**Les types de données**

|  |  |
| --- | --- |
| TINYINT  SMALLINT  MEDIUMINT  INT  BIGINT | -128 127  -32768 32767  -8388608 8388608  -2147483648 2147483648  -9223372036854775808 9223372036854775808 |
| UNSIGNED TINYINT  UNSIGNED SMALLINT  UNSIGNED MEDIUMINT  UNSIGNED INT  UNSIGNED BIGINT | 0 255  0 65535  0 16777215  0 4294967295  0 1.846744e+19 |
| INT(4) ZEROFILL | Ajoute des 0 si la taille du int ne contient pas au minimum 4 chiffre (pour 45, ça donnera 0045) |
| DECIMAL(5,3) ou NUMERIC(5,3) | Stockage en forme de chaine de caractère de 5 chiffre maximum, dont de 3 après la virgule (arrondissement automatique si trop de chiffre après la virgule) |
| FLOAT ou DOUBLE | Nombre décimal sans paramètre |
| CHAR(x) | A prendre **si on connais** exactement le nombre de caractère qu'il doit s'y trouver |
| VARCHAR(x) | A prendre **si on** **ne connais pas** exactement le nombre de caractère qu'il doit s'y trouver |
| TINYTEXT  TEXT  MEDIUMTEXT  LONGTEXT | 2^8 octets  2^16 octets  2^24 octets  2^32 octets |
| espece ENUM('chat', 'chien', 'tortue')  espece SET('chat', 'chien', 'tortue') | Seul chat chien ou tortue peuvent être entré, sinon stockera une chaine vide par défault  Comme ENUM mais peut stocket plusieurs en même temps (chat, chien – chat, tortue – chien, chat, tortue…) |
| DATETIME | AAAA-MM-JJ HH:MM:SS |
| DATE | AAAA-MM-JJ |
| TIME | HH:MM:SS |
| YEAR | Seulement l'année (de 1901 a 2155) |

**Création de base de données**

|  |  |
| --- | --- |
| CREATE DATABASE nom\_base SET 'utf8'; | Cree la base de données du nom nom\_base d'encodage utf8 |
| DROP DATABASE nom\_base; | Supprime la base nom\_base |
| DROP DATABASE IF EXISTS nom\_base; | Supprime la base nom\_base à condition qu'elle existe |
| USE nom\_base | A partir de cette ligne, les commandes entrée seront pour la base nom\_base |

**Création de tables**

|  |  |
| --- | --- |
| CREATE TABLE IF NOT EXISTS Nom\_table (  colonne1 description\_colonne1,  colonne2 description\_colonne2,  colonne3 description\_colonne3,  [PRIMARY KEY (colonne\_clé\_primaire)]  )  [ENGINE= INNODB]; | Cree table Nom\_table si elle n'existe pas  Nom colonne1 type (id SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,)  Nom colonne2 type (espece VARCHAR(40) NOT NULL,)  Nom colonne3 type (sexe CHAR(1),)  Clé primaire (PRIMARY KEY (id)) |
| DROP TABLE Nom\_table; | Supprime la table Nom\_table |

**MODIFICATION D'UNE TABLE**

|  |  |
| --- | --- |
| ALTER TABLE nom\_table  ADD COLUMN nom\_colonne description\_colonne; | Dans la table nom\_table  Ajoute colonne nom\_colonne |
| ALTER TABLE Commande  ADD CONSTRAINT fk\_client\_numero FOREIGN KEY (client) REFERENCES Client(numero); | Dans la table Commande  Ajoute nom\_foreign\_key la foreign key client qui fait reference a la table client(numero) |
| ALTER TABLE nom\_table  DROP COLUMN nom\_colonne; | Dans la table nom\_table  supprime colonne nom\_colonne |
| ALTER TABLE nom\_table  CHANGE ancien\_nom nouveau\_nom description\_colonne; | Dans la table nom\_table  Change nom de colonne par un autre (CHANGE nom prenom VARCHAR(10) NOT NULL;) |
| ALTER TABLE nom\_table  MODIFY nom\_colonne nouvelle\_description; | Dans la table nom\_table  Modifie la description (MODIFY nom VARCHAR(30) NOT NULL DEFAULT 'Blabla') |
|  |  |

