

LE
LANGUAGE
S.QL

TABLE DES MATIERES

1. INTRODUCTION.....	2
1.1. HISTORIQUE.....	3
1.2. DÉFINITION.....	3
2. ALGÈBRE RELATIONNELLE.....	5
2.1. OPÉRATEURS ENSEMBLISTES.....	5
2.1.1. <i>Union</i>	5
2.1.2. <i>Intersection</i>	6
2.1.3. <i>Différence</i>	7
2.2. OPÉRATEURS UNAIRES RELATIONNELS.....	8
2.2.1. <i>Sélection</i>	8
2.2.2. <i>Projection</i>	8
2.3. OPÉRATEURS BINAIRES RELATIONNELS.....	8
2.3.1. <i>Le produit cartésien</i>	8
2.3.2. <i>La jointure</i>	8
2.3.3. <i>La division</i>	9
3. LE LANGAGE SQL.....	10
3.1. LANGAGE DE DESCRIPTION DE DONNÉES.....	10
3.1.1. <i>CREATE</i>	10
3.1.2. <i>DROP</i>	10
3.1.3. <i>ALTER</i>	11
3.2. LANGAGE DE MANIPULATION DES DONNÉES.....	11
3.2.1. <i>SELECT</i>	11
3.2.2. <i>INSERT</i>	12
3.2.3. <i>DELETE</i>	12
3.2.4. <i>UPDATE</i>	12
4. LA SÉLECTION.....	13
4.1. SYNTAXE COMPLÈTE DU SELECT.....	13
4.2. FONCTIONS INTÉGRÉES.....	14
4.3. LA JOINTURE.....	15
4.3.1. <i>Méthode ensembliste</i>	15
4.3.2. <i>Méthode prédicative</i>	16
4.3.3. <i>Auto-jointure</i>	16
4.4. OPÉRATEUR DE PARTITIONNEMENT.....	17
4.4.1. <i>Group by</i>	17
4.4.2. <i>Having</i>	17
4.5. OPÉRATEURS DU WHERE.....	18
5. EXERCICES.....	19
5.1. ENONCÉS.....	19
5.2. CORRIGÉS.....	20
5.2.1. <i>Exercice 1</i>	20
5.2.2. <i>Exercice 2</i>	20
5.2.3. <i>Exercice 3</i>	21
5.2.4. <i>Exercice 4</i>	21
5.2.5. <i>Exercice 5</i>	22
5.2.6. <i>Exercice 6</i>	22
5.2.7. <i>Exercice 7</i>	22
5.2.8. <i>Exercice 8</i>	22
5.2.9. <i>Exercice 9</i>	23
5.2.10. <i>Exercice 10</i>	23
5.2.11. <i>Exercice 11</i>	23
5.2.12. <i>Exercice 12</i>	23
5.2.13. <i>Exercice 13</i>	24
5.2.14. <i>Exercice 14</i>	24

1. INTRODUCTION

1.1. Historique

S.Q.L. est un langage structuré permettant d'interroger et de modifier les données contenues dans une base de données relationnelle.

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 imposé comme norme. Il n'existe pas de S.G.B.D.R sans S.Q.L..

Malheureusement, malgré la norme S.Q.L., il existe un ensemble de dialectes. Les différences entre ces différents dialectes sont souvent minimes et tous respectent un minimum commun : ce que nous allons étudier ici.

1.2. Définition

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.

S.Q.L. est un langage interactif, mais il peut aussi être intégré dans un langage de programmation pour le développement d'applications.

S.Q.L. n'est pas le meilleur langage, en particulier pour la manipulation des données, mais c'est un standard.

