

- Stockage et organisation des données
- Systèmes de gestion de bases de données (SGBD)
- Architecture client/serveur
- Modèle entités-associations
- Modèle relationnel
- Schéma relationnel
- Normalisation d'un schéma de base de données
- Langage SQL

- 1960-  
Systèmes de gestion de fichiers, collection de données, bases de données (modèle réseau)  
- nécessité de savoir où et comment sont stockées quelles données
- 1970-  
Émergence du **modèle relationnel** et de son implémentation  
+ indépendance entre l'organisation logique et le stockage physique
- 1980-  
**SGBD relationnels**, modèles avancés  
relationnel étendu, OO (orienté objet), déductif, *etc.*  
et orientés application (spatial, scientifique)

# Organisation des données

## • Fichiers

- ex : tableur, UniProt
- Problème : programme spécifique pour l'accès
  - pas d'accès simultanés
  - redondances/incohérences

## • Système de Gestion de Bases de Données

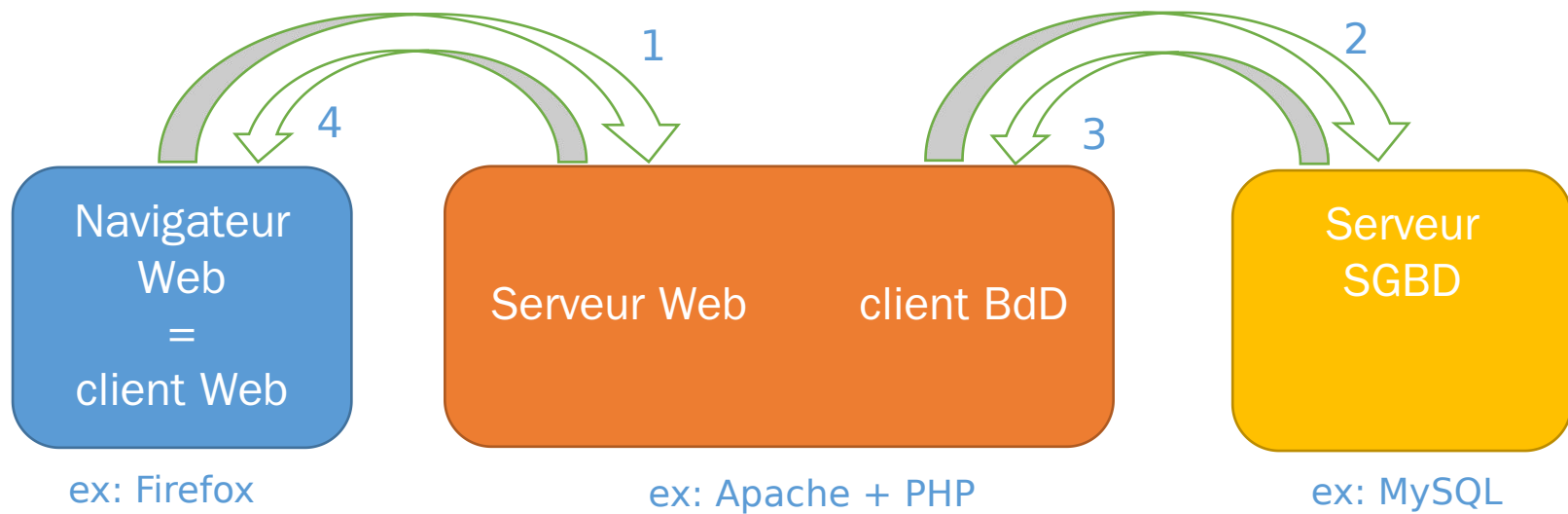
- intégrité des données
- partage et accès simultanés
- performances (On-Line Transaction Processing ou OLTP)
- sécurité et robustesse (reprise après une panne)
- indépendance physique (modification de l'organisation du stockage physique sans modification des applications qui utilisent le système)

```

ID Q8DPI7_STRR6 PRELIMINARY; PRT; 286 AA.
AC Q8DPI7;
DT 01-MAR-2003, integrated into UniProtKB/TrEMBL.
DT 01-MAR-2003, sequence version 1.
DT 02-MAY-2006, entry version 10.
DE DNA processing Smf protein.
...
DR Pfam; PF02481; SMF; 1.
DR TIGRFAMs; TIGR00732; dprA; 1.
KW Complete proteome.
SQ SEQUENCE 286 AA; 31583 MW; CF12DB83AE3663A2
MELFMKITNY EIYKLLKKSGL TNQQILKVLV YGENVDQELL L
FQIDDAHLSK EFQKFPSFSI LDDCYPWDLS EIYDAPVLLF Y
CSKQGAKSVE KVIQGLENEL VIVSGLAKGI DTAAHMAALQ
NKRLQDYIGN DHLVLSEYGP GEQPLKFHFP ARNRRIAGLC
AMEEGRDVFA IPGSILDGLS DGCHHLIQEG AKLVTSGQDV
//
  
