

Introduction aux BDD Relationnelles et à SQL

W4D3

Objectifs de la matinée

- BDD Relationnelles / Diagramme Entités - Relations
- SQL :
 - Différence entre SQL et d'autres langages de programmations
 - Utilisation de SQL avec des bases de données
 - Rôle en tant que Data Analyst versus d'autres rôles
 - Expliquer les relations entre données à l'intérieur d'une base de données
 - Utiliser SELECT

Base de données et Tables

- Base de données : un conteneur
- Table : une liste structurée de données



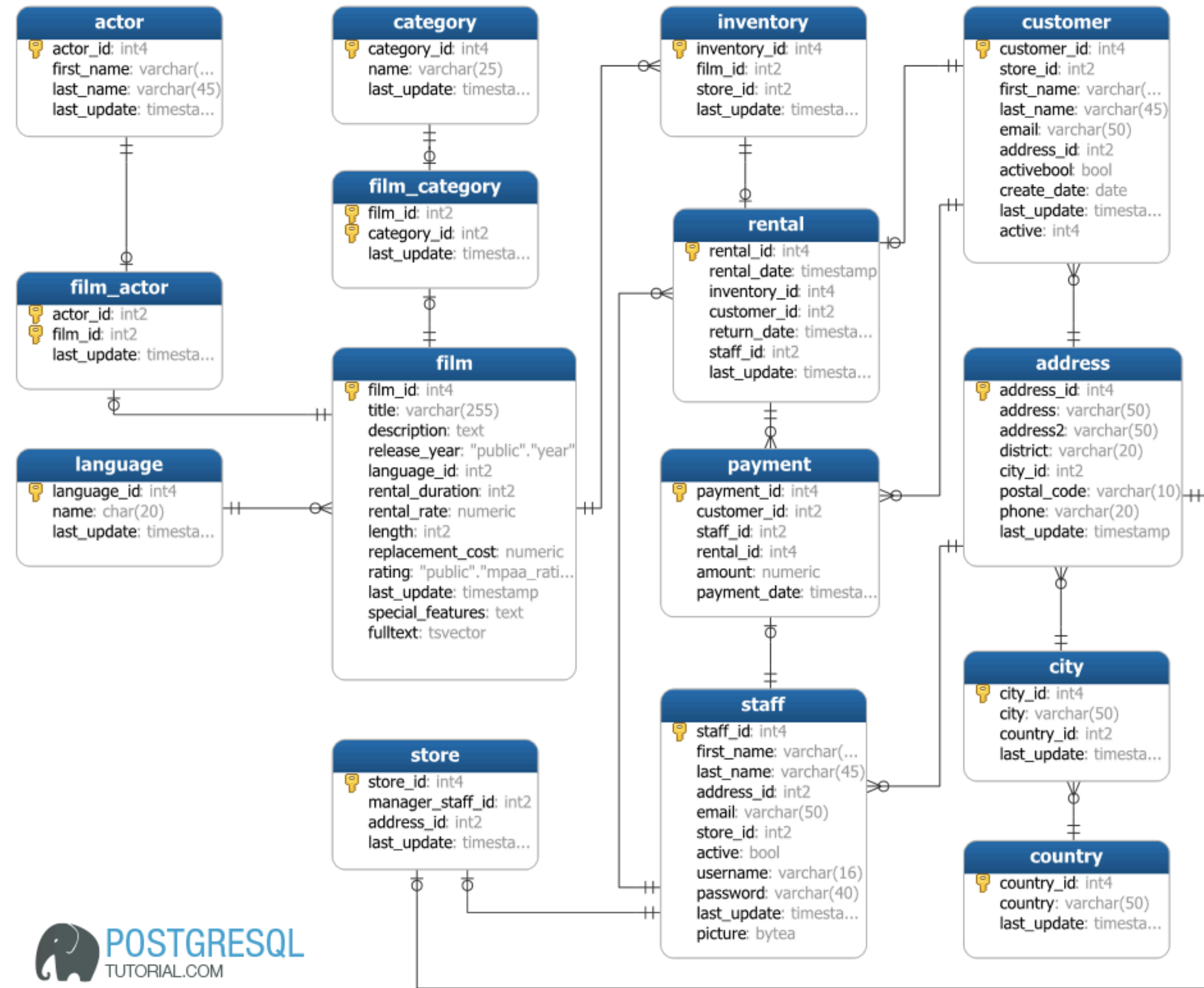
Colonnes et rangées

- Colonne : un seul champ
- Rangée : un enregistrement / échantillon

[illegible]