Dans tout ce qui suit les exemples seront donnés par rapport à la base de données suivante :

AVION(NUMAV, AVMARQ, AVTYPE, CAP, LOC)

PILOTE(NUMPIL, PILNOM, ADR)

VOL(NUMVOL, NUMPIL, NUMAV, VD, VA, HD, HA)

NUMAV : numéro d'avion

AVMARQ : marque de l'avion

AVTYPE : type de l'avion

CAP : capacité en nb de passagers

LOC : ville où est basé l'avion

NUMPIL : numéro du pilote

PILNOM : nom du pilote

ADR : adresse du pilote

NUMVOL : numéro du vol

VD : ville départ

VA : ville d'arrivée

HD : heure de départ

HA : heure d'arrivée

TABLE AVION

NUMAV	Marque	Type	Capacité	Localisation
100	AIRBUS	A320	300	Nice
101	BOIENG	B707	250	Paris
102	AIRBUS	A320	300	Toulouse
103	CARAVELLE	Caravelle	200	Toulouse
104	BOEING	B747	400	Paris
105	AIRBUS	A320	300	Grenoble
106	ATR	ATR42	50	Paris
107	BOEING	B727	300	Lyon
108	BOEING	B727	300	Nantes
109	AIRBUS	A340	350	Bastia

TABLE PILOTE

NUMPIL	Nom	Adresse
1	SERGE	Nice
2	JEAN	Paris
3	CLAUDE	Grenoble
4	ROBERT	Nantes
5	MICHEL	Paris
6	LUCIEN	Toulouse
7	BERTRAND	Lyon
8	HERVE	Bastia
9	LUC	Paris

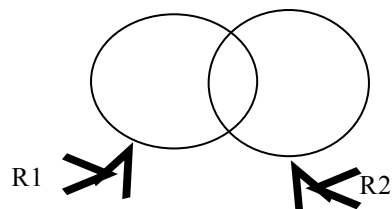
TABLE VOL

NUMVOL	Avion	Pilote	Ville Départ	Ville Arrivée	Heure Départ	Heure Arrivée
IT100	100	1	NICE	PARIS	7	9
IT101	100	2	PARIS	TOULOUSE	11	12
IT102	101	1	PARIS	NICE	12	14
IT103	105	3	GRENOBLE	TOULOUSE	9	11
IT104	105	3	TOULOUSE	GRENOBLE	17	19
IT105	107	7	LYON	PARIS	6	7
IT106	109	8	BASTIA	PARIS	10	13
IT107	106	9	PARIS	LYON	7	8
IT108	106	9	LYON	PARIS	19	20
IT109	107	7	PARIS	LYON	18	19
IT110	102	2	TOULOUSE	PARIS	15	16
IT111	101	4	NICE	NANTES	17	19

2. ALGÈBRE RELATIONNELLE

2.1. Opérateurs ensemblistes

2.1.1. Union



R1 U R2

Est utilisé pour relier 2 relations unicompatibles, c'est à dire ayant le même nombre d'attributs et des attributs respectifs définis sur le même domaine.

Pour l'union il faut une structure identique.

R1 : TABLE AVION PARIS NICE

VOL	Avion	Pilote	Ville Départ	Ville Arrivée	Heure Départ	Heure Arrivée
IT102	101	1	Paris	Nice	12h00	14h00
IT118	256	1024	Paris	Nice	9h36	10h30
IT112	812	2048	Paris	Nice	12h25	14h00
IT123	999	1246	Paris	Nice	19h00	20h00

R2 : TABLE AVION LYON MARSEILLE

VOL	Avion	Pilote	Ville Départ	Ville Arrivée	Heure Départ	Heure Arrivée
IT99	999	2	Lyon	Marseille	10h30	13h00

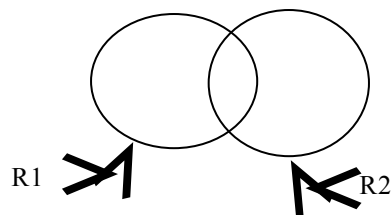
R1 U R2

Avion
101
256
812
999

Exemples :

- liste des vols Paris-Nice et Lyon-Marseille
- liste des avions Airbus et Boeing
- liste des vols sur Airbus ou sur un avion de plus de 200 places

2.1.2. Intersection



$R1 \cap R2$

Est utilisé pour relier 2 relations unicompatibles, c'est à dire ayant le même nombre d'attributs et des attributs respectifs définis sur le même domaine.

