

Le langage SQL – Rappels

Description du thème : Présentation des principales notions nécessaires pour réaliser des requêtes SQL	
Mots-clés :	Bases de données relationnelles, Open Office, champs, relations, requêtes, opérations relationnelles
Niveau :	Ce cours peut être commencé en 1 ^{ère} STG (les parties I à III comprise) et peut être complété en TGSI ou TCFE.
Domaine(s) :	Informatique de gestion
Type(s) de ressource :	Support de cours - travaux élèves en autonomie
Objectifs :	Le langage SQL : l'instruction SELECT pour traduire les opérations de sélection, de projection et de jointure.
Séance(s) développée(s) :	
Place de la séquence dans la progression annuelle :	En fin de 1ère STG ou au début de terminale.
Pré-requis :	Maîtrise du schéma relationnel. Connaissance des bases de données relationnelles.
Outils :	Microsoft Access ou Open Office.Org Base. Un vidéo projecteur pour l'évaluation des travaux des élèves ou pour la présentation du corrigé.
Conditions de réalisation :	Un ordinateur par personne. Le cours peut être divisé en 3 parties : Les parties I, II et III peuvent être enseignées d'une traite. Puis, on peut aborder la partie IV. De même ensuite, on peut traiter séparément la partie V, puis la partie VI.
Evaluation :	Formative
Temps approximatif de réalisation :	Parties I, II, III, cours et TD : 5 heures environ. Partie IV : 3 heures environ. Partie V : 2 heures environ. Partie VI : 3 heures environ.
Compétences B2i :	3-Créer, produire, traiter, exploiter des données.
Dossier professeur :	Dossier élève :
- Corrigé_requêtes.doc : corrigé des requêtes demandées	Les fichiers Anabase_sql.odt : base de données Base Anabase.mdb : base de données Access
Contact :	claude.pasqualini@ac-limoges.fr

Le but de ce propos n'est pas de présenter un nouveau cours sur SQL. Il en existe beaucoup qui sont disponibles sur Internet. L'objectif est plutôt de présenter une marche à suivre pour présenter les différentes parties du langage de requêtes. Je joins donc à ce cours quelques exercices d'illustrations. L'enseignant pourra choisir ceux qui correspondent le mieux à sa progression. Pour chaque requête, je fournis le résultat à obtenir.

La SGBD sur lequel ces requêtes ont été testées est OPEN OFFICE.OrgBase. La base de données Anabase est dans votre dossier sous le nom : anabase_sql.odt ainsi que le corrigé des exercices : corrigé_requetes.doc

Pour illustrer ce chapitre, j'ai décidé de m'appuyer sur un cas créé dans les années 80 et bien connu des enseignants en informatique : le cas « Anabase » dont voici quelques extraits :

Le congrès se déroule sur une semaine de 5 jours ouvrables. Il comporte différentes sessions d'une demi-journée chacune, consacrées à des conférences, débats sur un thème donné.

Un congressiste peut participer à ce congrès envoyé par un organisme (entreprise, administration, université ou lycée) ou non.

Il peut éventuellement être accompagné d'une personne. On suppose qu'un congressiste ne peut pas avoir plus d'un accompagnateur.

