

INTRODUCTION AUX BASES DE DONNEES

I./ Introduction

Systèmes de Gestion de Base de Données (SGBD)

C'est l'ensemble de logiciels et de matériels permettant la gestion d'un ensemble important de données appelées Base de Données.

I./ Notion de Base de Données

I.1/ Définition d'une base de données:

Une Base de Données (BD) est un ensemble de tables.

Commentaires :

- chaque table représente un objet ou une entité de l'entreprise concerné par cette base de données,
- En générale les tables sont reliées entre elles. C'est à dire Qu'à partir d'une information dans une table on peut accéder à des lignes correspondant d'une autre table. Ce qui permet de voir les deux tables comme une seul table

Exemple 1 : Base de données des anciens élèves TC

La Base de données précédente est composée de 4 tables :

- Tables : lycee
- Table : ancien_tc
- Table : poursuite_etudes
- Table : vie_professionnelle

Voici un échantillonnage des contenus :

Tables lycee :

LYCEE	ID_LYCEE	VILLE	PUBLIC
Guy de	0760035A	FECAMP	-1
François 1er	0760052U	LE HAVRE	-1
Porte Océane	0760054W	LE HAVRE	-1
Jules Siefried	0760056Y	LE HAVRE	-1
Robert	0760058A	LE HAVRE	-1
Guill. Le	0760072R	LILLEBONNE	-1
Jean Prévost	0760076V	MONTIVILLIER	-1

Table Ancien_TC :

n_adut	NOM	PRENOM	GROUPE	CODE POSTAL	VILLE	ADRESSE
6	AVENEL	Sabrina	3	76430	SAINT-	LE VILLAGE ROUTE
7	BAILLEUL	Géraldine	2	76700	HARFLEUR	1 rue des fontaines
8	BAMBIER	Alexandra	2	76330	NOTRE DAME	57 RESIDENCE LES
9	BAPTISTA	Silvy	2	76610	LE HAVRE	51 RUE GUILLOT ET
10	BAUDOU	Delphine	1	76600	LE HAVRE	71 RUE DE
11	BEGUIR	Adel	2	76620	LE HAVRE	10 IMPASSE SUGER
12	BELLECH	Frédéric	2	76620	LE HAVRE	9 RUE MAURICE LE
13	BERKANI	Linda	5	76410	CLEON	51 RUE JEAN JAURES
14	BEZAULT	Matthieu	4	76280	CRICQUETOT-	27 RUE DU 19 MARS
15	BIODORE	Stéphane	3	76610	LE HAVRE	19 rue JULES VALLES
16	BLANCHA	Jan	2	76170	LILLEBONNE	RUE MAURICE LEBAS
17	BOQUET	Emilie	2	76290	FONTENAY	3 PLACE DES
18	BORCARD	Rodolphe	2	76170	LILLEBONNE	LOTISSEMENT LA
19	BUTELET	Nicolas	2	76240	LE MESNIL	12 SQUARE MAURICE
20	CAVELIER	Sébastien	4	76280	CRICQUETOT-	COTE DU MOULIN
21	CHAKOUR	Zohra	3	76500	ELBEUF	38, rue Arthur Hulme
22	CHAUVRI	Lorna	2	76250	DEVILLE LES	14 RUE DE VERDUN
23	CHERON	Sandra	2	27130	VERNEUIL	LE HAUT BROU
24	CHOET	Elisabeth	1	27100	VAL DE	8 IMPASSE DES
25	CRAMPON	Cyrille	5	76700	HARFLEUR	1 RUE MARYSE

BAC	SEXE	TELEPHONE	ID_LYCEE	ANNEE
G3	F	35.20.00.15	0760054W	96
B	F	35.20.61.30	0760058A	96

G3	F	35.31.66.86	0760072R	96
G3	F	35.47.64.46	0760058A	96
D	F	35 42 45 39	0760052U	96
B	M	35.54.52.10	0760058A	96
A1	M	35.54.01.27	0760174B	96
G3	F	35.77.47.73	0760029U	96
B	M	35 20 78 99	0760076V	96
G3	M	35.45.83.06	0761735Y	96
B	M	35 38 81 72	0760072R	96
B	F	35 30 49 46	0760076V	96
G3	M	35 38 08 72	0760072R	96
B	M	35.80.70.15	0760091L	96
G3	M	35.20.98.40	0760061D	96
G3	F	35.77.15.45	0760029U	96
A2	F	35.75.15.93	0760096S	96
B	F	32.60.20.72	0270042Z	96
A1	F	32 59 43 39	0271582Y	96
G3	M	35.51.32.80	0762976X	96

Table poursuite_etudes :

n_adut	DIPLOME	SPECIALITE	ANNEE
7	Maitrise	BOURSE Erasmus	1997
10	FORMATIO	E.S.C. LE HAVRE-CAEN	1996
11	LICENCE	COMMERCE INTERNATIONAL	1996
13	FORMATIO	VENTE/DISTRIBUTION	1996
14	MAITRISE	Commerce et Vente	1997
16	FORMATIO	3è ANNEE MARKETING en	1996
17	licence	COMMERCE IUP	1996
18	D.U.T.A.	COMMERCE ALTERNANCE EN	1996
22	MAITRISE	COMMERCE INTERNATIONAL	1997
24	FORMATIO	commerciale I.S.D	1996
26	LICENCE	D.T.A à ROUEN	1996

Table vie_professionnelle :

n_adut	SECTEUR	FONCTION	ANNEE	adresse	ville
6	entreprise travail	secrétariat	1996	ADECO. Place de	Le Havre
8	ENTREPRISE	SECRETAIRE	1996		
9	GRANDE	CAISSIERE /	1996		
12	Optique	Vendeur à	1996	Optique generale.	Fontenay
15	SERVICE		1997		
19	RECHERCHE				
20	SERVICE		1996		
21	Recherche	ENQUETRICE	1997	INSEE	ELBEUF
23	FILLE AU PAIR AU		1997		
25	SERVICE		1996		

Exemple 2 : Vidéo Club

pour gérer cette entreprise on recense les objets suivants :