Pour l'intersection il faut une structure identique.

R1 : TABLE AVION AIRBUS

NUMAV	AVMARQ	Ville Départ	Ville Arrivée
100	AIRBUS	NICE	PARIS
100	AIRBUS	PARIS	TOULOUSE
105	AIRBUS	GRENOBLE	TOULOUSE
105	AIRBUS	TOULOUSE	GRENOBLE
109	AIRBUS	BASTIA	PARIS
102	AIRBUS	TOULOUSE	PARIS

R2 : TABLE AVION PARIS

NUMAV	AVMARQ	Ville Départ	Ville Arrivée
100	AIRBUS	NICE	PARIS
107	BOEING	LYON	PARIS
109	AIRBUS	BASTIA	PARIS
106	ATR	LYON	PARIS
102	AIRBUS	TOULOUSE	PARIS

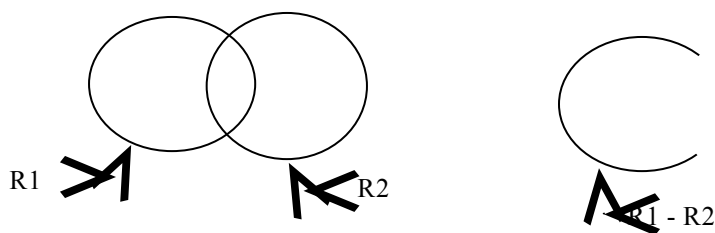
$R1 \cap R2$

NUMAV	AVMARQ	Ville Départ	Ville Arrivée
100	AIRBUS	NICE	PARIS
109	AIRBUS	BASTIA	PARIS
102	AIRBUS	TOULOUSE	PARIS

Exemples :

- Liste des avions AIRBUS allant à Paris
- liste des pilotes de boeing habitant toulouse

2.1.3. Différence



R1-R2

Est utilisé pour relier 2 relations unicompatibles, c'est à dire ayant le même nombre d'attributs et des attributs respectifs définis sur le même domaine.

R1 : TABLE AVION AIRBUS

NUMAV	AVMARQ	Ville Départ	Ville Arrivée
100	AIRBUS	NICE	PARIS
100	AIRBUS	PARIS	TOULOUSE
105	AIRBUS	GRENOBLE	TOULOUSE
105	AIRBUS	TOULOUSE	GRENOBLE
109	AIRBUS	BASTIA	PARIS
102	AIRBUS	TOULOUSE	PARIS

R2 : TABLE AVION PARIS

NUMAV	AVMARQ	Ville Départ	Ville Arrivée
100	AIRBUS	NICE	PARIS
107	BOEING	LYON	PARIS
109	AIRBUS	BASTIA	PARIS
106	ATR	LYON	PARIS
102	AIRBUS	TOULOUSE	PARIS

R2 - R1

NUMAV	AVMARQ	Ville Départ	Ville Arrivée
107	BOEING	LYON	PARIS
106	ATR	LYON	PARIS

R1 - R2

NUMAV	AVMARQ	Ville Départ	Ville Arrivée
100	AIRBUS	PARIS	TOULOUSE
105	AIRBUS	GRENOBLE	TOULOUSE
105	AIRBUS	TOULOUSE	GRENOBLE

Exemples :

- liste des Airbus qui ne vont pas à Paris
- liste des avions pour Paris qui ne sont pas des Airbus
- liste des pilotes qui ne conduisent pas d'ATR

2.2. Opérateurs unaires relationnels

Les opérateurs unaires sont les opérateurs qui ne font intervenir qu'une seule relation.

2.2.1. Sélection

Elle opère un découpage horizontal de la table.

Certains enregistrements et tous les attributs.

