) >= 100
ORDER BY art_coul;

```

R22 – Couleur des articles telle que le prix de vente moyen des articles de la couleur soit supérieur à 70.

art_coul	prix_moyen
	225
BLANC	89,5
NOIR	120
VERT	77

```

SELECT art_coul, avg(art_pv) AS prix_moyen
FROM articles
GROUP BY art_coul
HAVING avg(art_pv) > 70
ORDER BY art_coul;

```

Format de date dans une requête SQL : #date#

Exemple : Commandes passées le 06/06/94

SELECT cmd_num, cmd_clt, cmd_date

FROM commandes

WHERE cmd_date=#06/06/94#;

cmd_num	cmd_clt	cmd_date
C9401	C07	06/06/94
C9402	C06	06/06/94
C9403	C13	06/06/94
C9404	C01	06/06/94
C9405	C08	06/06/94

Fonctions sur dates et fonctions sur chaînes

YEAR(date) : retourne l'année de date

MONTH(date) : retourne le mois de date

DAY(date) : retourne le jour de date

LTRIM(chaîne) : supprime les blancs à gauche de chaîne

RTRIM(chaîne) : supprime les blancs à droite de chaîne

R23 – Magasins qui ont réalisé plus d'une vente dans la période du 06/06/94 au 07/06/94.

cmd_mag	nb_ventes
M01	2
M11	5
M12	2

```
SELECT cmd_mag, count(cmd_mag) AS nb_ventes
FROM commandes
WHERE cmd_nom between#06/06/94# and #07/06/94#
GROUP BY cmd_mag
HAVING count(cmd_mag)>1;
```

Ou

Ou

cmd_mag	mag_loc	nb_ventes
M01	Paris 5	2
M11	Bruxelles	5
M12	Liège	2

```
SELECT cmd_mag, mag_loc, count(cmd_mag) AS nb_ventes
FROM commandes, Magasins
WHERE cmd_nom between#06/06/94# and #07/06/94# and
cmd_mag=mag_num
GROUP BY cmd_mag, mag_loc
HAVING count(cmd_mag)>1;
```

R24 – Clients ayant commandé le 06/06/94.

clt_num	clt_nom	date_commande
C07	DECKERS	06/06/94
C08	DECLERCQ	06/06/94
C01	DEFRERE	06/06/94
C06	JAMAR	06/06/94
C13	MANTEAU	06/06/94

```
SELECT clt_num, clt_nom, cmd_nom AS Date_Commande
FROM commandes, clients
WHERE cmd_nom=#06/06/94# and cmd_clt=clt_num
ORDER BY clt_nom;
```

R25 – Rechercher le numéro, le nom, le poids, le nom du fournisseur des articles rouges.

```
SELECT art_num, art_nom, art_poids, four_nom
FROM articles, fournisseurs
WHERE art_coul='ROUGE' and art_four=four_num;
```

art_num	art_nom	art_poids	four_nom
A01	AGRAFEUSE	150	SARL ROULAND
A04	LAMPE	550	ELECTROLAMP
A10	CRAYON	20	LES STYLOS REUNIS
A12	CRAYON LUXE	20	LES STYLOS REUNIS

R26 – Liste des articles dont le prix d'achat est supérieur au prix de l'article 'A08'.

```
SELECT a1.art_num AS Numéro, a1.art_nom AS Libellé, a1.art_pa
AS Prix_achat, a2.art_num AS A08, a2.art_pa AS Achat_A08
FROM articles AS a1, articles AS a2
WHERE a1.art_pa>a2.art_pa and a2.art_num='A08';
```

Numéro	Libellé	Prix_achat	A08	Achat_A08
A02	CALCULATRICE	200	A08	120
A09	PESE-LETTRE- 1-1000	150	A08	120

Niveau de Select, **AS** est numéro de colonne
Niveau de From, **AS** est une table

R27 – Liste des clients qui n'ont rien commandé.

clt_num	clt_nom
C02	DECERF
C09	DEFYZ
C11	RAMJOIE
C14	JONAS
C15	DELVIENNE
C16	DEFPEZ

```
SELECT [clt_num], [clt_nom]
FROM clients
WHERE clt_num not in (select cmd_clt from commandes);
```

R28 – Numéro et nom des gérants des magasins qui ont vendu au moins un article ‘A02’.

mag_num	mag_ger
M01	BERTON Louis
M02	JANNEAU Luc

