



T.D. : le langage SQL (Structured Query Language)

Détail de l'activité

- Interroger une base de données à l'aide de requêtes SQL
- Utiliser la jointure dans leur interrogation.

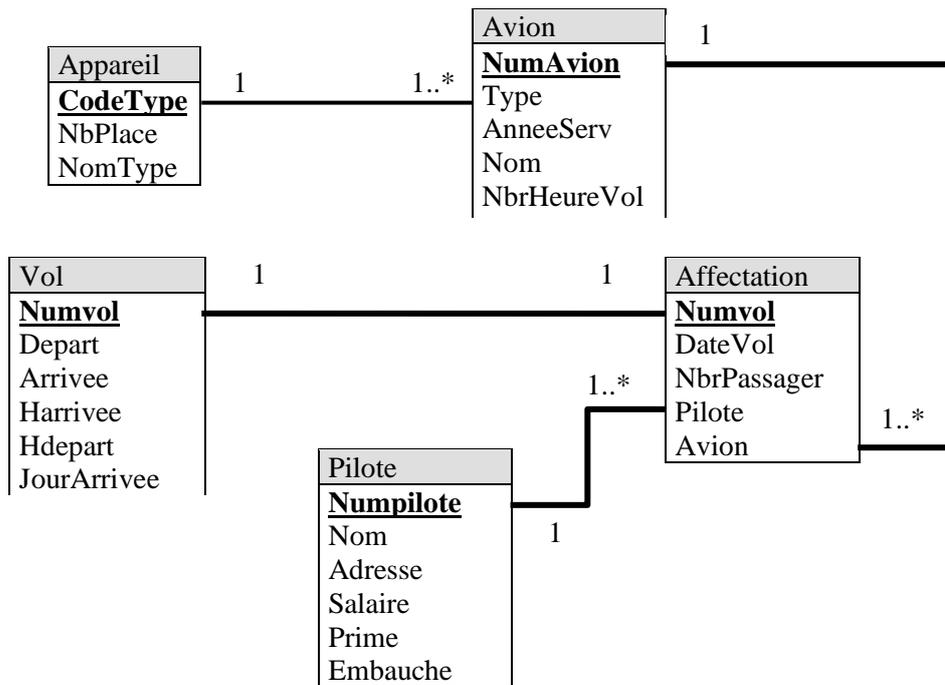
Logiciel :

XAMPP : kit d'installation d'Apache qui contient MySQL, PHP et un serveur Web

Mise en situation :

La base de données étudiée dans les exercices gère de façon simplifiée un trafic de transport aérien.
La structure de la base est donnée ci-dessous :

Schéma des tables



Description des tables

Vol		
<u>Numvol</u>	<u>Char(6)</u>	numéro identification d'un vol
Depart	Char(30)	ville de départ
Arrivee	Char(30)	ville d'arrivée
Harrivee	Date	heure d'arrivée
Hdepart	Date	heure de départ
JourArrivee	Int	jour d'arrivée (0: même jour, 1: lendemain)

Pilote		
<u>Numpilote</u>	Char(4)	numéro de matricule du pilote
Nom	Char(35)	nom du pilote
Adresse	Char(30)	adresse du pilote, limitée à la ville
Salaire	Int	salaire mensuel
Prime	Int	commission mensuel qui peut ne pas exister
Embauche	date	date d'embauche

Appareil		
<u>CodeType</u>	Char(3)	code normalisé désignant une famille d'avions
NbPlace	Int	nombre de places
NomType	Char(50)	nom de la famille d'avions

Avion		
<u>NumAvion</u>	Char(4)	numéro d'immatriculation d'un avion
Type	Char(3)	code normalisé désignant une famille d'avions
AnneeServ	Int	année de mise en service
Nom	Char(50)	nom donné à un avion qui peut ne pas être connu
NbrHeureVol	Int	nombre heures de vol depuis la mise en service

Affectation		
<u>Numvol</u>	Char(6)	numéro identification d'un vol
DateVol	Date	date du vol sous la forme: jj.mm.aa
NbrPassager	Int	nombre effectif de passagers
Pilote	Char(4)	numéro du pilote conduisant l'avion pour le vol
Avion	Char(4)	numéro d'immatriculation de l'avion affecté au vol



Sélection de colonnes : commande select

La commande *SELECT* effectue des opérations de sélection de données sur plusieurs tables relationnelles par projection.

