Grille de structuration d'un cours en ligne

Langage relationnel (SQL)

Enseignant
COULIBALY MOUSSA
Enseignant vacataire
Ingénieur conception des SI au CHU-T
moussa.coulibaly97@gmail.com
70278896

A- Système d'entrée

- Présentation

Ce cours sur le langage relationnel(SQL) permet de comprendre les concepts liés à la théorie des enregistrements des données, de connaître et de maîtriser le principe d'organisation et de manipulation des données selon le modèle de CODD et enfin approfondir le langage SQL.

- Objectifs du module
  - Objectif général

Au terme de ce module, vous serez à mesure de :

- comprendre les concepts liés à la théorie des enregistrements des données ;
- connaître le principe d'organisation et de manipulation des données selon le modèle de COD ;
- maîtriser le principe d'organisation et de manipulation des données selon le modèle de COD ;
- Savoir utiliser le langage relationnel ;
- traduire l'algèbre relationnel en langage SQL ;

  - Objectifs spécifiques

Au terme de cette séquence, vous serez capable de :

- **OG1 : comprendre les concepts liés à la théorie des enregistrements des données**
  - Définir un attribut ou colonne ;
  - Décrire un attribut ou colonne ;
  - Distinguer les différents domaines de valeurs d'un attribut ou colonne ;
  - Connaître un schéma relationnel ou relation ;
  - Définir une cardinalité d'un schéma ;
  - Connaître le degré d'un schéma ;
  - Citer les différentes contraintes ;
  - Définir les différentes contraintes;
  - Comprendre la notion de tuplet ou extension.
- **OG2 : connaître et de maîtriser le principe d'organisation et de manipulation des données selon le modèle de CODD**
  - *Comprendre les différentes parties de la Structure du système ;*
  - *Distinguer les différentes parties de la Structure du système ;*
  - *Citer les différents supports physiques des données ;*
  - *Connaître les différents supports physiques des données ;*

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

- *Maitriser l'organisation des fichiers ;*
- *Définir et connaître la gestion de la mémoire tampon (disque) ;*
- *Définir et connaître la gestion de la mémoire tampon (disque)*
- *Savoir la Répartition des bases relationnelles en fichiers;*
- *Revoir et maîtriser le langage de définition des données;*
- *Revoir et maîtriser l*e langage de manipulation des données ;

- **OG3 : Savoir utiliser le langage relationnel**
  - *Comprendre et utiliser les opérations de Base :*
    - *Union ;*
    - *Différence ;*
    - *Produit ;*
    - *Projection ;*
    - *Sélection ;*
  - *Savoir utiliser les opérations de base pour construire les opérations additionnelles :*
    - *Intersection ;*
    - *Quotient ;*
    - *Jointure ;*
    - *Jointure naturelle ;*
    - *Semi-jointure ;*
- **OG4 : traduire l'algèbre relationnel en langage SQL ;**
  - *employerles opérations de Base :*
    - *Union ;*
    - *Différence ;*
    - *Produit ;*
    - *Projection ;*
    - *Sélection ;*
  - *Savoir utiliser les opérations de base pour construire les opérations additionnelles :*
    - *Intersection ;*
    - *Quotient ;*
    - *Jointure ;*
    - *Jointure naturelle ;*
    - *Semi-jointure ;*

- Scénario pédagogique

M. Moussa COULIBALY					
US01 : comprendre les concepts liés à la théorie des enregistrements des données	• Définir un attribut ou colonne ; • Décrire un attribut ou colonne ; • Distinguer les différents domaines de valeurs d'un attribut ou colonne ;	**contenu** *Le modèle du CODD*	**Activités d'apprentissage** *Lire le support de cours et le cour en ligne,* *Participer à la séance de chat du 01*	**Ressources pédagogiques** *Support de cours ou cours en ligne*	**Modalité d'évaluation** *Examen en ligne pour une durée de 2h*

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

	○ Connaître un schéma relationnel ou relation ; ○ Définir une cardinalité d'un schéma ; ○ Connaître le degré d'un schéma ; ○ Citer les différentes contraintes ; ○ Définir les différentes contraintes; ○ Comprendre la notion de tuplet ou extension.		*Participer au forum à tout moment* *Faire les QCM et les activités en ligne* *Faire des recherche pour complément*		
• **US02 : connaître et de maîtriser le principe d'organisation et de manipulation des données selon le modèle de CODD**	• *Comprendre les différentes parties de la Structure du système ;* • *Distinguer les différentes parties de la Structure du système ;* • *Citer les différents supports physiques des données ;* • *Connaître les différents supports physiques des données ;* • *Maitriser l'organisation des fichiers ;* • *Définir et connaître la gestion de la mémoire tampon (disque) ;* • *Définir et connaître la gestion de la mémoire tampon (disque)* • *Savoir la Répartition des bases*	Modèle du CODD et entité/association	*Lire le support de cours et le cour en ligne,* *Participer à la séance de chat du 01* *Participer au forum à tout moment* *Faire les QCM et les activités en ligne* *Faire des recherche pour complément*	*Support de cours ou cours en ligne*	*Examen en ligne pour une durée de 2h*

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

	relationnelles en fichiers; • *Revoir et maîtriser le langage de définition des données;* • *Revoir et maîtriser l*e langage de manipulation des données ;				

US03 : Savoir utiliser le langage relationne	• *Comprendre et utiliser les opérations de Base :*
	  • *Union ;*
	  • *Différence ;*
	  • *Produit ;*
	  • *Projection ;*
	  • *Sélection ;*
	• *Savoir utiliser les opérations de base pour construire les opérations additionnelles :*
	  • *Intersection ;*
	  • *Quotient ;*
	  • *Jointure ;*
	  • *Jointure naturelle ;*
	  • *Semi-jointure ;*
		Modèle du CODD et entité/association	*Lire le support de cours et le cour en ligne,*
			Participer à la séance de chat du 01
			Participer au forum à tout moment
			Faire les QCM et les activités en ligne
			Faire des recherche pour complément
				Support de cours ou cours en ligne	*Examen en ligne pour une durée de 2h*

US04 : traduire l'algèbre relationnel en langage SQL ;	○ *employerles opérations de Base :*
	  ▪ *Union ;*
	  ▪ *Différence ;*
	  ▪ *Produit ;*
	  ▪ *Projection ;*
	  ▪ *Sélection ;*
	○ *Savoir utiliser les opérations de base pour construire les opérations additionnelles :*
		Modèle du CODD et entité/association	*Lire le support de cours et le cour en ligne,*
			Participer à la séance de chat du 01
			Participer au forum à tout moment
			Faire les QCM et les activités en ligne
				Support de cours ou cours en ligne	*Examen en ligne pour une durée de 2h*

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

	▪ *Intersection ;* ▪ *Quotient ;* ▪ *Jointure ;* ▪ *Jointure naturelle ;* ▪ *Semi-jointure ;*				

B- Système d'apprentissage

- Unité d'apprentissage 1 (OG1) : comprendre les concepts liés à la théorie des enregistrements des données

  - contenu

    - Objectif

Définir et distinguer tous les concepts liés à la théorie des enregistrements selon CODD. Dans cette partie vous allez apprendre et définir un attribut, les domaines de valeur, une relation, une cardinalité, un dégré, les différentes contraintes et la notion d'extension.

