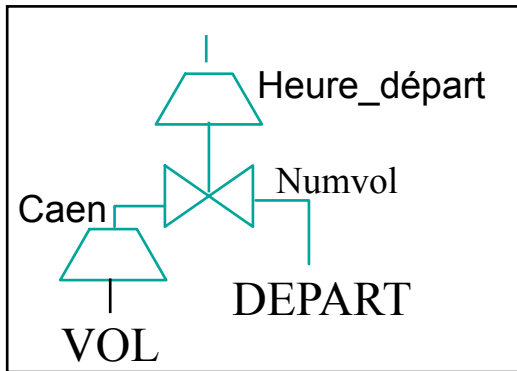


SQL, LE LANGAGE D'INTERROGATION DES SGBD RELATIONNELS

Généalogie du langage SQL



**Algèbre
relationnelle**

{V.Heure_Départ/ Vol (V)
et V.Ville_arrivée = 'Caen'
et \exists D / Départ (D)
et D.Numvol = V.Numvol
et D.Date = '19-12-95'}

**Calcul de
tuples**

Autre

SQL

Evolutions

- **Il existe trois versions normalisées, du simple au complexe :**
 - SQL1 86 version minimale
 - SQL1 89 addendum (intégrité)
 - SQL2 (92) langage complet à 3 niveaux
- **Une version 3 étendue (objets, règles) est la norme 99.**
- **La plupart des systèmes supportent SQL2 complet**

SQL (Structured Query Language)

➤ **SQL comprend quatre parties :**

- **Le langage de définition de schéma (Tables, Vues, Droits)**
- **Le langage de manipulation (Sélection et mises à jour)**
- **La spécification de modules appelables (Procédures)**
- **L'intégration aux langages de programmation (Curseurs)**

SQL

➤ **LANGAGE DE DEFINITIONS DE DONNEES**

- CREATE TABLE, ALTER TABLE, DROP TABLE, ...
- CREATE VIEW, DROP VIEW,
- CREATE INDEX, DROP INDEX, ...
- CREATE USER, ... DROP USER, CREATE GROUP

➤ **LANGAGE DE MANIPULATION DE DONNEES**

- SELECT OPEN
- INSERT FETCH
- UPDATE CLOSE
- DELETE

➤ **LANGAGE DE CONTROLE DE DONNEES**

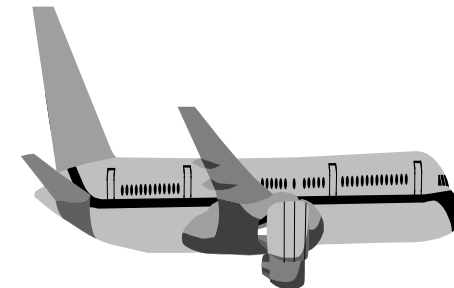
- GRANT et REVOKE
- BEGIN et END TRANSACTION
- COMMIT et ROLLBACK

Compagnie Aerienne (1)

AVION	Numav	Capacité	Type	Entrepot
	14	25	A400	Garches
	345	75	B200	Maubeuge

PILOTE	Matricule	Nom	Ville	Age	Salaire
	1	Figue	Cannes	45	28004
	2	Lavande	Touquet	24	11758

PASSAGER	Numab	Nomab
	1	Nifance
	8	Téarice



Compagnie Aérienne (2)

VOL	Numvol	Heure_départ	Heure_arrivée	Ville_départ	Ville_arrivée
	AL12	08-18	09-12	Paris	Lilles
	AF8	11-20	23-54	Vaux	Rio

DEPART	Numvol	Date_dep	Numav	Matricule
	AL12	31-DEC-95	14	1
	AL12	19-DEC-95	345	2

RESERVATION	Numab	Numvol	Date_dep
	1	AL12	31-DEC-95
	2	AL12	31-DEC-95



Description des données

➤ Création d ' une relation

```
CREATE TABLE Pilote (  
    pl#      Number(4) PRIMARY KEY,  
    plnom    char(20),  
    adresse  char(30));
```

➤ Création d ' une relation avec contraintes d ' intégrité

```
CREATE TABLE Pilote (  
    pl#      Number(4) PRIMARY KEY NOT NULL,  
    plnom    char(20 NOT NULL),  
    adresse  char(30));
```

➤ Création d ' index

```
CREATE UNIQUE INDEX IPL# ON Pilote(pl# ASC);  
CREATE UNIQUE INDEX IPL# ON Pilote(pl# DESC);
```


Description de données ...