Sa syntaxe est la suivante:

```
SELECT [ALL] | [DISTINCT] <liste des noms de colonnes> | *
FROM <Liste des tables>
[WHERE <condition logique>]
```

Il existe d'autres options pour la commande *SELECT*:

GROUP BY
HAVING
ORDER BY

Question N°1 : Tapez « `select * from avion;` ». Quel est la table ? Quels sont ses champs ?

Question N°2 : Tapez « `select numavion from avion;` » puis « `select numavion, type from avion;` ». Quels sont les champs affichés ?

Question N°3 : Listez tous enregistrements les pilotes de la table pilote.

Question N°4 : Listez uniquement le nom et les adresses des pilotes.

Tri des résultats : Il est possible en SQL d'organiser les résultats grâce à la clause *ORDER BY*. Elle est suivie des mots clés *ASC* ou *DESC*, qui précisent respectivement si le tri se fait de manière croissante (par défaut) ou décroissante. Le classement se fait sur des nombres ou des chaînes de caractères.

Question N°5 : Tapez les exemples « `select * from pilote order by nom;` », « `select * from pilote order by nom ASC;` » et « `select * from pilote order by nom DESC;` », quels sont les différences ?

Question N°6 : Sélectionnez les noms et les adresses des pilotes de la table « pilote ». (classement des noms par ordre alphabétique).

Question N°7 : Tapez « `select numpilote as "Numéro de pilote" from pilote ;` » Quel est le nom de la colonne ?.

Question N°8 : Tapez « `select numpilote, salaire + 111, nom from pilote;` » Quel est le nom de la deuxième colonne ?..

Question N°9 : Listez tous les noms des pilotes et de leur revenu - égal au salaire augmenté de leur prime- (revenu = salaire + prime).

Question N°10 : Même question mais remplacez le nom de la colonne « salaire + prime » par « Revenu ». (Utilisez la question N°7).

Restrictions simples

Une restriction consiste à sélectionner les lignes satisfaisant à une condition logique effectuée sur leurs attributs.

En SQL, les restrictions s'expriment à l'aide de la clause *WHERE* suivie d'une condition logique exprimée à l'aide de :

opérateurs logiques	comparateurs arithmétiques:
<ul style="list-style-type: none"> • AND • OR • NOT 	<ul style="list-style-type: none"> • = • != • > • < • >=
comparateurs de chaîne	



- IN
- BETWEEN
- LIKE

- <=
- <>
- !>
- !<

Question N°11 : Listez les avions dont l'année de mise en service est 2000. (AnneeServ)

Question N°12 : Listez les avions dont le type est 734.

Question N°13 :

Question N°14 : Tapez l'exemple : « **select** numavion, type, nbrheurevol **from** avion **where** nbrheurevol > 10000 ». Quel condition ces enregistrement vérifie-t-il ?

Question N°15 : Même question avec « **select** numavion, type, nbrheurevol **from** avion **where** nbrheurevol < 10000 **AND** AnnéeServ > 1995 »

Question N°16 : Sélectionnez les avions mis en service avant 1999.

Question N°17 : Sélectionnez les avions mis en service entre 1995 et 2000.

Question N°18 : Affichez la liste alphabétique des pilotes qui ont un salaire supérieur à 20000 et qui ne perçoivent aucune prime.

Question N°19 : Donnez la liste des vols qui relie Marseille à Francfort ou à Londres.

Restriction sur une comparaison de chaîne

Le prédicat *LIKE* permet de faire des comparaisons sur des chaînes grâce à des caractères, appelés caractères *jokers*:

- Le caractère % permet de remplacer une séquence de caractères (éventuellement nulle)
- La caractère _ permet de remplacer un caractère (l'équivalent du "blanc" au scrabble...)

Question N°20 :

Question N°21 : Tapez l'exemple « **select** * **from** pilote **where** nom **like** 'B%' »

Question N°22 : Listez les pilotes dont l'adresse est composée de 3 lettres. Même question avec 4 lettres.

Question N°23 : Donnez la liste des pilotes dont le nom contient la lettre « l ».

Regroupement de résultats

Il peut être intéressant de regrouper des résultats afin de faire des opérations par groupe (opérations statistiques par exemple). Cette opération se réalise à l'aide de la clause *GROUP BY*, suivie du nom de chaque colonne sur laquelle on veut effectuer des regroupements.