- Définitions

Un attribut est un nom donné à une colonne d'une relation ou table et prenant des valeurs dans un domaine.
Exemple : nom_cli.

Un domaine est l'ensemble des valeurs que peut prendre une colonne.
Exemple : nom_cli prendra des valeurs dans le domaine caractère numérique noté AN et dans les SGBD par CHAR ou VARCHAR.
Liste des domaines existants et l'équivalent dans les SGBD

N°	Langage naturel	Équivalent SGBD
01	Caractère	CHAR
02	Caractère variable	VARCHAR
03	Numérique	NUMBER, REAL, INTEGER, SMALLINT
04	Date	DATE, TIME
05	Booléen	BOOLEAN

Notions de clé primaire

Clé primaire : Groupe d'attributs minimum qui détermine un tuple d'une manière unique dans la table
– Exemple de clés :
• le numéro du Bon
• Le matricule étudiant
– La clé de la table OUVRAGES est l'attribut « cote », car la cote permet de déterminer de façon unique une ligne de la table.
– ATTENTION : la clé se détermine par rapport à toutes les valeurs possibles de l'attribut (ou les attributs) formant la clé primaire, et surtout pas par rapport aux valeurs déjà saisies
• Remarque : toute table doit obligatoirement avoir une clé primaire.

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

Clé Étrangère

• Les attributs cote et NumAuteur de la table ECRIT proviennent en fait respectivement des tables OUVRAGES et AUTEURS. Ces deux Attributs sont clés primaires dans chacune de ces tables.
• Définition : Nous appelons Clé étrangère toute clé primaire apparaissant dans une autre table.
• Exemple
– NumAuteur est une clé étrangère dans la table ECRIT
– cote est aussi une clé étrangère dans ECRIT
Par convention, Une clé étrangère est soulignée en pointillé (et/ou mise en italique)
Attention : la notion de clé est toujours liée à une table, un attribut (ou groupe d'attributs) est clé primaire, ou clé étrangère dans une table donnée.

Une Relation est composé de la liste des attributs représenté sous forme de tuplet.
Exemple : Client (num_cli, nom_cli, tel_cli, mail_cli)

Un schéma relationnel ou table est composé de la liste des attributs représenté sous forme de graphique.
Exemple : client

num_cli	nom_cli	tel_cli	mail_cli

Sous MySQL

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	**num_cli**	int(4)			Non	*Aucun(e)*		AUTO_INCREMENT
2	**nom_cli**	varchar(100)	latin1_swedish_ci		Non	*Aucun(e)*		
3	**tel_cli**	varchar(20)	latin1_swedish_ci		Non	*Aucun(e)*		
4	**mail_cli**	varchar(100)	latin1_swedish_ci		Non	*Aucun(e)*		

```
CREATE TABLE `client`.`client` ( `num_cli` INT(4) NOT NULL AUTO_INCREMENT , `nom_cli` INT(50) NOT NULL , `tel_cli` INT(20) NOT NULL , PRIMARY KEY (`num_cli`)) ENGINE = MyISAM;

ALTER TABLE `client` CHANGE `nom_cli` `nom_cli` VARCHAR(100) NOT NULL;

ALTER TABLE `client` ADD `mail_cli` VARCHAR(100) NOT NULL AFTER `tel_cli`;

ALTER TABLE `client` DROP `mail_cli`;
```

Un tuplet ou un enregistrement ou une **extension** est l'ensemble des valeurs pris par chaque colonne ou attribut.
Exemple

num_cli	nom_cli	tel_cli	mail_cli
1	DAO Ali	70876590	d.ali@uni-virtuel.bf

Une cardinalité d'un schéma est le nombre de tuplets ou d'enregistrement du schéma.
Exemple

num_cli	nom_cli	tel_cli	mail_cli
1	DAO Ali	70876590	d.ali@uni-virtuel.bf
2	KABORE Jean	76890900	K.jean@uni-virtuel.bf

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

3	BOLY Kadi	78097800	b.kadi@uni-virtuel.bf

Connaître le degré d'un schéma est le *nombre d'attributs (n) dans son schéma relationnel. Exemple le degré du schéma client est 4 :*

Citer et définir les différentes contraintes ;

Trois types de C.I. obligatoires

– Contrainte de clé : une relation doit posséder une clé primaire

– Contrainte d 'entité : un attribut d 'une clé ne doit pas posséder de valeurs nulles (vides)

– Contrainte de référence (pour les clés étrangères), c'est une contrainte exprimée entre deux tables. Tout tuple d'une relation faisant référence à une autre relation doit se référer à un tuple qui existe

• Intuitivement, cela consiste à vérifier que l'information utilisée dans un tuple pour désigner un autre tuple est valide, notamment si le tuple désigné existe bien

• En d'autre terme, quand on désigne un attribut comme clé étrangère, les seules valeurs que peut prendre cet attribut sont celles qui sont déjà saisies dans la table qu'il référence

Un des avantages des bases de données par rapport à une gestion de fichiers traditionnelle réside dans la possibilité d'intégrer des contraintes que doivent vérifier les données à tout instant.

• Exemple : on souhaite poser les contraintes suivantes :

– le nombre d'exemplaire de chaque OUVRAGE doit être supérieur à 0 (zéro)

– Chaque OUVRAGE doit avoir au moins un auteur

– Etc.

• Ceci est possible grâce à la notion de contraintes d'intégrité

• Définition :

– Contraintes d 'intégrité « sont des assertions qui doivent être vérifiées à tout moment par les données contenues dans la base de données »

Contrainte optionnelle

– Contrainte de domaine : liée au domaine de définition d 'un attribut.

– Exemple: NbExemplaire > 0

• Les contraintes d'intégrité sont vérifiées (exécutées) à chaque mise à jour de la base de données (ajout, suppression ou modification d'un tuple). Si, lors d'une mise à jour une contrainte n'est pas satisfaite, cette mise à jour ne peut pas avoir lieu.

