



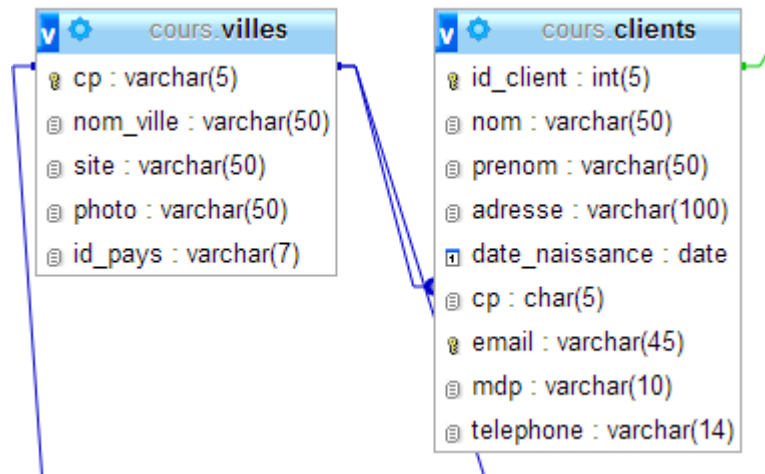
1.1 - FONTAMENTAUX : LES BASES

Cf aussi :
BDR_les_bases.pdf
CRUD.pdf
table.pdf

Notion	Définition ou syntaxe
SGBD	Système de Gestion de bases de Données : logiciel permettant de gérer une Base de Données
BDR	Base de Données Relationnelle Entrepôt de données composé de tables. R veut dire Relationnel ie à base d'algèbre relationnelle comme extension de l'algèbre ensembliste.
SQL	Structure Query Language : langage d'interrogation structuré Implémentation de l'algèbre relationnelle. Le langage SQL est composé de 4 sous langages : LDD, LMD, LCD, LCT.
LDD	Langage de Définition de Données. Gestion des conteneurs (database, table, user, index , ...) 3 verbes : CREATE, ALTER, DROP.
LMD	Langage de Manipulation de Données. Gestion des contenus (table). 4 verbes : INSERT, SELECT, UPDATE, DELETE (le CRUD : create, read, update, delete).
LCD	Langage de Contrôle de Données. Gestion des droits accordés ou retirés aux utilisateurs. 2 verbes : GRANT et REVOKE.
LCT	Langage de Contrôle de Transaction. Gestion des transactions. 2 verbes : COMMIT et ROLLBACK.
Table	Ensemble de lignes/colonnes Possède une clé primaire composée de 1 ou plusieurs colonnes. Possède éventuellement une ou plusieurs clés étrangères en référence à une ou des clés primaires d'une ou plusieurs autres tables.
Création d'une BD	CREATE DATABASE nomBD;
Utilisation d'une BD	USE nomBD;

Modification d'une BD	ALTER DATABASE ...;
Suppression d'une BD	DROP DATABASE nomBD;
Création d'une table	CREATE TABLE nomTable(colonne1 TYPE CONTRAINTE, colonne2 TYPE CONTRAINTE, ... PRIMARY KEY colonne);
Suppression d'une table	DROP TABLE nomTable;
Modification d'une table	ALTER TABLE nomTable ...;
Principaux types SQL	TEXTE : VARCHAR, CHAR NUMERIQUE : integer, long, double, real ... DATE : date et datetime BINAIRE : blob (Binary Large Object)
Ajout d'un enregistrement	INSERT INTO nomTable(liste de colonnes) VALUES(liste de valeurs);
La clause WHERE	Permet de sélectionner certains enregistrements aussi bien pour la sélection que la suppression et que la modification WHERE condition
Opérateurs de comparaison	=, !=, >, >=, <, <=
Opérateurs logiques	AND, OR, NOT
Opérateurs ensemblistes	IN, BETWEEN, LIKE
Suppression d'un enregistrement	DELETE FROM nomTable [WHERE condition];
Modification d'un enregistrement	UPDATE nomTable SET colonne1=valeur1, colonne2=valeur2 [WHERE condition];
Sélection de tous les enregistrements	SELECT * FROM nomTable;
Sélection de certaines colonnes d'une table (Projection)	SELECT col1, col2, coln FROM nomTable;
Sélection d'un enregistrement ou de quelques-uns (Restriction)	SELECT colonnes FROM nomTable WHERE condition;