- Film
- Client

• Rayon

Donc, La BD qui gère le vidéo club sera constituée de 3 tables : FILM, CLIENT, RAYON. La composition de ces tables est la suivante :

Table film

N_FILM	TITRE	GENRE	DUREE	Apparu
--------	-------	-------	-------	--------

Table client

N_CLIENT	NOM	ADRESSE	CODE POSTAL	VILLE	COTISATION
----------	-----	---------	-------------	-------	------------

Table rayon

N_exemplaire	N_FILM	N_CLIENT	DATE_EMPRUNT
--------------	--------	----------	--------------

Voici également un échantillonnage de leur contenu

table film :

N_FILM	TITRE	GENRE	DUREE	apparu
F1	L'ANGE BLEU	DRAME	100	11/10/90
F2	EMMANUELLE	EROTIQUE	95	10/2/87
F3	VERDOUX	COMEDIE	85	1/12/85
F4	LA CHANCE	DRAME	125	21/2/88
F5	LA VILLE DE Z	DRAME	140	11/1/82
F10	La crime était presque	POLICIER	100	1/12/88
F11	QUAI DES Orfevres	POLICIER	90	21/2/81

table client :

N_CLIENT	NOM	ADRESSE	CODE POSTAL	VILLE	COTISATION
1	DUPOND	1 rue la gare	76600	LE HAVRE	100
2	ALFRED	5 av. deville	76610	HARFLEUR	200
3	DURAND	1 rue capus	76000	ROUEN	50
4	TOTO	R. de ROME	76610	LE HAVRE	300
5	JIME	4, R. STRASBOURG	76600	LE HAVRE	210
5	LUBIN	3 r. St. Paul	76000	ROUEN	0

table rayon :

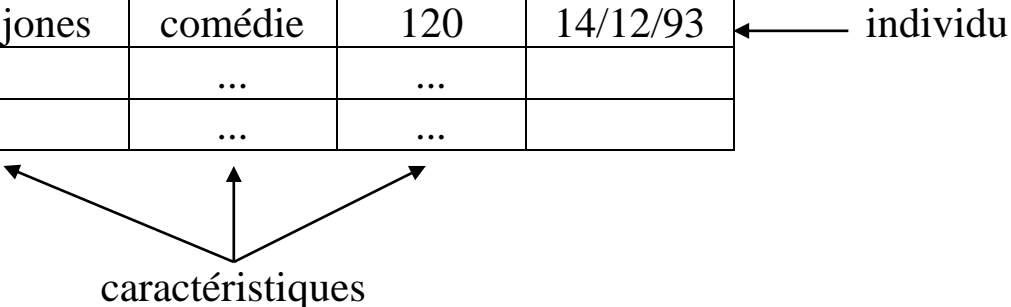
N_exemplaire	N_FILM	N_CLIENT	DATE_EMPRUNT
1	F1	1	04/11/1993
2	F1	0	01/12/1993
3	F1	0	11/10/1993
1	F2	3	01/01/1992
1	F3	0	01/01/1991
2	F3	2	12/07/1993
1	F4	0	02/05/1993
1	F5	3	02/09/1993

I.2/ Notion de table

Définition : Une table est un ensemble de lignes et de colonnes représentant un objet.

Exemple : la table film

n_film	titre	genre	durée	apparu
F1	L'ange bleu	drame	100	11/10/90
F2	Indiana jones	comédie	120	14/12/93
F3	
...	



Une Ligne : représente un *individu* appartenant à l'objet représenté par la table

Une colonne (ou attribut) : représente une valeur qui désigne une des *caractéristiques* de l'individu appartenant à l'objet désigné par la table.

Par exemple pour l'objet film, les caractéristiques d'un film sont : son titre, son numéro, sa date d'apparition, sa durée, et son genre.

Notion de domaine : La valeur d'une colonne appartient toujours à un domaine, en informatique on distingue 3 domaines principaux :

- *chaîne de caractères*,
- *numérique*,
- *la date*

Exemples: reprenons la BD vidéo club :

Table film

N_FILM	TITRE	GENRE	DUREE	apparu
--------	-------	-------	-------	--------

Table client

N_CLIENT	NOM	ADRESSE	CODE POSTAL	VILLE	COTISATION
----------	-----	---------	-------------	-------	------------

Table rayon

N_exemplaire	N_FILM	N_CLIENT	DATE_EMPRUNT
--------------	--------	----------	--------------

1. le domaine du titre est une *chaîne de caractères*,
2. le domaine de la durée est un *décimal*,
3. le domaine de la colonne apparu est *la date*

Exercice : On considère la BD des anciens élèves tc. Pour chaque table et chaque colonne, donner le domaine et la taille.

Solution :

Tables *lycee* :

LYCEE	ID_LYCEE	VILLE	PUBLIC
T(50)	T(8)	T(40)	Logique

Table *Ancien_TC* :

n_adut	NOM	PRENOM	GROUPE	CODE POSTAL	VILLE	ADRESSE
N ent	T(40)	T(200)	N ent	T(5)	T(40)	T(70)

BAC	SEXE	TELEPHONE	ID_LYCEE	ANNEE
T(6)	T(1)	T(12)	T(8)	N ent

Table *poursuite_etudes* :

n_adut	DIPLOME	SPECIALITE	ANNEE D'OBTENTION
N ent	T(20)	T(30)	N ent

Table *Vie_professionnelle* :

n_adut	SECTEUR D'ACTIVITES	FONCTION	ANNEE D'EMBAUCHE	adresse professionnelle	ville
N ent	T(200)	T(15)	N ent	T(70)	T(40)

II.3/ Notion de la clé d'une table

Définition : La clé d'une table est une colonne (ou un ensemble de colonnes) qui prend(ent) une valeur unique pour chaque ligne, permettant ainsi de repérer d'une manière unique tous les individus de la table.

Exemple : on considère la table film

n_film	titre	genre	durée	apparu
F1	L'ange bleu	drame	100	11/10/90
F2	Indiana jones	comédie	120	14/12/93
F3	

↑
clé

On remarque que tout film peut être repéré d'une façon unique par le numéro de film, d'où la colonne n_film constitue une clé pour la table film. Exercice : quelle est la clé de la table client ?