Les principales fonctions pouvant être effectuées par groupe sont:

- **AVG**: Calcule la moyenne d'une colonne (ou de chaque regroupement si elle est couplée à la clause *GROUP BY*)
- **COUNT**: Calcule le nombre de lignes d'une table (ou de chaque regroupement ...)
- **MAX**: Calcule la valeur maximale d'une colonne (ou de chaque regroupement ...)
- **MIN**: Calcule la valeur minimale colonne (ou de chaque regroupement ...)
- **SUM**: Effectue la somme des valeurs d'une colonne (ou de chaque regroupement ...)

Question N°24 : Exécutez la requête : « **select** avg(salaire) **from** pilote; »

Question N°25 : Donnez le nombre d'avions.

Question N°26 : Donnez la requête permettant d'obtenir le résultat suivant :

	moyenne	maximum	minimum
▶	20227,2727	24000	15600

La jointure

La jointure permet de mettre en relation deux (ou plusieurs tables).



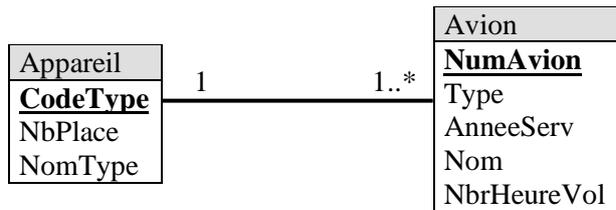
La condition de jointure est appelée « pivot de jointure ».

En SQL, l'expression d'une jointure se fait en précisant le nom des colonnes des tables sur lesquelles on fait la jointure, en désignant les colonnes des différentes tables en écrivant le nom de la table, suivie d'un point puis du nom de la colonne. La clause *WHERE* permet de préciser la qualification de la jointure.

Syntaxe :

```
SELECT <liste des noms de colonnes> | *
FROM Nom-de-la-table1, Nom-de-la-table2
WHERE table1.pivot=table2.pivot
```

Soient les tables Avion et Appareil : la première contient tous les avions de la base de données et la seconde donne pour chaque type d'avion son nombre de place et son nom.



La clef primaire de la table Avion est *NumAvion* et celle de Appareil est *CodeType*. La relation entre ces deux tables se fait par *CodeType* et *Type*.

La jointure des deux tables par la relation *CodeType = Type* permet de compléter les caractéristiques propres à chaque avion :

<u>NumAvion</u>	Type	Annee Serv	Nom	NbrHeure Vol	NbPlace	NomType
8437	734	1999	Paris - Dakar	11000	150	Boeing 747-400 COMBI
8124	AB7	2000	Malte	8000	180	Airbus A300
800	320	1999	British Air 2	10000	100	Boieng 747-100
700	320	1999	British Air 3	4500	100	Boieng 747-100
...

Pour la suite des questions utilisez l'annexe : *Liaison entre les tables.*

Question N°27 : Affichez la liste des avions avec numéro d'avion, nom de l'avion et nombre de place. (la condition de jointure est avion.type=appareil.codetype).

Question N°28 : On désire connaître les pilotes qui sont affectés à tous les vols. Complétez et exécutez la requête : « select NumVol,affectation.pilote,pilote.nom from affectation,pilote where = ; » Quelle est la signification de affectation.pilote et de pilote.nom ?

Question N°29 : Listez les avions qui sont affectés à tous les vols.

Question N°30 : Listez le numéro de vol, le numéro de l'avion et le nom des pilotes pour tous les vols de la table affectation.

La création de tables

La création de tables se fait à l'aide du couple de mots-clés *CREATE TABLE*.

Le nom donné à la table doit généralement (sur la plupart des SGBD) commencer par une lettre, et le nombre de colonnes maximum par table est de 254.

