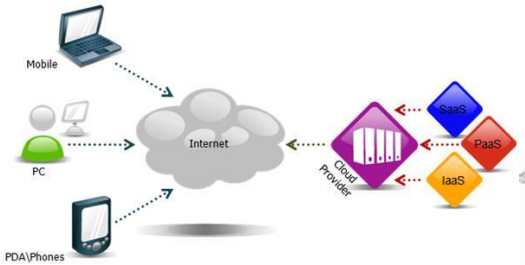
BASE DE DONNES

Rappels SQL LDD
Syntaxes et bonnes pratiques



Jérôme FESSY

Processus Conceptuel / Logique / Physique

Schémas relationnels :

- COMMANDE (**N°commande**, Date, Statut) ;
- PORTER (***N°article, N°commande***, Qte_commandée) ;
- ARTICLE (**N°article**, Désignation, Prix).

CREATE TABLE COMMANDE (…)
CREATE TABLE PORTER (….)
CREATE TABLE ARTICLE (…)
ALTER TABLE …
CREATE SEQUENCE …

SQL LMD
SELECT/INSERT/UPDATE/DELETE

SQL LDD
CREATE/DROP/ALTER

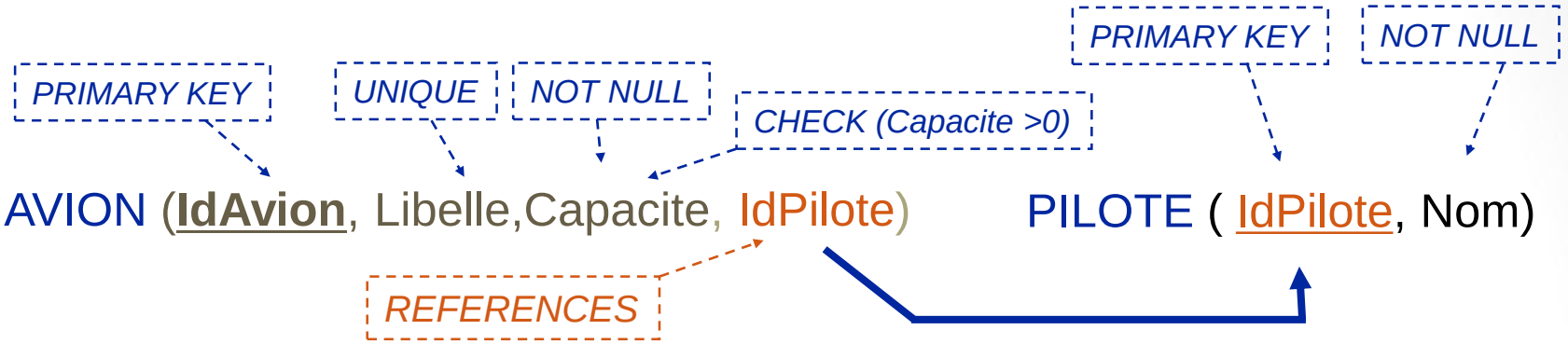
SQL1 1986 → SQL2 1992 → SQL3 2003 → SQL4

Norme SQL

SEQUENCE
TABLE
INDEX

Jérôme FESSY

Rappel : *CREATE TABLE et CONTRAINTES*




```
CREATE TABLE Pilote(
    IdPilote INTEGER  PRIMARY KEY,
    NomPilote VARCHAR(30) NOT NULL)
```

```
CREATE TABLE AVION (
    IdAvion INTEGER PRIMARY KEY,
    LibelleAvion VARCHAR(30) UNIQUE,
    Capacite INTEGER NOT NULL CHECK (Capacite>0) ,
    IdPilote INTEGER REFERENCES PILOTE( IdPilote) )
```

ECRITURE DE SCRIPT SQL

Définition de contraintes de clé primaire

```
CREATE TABLE Pilote(
IdPilote INTEGER  CONSTRAINT PK_Pilote PRIMARY KEY,
NomPilote VARCHAR(30) )
```

