

	TP 6 ET 7: BASE DE DONNÉES ET SI	(AU 2019/2020) PAR MR : I.CHANA
	<ul style="list-style-type: none"> - REQUÊTE SQL SUITE - ACCENT SUR LES REQUÊTES DE MISE AJOUR (UPDATE, DELETE, TRUNCATE) - OPÉRATION SUR LE CHAMPS DE FORMATS DATE, DATETIME - LA FONCTION CONCAT() 	

Important:

- **Le compte Rendu doit être bien fait,**
- **Les screenshot (copie écran) doivent se faire uniquement aux résultats d'exécution alors que les requêtes sql , solutions des questions, doivent être écrites en texte.**

Introduction :

Ce TP contient 2 parties :

- **La première** est une suite de manipulation de langage LMD de SQL en mettant l'accent sur les mise à jour en utilisant les order **update et delete, deuran cette partie nous allons aussi crer des tables en** utilisant l'ordre de création directe (Create table) ou par copie (create ... as select ..) , nous réaliserons aussi des requête d'extraction select pour consolider nos acquis en terme des requêtes d'interrogation
 - **La deuxième** partie est très important, car elle est consacrée à la manipulation des champs et des valeur de type **date (ou date_time)** , par un panel de fonctions qui permettent de réaliser des opération arithmétiques (addition, soustraction) entre deux champs ou deux valeurs de type **date (ou date_time)**, et d'autre fonction qui font la conversion entre les formats de dates ou la conversion d'une unité de temps à l'autre (exemple de Jours au Heures , ou de Année aux Jours etc.) sans oublier aussi les fonctions qui sont consacrées au formatage et affichage des dates selon le format désiré
- Reference de la 2eme partie: <https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>

Soit la table ETUDIANT suivante

Code_Etud	Nom	Prenom	Date_naissance	annee_inscription	Filiere	option	Moyenne	Resultat	mention
12	Saidi	Mohammed	2000-02-10 01:20:10	2017	GI	GL	11,88		
22	Rachidi	Najwa	2001-04-03 15:00:30	2018	GE		13		
55	Igon	Mouna	2002-05-12 13:50:00	2017	GI	TSR	14,5		
66	Waf	Rachid	2001-06-05 14:07:20	2017	GE		6		
77	Kafi	Najwa	1999-12-13 03:00:00	2017	GI	GL	7		
88	Rak	Ahmed	2002-08-23 04:10:30	2018	GE		14,13		
99	Fatih	Noura	2003-09-14 17:23:21	2017	GI	TSR	11,4		
111	Ramchi	Ilyas	2000-02-17 12:20:14	2017	GE		16,6		
122	Skafi	Ali	2000-09-09 05:01:44	2016	GI	TSR	12,88		
133	Nari	Khalil	2002-05-30 09:10:50	2017	TSI		13,8		
144	Kapo	Ilyas	1999-10-25 10:50:12	2018	GE	TSR	5,59		
155	Fouh	Kamilya	2002-11-11 11:54:20	2017	TM		10,5		
166	Richard	Anna	2003-02-27 13:23:40	2016	GE		11,9		
177	Baba	Ilyas	2000-02-23 11:13:33	2017	GI	GL	11,8		

Partie I) Langage de manipulation de données suite : Accent sur les requêtes de mise à jour (Update, delete, Truncate)

