Apprendre le langage SQL par l'exemple
Partie 1 : le DDL

Historique

- **1969-1970** : Edgar Frank Codd, salarié d'IBM définit les principes du modèle relationnel.
- **1976** : création du SEQUEL par IBM
- 1977 :QUEL (QUEry Language), par Zook en 1977
- QBE (Query By Example), par Zloof
- **1981** : SQL (Structured Query Language), par IBM

Normalisation du SQL

- 1986 : SQL 86 - ANSI
- 1989 : ISO et ANSI
- 1992 : SQL 2 - ISO et ANSI
- **1999** : SQL 3 - ISO

Les 3 sous-langages

Le SQL est un langage dont l'objet principal est de pouvoir manipuler les données d'un système d'informations.

- **LDD** (Langage de Définition de Données) ou **DDL** (Data Definition Language)
  Création, modification et suppression des objets (tables, index, séquences, déclencheurs ou triggers, vues, triggers, synonymes, liens de bases de données, etc.)
  CREATE, **ALTER**, **DROP**
- **LMD** (Langage de Manipulation de Données) ou **DML** (Data Manipulation Language)
  Ajout, modification, suppression et extraction des données
  INSERT, **UPDATE**, **DELETE**, **SELECT**
- **LCD** (Langage de Contrôle de Données) ou **CDL** (Control Data Language)
  Gestion des droits, validation des données
  GRANT, **REVOKE**, **COMMIT**, **ROLLBACK**

Définition d'une base de données

Une base de données est un dispositif de stockage des données - total ou partiel - du système d'information de l'entreprise.

Les objectifs (liste non exhaustive)

1. Assurer la **cohérence** des données
2. **Eviter la redondance** des informations
3. **Extraire** les informations
4. **Sécuriser** l'accès aux données
5. Assurer **l'indépendance totale entre les données et les traitements**

Les bases de données relationnelles obéissent à l'algèbre relationnel. Les bases NoSQL se différencient par un stockage en mode colonne.

Marché de la base de données relationnelles

Nous ne disposons que de vieilles données sur les parts de marché des bases de données relationnelles commerciales !

Le marché des bases de données en 2007		
	Parts de marché en 2007	Parts de marché en 2006
Oracle	46,5%	45,7%
IBM	25,9%	27%
Microsoft	21%	20,6%
Teradata	3,7%	3,6%
Sybase	2,9%	3,1%
		Source : Gartner 2008

Tableau extrait de Zdnet

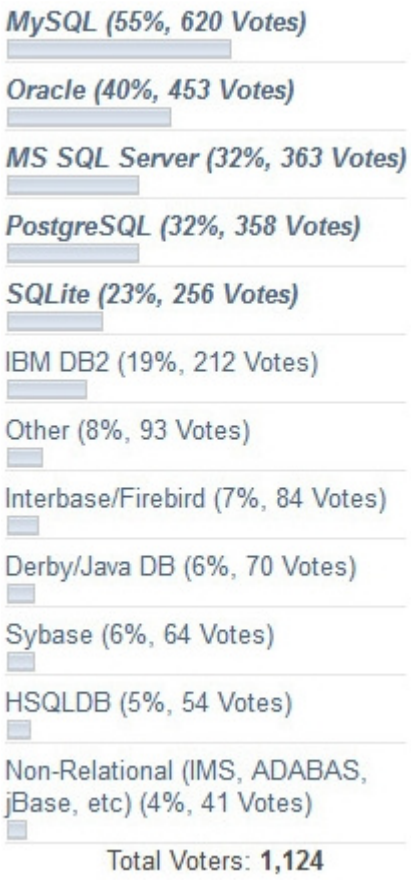
Les bases Open Source

Il est toujours de comparer sur la base de leurs prix des produits commerciaux à des produits Open Source dont la valeur d'achat est nulle !