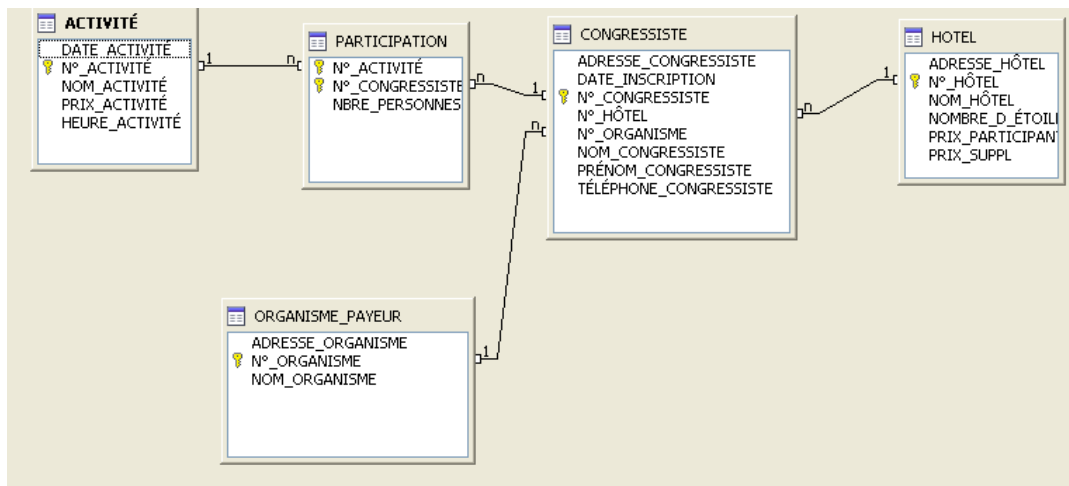
Le congressiste est affecté à un hôtel. Chaque hôtel a proposé un prix par participant et un supplément pour l'accompagnateur. Ces prix sont valables pour toutes les chambres d'un même hôtel.

Le bureau d'animation propose des activités culturelles (spectacles, excursions, concerts, visites de monuments...) pour meubler le temps des accompagnateurs et des congressistes qui ont des disponibilités dans leur emploi du temps. Chaque activité est repérée par un code activité, une désignation et un horaire de début et de fin.

Les congressistes et les accompagnateurs peuvent s'inscrire jusqu'à la veille au soir aux activités souhaitées.

Dans la suite du travail, nous allons nous limiter aux informations concernant les congressistes et leur organisme payeur, les hôtels ainsi que les activités proposées.

Le modèle relationnel obtenu est le suivant :



I- Rappels sur l'algèbre relationnelle

Il faut rappeler que le modèle relationnel est basé sur le modèle mathématique. Il existe deux types d'opérations que l'on peut effectuer sur les tables :

- **les opérations ensemblistes** : union, intersection et différence.
- **les opérations relationnelles** : projection, sélection, jointure, division.

Nous nous limiterons ici aux opérations relationnelles :

- **la projection** : son objectif est de fournir le résultat d'une requête en supprimant les attributs d'une table.
- **La sélection** : elle a pour but d'extraire des enregistrements (lignes) d'une table répondant à un ou plusieurs critères de recherche.
- **La jointure** : c'est la mise en correspondance de 2 tables selon un critère.

Utilisation de SQL pour représenter les opérations opérationnelles

Les 3 opérations vues ci-dessus sont exprimées en SQL avec la même instruction : **SELECT**.

II- La projection

Syntaxe : **select** liste attributs **from** table [order by nom attribut, ...]

Exemple : `SELECT NOM_CONGRESSISTE, PRÉNOM_CONGRESSISTE FROM CONGRESSISTE` permet de réaliser la projection de la table congressiste sur les attributs nom_congressiste et prénom_congressiste.

`ORDER BY` suivi de nom d'un attribut permet l'affichage des résultats trié suivant le nom d'attribut. Si l'on souhaite obtenir un tri suivant plusieurs critères, il faut préciser une suite d'attributs derrière le mot-clé `ORDER BY`.

Par défaut le tri est croissant. Le mot-clé DESC placé à la suite du nom de l'attribut permet de réaliser un tri décroissant.

On peut afficher également le résultat d'un calcul : une expression arithmétique faisant intervenir plusieurs attributs numériques. De plus, pour des raisons de lisibilité, on peut, grâce au mot-clé AS, donner un nom à l'expression :

SELECT attribut 1 + attribut 2 AS somme FROM table

Remarque : SQL n'élimine pas les doublons à moins que figure le mot-clé DISTINCT à la suite de SELECT.

II-1-Exercices de requêtes de projection

R1 : Afficher le nom, la date et l'heure de chaque activité.