- situations d'apprentissages/activités locales

Exercice 1

#	Nom	Type
1	num_cli	int(4)
2	nom_cli	varchar(100)
3	tel_cli	varchar(20)
4	mail_cli	varchar(100)

BP : 64 Ouagadougou Cité AN II – BURKINA FASO

Téléphone : 02 20 31 31 - Site web : www.uv.bf

Courriel : info@uv.bf

En considérant l'image ci-dessus :

1- la clé de la table est
a- num_cli
b- nom_cli
c- tel_cli
d- mail_cli

2-domaine de la clé est
a- alpha-numéric
b- booléen
c- date
d- entier
soit la relation client ci-dessous :

num_cli	nom_cli	tel_cli	mail_cli
1	DAO Ali	70876590	d.ali@uni-virtuel.bf
2	KABORE Jean	76890900	K.jean@uni-virtuel.bf
3	BOLY Kadi	78097800	b.kadi@uni-virtuel.bf

3- donner le schéma de la relation:
4- donner le dégre de la relation client :
5- donner un tuplet ou enregistrement :
6- donner la cardinalité de la relation :
7- citer et définir les 3 types de contraintes d'intégrité.

- Unité d'apprentissage 2 (OG2) **connaître et de maîtriser le principe d'organisation et de manipulation des données selon le modèle de CODD**

  - Contenu

    - Objectif

Comprendre et distinguer les différentes parties de la structure du système, connaître et citer les différents supports physiques des données, maîtriser l'organisation des fichiers, définir et connaître la gestion de la mémoire tampon, savoir la répartition des bases relationnelles en fichiers et revoir et maîtriser le LDD et LMD.

- Comprendre et distinguer les différentes parties de la Structure du système

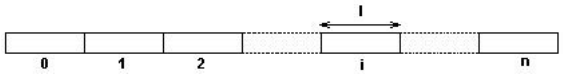
on passera en revue les principales organisations de fichiers en supposant que le lecteur a déjà des connaissances sur les supports physiques de stockage; en particulier, on supposera que les données sont physiquement enregistrés sur un support à accès sélectif (typiquement un disque magnétique). On admettra également que les organisations de fichiers permettent d'obtenir un enregistrement à partie de son adresse relative (encore appelée adresse logique) qui est un nombre entier donnant le déplacement par rapport au début de fichier; ceci nécessite évidemment un dispositif de conversion permettant d'obtenir l'adresse réelle à partir de l'adresse relative et de l'identification du fichier. On supposera que ceci est réalisé par le système de gestion du disque. Les organisations de fichiers doivent donc répondre au problème essentiel suivant : étant donné un identificateur d'enregistrement, comment obtenir son adresse relative ?
On repère chaque enregistrement par un identificateur ou identifiant appelé clé : la connaissance de la clé (qui peut ou non faire partie de l'enregistrement) doit permettre la connaissance de l'adresse relative de l'enregistrement. On

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

distingue 3 types d'organisation des enregistrements :

- Organisation relative

Dans l'organisation relative, les enregistrements sont supposés avoir une longueur fixe l et sont stockés séquentiellement dans l'ordre croissant des adresses relatives. Ils sont numérotés dans l'ordre de cette séquence. Ce numéro d'ordre est la clé de l'enregistrement:



L'adresse relative ai de l'enregistrement de clé i (en fait l'adresse relative de son début) est donc obtenue très simplement par :

$$ai = i \times l$$

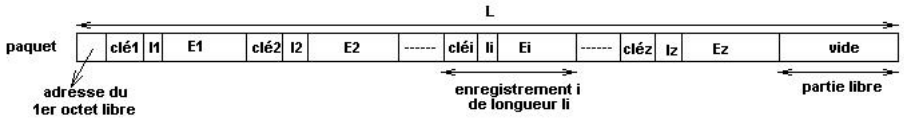
L'avantage quasi-unique de la méthode est la simplicité. Les inconvénients sont, en contre-partie, importants. Tout d'abord les enregistrements doivent être de longueur fixe. Deuxièmement, il faut associer un numéro à un enregistrement ce qui est plus difficile à faire qu'à dire.

Notons que l'on peut, à la limite, utiliser des enregistrements de longueur variable, mais il faut alors prendre pour l la longueur maximale des enregistrements ce qui conduit à de nombreux trous et donc une perte non négligeable de place de stockage.

- Organisation directe

Ici, on suppose que la clé est un identificateur figurant dans l'enregistrement. Le fichier est découpé en p paquets (buckets) de taille fixe L numérotés de 0 à N. Un paquet comprend plusieurs enregistrements qui peuvent être de taille variable. En général un paquet correspond à un bloc physique qui est l'unité élémentaire de transfert entre la mémoire secondaire (disque) et la mémoire principale. L'accès à un enregistrement se fait alors en deux temps :

Les enregistrements sont stockés séquentiellement dans le paquet de la manière suivante :



Une fois que l'adresse du paquet contenant l'enregistrement recherché est connue, la procédure g consiste à parcourir séquentiellement les enregistrements du paquet jusqu'à trouver celui qui correspond à la clé i on procède par comparaison.

f est la fonction de hachage (Hash code). Le choix de f doit correspondre à une répartition uniforme des enregistrements dans les paquets. On sait que cela est fait de manière satisfaisante si les enregistrements sont répartis au hasard. Pour cette raison f devrait idéalement aboutir à la production d'un nombre aléatoire. C'est pour cette raison que souvent, organisation directe et organisation aléatoire (random) sont synonymes. De nombreux procédés

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

définissent la fonction f. Pour simplifier nous donnerons ici le plus simple : celui du modulo. Supposons que le fichiers comprennent 57 paquets qui auront donc comme adresses les valeurs 0 à 56. Pour obtenir l'adresse relative du paquet qui contient un enregistrement de clé (supposée numérique) donnée, on effectue la division entière de celle-ci par 57 et on prend le reste.

Ainsi l'enregistrement de clé 100 se trouve dans le paquet d'adresse 43

l'enregistrement de clé 94 se trouve dans le paquet d'adresse 37

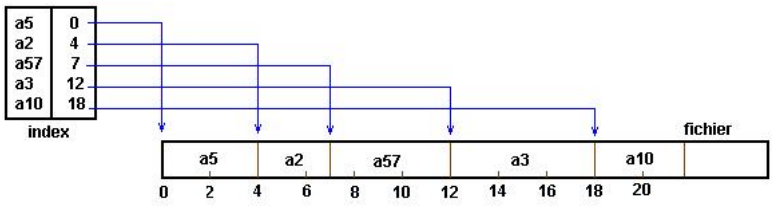
l'enregistrement de clé 121 se trouve dans le paquet d'adresse 7

Les avantages notables de cette organisation sont la simplicité et surtout les bonnes performances en fonctionnement normal. Les inconvénients principaux sont au nombre de deux :

1. le fichier doit avoir une taille quasi-fixe puisque le nombre de paquets est fixé
2. en cas de paquet plein, il y a débordement ce qui nécessite des procédures particulières qui dégradent fortement les performances.

- Organisations indexées

Le principe des organisations indexées est d'associer dans une table la clé et l'adresse relative des enregistrements ou des paquets. Cette table est appelée index ou table d'index. Ceci est analogue à la table des matières d'un livre.