➤ **Suppression d ' une relation**

```
DROP TABLE Pilote;
```

➤ **Modification du schéma d ' une relation**

```
ALTER TABLE Pilote ADD ( age number (2));
```

```
ALTER TABLE Pilote MODIFY(adresse char (50));
```

```
ALTER TABLE Pilote MODIFY(pl# not null);
```

➤ **Création d ' une vue**

```
CREATE VIEW VoldeParis AS (
```

```
    SELECT vol#, va, hd, ha
```

```
    FROM Vol
```

```
    WHERE vd = ' Paris ' );
```

CREATION DE TABLE : SYNTAXE

CREATE TABLE <nom_table> (<def_colonne> * [<def_contrainte_table>*]) ;

< def_colonne > ::= <nom_colonne> < type >

[CONSTRAINT nom_contrainte <

NOT NULL |

UNIQUE |

PRIMARY KEY |

CHECK (condition) |

REFERENCES nom_table (colonne) >]

< def_contrainte_table > ::= CONSTRAINT nom_contrainte <

UNIQUE (liste_colonnes) |

PRIMARY KEY (liste_colonnes) |

CHECK (condition) |

FOREIGN KEY (liste_colonnes) REFERENCES nom_table (liste_colonnes)>

Manipulation des données (1)

Insertion de données

➤ Syntaxe :

```
INSERT INTO <relation name> [( attribute [,attribute] ... )]  
{VALUES (<value spec.> [, <value spec.>] ...) | <query spec.>}
```

➤ Exemple :

```
INSERT INTO Pilote values ( 1, ' Antoine ', ' Annecy ' );  
INSERT INTO Pilote values ( 2, ' Arthur ', ' Adge ' );  
INSERT INTO Pilote values ( 3, ' Jean ', NULL);
```

```
INSERT INTO Pilote (pl#, plnom, adresse)  
  SELECT id, name, NULL  
  FROM New_Pilote;
```

Manipulation des données (2)

Suppression des données

➤ **Syntaxe :**

```
DELETE FROM <relation name>  
[WHERE <search condition>]
```

➤ **Exemple :**

```
DELETE AVION;
```

```
DELETE FROM Pilote WHERE pl# = 1;
```

Manipulation des données (3)

Modification des données

➡ Syntaxe :

```
UPDATE <relation name>  
SET <attribute = {value expression | NULL}  
[<attribute> = {value expression | NULL}] ...  
[WHERE <search condition>]
```

➡ Exemple

```
UPDATE Pilote  
SET adresse = ' Aurillac '  
WHERE pl# = 2;
```

Contrôle des accès

➤ Fin normale (validation) d ' une transaction

COMMIT;

➡ commit rend effectifs les changements effectués sur la base. Ils deviennent visibles aux autres utilisateurs

➤ Annulation d ' une transaction

ROLLBACK;

➡ rollback défait toutes les modifications effectuées sur la base depuis le début de la transaction

(1) PROJECTION, RESTRICTION

(1.1) Le principe

```
SELECT AVION.numav  
FROM AVION  
WHERE AVION.capacité > 200;
```

Projection seule:

```
SELECT AVION.numav  
FROM AVION;
```

Restriction seule:

```
SELECT AVION.*  
FROM AVION  
WHERE  
AVION.capacité > 200;
```

Exercices

Exercice n° 3.1: Quel est l'âge du pilote de matricule 15 ?

