

Correction — SQL

Gestion d'une salle de sport

28 questions — DDL · DML · Jointures implicites · Filtres · Dates · Sous-requêtes

Jointure implicite · NOW() · DATEDIFF() · Table ACTIVITES dès Q9

DDL — Questions 1 à 8

ALTER TABLE · ADD COLUMN · RENAME · CONSTRAINT

La table s'appelle encore COURS dans ces questions

Q1 — Ajouter une colonne telephone à MEMBRES

DDL

```
ALTER TABLE MEMBRES
```

```
ADD COLUMN telephone VARCHAR(15);
```

→ *ADD COLUMN ajoute une nouvelle colonne en fin de table. VARCHAR(15) = 15 caractères max.*

Q2 — Ajouter une colonne remise à COURS

DDL

```
ALTER TABLE COURS
```

```
ADD COLUMN remise DECIMAL(5,2) DEFAULT 0.00;
```

→ *DECIMAL(5,2) = 5 chiffres au total dont 2 après la virgule. DEFAULT 0.00 = valeur par défaut.*

Q3 — Modifier la colonne statut (30 caractères)

DDL

```
ALTER TABLE INSCRIPTIONS
```

```
MODIFY COLUMN statut VARCHAR(30) DEFAULT 'en attente';
```

→ *MODIFY COLUMN redéfinit le type d'une colonne existante sans supprimer les données.*

Q4 — Renommer la table COURS en ACTIVITES

DDL

```
RENAME TABLE COURS TO ACTIVITES;
```

→ [A](#) À partir de cette question, la table se nomme ACTIVITES dans toutes les requêtes suivantes.

```
-- Alternative standard SQL :
```

```
ALTER TABLE COURS RENAME TO ACTIVITES;
```

Q5 — Renommer la colonne ville en ville_residence

DDL

```
ALTER TABLE MEMBRES
```

```
RENAME COLUMN ville TO ville_residence;
```

→ *RENAME COLUMN* disponible depuis MySQL 8.0. La colonne s'appelle *ville_residence* dans la suite.

Q6 — Contrainte UNIQUE sur email dans MEMBRES

DDL

```
ALTER TABLE MEMBRES
```

```
ADD CONSTRAINT uq_email UNIQUE (email);
```

→ *UNIQUE* garantit qu'aucun doublon n'est possible sur email. La contrainte est nommée *uq_email*.

Q7 — Contrainte CHECK : prix > 0 dans COURS

DDL

```
ALTER TABLE COURS
```

```
ADD CONSTRAINT chk_prix CHECK (prix > 0);
```

→ CHECK empêche l'insertion de lignes avec $\text{prix} \leq 0$. Supporté depuis MySQL 8.0.16.

Q8 — Contrainte CHECK : nb_seances ≥ 1

DDL

```
ALTER TABLE LIGNES_INSCRIPTION  
ADD CONSTRAINT chk_nb_seances CHECK (nb_seances >= 1);
```

→ Garantit qu'une ligne d'inscription a au moins 1 séance. Empêche les valeurs nulles ou négatives.

SELECT — Questions 9 à 20

Jointure implicite (FROM ... WHERE) · ACTIVITES · NOW() · DATEDIFF()

La table COURS a été renommée ACTIVITES (Q4)

La colonne ville est devenue ville_residence (Q5)

Q9 — Inscriptions confirmées (prénom, nom, date)

SELECT

```
SELECT M.prenom, M.nom, I.date_inscription_cours
FROM MEMBRES M, INSCRIPTIONS I
WHERE I.id_membre = M.id_membre
      AND I.statut = 'confirmée';
```

Q10 — Inscriptions 2024 : membre, cours, séances

SELECT

```
SELECT M.nom, A.nom_cours, LI.nb_seances
FROM MEMBRES M, INSCRIPTIONS I, LIGNES_INSCRIPTION LI, ACTIVITES A
WHERE I.id_membre = M.id_membre
      AND LI.id_inscription = I.id_inscription
      AND A.id_cours = LI.id_cours
      AND YEAR(I.date_inscription_cours) = 2024;
```