Commentaires :

- une clé identifie tout individu appartenant à la table,
- Si dans une table il n'existe pas une colonne qui identifie toute seule les individus, alors dans ce cas on peut concatener (combiner) plusieurs colonnes pour obtenir un identifiant. La clé dans ce cas sera constituée de plusieurs colonnes.

Exemple : On considère la table rayon

N_exemplaire	N_FILM	N_CLIENT	DATE_EMPRUNT
1	F1	1	04/11/1993
2	F1	0	01/12/1993
3	F1	0	11/10/1993
1	F2	3	01/01/1992
1	F3	0	01/01/1991
2	F3	2	12/07/1993
1	F4	0	02/05/1993
1	F5	3	02/09/1993

Il n'existe pas un champ permettant d'identifier les cassettes en rayon.

On ne peut pas constituer une clé avec un seul champ.

La clé ici est donc la composition de deux champs ; Le champ numéro de film (*n_film*) avec le champ numéro d'exemplaire (*N_exemplaire*)

clé du rayon → (*n_film* + *N_exemplaire*)

III./ Création d'une Base de Données :

Interroger, et gérer une BD existant peut être une tâche abordable par un utilisateur non informaticien.

Par contre, la construction d'une BD n'est pas une tâche facile sur tout quand il s'agit de créer un Système d'information d'une entreprise.

Ceci nécessite l'utilisation des méthode d'analyse «informatique » comme MERISE et aussi des outils d'aide : les ateliers de Génie logiciel.

L'analyse conduit à l'établissement d'un schéma conceptuel de la base de données qui est indépendante de la technologie de l'implantation sur machine.

Une fois le schéma conceptuel est établie, le langage SQL ("Structure Query Language" langage d'interrogation structuré) peut être utiliser pour l'implantation de telle BD.

Il est hors de question, de former les TC à ces outils, nous allons cependant introduire ce concepts pour vous aider à mener à bien vos propres développement.

IV./ Consultation et interrogation des bases de données (le langage SQL

L'utilisateur manipule les données d'une BD au moyen d'un unique langage, SQL (Structure Query Language)

Ce langage permet à l'utilisateur de demander au SGBD :

- de consulter les données d'une ou de plusieurs tables,
- de ranger des données dans les tables et de les modifier,
- De définir les autorisation d'accès.
-

Ce langage (SQL) peut être utilisé par un non informaticien.

Il permet d'interroger une base de données en lui formulant des questions (requêtes) exprimées en langage ressemblant à des langages Humains.

SQL utilise des mots clés comme :

- *select*,
- *from*,
- *where*,
- *order by*,
- *group by*,
- *having, in, not in*,
- *between*,
- *and, or, not*,
- *=, /=*,
- *union*
- *insert, update, grant, revoke, ...*
- ...

Dans la suite, on va passer en revue l'essentiel de ce langage.

IV.1/ Notion de requête :

Définition : une requête est une phrase exprimée en SQL permettant d'interroger une base de données. Par exemple la requête suivante :

```
select NOM, prenom, telephone  
from ancien_tc  
where ville = "ELBEUF" ;
```

donne le nom, le prénom, et le téléphone de tous les anciens élèves tc de la ville ELBEUF.

En effet, la commande *select* se compose de plusieurs clauses (au moins deux) :

- Dans la première clause juste après *select*, on présente les champs à afficher.

- La deuxième clause désigne la table source d'informations, cette clause commence par le mot clé **from**.
- La troisième clause **where** détermine un critère de recherche (filtre).
- Une commande SQL se termine toujours par ; (point virgule)

Par exemple dans la requête ci-dessus, après **select**, on trouve les champs que l'on veut affichés : *nom, prenom, telephone*. Dans la deuxième clause après **from**, on trouve la table *ancien_tc* correspondant aux sources d'informations recherchées. La troisième clause **where** détermine un critère permettant de sélectionner les individus habitant la ville *ELBEUF*.

Dans ce qui suit suite, on va décrire d'autres aspects du langage SQL.

2.IV.2/ Des conditions de sélection plus complexes:

- La condition de sélection introduite par la clause WHERE peut être une expression avec les opérateurs AND, OR, NOT :

Exemple 1: *select nom, adresse, codepostal, ville*
 from ancien_tc
 where ville = "Le Havre" and bac = "G3" ;

Exemple 2: *select nom, adresse, codepostal, ville*
 from ancien_tc
 where ville = "Le Havre" OR ville = "MONTIVILLIER" ;

Exemple 3: *select nom, adresse, codepostal, ville*
 from ancien_tc
 where not(ville = "Le Havre");

- l'appartenance d'un champ à une liste de valeurs :

Exemple: *select nom from ancien_tc*
 where ville in ("Le Havre", "MONTIVILLIER") ;

- l'appartenance d'un champ à un intervalle :

Exemple: *select nom from ancien_tc*
 where annee between 90 and 95;

- l'utilisation d'un **masque** qui décrit une partie de la valeur d'un champ par exemple pour afficher les dresses des élèves dont le nom commence par CH on peut lancer la requête suivante :

```

select nom, prenom, codepostal , ville, adresse
from ancien_tc
where nom like "CH*" and annee = 96 ;

```

l'exécution de cette requête sous Access affichera :

CHAKOUR	Zohra	76500	ELBEUF	38, rue Arthur Hulme
CHAUVRI	Lorna	76250	DEVILLE LES	14 RUE DE VERDUN
CHERON	Sandra	27130	VERNEUIL SUR	LE HAUT BROU
CHOET	Elisabeth	27100	VAL DE REUIL	8 IMPASSE DES

IV.3/ Données extraites et données dérivées:

jusqu'à présent nos données proviennent directement de la base de données. Une autre possibilité de SQL, l'instruction select peut spécifier des données calculées (données dérivées) à partir des valeurs venant de la base de données.

Exemple: Considérons la table **produit** décrit ci-dessous

Table produit

<i>numéro</i>	<i>libelle</i>	<i>prix</i>	<i>qstock</i>	<i>stock_minimum</i>
---------------	----------------	-------------	---------------	----------------------

Considérons aussi la requête : *select libelle, (prix * qstock) from produit ;*

Cette requête va afficher les valeurs de tous les produits existant dans le magasin (un montant par produit).