- 1) Créer la table **ETUDIANT** , sachant que le type de **date_naissance** est de **DATETIME** et celui de **annee_inscription** est de **YEAR**
- 2) insérer les données correspondantes : Pour les questions 1 et 2 vous utilisez les codes ci-dessous (! n'essayez pas de faire copier coller c'est une image !)

```

create table ETUDIANT (
    Code_Etud int primary key,
    Nom        varchar(15),
    Prenom     varchar(15),
    Date_naissance datetime,
    annee_inscription year,
    Filiere    varchar(15),
    option_F   varchar(15),
    Moyenne    decimal(4,2),
    Resultat   varchar(15),
    mention    varchar(15)
);

insert into ETUDIANT values
(12,'Saidi','Mohammed','2000-02-10 01:20:10','2017','GI','GL',11.88,null,null),
(22,'Rachidi','Najwa','2001-04-03 15:00:30','2018','GE',null,13,null,null),
(55,'Igon','Mouna','2002-05-12 13:50:00','2017','GI','TSR',14.5,null,null),
(66,'Waf','Rachid','2001-06-05 14:07:20','2017','GE',null,6,null,null),
(77,'Kafi','Najwa','1999-12-13 03:00:00','2017','GI','GL',7,null,null),
(88,'Rak','Ahmed','2002-08-23 04:10:30','2018','GE',null,14.13,null,null),
(99,'Fatih','Noura','2003-09-14 17:23:21','2017','GI','TSR',11.4,null,null),
(111,'Ramchi','Ilyas','2000-02-17 12:20:14','2017','GE',null,16.6,null,null),
(122,'Skafi','Ali','2000-09-09 05:01:44','2016','GI','TSR',12.88,null,null),
(133,'Nari','Khalil','2002-05-30 09:10:50','2017','TSI',null,13.8,null,null),
(144,'Kapo','Ilyas','1999-10-25 10:50:12','2018','GE','TSR',5.59,null,null),
(155,'Fouh','Kamilya','2002-11-11 11:54:20','2017','TM',null,10.5,null,null),
(166,'Richard','Anna','2003-02-27 13:23:40','2016','GE',null,11.9,null,null),
(177,'Baba','Ilyas','2000-02-23 11:13:33','2017','GI','GL',11.8,null,null);

```

- 3) Mettre à jour (**Update**) la colonne **MOYENNE** et la colonne **RESULTAT** en donnant 12,0 aux étudiants ayant obtenus une note entre 11.70 et 12.0 et affectant la valeur **Racheté** à la colonne **RESULTAT** correspondante.
- 4) Mettre à jour la colonne **RESULTAT** pour le reste des étudiants (ceux qui ne sont pas rachetés) en affectant les valeurs suivantes :
- '**Validé**' si la moyenne est supérieur ou égale à 12 (Attention à ceux qui sont racheté il ne faut pas leurs changer la valeur de la colonne **RESULTAT**, il faut qu'elle reste toujours égale à **racheté**)
 - '**Non Validé**' si la moyenne est strictement inférieure à 12 et supérieure ou égale à 8
 - '**Exclut**' si la moyenne est strictement inférieure à 8
- 5) Mettre à jour la colonne **MENTION** en affectant les valeurs suivantes :
- '**A Bien**' si la moyenne est entre 12 et 13.99
 - '**Bien** ': Si la moyenne est entre 14 et 15.99
 - '**Tres bien**' si la moyenne est supérieure ou égale à 16
- 6) Supprimer les étudiant de la table **ETUDIANT** qui ont la moyenne strictement inférieure à 8
- 7) Quelle est la moyenne des moyennes des étudiants
- 8) Écrire la requête qui permet d'afficher le nombre des étudiants par filière
- 9) Écrire la requête qui permet d'afficher le nombre des étudiants qui ont validé par année d'inscription
- 10) Écrire la requête qui permet d'afficher les noms et les prénoms des étudiants triés par ordre décroissant de la moyenne
- 11) Écrire la requête qui permet d'afficher le nombre des étudiants qui ont le même nom
- 12) Écrire la requête qui permet d'afficher le plus petit et le plus grand étudiant vous vous basez bien sur la date de naissance
- 13) Créons la table **CP_Etudiant** qui est une copie de la table **ETUDIANT**
- pour se faire on exécute la commande suivante :

```
create table CP_Etudiant as select * from Etudian
```

- 14) Créer la table **Resut_Etudiant** qui ne contient que les colonnes **nom, prenom et resultat** de la table **ETUDIANT**
- 15) S'assurer que les deux tables **CP_Etudiant** et **Resut_Etudiant** sont bien créées?
- 16) Renommer la table **CP_Etudiant** par **COPIE_ET**
- 17) Vider la table **COPIE_ET** (supprimer tous ces enregistrements) en utilisant l'ordre **delete**
- 18) Vider la table **Resut_Etudiant** (supprimer tous ces enregistrements) en utilisant l'ordre **truncate**
- 19) Quelle est la différence entre les deux ordres **delete**, **truncate**
- 20) Supprimer les deux tables **COPIE_ET** et **Resut_Etudiant**

