



INTRODUCTION AU LANGUAGE SQL

SQL, C'EST QUOI?

- Langage lié aux bases de données
- MySQL, SQLite

```

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.19 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use elevage
Database changed
mysql> select * from adoption;

```

client_id	animal_id	date_reservation	date_adoption	prix	paye
1	8	2020-05-03	NULL	735.00	1
1	39	2008-08-17	2008-08-17	735.00	1
1	40	2008-08-17	2008-08-17	735.00	1
2	3	2011-03-12	2011-03-12	835.00	1
2	18	2008-06-04	2008-06-04	485.00	1
3	27	2009-11-17	2009-11-17	200.00	1
4	26	2007-02-21	2007-02-21	485.00	1
4	41	2007-02-21	2007-02-21	835.00	1
5	21	2009-03-08	2009-03-08	200.00	1
6	16	2010-01-27	2010-01-27	200.00	1
7	5	2011-04-05	2011-04-05	150.00	1
8	42	2008-08-16	2008-08-16	735.00	1
9	33	2007-02-11	2007-02-11	835.00	1
9	54	2011-02-13	2011-02-13	140.00	1
9	55	2011-02-13	2011-02-13	140.00	1
10	49	2010-08-17	2010-08-17	140.00	0
11	32	2008-08-17	2010-03-09	140.00	1
11	62	2011-03-01	2011-03-01	630.00	0
12	15	2020-05-05	NULL	200.00	1
13	57	2012-01-10	2012-01-10	700.00	1
14	58	2012-02-25	2012-02-25	700.00	1
15	30	2008-08-17	2008-08-17	735.00	1

22 rows in set (0.03 sec)

Base de données

Requête Historique

```

1 select *
2 from adoption ;

```

Table Formulaire

Nombre de lignes chargées : 20

client_id	animal_id	date_reserv	date_adopt	prix	paye
2	3	2011-03-12	2011-03-12	835	1
7	5	2011-04-05	2011-04-05	150	1
16	16	2010-01-27	2010-01-27	200	1
2	18	2008-06-04	2008-06-04	485	1
5	21	2009-03-08	2009-03-08	200	1
4	26	2007-02-21	2007-02-21	485	1
3	27	2009-11-17	2009-11-17	200	1
15	30	2008-08-17	2008-08-27	735	1
11	32	2008-08-17	2010-03-09	140	1
9	33	2007-02-11	2007-02-11	835	1
1	39	2008-08-17	2008-08-17	735	1
1	40	2008-08-17	2008-08-17	735	1
4	41	2007-02-21	2007-02-21	835	1
8	42	2008-08-16	2008-08-16	735	1
10	49	2010-08-07	2010-08-17	140	1
9	54	2011-02-13	2011-02-13	140	1
9	55	2011-02-13	2011-02-13	140	1
13	57	2012-01-10	2012-01-10	700	1
14	58	2012-02-25	2012-02-25	700	1
12	62	2011-03-01	2011-03-01	630	1



LA SYNTAXE DANS SQL

- Mot clé
- Généralement, les mots clés sont écrit en majuscule
- Une requête se termine par un « ; »

	Requête	Historique
1	<u>SELECT</u> *	
2	<u>FROM</u> adoption ;	
3		
4		
5		

UNE BASE DE DONNÉES?

Animal

Adoption

Client

Espèce

Race

CRÉER SA BDD

- Les types numériques : INTEGER
- Les nombres décimaux : DECIMAL
- Les chaînes de caractères : CHAR et VARCHAR
- Le format texte : TEXT
- Les types temporels : DATE, TIME, DATETIME, YEAR, TIMESTAMP

```
CREATE DATABASE nom_base;
```

```
1 CREATE TABLE [IF NOT EXISTS] Nom_table (  
2     colonne1 description_colonne1,  
3     [colonne2 description_colonne2,  
4     colonne3 description_colonne3,  
5     ...,]  
6     [PRIMARY KEY (colonne_clé_primaire)]  
7 )
```

```
1 ALTER TABLE nom_table ADD ... -- permet d'ajouter quelque chose (une colonne par exemple)  
2  
3 ALTER TABLE nom_table DROP ... -- permet de retirer quelque chose  
4  
5 ALTER TABLE nom_table CHANGE ...  
6 ALTER TABLE nom_table MODIFY ... -- permettent de modifier une colonne
```


LE MOT CLÉ SELECT

- Permet de ... sélectionner
- Permet l'affichage de résultats en sélectionnant une partie des données

```
1 SELECT colonne1, colonne2, ...  
2 FROM nom_table;
```

```
1 SELECT *  
2 FROM Animal;
```

LES CLAUSES

WHERE = permet de cibler une sélection

```
1 SELECT *
2 FROM Animal
3 WHERE espece='chien';
```

ORDER BY = Trier la sélection

```
1 SELECT *
2 FROM Animal
3 WHERE espece='chien'
4     AND nom IS NOT NULL
5 ORDER BY nom DESC;
```