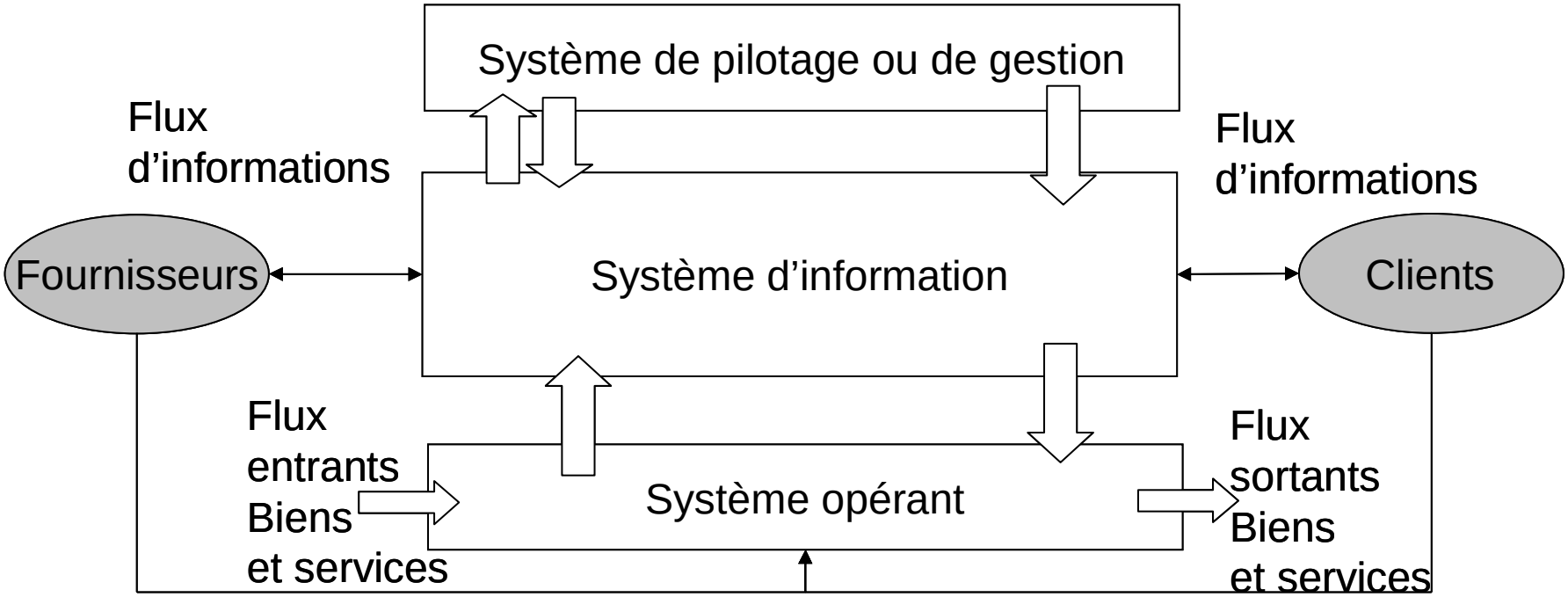
- MySQL, qui ne gère toujours pas à ce jour les contraintes de type CHECK !
- PostgreSQL, un fork de INGRES, qui a su prendre beaucoup d'autonomie
- Ingres, un des tout meilleurs moteurs de bases de données relationnelles
- Firebird, un fork de Interbase

Les usages en matière de bases de données

Le site DBeaver a récemment un sondage organisé sur les usages en matière de bases de données. Il en ressort que MySQL serait en 1ère position, devant Oracle, puis ex aequo Microsoft SQL Server et PostgreSQL.



Approche systémique



MERISE

La méthode MERISE est une méthode d'analyse du fonctionnement des organisations qui s'inspire en très grande partie de l'approche systémique. Initiée par l'INRIA, elle a vu le jour en 1979.

1. **Objectifs** assignés par la direction
2. Recueil des données par des **interviews** des opérationnels
3. Formalisation par un **dictionnaire des données**
4. **Modèle Conceptuel des Données**
5. **Modèle Logique des Données**
6. **Modèle Physique des Données**

Pour plus d'information à ce sujet, vous pouvez consulter mon support de cours relatif à Merise.

