

## Exonet 30 version 1.2

Propriétés	Description
<b>Intitulé long</b>	Pratique d'interrogation de base de données.
<b>Formation concernée</b>	Section de TS Service Informatique aux Organisations 1ère année. (Éventuellement classe de terminale Sciences et technologies de la gestion (STG))
<b>Matière</b>	Conception et maintenance de solutions applicatives.
<b>Présentation</b>	A travers une base de données relationnelle, effectuer des requêtes de manipulation puis de définition dans le cadre de l'extension de cette base. Approche progressive avec aide adaptée pour travail en autonomie.
<b>Notions</b>	Base de données, clauses SQL, manipulation de données.
<b>Transversalité</b>	Inscrit dans tout processus de traitement de l'information.
<b>Pré-requis</b>	Notions élémentaires liées au système de gestion de bases de données relationnelles,
<b>Outils</b>	Access, Postgresql (autre SGBD en adaptant le script Postgresql)
<b>Mots-clés</b>	SQL, PostgreSQL, Access
<b>Durée</b>	2 à 4H si mise en pratique.
<b>Auteur</b>	Fabio PASQUALINI
<b>Version</b>	1.3
<b>Date de publication</b>	10 avril 2011

### Contexte de travail

Vous travaillez dans le cadre d'un commerce de sucrerie (ce qui vous demande une grande volonté afin de ne pas entamer les stocks !). Il existe une base contenant des tables dont la description figure en annexe. La base est réalisée sous Access (fichier joint) mais peut-être transposée sur d'autres SGBDR. Le SQL utilisé est celui fourni avec Access, de faibles modifications permettent de l'utiliser avec d'autres SGBDR. Un script de création de la base sous PostgreSQL est fourni en fichier joint.

Suite à la description du travail à réaliser, des pistes sont données afin de mener à bien les requêtes qui vous semblent complexes. Ces pistes ne sont à utiliser qu'en dernier recours : après avoir essayé de trouver en s'aidant de ses connaissances, après avoir consulté l'aide en ligne mais sans avoir demandé à votre voisin (où est le plaisir de la copie quand on connaît le plaisir de la découverte !).

Il est possible de travailler directement sur Access ou de travailler sur feuille en TD. En cas de réalisation pratique, il faut toujours débiter par ouvrir sa requête en mode 'SQL' : le mode 'création' ne permet pas l'écriture d'un code SQL standard : le résultat fourni alors est loin du résultat escompté.

Toute ressemblance avec des personnes existantes ne pourraient être que le fruit du plus pur hasard. Les sucreries sont soit des produits courants soit de pures inventions.

## Questions

Il s'agit d'écrire les requêtes permettant de réaliser les listes ou opérations qui suivent.

1 - Clients (nom, adresse, téléphone) dont la deuxième lettre est A, la quatrième I et la cinquième N.

2 - Numéros des commandes passées par Mme RABIN (\*).

3 - Liste des articles (Désignation) achetés par Mme RABIN (\*).

Je sais que le Montant HT des commandes inclut le transport.

4 - Montant HT (total) des commandes de Mme RABIN (\*).

5 - Montant HT (total) des commandes de Mme RABIN (\*) hors transport (calculé d'après le prix unitaire hors taxes des articles achetés).

6 - Chiffre d'affaires total pour le mois d'octobre 2008.

7 - Chiffre d'affaires de chaque vendeur pour le mois d'octobre 2008 (Nom du vendeur, CA).

8 - Liste des chiffres d'affaires quotidiens supérieurs à 300€ (date, CA journalier) d'après les dates de commande.

9 - Liste des produits figurant au moins dans deux commandes classée du plus sollicité au moins sollicité (désignation produit, nombre de fois que ce produit figure dans une commande.)

10 - Créer une table « commission » qui contiendrait un code, un libellé de cette commission, un pourcentage entier qui correspondrait à la commission versée.

