SELECT 2+2;

SELECT true, false, TRUE, FALSE, True, False, 1, 0;

SELECT (TRUE AND FALSE);

SELECT (TRUE OR FALSE);

SELECT IF(TRUE, 'ok', 'pas ok');

SELECT IFNULL(NULL, 'ok');

SELECT IFNULL('coucou, suis-je nul(le) ?', 'ok');

SELECT IF (1 IS TRUE, 'ok', 'pas ok');

SELECT IF (0 IS TRUE, 'ok', 'pas ok');

SELECT IF (1 IS NOT TRUE, 'ok', 'pas ok');

SELECT IF (0 IS NOT TRUE, 'ok', 'pas ok');

SELECT \* FROM pays;

SELECT nom\_pays FROM pays;

SELECT champ1, champ2 FROM ma\_table;

SELECT nom\_pays FROM pays WHERE code\_pays = 'FR';

SELECT prenom FROM stagiaires WHERE abonnement\_newsletter = true;

SELECT prenom FROM stagiaires WHERE abonnement\_newsletter = 1;

SELECT prenom FROM stagiaires WHERE abonnement\_newsletter = '1';

SELECT prenom, nom FROM stagiaires ORDER BY nom;

SELECT prenom, nom FROM stagiaires ORDER BY nom, prenom;

SELECT prenom, nom FROM stagiaires WHERE abonnement\_newsletter = true ORDER BY nom;

SELECT count(\*) FROM stagiaires;

SELECT nom, (2+2) FROM stagiaires;

SELECT nom as name, prenom as firstname, date\_naissance as Birthdate FROM stagiaires;

SELECT nom as 'NOM DU STAGIAIRE', prenom as 'PRENOM DU STAGIAIRE', date\_naissance as 'DATE DE NAISSANCE' FROM stagiaires;

SELECT nom, prenom, YEAR(date\_naissance) FROM stagiaires;

SELECT nom as Nom, prenom as Prénom, YEAR(date\_naissance) as 'Année de naissance' FROM stagiaires;

SELECT nom FROM stagiaires WHERE civilite = 'M.' OR civilite = 'Mlle';

DELETE FROM stagiaires WHERE identifiant\_stagiaire = 2;

DELETE FROM stagiaires WHERE nom = 'GIAMBIASI';

DELETE FROM pays;

TRUNCATE passions;

DELETE FROM stagiaires WHERE YEAR(date\_naissance)<'1990';

-- afficher les noms, prénoms et l'âge de tous les stagiaires (l'âge sera calculé à partir de leur date de naissance et de la date d'aujourd'hui)

-- -- <https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>

SELECT nom, prenom, DATEDIFF(CURDATE(), date\_naissance)/365 as 'âge' FROM stagiaires;

SELECT nom, prenom, ROUND ( DATEDIFF(CURDATE(), date\_naissance)/365 ) as 'âge' FROM stagiaires;

SELECT nom, prenom, ( YEAR( CURDATE() ) - YEAR( date\_naissance ) ) as 'âge' FROM stagiaires;

SELECT nom, prenom, TIMESTAMPDIFF(YEAR, date\_naissance, CURDATE()) as 'âge' FROM stagiaires;

---

SELECT UPPER(prenom), LOWER(nom) from stagiaires;

SELECT REVERSE(prenom) from stagiaires;

SELECT prenom, LENGTH(prenom) FROM stagiaires;

SELECT prenom, MAX(LENGTH(prenom)) FROM stagiaires;

SELECT CONCAT (prenom, ' ', nom) as 'Nom complet' FROM stagiaires;

<https://dev.mysql.com/doc/refman/8.0/en/string-functions.html>

SELECT LEFT(prenom, 3) FROM stagiaires;

SELECT SUBSTRING(prenom, 1, 3) FROM stagiaires;

SELECT prenom, nom FROM stagiaires WHERE ISNULL(tel);

DROP TABLE pays;

DROP DATABASE nom\_de\_la\_base;

SELECT stagiaires\_identifiant\_stagiaire, AVG(note) FROM NotesFinales GROUP BY stagiaires\_identifiant\_stagiaire;

SELECT stagiaires\_identifiant\_stagiaire, ROUND(AVG(note),2) FROM NotesFinales GROUP BY stagiaires\_identifiant\_stagiaire;