```

SELECT mag_num, mag_ger
FROM magasins, commandes
WHERE exists (select lcd_cmd from lig_cmd where
cmd_num=lcd_cmd and lcd_art='A02')
and mag_num=cmd_mag; // Exits sert à faire une recherche
// exists (select .... from .... where

```

R29 – Liste des articles dont le prix de vente est supérieur au prix de vente de l’article de couleur blanche le moins cher.

art_num	art_nom	art_pv
A02	CALCULATRICE	235
A04	LAMPE	149
A05	LAMPE	149
A06	LAMPE	149
A07	LAMPE	149
A08	PESE-LETTRE- 1-500	200
A09	PESE-LETTRE- 1-1000	250

```

SELECT art_num, art_nom, art_pv
FROM articles
WHERE art_pv >
(select min(art_pv) from articles where art_coul ='BLANC');

```

R30 – Liste des articles dont la **somme** des ventes est supérieure à la moyenne des ventes de tous les articles. (ventes = lcd_pu*lcd_qte)

art_num	art_nom	somme
A02	CALCULATRICE	700
A03	CACHET-DATEUR	90
A04	LAMPE	140
A05	LAMPE	149
A06	LAMPE	149
A07	LAMPE	149
A08	PESE-LETTRE- 1-500	200
A09	PESE-LETTRE- 1-1000	250
A15	CRAYON LUXE	92

```

SELECT art_num, art_nom, sum(lcd_pu*lcd_qte) AS somme
FROM articles, lig_cmd
WHERE art_num=lcd_art
GROUP BY art_num, art_nom
HAVING sum(lcd_pu*lcd_qte)>(select avg(lcd_pu*lcd_qte)
from lig_cmd);

```

↓
 Pour chaque avg ou sum ,
 il faut un « Select »

R31 – Liste des articles rouges et ceux vendus dans un magasin de Paris.

art_num	art_nom	indicateur
A01	AGRAFEUSE	ROUGE
A04	LAMPE	ROUGE
A10	CRAYON	ROUGE
A12	CRAYON LUXE	ROUGE
A02	CALCULATRICE	Paris
A03	CACHET-DATEUR	Paris
A05	LAMPE	Paris
A09	PESE-LETTRE- 1-1000	Paris
A12	CRAYON LUXE	Paris
A13	CRAYON LUXE	Paris
A14	CRAYON LUXE	Paris
A15	CRAYON LUXE	Paris

```

SELECT art_num, art_nom, 'ROUGE' as indicateur
FROM articles
where art_coul='ROUGE'
UNION select art_num, art_nom, 'Paris' as indicateur
from articles, lig_cmd, commandes, magasins
where art_num=lcd_art and lcd_cmd=cmd_num
and cmd_mag=mag_num and mag_loc like 'Paris*'
ORDER BY 3 DESC;

```

R32 – Liste des articles rouges parmi ceux vendus dans un magasin de Paris

art_num	art_nom
A12	CRAYON LUXE