11 - Que doit-on modifier dans la base pour que cette table permette d'allouer une commission et une seule à chaque vendeur suivant le code commission, indiquer les requêtes permettant de réaliser cet objectif (champ supplémentaire, relation, index).

Les articles sont vendus en conditionnement d'un poids défini (par exemple 500 grammes) ou en conditionnement contenant un certain nombre de pièces (exemple : sachet de 40 carambars), cette information figure dans le champ 'Quantité'. Pour distinguer les types de conditionnement on indique G ou P dans le champ 'Descriptif'. Dans ce dernier cas, on renseigne la colonne 'Poids pièce' avec le poids d'une pièce. Dans les autres cas, on renseigne le poids pièce à la valeur 0 pour pouvoir effectuer des calculs sur toutes les lignes : un calcul effectué avec une valeur non renseignée ne renvoie rien !

12 - Prix au kilogramme de chaque article classé du moins coûteux au plus coûteux (désignation, prix au kilogramme).

13 - Poids total du stock en kilogramme.

(\*) : On considère que le nom RABIN n'existe qu'une seule fois.

## Aide

A n'utiliser qu'en cas de difficultés !!!!

Suivent des conseils pour aider à la réalisation des requêtes demandées. Ces conseils ne sont à consulter qu'en cas de difficultés.

1 – Il faut utiliser des caractères génériques : on utilise <#> en SQL standard pour remplacer n'importe quelle chaîne de caractère ( <\*> sous Access) et <\_> pour remplacer n'importe quel caractère ( <?> sous Access). Attention ces caractères ne sont supportés que par l'opérateur LIKE !

2 – Attention de ne pas utiliser de tables inutiles. Essayer une requête imbriquée.

3 – Là encore, on peut faire un énorme produit cartésien et des jointures ou utiliser des requêtes imbriquées. Essayer de trouver la solution par étapes : d'abord le code client de Mme RABIN, puis les commandes qu'elle a passées, etc. Vous aurez besoin de l'opérateur IN, à bien distinguer de = (votre requête imbriquée renvoie-t-elle une ligne ou plusieurs ?).

4 – On a besoin de « client » et « commande ».

5 – Il faut utiliser, entre autres, la table article qui est la seule à contenir le prix unitaire hors taxes. Vous aurez besoin de multiplier puis d'effectuer une somme (fonction de regroupement). Un produit cartésien est inévitable.

6 – Les dates sous Access s'encadrent du signe <#> : #10/10/2008#. Vous effectuez des opérations de comparaison sur les dates à peu près de la même façon que sur les nombres. Vous pouvez aussi utiliser des fonctions spécifiques aux dates. Attention, les dates sont parfois au format américain MM/JJ/AA. Il faut utiliser une fonction de regroupement.

7 – Il faut faire DES regroupements. Attention, la clause GROUP BY vient toujours après la clause WHERE.

8 – La clause HAVING sera utile.

9 – La clause ORDER BY vient en dernier. On peut remplacer le nom de la colonne triée par son numéro d'ordre : ORDER BY 3.

10 – On peut ajouter une contrainte directement sur le champ désiré (NOT NULL, PRIMARY KEY). Il faut nommer cette contrainte.

11 – Commencer par ajouter un nouveau champ (dans quelle table ?). Puis établir la relation en indiquant que l'on utilise une clé étrangère. Éventuellement on peut aussi créer un index sur ce champ.

12 et 13 – Il faut utiliser ici une " formule " qui permet d'effectuer deux calculs distincts suivant les deux cas qui se présentent : valeur P ou G du Descriptif. Il faut savoir que (a < 10) renvoie la valeur 0 si c'est faux et -1 si c'est vrai. On peut alors utiliser l'opérateur de comparaison comme n'importe quel opérateur qui renvoie une valeur numérique. Si on le souhaite, on peut mettre en facteurs la formule.