L'accès à un enregistrement suppose les opérations successives suivantes :

a) accès à l'index

b) recherche de la clé et donc de l'adresse relative

c) conversion de l'adresse relative en adresse absolue (système de gestion de la mémoire secondaire)

d) accès à l'enregistrement en mémoire secondaire

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

e) transfert de l'enregistrement en mémoire principale

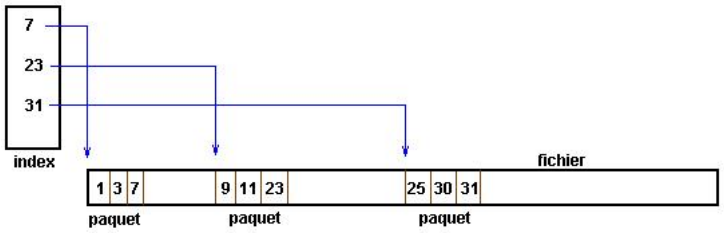
il existe de nombreuses variantes de l'utilisation des index ce qui correspond à différentes organisations de fichiers ; le tableau ci-dessous schématise la situation courante :

			fichier : trié	fichier : non trié
index	dense	trié	possible	IS3
		non trié		possible
	non dense	trié	ISAM, VSAM	
		non trié		

On voit que deux notions sont déterminantes : celle d'index trié et celle d'index dense.

Un index trié est un index dans lequel les clés sont rangés dans un certain ordre (chronologique, alphabétique, numérique,...) de manière à permettre une recherche dichotomique. Le fichier, quant à lui, peut aussi être trié ou non trié. Examinons l'intérêt de l'index trié. Supposons que l'index comporte n clés réparties en blocs de b clés (ce qui dans la pratique signifie qu'une accès disque permet de lire b clés au plus). Il y a donc n/b blocs.
Si l'index n'est pas trié, il faudrait en moyenne lire n/2b blocs (donc accès)pour obtenir une clé donnée. Si au contraire, l'index est trié, la recherche dichotomique correspond en moyenne à une lecture de log2(n/b) blocs(donc accès) pour obtenir une clé donnée. Un exemple numérique permet de mieux voir la différence entre les deux méthodes. Supposons que n = 10^6 clés et que b = 10^2 clés.. Alors on obtient :
pour un index non trié : 5 000 accès !
pour un index trié : 13 accès

La densité d'un index est le rapport d du nombre de clés sur le nombre d'enregistrements. Si d = 1, cela signifie qu'il y a une clé par enregistrement : on dit alors que l'index est dense. Par contre d<1 correspond à une situation où une clé correspond à un groupe d'enregistrements (un paquet): l'index est dit alors non dense.

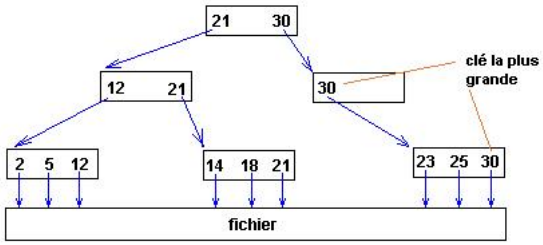


Un index est en soi un fichier que l'on peut atteindre avec un autre index, et ainsi de suite. On aboutit ainsi à la notion d'index multiniveau (ou hiérarchisé). La représentation des correspondances est alors un arbre dont un exemple est

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

donné ci-dessous (les clés sont supposées numériques).

NB : Dans les bases de données, un index est une structure de données utilisée et entretenue par le système de gestion de base de données (SGBD) pour lui permettre de retrouver rapidement les données.



B-arbres

Pour faciliter la gestion des index multiniveaux, on utilise la notion de B-arbre (arbre "balancé" ou "équilibré"). Un B-arbre d'ordre m est défini comme suit :

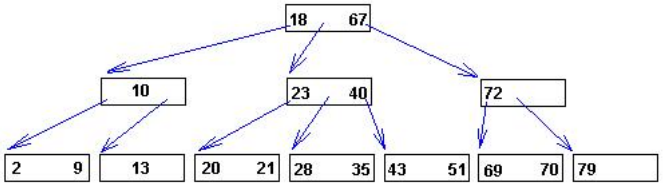
1) Chaque noeud contient k clés triées de gauche à droite avec m <= k <= 2m sauf pour la racine qui contient entre 1 et 2m clés.

2) Un noeud est soit terminal, soit possède k+1 fils.

3) Le ième fils possède des clés comprises entre la (i-1)ème et la ième clé du père.

4) L'arbre est équilibré : tous les noeuds terminaux sont au même niveau.

Voici par exemple un B-arbre d'ordre 1



- Connaître et citer les différents supports physiques des données

Plusieurs types de stockage sont utilisés par les SGBD.
Classés par :

1. ordre de rapidité d'accès;
2. coût de stockage des informations;
3. fiabilité.

Le type de mémoire que l'on rencontre sur les machines sont les suivantes:

1. Mémoire cache :
   très rapide et de petite taille, elle est ignorée par leSGBD
2. Mémoire principale :
   a. utilisée en permanence par la machine;

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

b. ne peut contenir la totalité de la BD;
c. perdue en cas de crash .

3. Mémoires à accès séquentiel (bandes magnétiques).
   a. utilisées principalement pour les archivages et les sauvegardes;
   b. accès très lent car il est séquentiel;
   c. très fiables.
4. Mémoire à accès direct :
   a. support utilisé pour le stockage des BD;
   b. des fractions de la base sont lues en mémoire principale pour le traitement puis réécrites sur disque;
   c. les données peuvent être lues dans un ordre quelconque;
   d. elle résiste aux pannes de système (sauf au "crash disk");
   e. elle nécessite d'archiver régulièrement sur bandes magnétiques. (« backup »)

- Maitriser l'organisation des fichiers

Sur les disques, les données sont réparties dans des fichiers.
Un fichier est une suite d'enregistrements (ou de données)…

1. organisés de façon logique
2. et découpés en blocs.

Même si la dimension des blocs dépend:

1. des caractéristiques de l'unité de disque
2. et du système d'exploitation,

… la taille des enregistrements varie elle aussi.
Il existe 2 modes de découpage des données en enregistrements:

1. enregistrements de longueur fixe;

Cette structure semble très simple puisque tous les enregistrements de la relation commande et Client auront la même longueur.
Problèmes:

a. l'effacement d'un enregistrement est difficile: il y a le problème de remplacement de l'information avec des enregistrements de longueurs différentes.
b. un enregistrement peut être à cheval entre deux blocs;

2. enregistrements de longueur variable.