Q11 — Cours du membre n°1 (séances, prix)

SELECT

```
SELECT A.nom_cours, LI.nb_seances, LI.prix_unitaire
FROM INSCRIPTIONS I, LIGNES_INSCRIPTION LI, ACTIVITES A
WHERE LI.id_inscription = I.id_inscription
      AND A.id_cours = LI.id_cours
      AND I.id_membre = 1;
```

Q12 — Montant par ligne, inscriptions confirmées

SELECT

```
SELECT M.nom, I.date_inscription_cours,  
       A.nom_cours, LI.nb_seances,  
       (LI.nb_seances * LI.prix_unitaire) AS montant_total  
FROM MEMBRES M, INSCRIPTIONS I, LIGNES_INSCRIPTION LI, ACTIVITES A  
WHERE I.id_membre = M.id_membre  
      AND LI.id_inscription = I.id_inscription  
      AND A.id_cours = LI.id_cours  
      AND I.statut = 'confirmée'  
ORDER BY M.nom DESC, montant_total DESC;
```

Q13 — Membres de Paris ou Lyon (OR puis IN)

SELECT

-- Avec **OR** :

```
SELECT nom, prenom, ville_residence FROM MEMBRES  
WHERE ville_residence = 'Paris' OR ville_residence = 'Lyon';
```

-- Avec **IN** :

```
SELECT nom, prenom, ville_residence FROM MEMBRES  
WHERE ville_residence IN ('Paris', 'Lyon');
```

→ La colonne s'appelle *ville_residence* depuis Q5.

Q14 — Cours Muscu entre 50 et 150 € (table ACTIVITES)

SELECT

```
SELECT nom_cours, categorie, prix
FROM ACTIVITES
WHERE categorie = 'Muscu'
      AND prix BETWEEN 50 AND 150
ORDER BY prix;
```

Q15 — Inscriptions janv–avr 2024 (confirmées + en attente)

SELECT

```
SELECT id_inscription, date_inscription_cours, statut
FROM INSCRIPTIONS
WHERE statut IN ('confirmée', 'en attente')
      AND date_inscription_cours BETWEEN '2024-01-01' AND '2024-04-30'
ORDER BY date_inscription_cours;
```

Q16 — Lignes Muscu/Cardio avec nb_seances > 2

SELECT

```
SELECT A.nom_cours, A.categorie, LI.nb_seances, LI.prix_unitaire
FROM LIGNES_INSCRIPTION LI, ACTIVITES A
WHERE A.id_cours = LI.id_cours
      AND A.categorie IN ('Muscu', 'Cardio')
      AND LI.nb_seances > 2;
```

Q17 — Membres inscrits après 2020 (avec année)

SELECT

```
SELECT nom, prenom, YEAR(date_inscription) AS annee_inscription
FROM MEMBRES
WHERE YEAR(date_inscription) > 2020
ORDER BY date_inscription ASC;
```

Q18 — Inscriptions du mois de mars 2024

SELECT

```
SELECT id_inscription, date_inscription_cours, statut
FROM INSCRIPTIONS
WHERE YEAR(date_inscription_cours) = 2024
      AND MONTH(date_inscription_cours) = 3;
```

→ Alternative : *BETWEEN '2024-03-01' AND '2024-03-31'*.

Q19 — Inscriptions de plus de 30 jours

SELECT

```
SELECT id_inscription, date_inscription_cours, statut  
FROM INSCRIPTIONS  
WHERE DATEDIFF(NOW(), date_inscription_cours) > 30  
ORDER BY date_inscription_cours ASC;
```

→ *DATEDIFF(NOW(), date)* calcule le nombre de jours entre aujourd'hui et la date d'inscription.