Exercice: afficher les montants de TVA de tous les produits en stock.

Solution : *Select libelle, ((20.5 * prix * qstock)/100) from produit;*

Exercice: afficher les montants de prix de vent (prix +TVA) de tous les produits en stock.

IV.4/ Requêtes utilisant plusieurs tables:

Ici la ressource des données se trouve sur plusieurs table

Exemple introductif : Déterminer les poursuites d'étude de l'ancien élève BEGUIR

On remarque que les données correspondant à cette interrogation se trouvent sur deux table : la table ancien_tc et la table poursuite_etudes

Solution n° 1 (solution naïve): on détermine d'abord les numéros de l'étudiant BEGUIR en posant la requête (1) suivante :

```
Select n_adut from ancien_tc where nom = "BEGUIR";
```

dont l'exécution donnerait le numéro 11.

En suite avec la requête (2) :

```
select diplôme, spécialité from poursuite_etudes where n_adut =11
```

L'exécution de cette requête donne l'affichage suivante :

LICENCE	COMMERCE INTERNATIONAL
---------	------------------------

Solution n° 2 : on combine les deux requêtes précédentes en une seule requête(3) comme suite :

```
select diplôme, spécialité from poursuite_etudes  
where n_adut = (Select n_adut from ancien_tc where nom = "BEGUIR");
```

La requête (3) précédente donne également affichage suivante :

LICENCE	COMMERCE INTERNATIONAL
---------	------------------------

En effet dans ce qui précède, pour obtenir la requête (3), on a remplacé dans la requête (2) la valeur 11 par la requête (1).

On peut également exprimer la requête 3 en remplaçant = par **in**, ceci donne une autre écriture plus générale de la requête (3) :

```
select diplôme, spécialité from poursuite_etudes  
where n_adut in (Select n_adut from ancien_tc where nom = "BEGUIR");
```

Exercice1: Afficher l'activité provisionnelle de l'ancien élève BELLECH

Exercice 2 : Trouver le lycee de l'élève BAUDOUX

On remarque que l'extraction de données issues de plusieurs tables s'exprime par des requêtes complexes et nécessitent un apprentissage adéquat, ce qui fera l'objet de plusieurs TD permettant de maîtriser cette technique. On verra aussi une méthode basée la technique de jointure.

V./ Les Système Gestion de Base de Données (SGBD) du marché :

Le marché de logiciel offre aujourd'hui des outils qui permet d'accéder à des BD sans qu'il soit nécessaire de maîtriser le langage SQL;

Citons quelques-uns :

SQLWindows,
Object Vision,
ACCESS
Oracle

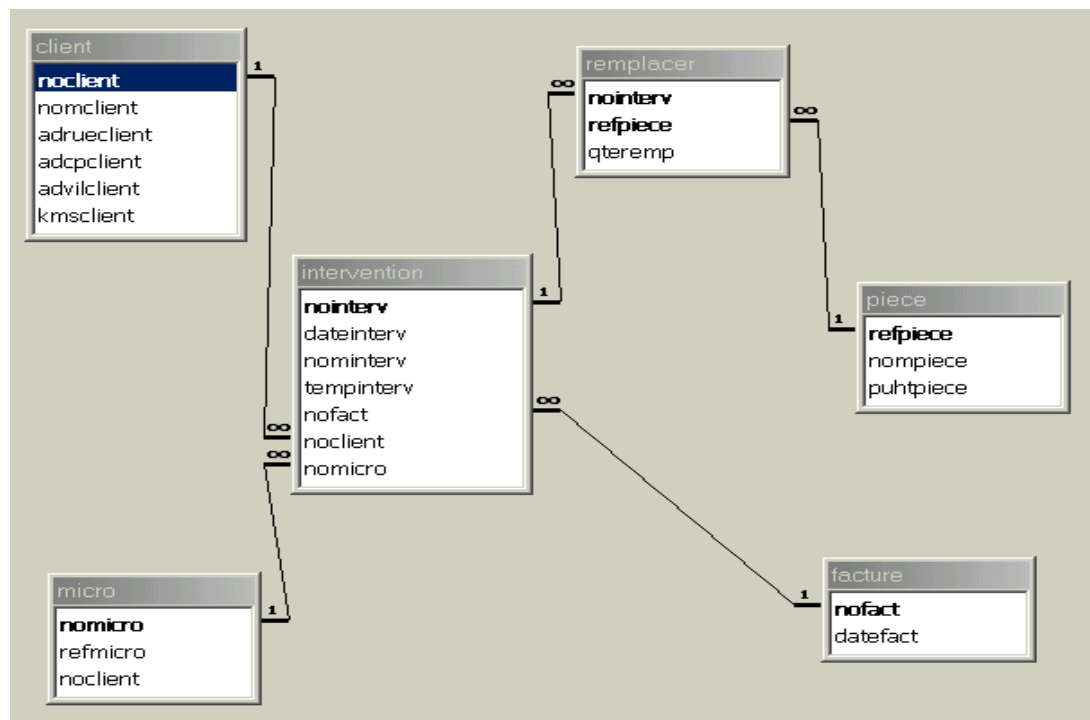
Ces outils sont cependant limités à des requêtes simples.

Exemple d'une Base de données ACCESS avec requêtes SQL

Cette BD est récupérable à l'adresse : <http://nakech.free.fr/micro.accdb>

A/ Présentation : Dans l'objectif de créer une base de données permettant de gérer un service après-vente d'une société de Micro-Informatique, on propose les 6 tables suivantes : **Client, Pièce, Micro, Facture, Intervention, et Remplacer.** Dans l'annexe I on trouve la structure, le contenu de cette BD, et la signification des champs.