Data Modeling

- Organiser et structurer l'info en plusieurs tables reliées
- Peut représenter un process business ou des relations entre process business



Evolution des modèles de données

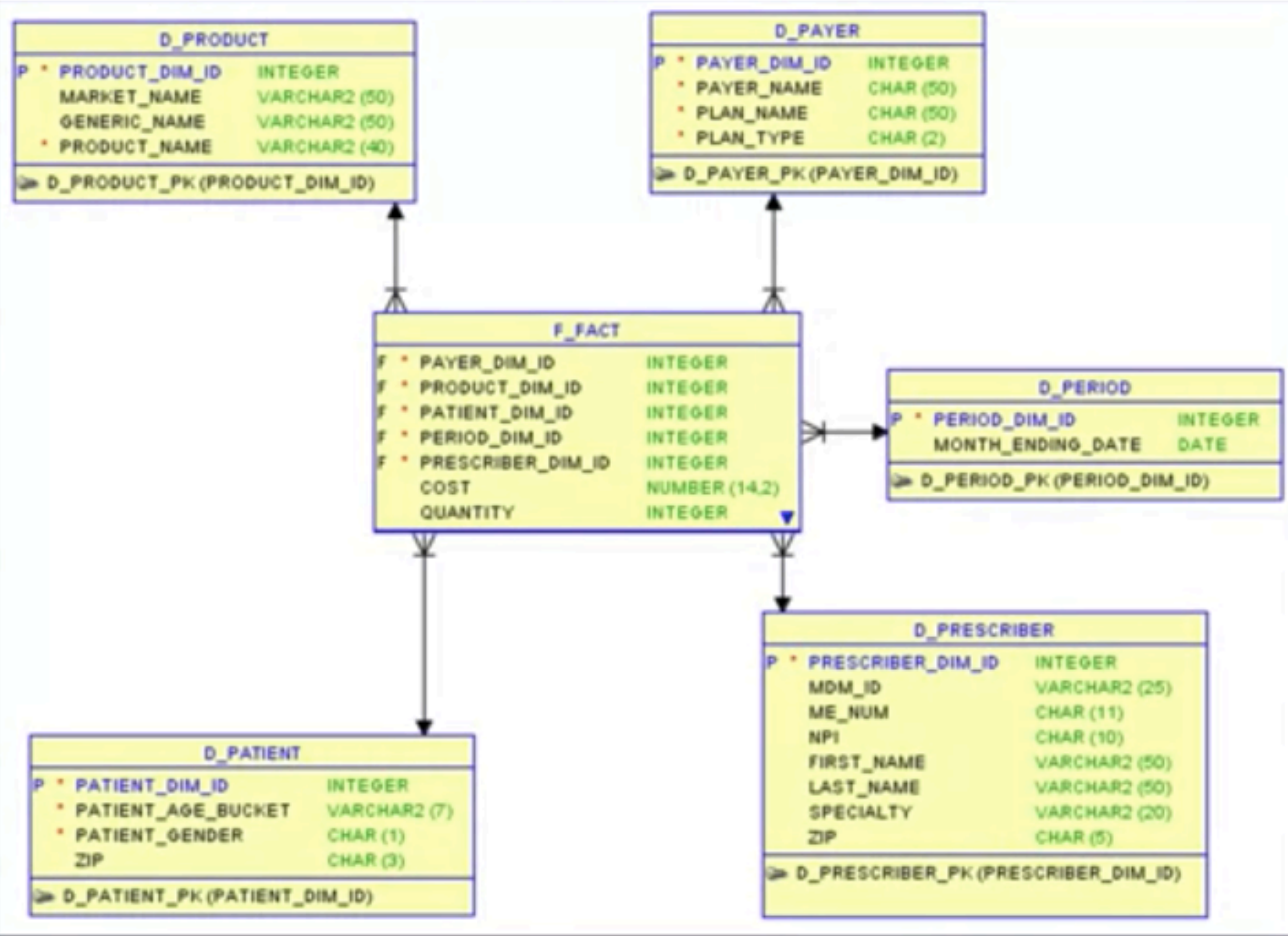
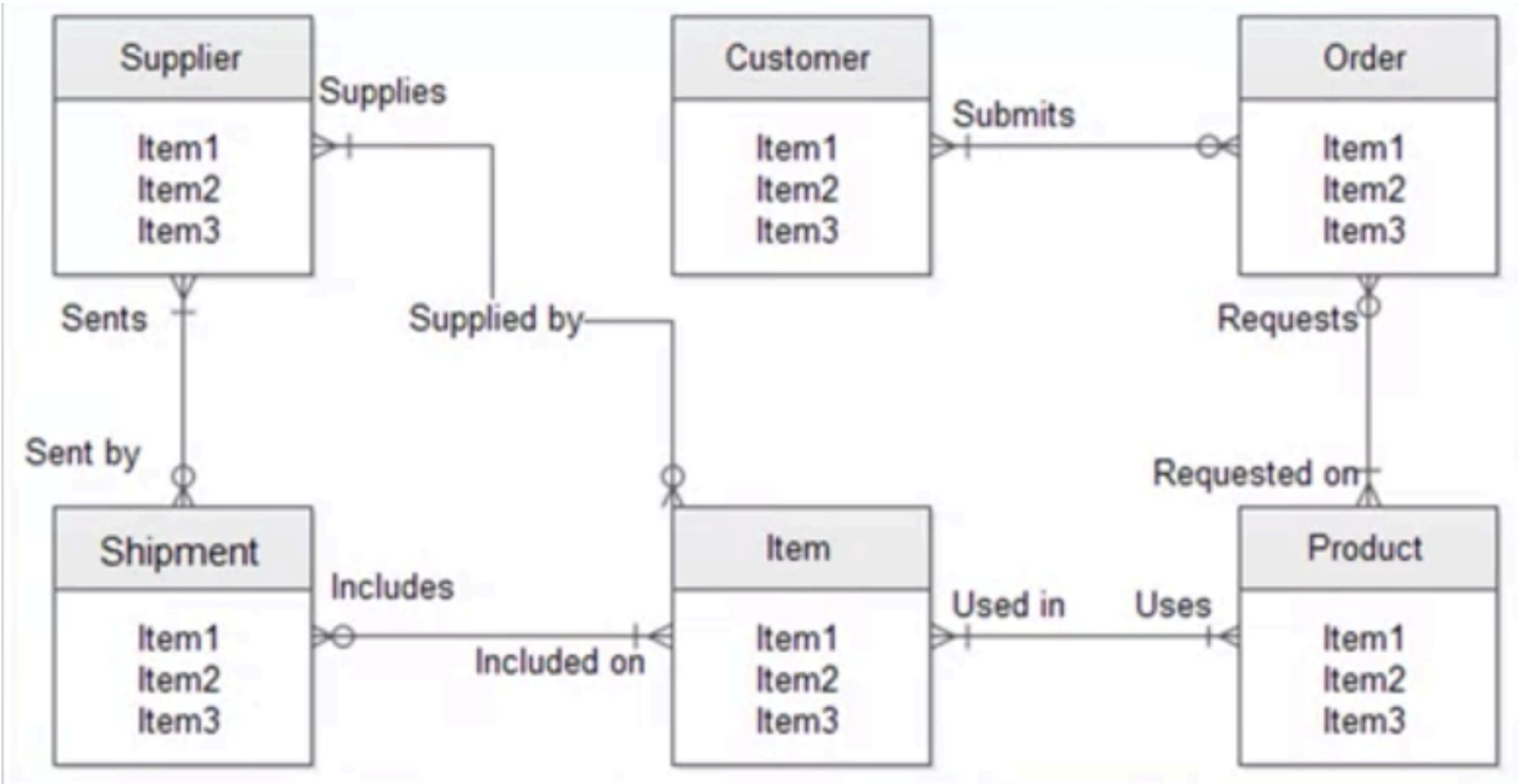
1960	Hierarchical	Difficult to represent M:N relationships (hierarchical only)
1969	Network	Structural level dependency No ad hoc queries (record-at-a-time access) Access path predefined (navigational access)
1970	Relational	Conceptual simplicity (structural independence) Provides ad hoc queries (SQL) Set-oriented access
1976	Entity Relationship	Easy to understand (more semantics) Limited to conceptual modeling (no implementation component)
1978	Semantic	More semantics in data model Support for complex objects
1985	Object-Oriented	Inheritance (class hierarchy) Behavior Unstructured data (XML)
1990	Extended Relational (O/R DBMS)	XML data exchanges
2009 Big Data	NoSql	Addresses Big Data problem Less semantics in data model Based on schema-less key-value data model Best suited for large sparse data stores