Exercice n° 3.2: Quels sont les pilotes de cette compagnie aérienne ?
(leur matricule, leur nom, leur ville, leur âge et leur salaire).

(1.2) Fonctionnalités diverses

(a) Tri

```
SELECT AVION.numav  
FROM AVION  
WHERE AVION.CAPACITE > 200  
ORDER BY AVION.numav DESC;  
(ou ASC)
```

(c) Omission du préfixe

```
SELECT numav  
FROM AVION  
WHERE CAPACITE > 200;
```

(b) Suppression des doubles

```
SELECT DISTINCT AVION.numav  
FROM AVION  
WHERE AVION.CAPACITE > 200;  
(ou ALL)
```

(d) Emploi de variables

```
SELECT A.numav  
FROM AVION A  
WHERE A.CAPACITE > 200;
```

Exercices

Exercice n° 3.3: Quels sont les types d'avions entreposés à Tarascon, avec leur capacité, par ordre de capacités croissantes ?

(e) Prédicats dans le critère de sélection:

=	!=	>	<	>=	<=	NOT	OR	AND	IN	BETWEEN	x and y
						LIKE		NOT NULL			

SELECT NUMAV

FROM AVION

WHERE CAPACITE > 200

AND ((ENTREPOT = 'Saint nom la Bretèche') **OR** (ENTREPOT = 'Mantes la Jolie'))

AND NUMAV != 300;

SELECT NUMAV

FROM AVION

WHERE ENTREPOT **IN** ('Saint nom la Bretèche' , 'Mantes la Jolie')

AND CAPACITE **BETWEEN** 200 **AND** 210

AND TYPE **LIKE** 'B_EING%';

Exercices

Exercice n° 3.4: Quelles sont les villes ayant un nom commençant par 'L' et où un pilote de moins de 20 ans gagne plus de 200000F ?

Exercice n° 3.5: Quels sont les numéros d'avions, avec leur entrepôt, ne résidant pas à Marolles-en-Hurepoix, ayant une capacité supérieure à 200 ?
(donnés par ordre de numéros décroissants).

(f) Les fonctions:

Lower(a), Soundex (a), Abs(a), Sqrt(n), Length(a), etc...

```
SELECT NUMAV  
FROM AVION  
WHERE LOWER (ENTREPOT) = 'le vésiney';
```

```
SELECT MATRICULE, NOM  
FROM PILOTE  
WHERE SOUNDEX (NOM) = SOUNDEX ('SINTEXUPERI');
```

```
SELECT UPPER (NOM)  
FROM PILOTE  
WHERE ADRESSE = 'Cordillère des Andes';
```

Exercices

Exercice n° 3.6: Quelles sont les villes d'où part un vol pour Ajaccio?
(donner les résultats en majuscule).

(g) Les agrégats

Avg, Count, Max, Min, Sum, Stddev, Variance

Exemple:

Quel est le nombre d'avions ayant une capacité supérieure à 200?

SELECT COUNT (NUMAV)

FROM AVION

WHERE CAPACITE > 200;

SELECT COUNT (DISTINCT NUMAV)

FROM AVION

WHERE CAPACITE > 200;

(ou ALL)

Exercices

Exercice n° 3.7: Quel est le nombre d'avions en services (i.e. dans la relation DEPART), le 19 décembre 1995 ?

Exercice n° 3.8: Quelle est la capacité moyenne des avions entreposés à Pogo Togo ?