Q20 — Date de début prévue = inscription + 7 jours

SELECT

```
SELECT id_inscription,  
       date_inscription_cours,  
       ADDDATE(date_inscription_cours,7) AS date_debut_prevue  
FROM INSCRIPTIONS  
ORDER BY date_inscription_cours;
```

Sous-requêtes — Questions 21 à 28

EXISTS · NOT EXISTS · IN · NOT IN — Jointure implicite · ACTIVITES

Q21 — Membres ayant au moins une inscription

Sous-req.

```
SELECT DISTINCT M.nom, M.prenom  
FROM MEMBRES M, INSCRIPTIONS I  
WHERE I.id_membre = M.id_membre
```

→ *EXISTS renvoie TRUE dès qu'au moins une ligne est trouvée. Plus efficace que IN sur de grandes tables.*

Q22 — Membres sans aucune inscription

Sous-req.

```
SELECT M.nom, M.prenom, M.email  
FROM MEMBRES M  
WHERE M.id_membre NOT IN (  
    SELECT id_membre FROM INSCRIPTIONS  
);
```

→ *NOT IN* exclut tous les membres présents dans *INSCRIPTIONS*.

Q23 — Cours (ACTIVITES) jamais inscrits

Sous-req.

```
SELECT A.nom_cours, A.categorie, A.prix
FROM ACTIVITES A
WHERE A.id_cours NOT IN (
    SELECT DISTINCT id_cours FROM LIGNES_INSCRIPTION
);
```

→ La table ACTIVITES remplace COURS depuis Q4. DISTINCT dans la sous-requête optimise la recherche.

Q24 — Membres ayant suivi au moins un cours Muscu

```
SELECT DISTINCT M.nom, M.prenom
FROM MEMBRES M ,INSCRIPTIONS I, LIGNES_INSCRIPTION LI, ACTIVITES A
WHERE LI.id_inscription = I.id_inscription
      AND A.id_cours = LI.id_cours
      AND I.id_membre=M.id_membre

      AND A.categorie = 'Muscu'

);
```

Q25 — Cours inscrits dans les mêmes inscriptions que le cours n°2

Sou-req.

```
SELECT DISTINCT A.nom_cours, A.categorie
FROM ACTIVITES A, LIGNES_INSCRIPTION LI1, LIGNES_INSCRIPTION LI2
WHERE LI1.id_cours = A.id_cours
      AND LI1.id_inscription=LI2.id_inscription
      AND LI2.id_cours = 2
      AND A.id_cours <> 2;
```

Q26 — Inscriptions confirmées janv–juin 2024, avec montant

SELECT

```
SELECT M.nom, I.date_inscription_cours,  
       A.nom_cours,  
       (LI.nb_seances * LI.prix_unitaire) AS montant_total  
FROM MEMBRES M, INSCRIPTIONS I, LIGNES_INSCRIPTION LI, ACTIVITES A  
WHERE I.id_membre = M.id_membre  
      AND LI.id_inscription = I.id_inscription  
      AND A.id_cours = LI.id_cours  
      AND I.statut = 'confirmée'  
      AND I.date_inscription_cours BETWEEN '2024-01-01' AND '2024-06-30'  
ORDER BY I.date_inscription_cours;
```

Q27 — Inscriptions en attente avec date de relance

SELECT

```
SELECT M.nom AS membre,  
       I.date_inscription_cours,  
       ADDDATE(I.date_inscription_cours,15) AS date_relance,  
       I.statut  
FROM MEMBRES M, INSCRIPTIONS I  
WHERE I.id_membre = M.id_membre  
      AND I.statut = 'en attente';
```

Q28 — Membres avant 2021 ayant suivi un cours Muscu

```
SELECT DISTINCT M.nom, M.prenom, YEAR(M.date_inscription) AS annee_inscription
FROM MEMBRES M, INSCRIPTIONS I, LIGNES_INSCRIPTION LI, ACTIVITES A
WHERE YEAR(M.date_inscription) < 2021
AND LI.id_inscription = I.id_inscription
AND M.id_membre= I.id_membre
AND A.id_cours = LI.id_cours
AND A.categorie = 'Muscu';
```