	NOM_ACTIVITÉ	DATE_ACTIVITÉ	HEURE_ACTIVITÉ
▶	Excursion à la Rhune	05/06/08 00:00	30/12/99 17:00
	Visite St Jean de Lu	05/06/08 00:00	30/12/99 09:30
	Concert Jazz	06/06/08 00:00	30/12/99 20:30
	Tournoi Pelote basq	07/06/08 00:00	30/12/99 20:30

R2 : Afficher les villes dont sont originaires les congressistes :

	ADRESSE_CONGRESSISTE
▶	Bordeaux
	Cognac
	La Rochelle
	Limoges
	Paris
	Royan
	St Junien
	St Victurnien

III- La sélection

La sélection s'exprime grâce à la clause WHERE.

Syntaxe : select * from table where condition.

Le critère de sélection (condition) peut faire intervenir un attribut, un opérateur et une constante.

Le plus souvent, la projection et la sélection sont composées :

Syntaxe : select liste attributs from table where condition.

Exemple: select * from congressiste where ADRESSE_CONGRESSISTE = 'St Junien'
donne le résultat suivant :

	N°_CONGRESSISTE	N°_HÔTEL	N°_ORGANISME	NOM_CONGRESSISTE	PRÉNOM_CONGRESSISTE	ADRESSE_CONGRESSISTE	TÉLÉPHONE_CONGRESSISTE	DATE_INSCRIPTION
▶	1	1	3	Millet	Alain	St Junien	0555112233	01/03/08 00:00
	6	2	1	Santerre	Jean	St Junien	0555667788	03/01/08 00:00

Recherche	Caractère	Exemple	Commentaires
Tous les attributs	*	Select * ...	
D'un attribut non renseigné	NULL	...where adresse is NULL	Adresse non renseignée
D'un caractère dans une chaîne	-	... where nom like 't-t'	Le nom contient un t En 1 ^{ère} et en 3 ^{ème} lettre
D'un ensemble de caractères Dans une chaîne	%	Where nom like 'd%d'	Le nom commence par un d et finit par un d.

Dans certains SGBD, le caractère % peut être remplacé par le caractère *.

Exemples d'opérateurs :

Opération	Opérateur SQL	Exemple	Commentaires
Est égal à	=	where numéro = 10	
Différent de	<> ou !=	where numéro <> 10	
Supérieur	>		
Supérieur ou égal	>=		
Inférieur	<		
Inférieur ou égal	<=		
Entre deux valeurs	Between ... and	Where prix between 15 and 20	Les bornes sont Incluses
Contient (pour une chaîne de caractères)	Like	Where nom like 'A %'	Recherche des noms commençant par la letter A
Ne contient pas	Not like		
Est vide	Is NULL		
N'est pas vide	Is not NULL		
Et	And	Where adresse = 'Limoges' and Num_organisme is null	
Ou	Or	Where adresse = 'Limoges' or adresse = 'St Junien'	
Non	Not		
Dans la liste	In	Where ville in (« Limoges », « St Junien », « Cognac »)	Recherche les villes dans la liste qui suit le mot-clé IN.

Gestion des dates :

Lorsque l'on veut utiliser une date comme critère de sélection, il n'existe pas vraiment de solutions standards :

Microsoft Access utilise la date sous la forme : #jj/mm/aaaa#

Open Office Base sous la forme {D 'aaaa-mm-jj' }

D'autres SGBD utilisent la forme 'aaaa-mm-jj'.

III-1 Exercices de requêtes de sélection

R3 : Afficher la liste des organismes limougeaux.

	N°_ORGANISME	NOM_ORGANISME	ADRESSE_ORGANISME
▶	1	IUT Limoges	Limoges
	3	Lycée Valadon	Limoges
+			

R4 : Afficher les noms et prénoms de congressistes ne dépendant pas d'un organisme.