```

Fiche UniProt

# Architecture Client - Serveur

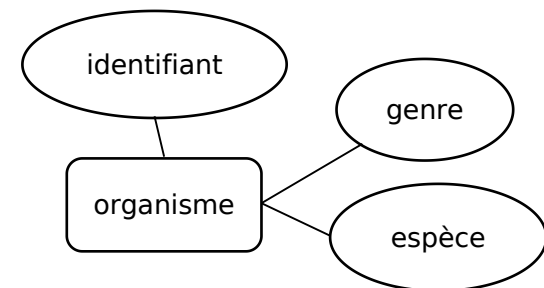
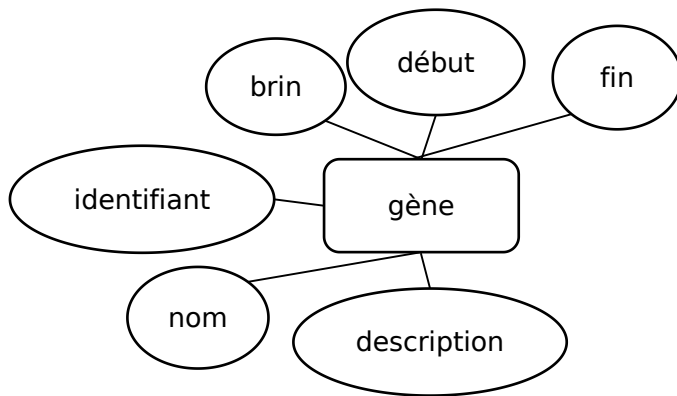


1. Demande d'une page Web (protocole HTTP)
  1. page dynamique (code PHP exécuté à la volé, connexion et requête au SGBD)
2. Requête au serveur de bases de données (SQL)
3. Renvoi des données
4. Mise en forme (HTML) et renvoi de la page Web

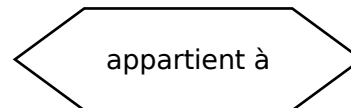
# Conception d'une base de données

## Modélisation de la réalité avec par exemple un modèle Entités-Associations

- Entité : Objet (matériel ou pas) ou concept
  - ex: séquence nucléique, organisme tel *Escherichia coli* ou *Saccharomyces cerevisiae*, une image d'arabette
  - avec une liste d'attributs. ex: nom de fichier, écotype d'arabette pris en photo, date de la photo, surface foliaire mesurée en cm<sup>2</sup>