Briques pour construire une BDD Relationnelle

- **Entités** : personnes, lieux, évènement. Distinguable, unique.
- **Attributs** : Une caractéristique de cette entité.
- **Relations** : Relation entre entités :
 - One to Many
 - Many to Many
 - One to One

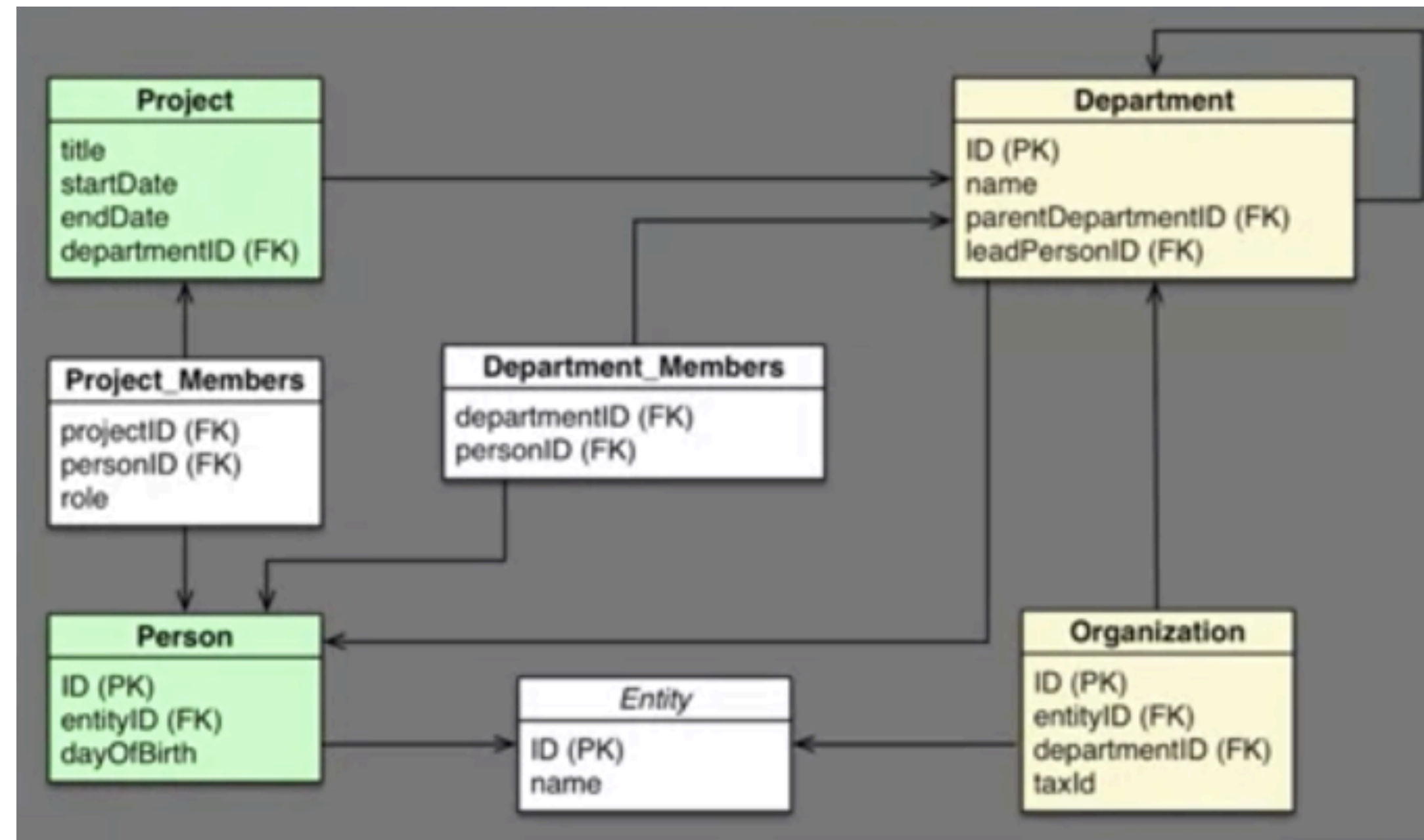
Diagrammes ER

- Représente une BDD Relationnelle



Clés primaires et clés étrangères

- Clé primaire : Une colonne dont les valeurs identifient de manière unique la rangée.
- Clé étrangère : Une ou plusieurs colonnes qui ensemble peuvent identifier une rangée dans une autre table.