Sa syntaxe est la suivante:

```
CREATE TABLE Nom_de_la_table (Nom_de_colonne1 Type_de_donnée,
```



```
Nom_de_colonne2 Type_de_donnée,  
...);
```

Les types de données :

Pour chaque colonne que l'on crée, il faut préciser le type de données que le champ va contenir. Celui-ci peut être un des types suivants:

Type de donnée	Syntaxe	Description
Type alphanumérique	CHAR(n)	Chaîne de caractères de longueur fixe n ($n < 16383$)
Type alphanumérique	VARCHAR(n)	Chaîne de caractères de n caractères maximum ($n < 16383$)
Type numérique	NUMBER(n,[d])	Nombre de n chiffres [optionnellement d après la virgule]
Type numérique	SMALLINT	Entier signé de 16 bits (-32768 à 32757)
Type numérique	INTEGER	Entier signé de 32 bits (-2E31 à 2E31-1)
Type numérique	FLOAT	Nombre à virgule flottante
Type horaire	DATE	Date sous la forme 16/07/99
Type horaire	TIME	Heure sous la forme 12:54:24.85
Type horaire	TIMESTAMP	Date et Heure

L'option *NOT NULL*, placée immédiatement après la type de donnée permet de préciser au système que la saisie de ce champ est obligatoire.

Par exemple :

```
create TABLE clients (Nom char(30), Prenom char(30) );
```

Question N°31 : Créez la table Appareil_v2 ;

La destruction de tables

La création de tables se fait à l'aide du couple de mots-clés *DROP TABLE* où table est le nom de la table à supprimer

```
DROP TABLE ;
```

Question N°32 : Créez une autre table Appareil_v3 puis supprimer la.

Insertion de données

L'insertion de nouvelles données dans une table se fait grâce à l'ordre *INSERT*, qui permet d'insérer de nouvelles lignes dans la table.

L'ordre *INSERT* attend la clause *INTO*, suivie du nom de la table, ainsi que du nom de chacune des colonnes entre parenthèses (les colonnes omises prendront la valeur *NULL* par défaut).

Les valeurs à insérer doivent être précisées avec la clause de *VALUES* : une seule ligne est insérée, elle contient comme valeurs, l'ensemble des valeurs passées en paramètre dans la parenthèse qui suit la clause *VALUES*. Les données sont affectées aux colonnes dans l'ordre dans lequel les colonnes ont été déclarées dans la clause *INTO*

Syntaxe :

```
INSERT INTO Nom_de_la_table (colonne1, colonne2, colonne3, ...)  
VALUES (Valeur1, Valeur2, Valeur3, ...)
```

Lorsque chaque colonne de la table est modifiée, l'énumération de l'ensemble des colonnes est facultatif.

Question N°33 : Insérez dans la table appareil_v2 les données suivantes :



CodeType	NbPlace	NomType
320	100	Boeing 747-100
47	150	Boeing 747-150
478	150	Boeing 747-150
487	150	Airbus 320-150
545	200	Boeing 200
546	250	Boeing 250
734	150	Boeing 747-400
737	80	Concorde
878	250	Airbus A300-250
aze	1005	Airbus Boeing
AB7	180	Airbus A300

Attention : les chaînes de caractères sont entourés par des ‘ et par des ". Exemple ‘Airbus A300’.

La suppression des données

La suppression de données dans une table se fait grâce à l'ordre *DELETE*. Celui-ci est suivi de la clause *FROM*, précisant la table sur laquelle la suppression s'effectue, puis d'une clause *WHERE* qui décrit la qualification, c'est-à-dire l'ensemble des lignes qui seront supprimées.

L'ordre *DELETE* est à utiliser avec précaution car l'opération de suppression est irréversible. Il faudra donc s'assurer dans un premier temps que les lignes sélectionnées sont bien les lignes que l'on désire supprimer!

Question N°34 : Supprimez la ligne dont le CodeType est ‘aze’.

La modification de données

La modification de données (aussi appelée *mise à jour*) consiste à modifier des tuples (des lignes) dans une table grâce à l'ordre *UPDATE*. La modification à effectuer est précisé après la clause *SET*. Il s'agit d'une affectation d'une valeur à une colonne grâce à l'opérateur = suivi d'une expression algébrique, d'une constante ou du résultat provenant d'une clause *SELECT*. La clause *WHERE* permet de préciser les tuples sur lesquels la mises à jour aura lieu

```
UPDATE Nom_de_la_table  
SET Colonne = Valeur_Ou_Expression  
WHERE qualification
```

Question N°35 : Tous les vols dont l'arrivée est ‘LONDRES’ voient leurs jours d'arrivés mise à jours à la valeur 3. Ecrivez cette requête.

Question N°36 : Tous les pilotes ont eu une augmentation de 10%. Mettez à jour la table.

Question N°37 : Les pilotes dont la prime est nulle se voient gratifier de 1000€. Mettez à jour la table



Question N°38 :