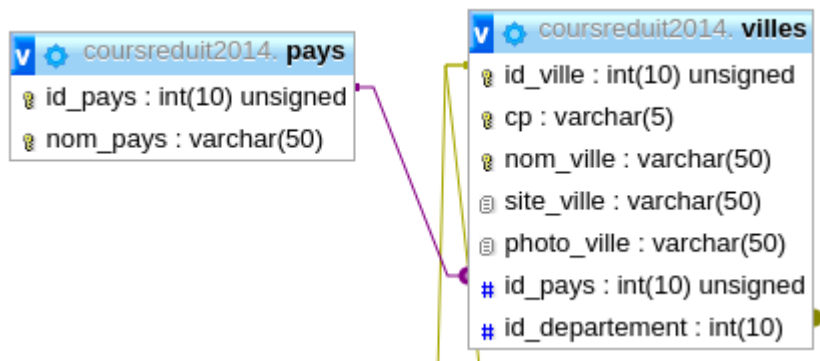
1.2 - FONDAMENTAUX : LES REQUÊTES MULTI-TABLES ET LES FONCTIONS AVANCÉES



Notion	Définition ou syntaxe ou exemple
La jointure	L'utilisation de 2 ou n tables dans un ordre SQL SELECT. Concaténation d'une liste de colonnes et contanénation de chaque enregistrement d'une table avec tous les enregistrements de l'autre table correspondant à un prédicat (une condition).
La jointure naturelle	La jointure avec une liaison d'égalité entre la PK et la FK.
La jointure naturelle entre la table villes et la table clients	SELECT * FROM villes NATURAL JOIN clients;
L'equi-jointure	La jointure avec l'utilisation de l'opérateur de comparaison = dans la condition de jointure.
La jointure entre la table villes et la table clients	SELECT * FROM villes INNER JOIN clients ON villes.cp = clients.cp;
La jointure externe	La juxtaposition des enregistrements d'une table avec ceux d'une autre table + ceux qui n'ont pas de correspondance.
La jointure externe entre la table villes et la table clients	SELECT * FROM villes LEFT OUTER JOIN clients ON villes.cp = clients.cp;
La jointure externe et la non correspondance	La juxtaposition des enregistrements d'une table mais seulement ceux qui n'ont pas de correspondance.
La liste des villes où il n'y a pas de clients	SELECT * FROM villes LEFT OUTER JOIN clients ON villes.cp = clients.cp WHERE clients.id_client IS NULL;
La fonction agrégat COUNT()	Fonction de groupe d'enregistrement de nature statistiques.
Le nombre de villes de la table villes	SELECT COUNT(*) FROM villes;
La clause GROUP BY. Le nombre de clients par ville	Le regroupement d'enregistrements. SELECT nom_ville, COUNT(*) FROM villes NATURAL JOIN clients GROUP BY nom_ville;
La clause HAVING	Condition sur un groupe.

La liste des villes où il y a plus de 5 clients	<pre>SELECT nom_ville, COUNT(*) FROM villes NATURAL JOIN clients GROUP BY nom_ville HAVING COUNT(*) > 5;</pre>
L'auto-jointure	Jointure sur la même table
La theta-jointure	Jointure avec un opérateur de comparaison différent de =.
Le produit cartésien	La concaténation des colonnes d'une table avec les colonnes d'une autre table et la combinaison de chaque enregistrement d'une table avec tous les enregistrements d'une autre table.

Exercez-vous !



id_pays	nom_pays	id_ville	nom_ville
1	France	1	Paris 11
1	France	2	Paris 12
1	France	3	Lyon
1	France	4	Lille
1	France	5	Versailles
1	France	6	Neuilly
1	France	7	Nanterre
1	France	8	Vincennes
1	France	9	Saint-Mande
2	Italie	10	ROME
2	Italie	11	MILAN

Code pays	Nom du pays	Code ville	Nom de la ville
41	Albanie	NULL	NULL
1	France	1	Paris 11
1	France	6	Neuilly
1	France	2	Paris 12
1	France	7	Nanterre
1	France	3	Lyon
1	France	8	Vincennes
1	France	4	Lille
1	France	9	Saint-Mande
1	France	5	Versailles
4	Hongrie	NULL	NULL
2	Italie	11	MILAN

1.3 - CORRIGÉS

```
SELECT p.id_pays, p.nom_pays, v.id_ville, v.nom_ville
FROM pays p
INNER JOIN villes v ON p.id_pays = v.id_pays;
```

```
SELECT p.id_pays AS "Code pays", p.nom_pays AS "Nom du pays",
v.id_ville AS "Code ville", v.nom_ville AS "Nom de la ville"
FROM pays p LEFT OUTER JOIN villes v
ON p.id_pays = v.id_pays
ORDER BY p.nom_pays
```