



M19 : SGBD 1

Langage de Manipulation de Donnée(LMD)

Formateur : Driouch (cfmoti.driouch@gmail.com)

Etablissement : OFPPT/CFMOTI 16/10/2011

<http://www.ista-ntic.net/>

Plan

- Définition
- Opérations de Sélection et projection
- Regroupement
- Produit cartésien et division
- Opérations de Jointure(ensembliste et prédicative)
- Les Fonctions intégré

Introduction

- LMD est le sous ensemble du langage SQL utilisé pour manipuler des données au sein d'un Système de Gestion de Bases de Données Relationnel.
- S.Q.L. signifie Structured Query Language.
- Il est issu de SEQUEL : Structured English Query Language, C'est le premier langage pour les S.G.B.D. Relationnels, Il a été développé par IBM en 1970 pour système R, son 1er SGBDR.
- S.Q.L. a été reconnu par l'ANSI (Association de Normalisation des Systèmes d'Information) puis s'est imposé comme norme, Il n'existe pas de S.G.B.D. Relationnel sans S.Q.L.
- S.Q.L. est un langage relationnel qui permet d'effectuer les tâches suivantes :
 - Définition et modification de la structure de la base de données
 - Interrogation et modification non procédurale (c'est à dire interactive) de la base de données
 - Contrôle de sécurité et d'intégrité de la base
 - Sauvegarde et restauration des bases

DRIOUCH B.

3

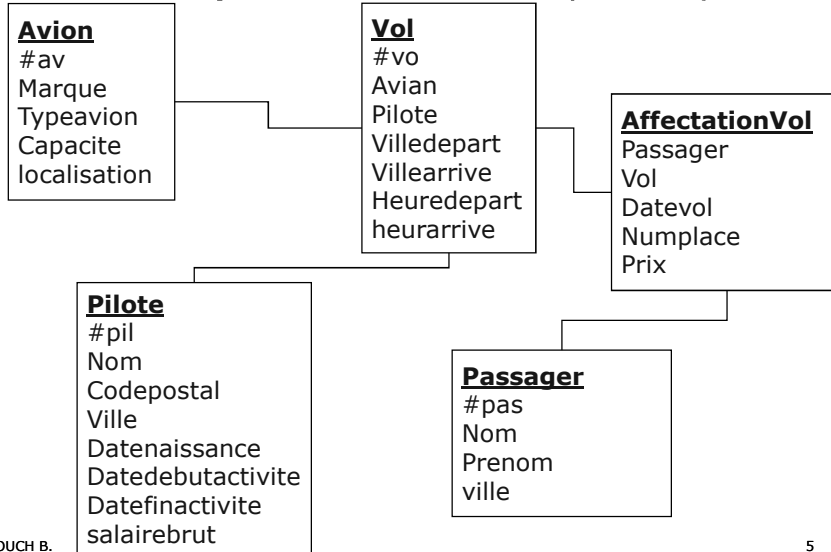
Opération de sélection

- L'instruction SELECT permet d'extraire des données et de les présenter triées et/ou regroupées suivant certains critères.
- Les enregistrements doivent vérifier certains critères exprimés dans des expression conditionnelles.
- Syntaxe de l'instruction SELECT
 - SELECT liste des attributs à afficher ou *
 - [DISTINCT] mot clé facultatif (élimination des doublons)
 - FROM liste des noms de tables
 - [WHERE] condition sur les attributs
 - [GROUP BY] nom de l'attribut de regroupement
 - [HAVING] condition de regroupement
 - [ORDER BY] critère de tri
 - [DESC] Ordre de tri descendant

DRIOUCH B.

4

Exemple VolAvion (MLD)



DRIOUCH B.