1

Définition de la contrainte lors de la définition de la colonne

```
CREATE TABLE Pilote(
IdPilote INTEGER,
NomPilote VARCHAR(30),
CONSTRAINT PK_Pilote PRIMARY KEY(idPilote) )
```

2

Définition de la contrainte comme une contrainte de table

```
CREATE TABLE Pilote(
IdPilote INTEGER,
NomPilote VARCHAR(30) )
/
ALTER TABLE PILOTE ADD CONSTRAINT PK_Pilote PRIMARY KEY(idPilote)
```

3

Définition de la contrainte en dehors du CREATE TABLE

ECRITURE DE SCRIPT

Définition de contraintes de référence

```
CREATE TABLE AVION (
IdAvion INTEGER, …,
IdPilote INTEGER CONSTRAINT FK_Pilote_Avion REFERENCES PILOTE( IdPilote) )
```

1

Définition de la contrainte lors de la définition de la colonne

```
CREATE TABLE AVION (
IdAvion INTEGER, …,
IdPilote INTEGER,
CONSTRAINT FK_Pilote_Avion FOREIGN KEY(idPilote) REFERENCES PILOTE( IdPilote) )
```

2

Définition de la contrainte comme une contrainte de table

```
CREATE TABLE AVION (
IdAvion INTEGER, …,
IdPilote INTEGER )
/
ALTER TABLE AVION
ADD CONSTRAINT FK_Pilote_Avion FOREIGN KEY(idPilote) REFERENCES PILOTE( IdPilote)
/
```

3

Définition de la contrainte en dehors du CREATE TABLE

ECRITURE DE SCRIPT : Bon usage d'écriture

Objectif : Ecrire un script homogène facile à lire et à maintenir

Nomenclature de nommage des :
TABLE; COLONNES; CONTRAINTES

1

```
CREATE TABLE AVION
( IdAvion          INTEGER  ,
  Libelle          VARCHAR(30) ,
  Capacite         INTEGER CONSTRAINT NN_Capacite  NOT NULL,
  TypeAvion        VARCHAR(10) ,
  IdPilote         INTEGER,
)
/
```

Création de la table sans contraintes

2

Ajout de l'ensemble des contraintes par table

3

```
ALTER TABLE AVION ADD CONSTRAINT PK_Avion_IdAvion  PRIMARY KEY (IdAvion);
ALTER TABLE AVION ADD  CONSTRAINT CK_CapaciteSup0 CHECK (Capacite>0) ;
ALTER TABLE AVION ADD  CONSTRAINT FK_Avion_Pilote FOREIGN KEY (IdPIlote)
                                                  REFERENCES PILOTE( IdPilote);
ALTER TABLE AVION ADD  CONSTRAINT CU_Capacite UNIQUE (Libelle);
```

Insertion de données dans les tables

Syntaxe SQL de la commande INSERT :

INSERT INTO < Nom_Table> [(<Liste_colonnes>)]
{ VALUES (<Liste_valeurs>) | **Requête_SQL** }

- Exemples :

INSERT INTO **AVION (IdAvion, LibelleAvion)** VALUES **(1, 'A380')** → OK

INSERT INTO **AVION (LibelleAvion, IdAvion)** VALUES **('A320', 2)** → OK

INSERT INTO **AVION** VALUES **(3, 'B747')** → OK mais déconseillé

INSERT INTO **AVION** VALUES **('B907', 4)** → KO

N.B. : répéter la commande INSERT pour chaque tuple à insérer

Jérôme FESSY

Modification et Suppression de données des tables

Syntaxe de la commande UPDATE

UPDATE <Nom_Table> SET <Attribut$_1$> = *expr$_1$* [,<Attribut$_2$>= *expr$_2$*] …

[WHERE *Condition*]

Exemple : *UPDATE AVION SET Capacite = 250 WHERE LibelleAvion= ‘ A320’*

Aspect ensembliste du UPDATE

Syntaxe de la commande DELETE

DELETE FROM <Nom_Table> [WHERE *Condition*]

Exemple : *DELETE FROM Pilote WHERE IdPilote >4*

Aspect ensembliste du DELETE

Jérôme FESSY