(2) LES REQUETES IMBRIQUEES

(2.1) Prédicat d'existence:

```
SELECT  
FROM  
WHERE EXISTS ( SELECT  
                FROM  
                WHERE );
```

Exemple:

Quels sont les pilotes qui habitent dans une ville où n'est basé aucun avion ?

```
SELECT PILOTE.NOM  
FROM PILOTE  
WHERE NOT EXISTS ( SELECT AVION.*  
                    FROM AVION  
                    WHERE AVION.ENTREPOT =PILOTE.VILLE);
```

(2.2) Requête à l'intérieur du IN

Exemple: Quels sont les noms des pilotes qui habitent dans une ville où sont localisés des avions de capacité supérieure à 250 ?

```
SELECT PILOTE.NOM  
FROM PILOTE  
WHERE VILLE IN ( SELECT ENTREPOT  
                  FROM AVION  
                  WHERE CAPACITE > 250);
```

Exercices

Exercice n°3.9: Exprimer la requête suivante avec un "EXISTS":
Quels sont les noms des pilotes qui habitent dans une ville où sont localisés des avions de capacité supérieure à 250 ?

Exercice n° 3.10:

Exprimer la requête suivante avec un "IN":
Quels sont les noms des passagers ayant réservé pour le 1er avril 1996 ?

3) LA JOINTURE

2 principales méthodes

Prédicative

Requêtes imbriquées



Exemple: Quels sont les noms des pilotes qui assurent un vol le vendredi 13 octobre 1995 ?

- **Prédicative**
SELECT PILOTE.NOM
FROM PILOTE, DEPART
WHERE PILOTE.MATRICULE = DEPART.MATRICULE
AND DEPART.DATE_DEP = '13-10-95';

- **Imbrication**
SELECT NOM
FROM PILOTE
WHERE MATRICULE **IN** (**SELECT** DEPART.MATRICULE
FROM DEPART
WHERE DATE_DEP ='13-10-95');

Remarque

Il y a de nombreuses autres façons d'exprimer une jointure.

Par exemple:

```
SELECT PILOTE.NOM  
FROM PILOTE  
WHERE EXISTS ( SELECT *  
                FROM DEPART  
                WHERE  
                PILOTE.MATRICULE = DEPART.MATRICULE  
                AND DEPART.DATE_DEP = '13-10-95');
```

Exercice

Exercice n° 3.11:

Quels sont les noms des pilotes qui assurent un vol le vendredi 13 octobre 1995, sur un avion de capacité supérieure à 250 ?

- a) de manière prédicative,
- b) à l'aide de requêtes imbriquées.

Exercices

Exercice n° 3.12: Quelles sont les paires de pilotes (matricules) habitant dans la même ville ?

Exercice n° 3.13: Quels sont les noms des pilotes qui conduisent l'avion de numéro '666' ?

Exercice n°3.14: Quels sont les noms des pilotes qui habitent dans la ville où est entreposé l'avion '666' ?

4) L'UNION

Quels sont les pilotes qui conduisent l' avion de numéro '666' OU qui habitent dans la ville ou est entreposé l'avion '666' ?

```
SELECT NOM  
FROM PILOTE,DEPART  
WHERE PILOTE.MATRICULE = DEPART. MATRICULE  
AND DEPART.NUMAV = '666'
```

UNION

```
SELECT NOM  
FROM PILOTE,AVION  
WHERE PILOTE.VILLE = AVION.ENTREPOT  
AND AVION.NUMAV = '666';
```

5) L'INTERSECTION

Quels sont les pilotes qui conduisent l' avion de numéro '666' ET qui habitent dans la ville ou est entreposé l'avion '666' ?

```
SELECT NOM  
FROM PILOTE,DEPART  
WHERE PILOTE.MATRICULE = DEPART. MATRICULE  
AND DEPART.NUMAV = '666'
```

INTERSECT

```
SELECT NOM  
FROM PILOTE,AVION  
WHERE PILOTE.VILLE = AVION.ENTREPOT  
AND AVION.NUMAV = '666';
```

6) LA DIFFERENCE

Quels sont les pilotes qui conduisent l' avion de numéro '666' SAUF ceux qui habitent dans la ville ou est entreposé l'avion '666' ?

```
SELECT NOM  
FROM PILOTE,DEPART  
WHERE PILOTE.MATRICULE = DEPART. MATRICULE  
AND DEPART.NUMAV = '666'
```

MINUS

```
SELECT NOM  
FROM PILOTE,AVION  
WHERE PILOTE.VILLE = AVION.ENTREPOT  
AND AVION.NUMAV = '666';
```

7) LA DIVISION

Quels sont les pilotes qui conduisent tous les avions ?