5

Opération de sélection

1. Requêtes sur une seule table:

```

SELECT      *
FROM        Vol
WHERE       HeureDepart > 16

```

	vol#	avion	pilote	villeddepart	villearrivee	heuredepart	heurearrivee
1	104	105	3	Toulouse	Grenoble	17.00	19.00
2	108	106	9	Brive	Paris	19.00	20.00
3	109	107	7	Paris	Lyon	18.00	19.00
4	113	105	2	Toulouse	Grenoble	17.00	19.00
5	116	101	4	Nice	Nantes	17.00	19.00

DRIOUCH B.

6

Opération de sélection

- Nous n'avons pas besoin de tous les attributs, et il s'agit d'un vol au départ de Toulouse et à destination de Grenoble, Nous allons donc restreindre les attributs retenus dans l'opération SELECT au moyen d'une **Projection** et compléter la clause **Where** :

```
SELECT      vol#,heuredepart,heurearrivee
FROM        vol
WHERE       heuredepart > 16
AND         villedepart LIKE 'Toulou%'
AND         villearrivee = 'Grenoble'
```

	vol#	heuredepart	heurearrivee
1	104	17.00	19.00
2	113	17.00	19.00

DRIOUCH B.

7

Opération de sélection

■ Utilisation du mot clé DISTINCT

Le résultat d'un SELECT étant un ensemble, il peut y avoir des doublons. Le mot clé DISTINCT permet de préciser que l'on ne veut qu'un seul exemplaire des lignes retenues dans la sélection.

- Ensemble des types d'avions dont la capacité est supérieure à 250 passagers.

```
SELECT      Marque,TypeAvion, Capacite
FROM        AVION
WHERE       Capacite > 250
```

	Marque	TypeAvion	Capacite
1	AIRBUS	A320	381
2	AIRBUS	A320	522
3	Boeing	B747	400
4	AIRBUS	A320	423
5	Atr	ATR42	500
6	Boeing	B727	300
7	Boeing	B727	300
8	AIRBUS	A340	350
9	AIRBUS	A340	345
10	AIRBUS	A340	600

DRIOUCH B.

8

Opération de sélection

- Nous ajoutons le mot clé **DISTINCT** à la sélection et ordonnons les données par capacités décroissantes :

```
SELECT      DISTINCT Marque, TypeAvion, Capacite
FROM        AVION
WHERE       Capacite > 250
ORDER BY    Capacite DESC
```

	Marque	TypeAvion	Capacite
1	AIRBUS	A340	600
2	AIRBUS	A320	522
3	Atr	ATR42	500
4	AIRBUS	A320	423
5	Boeing	B747	400
6	AIRBUS	A320	381
7	AIRBUS	A340	350
8	AIRBUS	A340	345
9	Boeing	B727	300

DRIOUCH B.

9

Opération de sélection

- Fonctions de groupe
- Nous pouvons travailler sur des données consolidées au travers de requêtes qui, à partir d'un ensemble de lignes sélectionnées, réalisent des opérations de synthèse sur ces données.
- Nous pouvons combiner ainsi des fonctions récapitulatives à la liste des attributs sélectionnés:
 - SUM somme des valeurs de l'attribut
 - MIN minimum des valeurs de l'attribut
 - MAX maximum des valeurs de l'attribut
 - AVG moyenne des valeurs
 - COUNT nombre d'enregistrements sélectionnés
- Nous avons à notre disposition des clauses spécifiques à la construction de requêtes récapitulatives. Nous avons ainsi les clauses :
 - GROUP BY permet d'exprimer sur quels attributs sont agrégés les données.
 - HAVING permet de définir une condition sur des valeurs de regroupement.

DRIOUCH B.

10

Opération de sélection

- Quelques exemples :
- Nombre d'avions dans la table AVION

```
SELECT COUNT(AV#)
FROM AVION
```

	(Aucun nom de colonne)
1	16

- Mécanisme des alias : NomColonne **AS** NomAlias

```
SELECT COUNT(AV#) as "Nombre Avions"
FROM AVION
```

	Nombre Avions
1	16

DRIOUCH B.