Différences de notation

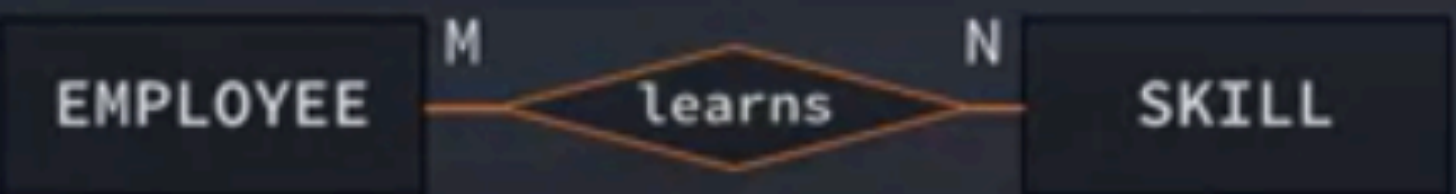
Chen Notation

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



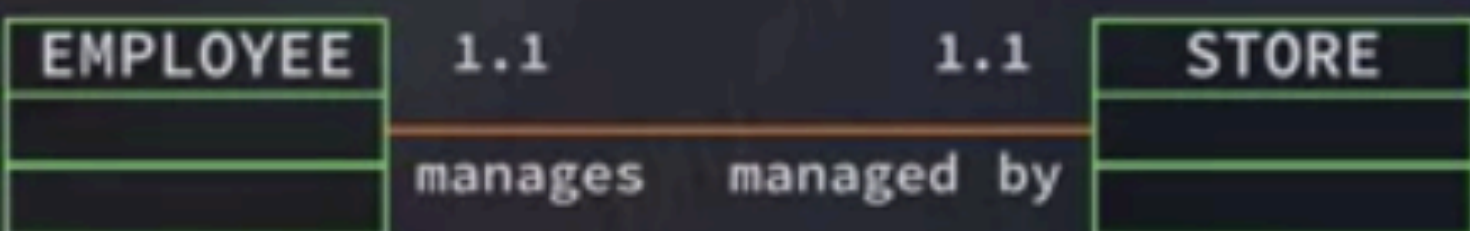