Schéma conceptuel de la base de données



B/ Signification des champs

Champ	Signification
NOCLIENT	Numéro client
NOMCLIENT	Nom du client
ADRUECLIENT	Adresse rue client
ADCPCLIENT	Adresse code postal client
ADVILCLIENT	Adresse ville client
KMSCLIENT	Kilométrage séparant le domicile du client du siège social
REFPIÈCE	Référence pièce

NOMPIECE	Nom de la pièce
PUHTPIECE	Prix unitaire hors taxe pièce
NOFACTURE	N° facture
DATEFACTURE	Date de facture
NOINTERV	N° d'intervention
DATEINTERV	Date d'intervention
NOMINTERV'	Nom du réparateur
TEMPSINTERV	Temps d'intervention
QTÉREMP	Quantité remplacée lors d'une intervention
NOMICRON	N° du micro
REFMICRO	référence du micro

C/ Exemple de contenu des tables :

table client

noclient	nomclient	adrueclient	adcpclient	advilclient	kmsclient
1	LOUPLE	12 impasse des Soupis	88390	Les Fourges	10
2	LINCO	7 rue du Lac	88390	Bouzey	15
3	OCTET	8 impasse Alix	54000	Nancy	70
4	JACQUES	17 route de Nancy	57000	Metz	130
5	MEUSIEN	8 rue Marguerite	55000	Verdun	140

table mirco

nomicro	refmicro	noclient
623s	HP per	1
642e	Maxdata per	1
642o	HP pav	2
643e	compaq persar	1
643s	HP per	1
644o	Olivetto	3
645n	IBM pro	4

table intervention

nointerv	dateinterv	nominterv	tempinterv	nofact	noclient	nomicro
142	29/3/02	GERARD	1,5	101	1	642o
143	30/3/02	CLAUDE	2	101	1	642o
144	1/4/02	GERARD	1	102	3	644o
145	1/4/02	ANTOINE	3	103	4	645n
146	2/4/02	CLAUDE	2	105	2	643e
147	3/4/02	GERARD	3	105	1	642o
148	4/4/02	ANTOINE	2	106	3	644o
149	5/4/02	ANTOINE	3	106	3	623s

table remplacer

nointerv	refpiece	qteremp
142	ali200	1
142	sca058	1
143	ali200	3
144	loc014	1
145	sim123	8
147	car125	1
148	car100	2
148	car125	4

149	car100	1
149	cop485	1
149	sim123	4

table facture

nofact	datefact
101	01/04/2002
102	02/04/2002
103	03/04/2002
104	04/04/2002
105	08/04/2002
106	10/04/2002

table piece

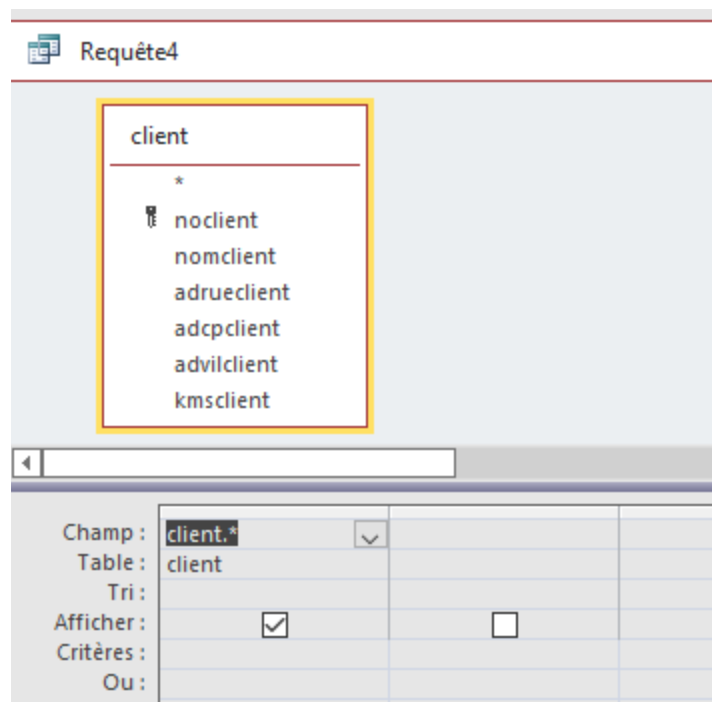
refpiece	nompiece	puhtpiece
ali200	alimentation 200w	47,00
car100	carte ide-hd	14,00
car125	carte ide mf	22,00
car152	carte video triden 32	30,00
car422	dd 20	90,00
car435	carte mere P	120,00
cop485	pro PIII	200,00
loc014	local bus	20,00
sca058	scanman	22,00
sim123	sdram 256	50,00

B/ Questions requêtes SQL :

1 / Ecrire la requête SQL permettant d'afficher la table Intervention

Réponse : `select * from Intervention`

Sous Access, pour créer une requête, on passe par les étapes : créer → création requête → obtenir la configuration suivante :



On passe ensuite : Bouton Affichage puis mode SQL ce qui donne la requête suivante :
`SELECT client.*, * FROM client;`

2/ Ecrire la requête SQL permettant d'insérer la ligne (149, car125,2) dans la table Remplacer

Réponse : `INSERT INTO piece VALUES ("149", "car125", 2);`

3 / Ecrire la requête SQL permettant d'augmenter de 10% les prix des pieces

Réponse: Update piece set puhtpiece = puhtpiece + puhtpiece * 0.1

4 / Ecrire la requête SQL affichant les client habitant à plus de 100 km de l'entreprise de réparation.