<=> Quels sont les pilotes tels que, quel que soit l'avion, ils le conduisent ?

<=> Quels sont les pilotes tels que, quel que soit l'avion, il existe un départ assuré par ce pilote sur cet avion ?

<=> Quels sont les pilotes tels qu' il n'existe pas d'avion tel qu' il n'existe pas de départ de ce pilote sur cet avion ?

SELECT MATRICULE

FROM PILOTE

WHERE NOT EXISTS (**SELECT** *

FROM AVION

WHERE NOT EXISTS (**SELECT** *

FROM DEPART

WHERE DEPART.MATRICULE =

PILOTE.MATRICULE

AND DEPART.NUMAV = AVION.NUMAV));

Exercices

Exercice n° 3.15: Quels sont les passagers qui ont réservé sur tous les vols ?

Exercice n° 3.16: Quel est le nombre de pilotes qui conduisent des avions en service (i.e. dans la relation départ)?

GROUP BY

- L'opérateur GROUP BY partitionne la table

EXEMPLE: Quel est le nombre de pilotes qui conduisent des avions en service, par avion ?

```
SELECT NUMAV, count (DISTINCT DEPART.MATRICULE)  
FROM DEPART  
GROUP BY NUMAV;
```

Exercice

Exercice n° 3.17: Quelle est la capacité moyenne des avions, par entrepôt et par type ?

LE GROUP BY ... HAVING

- L'opérateur **HAVING** permet d'introduire une qualification sur les sous-tables issues du **GROUP BY**:

Le HAVING est au GROUP BY ce que le WHERE est au FROM.

EXEMPLE:
Quels sont les avions qui assurent plus de 3 départs ?

```
SELECT DEPART.NUMAV  
FROM DEPART  
GROUP BY NUMAV  
HAVING COUNT (*) > 3;
```


Exercice

Exercice n° 3.18: Quelles sont les villes où sont entreposés plus de 2 avions de type "ICBM" ?

Synthèse du SELECT : forme générale

```
SELECT [DISTINCT| ALL] { * | <value exp.> [, <value exp.>]...}  
FROM relation [variable], relation [ variable]...  
[WHERE <search condition>]  
[GROUP BY <attribute> [,<attribute>]...]  
[HAVING <search condition>]  
[ORDER BY <attribute> [{ASC | DESC}] [,<attribute>[{ASC | DESC}] ]...]
```

EXPRESSION DE VALEURS

- Calculs arithmétiques
- Fonctions agrégats

CONDITION DE RECHERCHE

- Sélection, projection, jointure
- Recherche textuelle
- Recherche par intervalle
- Recherche sur valeur nulle

Forme générale de la condition

<search condition> ::= [NOT]

<nom_colonne> θ constante | <nom_colonne>

<nom_colonne> LIKE <modèle_de_chaîne>

<nom_colonne> IN <liste_de_valeurs>

<nom_colonne> θ (ALL | ANY | SOME) <liste_de_valeurs>

EXISTS <liste_de_valeurs>

UNIQUE <liste_de_valeurs>

<tuple> MATCH [UNIQUE] <liste_de_tuples>

<nom_colonne> BETWEEN constante AND constante

<search condition> AND | OR <search condition>

avec

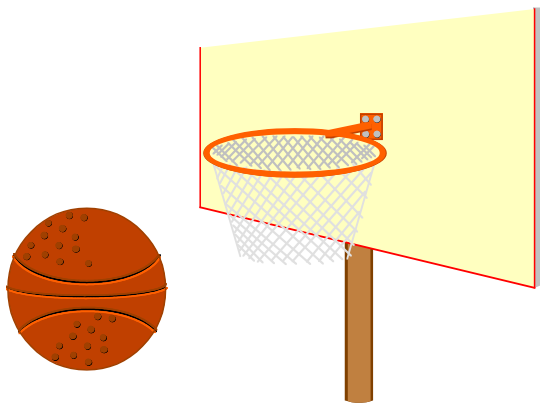
$\theta ::= < \mid = \mid > \mid \geq \mid \leq \mid <>$