	NOM_CONGRESSISTE	PRÉNOM_CONGRESSISTE
▶	Caillierez	Gabriel
	Delord	Thierry

R5 : Afficher la liste des congressistes situés dans les hôtels 2, 3 ou 4.

	N°_CONGRESSISTE	N°_HÔTEL	N°_ORGANISME	NOM_CONGRESSISTE	PRÉNOM_CONGRESSISTE	ADRESSE_CONGRESSISTE	TÉLÉPHONE_CONGRESSISTE	DATE_INSCRIPTION
▶	3	3	2	Namèche	Nicole	Cognac	0545334455	03/02/08
	6	2	1	Santerre	Jean	St Junien	0555667788	03/01/08
	7	2	2	Pasqualini	Gabrielle	Bordeaux	0556778899	03/10/08
	8	4	3	Messenger	Yves	St Victurnien	0555039900	03/04/08
	9	3		Delord	Thierry	Royan	0545001122	03/06/08
+								

R6 : Afficher la liste des activités ayant lieu le 5 ou le 7 juin 2008.

	N°_ACTIVITÉ	NOM_ACTIVITÉ	DATE_ACTIVITÉ	PRIX_ACTIVITÉ	HEURE_ACTIVITÉ
▶	1	Excursion à la Rhune	05/06/08	10	17:00
	2	Visite St Jean de Luz	05/06/08	3	09:30
	4	Tournoi Pelote basque	07/06/08	12	20:30
+					

R7 : Afficher la liste des congressistes (nom et prénom) par ordre alphabétique.

	NOM_CONGRESSISTE	PRÉNOM_CONGRESSISTE
▶	Brisset	Eric
	Caillierez	Gabriel
	Delord	Thierry
	Messenger	Yves
	Michelin	Marie-Pierre
	Millet	Alain
	Namèche	Nicole
	Pasqualini	Gabrielle
	Santerre	Jean

R8 : Afficher la liste des hôtels dont le prix participant est compris entre 35 et 52 €.

	NOM_HÔTEL	PRIX_PARTICIPANT
▶	Beau Rivage	50
	Continental	40

R9 : Afficher la liste des congressistes (nom, prénom) dont le nom commence par un « M ».

	NOM_CONGRESSISTE	PRÉNOM_CONGRESSISTE
▶	Millet	Alain
	Michelin	Marie-Pierre
	Messenger	Yves

R10 : Afficher la liste des congressistes (nom, prénom) dont le nom contient un « A ».

	NOM_CONGRESSISTE	PRÉNOM_CONGRESSISTE
▶	Namèche	Nicole
	Caillierez	Gabriel
	Santerre	Jean
	Pasqualini	Gabrielle
	Messenger	Yves

R11 Afficher la liste des congressistes (nom, prénom) dont le nom finit par un « T ».

	NOM_CONGRESSISTE	PRÉNOM_CONGRESSISTE
▶	Millet	Alain
	Brisset	Eric

R12 : Afficher la liste des hôtels dont le prix participant varie entre 35 et 55 € et dont le prix supplément varie entre 20 et 30 €.

	NOM_HÔTEL	PRIX_PARTICIPANT	PRIX_SUPPL
▶	Continental	40	25

R13 : Afficher pour chaque hôtel la somme des deux prix (prix participant + prix supplément).

	NOM_HÔTEL	PRIX_PARTICIPANT	PRIX_SUPPL	PRIX_TOTAL
▶	Les Flots Bleus	55	35	90
	Beau Rivage	50	32	82
	Itsas-mendia	25	15	40
	Continental	40	25	65

IV- Les fonctions

Il existe de nombreuses possibilités de calcul grâce à certaines fonctions. Les principales sont :

- AVG : moyenne de valeurs
- COUNT : nombre de valeurs
- SUM : somme de valeurs
- MAX : valeur maximale
- MIN : valeur minimale.

Ces fonctions peuvent s'appliquer après un SELECT :

Exemples :

- SELECT MIN(Prix_Participant) FROM Hôtel donne la valeur minimale des prix participant.
- SELECT AVG(Prix_Participant + Prix_Suppl) FROM Hôtel donne la moyenne des prix totaux par hôtel.
- SELECT COUNT(*) FROM Congressiste donne le nombre de congressistes.