Deux façons de traiter les enregistrements de longueur variable:

1. Par espaces réservés:
   a. On détermine la longueur maximale de l'enregistrement à longueur variable.
   b. À sa création, l'enregistrement est créé à sa longueur maximale.
   c. Les espaces non utilisés sont remplis de symboles spéciaux.
2. Par pointeurs :
   a. Dans le fichier, on ajoute un nouveau champ contenant des pointeurs aux enregistrements.
   b. À sa création, l'enregistrement de longueur variable est représenté par une liste enregistrements de longueur fixe.
   c. Les enregistrements appartenant à une même agence sont chaînés.

Imprégner

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

- Définir et connaître la gestion de la mémoire tampon (disque)

• Rôle du gestionnaire de buffer : passage des pages du disque vers la mémoire (et inversement) ou placer, au moment voulu, une page du disque vers la mémoire et inversement :

- Politique de remplacement (ex. LRU : least recently used)
- Gestion des pages mises à jour
- Partition de la mémoire
- Vérification des droits sur les pages

- Revoir et maîtriser le langage de définition des données (LDD ref conception et propriété BD);
- Revoir et maîtriser le langage de manipulation des données ((LMD ref conception et propriété BD) ;

- situations d'apprentissages/activités locales

Exercice 1

1- citer, définir et critiquer les différentes d'organisation des données sur l'ordinateur.
2- quelle est la technique de gestion des mémoires tampons dans la gestion des données des SGBD relationnels ?
3- . Supposons que n = 10^6 clés reparties en b = 10^2 clés. Calculer les accès selon
   a- les indexes triés
   b- les indexes non triés
4- quelle notion utilisent les SGBD relationnels pour gérer les indexes multi-niveaux ?
5- citer et définir les types de mémoires entrant dans le stockage de données ?

- Unité d'apprentissage 3 (OG3) : **utiliser le langage relationnel**

- contenu

Il existe 2 catégories d'opérateurs:

1. opérateurs de base : union, différence, produit, projection, sélection,
2. opérateurs additionnels qui s'expriment à partir des opérateurs de base : intersection, quotient, jointure,jointure naturelle, semi-jointure.

- Les (05) cinq opérateurs de base
  - Union : ∪

Soit r et s deux relations compatibles, c'est-à-dire deux parties du même produit E1 x ... x Ek, alors r ∪ s est l'ensemble des tuples qui sont à la fois dans r ou dans s. Exemple :

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

r =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon

s =

Nom	Numéro	Ville
Martin	98213	Paris
Duval	95565	Lille
Dupont	99123	Lyon

alors r ∪ s =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon
Duval	95565	Lille

si r et s contiennent respectivement m et n tuples, l'union r ∪ s contient au plus m+n tuples.

- Différence : -(MINUS or EXCEPT)

On suppose comme pour l'union que les deux relations r et s sont compatibles. Alors r - s est l'ensemble des tuples qui sont dans r mais pas dans s.

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

r = s =

Nom	Numéro	Ville
Durand	99345	Orléans
Duval	95564	Rennes
Martin	98214	Paris

- Produit : x

Si r est une relation d'arité p et s une relation d'arité q, alors r x s est l'ensemble des (pxq)-uplets formés en mettant bout à bout un tuple de r et un tuple de s.

r =

A	B
a1	b1
a2	b2
a3	b3

s =

C	D
c1	d1
c2	d2

alors r x s =

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b2	c1	d1
a2	b2	c2	d2
a3	b3	c1	d1
a3	b3	c2	d2

Remarques :
Si r et s contiennent respectivement m et n tuples, r x s contient exactement m x n tuples.
Lorsque r et s ont des noms d'attributs en commun, il faut les renommer pour éviter les confusions. Si r a les attributs A et B, s a les attributs B et C, on pourra désigner les attributs de r x s par A, B1, B2, C.
C'est le cas dans le produit r x r de r par elle-même ; si r a les attributs A et B, r x r a les attributs A1, A2, B1, B2.

- Projection : π

Soit r une relation d'arité k ayant pour attributs U = {A1, A2, ... Ak} . Soit i1, i2, ... ip des entiers distincts compris entre 1

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

et k. La projection de r sur les attributs de numéros i1, i2, ... ip notée :
π i1, i2, ... ip est obtenue en ne gardant des tuples de r que la partie correspondant aux attributs de numéros i1, i2, ... ip.

r =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon

alors $\Pi_{1,3}(r)$ =

Nom	Ville
Durand	Orléans
Martin	Paris
Duval	Rennes
Dupont	Lyon

Remarque : si la relation de départ a n tuples, la relation provenant de la projection a au plus n tuples, car plusieurs tuples peuvent se projeter sur le même tuple. C'est ici le cas de <Martin, 98213, Paris> et <Martin, 98214, Paris> qui produisent tous les deux <Martin, Paris>.
Les attributs de la projection peuvent être désignés par leurs noms plutôt que par leurs numéros. Dans le cas ci-dessus, la projection P **π** 1,3 (r) se note également P **π** nom, ville (r).

- Sélection : σ

Soit r une relation et f une formule logique construite à partir de

- constantes
- des attributs de r désignés par leurs noms ou leurs numéros,
- les opérateurs de comparaisons =, ≠, <, ≤, >, ≥
- les connecteurs logiques ÚV,Λ,¬ Ø

la relation **σ** f (r) désigne l'ensemble des tuples u de r tels que f(u) donne la valeur "vrai".
Remarque : la relation **σ** f (r) a en général moins de tuples que r.

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

r =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon

alors, $\sigma_{nom = Martin}(r)$ =

Nom	Numéro	Ville
Martin	98213	Paris
Martin	98214	Paris

$\sigma_{nom = Martin \vee ville = Orléans}(r)$ =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Martin	98214	Paris

Lorsque l'on souhaite désigner un attribut par son numéro, on précède son nom par le caractère $ pour éviter la confusion avec une constante. Ainsi **σ** nom= "Martin" (r) se récrit σ $1= "Martin"(r).

- Opérateurs additionnels
  - Intersection : ⋂ (INTERSECT)

Etant donné deux relations compatibles r et s, l'intersection r ⋂ s est constituée des tuples qui sont à la fois dans r et dans s.

- Opérateurs additionnels
  - Intersection : ⋂ (INTERSECT)

Etant donné deux relations compatibles r et s, l'intersection r ⋂ s est constituée des tuples qui sont à la fois dans r et dans s.

L'intersection s'exprime avec les opérateurs de base : r ⋂ s = r - (r - s) = s - (s - r).

