

SQL DS 2

Document de Révision SQL et Algèbre Relationnelle

I. SQL (Structured Query Language)

1. Sélection (Selection)

SQL : Récupérer les étudiants dont l'âge est supérieur à 18 ans.

```
SELECT * FROM Etudiants  
WHERE age > 18;
```

Algèbre :

$\sigma_{age > 18}(\text{Etudiants})$

2. Projection (Projection)

SQL : Récupérer uniquement les noms et âges des étudiants.

```
SELECT nom, age FROM Etudiants;
```

Algèbre :

$\pi_{nom, age}(\text{Etudiants})$

3. Produit Cartésien (Cartesian Product)

SQL : Récupérer le produit cartésien entre les tables `Etudiants` et `Cours`.

```
SELECT * FROM Etudiants, Cours;
```

Algèbre :

$\text{Etudiants} \times \text{Cours}$

4. Union (Union)

SQL : Récupérer tous les noms d'étudiants et de professeurs.

```
SELECT nom FROM Etudiants
UNION
SELECT nom FROM Professeurs;
```

Algèbre :

$$\pi_{nom}(\text{Etudiants}) \cup \pi_{nom}(\text{Professeurs})$$

5. Intersection (Intersection)

SQL : Récupérer les étudiants inscrits à la fois dans Cours1 et Cours2 .

```
SELECT nom FROM Cours1
INTERSECT
SELECT nom FROM Cours2;
```

Algèbre :

$$\pi_{nom}(\text{Cours1}) \cap \pi_{nom}(\text{Cours2})$$

6. Différence (Difference)

SQL : Récupérer les étudiants qui ne sont pas inscrits à Cours1 .

```
SELECT nom FROM Etudiants
WHERE nom NOT IN (SELECT nom FROM Cours1);
```

Algèbre :

$$\pi_{nom}(\text{Etudiants}) - \pi_{nom}(\text{Cours1})$$

7. Jointure (Join)

SQL : Récupérer les étudiants avec leurs cours.

```
SELECT Etudiants.nom, Cours.nom_cours
FROM Etudiants
```

```
JOIN Cours ON Etudiants.id = Cours.etudiant_id;
```

Algèbre :

$\text{Etudiants} \bowtie_{\text{Etudiants.id=Cours.etudiant_id}} \text{Cours}$

II. Concepts Importants de l'Algèbre Relationnelle

1. Clés Primaires et Étrangères

- **Clé Primaire** : Un attribut qui identifie de manière unique chaque enregistrement dans une table.
- **Clé Étrangère** : Un attribut qui crée un lien entre deux tables.

2. Fonctions d'Agrégation

SQL : Calculer le nombre d'étudiants dans une classe.

```
SELECT COUNT(*) FROM Etudiants WHERE classe = 'L3';
```

Algèbre :

$\text{COUNT}(\sigma_{\text{classe}='L3'}(\text{Etudiants}))$

3. Grouper (Group By)

SQL : Récupérer le nombre d'étudiants par classe.

```
SELECT classe, COUNT(*) as nombre_etudiants
FROM Etudiants
GROUP BY classe;
```

Algèbre :

$\pi_{\text{classe}, \text{COUNT}(*)}(\text{Etudiants})$