Remarque: <liste_de_valeurs> peut être dynamiquement déterminée par une requête

CONCLUSION

- SQL est un langage d'interrogation très lisible, par tous les utilisateurs
- SQL est un langage dans lequel il est relativement facile d'écrire, pour un programmeur, à condition de bien visualiser l'ordre d'application des instructions:

(1)	From
(2)	Where
(3)	Group by
(4)	Having
(5)	Select
(6)	Order by



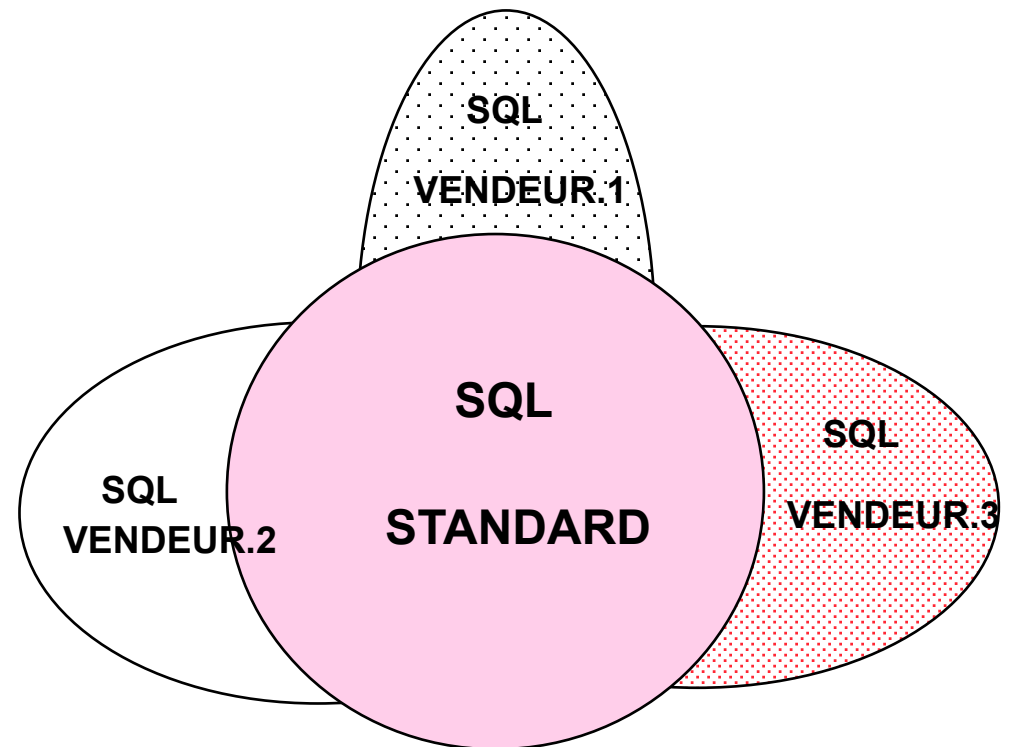
LA NORMALISATION DE SQL

- **Groupe de travail ANSI/X3/H2 et ISO/IEC JTC1/SC2**
- **Documents ISO :**
 - **SQL1 - 86 : Database Language SQL X3.135 ISO-9075-1987)**
 - **SQL1 - 89 : Database Language SQL with Integrity Enhancement X3.168 ISO-9075-1989**
 - **SQL2 - 92 : Database Language SQL2 X3.135 ISO-9075-1992**
- **Arguments pour :**
 - **Réduction des coûts d'apprentissage**
 - **Portabilité des applications**
 - **Longévité des applications**
 - **Langage de communication inter-systèmes**
- **Arguments contre :**
 - **Manque de rigueur théorique**
 - **Affaiblit la créativité**

POSITION DES VENDEURS

➤ Problèmes :

- SQLCODE (0 ou <0 si erreur)
- Requêtes imbriquées
- Dynamique SQL (Prepare, Execute)
- Méta-base normalisée
- Modèles internes (Index, Espaces,...)