- Quotient : ÷(DIVIDE)

Soit r une relation d'arité n et s une relation dont les attributs Us sont une partie stricte de Ur ; s a donc une arité p inférieure à n. Alors r ÷ s est l'ensemble des (n-p)-uplets appelés t portant sur l'ensemble d'attributs UT= UR - US tels

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

que, pour tout tuple u de s, le tuple t.u sur UR obtenu en concaténant t et u est dans r (en faisant abstraction de l'ordre des attributs).

r =

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b2	c2	d2
a3	b3	c1	d1
a3	b3	c2	d2
a1	b1	c3	d3

et s =

C	D
c1	d1
c2	d2

alors r ÷ s =

A	B
a1	b1
a3	b3

car le produit (r ÷ s) × s donne les tuples
a1 b1 c1 d1
a1 b1 c2 d2
a3 b3 c1 d1
a3 b3 c2 d2
qui sont tous dans r
a2 b2 n'est pas dans le quotient car le produit par s donne les deux tuples
a2 b2 c1 d1
a2 b2 c2 d2
le premier d'entre eux ne fait pas partie de r.
Remarque : si l'on s'arrange pour que les attributs de s soient les derniers attributs de r, alors r ÷ s est la plus grande relation q sur UR - US telle que r ⊇ q × s.
Le quotient s'exprime avec les opérateurs de base. Pour l'exemple ci-dessus, r ÷ s = **π** A,B (r) - **π** A,B ((**π** A,B (r) ×s) - r).

- jointure : ►◄c

r ►◄**s** c'est l'ensemble des tuples de r × s qui satisfont la condition c. C'est donc □c (r × s)

- jointure naturelle : ►◄

On suppose que r et s ont des attributs nommés, et qu'un certain nombre d'attributs sont partagés par les deux relations, alors la jointure naturelle r►◄s est obtenue en

- calculant le produit cartésien r × s
- sélectionnant les tuples dans lesquels les attributs de mêmes noms ont mêmes valeurs,
- supprimant les valeurs doubles.

Pour résumer r ►◄ s = **π** Ur ∪ Us (**σ** = (r × s)) où **σ** = (r × s) sélectionne dans le produit les tuples qui ont mêmes

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

valeurs sur les attributs en commun.

soit r =

A	B	C
a1	b1	c1
a2	b1	c1
a3	b1	c2
a4	b2	c3

et s =

B	C	D
b1	c1	d1
b1	c1	d2
b2	c3	d3

alors r ►◄ s =

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a2	b1	c1	d1
a2	b1	c1	d2
a4	b2	c3	d3

r ►◄ s s'exprime évidemment avec les opérateurs de base. Pour l'exemple ci-dessus cela donne
r ►◄ s = π 1,2,3,6 (σ $2=$4 ∧ $3=$5 (r × s))
Remarques :
Si r et s n'ont aucun attribut en commun, c'est-à-dire si Ur ⋂ Us = ∅, alors r ►◄ s = r × s
Si r et s ont tous leurs attributs en commun, c'est-à-dire si Ur = Us, alors r ►◄ s = r ⋂ s.

- Semi-jointure ►<

r ►< s est la projection sur les attributs de r de la jointure naturelle r ►◄ s.
C'est donc **π** Ur (r ►◄ s).

- situations d'apprentissages/activités locales

Considérons les trois schémas Etudiants, Inscriptions et Enseignements suivants :
Etudiants (nom-et, num-et, adr)
Inscriptions (num-et, module)
Enseignements (module, jour, heure, salle, nom-ens)

Etudiants

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

Nom-et	Num-et	adr
Martin	347	Orléans
Durand	1024	St Jean le Blanc
Duval	425	Orléans
Dupuis	248	St Cyr en val
Martin	504	Olivet

Inscriptions

Num-et	module
347	BD2
1024	BD3
248	RO
1024	BD3
504	SYS
347	RO

Enseignements

Module	Jour	Heure	Salle	Nom-ens
SYS	Lun	9	E05	Rousseau
BD2	Mar	10	E12	Proust
BD3	Lun	14	E05	Durand
BD2	Jeu	16	E08	Durand
RO	Lun	14	E08	Rousseau

Ces trois tables ont la signification suivante : l'étudiant de numéro 347 par exemple se nomme Martin et habite à Orléans ; il est inscrit aux modules BD2 et RO ; au titre de BD2 il suit les enseignements du mardi 10h en salle E12 assuré par Proust et du jeudi 16h en salle E08 assuré par Durand.
Quelques requêtes

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

R1- Numéros d'inscriptions des étudiants se nommant Martin :
R2- Modules d'enseignements qui ont lieu le lundi en salle E05
R3- Modules suivis par les étudiants se nommant Martin :
R4- Couples étudiant-enseignant tels que l'étudiant connaît l'enseignant : il suit au moins un cours assuré
R5- Etudiants qui suivent tous les enseignements de l'enseignant Durand

Réponses

R1 **π** num-et (**σ** nom-et = « Martin » (étudiants))
R2 **π** module (**σ** jour = lun ∧ salle = E05 (enseignements))
R3 **π** module (**σ** nom-et = « Martin » (étudiants ►◄ inscriptions))
π module ((**σ** nom-et = « Martin » (étudiants)) ►◄ inscriptions)

Réponses

R4 **π** nom-et, nom-ens (**π** nom-et, module (étudiants ►◄ inscriptions)►◄ **π** module, nom-ens(enseignements))
R5 : (on suppose que lorsqu'un étudiant est inscrit à un module il suit tous les enseignements de ce module) :
R5 : on fait la division de l'ensemble des couples num-et - module par l'ensemble des modules dans lesquels enseigne Durand
π num-et, module (inscriptions) ÷ **π** module (**σ** nom-ens = « Durand » (enseignements))

- Unité d'apprentissage 4 (OG4) : SQL
  - *Comprendre et utiliser les opérations de Base :*

Syntaxe générale de la commande SELECT

Voici la syntaxe générale d'une commande SELECT :
*SELECT [ALL | DISTINCT] { * | expression [AS nom_affiché] } [, ...]*
FROM nom_table [[AS] alias] [, ...]
[WHERE prédicat]
[GROUP BY expression [, ...]]
[HAVING condition [, ...]]
[{UNION | INTERSECT | EXCEPT [ALL]} requête]
[ORDER BY expression [ASC | DESC] [, ...]]
En fait l'ordre SQL SELECT est composé de 7 clauses dont 5 sont optionnelles :
SELECT : Cette clause permet de spécifier les attributs que l'on désire voir apparaître dans le résultat de la requête.
FROM : Cette clause spécifie les tables sur lesquelles porte la requête.
WHERE : Cette clause permet de filtrer les n-uplets en imposant une condition à remplir pour qu'ils soient présents dans le résultat de la requête.
GROUP BY : Cette clause permet de définir des groupes (i.e. sous-ensemble ; cf.).
HAVING : Cette clause permet de spécifier un filtre (condition de regroupement des n-uplets) portant sur les résultats ().
UNION, INTERSECT et EXCEPT : Cette clause permet d'effectuer des opérations ensemblistes entre plusieurs résultats de requête (i.e. entre plusieurs SELECT) ().
ORDER BY : Cette clause permet de trier les n-uplets du résultat ().

Clause select

Pour illustrer par des exemples les sections qui suivent, nous utiliserons une table dont le schéma est le suivant :
employee(id_employee, surname, name, salary)

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

Cette table contient respectivement l'identifiant, le nom, le prénom et le salaire mensuel des employés d'une compagnie.

