Bases de données - Requêtes SQL

Définition d'une requête

Une requête SQL permet de retourner un ou plusieurs attributs d'une table. Voir un ou plusieurs calculs fait à partir des champs d'attributs d'une table. La sélection des attributs se fait avec le mot clef « SELECT » et le choix de la table dans laquelle on fait la sélection se fait avec le mot clef « FROM ».

Donc toute requête SQL comporte au moins un « SELECT » et un « FROM ».

Remarques : Les requêtes écrites ci-dessous utilisent l'exemple de DB ci-dessus.

Requêtes simples

Sélection de tous les champs d'un ou plusieurs attributs

Sélection de tous les champs des attributs « Nom » et « Prenom » dans la table « Formateurs ».

SELECT *Nom,Prenom* FROM *Formateurs*

Cela retourne un tableau avec les étiquettes des attributs qui seront « Nom » et « Prenom ». On peut si on le souhaite renommer ces attributs (Changer les étiquettes) avec le mot clef « AS » :

SELECT *Nom* AS *'Nom de famille'*, *Prenom* AS *Prénom*
FROM *Formateurs*

Sélection de certains champs d'un ou plusieurs attributs avec une ou plusieurs conditions

Sélection des champs des attributs « Batiment », « Etage » et « Id » dans la table « Salles » mais uniquement pour les salles ayant une capacité de 12 stagiaires : mot clé : « WHERE ».

SELECT *Batiment,Etage,Id* FROM *Salles* WHERE *Capacite=12*

Pour le complémentaire on peut utiliser le mot clé « NOT ». Exemple pour la même requête

SELECT *Batiment,Etage,Id* FROM *Salles* WHERE NOT *Capacite<>12*

Types de conditions : Egalité : **=** Différence : **<>** ⚠ différent de Python : « == » et « != »

Supérieur à (strictement) : **>** Inférieur à (strictement) : **<**

Supérieur ou égal à : **>=** Inférieur ou égal à : **<=**

La condition peut aussi être l'appartenance (ou non) à une liste : **IN** ou : **NOT IN**

Liste des noms, prénoms et numéros de téléphone des formateurs qui ont un homonyme dans la liste des stagiaires. La liste est définie à l'aide d'une sous requête écrite entre parenthèses. Pour que ce soit bien le doublet dont on teste la présence dans la liste, il est mis entre parenthèses.

SELECT *Nom,Prenom,Tel* FROM *Formateurs*
WHERE (*Nom,Prenom*) IN (SELECT *Nom,prenom* FROM *Etudiants*)

Pour plusieurs conditions, on utilise les mots clé « AND » et « OR » suivant l'algèbre booléenne.

Exemple : Liste des salles (« Batiment », « Etage » et « N° de salle ») pouvant accueillir au moins 20 stagiaires et qui de plus ont une imprimante ou un vidéo-projecteur.

SELECT *Batiment,Etage,Id* AS *'N° de salle'* FROM *Salles*
WHERE *Capacite>=20* AND (*Imprimante* OR *Video_proj*)

⚠ **L'opérateur « AND » est prioritaire sur l'opérateur « OR » (D'où ici les parenthèses)**

Ordonnancement des résultat

Sélection des champs des attributs « Etage », « Id » et « capacite » dans la table « Salles » mais uniquement pour des salles du « batiment » 'A' . les résultats étant ordonnés dans l'ordre croissant des capacités : mot clé : « ORDER BY ».

SELECT *Etage,Id,Capacite* FROM *Salles*
WHERE *Batiment='A'* ORDER BY *Capacite*

Ordre décroissant : Ajouter après « ORDER BY Attribut » le mot clé : **« DESC »**

Limiter en nombre les résultats : Pour seulement n résultats, ajouter **« LIMIT n »**

Ne pas commencer au 1ier (ou dernier) : Pour commencer au k^{ième} ajouter **« OFFSET k »**

Exemple : Sélection des champs des attributs « Etage », « Id » et « Capacite » dans la table « Salles » mais pour des salles du « batiment » 'A'. On ne présentera que les 2, 3, 4 et 5^{ième} salles par ordre décroissant de capacité.

SELECT *Etage,Id,Capacite* FROM *Salles* WHERE *Batiment='A'*
ORDER BY *Capacite,Etage* DESC LIMIT 4 OFFSET 1

Requêtes d'agrégation

Sélection sans redondances

Lister, dans l'ordre croissant de capacité et du nombre de postes informatique, les différentes configurations de salles dont dispose le centre de formation. La configuration d'une salle étant définie par sa capacité et son nombre de postes d'informatique. On utilise pour cela le mot clé « DISTINCT ».

SELECT DISTINCT *Capacite,Nbr_info* FROM *Salles*
ORDER BY *Capacite,Nbr_info*

Comptage d'un nombre d'éléments

Comptage du nombre de salles.