```

SELECT art_num, art_nom
FROM articles
WHERE art_coul='ROUGE' and art_num in
(select art_num
from articles, lig_cmd, commandes, magasins
where art_num=lcd_art and lcd_cmd=cmd_num and cmd_mag=mag_num and
mag_loc like 'Paris*');

```

Données

N° Employé	Nom	Qualification	Service	N° Chef
E01	Bernard	Q1	SE1	E01
E02	Dupont	Q1	SE1	E01
E03	Eyraud	Q1	SE2	E08
E04	Fayard	Q2	SE1	E02
E05	Guillot	Q2	SE8	E15
E06	Henri	Q3	SE8	E05
E07	Kamel	Q3	SE8	E05
E08	Lemaire	Q1	SE2	E01
E09	Martins	Q2	SE8	E15
E10	Peyron	Q2	SE3	E18
E11	Jeanne	Q2	SE3	E18
E12	Pierre	Q3	SE7	E27
E13	Jacques	Q3	SE7	E27
E14	Martine	Q3	SE7	E27
E15	Gilles	Q1	SE8	E01
E16	Luc	Q3	SE8	E09
E17	Alcide	Q3	SE8	E09
E18	Pierrette	Q1	SE3	E01
E19	Nicolas	Q2	SE6	E33
E20	Justin	Q3	SE8	E09
E21	Justine	Q3	SE3	E10
E22	Jeanne	Q1	SE4	E01
E23	Christian	Q2	SE4	E42
E24	Serje	Q2	SE5	E29
E25	Flora	Q3	SE7	E27
E26	Nabil	Q1	SE7	E46
E27	Patrick	Q2	SE7	E26
E28	Daniel	Q3	SE6	E30
E29	Morgon	Q1	SE5	E01
E30	Baptiste	Q2	SE6	E33
E31	Quillot	Q2	SE4	E42
E32	Dejardin	Q1	SE2	E08
E33	Leroy	Q1	SE6	E39
E34	Damien	Q3	SE6	E19
E35	Coulon	Q3	SE6	E19
E36	Dassault	Q1	SE2	E04
E37	Munoz	Q2	SE1	E02
E38	Maganinho	Q3	SE6	E19
E39	Bertelino	Q1	SE6	E01
E40	Seyrasse	Q3	SE6	E19
E41	Vermillon	Q3	SE6	E30
E42	Petit	Q1	SE4	E22
E43	Grosjean	Q3	SE6	E30
E44	Legros	Q3	SE6	E30
E45	Pfeiffer	Q3	SE6	E30
E46	Dellia	Q1	SE7	E01
E47	Puteau	Q3	SE5	E48
E48	Delbosse	Q2	SE5	E29

Qualification

Qualification	Salaire Hor.
Q1	100
Q2	70
Q3	40

Site

N° Site	Nom Site
SI1	Clermont
SI2	Aubière
SI3	Romagnat

Service

N° Service	Intitulé	Localisation	N° Chef
SE1	Direction	Clermont	E01
SE2	Informatique	Clermont	E08
SE3	Personnel	Clermont	E18
SE4	Contrôle Gestion	Aubière	E22
SE5	Achats	Aubière	E29
SE6	Production	Romagnat	E39
SE7	Logistique	Romagnat	E46
SE8	Maintenance	Romagnat	E15

Entreprise

N° Entreprise	Nom Entreprise
EN1	Unilog
EN2	Cognos
EN3	Datalog
EN4	Mathos
EN5	Mecabis
EN6	Equip

Travail

N° Projet	N° Entreprise	Montant
PR01	EN2	500000
PR01	EN3	300000
PR02	EN1	300000
PR02	EN4	350000
PR03	EN5	450000
PR03	EN6	500000
PR04	EN1	450000
PR04	EN2	450000

E49	Laporte	Q2	SE2	E03
-----	---------	----	-----	-----

N° Projet	Intitulé	Date début	Date fin	Service
PR01	EIS	07/11/97	06/03/98	SE2
PR02	ORDO	02/10/97	10/10/98	SE7
PR03	DEKA7	05/09/97	04/02/98	SE6
PR04	ERP	07/11/97	06/05/98	SE2

Implication

N° Projet	N° Employé	Durée
PR01	E03	600
PR01	E32	1000
PR02	E25	1800
PR02	E26	1800
PR02	E27	1800
PR02	E46	1800
PR03	E08	100
PR03	E28	600
PR03	E30	600
PR03	E33	500
PR03	E41	600
PR03	E43	600
PR03	E44	600
PR03	E45	600
PR04	E03	600
PR04	E36	1000
PR04	E49	1000

R1 Liste de tous les employes

NuEmp	NomEmp	NumQual	NumSer	NumChef
E01	Bernard	Q1	SE1	E01
E02	Dupont	Q1	SE1	E01
E03	Eyrond	Q1	SE2	E08
E04	Fayard	Q2	SE1	E02
E05	Guillot	Q2	SE8	E15
E06	Henri	Q3	SE8	E05
E07	Kamel	Q3	SE8	E05
E08	Lemaire	Q1	SE2	E01
E09	Martins	Q2	SE8	E15
E10	Peyron	Q2	SE3	E18
E11	Jeanne	Q2	SE3	E18
E12	Pierre	Q3	SE7	E27
E13	Jacques	Q3	SE7	E27
E14	Martine	Q3	SE7	E27
E15	Gilles	Q1	SE8	E01
E16	Luc	Q3	SE8	E09
E17	Alcide	Q3	SE8	E09
E18	Pierrette	Q1	SE3	E01
E19	Nicolas	Q2	SE6	E33
E20	Justin	Q3	SE8	E09
E21	Justine	Q3	SE3	E10
E22	Jeanne	Q1	SE4	E01
E23	Christian	Q2	SE4	E42
E24	Serje	Q2	SE5	E29
E25	Flora	Q3	SE7	E27
E26	Nabil	Q1	SE7	E46
E27	Patrick	Q2	SE7	E26
E28	Daniel	Q3	SE6	E30
E29	Morgon	Q1	SE5	E01
E30	Baptiste	Q2	SE6	E33
E31	Quillot	Q2	SE4	E42
E32	Dejardin	Q1	SE2	E08
E33	Leroy	Q1	SE6	E39
E34	Damien	Q3	SE6	E19
E35	Coulon	Q3	SE6	E19
E36	Dassault	Q1	SE2	E04
E37	Munoz	Q2	SE1	E02
E38	Maganinho	Q3	SE6	E19
E39	Bertelino	Q1	SE6	E01
E40	Seyrasse	Q3	SE6	E19
E41	Vermillon	Q3	SE6	E30
E42	Petit	Q1	SE4	E22
E43	Grosjean	Q3	SE6	E30
E44	Legros	Q3	SE6	E30
E45	Pfeiffer	Q3	SE6	E30
E46	Dellia	Q1	SE7	E01
E47	Puteau	Q3	SE5	E48

