



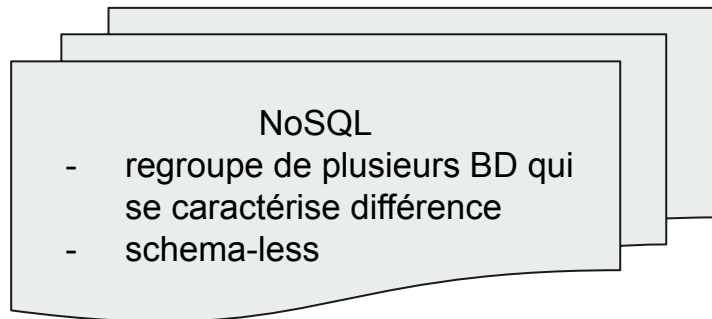
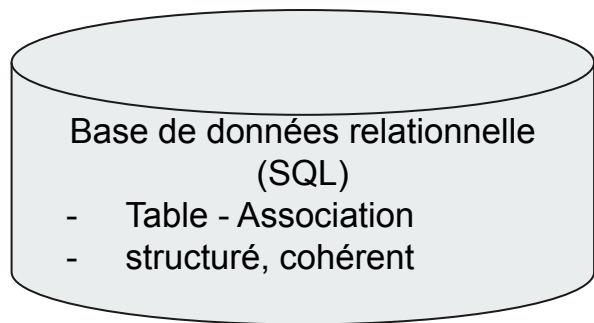
# L'accès aux bases de données

Quel paradigme? Quelles solutions?

Nga & Luca

# Introduction

- Les bases de données sont importantes, et une base de données ne sert qu'à lire/écrire des données
- Les langages peuvent lire et écrire dans des fichiers et dans de multiples formats  
-> comment peut-on accéder aux bases de données



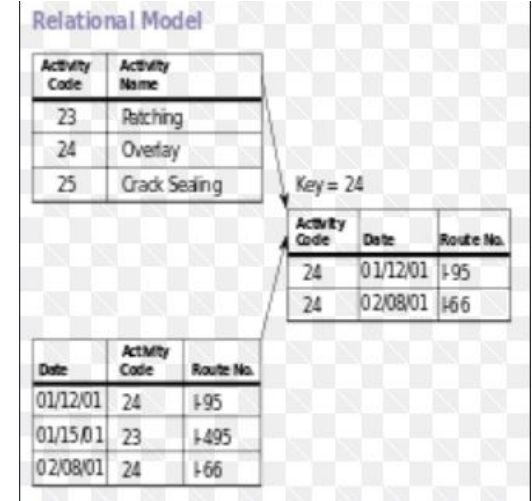
# Base de données relationnelles - DBR

1. l'information doit être stockée dans des **tableaux à deux dimensions**, colonnes et lignes (enregistrements)
2. l'association entre des tables est représenté par clé étrangère
3. doit respecter la norme ACID

## Atomicité, Cohérence, Isolation et Durabilité

-> garantissent qu'une transaction est exécutée de façon fiable

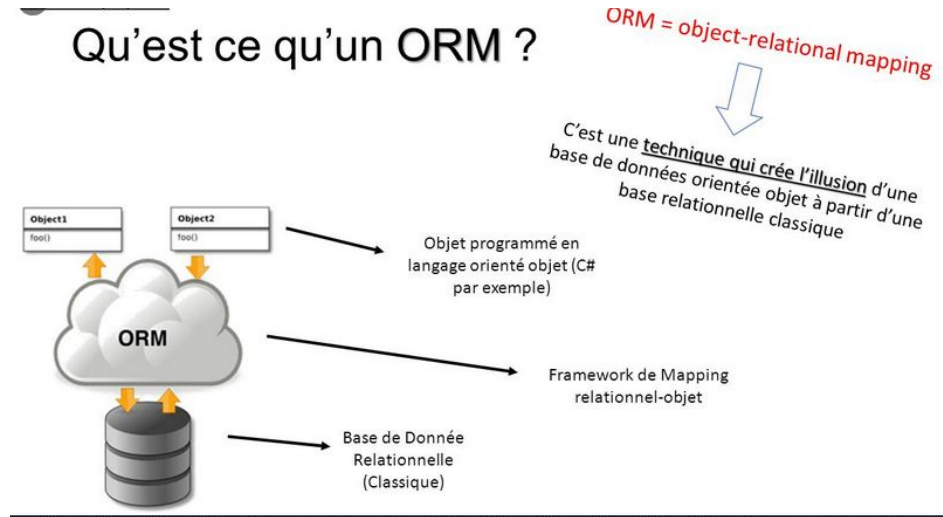
4. Langage de requête structuré SQL
  - a. LDD: créer les objets de la base de données (tables, vue, index, schéma,...)  
CREATE, ALTER, DROP
  - b. LMD: utiliser des commandes de filtrer, supprimer, mettre à jour  
SELECT, INSERT, UPDATE, DELETE
  - c. LCD: gérer les droits des utilisateurs, administrer les transactions



# ORM avec DB Relationnelle

- RDB doit se rapprocher de l'objet, et le langage-objet doit faire de même
- Intérêt de ORM
  - facilite la vie au codeur: gérer des classes (pas tables) et les méthodes de classe (lancer des requêtes)
  - l'indépendance vis-à-vis de la base de données
    - ORM fonctionne de la même manière avec SQLite, autre GSDBR
  - il est une couche d'abstraction qui exploite la base de données

## Qu'est ce qu'un ORM ?





## Les deux principaux paradigmes ORM: le Data Mapper pattern et l'Active Record pattern

### Active Record pattern

```
$user = new User;  
$user->username = 'philipbrown';  
$user->save();
```



### Exemple Active Record

```
c = Contact()  
c.nom = "Guido Van Rossum"  
c.tel = "06 01 02 03 04"  
c.save()
```

```
# création d'un nouveau contact  
# affectation des champs  
  
# Ecriture dans la base
```



## Le Data Mapper pattern

### Exemple Data Mapper Pattern

```
session = create_session()    # Création d'une session

c = Contact()                 # Nouveau contact
c.nom = "Guido Van Rossum"    # affectation
c.tel = "06 01 02 03 04"

session.add(c)                # ajout du nouveau contact
session.commit()              # validation de la transaction
```

# Librairie ORM dans Python



1. ORM pour RDB
  - a. Django: framework python consacré au développement web 2.0 (2005)
  - b. une belle boîte à outils qui aide et oriente le développeur
  - c. inspire s/modèle MVC
  - d. gère de faire la requête compatible en fonction du GBDR
2. ORM pour NoSQL
  - a. Flask



# Conclusion

---