L'opérateur étoile (*)

Le caractère étoile (*) permet de récupérer automatiquement tous les attributs de la table générée par la clause FROM de la requête.

Pour acher la table employee on peut utiliser la requête :

SELECT * FROM employee

Les opérateurs DISTINCT et ALL

Lorsque le SGBD construit la réponse d'une requête, il rapatrie toutes les lignes qui satisfont la requête, généralement dans l'ordre ou il les trouve, même si ces dernières sont en double (comportement ALL par défaut). C'est pourquoi il est souvent nécessaire d'utiliser le mot clef DISTINCT qui permet d'éliminer les doublons dans la réponse.

Par exemple, pour acher la liste des prénoms, sans doublon, des employés de la compagnie,

il faut utiliser la requête :

SELECT DISTINCT name FROM employee

Les opérations mathématiques de base

Il est possible d'utiliser les opérateurs mathématiques de base (i.e. +, -, * et /) pour générer de nouvelles colonnes à partir, en générale, d'une ou plusieurs colonnes existantes.

Pour acher le nom, le prénom et le salaire annuel des employés, on peut utiliser la requête :

SELECT surname, name, salary*12 FROM employee

L'opérateur AS

Le mot clef AS permet de renommer une colonne, ou de nommer une colonne créée dans la requête.

Pour acher le nom, le prénom et le salaire annuel des employés, on peut utiliser la requête :

SELECT surname AS nom, name AS prénom, salary*12 AS salaire FROM employee

L'opérateur de concaténation

L'opérateur || (double barre verticale) permet de concaténer des champs de type caractères.

Pour acher le nom et le prénom sur une colonne, puis le salaire annuel des employés, on peut utiliser la requête :

SELECT surname || ' ' || name AS nom, salary*12 AS salaire FROM employee

La clause FROM

L'opérateur *AS*

Le mot clef AS permet de renommer une table, ou de nommer une table créée dans la requête (c'est à dire une sous-requête) afin de pouvoir ensuite y faire référence. Le renommage du nom d'une table se fait de l'une des deux manières suivantes :

FROM nom_de_table AS nouveau_nom

FROM nom_de_table nouveau_nom

Une application typique du renommage de table est de simplifier les noms trop long :

*SELECT * FROM nom_de_table_1 AS t1, nom_de_table_1 AS t2 WHERE t1.A_1 = t2.A_2*

Attention, le nouveau nom remplace complètement l'ancien nom de la table dans la requête. Ainsi, quand une table a été renommée, il n'est plus possible d'y faire référence en utilisant son

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

ancien nom. La requête suivante n'est donc pas valide :

SELECT * FROM nom_table AS t WHERE nom_table.a > 5

Sous-requête

Les tables mentionnées dans la clause FROM peuvent très bien correspondre à des tables résultant d'une requête, spécifiée entre parenthèses, plutôt qu'à des tables existantes dans la base de données. Il faut toujours nommer les tables correspondant à des sous-requêtes en utilisant l'opérateur AS.

Par exemple, les deux requêtes suivantes sont équivalentes :

SELECT * FROM table_1, table_2

SELECT * FROM (SELECT * FROM table_1) AS t1, table_2

La clause ORDER BY

Comme nous l'avons déjà dit, la clause ORDER BY permet de trier les n-uplets du résultat et sa syntaxe est la suivante :
ORDER BY expression [ASC | DESC] [, ...]
expression désigne soit une colonne, soit une opération mathématique de base (nous avons abordé ce type d'opérations dans « La clause SELECT ») sur les colonnes.
ASC spécifie l'ordre ascendant et DESC l'ordre descendant du tri. En l'absence de précision
ASC ou DESC, c'est l'ordre ascendant qui est utilisé par défaut.
Quand plusieurs expressions, ou colonnes sont mentionnées, le tri se fait d'abord selon les premières, puis suivant les suivantes pour les n-uplet qui sont égaux selon les premières.
Le tri est un tri interne sur le résultat final de la requête, il ne faut donc placer dans cette clause que les noms des colonnes mentionnés dans la clause SELECT.
La clause ORDER BY permet de trier le résultat final de la requête, elle est donc la dernière clause de tout ordre SQL et ne doit figurer qu'une seule fois dans le SELECT, même s'il existe des requêtes imbriquées ou un jeu de requêtes ensemblistes.
En l'absence de clause ORDER BY, l'ordre des n-uplet est aléatoire et non garanti. Souvent, le fait de placer le mot clef DISTINCT saut à établir un tri puisque le SGBD doit se livrer à une comparaison des lignes, mais ce mécanisme n'est pas garanti car ce tri s'effectue dans un ordre non contrôlable qui peut varier d'un serveur à l'autre.

La clause WHERE

Comportement

Comme nous l'avons déjà dit, la clause WHERE permet de filtrer les n-uplets en imposant une condition à remplir pour qu'ils soient présents dans le résultat de la requête ; sa syntaxe est la suivante :

WHERE prédicat

Concrètement, après que la table intermédiaire (i.e. virtuelle) de la clause FROM a été

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

construite, chaque ligne de la table est confrontée au prédicat prédicat afin de vérifier si la ligne satisfait (i.e. le prédicat est vrai pour cette ligne) ou ne satisfait pas (i.e. le prédicat est faux ou NULL pour cette ligne) le prédicat. Les lignes qui ne satisfont pas le prédicat sont supprimées de la table intermédiaire.

Le prédicat n'est rien d'autre qu'une expression logique. En principe, celle-ci fait intervenir une ou plusieurs lignes de la table générée par la clause FROM, cela n'est pas impératif mais, dans le cas contraire, l'utilité de la clause WHERE serait nulle.

Expression simple

Une expression simple peut être une variable désignée par un nom de colonne ou une constante. Si la variable désigne un nom de colonne, la valeur de la variable sera la valeur située dans la table à 'intersection de la colonne et de la ligne dont le SGBD cherche à vérifier si elle satisfait le prédicat de la clause WHERE. Les expressions simples peuvent être de trois types : numérique, chaîne de caractères ou

date. Une expression simple peut également être le résultat d'une sous-requête, spécifiée entre parenthèses, qui retourne une table ne contenant qu'une seule ligne et qu'une seule colonne (i.e. une sous-requête retournant une valeur unique).

Prédicat simple

Un prédicat simple peut être le résultat de la comparaison de deux expressions simples au moyen de l'un des opérateurs suivants :

= égal

!= di_érent

< strictement inférieur

<= inférieur ou égal

> strictement supérieur

>= supérieur ou égal

Dans ce cas, les trois types d'expressions (numérique, chaîne de caractères et date) peuvent être comparés. Pour les types date, la relation d'ordre est l'ordre chronologique.

