

Requetes SQL

Ordre d'interpretation des requetes

SQL interprète les requêtes dans l'ordre suivant :

1. FROM: SQL va d'abord chercher les tables à partir desquelles les données seront récupérées. Cela peut impliquer l'ouverture de fichiers de données ou l'accès à des données stockées en mémoire.
2. WHERE: une fois que les tables sont sélectionnées, SQL applique les conditions de filtre spécifiées dans la clause WHERE pour sélectionner les enregistrements qui répondent aux exigences de la requête. Cela peut impliquer l'utilisation d'opérateurs logiques tels que AND et OR, ainsi que des opérateurs de comparaison tels que =, <, >, etc.
3. GROUP BY: si la requête inclut une clause GROUP BY, SQL regroupe les enregistrements en fonction des valeurs communes dans les colonnes spécifiées.
4. HAVING: si la requête inclut une clause HAVING, SQL filtre les groupes résultants en fonction des conditions spécifiées pour sélectionner uniquement les groupes qui répondent à certaines exigences.
5. SELECT: une fois que les enregistrements sont sélectionnés et regroupés (si nécessaire), SQL récupère les colonnes de données spécifiées dans la clause SELECT.
6. ORDER BY: si la requête inclut une clause ORDER BY, SQL trie les résultats en fonction des valeurs des colonnes spécifiées.
7. LIMIT: si la requête inclut une clause LIMIT, SQL limite le nombre d'enregistrements renvoyés dans les résultats.

Requêtes de base

- Insertion de données

```
INSERT INTO eleve  
VALUES (1, 'David', 'Bowie', '1947-01-08', 6);
```

- Simple Select du nom et prénom

```
SELECT nom, prenom FROM data_scientist.eleve;
```

- Insertion du reste de la table

```
INSERT INTO eleve (nom, prenom, date_naissance, classe_id)  
VALUES ('Elton', 'John', '1947-03-25', 6),  
('Cartman', 'Eric', '1989-07-01', 7),
```

```
( 'Norton', 'Edward', '1959-08-18', 5),  
( 'Nicks', 'Stevie', '1948-05-26', 4),  
( 'Kilmister', 'Lemmy', '1945-12-24', 3);
```

- Modifier un champs dans une table

```
ALTER TABLE eleve  
CHANGE date_Naissance date_naissance DATE NOT NULL;
```

- Mettre à jour la table

```
UPDATE eleve  
SET nom= 'Bowie', prenom= 'David'  
WHERE nom like 'David'
```

- Selectionner et trier en ascendant

```
SELECT nom, prenom, date_naissance FROM data_scientist.eleve ORDER BY  
date_naissance ASC;
```

- Création de vues

```
CREATE VIEW 'Vue_eleves' AS  
SELECT nom AS nom, prenom AS Prenom, classes.niveau AS Classe  
FROM eleves  
LEFT JOIN classes  
ON classes_id = classe_id.id
```

Exercice BEER

1. Quels sont les tickets qui comportent l'article d'ID 500, afficher le numéro de ticket uniquement ?

```
select numero_ticket from beer.ventes where id_article ="500"
```

2. Afficher les tickets du 15/01/2014

```
SELECT numero_ticket, date_vente FROM beer.ticket WHERE date_vente = "2014-  
01-15";
```

3. Afficher les tickets émis du 15/01/2014 et le 17/01/2014

```
SELECT numero_ticket, date_vente FROM beer.ticket WHERE date_vente = "2014-01-15" OR DATE_VENTE = "2014-01-17";
```

4. Editer la liste des articles apparaissant à 50 et plus exemplaires sur un ticket

```
SELECT beer.ventes.QUANTITE, beer.article.NOM_ARTICLE from beer.ventes  
INNER JOIN beer.article  
ON beer.ventes.ID_ARTICLE = beer.article.ID_ARTICLE  
WHERE beer.ventes.QUANTITE >= 50
```

5. Quelles sont les tickets émis au mois de mars 2014

```
SELECT ticket.NUMERO_TICKET, DATE_VENTE from beer.ticket  
WHERE DATE_VENTE LIKE '%2014-03%';
```