Pré-requis

Pour pouvoir mettre en œuvre les exercices présentés dans ce document, vous devrez installer le moteur de base de données Oracle Database 11g, ainsi que SQL Developer 3. Ces logiciels sont téléchargeables, après vous être inscrit, à partir des liens suivants :

- Oracle SQL Developer
- Oracle Database Software Downloads
- Oracle JDK

Tous les logiciels Oracle s'installent indifféremment sur Windows ou sur Linux.

Documentation

Internet regorge d'informations sur le SQL. Je vous recommande, pour ma part, d'aller à la source : Documentation SQL Oracle Database 11gR2 SQL

Distributions Linux supportées par Oracle

Oracle ne prend en charge le support que sur trois distributions Linux :

- **Oracle Enterprise Linux**, basée sur Red Hat
- **Red Hat**
- **SUSE Linux Enterprise Server** : Oracle Database 10 g et 11 g sont certifiées pour la SLES 10

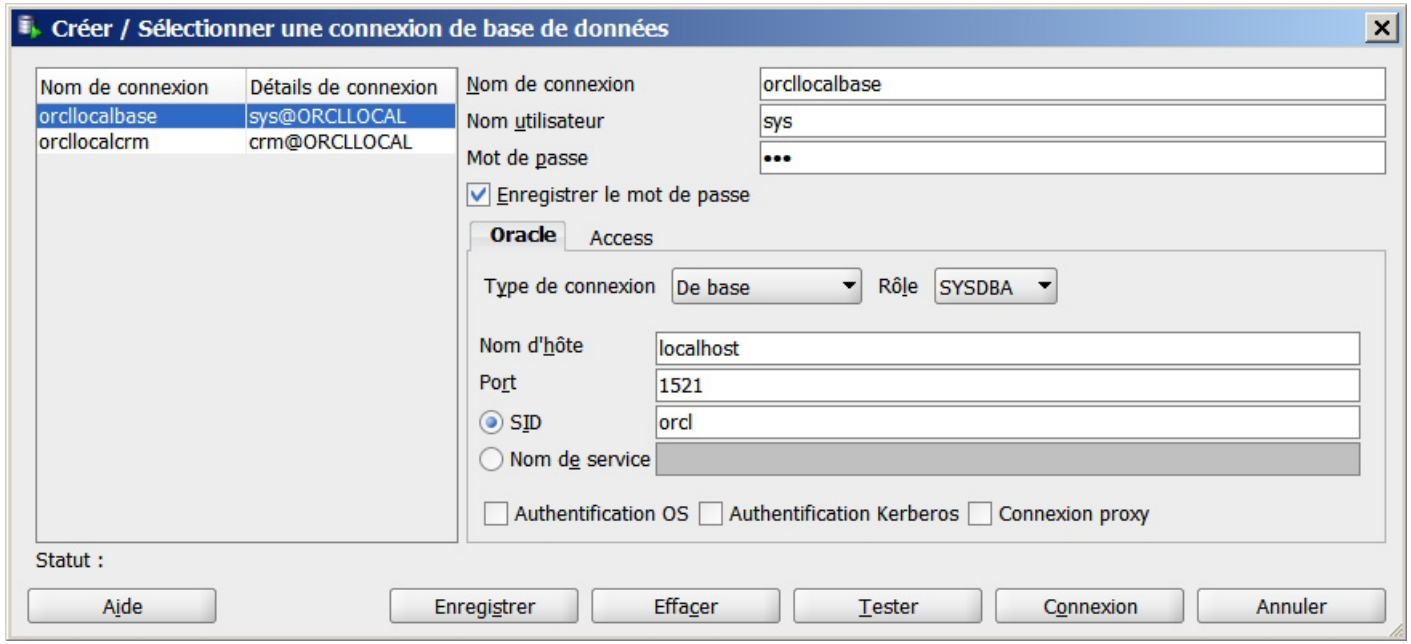
Source : **Linux Oracle FAQ**

Mise en œuvre de SQL Developer

SQL Developer 3 exige que vous disposiez du **JDK** 6 (Java SE Development Kit fourni par Oracle) sur votre système d'exploitation. Oracle propose un package complet comprenant SQL Developer et le JDK, uniquement pour la version 32 bits.