Dans ce cas, la chaîne de caractères faisant l'objet du test est à gauche et correspond à une expression simple du type chaîne de caractères, il s'agit généralement d'un nom de colonne.

L'expression régulière, qui s'écrit entre apostrophe simple, comme une chaîne de caractères, est située à droite de l'opérateur.

Un prédicat simple peut enfin correspondre à l'un des tests suivants :

expr IS NULL test sur l'indétermination de expr

BP : 64 Ouagadougou Cité AN II – BURKINA FASO

Téléphone : 02 20 31 31 - Site web : www.uv.bf

Courriel : info@uv.bf

expr IN (expr_1 [, ...]) comparaison de expr à une liste de valeurs
expr NOT IN (expr_1 [, ...]) test d'absence d'une liste de valeurs
expr IN (requête) même chose, mais la liste de valeurs est le résultat d'une requête
expr NOT IN (requête) sous-requête qui doit impérativement retourner une table
ne contenant qu'une colonne
EXIST (requête) vraie si la sous-requête retourne au moins un n-uplet
vraie si au moins un n-uplet de la sous-requête vérifie la condition
expr operateur ANY (requête) comparaison « expr opérateur n-uplet » ; la sous-requête
doit impérativement retourner une table ne contenant qu'une colonne ; IN est équivalent à = ANY
vraie si tous les n-uplets de la sous-requête vérifient la condition
expr operateur ALL (requête) comparaison « expr opérateur n-uplet » ; la sous-requête
doit impérativement retourner une table ne contenant qu'une colonne
Dans ce tableau, expr désigne une expression simple et requête une sous-requête.

Prédicat composé

Les prédicats simples peuvent être combinés au sein d'expression logiques en utilisant les opérateurs logiques AND (et logique), OR (ou logique) et NOT (négation logique).

- *Union ;*

Expression1 SQL union Expression2 SQL et Expression1 et Expression2 sont identiques au point de vu schema

Exemple :

```
1 SELECT name, id_address FROM intermediary
2 UNION
3 SELECT name, id_address FROM entity;
```

- *Différence munis except;*

Expression1 SQL minus or except Expression2 SQL et Expression1 et Expression2 sont identiques au point de vu schema

```
1 SELECT name, id_address FROM entity
2 EXCEPT
3 SELECT name, id_address FROM intermediary ;
```

autrement

- *Produit ;*

*Select * from table1, table2*

- *Projection ;*

Soit un schema suivant table1(chp1, chp2,…, chpk,…chpn)
On choisit des champs non pas tous les champs dans la clause select pour faire une projection.
Select chp1, chp7, chpi,chpn from table1
***Exemple** client(numcli, nomcli, prencli,telcli)*
Select numcli nomcli from client

- *Sélection ;*

Select from where condition
Exemple
Select numcli, nomcli from client where telcli='70345678'

- *Savoir utiliser les opérations de base pour construire les opérations additionnelles :*

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

- *Intersection ;*

```sql
SELECT name, id_address FROM entity
INTERSECT
SELECT name, id_address FROM intermediary ;
```

autrement

```sql
SELECT *
FROM adulte
WHERE numero_securite_sociale IN (
    SELECT numero_securite_sociale FROM marseillais
) ;
```

- *Quotient ;*

Tables :
PARTICIPER EPREUVE
Athlète Epreuve Epreuve
Les athlètes ayant participé au moins à toutes les épreuves de la table EPREUVE
SELECT Athlète FROM PARTICIPER
GROUP BY Athlète
HAVING COUNT(*) =
(SELECT COUNT(DISTINCT Epreuve) FROM EPREUVE) ;

- *Jointure ;*

*Select * from table1 as t1, table2 as t2 where t1.chpcum=t2.chpcum*

- situations d'apprentissages/activités locales

Créez les tables du schéma relationnel vu en travaux dirigés section 3.5.

Schéma relationnel :

– film (num_film, num_realisateur, titre, genre, annee)

– cinema (num_cinema, nom, adresse)

– individu (num_individu, nom prénom)

– jouer (num_acteur, num_film, role)

– projection (num_cinema, num_film, jour)

N'oubliez surtout pas :

– de choisir correctement le domaine de définition (i.e. le type) de chacun des attributs ;

– de bien préciser la clé primaire de chaque relation ;

– les contraintes d'intégrité référentielles (i.e. les clefs étrangères).

1. Quel est le contenu de la table individu ?

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf

2. Quels sont les prénoms des individus en conservant les doublons ?
3. Quels sont les prénoms des individus en conservant les doublons, mais en les classant par ordre alphabétique ?
4. Quels sont les prénoms des individus sans doublons ?
Observez le résultat en effectuant un classement alphabétique et sans effectuer de classement.
5. Quels sont les individus dont le prénom est John ?
6. Quel est le nom des individus dont le prénom est John ?
7. Dressez la liste de toutes les associations possibles entre un individu et un film (il n'y a pas nécessairement de lien entre l'individu et le film qu'on lui associe). Observez le nombre de lignes retournées. Était-il prévisible ?
8. Quels sont les individus qui sont des acteurs ?
9. Dressez la liste de toutes les associations possibles entre un acteur et un film (il n'y a pas nécessairement de lien entre l'acteur et le film qu'on lui associe). Observez le nombre de lignes retournées.
10. Quels sont les noms et prénoms des réalisateurs ?
11. Quels sont les noms et prénoms des acteurs ?
12. Quels sont les noms et prénoms des acteurs qui sont également réalisateurs ?
Remarque : vous ne pouvez utiliser le mot clef INTERSECT puisque nous ne l'avons pas encore vu.
13. Quels acteurs a-t-on pu voir au cinéma Le Fontenelle depuis l'an 2000 ?
14. Quels sont les titres des films où Nicole Kidman a joué un rôle et qui ont été projetés au cinéma Le Fontenelle ?

- Références

Guézélou, P. (2006). Modélisation des données : Approche pour la conception des bases des données.
Hernandez, M. J. & Viescas, J. L. (2001). Introduction aux requêtes SQL. Eyrolles.
Kauman, J., Matsik, B. & Spencer, K. (2001). Maîtrisez SQL (Wrox Press, Ed.). CampusPress.
Labbé, C. (2002). Modéliser les données. Pratiko.
Petrucci, L. (2006). Base de données. Présentation projettée et travaux dirigés. (IUT GTR Villetaneuse)
Saglio, J.-M. (2002). Dominante informatique : Module bases de données.
SQL Anywhere Studio. (2005a). ASA - Guide de programmation : Programmation avec Embedded
SQL Anywhere Studio. (2005b). ASA - Guide de programmation : Utilisation de SQL dans les applications.
Szulman, S. (2005). Base de données et SGBD. Présentation projettée.

BP : 64 Ouagadougou Cité AN II – BURKINA FASO
Téléphone : 02 20 31 31 - Site web : www.uv.bf
Courriel : info@uv.bf