## Annexes

### TABLE client

code_c	nom	adresse	cp	ville	telephone
17	TARINAUX Lucien	12 rue de la Justice	51100	REIMS	03.26.25.48.87
46	MARTUSE	103 avenue Lear	51100	REIMS	03.26.03.25.26
47	RABIN Sandrine	21 rue de la Méditerranée	51100	REIMS	03.26.14.15.25
48	SILLARD Laurence	15 rue Pasantiers	51100	REIMS	03.26.11.11.25
49	COTOY Sylvie	12 rue des écus	51100	REIMS	03.26.10.25.75
50	HELLOU Bernard	21 rue de la Méditerranée	51100	REIMS	03.26.12.25.42
51	HENTION Martine	50 allée des bons enfants	51100	REIMS	03.26.12.25.86
52	SIBAT Evelyne	14 rue de la Baltique	51100	REIMS	03.26.12.23.33
53	MARIN Dominique	24 rue de la Baltique	51100	REIMS	03.26.10.10.23
54	DURDUX Monique	15 allée des Béarnais	51150	VITRY LE FRANCOIS	03.26.42.42.33
55	CANILLE Walter	14 rue Lanterneau	51100	REIMS	03.26.12.12.87
56	Antoinette	1, rue de la Méditerranée	51140	ROMAIN	03.26.78.89.54
57	GAUTON Nadine	5 place des Oiseaux	51200	FISMES	03.26.53.56.55
58	LEGROS Christian	18 place des Oiseaux	51200	FISMES	03.26.44.55.66
59	DUMOITIERS Lucille	12 place Centrale	02320	LONGUEVAL	03.26.86.43.25
60	BOUCHE Carole	4, rue Brulé	51200	FISMES	03.26.33.96.85

### TABLE commande

numero	code_v	code_c	date_livraison	date_commande	total_ht	total_tva	etat
10178	15	47	05/09/08	05/09/08	177.00	9.75	1
10179	15	47	13/10/08	13/10/08	192.00	10.50	1
10180	15	48	10/10/08	10/10/08	98.00	5.40	1
10181	15	49	11/10/08	11/10/08	175.00	9.60	1
10182	15	50	11/10/08	11/10/08	116.00	6.40	1
10183	15	51	11/10/08	11/10/08	118.00	6.50	1
10184	15	52	12/10/08	12/10/08	102.00	5.60	1
10185	15	53	12/10/08	12/10/08	19.00	1.05	1
10186	15	54	10/10/08	10/10/08	101.00	5.55	1
10187	15	55	10/10/08	10/10/08	65.00	3.58	1
10188	17	56	12/10/08	12/10/08	121.00	6.66	1
10189	17	57	10/10/08	10/10/08	110.00	6.05	1
10190	17	58	13/10/08	13/10/08	123.00	6.76	1
10191	17	59	13/10/08	13/10/08	107.50	5.91	1
10192	17	60	10/11/08	10/11/08	237.00	13.04	1

### TABLE ligne\_commande

numero	numero_ligne	reference	quantite_demandee
10188	3	1004	1
10179	7	1007	2
10191	4	1015	1
10191	1	1016	1
10188	1	1016	2
10186	1	1016	1
10189	1	1017	1
10192	3	1017	1
10186	2	3002	2
10192	9	3004	1

numero	numero_ligne	reference	quantite_demandee
10192	4	3016	1
10190	1	3010	1
10180	3	3016	1
10192	6	3017	1
10179	4	4002	1
10185	1	4002	1
10181	4	4002	2
10184	3	4004	1
10178	5	4004	1
10187	2	4010	1
10187	3	4011	1
10182	2	4012	1
10181	5	4012	2
10183	5	4013	1
10191	3	4013	1
10178	1	4015	1
10187	1	4015	1
10190	2	4015	1
10189	2	4016	1
10192	10	4019	1
10181	1	4020	1
10192	7	4022	1
10183	1	4025	1
10178	2	4025	1
10192	1	4025	1
10184	1	4025	1
10181	2	4026	1
10183	2	4027	1
10183	3	4029	1
10192	2	4030	1
10179	1	4031	1
10178	3	4031	1
10189	3	4031	1
10184	2	4031	2
10192	5	4031	1
10179	2	4032	1
10190	3	4032	1
10189	4	4033	1
10182	1	4034	1
10178	4	4036	1
10179	3	4037	1
10183	4	4039	1
10178	7	4042	1
10181	3	4045	1
10191	2	4052	1
10192	8	4052	1
10188	2	4052	1
10178	6	4053	1
10180	1	4053	1
10190	4	4053	1
10179	5	4054	1
10181	6	4054	1
10182	3	4055	1
10180	2	4055	1
10179	6	4057	1
10182	4	4057	1