----

-- adapter en fonction

UPDATE nom\_table SET nom\_champ\_a\_modifier = valeur\_a\_modifier

WHERE nom\_champs\_conditionnel = valeur;

UPDATE nom\_table

SET

nom\_champ\_a\_modifier = valeur\_a\_modifier,

second\_nom\_champ\_a\_modifier = autre\_valeur

WHERE nom\_champs\_conditionnel = valeur;

UPDATE stagiaires SET nom = 'DUPOND' WHERE identifiant\_stagiaire = 2;

---

UPDATE stagiaires SET adresse = 'avant l''arbre rouge', adresse\_complement = 'à côté de l\'arbre vert' WHERE prenom = 'Franck' AND nom = 'DUPOND';

---

SELECT \* FROM passions ORDER BY nom\_passion LIMIT 0, 2;

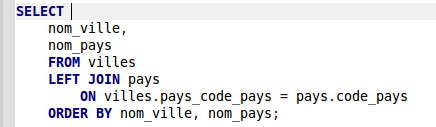
SELECT nom FROM stagiaires WHERE nom LIKE '%am%';

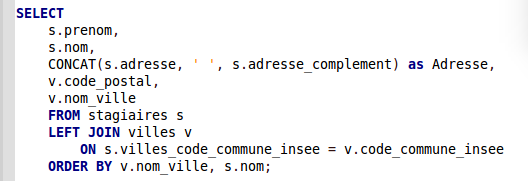
SELECT prenom, nom FROM stagiaires WHERE prenom LIKE '%i%' OR nom LIKE '%i%';

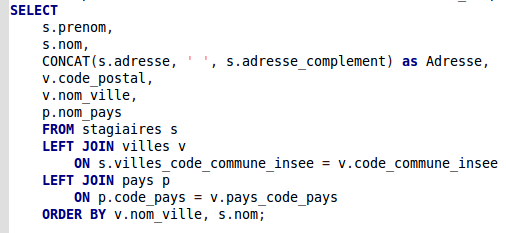
SELECT DISTINCT(villes\_code\_commune\_insee) FROM stagiaires;

SELECT COUNT(DISTINCT(villes\_code\_commune\_insee)) FROM stagiaires;

SELECT nom\_ville, pays\_code\_pays FROM villes ORDER BY nom\_ville, pays\_code\_pays;







SELECT s.prenom, s.nom, CONCAT(s.adresse, ' ', s.adresse\_complement) as Adresse, v.code\_postal, v.nom\_ville, p.nom\_pays FROM stagiaires s LEFT JOIN villes v ON s.villes\_code\_commune\_insee = v.code\_commune\_insee LEFT JOIN pays p ON p.code\_pays = v.pays\_code\_pays ORDER BY v.nom\_ville, s.nom;

SELECT nom, passions.nom\_passion FROM stagiaires LEFT JOIN aime on aime.stagiaires\_identifiant\_stagiaire = stagiaires.identifiant\_stagiaire LEFT JOIN passions ON passions.id\_passion = aime.passions\_id\_passion;

SELECT nom, passions.nom\_passion FROM stagiaires LEFT JOIN aime on aime.stagiaires\_identifiant\_stagiaire = stagiaires.identifiant\_stagiaire INNER JOIN passions ON passions.id\_passion = aime.passions\_id\_passion;

[**http://sql-joins.leopard.in.ua/**](http://sql-joins.leopard.in.ua/)

[**https://drive.google.com/drive/folders/15nU3MbCK4aYk1XbhBtw8gZLuFOAiirSe?usp=sharing**](https://drive.google.com/drive/folders/15nU3MbCK4aYk1XbhBtw8gZLuFOAiirSe?usp=sharing)

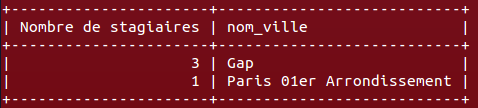
(jointures)

SELECT nom, passions.nom\_passion FROM stagiaires LEFT JOIN aime on aime.stagiaires\_identifiant\_stagiaire = stagiaires.identifiant\_stagiaire LEFT JOIN passions ON passions.id\_passion = aime.passions\_id\_passion WHERE passions.id\_passion = 1;