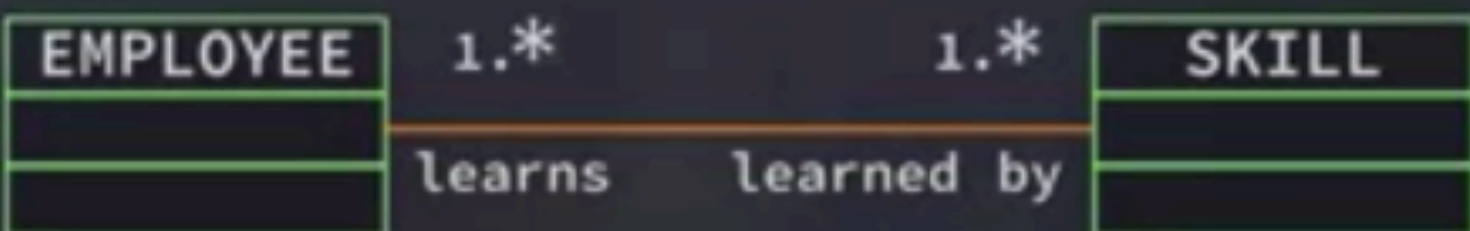
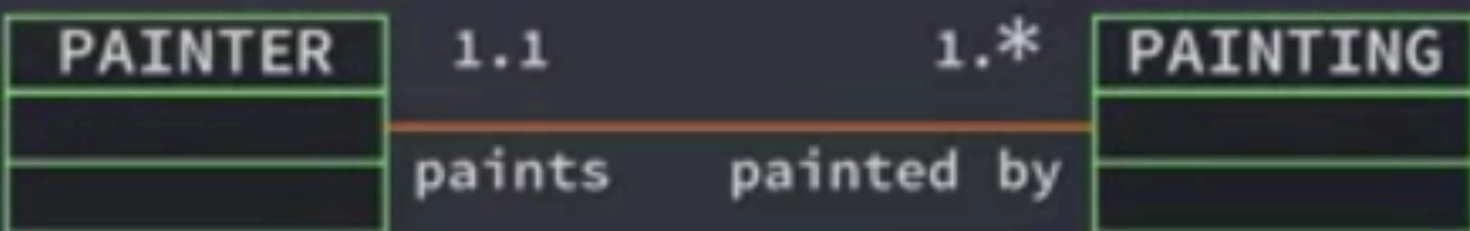
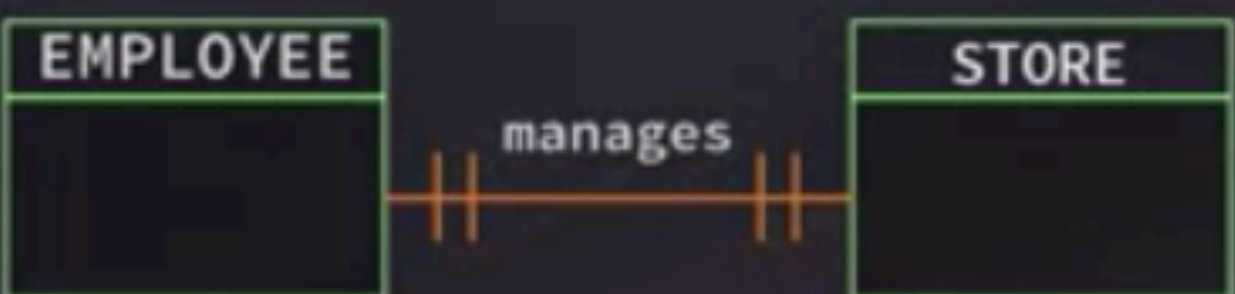
Crow's Foot Notation