Connexion SQL Developer

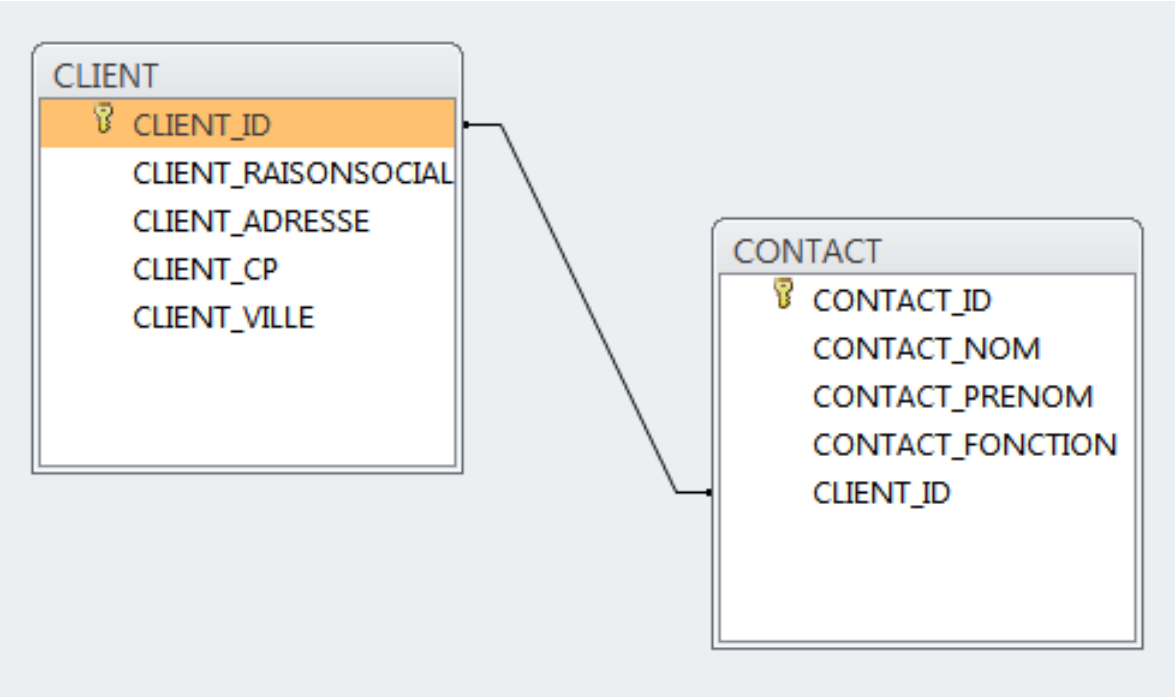
Après avoir installée la base Oracle, vous devez lancer Oracle SQL Developer.



Vous devez reprendre l'identificateur de l'instance que vous avez choisi lors de l'installation d'Oracle. L'utilisateur SYS contrairement à SYSTEM doit se connecter avec le rôle SYSDBA.

Le Modèle Physique des Données

L'exemple sur lequel nous nous appuierons est une application de Gestion de la Relation Clientèle, basée sur deux entités dans le Modèle Conceptuel des Données, transformées en tables dans le Modèle Physique des Données.



Création d'un tablespace

Le tablespace est un espace de disque logique. Conformément au modèle ISO relatif aux bases de données, son objet est d'isoler le niveau utilisateur (DDL) du niveau physique (système de fichiers).

```
CREATE TABLESPACE crm
DATAFILE 'd:\oracle\oradata\orcl\crm01.dbf' SIZE 1M;
```

Le moteur InnoDB sous MySQL emploie le concept d'espace de disque logique. Mais il est unique.

Création d'un schéma (utilisateur)

Un schéma est l'ensemble des objets de l'application. Sous Oracle, le schéma est aussi un utilisateur. Une fois les objets de l'application créée, le compte de l'utilisateur sera désactivé.

```
CREATE USER crm IDENTIFIED BY crm
DEFAULT TABLESPACE CRM
TEMPORARY TABLESPACE TEMP
QUOTA UNLIMITED ON crm;
```

Dans d'autres moteurs tels que MySQL, il faut créer une base de données à l'aide de l'instruction **CREATE DATABASE**. Sous Oracle, la base, c'est l'instance, le service – ou démon - qui lui est associé ! La notion de SCHEMA introduite récemment dans MySQL est proche de la notion de DATABASE.

