7 Bases de données S2

On se limite volontairement à une description applicative des bases de données en langage SQL. Il s'agit de permettre d'interroger une base présentant des données à travers plusieurs relations. On ne présente ni l'algèbre relationnelle ni le calcul relationnel.

Notions	Commentaires
Vocabulaire des bases de données : tables ou relations, attributs ou colonnes, domaine, schéma de tables, enregistrements ou lignes, types de données.	On présente ces concepts à travers de nombreux exemples. On s'en tient à une notion sommaire de domaine : entier, flottant, chaîne ; aucune considération quant aux types des moteurs SQL n'est au programme. Aucune notion relative à la représentation des dates n'est au programme ; en tant que de besoin on s'appuie sur des types numériques ou chaîne pour lesquels la relation d'ordre coïncide avec l'écoulement du temps. Toute notion relative aux collations est hors programme ; en tant que de besoin on se place dans l'hypothèse que la relation d'ordre correspond à l'ordre lexicographique usuel.
Clé primaire.	Une clé primaire n'est pas forcément associée à un unique attribut même si c'est le cas le plus fréquent. La notion d'index est hors programme.
Entités et associations, clé étrangère. **[P] On peut se limiter à une présentation informelle.** **[C] On ne présuppose aucun vocabulaire, formalisme, règles ou normalisation.**	On s'intéresse au modèle entité–association au travers de cas concrets d'associations $1-1, 1-*, *-*$. Séparation d'une association $*-*$ en deux associations $1-*$. L'utilisation de clés primaires et de clés étrangères permet de traduire en SQL les associations $1-1$ et $1-*$.
Requêtes `SELECT` avec simple clause `WHERE` (sélection), projection, renommage `AS` **[E] on peut omettre `AS`.** Utilisation des mots-clés `DISTINCT`, `LIMIT`, `OFFSET`, `ORDER BY`. Opérateurs ensemblistes `UNION`, `INTERSECT` et `EXCEPT`, produit cartésien.	Les opérateurs au programme sont `+`, `-`, `*`, `/` (on passe outre les subtilités liées à la division entière ou flottante), `=`, `<>`, `<`, `<=`, `>`, `>=`, `AND`, `OR`, `NOT`, `IS NULL`, `IS NOT NULL`. **[C] On précise explicitement si l'on souhaite un résultat sans doublons.**
Jointures internes T_1 `JOIN` T_2 … `JOIN` T_n `ON` ϕ, externes à gauche T_1 `LEFT JOIN` T_2 `ON` ϕ.	On présente les jointures (internes) en lien avec la notion d'associations entre entités.
Agrégation avec les fonctions `MIN`, `MAX`, `SUM`, `AVG` et `COUNT`, y compris avec `GROUP BY`.	Pour la mise en œuvre des agrégats, on s'en tient à la norme SQL99. On présente quelques exemples de requêtes imbriquées.
Filtrage des agrégats avec `HAVING`.	On marque la différence entre `WHERE` et `HAVING` sur des exemples.

Mise en œuvre
La création, la suppression et la modification de tables au travers du langage SQL sont hors programme. La mise en œuvre effective se fait au travers d'un logiciel permettant d'interroger une base de données à l'aide de requêtes SQL. Récupérer le résultat d'une requête à partir d'un programme n'est pas un objectif. Même si aucun formalisme graphique précis n'est au programme, on peut décrire les entités et les associations qui les lient au travers de diagrammes sagittaux informels. Sont hors programme : la notion de modèle logique *vs* physique, les bases de données non relationnelles, les méthodes de modélisation de base, les fragments DDL, TCL et ACL du langage SQL, l'optimisation de requêtes par l'algèbre relationnelle.