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



UML Class Diagram Notation

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



Comprendre la donnée avant de coder

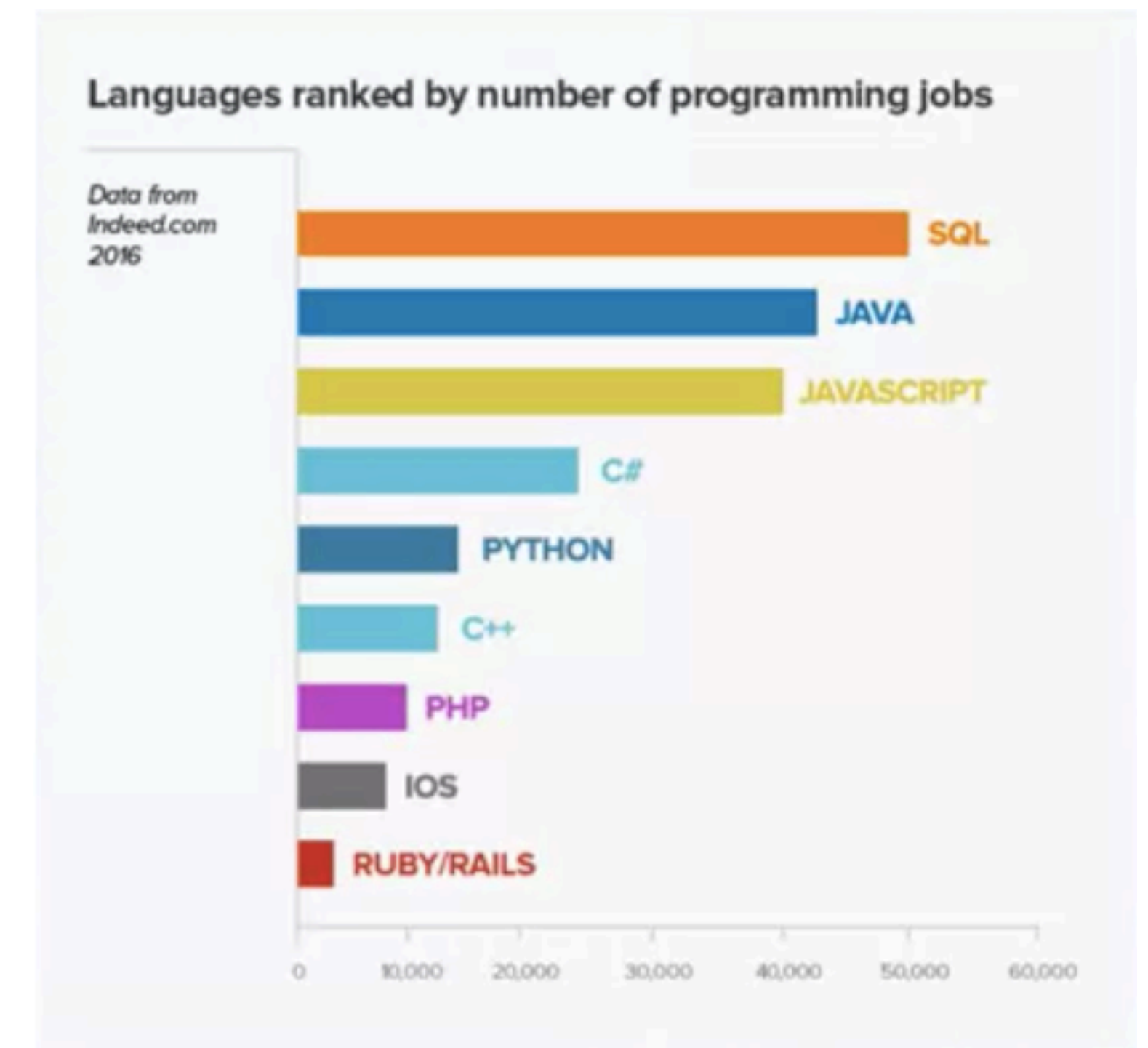
- Process business
- Comment la donnée est organisée et se structure

Qu'est ce que SQL ?

- Structured Query Language (SQL)
- Langage standard pour la communication avec et la gestion de BDD relationnelles.
- requête, insérer, mettre à jour, modifier
- Déclaratif
- Non procédural

Qui utilise SQL ?

- Backend, QA, Data Architect, Data Administrator, Data Analyst, Data Scientist, etc.
- Le Data Administrator :
 - Gouverne la BDD
 - Gère les permissions utilisateurs
 - Gère et crée les tables
- Le Data Analyst :
 - End user de la BDD
 - Utilise des requêtes SQL pour lire la donnée



Data Base Management Systems

- Un seul SQL mais plusieurs DBMS :
 - **MySQL**
 - PostgreSQL
 - SQLite
 - Microsoft SQL Server
 - IBM DB2 Oracle
 - Etc.
- Plusieurs différences de syntaxe.

SQL et le Big Data

- NoSQL - Not only SQL
- Rechercher l'info sans utiliser de BDD relationnelles

Syntaxe SELECT

- **Commande basique** : `SELECT nom_du_champ FROM nom_du_tableau`
- **Plusieurs colonnes** : `SELECT prenom, nom FROM client`
- **Toutes les colonnes** : `SELECT * FROM client`

Plus tard...

- Joindre un autre tableau aux résultats
- Filtrer pour ne sélectionner que quelques enregistrements
- Classer les résultats
- Grouper les résultats pour faire uniquement des statistiques (note moyenne, prix le plus élevé, etc.)

```
SELECT *  
FROM table  
WHERE condition  
GROUP BY expression  
HAVING condition  
{ UNION | INTERSECT | EXCEPT }  
ORDER BY expression  
LIMIT count  
OFFSET start
```