Les privilèges système

Pour pouvoir créer des objets, il faut disposer de privilèges.

```
GRANT CREATE SESSION TO crm ;
GRANT CREATE TABLE TO crm ;
GRANT CREATE SEQUENCE TO crm ;
GRANT CREATE TRIGGER TO crm ;
GRANT CREATE SYNONYM TO crm ;
GRANT CREATE VIEW TO crm ;
GRANT CREATE DATABASE LINK TO crm ;
```

Vous pouvez aussi passer par un rôle, qui permet d'associer des privilèges à plusieurs utilisateurs.

Principaux type de données

Datatype	Type	Taille maximale
BLOB	Binaire	(2^{32} -1 octets) * (taille bloc)
CHAR [(size [BYTE | CHAR])]	Caractère	2000 octets
LONG	Caractère	2^{31} -1 octets
NCHAR[(size)]	Caractère	1 à 2000 octets
VARCHAR2(size [BYTE | CHAR])	Caractère	4000 octets
NVARCHAR2(size)	Caractère UTF	4000 octets.
DATE	Date	7 octets
NUMBER[(precision [, scale]])	Numérique	Précision : 1 à 38 Echelle : -84 to 127.

Contraintes

Elles sont issues de l'analyse **Merise** et des **règles de gestion** du dictionnaire de données.

1. La **clé primaire**
   Elle identifie la ligne de manière unique dans la mise en relation de l'enregistrement avec les lignes d'une autre table
2. La **contrainte d'unicité**
   Elle évite d'avoir des doublons sur le contenu des champs, en dehors de ceux qui constituent la clé primaire.
3. Les **contraintes CHECK**
   Elles permettent d'assurer la cohérence des données saisies.
4. Les **contraintes d'intégrité référentielle**
   Elles évitent d'inscrire des données dans une table « fille », si la référence n'existe pas préalablement dans la table « parent ».

Création de table

```
CREATE TABLE client
(
  client_id NUMBER(6, 0),
  client_raisonsociale VARCHAR2(50),
  client_adresse VARCHAR2(50),
  client_cp NUMBER(5, 0),
  client_ville VARCHAR2(50),
  CONSTRAINT client_pk PRIMARY KEY(client_id),
  CONSTRAINT client_uk UNIQUE(client_raisonsociale,client_cp),
  CONSTRAINT client_chk_cp CHECK(client_cp BETWEEN 1000 AND 95999),
  CONSTRAINT client_chk_id CHECK(CLIENT_CP IS NOT NULL),
  CONSTRAINT client_chk_raisonsociale CHECK(client_raisonsociale IS NOT NULL),
  CONSTRAINT client_chk_ville CHECK(client_ville IS NOT NULL)
);
```

Numérotation automatique

Contrairement à la plupart des moteurs de base de données, Oracle ne possède pas de type de données qui permet la numérotation automatique de la colonne qui constitue la clé primaire d'une table !

1. Il faut d'abord **créer une séquence**.
2. Il faut ensuite **créer un trigger** (un déclencheur) – un programme événementiel – qui incrémentera la colonne avec la valeur de la séquence.

Création de séquence

Vous devez fixer la valeur minimale, l'incrément et, au besoin, la valeur maximale (MAXVALUE).

```
CREATE SEQUENCE client_seq INCREMENT BY 1 MINVALUE
1;
```

Création d'un déclencheur (trigger)

```
CREATE OR REPLACE
TRIGGER client_trg BEFORE INSERT ON client
FOR EACH ROW
BEGIN
 IF :NEW.client_id IS NULL THEN
   SELECT client_seq.NEXTVAL INTO :NEW.client_id
   FROM DUAL;

 END IF;

END;
```

Création d'un index

Le rôle d'un index est avant tout d'accélérer la recherche.

```
CREATE INDEX client_idx ON CLIENT (client_ville);
```