SELECT COUNT(*) AS *'Nombre de salles'* FROM *Salles*

Cette requête ne renvoie pas une liste mais simplement un nombre entier correspondant au nombre de lignes de la requête : « SELECT * FROM Salles »

Listing d'un comptage par regroupement

Comptage du nombre de salles par bâtiment. On utilise alors le mot clé « GROUP BY » :

SELECT *Batiment*,COUNT(*) AS *'Nombre de salles'*
FROM *Salles* GROUP BY *Batiment*

Cette requête renvoie une liste où chaque ligne est un comptage.

Comptage d'un nombre d'attributs différents

Comptage du nombre de bâtiments. On utilise alors le mot clé « DISTINCT » à l'intérieur de la commande de comptage « COUNT » :

SELECT COUNT(DISTINCT *Batiment*)
AS *'Nombre de bâtiments'* FROM *Salles*

Comptage d'un nombre d'attributs différents par regroupement sur un autre attribut

Comptage du nombre d'étages par bâtiments. On utilise alors les mots clé « DISTINCT » et « COUNT » pour le comptage d'attributs différents et le mot clé « GROUP BY » pour regrouper ce comptage par bâtiments différents.

```
SELECT Batiment,COUNT(DISTINCT(Etage))
  AS 'Nombre d'étages' FROM Salles
  GROUP BY Batiment
```

Opérations sur les attributs

Il est possible de créer des lignes qui n'existent pas dans la table en effectuant des opérations sur les attributs de la table.

Listing des différentes salles (bâtiment, étage et Numéro de salle) et de leur taux d'équipement en postes informatiques (en %) classées par bâtiments et étages :

```
SELECT Batiment,Etage,Id AS Numéro, 100*Nbr_info/Capacite
  AS 'Taux d'équipement informatique en %' FROM Salles
  ORDER BY Batiment,Etage
```

Remarque : Les attributs « Nbr_info » et « Capacite » sont du type « INTEGER » (entier) donc le résultat du calcul retourné est un entier qui est l'arrondi à l'entier inférieur. Donc si on ne met pas le « 100* » les résultats seront soit « 0 » (pour $0 \leq$ taux $<$ 100%) soit « 1 » (taux $\geq$ 100%)

Il convient donc pour les opérations d'être vigilant sur le type des attributs :

« INTEGER » pour entier « DOUBLE » pour un flottant « DECIMAL(n,m) » pour un nombre décimal de n chiffres et m décimales.

Autres mots clés pour des opérations :
- **CEIL(x)** pour l'arrondi de x à l'entier supérieur
- **ABS(x)** pour la valeur absolue de x
- **POWER(x,n)** pour le résultat x^n

Certains mots clés s'appliquent à des chaines de caractère :

« CHAR_LENGTH(ch) » renvoie le nombre de caractère de la chaine « ch »

« Ch1||ch2 » renvoie la concaténation des chaines de caractère « ch1 » et « ch2 »

Opérations sur des regroupements

Certaine opérations ne s'appliquent que sur un groupement de plusieurs enregistrements. On a la possibilité de sélectionner dans un regroupement :

- ☞ la valeur maximale mot clé : **« MAX »**
- ☞ la valeur minimale mot clé : **« MIN »**
- ☞ la valeur moyenne mot clé : **« AVG »**
- ☞ la somme des valeurs mot clé : **« SUM »**

Listing de la capacité d'accueil en stagiaires de chaque étage ordonné par bâtiment et étage.

```
SELECT Batiment,Etage,SUM(Capacite) FROM Salles
  GROUP BY Batiment,Etage ORDER BY Batiment,Etage
```

Sélection de résultats d'opérations sur les attributs

Il est possible d'effectuer une sélection sur les résultats des opérations par regroupement. Pour cela on utilise le mot clé « HAVING ». Ce mot clé est l'équivalent du mot clé « WHERE » qui ne s'applique lui que sur les attributs.

Listing des étages (bâtiments, étages) dont la plus grande salle peut accueillir au moins 20 stagiaires.

```
SELECT Batiment,Etage FROM Salles
  GROUP BY Batiment,Etage HAVING MAX(Capacite) >= 20
```

Listing des étages (bâtiments et étages) avec une moyenne de capacité d'accueil des stagiaires au moins égale à 20.

```
SELECT Batiment,Etage,AVG(Capacite) AS Moyenne
  FROM Salles  GROUP BY Batiment,Etage HAVING Moyenne>=20
```

Requêtes avec jointure

Produit cartésien

Le produit cartésien de deux tables donne une table dont les attributs sont concaténés et dont les enregistrements sont toutes les combinaisons possibles entre les différents enregistrements des tables.