Correction avec les fonctions de DATE

```
SELECT NUMERO_TICKET, DATE_VENTE  
FROM ticket  
WHERE ANNEE = 2014 AND MONTH(DATE_VENTE) LIKE 03;
```

6. Quelles sont les tickets émis entre les mois de mars et avril 2014 ?

```
SELECT ticket.NUMERO_TICKET, DATE_VENTE from beer.ticket  
WHERE DATE_VENTE >= '2014-03-01' and DATE_VENTE <= '2014-04-30';
```

7. Quelles sont les tickets émis au mois de mars et juin 2014 ?

```
SELECT ticket.NUMERO_TICKET, DATE_VENTE from beer.ticket  
WHERE DATE_VENTE LIKE '%2014-03%' OR DATE_VENTE LIKE '%2014-06%';
```

Correction avec les fonctions de DATE

```
SELECT *  
FROM ticket  
WHERE ANNEE = 2014  
AND MONTH(DATE_VENTE) IN(3,6);
```

8. Afficher la liste des bières classée par couleur. (Afficher l'id et le nom)

```
SELECT article.ID_ARTICLE, article.NOM_ARTICLE
FROM article
INNER JOIN beer.couleur
    ON article.ID_COULEUR = couleur.ID_COULEUR
ORDER BY article.ID_COULEUR ASC
```

9. Afficher la liste des bières n'ayant pas de couleur. (Afficher l'id et le nom)

```
SELECT article.ID_ARTICLE, article.NOM_ARTICLE
FROM article
LEFT JOIN beer.couleur
    ON article.ID_COULEUR = couleur.ID_COULEUR
WHERE article.ID_Couleur is null
```

10. Lister pour chaque ticket la quantité totale d'articles vendus. (Classer par quantité décroissante)

```
SELECT SUM(ventes.QUANTITE), NUMERO_TICKET
FROM beer.ventes
GROUP BY ventes.NUMERO_TICKET
```

11. Lister chaque ticket pour lequel la quantité totale d'articles vendus est supérieure à 500 (Classer par quantité décroissante)

```
SELECT NUMERO_TICKET
FROM beer.ventes
WHERE SUM(ventes.QUANTITE) > 500
GROUP BY ventes.NUMERO_TICKET
```

12. Lister chaque ticket pour lequel la quantité totale d'articles vendus est supérieure à 500. On exclura du total, les ventes ayant une quantité inférieur à 50 (classer par quantité décroissante)

```
SELECT NUMERO_TICKET, SUM(ventes.QUANTITE)
FROM beer.ventes
WHERE QUANTITE < 50
GROUP BY ventes.NUMERO_TICKET
HAVING SUM(ventes.QUANTITE) > 500
ORDER BY SUM(ventes.QUANTITE) asc;
```

13. Lister les bières de type 'Trappiste'. (id, nom de la bière, volume et titrage)

```
SELECT article.ID_ARTICLE, article.ID_TYPE, article.VOLUME,  
article.NOM_ARTICLE, article.titrage  
FROM article  
INNER JOIN beer.type  
ON article.ID_TYPE = beer.type.ID_TYPE  
WHERE article.ID_TYPE = 13;
```

14. Lister les marques de bières du continent 'Afrique'

```
SELECT NOM_MARQUE  
FROM marque  
INNER JOIN pays  
ON marque.ID_PAYS = pays.ID_PAYS  
WHERE pays.ID_CONTINENT = 1
```

15. Lister les bières du continent 'Afrique'

```
SELECT DISTINCT NOM_ARTICLE  
FROM article  
INNER JOIN marque  
ON marque.ID_MARQUE = article.ID_MARQUE  
INNER JOIN pays  
on pays.ID_PAYS = marque.ID_PAYS  
WHERE pays.ID_CONTINENT = 1;
```