Le regroupement de plusieurs résultats

Lorsqu'on utilise des fonctions on a parfois besoin de regrouper les résultats suivant certaines colonnes.

Par exemple, on pourrait souhaiter connaître pour chaque hôtel, le nombre de fois qu'il a été choisi par les congressistes. La fonction à utiliser est COUNT et il faut effectuer dans la table CONGRESSISTE un regroupement par hôtel :

```
SELECT N°_HÔTEL, COUNT( * ) FROM CONGRESSISTE GROUP BY N°_HÔTEL.
```

GROUP BY indique les colonnes à regrouper.

Tous les attributs placés derrière la clause SELECT doivent être présents derrière la clause GROUP BY. La proposition inverse n'est pas vraie.

La clause GROUP BY est obligatoire dès lors qu'il y a à la fois des attributs et des fonctions de calcul derrière la clause SELECT.

L'élimination de certains groupes

Lorsque l'on obtient le résultat d'une requête issue d'un regroupement, on peut souhaiter ne sélectionner que certains groupes parmi d'autres.

Pour cela, il faut ajouter à GROUP BY la clause HAVING qui permet la sélection de groupes parmi d'autres.

Attention à ne pas confondre HAVING et WHERE : HAVING permet la sélection de groupes à la suite d'une requête avec regroupement alors que WHERE permet de sélectionner des lignes pour construire la requête.

Exemple : dans le cas précédent, on ne souhaite afficher que les hôtels qui hébergent moins de 10 congressistes :

```
SELECT N°_HÔTEL, COUNT( * ) FROM CONGRESSISTE GROUP BY N°_HÔTEL HAVING COUNT(*) <10.
```

IV-1- Exercices de requêtes avec fonctions

R14 : Afficher le nombre total de congressistes.

	nombre
▶	9

R15 : Afficher le prix moyen d'une chambre à Biarritz et la même chose pour les suppléments.

	MOY_PARTICIPANT	MOY_SUPPL
▶	52,5	33,5

R16 : Afficher pour chaque activité le nombre de congressistes l'ayant choisie.

	N°_ACTIVITE	NB_FOIS
▶	1	3
	2	2
	3	4
	4	5

R17 : Pour chaque congressiste participant à plus de 2 activités, afficher la moyenne du nombre de personnes.

	N°_CONGRESSISTE	moyenne
▶	1	1,5
	3	2

V- La jointure

La jointure permet de lier des tables en faisant correspondre clés étrangères et clés primaires. Nous nous limiterons ici à l'équi-jointure, c'est à dire une jointure basée sur l'égalité des valeurs des attributs (ils n'ont pas forcément le même nom , mais doivent être définis sur le même domaine).

Syntaxe : Select liste d'attributs **from** table1, table2 **where** table1.attribut1 = table2.attribut2

Attribut1 et attribut2 sont reliés dans la base de données par un lien clé primaire- clé étrangère.

Remarque 1 : Si les noms des tables sont longs, il est possible d'utiliser des étiquettes :

Select liste d'attributs **from** table1 T, table2 S **where** T.attribut1 = S.attribut2

Remarque 2: Quand un même attribut figure dans les deux tables, il faut indiquer la table en temps que préfixe. Un attribut qui ne figure dans une seule table n'a pas besoin de préfixe.

Exemple : select nom, P.numéro from table1 P, table2 V where P.numéro = V.numéro order by nom

V-1- Exercices de requêtes de jointure

R18 : Afficher pour chaque congressiste son numéro, son nom et le nom de l'hôtel où il loge.