**Insertion de données**

|  |  |
| --- | --- |
| INSERT INTO Animal (espece, sexe, date\_naissance)    VALUES ('tortue', 'F', '2009-08-03 05:12:00'); | Insert dans Animal et colonne espece, sexe, date\_naissance  Les valeurs respectivement 'tortue', 'F', '2009-08-03 05:12:00' |
| INSERT INTO Animal (espece, sexe, date\_naissance, nom)  VALUES ('chien', 'F', '2008-12-06 05:18:00', 'Caroline'),  ('chat', 'M', '2008-09-11 15:38:00', 'Bagherra'),  ('tortue', NULL, '2010-08-23 05:18:00', NULL); | Insertion multiple |
| ~~INSERT INTO Animal~~  ~~SET nom='Bobo', espece='chien', sexe='M', date\_naissance='2010-07-21 15:41:00';~~ | ~~Autre façon d'insérer des donnée (propre à SQL et interdit l'insertion multiple)~~ |
|  |  |

**SÉLECTION DE DONNÉES**

|  |  |
| --- | --- |
| SELECT nom\_colonne  FROM Animal | Selectionne nom\_colonne  Dans la table nom\_table |
| UNION | Relie 2 select (à mettre à la fin du premier et avant la seconde) |
| SELECT DISTINCT espece  FROM Animal; | Selectionne sans les doublons (donnera une seule fois le mot chien si il y en a plusieur par exemple) |
| WHERE espece='chien'; | Condition espèce = chien |
| WHERE espece LIKE c%  WHERE espece LIKE \_  WHERE espece LIKE BINARY '%Lu%';  WHERE espece NOT LIKE %a% | Condition espèce contient c… (n'importe quel chaine après le c  Condition espèce contient 1seule caractère (P\_rl\_ pourrais etre egale a Parle – Perle – Perla)  La majuscule est obligatoire  Espece ne contient pas la lettre a |
| WHERE date\_naissance BETWEEN '2008-01-05' AND '2009-03-23'; | Date de naissance compris entre 2008-01-05 et 2009-03-23 |
| WHERE nom IN ('Moka', 'Bilba', 'Tortilla', 'Balou', 'Dana', 'Redbul', 'Gingko'); | Condition nom = Moka ou Bilba ou Tortilla… |
| **Comparaisons where :**  =  <  <=  >  >=  <> ou !=  <=> | **Comparaisons where :**  Égal  Inférieur  Inférieur ou égal  Supérieur  Supérieur ou égal  Différent  égal (valable pour NULL aussi) |
| **Plusieurs valeurs dans where**  AND  OR  XOR  NOT | **Plusieurs valeurs dans where**  ET  OU  OU exclusif  NON |
| **APRES LE WHERE SEULEMENT**  ORDER BY nom\_colonne  ORDER BY nom\_colonne DESC  ORDER BY nom\_colonne, nom\_colonne2; | **APRES LE WHERE SEULEMENT**  Trie par nom\_colonne croissant  Trie par nom\_colonne décroissant  Trie par nom\_colonne et ensuite par nom\_colonne2 |
| LIMIT 6 OFFSET 3;.. | Nombre de ligne selectionner = 3 a partir de la ligne 3 |

**SUPPRESSION ET MODIFICATION DE DONNÉES**

|  |  |
| --- | --- |
| DELETE FROM nom\_table  WHERE critères; | Supprime dans nom\_table  Les éléments dont la condition est… |
| UPDATE Animal  SET sexe='F', nom='Pataude'  WHERE id=21; | Modifie dans la classe Animal  Sexe = F et nom = Pataude  A condition que l'ID soit 21 |

**Jointures**

|  |  |
| --- | --- |
| **APRES FROM SEULEMENT**  INNER JOIN Animal  ON Espece.id = Animal.espece\_id | **APRES FROM SEULEMENT**  Joins avec la table Animal  Ou l'id d'espece = l'id d'Animal |
| WHERE espece.id = Animal.espece\_id | Equivalent a inner join avec la condition where |

**Sous requête**

|  |  |
| --- | --- |
| SELECT MIN(date\_naissance)  FROM (  SELECT Animal.id, Animal.nom  FROM Animal  INNER JOIN Espece  ON Espece.id = Animal.espece\_id  WHERE sexe = 'F'  AND Espece.nom\_courant IN ('Tortue d''Hermann', 'Perroquet amazone')  ) AS tortues\_perroquets\_F; | Sélectionne la date de naissance la plus basse  Dans… (sous-requête dans from)  que l'on appellera Tortues\_perroquer\_F |
| SELECT id, nom, espece\_id  FROM Race  WHERE espece\_id < (  SELECT id  FROM Espece  WHERE nom\_courant = 'Tortue d''Hermann'); | Sélectionne id, nom, espèce  Dans la classe Race  Condition l'id d'espèce est égale a (sous requête) |