16. Lister les tickets (année, numéro de ticket, montant total payé). En sachant que le prix de vente est égal au prix d'achat augmenté de 15% et que l'on n'est pas assujetti à la TVA.

```
SELECT ANNEE, ventes.NUMERO_TICKET, SUM(ventes.QUANTITE *  
article.PRIX_ACHAT) *1.15 AS Montant_total_payé  
FROM ventes  
INNER JOIN article  
ON article.ID_ARTICLE = ventes.ID_ARTICLE  
GROUP BY ANNEE , ventes.NUMERO_ticket;
```

17. Donner le C.A. par année.

```
SELECT SUM(ventes.QUANTITE*article.PRIX_ACHAT), ANNEE  
FROM ventes  
INNER JOIN article  
ON article.ID_ARTICLE = ventes.ID_ARTICLE  
group by ANNEE;
```

18. Lister les quantités vendues de chaque article pour l'année 2016.

```
SELECT SUM(ventes.QUANTITE) AS QT, article.NOM_ARTICLE, ARTICLE.ID_ARTICLE
FROM ventes
INNER JOIN article
    ON article.ID_ARTICLE = ventes.ID_ARTICLE
WHERE ANNEE = '2016'
GROUP BY NOM_ARTICLE, ANNEE, ARTICLE.ID_ARTICLE
```

19. Lister les quantités vendues de chaque article pour les années 2014,2015 ,2016.

```
SELECT SUM(ventes.QUANTITE) AS QT, article.NOM_ARTICLE, ARTICLE.ID_ARTICLE,
ANNEE
FROM ventes
INNER JOIN article
    ON article.ID_ARTICLE = ventes.ID_ARTICLE
WHERE ANNEE in (2014,2015,2016)
GROUP BY NOM_ARTICLE, ARTICLE.ID_ARTICLE, ANNEE
```

20. Lister les articles qui n'ont fait l'objet d'aucune vente en 2014.

```
SELECT ID_ARTICLE, NOM_ARTICLE
FROM article
WHERE (SELECT SUM(QUANTITE)
      FROM ventes
      WHERE ID_ARTICLE = article.ID_ARTICLE AND ANNEE = 2014) IS NULL;
```

21. Coder de 3 manières différentes la requête suivante : Lister les pays qui fabriquent des bières de type 'Trappiste'.

- Sous requete
- Distinct
- Group By

```
SELECT nom_pays
FROM PAYS
WHERE ID_PAYS in (
    SELECT ID_PAYS
    FROM marque
    WHERE ID_MARQUE in (
        SELECT ID_MARQUE
        FROM article
        WHERE ID_TYPE in (
            SELECT ID_TYPE
```

```
FROM type
WHERE ID_TYPE = 13))));
```

22. Lister les tickets sur lesquels apparaissent un des articles apparaissant aussi sur le ticket 2014-856 (le ticket 856 de l'année 2014).

```
SELECT DISTINCT NUMERO_TICKET, ANNEE
FROM ventes
WHERE ID_ARTICLE IN (
    SELECT ID_ARTICLE
    FROM VENTES
    WHERE NUMERO_TICKET LIKE '856' AND ANNEE LIKE '2014');
```

23. Lister les articles ayant un degré d'alcool plus élevé que la plus forte des trappistes.

24. Editer les quantités vendues pour chaque couleur en 2014.

25. Donner pour chaque fabricant, le nombre de tickets sur lesquels apparait un de ses produits en 2014.

26. Donner l'ID, le nom, le volume et la quantité vendue des 20 articles les plus vendus en 2016.

27. Donner l'ID, le nom, le volume et la quantité vendue des 5 'Trappistes' les plus vendus en 2016.

28. Donner l'ID, le nom, le volume et les quantités vendues en 2015 et 2016, des bières dont les ventes ont été stables. (Moins de 1% de variation)

29. Lister les types de bières suivant l'évolution de leurs ventes entre 2015 et 2016. Classer le résultat par ordre décroissant des performances.

30. Existe-t-il des tickets sans vente ?

31. Lister les produits vendus en 2016 dans des quantités jusqu'à -15% des quantités de l'article le plus vendu.