CORRECTION DES EXERCICES

Correction de l'exercice n° 3.1

Exercice n° 3.1: Quel est l'âge du pilote de matricule 15 ?

```
SELECT PILOTE.AGE  
FROM PILOTE  
WHERE MATRICULE = 15;
```


Correction de l'exercice n° 3.2

Exercice n° 3.2: Quels sont les pilotes de cette compagnie aérienne (leur matricule, leur nom, leur ville, leur âge et leur salaire).

```
SELECT *  
FROM PILOTE;
```

Correction de l'exercices n°3.3

Exercice n° 3.3: Quels sont les types d'avions entreposés à Tarascon, avec leur capacité, par ordre de capacités croissantes ?

```
SELECT distinct TYPE, CAPACITE  
FROM AVION  
WHERE ENTREPOT = 'Tarascon'  
ORDER BY CAPACITE ASC;
```

Correction de l'exercice N° 3.4

Exercice n° 3.4: Quelles sont les villes ayant un nom commençant par 'L' et où un pilote de moins de 20 ans gagne plus de 200000F ?

```
SELECT VILLE  
FROM PILOTE  
WHERE AGE < 20  
AND SALAIRE > 200000  
AND VILLE LIKE 'L%';
```

Correction de l'exercice n° 3.5

Exercice n° 3.5: Quels sont les numéros d'avions, avec leur entrepôt, ne résidant pas à Marolles-en-Hurepoix, ayant une capacité supérieure à 200 ?
(donnés par ordre de numéros décroissants).

```
SELECT NUMAV, ENTREPOT  
FROM AVION  
WHERE ENTREPOT != 'Marolles-en-Hurepoix'  
AND CAPACITE > 200  
ORDER BY NUMAV DESC;
```

Correction de l'exercice 3.6

Exercice n° 3.6: Quelles sont les villes d'où part un vol pour Ajaccio?
(donner les résultats en majuscule).

```
SELECT UPPER (VILLE_DEPART)  
FROM VOL  
WHERE VILLE_ARRIVEE = 'Ajaccio';
```

Correction de l'exercice n°3.7

Exercice n° 3.7: Quel est le nombre d'avions en services (i.e. dans la relation DEPART), le 19 décembre 1995 ?

```
SELECT COUNT (DISTINCT NUMAV)
FROM DEPART
WHERE DATE_DEP = '19-12-95';
```

Correction de l'exercice 3.8

Exercice n° 3.8 : Quelle est la capacité moyenne des avions entreposés à Pogo Togo ?

```
SELECT AVG (ALL CAPACITE)  
FROM AVION  
WHERE AVION.ENTREPOT = 'Pogo Togo';
```

Correction de l'exercice 3.9

Exercice n°3.9: Exprimer la requête suivante avec un "EXISTS":

Quels sont les noms des pilotes qui habitent dans une ville où sont localisés des avions de capacité supérieure à 250 ?

```
SELECT PILOTE.NOM  
FROM PILOTE  
WHERE EXISTS ( SELECT *  
                FROM AVION  
                WHERE CAPACITE > 250  
                AND AVION .ENTREPOT = PILOTE.VILLE);
```


Correction de l'exercice n° 3.10

Exercice n° 3.10: Quels sont les noms des passagers ayant réservé pour le 1er avril 1996 ?
(avec un "IN").

```
SELECT PASSAGER.NOMAB
FROM PASSAGER
WHERE NUMAB IN (SELECT RESERVATION.NUMAB
                FROM RESERVATION
                WHERE RESERVATION.DATE_DEP = '01-04-96');
```