11

Opération de sélection

- Groupement
- Je peux aussi vouloir comptabiliser le nombre d'avions par marque. La clause GROUP BY me permet alors de définir les conditions de regroupement des calculs récapitulatifs :

```
SELECT COUNT(AV#) as "Nombre Avions"
FROM AVION
GROUP BY Marque
```

	Nombre Avions
1	8
2	1
3	5
4	2

```
SELECT Marque,COUNT(AV#) as "Nombre Avions"
FROM AVION
GROUP BY Marque
ORDER BY "Nombre Avions" DESC
(ORDER BY COUNT(AV#) DESC)
```

	Marque	Nombre Avions
1	AIRBUS	8
2	Boeing	5
3	Caravelle	2
4	Atr	1

DRIOUCH B.

12

Opération de sélection

■ HAVING

On veut aussi ne pas vouloir conserver dans mon résultat les marques dont le nombre d'avions est inférieur à 3. J'introduis alors dans ma requête une clause **HAVING** qui exprime une condition sur une opération de regroupement.

```
SELECT Marque,COUNT(AV#) as "Nombre Avions"  
FROM AVION  
GROUP BY Marque  
HAVING COUNT(AV#) >= 3  
-- on ne peut pas utiliser "Nombre Avions"  
ORDER BY COUNT(AV#) DESC
```

	Marque	Nombre Avions
1	AIRBUS	8
2	Boeing	5

■ Capacités minimum et maximum des Boeing.
SELECT MIN(Capacite) as "Capacite Mini Boeing",
MAX(Capacite) as "Capacite Maxi Boeing"
FROM AVION
WHERE Marque = 'BOEING'

	Capacite Mini Boeing	Capacite Maxi Boeing
1	250	400

DRIOUCH B.

13

Opération sur la sélection

■ On peut utilisé dans la sélection les opérateur suivant:

- Opérateurs arithmétiques
- Opérateurs de comparaison
- Opérateurs logiques
- Opérateur(fonctions) de traitement de chaînes
- Opérateur(fonctions) de traitement de date

DRIOUCH B.

14

Exemples

- Liste des vols au départ de Toulouse et a destination de Paris entre 12 et 15 heures.

```
SELECT vol#, villedepart, villearrivee, heuredepart, heurearrivee  
FROM Vol  
WHERE Heuredepart BETWEEN 12 AND 15  
AND villedepart LIKE 'TOULO%' AND villearrivee LIKE 'PAR%'
```

- -- Liste des noms de pilotes

```
SELECT nom  
FROM Pilote
```

- -- Liste des Pilotes dont le nom contient la chaîne el

```
SELECT nom  
FROM Pilote  
Where nom LIKE '%el%'
```

- -- Liste des Pilotes dont le 3ème caractère du nom est un r

```
SELECT nom  
FROM Pilote  
Where nom LIKE '___r%'  
DRIOUCH B.
```

15

Exemples

- -- Liste des Pilotes dont le nom comporte les caractères u,c,ou l

```
SELECT nom  
FROM Pilote  
Where nom LIKE '%[cul]%'
```

- -- Liste des Pilotes dont le nom commence par le caractère m ou l

```
SELECT nom  
FROM Pilote  
Where nom LIKE '[ml]%'
```

- /* Liste des Pilotes dont le nom commence par le caractère m ou l et se termine par l */

```
SELECT nom  
FROM Pilote  
Where nom LIKE '[ml]%'
```

DRIOUCH B.

16

Produit Cartésien

- Liste des affectations possibles (Nom Pilote, Marque Avion) entre pilote et avion :

Exp:

```
SELECT DISTINCT Nom,
Marque
FROM Pilote, Avion
```

```
SELECT DISTINCT Nom,
Marque, Prénom
FROM Pilote, Avion, Passager
```

DRIOUCH B.

