

# Introduction aux bases de données

Oracle 11g

# Introduction

- ▶ Retour sur la séance précédente
- ▶ Point de vue de l'étudiant
- ▶ Point de vue de l'enseignante

# Introduction

- ▶ Oracle 11g, le serveur. Le serveur est au Collège au Local P143.(à vérifier)
  - ▶ Pas besoin d'installer le serveur sur les postes des étudiants
- ▶ SQL Developer, l'interface client pour accéder aux objets.
  - ▶ TOUS les étudiants doivent télécharger SQL Developer sur leur poste de travail
    - ▶ C'est fait
- ▶ Connexion au serveur Oracle. (c'est fait)
- ▶ Présenter SQL developer (retour)
- ▶ Quelques objets d'Oracle
  - ▶ Users, Tables, vues, séquences etc .....
- ▶ Conclusion.

# Introduction

- ▶ Historique: → **Les problèmes**
  - ▶ SGF (systèmes de gestion de fichiers) vers le début des années 60.
  - ▶ Chaque application a ses propres fichiers.
  - ▶ Si plusieurs applications utilisent le même fichier, alors la description de celui-ci doit se faire au niveau de chacune des applications.
  - ▶ Redondance accrue des données
  - ▶ Difficulté de mise à jours
- ▶ Base de données: → **La solution**. (la première fois en 1964)
  - ▶ Les données sont indépendantes des programmes
  - ▶ La redondance est réduite
  - ▶ La mise à jour est facilitée
  - ▶ Les accès aux données sont contrôlés
  - ▶ Nous avons une intégrité des données.

# Introduction

## ► Définitions

- Une base de données est un ensemble de données modélisant les objets d'une partie du monde réel et se servant de support à une application informatique.
- Un SGBD (système de gestion de base de données) peut-être perçu comme un ensemble de logiciels système permettant aux utilisateurs d'insérer, de modifier et de rechercher efficacement des données spécifiques dans une grande masse d'information partagée par de multiples utilisateurs.
- Exemples

# Introduction

## ► Rôles des SGBDs

- Non redondance des données : permet de réduire le risque d'incohérence lors des mises à jour, de réduire les mises à jour et les saisies.
- Partage des données : ce qui permet de partager les données d'une base de données entre différentes applications et différents usagers.
- Cohérence des données : ce qui permet d'assurer que les règles auxquelles sont soumises les données sont contrôlées surtout lors de la modification des données.
- Sécurité des données : ce qui permet de contrôler les accès non autorisés ou mal intentionnés. Il existe des mécanismes adéquats pour autoriser, contrôler ou d'enlever des droits à n'importe quel usager à tout ensemble de données de la base de données.

# Introduction

- ▶ Le modèle **relationnel**: début des années 70
  - ▶ Définit par le Britannique Edgar Frank Codd et basé sur l'algèbre relationnelle.
  - ▶ Les données sont organisées dans des tables ou des relations.
  - ▶ Les liens qui existent entre les relations est exprimé sous forme de clé étrangère ou clé primaire.
  - ▶ C'est le modèle le plus répandu de nos jours.
  - ▶ Les SGBDS relationnels utilisent le langage SQL pour manipuler les données de la base de données.

# Introduction

- ▶ Exemples de SGBDR: (dans l'ordre de popularité)
  - ▶ Oracle(Oracle)
  - ▶ MySQL (Oracle)
  - ▶ SQL server(Microsoft)
  - ▶ PostgreSQL
  - ▶ DB 2 (IBM)
  - ▶ SQLite
- ▶ Autres SGBD
  - ▶ MongoDB (orienté documents)
  - ▶ Cassandra (orienté colonnes)

Source: <https://db-engines.com/en/ranking>

# Introduction

## Présentation du SGBDR Oracle

- ▶ Architecture:
  - ▶ Noyau: assure la gestion de l'intégrité des données, le stockage des données, les accélérateurs (index, clusters ), gestion des connexions à la BD et l'exécution des requêtes.
  - ▶ Le dictionnaire: décrit de manière dynamique les objets et les utilisateurs de la base de données.
  - ▶ La couche SQL: interface d'accès aux objets. Tous les accès se font en utilisant le langage SQL
  - ▶ La couche PL/SQL: extension de la couche SQL.

