TP1 Bases de données TIS3

**Thème :** Environnement Unix + Oracle. Base de donnée « Le grand bazar ».

Démarrer sur les machines

**étape 1 : se connecter au serveur im2ag-oracle.e.ujf-grenoble.fr**

- choisir un PC, le redémarrer et demander une session LINUX (sauf s'il est déjà en mode linux)

- connectez-vous avec votre login\_agalan/motdepasse\_agalan (cf compte à Polytech)

- ouvrez un terminal, et connectez vous à la machine oracle :

ssh -X indiquez im2ag-oracle.e.ujf-grenoble.fr -l votre\_login

**étape 2 :**

tapez *env*

vérifier que la variable ORACLE\_SID vaut bien im2ag

si ce n'est pas le cas tapez : *setenv ORACLE\_SID im2ag*

vérifiez en tapant *env* que la variable a bien changé.

**étape 3 : récupérer le fichier des tables et des données**

- connectez-vous à Sql :

*sqlplus login\_agalan / mot de passe agalan*

- chargez la base, pour cela tapez :

*start /tmp/relations1\_bazar2020.txt*

*start /tmp/base\_bazar2020.txt*

- vérifiez que les données ont bien été chargées :

*select \* from produit;* (vous devez voir 51 produits)

*select \* from vente;* (248 ventes)

*select \* from achat ;* (252 achats).

**étape 4 : ouvrir un éditeur de texte pour saisir vos requêtes**

Pensez à régulièrement enregistrer votre fichier

Copiez/collez les résultats de vos requêtes dans ce fichier, vous aurez ainsi la trace de votre travail

Vous êtes maintenant opérationnels avec une fenêtre sur sql, et l'autre pour faire de la saisie des requêtes et de la récupération de résultats

Le TP

**Schéma de Données**:

On rappelle les trois tables du Grand Bazar :

**PRODUIT**(NPRO **integer** **primary** **key** , NOMP **char**(10), QTS **integer**, COULEUR **char**(10));

Donne pour chaque produit identifié par son numéro NPRO, son nom NOMP, sa quantité en stock QTS et sa couleur.

**VENTE**(NVEN **integer** **primary** **key**, NOMC **char**(10), NPRV **integer**, QTV **integer**, DATEV date);

Décrit toutes les ventes de produit réalisées. Pour chaque vente, identifiée par son numéro NVEN, on conserve le nom du client NOMC, le numéro du produit vendu NPRV, la quantité QTV de produit vendu, et la date DATEV de la vente.

**ACHAT**(NACH **integer** **primary** **key**, NOMF **char**(10), NPRA **integer**, QTA **integer**, DATEA date);

Décrit les commandes de produits aux fournisseurs. Pour chaque achat, identifié NACH, on donne le nom du fournisseur NOMF, le numéro du produit commandé NPRA, la quantité QTA commandée, la date DATEA de l'achat (de la commande).

Questions :

**Il faut enregistrer vos requêtes ainsi que les réponses à vos requêtes**

**Pour chaque requête, nous vous indiquons les éléments permettant de 'vérifier' votre réponse. Attention, cette vérification est nécessaire, mais n'assure pas que votre réponse est correcte.**

**Le TP doit être réalisé en 2h.**

**Si vous avez peur de manquer de temps, faites la requête 1 puis les requêtes paires.**

1 - Donner les noms et couleurs de tous les produits par ordre alphabétique inverse. ->50 nuplets

*On utilisera l'ordre de classement order by que l'on met en fin de requête :*

*select distinct qts from produit*

*order by qts -> les quantités en stocks seront affichés en ordre croissant*

*select distinct qts from produit*

*order by qts desc -> les quantités en stocks seront affichés en ordre décroissant*

*select distinct couleur, qts from produit*

*order by couleur, qts -> les couleurs et les quantités en stock de chaque produit sont imprimés en ordre croissant de npro, puis par qts (pour une même couleur)*

*select distinct couleur, qts from produit*

2 - Donner les numéros de vente, le nom du client, la quantité vendue pour les ventes du produit ‘mouchoir’, réalisées avant le '10-SEP-19' (ce format de date est le format par défaut d'oracle) -> 10 n-uplets

3- Quelle est la moyenne des quantités en stock des produits vendus pendant le dernier semestre?