	Nom	Marque
3	Bertrand	Boeing
4	Bertrand	Caravelle
5	Bost	AIRBUS
6	Bost	Atr
7	Bost	Boeing
8	Bost	Caravelle
9	Driss	AIRBUS
10	Driss	Atr
11	Driss	Boeing
12	Driss	Caravelle
13	Hervé	AIRBUS
14	Hervé	Atr

17

Division

- liste des pilotes qui conduisent tous les avions :

```
Select Nom From pilote, vol
where pilote.pil#=vol.pilote and vol.avion in (Select av# from avion)
group by pilote
Having Count(Distinct avion)=(Select Count(Distinct av#) from Avion)
```

- liste des avions qui vont dans toutes les villes

```
Select avion From vol
where vol.villearrivee in (Select villearrivee from vol)
group by avion
Having Count(Distinct villearrivee)=(Select Count(Distinct villearrivee)
from vol)
```

DRIOUCH B.

18

Les fonctions scalaire

- Avec la clause WHERE, il nous est possible d'utiliser les prédicats suivants :

ALL , ANY , BETWEEN , EXISTS , IN , LIKE

NOT BETWEEN, NOT EXISTS, NOT IN, NOT LIKE

ANY : Permet une comparaison d'une colonne avec une liste de valeurs fournies par une sous-interrogation. Si une des valeurs de la liste rend la condition vraie, alors la ligne est sélectionnée.

- SELECT vol#, villedepart from vol where avion = any (select av# from avion where Marque like 'boeing')

ALL : permet une comparaison d'une colonne avec l'ensemble des valeurs d'une liste de valeurs fournies par une sous-interrogation. Si toutes les valeurs de la liste rendent la condition vraie, alors la ligne est sélectionnée.

- SELECT distinct Marque, Capacite from avion where capacite > all(select capacite from avion where marque like 'boeing')

EXISTS : est un prédicat évalué à vrai si la sous-requête qui le suit donne au moins une ligne c'est à dire un résultat non vide.

BETWEEN : Les limites sont incluses. La limite inférieure doit être définie la première.

DRIOUCH B.

19

Les Opérations de Jointure

2. Requêtes intégrant plusieurs tables

- La **jointure** consiste à rechercher entre deux tables ayant un attribut commun (même type et même domaine de définition) tous les tuples (toutes les lignes) pour lesquels ces attributs ont la même valeur.
 - la méthode ensembliste réalise l'intersection de deux ensembles et s'exprime sous forme de requêtes imbriquées.
 - la méthode prédicative vérifie l'égalité de deux attributs et s'exprime sous la forme d'une seule sélection conditionnelle.

DRIOUCH B.

20

Les Opérations de Jointure

■ La méthode ensembliste

Schéma de construction :

SELECT liste d'attributs

FROM table1

WHERE attribut de jointure

IN (SELECT attribut de jointure
FROM table2
[WHERE condition])

La requête à l'intérieur des parenthèses est dite requête interne ou sous-requête. Elle est évaluée en premier, constituant ainsi un premier ensemble dont on réalisera l'intersection (IN) avec l'ensemble issu de l'évaluation de la requête externe.

Les attributs sélectionnés, et retenus dans le jeu de résultat, sont **nécessairement** issus de la requête externe. Il s'agit donc d'une méthode assez restrictive.

DRIOUCH B.

21

Les Opérations de Jointure

■ Exemple : Liste des pilotes affecté a des vol de départ paris

SELECT PIL# as "Code Pilote", NOM

FROM PILOTE

WHERE PIL# IN (SELECT PILOTE
FROM VOL
WHERE Villedepart = 'Paris')

SELECT PIL# as "Code Pilote", NOM

FROM PILOTE

WHERE PIL# IN (SELECT PILOTE
FROM VOL
WHERE avion IN (SELECT av#
FROM avion
Where Marque like 'boeing'));

	Code Pilote	NOM
1	1	Serge
2	2	Jean
3	7	Bertrand
4	9	Luc

DRIOUCH B.