Exemples :

- liste des avions avec toutes les informations
- liste des vols pour Nice
- liste des avions qui ont plus de 200 places

2.2.2. Projection

Elle opère un découpage vertical de la table.

Certains attributs et tous les enregistrements.

Exemples :

- liste de tous les numéros d'avions
- liste des noms des pilotes
- liste des marques d'avions

2.3. Opérateurs binaires relationnels

Les opérateurs binaires sont ceux qui font intervenir deux relations.

2.3.1. Le produit cartésien

Il consiste à rassembler deux tables en une seule

R1	R2	R1 x R2	
A1	B1	A1	B1
A2	B2	A2	B1
A3		A3	B1
		A1	B2
		A2	B2
		A3	B2

2.3.2. La jointure

La **jointure** consiste à rechercher entre deux tables ayant un attribut commun (même type et même domaine de définition) tous les tuples pour lesquels ces attributs ont la même valeur.

La jointure entre les tables AVION et PILOTE consiste à rechercher toutes les informations du pilote de chaque avion. Concrètement elle réalise la concaténation des tuples (ou n-uplets) de Avion avec Pilote chaque fois que AVION.NUMPIL = VOL.NUMPIL.

Vol

Vol	Av	Pil	VD	VA	HD	HA
IT101	100	2	PARIS	TOULOUSE	11	12
IT100	100	1	NICE	PARIS	7	9
IT102	101	1	PARIS	NICE	12	14
IT103	105	3	GRENOBLE	TOULOUSE	9	11
IT104	105	3	TOULOUSE	GRENOBLE	17	19
IT105	107	7	LYON	PARIS	6	7
IT106	109	8	BASTIA	PARIS	10	13

Pilote

Pil	Nom
1	SERGE
2	JEAN
3	CLAUDE
7	BERTRAND
8	HERVE

Vol	Av	Pil	VD	VA	HD	HA	Nom
IT101	100	2	PARIS	TOULOUSE	11	12	JEAN
IT100	100	1	NICE	PARIS	7	9	SERGE
IT102	101	1	PARIS	NICE	12	14	SERGE
IT103	105	3	GRENOBLE	TOULOUSE	9	11	CLAUDE
IT104	105	3	TOULOUSE	GRENOBLE	17	19	CLAUDE
IT105	107	7	LYON	PARIS	6	7	BERTRAND
IT106	109	8	BASTIA	PARIS	10	13	HERVE

2.3.3. La division

R1

B	A
B1	A1
B1	A2
B2	A2
B3	A1
B4	A1
B4	A2

R2

C
A1
A2

R3

D
B1
B4

La division consiste à trouver la liste des valeurs d'un attribut de R1 telles qu'il existe un tuple dans R1 pour chaque valeur de R2 pour l'attribut correspondant.

exemples :

- liste des pilotes qui conduisent tous les avions
- liste des avions qui vont dans toutes les villes

3. LE LANGAGE SQL

3.1. Langage de Description de Données

3.1.1. CREATE

3.1.1.1. CREATE TABLE

Création de la description d'une table avec la liste de tous ses attributs et leur type.

```
CREATE TABLE PILOTE (NUMPIL INTEGER NOT NULL,  
                      PILNOM CHAR(12),  
                      ADR CHAR(20));
```

```
CREATE TABLE AVION  (NUMAV INTEGER NOT NULL,  
                      AVMARQ CHAR(12),  
                      AVNOM CHAR(12),  
                      CAP INTEGER,  
                      LOC CHAR(20),  
                      UNIQUE (NUMAV));
```

```
CREATE TABLE VOL    (NUMVOL INTEGER NOT NULL,  
                      NUMPIL INTEGER NOT NULL,  
                      NUMAV INTEGER NOT NULL,  
                      VD CHAR(20),  
                      VA CHAR(20),  
                      HD DECIMAL(2,2),  
                      HA DECIMAL(2,2),  
                      UNIQUE (NUMVOL));
```

3.1.1.2. CREATE VIEW

```
CREATE VIEW VOL_NICE (NUMVOL, NUMPIL, NUMAV, VD, VA, HD, HA)  
AS SELECT (NUMVOL, NUMPIL, NUMAV, VD, VA, HD, HA)  
FROM VOL  
WHERE VD=« NICE »
```

Une vue est utilisée pour :

- obtenir une table intermédiaire constituant un extrait d'une ou plusieurs tables
- définir les règles de gestion permettant d'assurer le contrôle d'intégrité dans la base de données.