# Introduction

- ▶ Une table:
  - ▶ objets contenant les données des utilisateurs ou appartenant au système. Une table est composée de colonnes (Champs ou attributs) et de lignes (enregistrements ou occurrences).
  - ▶ Exemples: Table Joueurs

Nom	Prénom	Nb_Points
Patoche	Alain	44
Zouba	Alex	88
Lafleur	Serge	54
Patoche	Peter	102

# Introduction

- ▶ Dans la table joueurs:
  - ▶ **Une ligne** représente un enregistrement de la table. (Patoche, Alain, 44) est un enregistrement.
  - ▶ **Les colonnes** Nom, prénom et Nb\_points sont des attributs de la table joueurs. Ils sont également appelés «champs»
  - ▶ Lafleur est une valeur de l'attribut NOM.
- ▶ Remarque: dans la table joueurs, il existe deux enregistrements ayant la même valeur pour l'attribut nom.
- ▶ Afin de distinguer les enregistrements nous introduisons la notion de clé primaire ou **PRIMARY KEY**.

# Introduction

- ▶ Une clé primaire **PRIMARY KEY** est un attribut d'une table permettant d'identifier chaque enregistrement de manière unique.
- ▶ Dans la table suivantes, tous les joueurs ont un numéro unique. Le numéro est donc une clé primaire

Numero	Nom	Prénom	Nb_Points
21	Patoche	Alain	44
55	Zouba	Alex	88
93	Lafleur	Serge	54
78	Patoche	Peter	102

# Introduction

## ► Définitions:

- SQL pour Structured Query Language est un langage de définition de données (DDL, Data Definition Language), de manipulation de données (DML, Data Manipulation Language) et de contrôle de données (DCL, Data Control Language)
- DDL: CREATE, ALTER , DROP
- DML: INSERT INTO, UPDATE, DELETE
- DCL: GRANT, REVOKE
- La commande SELECT, qui permet d'extraire les enregistrements

# Introduction

## ► Conseils Généraux pour le langage SQL:

- SQL n'est pas sensible à la casse, cependant il est conseillé d'utiliser les mots réservés (commandes, le type de données ...) en majuscules
- Il ne faut pas oublier le point virgule à la fin de chaque ligne de commande.
- Utiliser les deux traits -- pour mettre une ligne en commentaire
- Utiliser /\* et \*/ pour mettre plusieurs lignes en commentaire
- Utiliser des noms significatifs pour les objets que vous créez
- Ne pas utiliser de mots réservés comme noms d'objets (tables, vue, colonne..)
- Mettre une clé primaire pour chacune des tables que vous créez
- Si vous avez à contrôler l'intégrité référentielle, alors il faudra déterminer l'ordre dans lequel vous allez créer vos tables.

# Introduction

La commande SELECT:

La commande SELECT est la commande la plus simple à utiliser avec SQL. Cette commande n'affecte en rien la base de données et permet d'extraire des données d'une ou plusieurs tables. La syntaxe simplifiée n'utilise pas de jointure et elle se présente comme suit :

```
SELECT <nom_de_colonne1,...nom_de_colonnen>
```

```
FROM <nom_de_table>
```

```
WHERE <condition>
```

```
ORDER BY <nom_de_colonne>;
```

**Remarque: le résultat d'une requête SELECT est une table.**

# Introduction

Quelques opérateurs de comparaison

=, !=, <, <=, >, >=, Like, IN, NOT IN

IS NULL, IS NOT NULL

BETWEEN x AND Y ...

(voir page 20 du document pour tous les opérateurs de comparaison)

Les opérateurs logiques OR et AND peuvent être utilisés pour combiner deux conditions

Exemples :