22

Les Opérations de Jointure

- **Autre exemple** : Traitement du sauf (négation NOT) ou différence.

- Liste des pilotes qui ne sont pas affectés à des vols.

```
SELECT PIL# as "Code Pilote",NOM  
FROM PILOTE  
WHERE PIL# NOT IN (SELECT PILOTE  
                   FROM VOL)
```

On ne pourra pas non plus écrire la requête suivante :

```
SELECT PIL# as "Code Pilote",NOM,VILLEDEPART  
FROM PILOTE  
WHERE PIL# IN (SELECT PILOTE FROM VOL)
```

Car l'attribut VILLEDEPART n'appartient pas à la table sur laquelle porte la requête externe. Vous obtenez un message d'erreur

DRIOUCH B.

23

Les Opérations de Jointure

- **La méthode prédicative**

- La requête comporte une seule instruction SELECT qui traite plusieurs tables dont la liste apparaît dans la clause FROM. La traduction de la jointure se fait par une équation de jointure (égalité entre 2 attributs) exprimée au niveau de la clause FROM (ou Where sous forme de produit cartésien).
- Il existe quatre natures (Types) de jointure qui sont respectivement exprimées par les mots clés INNER, RIGHT OUTER, LEFT OUTER ou FULL OUTER dont nous verrons les différents cas d'usage dans les exemples suivants.
- Cette méthode est en général préférable à la méthode ensembliste car plus performante.

DRIOUCH B.

24

Les Opérations de Jointure

- Jointure interne (INNER JOIN)
- Liste des avions affectés à des vols avec la mention de la ville de départ du vol.

```
SELECT AV#, MARQUE, TYPEAVION, VILLEDEPART
FROM AVION INNER JOIN VOL
ON AVION.AV# = VOL.AVION
```

	AV#	MARQUE	TYPEAVION	VILLEDEPART
1	100	AIRBUS	A320	Nice
2	100	AIRBUS	A320	Paris
3	101	Boeing	B707	Paris
4	105	AIRBUS	A320	Grenoble
5	105	AIRBUS	A320	Toulouse
6	107	Boeing	B727	Lyon
7	109	AIRBUS	A340	Bastia
8	106	Atr	ATR42	Paris
9	106	Atr	ATR42	Brive

La clause INNER JOIN, jointure interne, ne retient que les lignes des deux tables pour lesquelles l'expression exprimée au niveau de ON se vérifie.

DRIOUCH B.

25

Les Opérations de Jointure

- Jointure Externe Gauche & Droite (LEFT JOIN, RIGHT JOIN)

Si nous voulons la liste de tous les avions avec, pour ceux qui volent, des informations sur les villes de départ, nous écrirons :

```
SELECT AV#, MARQUE , TYPEAVION, VILLEDEPART
FROM AVION LEFT OUTER JOIN VOL
ON AVION.AV# = VOL.AVION
```



```
SELECT AV#, MARQUE , TYPEAVION, VILLEDEPART
FROM VOL RIGHT OUTER JOIN AVION
ON AVION.AV# = VOL.AVION
```

	AV#	MARQUE	TYPEAVION	VILLEDEPART
1	100	AIRBUS	A320	Nice
2	100	AIRBUS	A320	Paris
3	101	Boeing	B707	Paris
4	101	Boeing	B707	Nice
5	102	AIRBUS	A320	Toulouse
6	103	Caravelle	Caravelle	NULL

Pour les avions qui ne sont pas en service, l'attribut VILLEDEPART aura la valeur **NULL**

DRIOUCH B.

26

Les Opérations de Jointure

■ Compléments sur des jointures

■ INNER

Spécifie toutes les paires correspondantes de lignes renvoyées. Supprime les lignes n'ayant pas de correspondance entre les deux tables. Ceci est l'option par défaut si aucun type de jointure n'est spécifié.