3.1.1.3. CREATE INDEX

```
CREATE INDEX VILLE_D ON VOL (VD);  
CREATE UNIQUE INDEX CLE ON PILOTE (NUMPIL);
```

La création d'index sert à améliorer les performances lors de recherche dans la table sur cet attribut (VD dans VOL ou NUMPIL dans PILOTE)

3.1.2. DROP

DROP est utilisé pour supprimer une définition de table, de vue ou d'index.

```
DROP TABLE VOL;  
DROP INDEX CLE;
```

3.1.3. ALTER

ALTER est utilisé pour modifier une définition de table, de vue ou d'index.

```
ALTER TABLE PILOTE ADD SALAIRE SMALLINT;  
ALTER TABLE PILOTE SUBTRACT SALAIRE;
```

3.2. Langage de Manipulation des Données

3.2.1. SELECT

Le SELECT sert à interroger les données et à les présenter triées et/ou regroupées suivant certains critères.

3.2.1.1. Sélection

Il s'agit de retrouver les enregistrements qui vérifient certains critères

Exemple : trouver les vols qui arrivent à 19 heures

```
SELECT *  
FROM VOL  
WHERE HA >= 19h00;
```

3.2.1.2. Projection

Il s'agit de n'afficher que certains attributs dans une table

Exemple : lister les numéros de vols

```
SELECT NUMVOL  
FROM VOL;
```

3.2.1.3. Opérateurs ensemblistes

On peut travailler en SQL avec la méthode ensembliste. pour cela, il faut considérer que chaque table est un **ensemble** de tuples et que chaque SELECT produit un ensemble de tuples.

On utilise ensuite les opérateurs classiques sur les ensembles :

- UNION
- IN
- NOT IN

Exemples :

- Liste des avions AIRBUS allant à Paris (INTERSECTION)
(SELECT NUMAV FROM AVION
WHERE AVMARQ = AIRBUS)
INTERSECT
(SELECT NUMAV FROM VOL
WHERE VA = PARIS)

qui peut aussi s'écrire

```
SELECT * FROM AVION  
WHERE AVMARQ = AIRBUS  
AND NUMAV IN  
(SELECT NUMAV FROM VOL  
WHERE VA = Paris)
```

C'est à dire trouver tous les avions dont le numéro appartient à l'ensemble des numéros d'avions des vols à destination de Paris.

- Liste des avions AIRBUS n'allant pas à Paris (DIFFERENCE)

```
(SELECT NUMAV FROM AVION
WHERE AVMARQ = AIRBUS)
```

MINUS

```
(SELECT NUMAV FROM VOL
WHERE VA = PARIS)
```

qui peut aussi s'écrire

```
SELECT * FROM AVION
WHERE AVMARQ = AIRBUS
AND NUMAV NOT IN
(SELECT NUMAV FROM VOL
WHERE VA = Paris)
```

C'est à dire trouver tous les avions dont le numéro n'appartient pas à l'ensemble des numéros d'avions des vols à destination de Paris.

- Liste des avions de marque AIRBUS ou de plus de 200 places (UNION)

```
(SELECT NUMAV FROM AVION
WHERE AVMARQ = AIRBUS)
```

UNION

```
(SELECT NUMAV FROM AVION
WHERE CAP > 200)
```

qui peut aussi s'écrire

```
SELECT * FROM AVION
WHERE AVMARQ = AIRBUS
OR CAP > 200
```

3.2.2. INSERT

Permet d'ajouter un enregistrement dans une table.

```
INSERT
INTO AVION
VALUES (110, « AIRBUS », « A320 », 320, « NICE »);
```

3.2.3. DELETE

Permet de supprimer un enregistrement d'une table.

```
DELETE
FROM AVION
WHERE CAP < 100;
```

3.2.4. UPDATE

Permet de modifier les valeurs de certains attributs d'un ou plusieurs enregistrements dans une table.