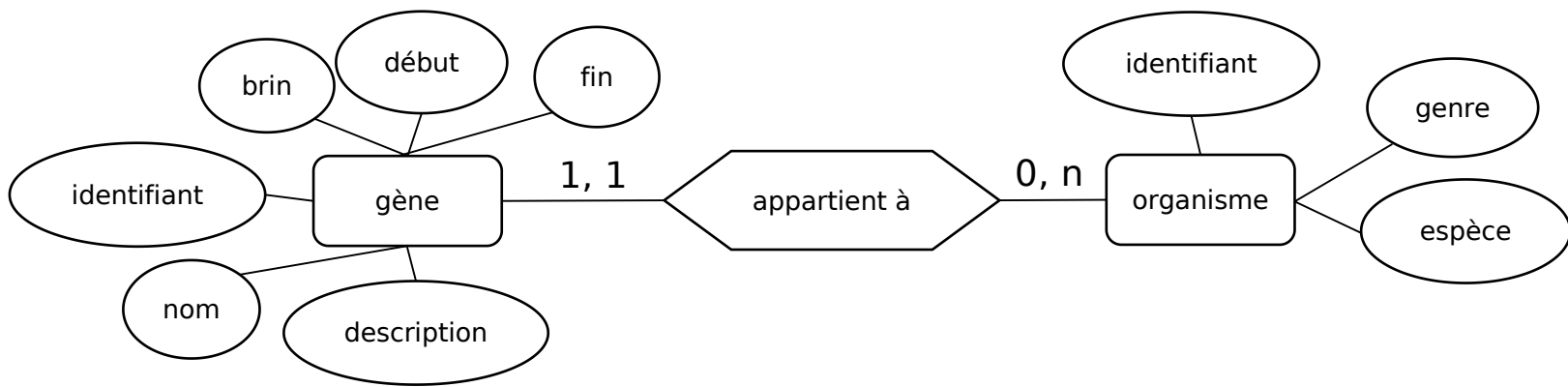


- Association : lien entre deux entités



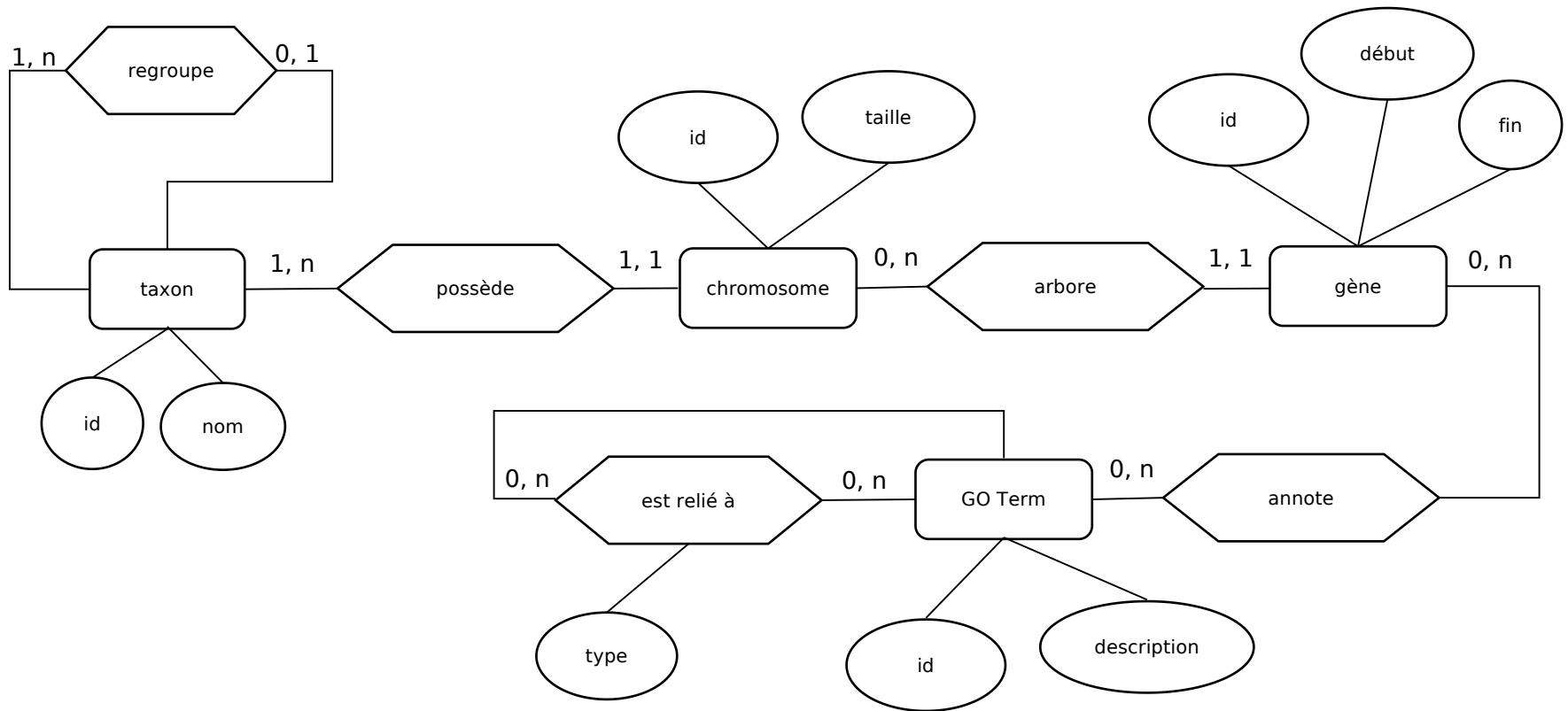
- Cardinalités

- 1:1 (ex: 1 écotype a un 1 génotype)
- 1:n (ex: plusieurs photos d'un plantule d'arabette)
- n:m (ex: plusieurs étudiants analysent plusieurs photos)



- Un gène appartient à un et un seul organisme
- Un organisme peut avoir 0 à plusieurs gènes

# Exemple de schéma entités-associations



- Un taxon (ex: bactérie) peut regrouper 1 à plusieurs autre taxon (ex: acidobactéries, actinobactéries, ...)
- Un taxon appartient à un groupe taxonomique
- Un taxon possède 1 à plusieurs chromosomes
- Un chromosome appartient à un et un seul taxon
- Un chromosome peut arborer des gènes
- Un gène est porté par un et un seul chromosome
- ...

- Une relation = une table
  - colonnes : attributs
  - lignes : entités ou associations
  - similaire à une feuille de calcul mais en plus propre
- Emergence avec le standard/norme en 1992 et le langage SQL puis adoption généralisée
- Quelques systèmes répandus
  - Propriétaires
    - Oracle, SQLServer (microsoft), Access (fichier)
  - Libres et Open source
    - MySQL, PostgreSQL, SQLite (fichier)
- Principaux atouts du schéma relationnel
  - indépendance schéma logique - stockage physique
  - efficacité des opérations grâce à l'algèbre relationnel (manipulation d'ensembles)



# Passage du modèle Entités-Associations au modèle relationnel

---

- Entité → table