**TABLE produit**

reference	designation	quantite	descriptif	prix_unitaire_ht	stock	poids_piece
1004	FEU DE JOIE LIQUEUR ASSORT.	500	G	23.00	50	0
1007	TENDRE FRUIT	500	G	18.00	120	0
1015	CARACAO	500	G	24.50	50	0
1016	COKTAIL	500	G	33.00	40	0
1017	ORFIN	500	G	32.00	40	0
3002	CARRE PECTO	500	G	29.00	40	0
3004	ZAN ALESAN	25	P	15.00	50	20
3010	PATES GRISES	500	G	35.00	100	0
3016	CAMEL AU LAIT	500	G	20.00	100	0
3017	VIOLETTE TRADITION	500	G	25.00	100	0
4002	SUCETTE BOULE FRUIT	25	P	14.00	100	40
4004	SUCETTE BOULE POP	25	P	21.00	50	40
4010	CARAMBAR	40	P	18.00	20	15
4011	CARANOUGA	40	P	18.00	100	15
4012	CARAMBAR FRUIT	40	P	18.00	100	15
4013	CARAMBAR COLA	40	P	18.00	50	15
4015	SOURIS REGLISSE	500	G	24.00	50	0
4016	SOURIS CHOCO	500	G	24.00	50	0
4019	SCHTROUMPFS VERTS	500	G	24.00	50	0
4020	CROCODILE	500	G	21.00	50	0
4022	PERSICA	500	G	28.00	20	0
4025	COLA CITRIQUE	500	G	21.00	50	0
4026	COLA LISSE	500	G	25.00	50	0
4027	BANANE	1000	G	23.00	20	0
4029	OEUF SUR LE PLAT	500	G	25.00	20	0
4030	FRAISIBUS	500	G	25.00	50	0
4031	FRAISE TSOIN-TSOIN	500	G	25.00	40	0
4032	METRE REGLISSE ROULE	500	G	19.00	50	0
4033	MAXI COCOBAT	1000	G	19.00	20	0
4034	DENTS VAMPIRE	500	G	22.00	50	0
4036	LANGUE COLA CITRIQUE	500	G	21.00	40	0
4037	OURSON CANDI	1000	G	21.00	50	0
4039	SERPENT ACIDULE	500	G	21.00	20	0
4042	TETINE CANDI	500	G	20.00	40	0
4045	COLLIER PECCOS	15	P	21.00	50	50
4052	TWIST ASSORTIS	500	G	22.00	50	0
4053	OURSON GUIMAUVE	500	G	35.00	10	0
4054	BOULE COCO MULER	500	G	34.00	10	0
4055	COCOMALLOW	500	G	33.00	10	0
4057	CRIC-CRAC	500	G	33.00	10	0

**TABLE vendeur**

code_v	nom	adresse	cp	ville	telephone
15	FILLARD Sylvain	77 rue du l'Adriatique	51100	REIMS	03.26.12.25.25
17	BAUDOT Marc	16 rue de Reims	51000	CHALONS EN CHAMPAGNE	03.26.10.58.59

## Corrigé

La correction tient compte de la syntaxe utilisée par Access.