SELECT l.nom\_langage, s.prenom from stagiaires s left join NotesFinales n ON s.identifiant\_stagiaire = n.stagiaires\_identifiant\_stagiaire left join langage l ON n.langage\_id\_langage = l.id\_langage where n.note > 10;

SELECT s.prenom as Stagiaire, ROUND(AVG(n.note),2) as Moyenne from stagiaires s left join NotesFinales n ON s.identifiant\_stagiaire = n.stagiaires\_identifiant\_stagiaire left join langage l ON n.langage\_id\_langage = l.id\_langage GROUP BY n.stagiaires\_identifiant\_stagiaire;

SELECT l.nom\_langage as Langage, ROUND(AVG(n.note),2) as Moyenne from stagiaires s left join NotesFinales n ON s.identifiant\_stagiaire = n.stagiaires\_identifiant\_stagiaire left join langage l ON n.langage\_id\_langage = l.id\_langage GROUP BY n.langage\_id\_langage;



SELECT count(s.villes\_code\_commune\_insee) as 'Nombre de stagiaires', v.nom\_ville FROM stagiaires s LEFT JOIN villes v ON s.villes\_code\_commune\_insee = v.code\_commune\_insee group by v.nom\_ville;

--

<https://dev.mysql.com/doc/refman/8.0/en/union.html>

ALTER TABLE `villes`

ADD KEY `pays\_code\_pays` (`pays\_code\_pays`);

ALTER TABLE `villes`

ADD CONSTRAINT `villes\_ibfk\_1` FOREIGN KEY (`pays\_code\_pays`) REFERENCES `pays` (`code\_pays`) ON DELETE CASCADE ON UPDATE CASCADE;

[**https://www.sololearn.com/Course/SQL/**](https://www.sololearn.com/Course/SQL/)

[**https://www.w3schools.com/quiztest/quiztest.asp?qtest=SQL**](https://www.w3schools.com/quiztest/quiztest.asp?qtest=SQL)

# 

# Ressources

* [CHEAT SHEET](https://drive.google.com/drive/folders/1LGsd7FkGUzYhzS1gtpzQAomS6KMwHONF?usp=sharing)
* [Compléments cours](https://drive.google.com/drive/folders/1FtJiHXexvGhRSqLVFu9GQdgM7Sp3wC8C?usp=sharing)
* [Jointures](https://drive.google.com/drive/folders/15nU3MbCK4aYk1XbhBtw8gZLuFOAiirSe?usp=sharing)

**Outils clients pour gérer les bases de données**

* <https://www.phpmyadmin.net/>
* <https://www.adminer.org/en/phpmyadmin/>
* [mysql work bench](https://dev.mysql.com/downloads/workbench/) - installable
* <http://dbeaver.jkiss.org/> - installable
* <http://topnew.net/sidu/>
* <http://mywebsql.net/>
* <http://sqlbuddy.com/>
* <http://phpminiadmin.sourceforge.net/>
* <https://github.com/fusonic/chive/>
* <http://www.dbninja.com/>
* <https://teamsql.io/>

**Certifications**

* **Mysql Certification**
  + <https://education.oracle.com/pls/web_prod-plq-dad/ou_product_category.getFamilyPage?p_family_id=406&p_mode=Certification>
  + <https://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=654&get_params=p_id:259&p_org_id=38&lang=F#tabs-1>
* **Google Mobile Web Specialist**
  + <https://developers.google.com/training/certification/mobile-web-specialist/>
  + <https://developers.google.com/training/web/>

**Jeux SQL**

* <http://wwwlgis.informatik.uni-kl.de/extra/game/?lang=en>

# Vues

-- Création d'une vue (uniquement sur les requêtes SELECT)

CREATE VIEW nomdelavue AS requete;

-- utilisation (comme une table normale)

SELECT \* FROM nomdelavue;

-- suppression

DROP VIEW nomdelavue;

-- Exemples

**CREATE VIEW *moyenne* AS** SELECT l.nom\_langage as Langage, ROUND(AVG(n.note),2) as Moyenne from stagiaires s left join NotesFinales n ON s.identifiant\_stagiaire = n.stagiaires\_identifiant\_stagiaire left join langage l ON n.langage\_id\_langage = l.id\_langage GROUP BY n.langage\_id\_langage;

SELECT \* FROM moyenne;

SELECT \* FROM moyenne WHERE Langage = 'PHP';