

# Rapport : Gestion des événements et inscriptions avec MySQL

**Module** : Base de Données MySQL

**Nom**: Aya ouazzani akkar hammoudi

## SOMMAIRE :

-Réaliser les ateliers sous MySQL (création des tables, procédures, fonctions, triggers).

-des captures d'écran montrant l'exécution des requêtes

### 1)Création de la base et des tables :

#### Création de la base

```
1 CREATE DATABASE IF NOT EXISTS AdvancedEventSolutions;  
2 USE AdvancedEventSolutions;
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0006 seconde(s).)

```
CREATE DATABASE IF NOT EXISTS AdvancedEventSolutions;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

⚠ Note: #1007 Ne peut créer la base 'advancedeventsolutions'; elle existe déjà

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0004 seconde(s).)

```
USE AdvancedEventSolutions;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

#### Création des tables

```
3 -- Table Experts  
4 CREATE TABLE IF NOT EXISTS Experts (  
5     id VARCHAR(10) PRIMARY KEY,  
6     nomExp VARCHAR(50),  
7     prenomExp VARCHAR(50),  
8     telExp VARCHAR(20),  
9     SpecialiteExp VARCHAR(50),  
10    EmailExp VARCHAR(50)  
11 );  
12 -- Table Evenements  
13 CREATE TABLE IF NOT EXISTS Evenements (  
14     id VARCHAR(10) PRIMARY KEY,  
15     theme VARCHAR(100),  
16     date_debut DATE,  
17     date_fin DATE,  
18     description VARCHAR(255),  
19     cout_journalier DOUBLE,  
20     expert_id VARCHAR(10),  
21     FOREIGN KEY (expert_id) REFERENCES Experts(id)  
22 );
```

```

24 -- Table Ateliers
25 CREATE TABLE IF NOT EXISTS Ateliers (
26     id VARCHAR(10) PRIMARY KEY,
27     nomAtelier VARCHAR(100),
28     descriptionAtelier VARCHAR(255),
29     evenement_id VARCHAR(10),
30     FOREIGN KEY (evenement_id) REFERENCES Evenements(id)
31 );
32
33 -- Table Participants
34 CREATE TABLE IF NOT EXISTS Participants (
35     id VARCHAR(10) PRIMARY KEY,
36     nomPart VARCHAR(50),
37     prenomPart VARCHAR(50),
38     sexePart VARCHAR(10),
39     telPart VARCHAR(20),
40     emailPart VARCHAR(50)
41 );
42
43
44 -- Table Inscriptions
45 CREATE TABLE IF NOT EXISTS Inscriptions (
46     id VARCHAR(10) PRIMARY KEY,
47     dateInscription DATE,
48     statutInscription VARCHAR(50),
49     participant_id VARCHAR(10),
50     evenement_id VARCHAR(10),
51     FOREIGN KEY (participant_id) REFERENCES Participants(id),
52     FOREIGN KEY (evenement_id) REFERENCES Evenements(id)
53 );

```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0112 seconde(s).)

```
-- Table Experts CREATE TABLE IF NOT EXISTS Experts ( id VARCHAR(10) PRIMARY KEY, nomExp VARCHAR(50), prenomExp VARCHAR(50), telExp VARCHAR(20), SpecialiteExp VARCHAR(50), EmailExp VARCHAR(50) );
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0331 seconde(s).)

```
-- Table Evenements CREATE TABLE IF NOT EXISTS Evenements ( id VARCHAR(10) PRIMARY KEY, theme VARCHAR(100), date_debut DATE, date_fin DATE, description VARCHAR(255), cout_journalier DOUBLE, expert_id VARCHAR(10), FOREIGN KEY (expert_id) REFERENCES Experts(id) );
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0466 seconde(s).)

```
-- Table Ateliers CREATE TABLE IF NOT EXISTS Ateliers ( id VARCHAR(10) PRIMARY KEY, nomAtelier VARCHAR(100), descriptionAtelier VARCHAR(255), evenement_id VARCHAR(10), FOREIGN KEY (evenement_id) REFERENCES Evenements(id) );
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0099 seconde(s).)

```
-- Table Participants CREATE TABLE IF NOT EXISTS Participants ( id VARCHAR(10) PRIMARY KEY, nomPart VARCHAR(50), prenomPart VARCHAR(50), sexePart VARCHAR(10), telPart VARCHAR(20), emailPart VARCHAR(50) );
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0349 seconde(s).)

```
-- Table Inscriptions CREATE TABLE IF NOT EXISTS Inscriptions ( id VARCHAR(10) PRIMARY KEY, dateInscription DATE, statutInscription VARCHAR(50), participant_id VARCHAR(10), evenement_id VARCHAR(10), FOREIGN KEY (participant_id) REFERENCES Participants(id), FOREIGN KEY (evenement_id) REFERENCES Evenements(id) );
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