```
SELECT *
FROM employe;
```

R2 Liste des employés de qualification Q1

```
SELECT [NomEmp], [NumQual]
FROM employe
WHERE NumQual="Q1";
```

NomEmp	NumQual
Bernard	Q1
Dupont	Q1
Eyrond	Q1
Lemaire	Q1
Gilles	Q1
Pierrette	Q1
Jeanne	Q1
Nabil	Q1
Morgon	Q1
Dejardin	Q1
Leroy	Q1
Dassault	Q1
Bertelino	Q1
Petit	Q1
Dellia	Q1

NuEmp	NomEmp	NumQual	NumSer	NumChef
E48	Delbosse	Q2	SE5	E29
E49	Laporte	Q2	SE2	E03

R3 Nom et salaire horaire du directeur

Directeur	Salaire
Bernard	100

```
SELECT NomEmp AS Directeur, Salaire
FROM employe, qualification
WHERE NuEmp=NumChef and
employe.NumQual=Qualification.NumQual;
```

PS : Il faut trouver un lien entre la table « employe »
et « Qualification » c'est « NumQual ». Après, il faut bien définir l'adresse exact de cet adresse comme
« nom_de_table.nom_de_champ » exemple : employe.NumQual

R4 Pour chaque service, donner le numéro matricule et le nom du chef

NumSer	NumChef	NomEmp
SE1	E01	Bernard
SE2	E08	Lemaire
SE3	E18	Pierrette
SE4	E22	Jeanne
SE5	E29	Morgon
SE6	E39	Bertelino
SE7	E46	Dellia
SE8	E15	Gilles

```
SELECT s.NumSer, s.NumChef, e.NomEmp
FROM employe as e, Service as s
WHERE e.NuEmp=s.NumChef;
```

R5 A quel site se trouve chaque employé ?

NomEmp	NomSite
Alcide	Romagnat
Alcide	Aubière
Alcide	Clermont
Baptiste	Romagnat
Baptiste	Aubière
Baptiste	Clermont
Bernard	Clermont
Bernard	Aubière
Bernard	Romagnat
Bertelino	Clermont
Bertelino	Aubière
Bertelino	Romagnat
Christian	Aubière
Christian	Clermont
Christian	Romagnat
Coulon	Clermont
Coulon	Aubière