Pour cette requête, on utilisera

* la variable *sysdate* qui donne la date du jour

exemple : select \* from vente where datev< sysdate

* la fonction *sql add\_months(date, nbdemois)* qui ajoute le nombre de mois à la date indiqué. Si nbdemois est négatif, le nombre de mois sera retranché.

exemples : select \* from vente where datev< add\_months('20-OCT-13', 1)

select \* from vente where datev< add\_months('20-OCT-13', -1)

On donnera 2 réponses à cette requête :

* 1 réponse imbriquée, -> 37.75
* 1 réponse non imbriquée -> 37.4615385

On constate donc que ces 2 requêtes ne donnent pas la même réponse. Ceci est dû au fait que la réponse imbriquée traite les données de façon ensembliste (elle calcule sur 12 produits), et pas la requête non imbriquée (qui, elle, traite 13 n-uplets dont 2 avec le produit 12)

4 - Nombre de produits fournis par le fournisseur de nom xuong ? -> 11

5 - Donner les noms des clients ayant acheté au moins un produit de couleur verte par ordre alphabétique. Ecrire avec une requête imbriquée puis avec un opérateur ensembliste. -> 7 rows selected.

6 - Donner le nom des fournisseurs avec lesquels aucune commande de produit n'a été réalisée depuis le 09/10/2015. Le résultat sera trié dans l’ordre alphabétique. -> 1 rows selected.

7 – Quel est le nom du produit que l’on a le plus en stock ? On écrira 2 versions de cette requête : avec des blocs imbriqués, avec des sous-requêtes. -> tasse

8 – Donner pour chaque fournisseur la liste des produits verts dont la quantité achetée est supérieure à la quantité moyenne vendue de l’ensemble des produits verts. On écrira 2 versions de cette requête : avec des blocs imbriqués, avec des sous-requêtes. -> 9 rows selected.

9 – Combien de produits de chaque couleur ? le résultat sera trié par ordre décroissant de la quantité. -> 14 rows selected.

10 – Combien de produits distincts de couleur verte achète chaque client ? le résultat sera trié par ordre croissant des noms de clients. -> 7 rows selected.

11 – Combien de produits de chaque couleur ? On affichera uniquement les couleurs où 2 produits distincts au moins sont disponibles. Le résultat sera trié par ordre décroissant de la quantité. -> 10 rows selected.

12 – Combien de produits de couleur verte achète chaque client ? On affichera uniquement les clients ayant acheté au moins 2 produits distincts de couleur verte. Le résultat sera trié par ordre croissant des noms de clients. -> 4 rows selected.

On écrira 2 versions de la requête (1 imbriquée, l’autre non)

13 – Quel est le plus gros client ? -> poutou

On écrira 2 versions de la requête.

14 – Quel est le client qui achète le moins souvent ? -> thureau

On écrira 2 versions de la requête.

15 - quelle est la couleur de produit la plus vendue (en nombre de produits) ? -> bleu

On écrira 2 versions de la requête.

*16 - Pour chaque fournisseur, donner le produit dont la quantité achetée est la plus élevée -> 15 rows selected.*

On écrira 2 versions de la requête.

*17 - Quels sont les fournisseurs de tous les produits de couleur bleue ? -> 3 rows selected.*

On écrira 2 versions de la requête.

Réécrire toutes les requêtes avec :

PRODUIT(NPRO, NOMP, QTS, COULEUR);

VENTE(NVEN, NOMC, NPRO, QTV, DATEV);

ACHAT(NACH, NOMF, NPRO, QTA, DATEA);

Pour cela :

- sous sql, chargez la base, pour cela tapez :

*start /tmp/relations2\_bazar2020.txt*

*start /tmp/base\_bazar2020.txt*

- vérifiez que les données ont bien été chargées :

*select \* from produit;* (vous devez voir 51 produits)

*select \* from vente;* (248 ventes)

*select \* from achat ;* (252 achats).

Réécrire toutes les requêtes avec :

PRODUIT(NPRO, NOM, QT, COULEUR);

VENTE(NVEN, NOM, NPRO, QT, DA);

ACHAT(NACH, NOM, NPRO, QT, DA);

Pour cela :

- sous sql, chargez la base, pour cela tapez :

*start /tmp/relations3\_bazar2020.txt*

*start /tmp/base\_bazar2020.txt*

- vérifiez que les données ont bien été chargées :

*select \* from produit;* (vous devez voir 51 produits)

*select \* from vente;* (248 ventes)

*select \* from achat ;* (252 achats).