Correction de l'exercice n° 3.11

Exercice n° 3.11:

Quels sont les noms des pilotes qui assurent un vol le vendredi 13 octobre 1995, sur un avion de capacité supérieure à 250 ?

a) de manière prédicative,

```
SELECT PILOTE.NOM  
FROM PILOTE, DEPART, AVION  
WHERE PILOTE.MATRICULE = DEPART.MATRICULE  
AND DEPART.DATE_DEP = '13-10-95'  
AND DEPART.NUMAV = AVION.NUMAV  
AND AVION.CAPACITE > 250;
```

Correction de l'exercice n°3.11 (suite)

Exercice n° 3.11:

Quels sont les noms des pilotes qui assurent un vol le vendredi 13 octobre 1995, sur un avion de capacité supérieure à 250 ?

b) à l'aide de requêtes imbriquées

```
SELECT NOM
FROM PILOTE
WHERE MATRICULE IN ( SELECT DEPART.MATRICULE
                      FROM DEPART
                      WHERE DATE_DEP ='13-10-95'
                      AND NUMAV IN ( SELECT AVION.NUMAV
                                     FROM AVION
                                     WHERE CAPACITE > 250));
```

Correction de l'exercice n° 3.12

Exercice n° 3.12: Quelles sont les paires de pilotes (matricules) habitant dans la même ville ?

➡ Il s'agit d'une autojointure

```
SELECT P1.MATRICULE, P2.MATRICULE  
FROM PILOTE P1, PILOTE P2  
WHERE P1.VILLE = P2.VILLE  
AND P1.MATRICULE > P2.MATRICULE;
```

Correction de l'exercice n° 3.13

Exercice n° 3.13: Quels sont les noms des pilotes qui conduisent l' avion de numéro '666' ?

```
SELECT NOM  
FROM PILOTE,DEPART  
WHERE PILOTE.MATRICULE = DEPART. MATRICULE  
AND DEPART.NUMAV = '666';
```

Correction de l'exercice n°3.14

Exercice n°3.14: Quels sont les noms des pilotes qui habitent dans la ville où est entreposé l'avion '666' ?

```
SELECT NOM  
FROM PILOTE,AVION  
WHERE PILOTE.VILLE = AVION.ENTREPOT  
AND AVION.NUMAV = '666';
```

Correction de l'exercice n° 3.15

Exercice n° 3.15: Quels sont les passagers qui ont réservé sur tous les vols ?

```
SELECT NUMAB
FROM PASSAGER
WHERE NOT EXISTS (SELECT *
                  FROM VOL
                  WHERE NOT EXISTS (SELECT *
                                    FROM RESERVATION
                                    WHERE RESERVATION.NUMAB =
                                           PASSAGER.NUMAB
                                           AND RESERVATION.Numvol =
                                           VOL.NUMVOL));
```

Correction de l'exercice n° 3.16

Exercice n° 3.16: Quel est le nombre de pilotes qui conduisent des avions en service (i.e. dans la relation départ)?

```
SELECT count (DISTINCT DEPART.MATRICULE)  
FROM DEPART;
```


Correction de l'exercice n°3.17

Exercice n° 3.17: Quelle est la capacité moyenne des avion, par entrepôt et par type ?

```
SELECT ENTREPOT, TYPE, AVG (CAPACITE)  
FROM AVION  
GROUP BY ENTREPOT,TYPE;
```

Correction de l'exercice n° 3.18

Exercice n° 3.18: Quelles sont les villes où sont entreposés plus de 2 avions de type "ICBM" ?

```
SELECT ENTREPOT  
FROM AVION  
WHERE TYPE = 'ICBM'  
GROUP BY ENTREPOT  
HAVING COUNT (*) > 2;
```