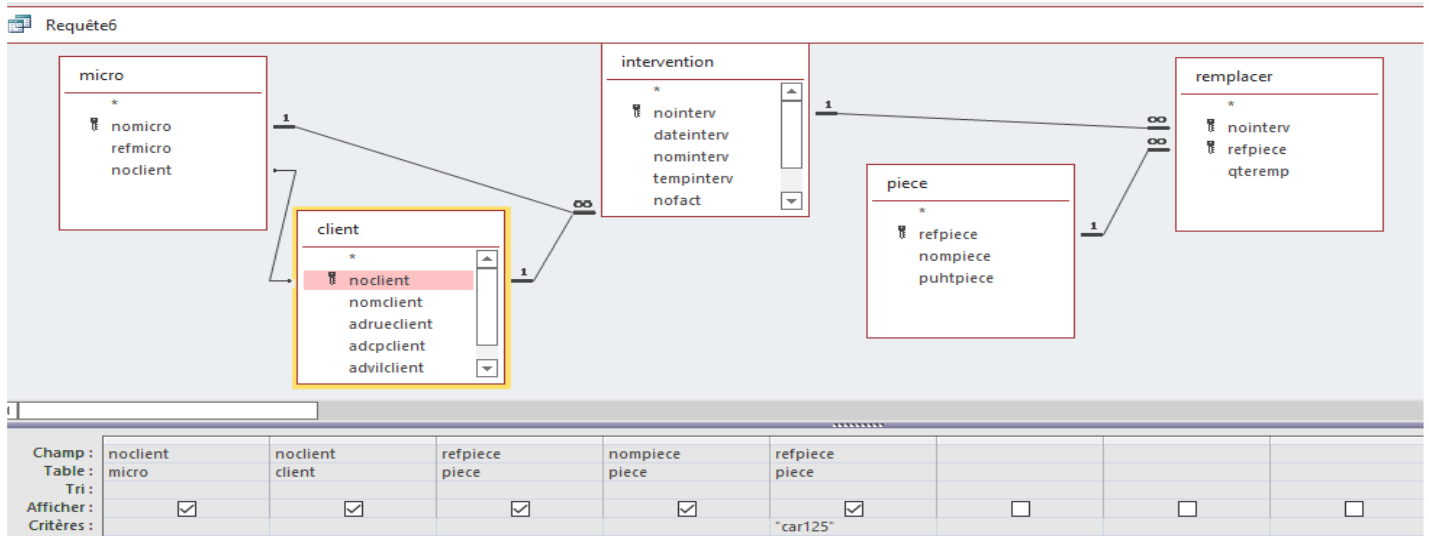
Réponse: SELECT * FROM client WHERE ((client.kmsclient)>100);

5 / Ecrire la requête SQL qui permet de modifier le prix unitaire de la pièce n° car100, le nouveau prix est 16 €

Réponse: Update piece set puhtpiece = 16 where nompiece = "car100"

6 / Ecrire la requête SQL Affichant une liste contenant le nom de client, la référence de micro (refmicro) concernant les micros pour lesquels on remplace la pièce car125 .

La creation de cette requete sous access donne :



Ce qui donne en sql

```
SELECT micro.noclient, client.noclient, piece.refpiece, piece.nompiece, piece.refpiece
FROM piece INNER JOIN ((micro INNER JOIN (client INNER JOIN intervention ON
client.noclient = intervention.noclient) ON (client.noclient = micro.noclient) AND
(micro.nomicro = intervention.nomicro)) INNER JOIN remplacer ON intervention.nointerv =
remplacer.nointerv) ON piece.refpiece = remplacer.refpiece
WHERE ((piece.refpiece)="car125");
```

7 / Ecrire la requête SQL permettant d'afficher le coût de heures réparation de l'ordinateur 642o.

Réponse :
 SELECT Sum(intervention.tempinterv) AS SommeDetempinterv, micro.nomicro
 FROM micro INNER JOIN intervention ON micro.nomicro = intervention.nomicro
 GROUP BY micro.nomicro
 HAVING ((micro.nomicro)="642o");

9/ Donner les effets des requetes SQL suivante :

i/ select sum (tempinterv), nomclient from intervention, client
 where (client.noclient = intervention.noclient) group by nomclient;

Réponse : Cette requête donne le temps d'intervention pour tous les clients

ii/ select sum (tempinterv) * 20 + sum (puhtpiece * Qteremp) * 1.20
 from intervention, remplacer , client , piece
 where (Client.Noclient = intervention.Noclient)
 and (piece.refpiece = remplacer.refpiece)
 and (remplacer.nointerv = intervention.nointerv)


```
and (nomicro = "642o");
```

Réponse :

Cette requête permet d'afficher la réparation de l'ordinateur 642o

Remarques : - Le coût d'une heure d'intervention est égale à 20 Euro
- Le TVA = 20%

```
iii/ select count(*), noclient from intervention group by noclient;
```

Réponse : *Cette requête donne le nombre d'intervention de chaque client*

Liens d'accès pour récupérer les TP Access

- [TP1 : Création d'une bd location_materiels sous Access](#)
- [TP2 Requête](#)
- [TP3 Formulaires](#)
- [Lien pour récupérer la bd location_materiels \(d'Access\)](#)

Annexe

Exemple D'une Base de données avec requêtes SQL(Oracle)

On considère ici une base de données décrivant une organisation de ventes. Cette base de données est constituée de 4 tables : employee, departement, produit, et vente dont la structure et le contenu est décrit ci-dessous.

table employee (La liste des employés)

numero	nom	salaire	adresse	dept
E7	Michel	7500.00	Paris	D2
E6	Alain	8500.00	Le Havre	D1
E5	Benoit	8000.00	Rouen	D3
E4	Said	5000.00	Evry	D3
E3	Kaulin	7000.00	Tours	D2
E2	Arthur	6000.00	Le Havre	D1
E1	Louis	8000.00	Rouen	D1
E8	Arthur	4000.00	Le Havre	D4

table departement (La liste des départements)

dnum	dnom	etage	mgr
D4	HiFi	3	E8
D3	Livres	2	E5
D2	Alimentation	3	E3
D1	Jouets	1	E1
D5	Bazar	2	E1

table produit (La liste des produits vendus)

pnum	pnom	poid	prix
P5	Linux	3	5.00
P4	Java	3	5.00
P3	Eclair	1	1.00
P2	Barbie	3	4.50
P1	CD	2	2.00
P6	Leffe	2	3.50

table vente (les ventes par département)

dnum	pnum	qt
D1	P2	100
D2	P3	200
D2	P6	300
D3	P4	100
D5	P6	100
D5	P5	200
D5	P4	400
D5	P3	300
D5	P2	100
D3	P5	300
D5	P1	200

1. Requêtes Simples

a) «Liste de tous les employés»

select * from employee;

Résultat :

numero	nom	salaire	adresse	dept
E7	Michel	7500.00	Paris	D2
E6	Alain	8500.00	Le Havre	D1
E5	Benoit	8000.00	Rouen	D3
E4	Said	5000.00	Evry	D3
E3	Kaulin	7000.00	Tours	D2
E2	Arthur	6000.00	Le Havre	D1
E1	Louis	8000.00	Rouen	D1
E8	Arthur	4000.00	Le Havre	D4

b) Avec colonnes choisies: «Le numéro et nom de tous les employés»

select numero, nom from employee;