  - colonnes = attributs
  - lignes = individus
- Association → table (cas général)
  - colonnes = identifiant des entités reliées + attributs
  - lignes = associations entre 2 entités
  - cas particulier des associations 1:n
    - l'identifiant référencé peut être placée dans la table qui fait référence car l'association est unique
  - cas particulier des association 1:1
    - les 2 entités peuvent être placées dans la même table
- Besoin d'identifier un individu dans une table → clé primaire
  - clé primaire : attribut ou ensemble d'attributs (peut inclure toute les colonnes)
  - ex : nom + prénom + date de naissance ; ou n° de sécu
  - contrainte d'intégrité : la valeur d'une clé primaire d'une ligne doit être unique parmi toutes les lignes de la table
- Clé étrangère : attribut ou ensemble d'attributs d'une table permettant de référencer les lignes d'une autre table

# Optimisation / Normalisation d'un schéma de base de données


---

- Objectif : empêcher ou réduire la redondance
  - facilite l'intégrité des données
  - facilite les mises à jour
- 1ère forme normale (1NF) : tous les attributs sont atomiques (une seule valeur par attribut)
  - ex: étudiants(INE, UEs) → étudiants(INE, UE)
- 2ème forme normale (2NF) : en 1NF + tout attribut non clé ne dépend pas d'une partie de la clé
 

ex: commande(date, n°client, n°produit, quantité, prix\_unitaire)

en 2NF:


commande(date, n°client, n°produit, quantité)   produit(n°produit, prix\_unitaire)