HAVING = Cibler spécifiquement des données

```
1 select *
2     from espece
3     where prix > 100
4     having prix < 500
5
```

LIKE = Cherche un String particulier

```
1 select *
2     from espece
3     where description like '%saute%'
4
```

IN = Création de
set de critère

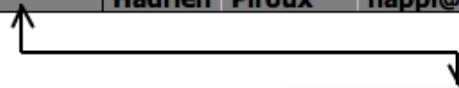
```
1 select *
2     from race
3     where nom in ('Sphynx', 'Berger allemand', 'Maine Coon')
4
```

AJOUTER UNE CLÉ DANS UNE BDD

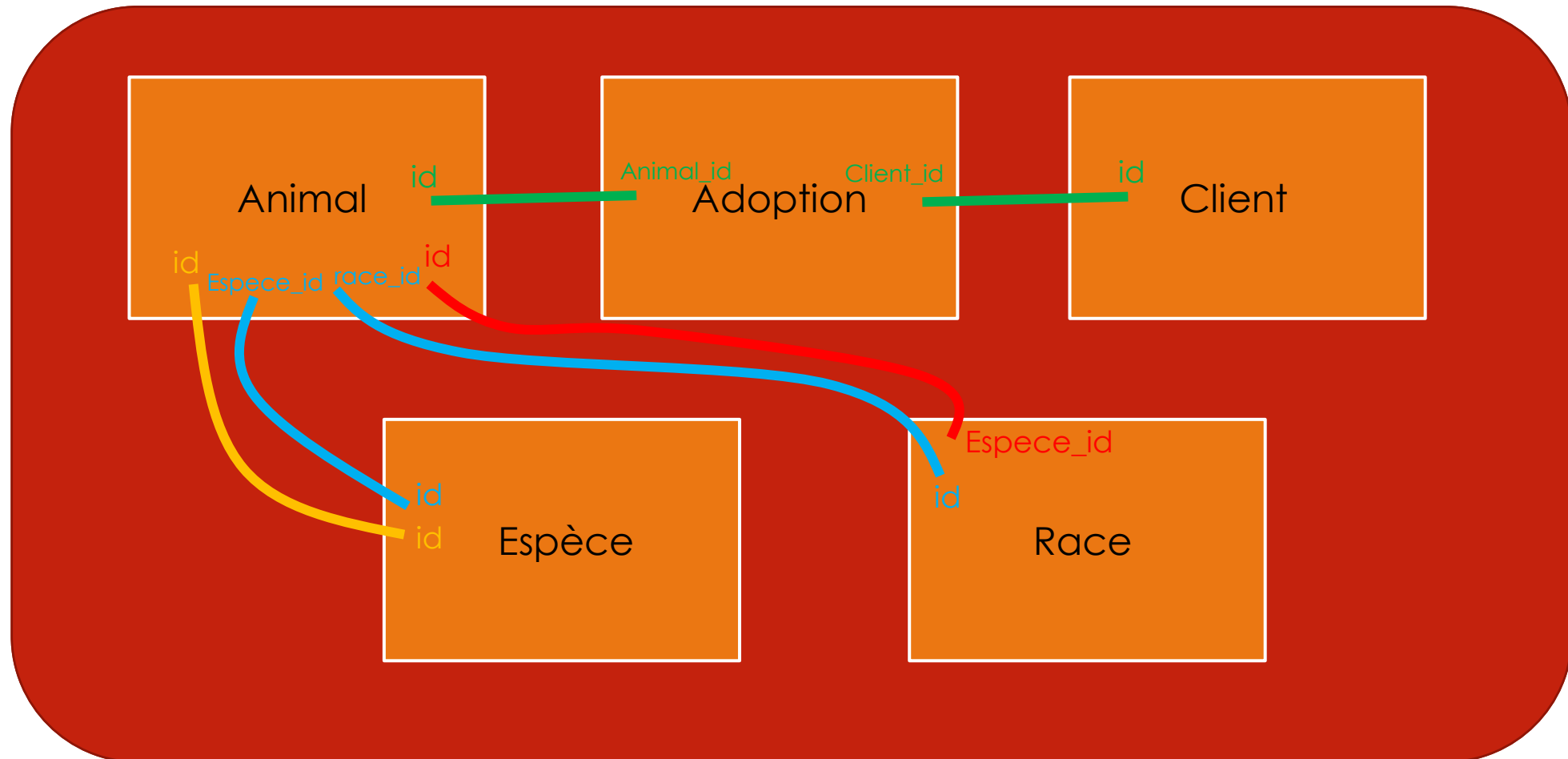
- Clé primaire = la donnée qui permet d'identifier de manière unique un enregistrement dans une table (contrainte d'unicité) => exemple du client
- Clé étrangère = contrainte garantissant l'intégrité référentielle de la BDD (en gros, on sait que pour chaque info liée de la table A à la table B, l'info existe dans la table B). On évite les bêtises :D

Numéro	Nom	Prénom	Email
1	Jean	Dupont	jdupont@email.com
2	Marie	Malherbe	mama@email.com
3	Nicolas	Jacques	Jacques.nicolas@email.com
4	Hadrien	Piroux	happi@email.com

Numéro	Client	Produit	Quantité
1	3	Tube de colle	3
2	2	Rame de papier A4	6
3	2	Ciseaux	2



ET DONC?