Exemple : modifier la capacité de tous les avions basés à Nice

```
UPDATE AVION
SET CAP=CAP*1,1
WHERE LOC=« NICE »;
```

4. LA SÉLECTION

4.1. Syntaxe complète du SELECT

SELECT Liste des attributs à afficher
[DISTINCT, *] mots clés facultatifs
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

UNION | INTERSECT | MINUS

autre SELECT

4.1.1.1. 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 de ces enregistrements.

Exemple : liste des types d'avions de plus de 200 places.

```
SELECT AVTYPE, CAP
FROM AVION
WHERE CAP > 250;
```

les types A320 et B727 vont apparaitre plusieurs fois

```
SELECT DISTINCT (AVTYPE, CAP)
FROM AVION
WHERE CAP > 250;
```

on obtient :

A320	300
B707	250
B747	400
B727	300
A340	350

4.2. Fonctions intégrées

Des fonctions intégrées peuvent être combinées à la liste des attributs.

SUM : Somme des valeurs de l'attribut pour les enregistrements sélectionnés
MIN : Minimum des valeurs de l'attribut pour les enregistrements sélectionnés
MAX : Maximum des valeurs de l'attribut pour les enregistrements sélectionnés
AVG : Moyenne des valeurs de l'attribut pour les enregistrements sélectionnés
COUNT : Nombre d'enregistrements sélectionnés

- Nombre d'avions dans la table

```
SELECT COUNT(NUMAV)
FROM AVION;
```

- Nombre d'avions en service

```
SELECT COUNT(DISTINCT NUMAV)
FROM VOL;
```

- Avion de plus petite capacité

```
SELECT AVTYPE
FROM AVION
WHERE CAP=MIN(CAP);
```

- Capacités mini et maxi des boeings

```
SELECT MIN(CAP), MAX(CAP)
FROM AVION
WHERE AVMARQ=« BOEING »;
```

- Capacité moyenne des avions localisés à Paris

```
SELECT AVG(CAP)
FROM AVION
WHERE LOC=« PARIS »;
```

- Capacité totale des avions

```
SELECT SUM(CAP)
FROM AVION
```

4.3.2. Méthode prédicative

Il y a un seul SELECT pour toute la requête.

La liste de toutes les tables concernées apparaît dans le FROM

La traduction de la jointure se fait par l'équation de jointure (égalité entre 2 attributs)

Exemples :

- Type et capacité des avions en service

```
SELECT AVION.NUMAV, AVMARQ,CAP
FROM VOL,AVION
WHERE VOL.NUMAV = AVION.NUMAV ;
```

- Nom des pilotes en service

```
SELECT DISTINCT PILNOM
FROM VOL, PILOTE
WHERE VOL.NUMPIL=PILOTE.NUMPIL;
```

- Nom des pilotes assurant un vol au départ de Paris

```
SELECT PILNOM
FROM VOL, PILOTE
WHERE VOL.NUMPIL= PILOTE.NUMPIL
AND VD=« Paris »
```

- Nom des pilotes conduisant un Airbus

```
SELECT PILNOM
FROM VOL, PILOTE, AVION
WHERE VOL.NUMPIL= PILOTE.NUMPIL
AND VOL.NUMAV=AVION.NUMAV
AND AVMARQ=« Airbus »;
```

4.3.3. 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.