```
SELECT e.NomEmp, si.NomSite
FROM employe AS e, Site AS si, Service AS s
WHERE e.NumSer=s.NumSer and si.NumSite=si.NumSite
order by e.NomEmp;
```

PS : Dans ce cas, li faut créer des tables virtuelles avec AS et il faut aussi trouver des champs communs dans les tables concernées

PS : Je n'ai pas copié la liste entière (dans cette page) qu'on obtient avec cette requête.

Tables de Q 3
Employe

numemp	nomemp	numqual	numser	numchef
E01	Bernard	Q1	SE1	E01
E02	Dupont	Q1	SE1	E01
E03	Eyrond	Q1	SE2	E08
E04	Fayard	Q2	SE1	E02
E05	Guillot	Q2	SE8	E15
E06	Henri	Q3	SE8	E05
E07	Kamel	Q3	SE8	E05
E08	Lemaire	Q1	SE2	E01
E09	Martins	Q2	SE8	E15
E10	Peyron	Q2	SE3	E18
E11	Jeanne	Q2	SE3	E18
E12	Pierre	Q3	SE7	E27
E13	Jacques	Q3	SE7	E27
E14	Martine	Q3	SE7	E27
E15	Gilles	Q1	SE8	E01
E16	Luc	Q3	SE8	E09
E17	Alcide	Q3	SE8	E09
E18	Pierrette	Q1	SE3	E01
E19	Nicolas	Q2	SE6	E33
E20	Justin	Q3	SE8	E09
E21	Justine	Q3	SE3	E10
E22	Jeanne	Q1	SE4	E01
E23	Christian	Q2	SE4	E42
E24	Serje	Q2	SE5	E29
E25	Flora	Q3	SE7	E27
E26	Nabil	Q1	SE7	E46
E27	Patrick	Q2	SE7	E26
E28	Daniel	Q3	SE6	E30
E29	Morgon	Q1	SE5	E01
E30	Baptiste	Q2	SE6	E33
E31	Quillot	Q2	SE4	E42
E32	Dejardin	Q1	SE2	E08
E33	Leroy	Q1	SE6	E39
E34	Damien	Q3	SE6	E19
E35	Coulon	Q3	SE6	E19
E36	Dassault	Q1	SE2	E04
E37	Munoz	Q2	SE1	E02
E38	Maganinho	Q3	SE6	E19
E39	Bertelino	Q1	SE6	E01
E40	Seyrasse	Q3	SE6	E19
E41	Vermillon	Q3	SE6	E30
E42	Petit	Q1	SE4	E22
E43	Grosjean	Q3	SE6	E30
E44	Legros	Q3	SE6	E30
E45	Pfeiffer	Q3	SE6	E30
E46	Dellia	Q1	SE7	E01

Entreprise

numentr	nomentr
EN1	Unilog
EN2	Cognos
EN3	Datalog
EN4	Mathos
EN5	Mecabis
EN6	Equip

Implication

numproj	numentr	montant
PR01	EN2	500000
PR01	EN3	300000
PR02	EN1	300000
PR02	EN4	350000
PR03	EN5	450000
PR03	EN6	500000
PR04	EN1	450000
PR04	EN2	450000

Qualification

numqual	salaire
Q1	100
Q2	70
Q3	40

Site

numsite	nomsite
S11	Clermont
S12	Aubière
S13	Romagnat

Travail

numproj	numemp	duree
PR01	E03	600
PR01	E32	1000
PR02	E25	1800
PR02	E26	1800
PR02	E27	1800
PR02	E46	1800
PR03	E08	100
PR03	E28	600
PR03	E30	600
PR03	E33	500
PR03	E41	600
PR03	E43	600
PR03	E44	600
PR03	E45	600
PR04	E03	600
PR04	E36	1000
PR04	E49	1000

numemp	nomemp	numqual	numser	numchef
E47	Puteau	Q3	SE5	E48
E48	Delbosse	Q2	SE5	E29
E49	Laporte	Q2	SE2	E03

Projet

numproj	nomproj	datedeb	datefin	numser
PR01	EIS	07/11/1997	06/03/1998	SE2
PR02	ORDO	02/10/1997	10/10/1998	SE7
PR03	DEKA7	05/09/1997	04/02/1998	SE6
PR04	ERP	07/11/1997	06/05/1998	SE2

Service

numser	nomser	numsite	numchef
SE1	Direction	SI1	E01
SE2	Informatique	SI1	E08
SE3	Personnel	SI1	E18
SE4	Contrôle Gestion	SI2	E22
SE5	Achats	SI2	E29
SE6	Production	SI3	E39
SE7	Logistique	SI3	E46
SE8	Maintenance	SI3	E15

Réponse 1)

Producteurs (Num-producteur, Genre, Nom, prenom, Ad-rue, CP, Ville, Pays)

Connaître (Num-producteur, Num-films, statuts_producteur)

Films(Num-films, Num-real, Titre, An_sortie, Genre, Pays, Coulr)

Realisateurs(Num-real, Nom, Prenom)

Interprete(Num-inter, Nom, Prenom)

Assure(Num-inter, Num-films, Role)

Réponse 2)

Q1)

```
SELECT numclient, nomclient, categorie
FROM Clients
WHERE nomclient Like 'DU*';
```

Q2)

```
SELECT numproduit, libelle, prixbrut AS Bon-Marché
FROM Produits
WHERE prixbrut <50
order by prixbrut;
```

Q3)

```
SELECT C.numclient, P. numproduit
FROM Clients AS C, Produits AS P, Livraisons AS L
WHERE P.numproduit=L.numproduit
ORDER BY C.numclient ;
```

Q4)

```
SELECT C.nomclient, K.categorie
FROM Clients AS C, Categorie AS K
WHERE C.categorie= K.categorie
ORDER BY C.nomclient ;
```

R) 3

R1

PS. Si on veut aussi afficher le nom de chef

nomemp
Bernard
Bertelino
Dellia
Dupont
Gilles
Jeanne
Lemaire
Morgon
Pierrette

```
SELECT [nomemp]
FROM Employe
WHERE numchef='E01'
ORDER BY [nomemp];
```

Si on ne veut pas faire afficher le nom directeur

nomemp
Bertelino
Dellia
Dupont
Gilles
Jeanne
Lemaire
Morgon
Pierrette

```
SELECT nomemp
FROM Employe
where numchef= 'E01' and not(numemp='E01')
order by nomemp;
```