Ce produit cartésien s'obtient en écrivant les 2 noms des tables séparés par une virgule.

Exemple on a 2 tables : « Etudiants » (2 attributs et 2 enregistrements) une clé primaire « Id_e »

Et : « Copies » (2 attributs et 2 enregistrements) une clé primaire « Id_c »

Et une clé étrangère « Etudiant » de la table « Etudiants »

Etudiants	
Id_e	Prenom
1	Sophie
2	Luc

Copies		
Id_c	Etudiant	Note
1	1	15
2	1	18
3	2	16

Le produit cartésien « Etudiant,Copies » donne la table :

Etudiants,Copies				
Id_e	Prenom	Id_c	Etudiant	Note
1	Sophie	1	1	15
1	Sophie	2	1	18
1	Sophie	3	2	16
2	Luc	1	1	15
2	Luc	2	1	18
2	Luc	3	2	16

Remarque sur les dimension du produit cartésien : Pour une table 1 de a1 attributs et e1 enregistrements et une table 2 de a2 attributs et e2 enregistrements le produit cartésien donne une table de (a1+a2) attributs et (e1×e2) enregistrements.

Jointure entre deux tables

En général on fait le produit cartésien de deux tables dont un attribut (clé étrangère) de l'une contient des valeurs correspondant à des valeurs d'un attribut de l'autre (clé primaire).

Dans notre exemple la clé étrangère de la table « Copies » : (« Etudiant ») correspond à la clé primaire de la table « Etudiants » : (« Id_e »). C'est-à-dire que toutes les valeurs des champs de l'attribut « Etudiant » de la table « Copies » sont présents dans l'un au moins des champs de l'attribut « Id_e » de la table « Etudiants ». Chaque note peut donc être attribuée à un étudiant qui a un prénom.

En ne sélectionnant que les enregistrements où la clé étrangère des copies : « Copies.Etudiant »

est égale à la clé primaire des Etudiants : « Etudiants.Id_e »

On fait la jointure entre les tables « Etudiants » et « Copies ».

La requête SQL : **SELECT Etudiants.Prenom,Copies.Note**
FROM Etudiants,Copies
WHERE Etudiants.Id_e=Copies.Etudiant

Prenom	Note
Sophie	15
Sophie	18
Luc	16

Permet d'obtenir la liste des notes avec les prénoms de chacune des copies. Ainsi cette requête renvoi la liste (ou table) ci-contre :

Remarque : le préfixe (nom de la table) devant le nom de l'attribut n'est pas obligatoire mais devient indispensable lorsque les deux tables ont des noms d'attributs identiques.

En général on préfère écrire cette requête avec une autre syntaxe utilisant les mots clés « JOIN » qui fait le produit cartésien et le mot clé « ON » qui fait la sélection permettant la jointure :

Exemple de Jointure Simple

Liste des désignations des stages avec les noms et prénoms et adresses mail des Responsables.

Liste des dates et créneaux réservés par la formatrice Nerrière Stéphanie.

Exemples de Jointure Multiples

Liste des stagiaires (Nom, Prénom) inscrits au Stage « Comptabilité analytique »

Félicie pourra-t-elle utiliser une imprimante le 12 Octobre 2022 ?

Dans combien de salles différentes, la formatrice Nerrière Stéphanie interviendra-t-elle pour le stage du 24/10/2022 au 28/10/2022.

Requêtes par combinaisons

Pour extraire des attributs de plusieurs tables on peut utiliser des constructions ensemblistes qui permettent de les combiner. Pour cela on a trois mots clé possibles.

- ☞ qui renvoie la concaténation des enregistrements des 2 listes
- ☞ qui renvoie les enregistrements de la 1^{ière} liste qui ne sont pas dans la 2nd.
- ☞ qui renvoie uniquement les enregistrements présents dans les deux listes.

Pour une telle requête on écrit la requête générant la première liste, puis le mot clé « UNION » ou « EXCEPT » ou « INTERSECT » et enfin la requête générant la seconde liste.

Remarque :

Exemple : Liste des identifiants des formateurs qui ne sont pas responsable d'un stage.

SELECT ***Id*** **FROM** ***Formateurs***
EXCEPT
SELECT ***Responsable*** **FROM** ***Stages***

Exemple de combinaison simple

Liste des Noms et prénoms des formateurs qui sont (ou ont été) aussi des stagiaires. On suppose qu'il n'y a pas d'homonynes.

Exemple de combinaison avec conditions

Listes des Numéros de Salles qui ont une capacité d'au moins 20 stagiaires et qui sont libres toute la semaine du 24 au 28 Octobre 2022.

Exemple de combinaison avec jointures

Listes des noms et prénoms des personnes ayant fréquenté le centre formation la semaine du 24 au 28 Octobre 2022.