■ FULL [OUTER] : Peu utilisée

Précise qu'une ligne de la table de gauche ou de droite, qui ne correspond pas à la condition de jointures, est comprise dans l'ensemble de résultats et que les colonnes de sortie sans correspondance dans l'autre table ont des valeurs nulles. Ceci est fourni **en plus** de toutes les lignes renvoyées par INNER JOIN.

DRIOUCH B.

27

Les Opérations de Jointure

■ Compléments sur des jointures

■ LEFT [OUTER]

Spécifie que toutes les lignes de la table de gauche ne respectant pas la condition de jointure sont comprises dans l'ensemble de résultats, et que les colonnes de sortie de l'autre table sans correspondance ont des valeurs NULL.

■ RIGHT [OUTER] :

Spécifie que toutes les lignes de la table de droite ne respectant pas la condition de jointure sont comprises dans l'ensemble de résultats, et que les colonnes de sortie correspondant à l'autre table ont des valeurs NULL.

■ JOIN

Indique que l'opération de jointure spécifiée doit avoir lieu entre les tables ou vues données.

■ ON <condition de Jointure>

Indique la condition sur laquelle se base la jointure

DRIOUCH B.

28

Les Opérations de Jointure

■ Auto-jointure

- L'auto-jointure est la jointure entre une table et elle-même, pour sélectionner des enregistrements correspondant à d'autres de la même table. Il est nécessaire de recourir alors à des alias pour définir la table déjà utilisée.

■ Exemple d'auto-jointure :

■ Liste des avions de même capacité

```
SELECT AVION.AV#, AVION.MARQUE, AVION.TYPEAVION,
       AVION.CAPACITE, AV2.AV#, AV2.MARQUE, AV2.TYPEAVION,
       AV2.CAPACITE
```

```
FROM AVION INNER JOIN AVION AS "AV2"
```

```
ON AVION.CAPACITE = AV2.CAPACITE
```

```
WHERE AVION.AV# > AV2.AV#
```

- **A noter** : L'expression de la clause WHERE pour éliminer les paires de lignes figurant plusieurs fois.

DRIOUCH B.

29

Exemple avec (Inner, Left, Right, Full)

Stg	
ids	Nom
1	A
2	B
3	C

Note		
ids	ide	Note
1	1	5
1	2	6
2	2	5

Examen	
Idex	DateEx
1	X
2	Y
3	Z

INNER entre stg, note et Examen				
ids	Nom	ide	DateEx	Note
1	A	1	X	5
1	A	2	Y	6
2	B	2	Y	5

LEFT entre stg, note et Examen				
Ids	Nom	ide	DateEx	Note
1	A	1	X	5
1	A	2	Y	6
2	B	2	Y	5
3	C	Null	Null	Null

RIGHT entre stg, note et Examen				
Ids	Nom	ide	DateEx	Note
1	A	1	X	5
1	A	2	Y	6
2	B	2	Y	5
Null	Null	3	Z	Null

FULL entre stg, note et Examen				
Ids	Nom	ide	DateEx	Note
1	A	1	X	5
1	A	2	Y	6
2	B	2	Y	5
3	C	Null	Null	Null
Null	Null	3	Z	Null

DRIOUCH B.

30

Les Opérations de Jointure

■ Requêtes sur 3 tables (jointes 2 à 2)

- Liste des vols avec informations sur pilotes et avions

SELECT VOL#, VILLEARRIVEE, VILLEDEPART, HEUREDEPART,
AV#,MARQUE, TYPEAVION, PIL#, NOM

FROM

((AVION INNER JOIN VOL ON AVION.AV# = VOL.AVION)

INNER JOIN PILOTE ON VOL.PILOTE = PILOTE.PILOTE#)