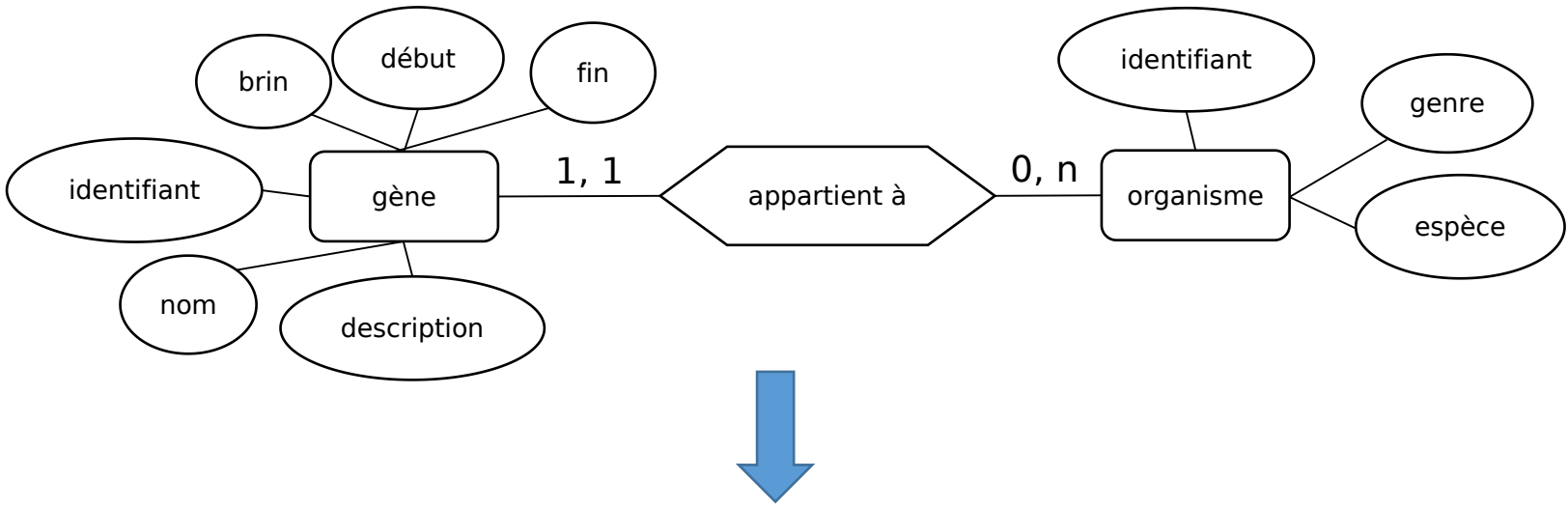

- 3ème forme normale (3NF) : en 2NF + tout attribut n'appartenant pas à la clé ne dépend pas d'un attribut non clé
 

ex: étudiant(ine, nom, prénom, code\_postal, ville)

en 3NF:

étudiant(ine, nom, prénom, code\_postal)   ville(code\_postal, ville)





gène

<u>identifiant</u>	nom	description	brin	début	fin	<u>organisme</u>
id0001	polA	polymerase	+	4 046 966	4 049 752	83333
id0002	recA	recombination	-	2 823 769	2 822 709	83333
id0003	polA	DNA polymerase	+	2 973 182	2 975 824	224308

organisme

<u>identifiant</u>	genre	espèce	strain
83333	Escherichia	coli	K12 MG1655
224308	Bacillus	subtilis	subsp. subtilis str. 168

- SQL : Structured Query Language
- Permet de récupérer des données spécifiques
  - certaines colonnes (SELECT)
  - de certaines tables (FROM)
  - certaines lignes (WHERE)
  - de regrouper certaines lignes (GROUP BY)
  - de filter après le regroupement (HAVING)
  - de trier les résultats (ORDER BY)

• Exemple :

<u>identifiant</u>	<u>nom</u>	<u>description</u>	<u>brin</u>	<u>début</u>	<u>fin</u>	<u>organisme</u>
id0001	polA	polymerase	+	4 046 966	4 049 752	83333
id0002	recA	recombination	-	2 823 769	2 822 709	83333
id0003	polA	DNA polymerase	+	2 973 182	2 975 824	224308

SELECT nom, description  
FROM gene

- Tous les gènes de l'organisme 83333

```
SELECT *  
FROM gene  
WHERE organisme = 83333
```

- Tous les gènes de l'organisme 83333 triés par position de début

```
SELECT *  
FROM gene  
WHERE organisme = 83333  
ORDER BY début
```

- Tous les gènes de l'organisme 83333, sur le brin +, faisant au moins 1000bp et triés par position de début

```
SELECT *  
FROM gene  
WHERE organisme = 83333 AND brin='+'  
      AND fin - début >= 1000  
ORDER BY début
```