	N°_CONGRESSISTE	NOM_CONGRESSISTE	NOM_HÔTEL
▶	1	Millet	Les Flots Bleus
	2	Brisset	Les Flots Bleus
	4	Michelin	Les Flots Bleus
	5	Caillierez	Les Flots Bleus
	6	Santerre	Beau Rivage
	7	Pasqualini	Beau Rivage
	3	Namèche	Itsas-mendia
	9	Delord	Itsas-mendia
	8	Messager	Continental

R19 : Même résultat que dans R18, mais afficher en plus le nom de l'organisme dont il dépend. Que remarquez-vous ?

	N°_CONGRESSISTE	NOM_CONGRESSISTE	NOM_HOTEL	NOM_ORGANISME
▶	1	Millet	Les Flots Bleus	Lycée Valadon
	2	Brisset	Les Flots Bleus	Lycée Valadon
	3	Namèche	Itsas-mendia	Microsoft
	4	Michelin	Les Flots Bleus	Valéo
	6	Santerre	Beau Rivage	IUT Limoges
	7	Pasqualini	Beau Rivage	Microsoft
	8	Messager	Continental	Lycée Valadon

R20 : Afficher pour chaque congressiste son numéro, son nom et le nom de l'hôtel. Restituer cette liste par hôtel et par ordre alphabétique des congressistes.

	NOM_HÔTEL	N°_CONGRESSISTE	NOM_CONGRESSISTE
▶	Les Flots Bleus	2	Brisset
	Les Flots Bleus	5	Caillierez
	Les Flots Bleus	4	Michelin
	Les Flots Bleus	1	Millet
	Beau Rivage	7	Pasqualini
	Beau Rivage	6	Santerre
	Itsas-mendia	9	Delord
	Itsas-mendia	3	Namèche
	Continental	8	Messager

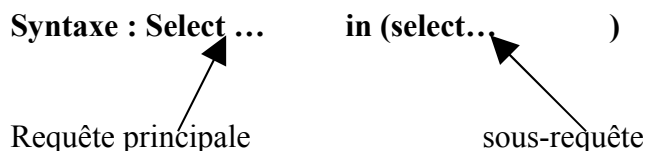
R21 : Même chose que pour R20, mais uniquement pour les congressistes des hôtels de Biarritz.

	NOM_HÔTEL	N°_CONGRESSISTE	NOM_CONGRESSISTE
▶	Les Flots Bleus	2	Brisset
	Les Flots Bleus	5	Caillierez
	Les Flots Bleus	4	Michelin
	Les Flots Bleus	1	Millet
	Beau Rivage	7	Pasqualini
	Beau Rivage	6	Santerre

VI- Les requêtes imbriquées

Les requêtes imbriquées sont basées sur le mot-clé IN (vu au III) qui permet de sélectionner des enregistrements parmi une liste.

Le principe consiste à construire une sous-requête qui donne un résultat équivalent à une liste. Ensuite la requête principale permet de sélectionner des lignes dans la liste précédemment construite.



Le mot-clé IN peut-être remplacé par un opérateur de comparaison (=, <, >, <=, >=). Dans ce cas, le résultat de la sous-requête ne consiste *qu'en une seule*

Utilisation des requêtes imbriquées pour réaliser des jointures

Lorsque le résultat d'une requête ne se trouve que dans une seule table, il est conseillé de la réaliser avec des requêtes imbriquées. En effet, pour le SGBD, cette solution est beaucoup plus rapide.

Exemple : si je veux connaître le nom et le prénom des congressistes envoyés par l'organisme : le Lycée Valadon de Limoges, je peux écrire la requête suivante :

```
SELECT nom_congressiste, prénom_congressiste
FROM CONGRESSISTE
WHERE N°_organisme IN ( SELECT N°_organisme
                        FROM ORGANISME_PAYEUR
                        WHERE nom_organisme = 'Lycée Valadon'
                        AND adresse_organisme = 'Limoges')
```

La requête imbriquée renvoie le n° organisme payeur recherché et la première requête permet de sélectionner les congressistes dépendant de l'organisme payeur ainsi obtenu.

Le SGBD exécute toujours en premier la requête imbriquée.