Partie II) Fonctions de manipulation des dates

II.1) Extraction dans un champ **datetime**

Le format d'un champ "DATETIME" dans MySQL est "YYYY-mm-dd hh:ii:ss". Il est possible d'extraire chaque partie des dates avec des fonctions basiques DATE(), TIME(), YEAR()

- 21) Exécuter la requête ci-dessous et interpréter le résultat

```
SELECT date_naissance, DATE(date_naissance), YEAR(date_naissance), MONTH(date_naissance),
DAY(date_naissance), TIME(date_naissance), HOUR(date_naissance), MINUTE(date_naissance),
SECOND(date_naissance) FROM ETUDIANT LIMIT 2;
```

Ces fonctions sont accessibles sans restriction et peuvent être utilisées dans toutes les parties d'une requête SQL: SELECT, ON, WHERE, GROUP BY....

- 22) On peut réaliser d'autres extractions un peu plus complexes avec la fonction EXTRACT() pour des cas plus particuliers.. Exécuter la requête ci-dessous et interpréter le résultat

```
SELECT date_naissance, EXTRACT(YEAR_MONTH FROM date_naissance), EXTRACT(DAY_MINUTE
FROM date_naissance) FROM ETUDIANT LIMIT 2;
```

II.2) Changer le format d'une date

23) La fonction **DATE_FORMAT()**: permet de modifier le format d'une date .

Exécuter la requête ci-dessous et interpréter le résultat

```
SELECT date_naissance, DATE_FORMAT(date_naissance, '%Y %M %d'),  
DATE_FORMAT(date_naissance, '%d-%m-%y %h:%i:%s') FROM ETUDIANT LIMIT 3;
```

24) La fonction **GET_FORMAT()** : fait appel aux formats prédéfinis selon les différentes norme (USA, Europe, ISO), Le premier paramètre de la fonction **GET_FORMAT()** doit être **DATE**, **DATETIME** ou **TIME** et permet de définir quelle partie on veut afficher.

- Exécuter la requête ci-dessous et interpréter le résultat en expliquant la différence entre **GET_FORMAT()** et **DATE_FORMAT()**

```
SELECT date_naissance, DATE_FORMAT(date_naissance, GET_FORMAT(DATE, 'EUR')),  
DATE_FORMAT(date_naissance, GET_FORMAT(DATE, 'USA')), DATE_FORMAT(date_naissance,  
GET_FORMAT(DATETIME, 'ISO')) FROM ETUDIANT LIMIT 2;
```

II.3) les opérations arithmétiques sur une date

On peut exécuter des opérations arithmétique usuel entre nombre par exemple **select 4*20 -3**

Addition : Les opérations arithmétiques sont bien sûr possible. Plusieurs fonctions et notations permettent d'obtenir le même résultat. Si je veux ajouter 5 jours à une date, je peux utiliser les fonctions **ADDDATE()**, **DATE_ADD()** ou tout simplement le signe + avec le mot clé **INTERVAL**:

25) Exécuter la requête ci-dessous et interpréter le résultat.

```
SELECT date_naissance, ADDDATE(date_naissance, 5), ADDDATE(date_naissance, INTERVAL 5 DAY),  
DATE_ADD(date_naissance, INTERVAL 5 DAY), date_naissance + INTERVAL 5 DAY FROM ETUDIANT  
LIMIT 2;
```

26) La première notation de la fonction ADDDATE() ne permet que d'ajouter un nombre de jour. L'utilisation du mot clef "INTERVAL", dans les autres notations, permet de créer des intervalles de temps plus complexes.

- Exécuter la requête ci-dessous et interpréter le résultat.

```
SELECT date_naissance, date_naissance + INTERVAL '1 10' YEAR_MONTH, date_naissance + INTERVAL  
1 MONTH + INTERVAL 10 DAY FROM ETUDIANT LIMIT 1;
```

- Expliquer la différence entre ADDDATE(), DATE_ADD(), et l'utilisation de INTERVAL

Soustraction : On peut aussi faire la même chose pour une soustraction avec les fonctions SUBDATE(), DATE_SUB() ou en spécifiant une valeur négative.