Contrainte d'intégrité référentielle

```
CREATE TABLE contact
(
contact_id NUMBER(6, 0),
contact_nom VARCHAR2(30),
contact_prenom VARCHAR2(20) ,
contact_fonction VARCHAR2(50),
client_id NUMBER(6, 0),
CONSTRAINT contact_pk PRIMARY KEY(contact_id),
CONSTRAINT contact_uk
UNIQUE(contact_nom,contact_prenom,contact_fonction,client_id),
CONSTRAINT contact_fk_client FOREIGN KEY(client_id) REFERENCES CLIENT
(client_id),
CONSTRAINT contact_chk_id CHECK(contact_id IS NOT NULL),
CONSTRAINT contact_chk_nom CHECK(contact_nom IS NOT NULL),
CONSTRAINT contact_chk_prenom CHECK(contact_prenom IS NOT NULL)
);
```

Modification de table

Ajout de champ

```
ALTER TABLE contact
ADD (contact_email VARCHAR2(100) );
```

Ajout de contrainte

```
ALTER TABLE contact
ADD CONSTRAINT contact_chk_email CHECK
(REGEXP_LIKE(contact_email,'[a-z0-9-_\.]@[a-z0-9-
_\.]\.(fr|org|com|net)$', 'i'));
```

Renommer une colonne

Le fait de renommer une colonne peut amener à supprimer les contraintes qui en dépendent préalablement.

```
ALTER TABLE contact
DROP CONSTRAINT contact_chk_email;

ALTER TABLE contact RENAME COLUMN contact_email TO
contact_mail;

ALTER TABLE contact
ADD CONSTRAINT contact_chk_email CHECK
(REGEXP_LIKE(contact_mail, '[a-z0-9-_\.]@[a-z0-9-
_\.]\.(fr|org|com|net)$', 'i'));
```

Autres usages de la commande ALTER

La commande ALTER vous permet, à titre d'exemple, de réduire la place occupée par une table. Il faut autoriser préalablement le déplacement des lignes.

```
ALTER TABLE client ENABLE ROW MOVEMENT;
ALTER TABLE client SHRINK SPACE COMPACT;
ALTER TABLE client DISABLE ROW MOVEMENT;
```

Elle permet aussi de reconstruire les index.

```
ALTER INDEX client_pk REBUILD;
```

Suppression

La suppression d'une colonne de table se fait par la commande ALTER.

```
ALTER TABLE contact
DROP COLUMN contact_email;
```

La suppression d'une table entraîne celle des objets qui lui sont associés, à savoir les déclencheurs et les index.

```
DROP TABLE client;
```

La suppression de la séquence devra se faire manuellement.

```
DROP SEQUENCE client_seq;
```

Récupération d'une table supprimée

Dans l'environnement de l'utilisateur, sous SQL Developer, vous disposez simplement de la possibilité de récupération d'une table supprimée à partir de la *corbeille* :

Pour visualiser tous les objets de la corbeille en SQL:

```
SELECT * FROM dba_recyclebin;
```

Pour récupérer les données d'une table supprimée

```
FLASHBACK TABLE crm."BIN$CiHXEwOgRu2ob7NiIIrOvw==$0"
TO BEFORE DROP RENAME TO CONTACT;
```

Purge

Suppression avec purge

Si vous ne souhaitez pas que votre table aille à la corbeille :

```
DROP TABLE crm.contact PURGE;
```

Purge d'une table en deux temps

Vous pouvez aussi procéder en deux temps.

```
DROP TABLE crm.contact;
PURGE TABLE crm."BIN$dfG+W4y4SSOJbEjjCiUpVA==$0";
```

Vider la corbeille

Purge des objets d'un utilisateur

```
PURGE recyclebin;
```

Purge de la corbeille

```
PURGE dba_recyclebin;
```

Activer / Désactiver la corbeille

Désactiver la corbeille

```
ALTER SESSION SET recyclebin = OFF;
ALTER SYSTEM SET recyclebin = OFF;
```

Activer la corbeille

```
ALTER SESSION SET recyclebin = ON;
ALTER SYSTEM SET recyclebin = ON;
```