Si l'on avait voulu la même requête en faisant afficher le nom de l'organisme payeur, cette solution n'aurait pas été adéquate. Il aurait fallu écrire une jointure comme décrite au chapitre V.

Utilisation des requêtes imbriquées avec des fonctions

Les requêtes utilisant les fonctions AVG, COUNT, SUM, MAX et MIN ne renvoient qu'une seule valeur.

Ces requêtes peuvent être utilisées comme sous requêtes afin que l'on puisse faire des comparaisons en utilisant les opérateurs classiques : =, <, >, <>...

Exemple : `select N°_Produit, nom_Produit , prix_produit = (select max(prix_produit) from Produit)`

Permet d'afficher les caractéristiques du produit le plus cher.

V1-1- Exercices de requêtes de requêtes imbriquées

R22 : Afficher le nom et le prénom des congressistes allant au concert de Jazz.

Nom_congressiste	prénom_congressiste
Millet	Alain
Namèche	Nicole
Caillierez	Gabriel
Santerre	Jean

R23 : Afficher le nom et le prénom des congressistes n'allant pas au tournoi de pelote basque.

	Nom_congressiste	prénom_congressiste
▶	Brisset	Eric
	Michelin	Marie-Pierre
	Caillierez	Gabriel
	Delord	Thierry

R24 : Afficher le nom et le prénom des congressistes ayant choisi une chambre d'hôtel dont le montant par jour est supérieur à 70 euros (prix chambre + prix supplémentaire).

	Nom_congressiste	prénom_congressiste
▶	Millet	Alain
	Brisset	Eric
	Michelin	Marie-Pierre
	Caillierez	Gabriel
	Santerre	Jean
	Pasqualini	Gabrielle

R25 : Afficher le nom et le prénom des congressistes ayant choisi l'activité la moins chère.

	Nom_congressiste	prénom_congressiste
▶	Millet	Alain
	Namèche	Nicole

Exemples de sites Internet présentant un cours SQL :

<http://sql.developpez.com/>: le cours complet de SQL de A à Z.

<http://cerig.efpg.inpg.fr/tutoriel/bases-de-donnees/chap18.htm>

<http://webtic.free.fr/sql/>

<http://langagesql.free.fr/accueil.php>: un petit site amusant, assez limité.

MEMENTO

Termes	Définitions
Modèle relationnel	Organisation des données sous forme de tables permettant leur manipulation par des opérateurs de l'algèbre relationnelle
Base de données relationnelle	Base de données structurée selon un modèle relationnel et régi par un Système de Gestion de Bases de Données Relationnel (SGBD-R). Exemples de SGBD-R libres : Base de Open Office, MySql, PostgreSQL... Exemples de SGBD-R propriétaires : Access, SQL Server, Oracle, DB2, Informix...
Opération relationnelle	Elle consiste à créer une nouvelle table, résultat d'une opération sur une ou plusieurs tables. Ces opérations sont ensemblistes (union, intersection, différence, produit cartésien) ou relationnelles (sélection, projection, jointure, division).
Attribut (champ), (colonne)	Nom d'une information stockée dans une table.
Tuple (ligne), (occurrence)	Correspond à une ligne d'une table,
Clé primaire	Attribut ou ensemble d'attributs qui permet d'identifier un tuple de manière unique.
Clé étrangère	Attribut ou ensemble d'attributs faisant référence à une clé primaire d'une autre table.
Intégrité référentielle	Une clé étrangère doit toujours faire référence à une clé primaire existante.
Sélection	Permet de rechercher des lignes dans une table à partir d'un critère de sélection. Le résultat est identique à la table de départ.
Projection	Permet de sélectionner des colonnes d'une table. À l'issue de cette opération, la table résultat peut contenir des tuples identiques.
Jointure	Permet de combiner 2 tables en se basant le plus souvent sur l'égalité de la clé étrangère de l'une et la clé primaire de l'autre. La table résultat contient les attributs des 2 tables de départ.