27) Exécuter et interpréter les requêtes ci-dessous et déterminer la différence entre DATE_SUB et SUBDATE

```
SELECT DATE_SUB("2017-06-15 09:34:21", INTERVAL 3 HOUR);
```

```
SELECT DATE_SUB("2017-06-15", INTERVAL -2 MONTH);
```

```
SELECT DATE_SUB("2017-06-15 09:34:21", INTERVAL 15 MINUTE);
```

```
SELECT SUBDATE("2017-06-15", INTERVAL 10 DAY);
```

```
SELECT SUBDATE("2017-06-15 09:34:21", INTERVAL 15 MINUTE);
```

28) Il est possible de calculer la différence entre deux dates ou deux heures avec les fonctions DATEDIFF() et TIMEDIFF(). DATEDIFF() ne renvoie qu'un nombre de jour. Les deux fonctions ne savent travailler que sur leur partie respective.

- Exécuter et interpréter le résultat de la requête suivante

```
SELECT DATEDIFF('2020-02-14 18:33:15', '2019-11-14 20:33:15'), TIMEDIFF('2017-12-31 12:36:00',  
'2017-12-25 23:59:00');
```

NB : la date de premier paramètre doit être supérieure à la date de deuxième paramètre

29) Écrire la requête qui permet d'afficher la différence d'âge entre le plus petit et le plus grand étudiant. Vous vous basez bien sur la date de naissance (se référer à la question 12)

30) Pour obtenir le différentiel dans d'autres unités il faut utiliser la fonction `TIMESTAMPDIFF()`. Attention aux paramètres, l'ordre des deux dates est inversé par rapport aux fonctions `TIMEDIFF()`, `DATEDIFF()` précédentes. En effet on met en premier la date inférieure

- Exécuter et interpréter le résultat de la requête suivante

```
SELECT TIMESTAMPDIFF(MINUTE, '2012-10-25 23:59:00', '2012-12-31 12:36:00');
```

31) Refaire la même requête en affichant la différence en terme de :

- Secondes
- Heures
- Jours
- mois
- années

32) Etant donné que

- la fonction `now()` retourne la date actuelle pour tester, exécuter → `select now()`

- la fonction `Concat()` fait la concaténation des champs pour tester exécuter la requête suivante

```
select concat( 'Le nom complet est :' NOM, ' ', PRENOM ) from ETUDINAT ;
```

Écrire la requête qui affiche le nom et prénom et l'âge de tous les étudiants par rapport à la date d'aujourd'hui en utilisant l'année et les mois **Exemple Saidi Mohammed a 20 Ans**

III.4) Convertir une date

Il existe plusieurs fonctions pour convertir les dates comme :

-`TO_DAYS()` : obtenir le nombre de jour depuis l'an 0

-`TIME_TO_SEC()`: convertir l'heure en nombre de secondes

- `DAYOFYEAR()`: obtenir le jour de l'année

Et si la date est stockée dans un champs de type autre que "DATE", "DATETIME" ou "TIME" il est toujours possible d'utiliser la fonction `CAST()` pour pouvoir utiliser les nombreuses fonctions de manipulation de date.

34) exécuter et interpréter la requête suivante :

```
SELECT CAST('2020-02-22' AS DATETIME);
```

35) Et si la date est sous un format différent, il est dans ce cas possible d'utiliser la fonction `STR_TO_DATE()`, qui permet d'indiquer à MySQL le format de la date passée en paramètre

- Exécuter la requête suivante :

```
SELECT STR_TO_DATE('22/02/2020', '%d/%m/%Y'), CAST('22/02/2020' AS DATE);
```

36) Interpréter et expliquer pourquoi la deuxième colonne correspondante à `CAST('22/02/2020' AS DATE)` a affiché une valeur **null**, déduire l'intérêt de la fonction `STR_TO_DATE()` par rapport à la fonction `CAST()`