# Exemples de requêtes SQL

- Notion de jointure entre les tables

- Tous les gènes dont l'organisme est du genre *Bacillus*

```
SELECT gene.*
```

```
FROM gene JOIN organisme ON (gene.organisme = organisme.identifiant)
```

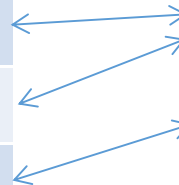
```
WHERE genre='Bacillus'
```

gène

identifiant	nom	description	brin	début	fin	organisme
id0001	polA	polymerase	+	4 046 966	4 049 752	83333
id0002	recA	recombinati on	-	2 823 769	2 822 709	83333
id0003	polA	DNA polymerase	+	2 973 182	2 975 824	224308

organisme

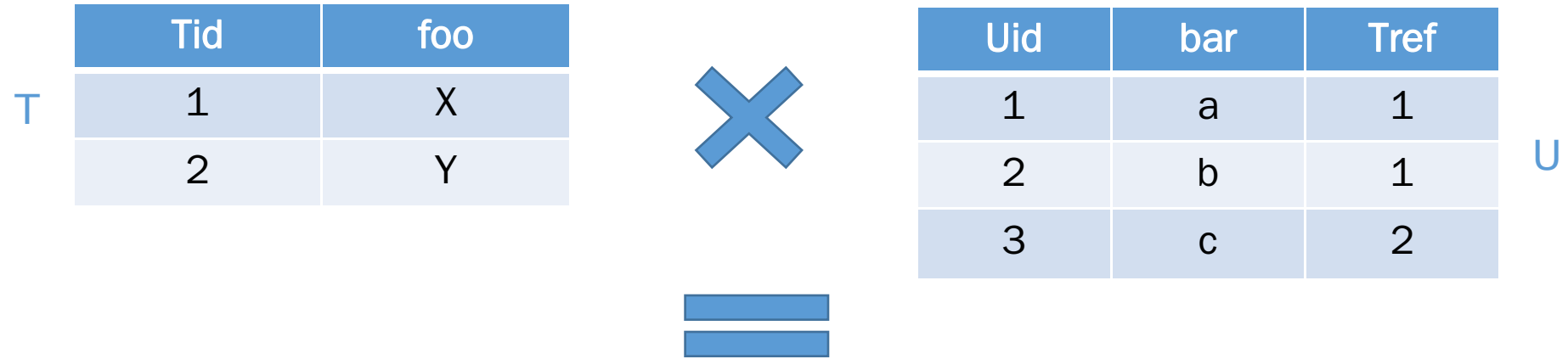
identifiant	genre	espèce	strain
83333	Escheric hia	coli	K12 MG1655
224308	<b>Bacillus</b>	subtilis	subsp. subtilis str. 168



# Jointure entre 2 tables

- produit cartésien

```
SELECT *
FROM T, U
```



T x U

T.Tid	T.foo	U.Uid	U.bar	U.Tref
1	X	1	a	1
1	X	2	b	1
1	X	3	c	2
2	Y	1	a	1
2	Y	2	b	1
2	Y	3	c	2



# Jointure entre 2 tables

- produit cartésien (FROM) + filtrage des lignes (WHERE)


```
SELECT *
```

```
FROM T, U
```

```
WHERE T.Tid = U.Tref
```

T


Tid	foo
1	X
2	Y

  
Tid = Tref

U

Uid	bar	Tref
1	a	1
2	b	1
3	c	2

=

T  U  
Tid = Tref

T.Tid	T.foo	U.Uid	U.bar	U.Tref
1	X	1	a	1
1	X	2	b	1
2	Y	3	c	2

# Jointure entre 2 tables


## • opération de jointure JOIN

SELECT \*

FROM T JOIN U ON (T.id = U.Tref)

T


Tid	foo
1	X
2	Y

  
Tid = Tref

U

Uid	bar	Tref
1	a	1
2	b	1
3	c	2

=

T  U  
Tid = Tref

T.Tid	T.foo	U.Uid	U.bar	U.Tref
1	X	1	a	1
1	X	2	b	1
2	Y	3	c	2