LES JOINTURES

Animal

<i>id</i>	<i>sexe</i>	<i>nom</i>	<i>race_id</i>	<i>espece_id</i>
24	M	Cartouche	NULL	1
25	M	Zambo	1	1
33	M	Caribou	4	2

Espece

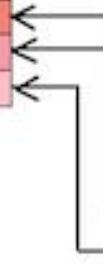
<i>id</i>	<i>nom_courant</i>	<i>nom_latin</i>
1	Chien	Canis canis
2	Chat	Felix silvestris

Animal

<i>id</i>	<i>sexe</i>	<i>nom</i>	<i>race_id</i>	<i>espece_id</i>
24	M	Cartouche	NULL	1
25	M	Zambo	1	1
33	M	Caribou	4	2

Espece

<i>id</i>	<i>nom_courant</i>	<i>nom_latin</i>
1	Chien	Canis canis
2	Chat	Felix silvestris



<i>id</i>	<i>sexe</i>	<i>nom</i>	<i>race_id</i>	<i>espece_id</i>	<i>id</i>	<i>nom_courant</i>	<i>nom_latin</i>
24	M	Cartouche	NULL	1	1	Chien	Canis canis
25	M	Zambo	1	1	1	Chien	Canis canis
33	M	Caribou	4	2	2	Chat	Felix silvestris

L'ÉCRITURE D'UNE JOINTURE


```
1 SELECT e.id,  
2       e.description,  
3       a.nom  
4 FROM Espece AS e      -- On donne l'alias "e" à Espece  
5 INNER JOIN Animal AS a -- et l'alias "a" à Animal.  
6     ON e.id = a.espece_id  
7 WHERE a.nom LIKE 'Ch%';
```

```
1 SELECT Espece.id AS id_espece,  
2       Espece.description AS description_espece,  
3       Animal.nom AS nom_bestiole  
4 FROM Espece  
5 INNER JOIN Animal  
6     ON Espece.id = Animal.espece_id  
7 WHERE Animal.nom LIKE 'Ch%';
```





Id_espece	Description_espece	Nom_bestiole
24	XXXXXXXXXXXXX	Cartouche
25	XXXXXXXXXXXXX	Zambo

EXEMPLE DE JOINTURES

- Les jointures internes 
 - C'est très simple! Tu exiges d'avoir des données de part et d'autres de la jointure moyennant la condition des clés étrangères!

```
1 select race.nom as race, animal.nom as nom
2   from animal
3   inner join race on animal.race_id = race.id
4   where animal.espece_id = 2
5
```

- Les jointures externes gauche 
 - Une jointure à gauche signifie que l'on prend TOUTES les lignes de la table de gauche même s'il n'y a pas de correspondance à droite

- Les jointures externes droite 
 - Ici, ce sont les colonnes de la table à droite qui sont toutes conservées même s'il n'y a pas de correspondances à gauche

nom_animal	race
Choupi	NULL
Callune	Bleu russe
Caribou	Bleu russe
Cawette	Bleu russe
Capou	Maine coon
Cracotte	Maine coon



LES FONCTIONS SCALAIRES

- Fonctions permettant la manipulation de données et faire du calcul
- ROUND() = arrondir un chiffre
- POW() = exposant
- SQRT() = racine carrée
- SUBSTR() = coupe une chaîne de caractère
- REPLACE() = Remplacement d'élément dans un string
-

Code : SQL

```
SELECT nom, prix, ROUND(prix)
FROM Race;
```

nom	prix	ROUND(prix)
Berger allemand	485.00	485
Berger blanc suisse	935.00	935
Singapura	985.00	985
Bleu russe	835.00	835
Maine coon	735.00	735
Sphynx	1235.00	1235
Nebelung	985.00	985

LES FONCTIONS D'AGRÉGATIONS

- Fonctions permettant d'agréger des données. Sert surtout aux stats
- COUNT() = comptage d'éléments
- MIN() / MAX() = minimum et maximum
- SUM() = somme
- AVG() = moyenne

Code : SQL

```
SELECT MIN(prix)  
FROM Race;
```

MIN(prix)
485.00

PLUS D'INFOS?

- <https://openclassrooms.com/fr/courses/4449026-initiez-vous-a-lalgebre-relationnelle-avec-le-langage-sql>
- <https://openclassrooms.com/fr/courses/6971126-implementez-vos-bases-de-donnees-relationnelles-avec-sql/>
- <https://www.sololearn.com/learning/1060>



MERCI POUR VOTRE ATTENTION 😊