	VOL#	VILLEARRIVEE	VILLEDEPART	HEUREDEPART	AV#	MARQUE	TYPEAVION	PIL#	NOM
1	100	Paris	Nice	7.20	100	AIRBUS	A320	1	Serge
2	101	Toulouse	Paris	11.20	100	AIRBUS	A320	2	Jean
3	102	Nice	Paris	12.35	101	Boeing	B707	1	Serge
4	103	Toulouse	Grenoble	9.00	105	AIRBUS	A320	3	Roger
5	104	Grenoble	Toulouse	17.00	105	AIRBUS	A320	3	Roger
6	105	Paris	Lyon	6.00	107	Boeing	B727	7	Bertrand
7	106	Paris	Bastia	10.00	109	AIRBUS	A340	8	Hervé
8	107	Brive	Paris	7.00	106	Atr	ATR42	9	Luc
9	108	Paris	Brive	19.00	106	Atr	ATR42	9	Luc
10	109	Lyon	Paris	18.00	107	Boeing	B727	7	Bertrand

DRIOUCH B.

31

Les fonctions intégrées

- On présente les principales fonctions intégrées proposées par SQL Server, aussi sous des formes similaires sinon identiques sous les principaux SGBDR du marché.

- Fonctions de conversion
- Fonctions de traitement de chaînes
- Fonctions de manipulation de dates
- Quelques fonctions intégrées

- SQL server propose de nombreuses fonctions intégrées qui permettent de manipuler des données utilisateurs ou des données du système.

DRIOUCH B.

32

Les fonctions intégrées

■ Fonctions de conversion

Certaines conversions ne peuvent être automatiquement réalisées par le système. Nous devons alors réaliser ces conversions de manière explicite au moyen des fonctions de conversion `CAST(expression AS data_type[(length)])` et `CONVERT(data_type[(length)], expression [,style])`.

`SELECT CONVERT(DATETIME,'10-14-2011',110) AS "Date au Format USA"`

	Date au Format USA
1	2011-10-14 00:00:00

`SELECT CONVERT(varchar,Getdate(),110) AS "Date au Format USA"`

	Date au Format USA
1	10-14-2011

`SELECT CONVERT(Decimal(10,3),sum(prix)) AS "TOTAL CA Net" FROM affectevol`

	TOTAL CA Net
1	5400.000

`SELECT Cast(Getdate() as varchar) as "Date Texte"`

	Date Texte
1	oct 14 2011 9:03AM

DRIOUCH B.

33

Les fonctions intégrées

■ Fonctions de traitement de chaînes

Quelques exemples dans ce tableau car elles sont nombreuses

Voir aide à l'index fonctions, chaîne

Liste des noms des pilotes formatés. Le premier caractère de gauche est mis en majuscules, Les autres caractères en minuscules

<code>LEFT(str,n)</code> , <code>RIGHT(str,n)</code>	Extraire des caractères à gauche ou à droite
<code>UPPER(str)</code> , <code>LOWER(str)</code>	Mettre en majuscule ou minuscules
<code>LTRIM(str)</code> , <code>RTRIM(str)</code>	Suppression des espaces à gauche ou à droite
<code>SUBSTRING(str,pos,n)</code>	Extraction d'une sous chaîne
<code>REVERSE(str)</code>	Inversion d'une chaîne
<code>LEN(str)</code>	Longueur d'une chaîne
<code>ASCII(char)</code>	Valeur ascii d'un caractère
<code>NCHAR(n)</code>	Renvoie le caractère unicode en fonction de la valeur donnée
<code>REPLACE(str,str0,str1)</code>	Remplacement d'une occurrence de chaîne str0 par une autre str1

`SELECT Upper(Substring(Nom,1,1)) + Lower(Substring(Nom,2,Len(Nom)-1))`
`FROM PILOTE`

DRIOUCH B.

34

Les fonctions intégrées

■ Fonctions de manipulation de dates

Fonctions intégrées permettant de manipuler des valeurs de type DATETIME.