Résultat :

numero	nom
E7	Michel
E6	Alain
E5	Benoit
E4	Said
E3	Kaulin
E2	Arthur
E1	Louis
E8	Arthur

On peut qualifier les champs par leur table et écrire

select employee.numero, employee.nom from employee;

c) Avec clause de restriction, « le numéro et le nom des employés du département "D1" »

select numero, nom

from employee

where dept = 'D1';

Résultat :

numero	nom
E6	Alain
E2	Arthur
E1	Louis

d) Avec conditions mixées, «le numéro et le nom des employés du département "D1" et ayant un salaire > 7000»

select numero, nom from employee where dept = 'D1' and salaire > 7000;

Résultat :

numero	nom
E6	Alain
E1	Louis

e) On peut spécifier un ordre d'affichage: «le nom et le salaire des employés classés par ordre des salaires»

select nom, salaire from employee order by salaire;

L'ordre est croissant de façon implicite.

Résultat :

nom	salaire
Arthur	4000.00
Said	5000.00
Arthur	6000.00
Kaulin	7000.00
Michel	7500.00
Benoit	8000.00
Louis	8000.00
Alain	8500.00

f) Idem, mais si les salaires sont identiques, afficher les noms en ordre aussi.

select nom, salaire from employee order by salaire, 1 desc;

g) Noms en ordre inverse ici.

nom	salaire
Arthur	4000.00
Said	5000.00
Arthur	6000.00
Kaulin	7000.00
Michel	7500.00
Benoit	8000.00
Louis	8000.00
Alain	8500.00

h) On peut combiner les clauses where et order by, «le nom et le salaires des employés de "D1" affichés par salaires croissants»
select nom, salaire from employee where dept = 'D1' order by salaire;

Résultat :

nom	salaire
Arthur	6000.00
Louis	8000.00
Alain	8500.00

i) Usage de distinct, qui permet d'éliminer les lignes doubles.

Avec

select dnum from vente;

on a:

dnum
D1
D2
D2
D3
D5
D5
D5
D5
D5
D3
D5

et avec *select distinct dnum from vente;*

on a:

dnum
D1
D2
D3
D5

2. Requêtes Simples avec expressions

a) «Donner pour chaque produit son poids en gramme (poid x 1000)»

*select pnum, poid * 1000 from produit;*

Résultat :

pnum	(expression)
P5	3000
P4	3000
P3	1000
P2	3000
P1	2000
P6	2000

Le champ calculé n'as pas de nom, mais on peut le lui donner, e.g. Gramme

b)

*select pnum, poid * 1000 Gramme from produit;*

Résultat :

pnum	gramme
------	--------

P5 3000
P4 3000
P3 1000
P2 3000
P1 2000
P6 2000

3. Interrogation de plusieurs tables (Jointure relationnelle)

a) Parfois, on a besoin de consulter deux tables. «Afficher toutes les informations sur les employés et le département où ils travaillent.»

select employee.*, department.* from employee, department where employee.dept=department.dnum;

Dans la clause from, on annonce les deux tables. Le lien se fait par la colonne commune, à savoir dept de employee et dnum de department. On appelle ce lien, *critère de jointure*.

numero	nom	salaire	adresse	dept	dnum	dnom	etage	mgr
E7	Michel	7500.00	Paris	D2	D2	Alimentation	3	E3
E6	Alain	8500.00	Le Havre	D1	D1	Jouets	1	E1
E5	Benoit	8000.00	Rouen	D3	D3	Livres	2	E5
E4	Said	5000.00	Evry	D3	D3	Livres	2	E5
E3	Kaulin	7000.00	Tours	D2	D2	Alimentation	3	E3
E2	Arthur	6000.00	Le Havre	D1	D1	Jouets	1	E1
E1	Louis	8000.00	Rouen	D1	D1	Jouets	1	E1
E8	Arthur	4000.00	Le Havre	D4	D4	HiFi	3	E8

Remarquer que le département D5 ne figure pas, car personne n'y travaille.

b) Sans clause de restriction, on a ce qu'on appelle le produit cartésien (toutes les combinaisons):

select employee.*, department.* from employee, department;

numero	nom	salaire	adresse	dept	dnum	dnom	etage	mgr
E7	Michel	7500.00	Paris	D2	D2	Alimentation	3	E3
E7	Michel	7500.00	Paris	D2	D1	Jouets	1	E1
E7	Michel	7500.00	Paris	D2	D3	Livres	2	E5
E8	Arthur	4000.00	Le Havre	D4	D4	HiFi	3	E8
E8	Arthur	4000.00	Le Havre	D4	D5	Bazar	2	E1

40 (8 x 5) lignes au total.

c) Mais il est plus intéressant se spécifier les informations que l'on souhaite: Donner pour chaque employé, son numéro, son nom, ainsi que le département et l'étage où il travaille.

select employee.numero, employee.nom, department.dnom, department.etage from employee, department where employee.dept=department.dnum;

Résultat :

numero	nom	dnom	etage
E7	Michel	Alimentation	3
E6	Alain	Jouets	1
E5	Benoit	Livres	2
E4	Said	Livres	2
E3	Kaulin	Alimentation	3
E2	Arthur	Jouets	1
E1	Louis	Jouets	1
E8	Arthur	HiFi	3

d) On peut rajouter une condition supplémentaire au critère de jointure. Quels sont les employés du 3 e étage?

select nom from employee, department where employee.dept=department.dnum and etage = "3";

Résultat :

nom
Michel
Kaulin
Arthur

e) Parfois, on a besoin de consulter sur trois table, et joindre donc les trois.

