Aide-mémoire TP Partie II

Via les TP guidés de la Partie II, vous avez appris certaines nouvelles choses sur la syntaxe SQL. Pour faciliter la révision, vous trouvez ci-dessous un petit résumé. Cependant, une révision complète des 4 dernières séances de TP est recommandée pour réussir le CC3 de la partie II.

1. Créer une table à partir d'une requête SELECT (réf. TP1 Question Q3)
   CREATE TABLE <nom_table> AS SELECT …

2. GROUP BY … HAVING … (voir le mini-cours avec le TP2)

3. Fonctions d'agrégat
   x étant donné un attribut d'une relation :
   AVG(x) donne la moyenne des valeurs de x,
   COUNT(x) donne le nombre de tuples ayant une valeur sur x,
   MAX(x) donne la valeur maximum de x,
   MIN(x) donne la valeur minimum de x,
   SUM(x) donne la somme des valeurs de x,
   COUNT(*) donne le nombre de tuples

   Remarque : Toutes ces fonctions ignorent les valeurs NULL sauf COUNT(*).

4. Jointure
   Nous avons vu deux façons d'écrire une requête SQL pour effectuer des jointures durant le TP2 (réf. Question Q14).
   Forme procédurale (ou requête imbriquée) : nous mettons une seule relation dans la clause FROM de chaque sous-requête ; le résultat d'une sous-requête est utilisé par un prédicat dans la clause WHERE d'une autre sous-requête. Ex.
   SELECT Nom
   FROM Buveur
   WHERE NumB IN (SELECT NumB FROM ABU WHERE NumV=12);
   Forme relationnelle : nous mettons toutes les relations consultées dans la clause FROM, et les conditions de jointures dans la clause WHERE. Ex.
   SELECT Nom
   FROM Buveur, Abu
   WHERE Buveur.NumB=Abu.NumB AND NumV=12;

   Bien évidemment, il y a encore d'autres manières pour faire des jointures. Sans précision dans la question, vous être libre de choisir la manière. Dans ce cas-là, le plus important est que votre requête permet de répondre correctement à la question posée.

5. Union
   Oracle utilise le mot clé « *UNION* » entre deux sous-requêtes pour faire une union.
   Exemple : liste des buveurs habitant Toulouse ou Pau.

*SELECT * FROM Buveur*
WHERE Ville='Toulouse'
UNION
*SELECT * FROM Buveur*
WHERE Ville='Pau';

6. Différence
   Oracle utilise le mot clé « *MINUS* » entre deux sous-requêtes pour faire une différence.
   Exemple : liste des crus qui n'ont pas de millésime 2015.
   SELECT CRU FROM Vin
   MINUS
   SELECT CRU FROM Vin
   WHERE millésime=2015;

7. Intersection
   Oracle utilise le mot clé « *INTERSECT* » entre deux sous-requêtes pour faire une intersection.
   Exemple : liste des crus ayant la millésime 2015 et également la millésime 2017.
   SELECT CRU FROM Vin
   WHERE millésime=2015
   INTERSECT
   SELECT CRU FROM Vin
   WHERE millésime=2017;

8. Attribut ROWNUM (ref. TP2 bis)
   Pour chaque tuple renvoyé par une requête, la pseudo-colonne ROWNUM renvoie un nombre indiquant l'ordre dans lequel Oracle sélectionne le tuple dans une relation ou un résultat de jointure. Le premier tuple sélectionné a un ROWNUM de 1, le deuxième a 2, et ainsi de suite.
   Vous pouvez utiliser ROWNUM pour limiter le nombre de tuples renvoyés par une requête, comme dans cet exemple :
   *SELECT * FROM Pilote WHERE ROWNUM < =3;*
   Les conditions testant les valeurs ROWNUM supérieures à un entier positif (ou égale à un entier supérieur à 1) sont toujours fausses. Par exemple, cette requête ne renvoie aucun tuple :
   *SELECT * FROM Pilote WHERE ROWNUM > 1 ;*
   Le premier tuple extrait se voit attribuer un ROWNUM de 1 et rend la condition fausse. Le deuxième tuple à extraire est maintenant le premier et se voit également attribuer un ROWNUM de 1 et rend la condition fausse. Tous les tuples échouent par la suite à satisfaire la condition, donc aucun tuple n'est renvoyé.

9. Création de vue (ref. TP3)
   CREATE OR REPLACE VIEW <nom_vue> (attribut 1, attribut 2, ...)
   AS SELECT ... ;