DATEADD(datepart , number, date)	Ajout d'un intervalle de temps à une date
DATEDIFF(datepart , startdate , enddate)	Intervalle de temps entre deux dates
DATEPART(datepart , date)	Extraction d'une partie de date
DATENAME(datepart ,date)	Chaîne représentant une partie de date
DAY(datetime), MONTH(datetime) , YEAR(datetime)	Renvoie d'une partie de date
GETDATE()	Date du système
GETUTCDATE()	Renvoie la date actuelle du système en temps UTC (Universal Time Coordinate ou Greenwich Mean Time)

- Nombre de jours entre la date de naissance et la date du jour.
SELECT NOM,DATEDIFF(DAY,DateNaissance,GETDATE()) as "Nombre jours depuis Naissance"
FROM Pilote
- Extrait de portions de la date de naissance avec DatePart et DateName
SELECT DATEPART(MONTH,DateNaissance),
DATENAME(MONTH,DateNaissance)
FROM PILOTE

DRIOUCH B.

35

Exercice

Gestion Stock

- Clients (Ref_cli,DescriptionCl,Contact,villeCl,solvabilite,telCl)
- Fournisseurs (Ref_fou,descriptionF,VilleF, TelF)
- Produits (Ref_pro, DescriptionP, Ref_fou, Ref_cat, PrixU, Quantite)
- Categorie (Ref_cat, DescriptionCa)
- Commande (Ref_com, Ref_cli, DateCom, Date_liv)
- DetailCommande (Ref_com,Ref_pro, Qtite)

Questions :

1. Liste des commandes du 1er Trimestre de l'année 1997.
2. Liste des commandes dont la différence entre la date de commande et la date de livraison est supérieure à 10 jours.
3. La durée entre dateCom et Date Actuelle en jour, mois et année de chaque commande
4. Liste des commandes en affichant les produits commandés avec leurs prix et quantités respectifs ainsi que la date de commande et le client.
5. Liste des catégories dont la désignation contient la lettre « N ».
6. Lister les fournisseurs qui ne figurent pas dans la table Produit.
7. Liste des produits affichant les quantités maximale et minimale commandées par Produit.
8. Liste des produits affichant une nouvelle colonne «Montant Total par Produit».

DRIOUCH B.

36

■ EN Résumé

- **SELECT** DISTINCT liste des attributs résultats
FROM liste des tables
WHERE conditions de prise en compte
GROUP BY critères de regroupement
HAVING conditions sur regroupements
ORDER BY critères de tri des résultats

- **SELECT** colonnes ↗ PROJECTION
 permet de définir les colonnes résultats
 (colonnes de tables ou fonctions statistiques)
SELECT DISTINCT → élimination des doublons
SELECT * → sélectionne toutes les colonnes
 Fonctions statistiques
 → SUM(colonne) : cumul
 → AVG(colonne) : moyenne
 → COUNT(*) : comptage lignes
 → MIN(colonne) : plus petite valeur
 → MAX(colonne) : plus grande valeur

- **FROM** tables ↗ PRODUIT CARTESIEN
 permet d'indiquer les tables
 → où les colonnes suivant le **SELECT** se trouvent
 → où les colonnes permettant les jointures se trouvent

DRIOUCH B.

WHERE conditions ↗ RESTRICTION
 permet d'indiquer des critères de prise en compte des lignes des tables se trouvant derrière FROM
 Opérateurs → = , < , > , <= , >= , <>

→ AND , OR, NOT,
 → IN , BETWEEN , LIKE

GROUP BY colonnes
 permet d'effectuer des calculs par groupe de lignes sélectionnées
 Les critères doivent être des colonnes de la projection
 → on obtient une ligne résultat par critère

HAVING conditions ↗ RESTRICTION sur groupe
 équivalent à WHERE mais pour des lignes issues d'un GROUP BY

ORDER BY colonnes
 Trie les lignes de la table resultat
 → ASC : croissant ↗
 → DESC : décroissant ↘

37