Exemple :

- Nom des avions ayant même capacité

```
SELECT AVION1.AVTYPE, AVION1.CAP,
      « Même capacité que : », AVION2.AVTYPE
FROM AVION1, AVION2
WHERE AVION1.NUMAV<>AVION2.NUMAV
AND AVION1.CAP = AVION2.CAP;
```

4.4. Opérateur de partitionnement

4.4.1. Group by

Ce mot clé permet d'effectuer des regroupements sur lesquels s'opèrent les fonctions intégrées.

Exemples :

- Nombre d'avions de chaque marque

```
SELECT AVMARQ, COUNT(NUMAV)
FROM AVION
GROUP BY AVMARQ;
```

- Nombre de pilotes différents pour chaque avion en service

```
SELECT NUMAV, COUNT(DISTINCT NUMPIL)
FROM VOL
GROUP BY NUMAV;
```

4.4.2. Having

Le mot clé **HAVING** associé au GROUP BY permet d'exprimer des conditions sur les regroupements :

- Numéros des pilotes assurant plus d'un vol

```
SELECT NUMPIL
FROM VOL
GROUP BY NUMPIL
HAVING COUNT(NUMVOL)>1
ORDER BY NUMPIL
```

- Numéros des pilotes et Nombre de vols assurés au départ de Nice

```
SELECT NUMPIL, COUNT(NUMVOL)
FROM VOL
WHERE VD=« NICE »
GROUP BY NUMPIL
HAVING COUNT(NUMVOL)>1
```

- Numéros des pilotes et nombre de vols qui ont plusieurs vols dont un au moins au départ de Nice

```
SELECT NUMPIL, COUNT(NUMVOL)
FROM VOL
WHERE NUMPIL IN (SELECT NUMPIL
                  FROM VOL
                  WHERE VD=« NICE »)
GROUP BY NUMPIL
HAVING COUNT(NUMVOL)>1;
```

4.5. Opérateurs du WHERE

Pour exprimer les conditions dans la clause WHERE on dispose de certains opérateurs :

- >, <, =, <>, <=, >= pour les comparaisons
- BETWEEN
- IN et NOT IN : expriment l'appartenance (ou non) d'un tuple à l'ensemble résultat du SELECT imbriqué
- LIKE suivi d'une expression représentant un ensemble de valeurs. Dans ces expressions % désigne un ensemble de caractères, _ remplace une lettre
- EXISTS et NOT EXISTS : expriment l'appartenance (ou non) d'un tuple à l'ensemble résultat du SELECT imbriqué
- IS NULL et IS NOT NULL : testent si un attribut possède ou non une valeur

Exemples :

- Nom des pilotes dont la 2^e lettre est un S

```
SELECT PILNOM
FROM PILOTE
WHERE PILNOM LIKE « _S% »
```

- Nom des pilotes en service

```
SELECT PILNOM
FROM VOL, PILOTE
WHERE PILOTE.NUMPIL=VOL.NUMPIL
```

OU

```
SELECT PILNOM
FROM PILOTE
WHERE NUMPIL EXISTS (SELECT NUMPIL
                     FROM VOL);
```

- Nom des pilotes n'ayant aucun vol

```
SELECT PILNOM
FROM PILOTE
WHERE NUMPIL NOT EXISTS (SELECT NUMPIL
                        FROM VOL);
```

5. EXERCICES**5.1. Enoncés**

- 1 Quels sont les noms des pilotes qui habitent dans la ville de localisation d'un Airbus ?
 - 2 Quels sont les noms des pilotes qui conduisent un Airbus et qui habitent dans la ville de localisation d'un Airbus ?
 - 3 Quels sont les noms des pilotes qui conduisent un Airbus ou qui habitent dans la ville de localisation d'un Airbus ?
 - 4 Quels sont les noms des pilotes qui conduisent un Airbus sauf ceux qui habitent dans la ville de localisation d'un Airbus ?
 - 5 Quels sont les vols ayant un trajet identique (VD, VA) à ceux assurés par Serge ?
 - 6 Pour chaque ville, donner la capacité moyenne des avions qui s'y trouvent et cela par type d'avion.
 - 7 Donner toutes les paires de pilotes habitant la même ville (sans doublon).
 - 8 Pour chaque ville, donner le nombre et les capacités minimum et maximum des avions qui s'y trouvent?
 - 9 Quelle est la capacité moyenne des avions pour chaque ville ayant plus de 10 avions ?
 - 10 Quels sont les vols au départ de Nice entre 12h et 14h ?
 - 11 Quels sont les pilotes dont le nom commence par "S" ?
 - 12 Quels sont les noms des pilotes qui conduisent un avion que conduit aussi le pilote n°10 ?
 - 13 Donner toutes les paires de villes telles qu'un avion localisé dans la ville de départ soit conduit par un pilote résidant dans la ville d'arrivée.
 - 14 Sélectionner les numéros des pilotes qui conduisent tous les Airbus ?
-