Donner le nom de chaque département ainsi que le noms des produits qu'il vend.

select dnom, pnom from produit , vente, department where produit.pnum = vente.pnum and department.dnum = vente.dnum;

Résultat :

dnom	pnom
Livres	Java

Livres	Linux
Alimentation	Eclair
Alimentation	Leffe
Jouets	Barbie
Bazar	CD
Bazar	Barbie
Bazar	Eclair
Bazar	Java
Bazar	Linux
Bazar	Leffe

e) On peut avoir besoin de consulter deux fois la même table. Par exemple pour *afficher par paires d'employés habitant la même ville*.

select x.nom, y.nom from employee x, employee y where x.adresse = y.adresse and x.numero > y.numero;

On emploie alors des variables (Louisas) comme ici x et y, pour se référer à deux employés simultanément.

nom	nom
Arthur	Alain
Alain	Arthur
Arthur	Arthur
Benoit	Louis

La condition supplémentaire $x.numero > y.numero$, est une astuce pour éliminer les paires symétriques ou deux fois la même personne.

f) Une requête caractéristique est *le nom des employés qui gagnent plus que leur manager*. Jointure entre trois tables, dont deux fois la même.

select x.nom from employee x, employee y, department d where x.dept = d.dnum and d.mgr = y.numero and y.salaire < x.salaire;

Ce qui donne:

nom
Michel
Alain

4. fonctions incorporées

select count() from employee;*

Résultat :

(count(*))
8

select count (distinct nom) from employee;

Résultat :

(count)
7

select avg(qt) from vente;

Résultat :

(avg)
209.09

select avg(qt) from vente where pnum = "P4";

Résultat :

(avg)
250.00

select sum(qt) / count() from vente where pnum = "P4";*

Résultat :

(expression)
250.00

5. Usage de group by

select pnum, avg(qt) from vente group by pnum;

Résultat :

pnum	(avg)
P2	100.00
P3	250.00
P6	200.00
P4	250.00
P5	250.00
P1	200.00

select pnum, avg(qt) from vente group by pnum order by 2;

Résultat :

pnum	(avg)
P2	100.00
P6	200.00
P1	200.00
P3	250.00
P4	250.00
P5	250.00

```
select pnum, sum(qt)
from vente
```

group by pnum having avg(qt) >200;

Résultat :

pnum	(sum)
P3	500
P4	500
P5	500

requete complete

select d.dnom, sum(qt) from department d, vente s where d.etage >1 and d.dnum = s.dnum group by d.dnom having avg(qt)>200 order by 2 desc;

Résultat :

dnom	(sum)
Bazar	1300
Alimentation	500

{produit dont le poids s'ecarte le plus de la moyenne}

- *create table t1 (moy decimal(8,2));*
- *insert into t1 select avg(poid) from produit;*
- *select * from t1;*

Résultat :

moy
2.33

- *create table t2 (pnum char(4), ecart decimal(8,2));*
- *insert into t2 select pnum, poid - moy from produit, t1;*
- *update t2 set ecart = ecart*(-1) where ecart <0;*
- *select * from t2;*

Résultat :

pnum	ecart
P5	0.67
P4	0.67
P3	1.33
P2	0.67
P1	0.33
P6	0.33

- *select pnum from t2 where ecart = (select max(ecart) from t2);*
- *drop table t1; drop table t2;*

Résultat :

pnum
P3

6. Caractéristiques Avancées

- **LIKE**

select nom, adresse from employee where adresse like "Raba_";

Résultat :

nom	adresse
Louis	Rouen
Benoit	Rouen

select nom, adresse from employee where adresse like "%a%";

Résultat :

nom	adresse
Alain	Le Havre
Benoit	Rouen
Said	Evry
Kaulin	Tours
Arthur	Le Havre
Louis	Rouen
Arthur	Le Havre

select nom, adresse from employee where nom like "%e_";

Résultat :

nom	adresse
Kaulin	Tours
Arthur	Le Havre
Arthur	Le Havre

- **BETWEEN**

select salaire, nom from employee where salaire between 7000 and 8000;

Résultat :

salaire	nom
7500.00	Michel
8000.00	Benoit
7000.00	Kaulin
8000.00	Louis

- **IN**

select numero, nom from employee where numero in ("E5", "E7", "E2");

Résultat :

numero	nom
E2	Arthur
E7	Michel
E5	Benoit
bloc	SELECT

select numero, nom from employee where dept in (select dnum from department where etage = 3);

Résultat :

numero	nom
E7	Michel
E3	Kaulin
E8	Arthur

select numero, nom from employee where dept in (select dnum from department where mgr in (select numero from employee where adresse="Tours"));

Résultat :

numero	nom
E7	Michel
E3	Kaulin

select numero, nom from employee x where dept in (select dnum from department where mgr in (select numero from employee y where x.salaire > y.salaire));

Résultat :

numero	nom
E7	Michel
E6	Alain

usage mixte de join et group select

*select numero, nom from employee x, department d where d.dnum = x.dept and mgr in
(select numero from employee y where x.salaire > y.salaire);*

Résultat :

numero	nom
E7	Michel
E6	Alain

- **EXIST** (cf IN ci-dessus)

*select numero, nom from employee e where exists (select * from department d where d.dnum=e.dept and etage = 3);*

Résultat :

numero	nom
E7	Michel
E3	Kaulin
E8	Arthur

not EXISTS

*select dnum, dnom from department d where not exists (select * from employee e where e.dept = d.dnum);*

Résultat :

dnum	dnom
D5	Bazar

le dept qui vend tous les produits

*select dnum from department d where not exists (select * from produit p
where not exists (select * from vente s where d.dnum = s.dnum and p.pnum=s.pnum));*

Résultat :

dnum
D5

insert into t (SQL d oracle)

select d.dnum, dnom, etage, mgr, pnum, qt from department d, vente s where d.dnum = s.dnum;

insert into t (dnum, dnom, etage, mgr)

*select * from department d where not exists (select * from vente s where s.dnum = d.dnum);*

*select * from t;*

drop table t;

Résultat :

dnum	dnom	etage	mgr	pnum	qt
D1	Jouets	1	E1	P2	100
D2	Alimentation	3	E3	P3	200
D2	Alimentation	3	E3	P6	300
D3	Livres	2	E5	P4	100
D5	Bazar	2	E1	P6	100
D5	Bazar	2	E1	P5	200
D5	Bazar	2	E1	P4	400
D5	Bazar	2	E1	P3	300
D5	Bazar	2	E1	P2	100
D3	Livres	2	E5	P5	300
D5	Bazar	2	E1	P1	200
D4	HiFi	3	E8		