

TP	TP 7.2	TSI2
	Bases de données : requêtes SQL (tables multiples)	Période 1
		1h

CORRECTION

L'objectif de ce TP est d'écrire des requêtes concernant les écoles et les statistiques de réussite des étudiants. La base de données "**écoles_ingénieurs.sql**" contient toutes les tables nécessaires à l'élaboration de statistiques sur les admissions en écoles.

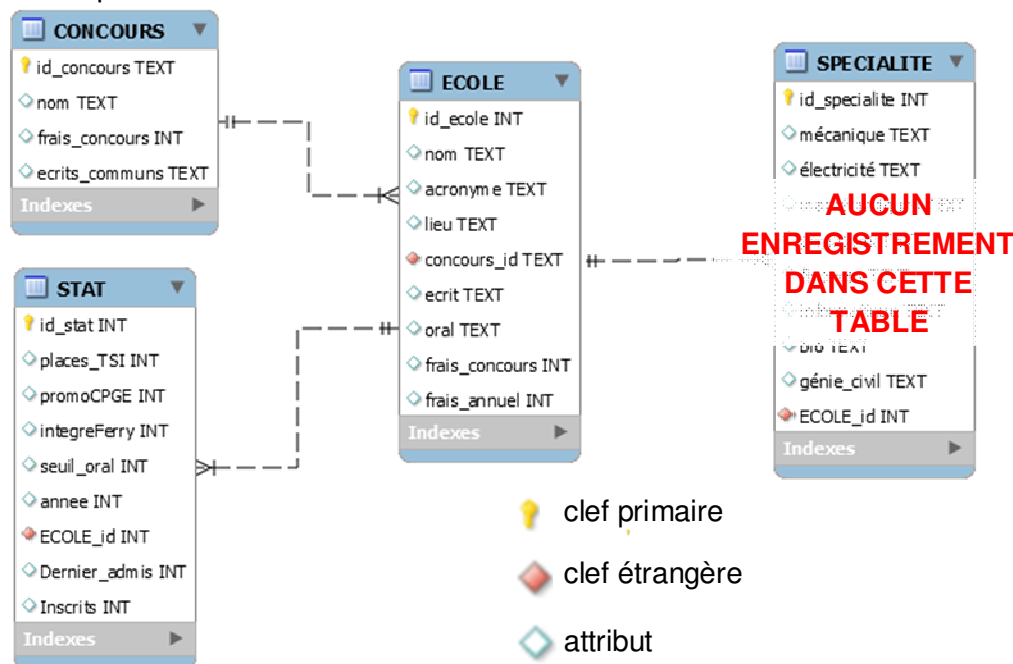


Figure 1 : Schéma de la base de données (logiciel MySQL Worbench)

1 Analyse des tables et de leurs attributs

- 1) Ouvrir le fichier **écoles_ingénieurs.sqlite** et lister les clefs primaires pour chacune des tables.

clefs primaires : id_concours, id_ecole et id_stat

- 2) Lister les clefs étrangères et indiquer les liens qui sont ainsi créés entre les tables. Pour indiquer le comportement de ce lien on indiquera quelle doit être l'occurrence présente dans quelle table afin que l'enregistrement puisse être validé par la clé étrangère; exemple : un enregistrement de la table **ECOLE** n'est valide que si l'occurrence de **concours_id** existe dans la table **CONCOURS** pour l'attribut **id_concours** : CCINP, CCS ou Epita).

Clefs étrangères : concours_id, ecole_id

id_ecole de ECOLE doit exister pour qu'un enregistrement ecole_id dans la table STAT ou SPECIALITE puisse se faire

- 3) Le comportement décrit à la question précédente est le comportement de la table enfant par rapport à la table parente. Quel est le comportement imposé par la clé étrangère sur la table parent vis-à-vis de la table enfant (on s'appuiera sur l'exemple des tables CONCOURS et ECOLE) ?

On ne peut supprimer un concours de la table CONCOURS que si aucun enregistrement dans la table ECOLE ne fait référence à ce concours.

2 Requêtes simples

- 4) Ecrire une requête permettant d'afficher la liste des écoles, leur acronyme (lettres représentatives du nom de l'école) et la ville où elles sont implantées, en triant l'affichage par ordre alphabétique des acronymes.

SELECT nom, acronyme, lieu FROM ecole ORDER BY acronyme

- 5) Ecrire une requête SQL permettant d'afficher la liste acronymes en supprimant les doublons.

SELECT DISTINCT acronyme FROM ecole

- 6) Ecrire une requête SQL pour afficher le nom et l'acronyme des écoles implantées sur Paris.

SELECT nom, acronyme FROM ecole WHERE lieu=="Paris"

3 Jointure symétrique et sélection

Combiner deux tables suivant un attribut commun :

SELECT * FROM t₁ JOIN t₂ ON t1.attribute1 == t2.attribute2 ;

- 7) Afficher les acronymes des écoles intégrées et le nombre d'étudiants du lycée Jules Ferry qui ont intégré en 2022 (les étudiants ont intégré si **integreFerry > 0**).