5.2.5. Exercice 5

- Quels sont les vols ayant un trajet identique (VD, VA) à ceux assurés par Serge ?

Prédicative

```
SELECT NUMVOL
FROM VOL1, VOL, PILOTE
WHERE VOL1.VD = VOL.VD
AND VOL1.VA = VOL.VA
AND VOL1.NUMPIL=PILOTE.NUMPIL
AND PILNOM=« Serge »;
```

Ensembliste

```
SELECT NUMVOL
FROM VOL
WHERE (VD, VA) IN (SELECT (VD, VA)
                   FROM VOL
                   WHERE NUMPIL IN (SELECT NUMPIL
                                   FROM PILOTE
                                   WHERE PILNOM = « Serge »));
```

5.2.6. Exercice 6

- Pour chaque ville, donner la capacité moyenne des avions qui s'y trouvent et cela par type d'avion.

Prédicative

```
SELECT LOC, AVNOM, AVG(CAP)
FROM AVION
GROUP BY LOC, AVNOM
```

5.2.7. Exercice 7

- Donner toutes les paires de pilotes habitant la même ville (sans doublon).

Prédicative

```
SELECT PILOTE.PILNOM, PILOTE1.PILNOM
FROM PILOTE, PILOTE1
WHERE PILOTE1.ADR=PILOTE.ADR
AND PILOTE1.NUMPIL>PILOTE.NUMPIL
```

5.2.8. Exercice 8

- Pour chaque ville, donner le nombre et les capacités minimum et maximum des avions qui s'y trouvent?

Prédicative

```
SELECT LOC, COUNT( NUMAV), MIN(CAP), MAX(CAP)
FROM AVION
GROUP BY LOC
```

5.2.9. Exercise 9

- Quelle est la capacité moyenne des avions pour chaque ville ayant plus de 10 avions ?

Prédicative

```
SELECT LOC, AVG(CAP)
FROM AVION
GROUP BY LOC
HAVING COUNT(NUMAV) > 10
```

5.2.10. Exercise 10

- Quels sont les vols au départ de Nice entre 12h et 14h ?

```
SELECT *
FROM VOL
WHERE VD=« NICE »
AND HD BETWEEN 12 AND 14;
```

5.2.11. Exercice 11

- Quels sont les pilotes dont le nom commence par "S" ?

```
SELECT PILNOM
FROM PILOTE
WHERE PILNOM LIKE « S% »
```

5.2.12. Exercice 12

- Quels sont les noms des pilotes qui conduisent un avion que conduit aussi le pilote n°10 ?

Prédicative

```
SELECT PILNOM
FROM PILOTE,VOL, VOL1
WHERE PILOTE.NUMPIL=VOL1.NUMPIL
AND VOL.NUMPIL = 10
AND VOL.NUMAV= VOL1.AV #
AND VOL1.NUMPIL<>10
```

Ensembliste

```
SELECT PILNOM
FROM PILOTE
WHERE NUMPIL <>10
AND NUMPIL IN (SELECT NUMPIL
                FROM VOL
                WHERE NUMAV IN (SELECT NUMAV
                                FROM VOL
                                WHERE NUMPIL=10));
```