### Question 1

```
SELECT nom, adresse,
       cp, ville, telephone
FROM client
WHERE nom LIKE "?A?IN*";
```

### Question 2

```
SELECT numero
FROM commande
WHERE code_c =
      (SELECT code_c
       FROM client
        WHERE nom LIKE "RABIN *");
```

### Question 3

```
SELECT designation AS [Articles achetés par Mme RABIN]
FROM produit
WHERE reference IN
      (SELECT reference
       FROM ligne_commande
        WHERE numero IN
          (SELECT numero
           FROM commande
            WHERE code_c =
              (SELECT code_c
               FROM client
                WHERE nom LIKE "RABIN *")));
```

### Question 4

```
SELECT Sum(total_ht) AS [Montant HT Mme RABIN]
FROM commande
WHERE commande.code_c=
      (SELECT code_c
       FROM client
        WHERE nom LIKE "RABIN *");
```

### Question 5

```
SELECT Sum(quantite_demandee*prix_unitaire_HT) AS [Montant HT Mme RABIN]
FROM ligne_commande, produit
WHERE produit.reference=ligne_commande.reference
AND numero IN
      (SELECT numero
       FROM commande
        WHERE code_c=
          (SELECT code_c
           FROM client
            WHERE nom Like "RABIN *"));
```

### Question 6

```
SELECT SUM(total_ht) AS [Total octobre 2008]
FROM commande
WHERE date_commande Between #01/10/2008# And #31/10/2008#;
```

### Question 7

```
SELECT nom, SUM(total_ht) AS [Total octobre]
FROM commande, vendeur
WHERE DatePart("m",date_commande)=10
AND DatePart("y",date_commande)=2008
AND commande.code_v=vendeur.code_v
GROUP BY nom;
```

### Question 8

```
SELECT date_commande,
Sum(total_ht) AS [CA journalier supérieur à 300€]
FROM commande
GROUP BY date_commande
HAVING Sum(total_ht)>300;
```

### Question 9

```
SELECT designation, Count(ligne_commande.reference)
FROM ligne_commande, produit
WHERE produit.reference=ligne_commande.reference
GROUP BY designation
HAVING Count(ligne_commande.reference)>=2
ORDER BY 2 DESC;
```

### Question 10

```
CREATE TABLE commission
(
code_commission      INTEGER  CONSTRAINT pk_code_com PRIMARY KEY,
libelle              VARCHAR(30),
pourcentage          INTEGER
);
```

### Question 11

```
ALTER TABLE vendeur
ADD COLUMN code_commission INTEGER
CONSTRAINT nn_comm NOT NULL;
```

*Les lignes déjà existantes contenant une valeur nulle ne satisferont pas à la contrainte, toutefois, la contrainte ne s'appliquant qu'au cours de l'enregistrement d'une ligne, les valeurs nulles déjà existantes seront acceptées.*

```
ALTER TABLE vendeur
ADD CONSTRAINT fk_comm
FOREIGN KEY (code_commission)
REFERENCES commission (code_commission);
```

*On pourra vérifier la création de cette contrainte dans la fenêtre 'relations'.*

```
CREATE INDEX idx_comm
ON vendeur (code_commission)
WITH IGNORE NULL;
```

*La clause IGNORE NULL permet de ne pas tenir compte des valeurs nulles dans l'index rendant ainsi obligatoire de mettre à jour les lignes non respectueuses de la contrainte NOT NULL.*

**Question 12**

```
SELECT designation,  
       ((-1000*prix_unitaire_HT) / quantite) / (((descriptif="G") + ((descriptif="P") * poids_piece))  
       AS [Prix au kilo]  
       FROM produit  
       ORDER BY 2;
```

*Il faut s'assurer qu'aucune division par zéro ne se fera.*

**Question 13**

```
SELECT SUM((((descriptif="G")*quantite)  
          + ((descriptif="P")*quantite*poids_piece))*Stock) / 1000  
       AS [Poids total en kg]  
       FROM produit;
```

*Il est préférable de diviser par 1000 à la fin du traitement.*