**SELECT acronyme, integreFerry FROM ecole JOIN STAT ON id_ecole==ECOLE_id
WHERE annee==2022 AND integreferry > 0**

- 8) Afficher la liste des acronymes et des noms des écoles prenant des étudiants en TSI en 2022, en triant le résultat par nombre décroissant des places (**ORDER BY attribut DESC** permet d'obtenir un tri décroissant, **ORDER BY attribut** permet d'obtenir un tri croissant). Pour vérifier que le tri est correct on pourra afficher également la valeur de l'attribut **places_TSI**.

**SELECT acronyme, nom, places_TSI FROM ecole JOIN STAT ON id_ecole==ECOLE_id
WHERE annee==2022 ORDER BY places_TSI**

- 9) Afficher la liste des acronymes des noms d'écoles et des barres d'oral en 2022 par ordre décroissant des barres à l'oral (**WHERE seuil_oral** permet d'éliminer les écoles dont on ne connaît pas les barres d'admission).

**SELECT acronyme, nom, seuil_oral FROM ecole JOIN STAT ON id_ecole==ECOLE_id
WHERE annee==2022 AND seuil_oral ORDER BY seuil_oral DESC**

4 Assemblage

4.1 Union

Relation contenant les attributs appartenant aux tables **t1** ou **t2** (suppression des lignes identiques) :

t1 UNION t2

- 10) En utilisant une UNION, afficher la liste des acronymes des écoles dans lesquels des étudiants du lycée ont intégré en 2022 ou en 2015.

**SELECT acronyme FROM ECOLE JOIN STAT
ON id_ecole==ECOLE_id WHERE integreFerry!=0 and annee==2022
UNION
SELECT acronyme FROM ECOLE JOIN STAT
ON id_ecole==ECOLE_id WHERE integreFerry!=0 and annee==2015 ;**

4.2 Différence

Relation contenant les lignes appartenant à **t1** moins celles appartenant à **t2** : **t1 EXCEPT t2**

- 11) En utilisant une soustraction, afficher la liste des acronymes des écoles dans lesquelles des étudiants ont intégrés en 2022 seulement (en 2022 mais pas dans toutes les autres années sauf 2022).

```
SELECT acronyme FROM ECOLE JOIN STAT  
ON id_ecole==ECOLE_id WHERE annee==2022 and integreFerry!=0  
EXCEPT  
SELECT acronyme FROM ECOLE JOIN STAT  
ON id_ecole==ECOLE_id WHERE integreFerry!=0 and annee!=2022
```

4.3 Intersection

Relation contenant les lignes appartenant à **t1** et à **t2** : **t1 INTERSECT t2**

- 12) En utilisant une intersection, afficher la liste des acronymes des écoles dans lesquelles des étudiants ont intégrés en 2022 et en 2015.

```
SELECT acronyme FROM ECOLE JOIN STAT ON id_ecole==ECOLE_id  
WHERE annee==2014 AND integreFerry!=0  
INTERSECT  
SELECT acronyme FROM ECOLE JOIN STAT ON id_ecole==ECOLE_id  
WHERE annee==2015 AND integreFerry!=0
```

5 Agrégation

Ces fonctions permettent d'effectuer des calculs statistiques basiques sur les valeurs : **MIN()**, **MAX()**, **SUM()** (somme), **AVG()** (moyenne), **COUNT()** (comptage du nombre de lignes).

La clause **GROUP BY** permet de rassembler les enregistrements ayant l'attribut spécifié en commun.

La clause **HAVING** permet d'imposer une condition sur le résultat de l'agrégation (valeur minimum ...)

- 13) Afficher le nombre d'étudiants du lycée Jules Ferry ayant intégré en 2022.

```
SELECT SUM(integreFerry) FROM stat WHERE annee==2022
```

- 14) Afficher l'acronyme, le nom des écoles et la somme **ferry** des étudiants de Jules Ferry qui ont intégré en les triant par ordre décroissant de **ferry**.

```
SELECT acronyme, nom, SUM(integreFerry) as ferry FROM ecole JOIN stat  
ON id_ecole==ECOLE_id GROUP BY acronyme ORDER BY ferry DESC
```

- 15) Afficher l'acronyme et le nom des écoles en les triant par ordre décroissant de la taille moyenne **taille** des promotions (si la promotion de l'année n'est pas nulle).

```
SELECT acronyme, nom, AVG(promoCPGE) as taille  
FROM stat JOIN ecole ON id_ecole==ECOLE_id  
WHERE promoCPGE!=0 GROUP BY acronyme ORDER BY taille DESC
```

- 16) Afficher l'acronyme et le nom des écoles en les triant par ordre croissant du taux **taux** d'étudiants TSI en 2022 (rapport *100 entre le nombre de places TSI et la taille de la promotion).

```
SELECT acronyme, nom, 100*places_TSI/promoCPGE as taux  
FROM ecole JOIN stat ON id_ecole==ECOLE_id  
WHERE annee==2022 and places_TSI!=0  
ORDER BY taux
```