## 2) Insertion des données initiales :

```

1  -- Experts
2  INSERT INTO Experts(id, nomExp, prenomExp, telExp, SpecialiteExp, EmailExp) VALUES
3  ('EXP01','Dupont','Jean','0611111111','IA','jean.dupont@mail.com'),
4  ('EXP02','Martin','Sophie','0622222222','Cloud','sophie.martin@mail.com');
5
6  -- Participants
7  INSERT INTO Participants(id, nomPart, prenomPart, sexePart, telPart, emailPart) VALUES
8  ('P001','Ahmed','Ali','H','0633333333','ali.ahmed@mail.com'),
9  ('P002','Leila','Fatima','F','0644444444','fatima.leila@mail.com');
10
11 -- Evenements
12 INSERT INTO Evenements(id, theme, date_debut, date_fin, description, cout_journalier, expert_id) VALUES
13 ('E001','Séminaire IA','2025-11-10','2025-11-12','Introduction à l'IA',500,'EXP01');
14
15 -- Ateliers
16 INSERT INTO Ateliers(id, nomAtelier, descriptionAtelier, evenement_id) VALUES
17 ('A001','Atelier Machine Learning','Exercices pratiques sur ML','E001');
18
19 -- Inscriptions
20 INSERT INTO Inscriptions(id, dateInscription, statutInscription, participant_id, evenement_id) VALUES
21 ('INS001','2025-06-15','Confirmé','P001','E001');
22

```

```

14
15 -- Ateliers
16 INSERT INTO Ateliers(id, nomAtelier, descriptionAtelier, evenement_id) VALUES
17 ('A001','Atelier Machine Learning','Exercices pratiques sur ML','E001');
18
19 -- Inscriptions
20 INSERT INTO Inscriptions(id, dateInscription, statutInscription, participant_id, evenement_id) VALUES
21 ('INS001','2025-06-15','Confirmé','P001','E001');
22

```

✓ 2 lignes insérées. (traitement en 0,0169 seconde(s).)

```

-- Experts INSERT INTO Experts(id, nomExp, prenomExp, telExp, SpecialiteExp, EmailExp) VALUES
('EXP01','Dupont','Jean','0611111111','IA','jean.dupont@mail.com'),
('EXP02','Martin','Sophie','0622222222','Cloud','sophie.martin@mail.com');

```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ 2 lignes insérées. (traitement en 0,0064 seconde(s).)

```

-- Participants INSERT INTO Participants(id, nomPart, prenomPart, sexePart, telPart, emailPart) VALUES
('P001','Ahmed','Ali','H','0633333333','ali.ahmed@mail.com'), ('P002','Leila','Fatima','F','0644444444','fatima.leila@mail.com');

```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ 1 ligne insérée. (traitement en 0,0045 seconde(s).)

```

-- Evenements INSERT INTO Evenements(id, theme, date_debut, date_fin, description, cout_journalier, expert_id) VALUES
('E001','Séminaire IA','2025-11-10','2025-11-12','Introduction à l'IA',500,'EXP01');

```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ 1 ligne insérée. (traitement en 0,0044 seconde(s).)

```

-- Ateliers INSERT INTO Ateliers(id, nomAtelier, descriptionAtelier, evenement_id) VALUES ('A001','Atelier Machine Learning','Exercices
pratiques sur ML','E001');

```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ 1 ligne insérée. (traitement en 0,0047 seconde(s).)

```

-- Inscriptions INSERT INTO Inscriptions(id, dateInscription, statutInscription, participant_id, evenement_id) VALUES ('INS001','2025-
06-15','Confirmé','P001','E001');

```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

## 1 SELECT \* FROM Experts;

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0004 seconde(s).)

SELECT \* FROM Experts;

☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

			id	nomExp	prenomExp	telExp	SpecialiteExp	EmailExp	
<input type="checkbox"/>	Éditer	Copier	Supprimer	EXP01	Dupont	Jean	0611111111	IA	jean.dupont@mail.com
<input type="checkbox"/>	Éditer	Copier	Supprimer	EXP02	Martin	Sophie	0622222222	Cloud	sophie.martin@mail.com

↑ ☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Opérations sur les résultats de la requête

## 1 SELECT \* FROM Evenements;

2

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0005 seconde(s).)

SELECT \* FROM Evenements;

☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

				id	theme	date_debut	date_fin	description	cout_journalier	expert_id
<input type="checkbox"/>	Éditer	Copier	Supprimer	E001	Séminaire IA	2025-11-10	2025-11-12	Introduction à l'IA	500	EXP01

↑ ☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

### Procédure : Insérer un nouvel événement en vérifiant l'expert

```
1 DELIMITER $$
2 CREATE PROCEDURE inserer_evenement(
3     IN p_id VARCHAR(10),
4     IN p_theme VARCHAR(100),
5     IN p_date_debut DATE,
6     IN p_date_fin DATE,
7     IN p_description VARCHAR(255),
8     IN p_cout DOUBLE,
9     IN p_expert_id VARCHAR(10)
10 )
11 BEGIN
12     IF EXISTS (SELECT 1 FROM Experts WHERE id = p_expert_id) THEN
13         INSERT INTO Evenements(id, theme, date_debut, date_fin, description, cout_journalier, expert_id)
14         VALUES (p_id, p_theme, p_date_debut, p_date_fin, p_description, p_cout, p_expert_id);
15     ELSE
16         SIGNAL SQLSTATE '45000'
17         SET MESSAGE_TEXT = 'Erreur : L'expert n'existe pas.';
18     END IF;
19 END $$
20 DELIMITER ;
```



✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0102 seconde(s).)

```
CREATE PROCEDURE inserer_evenement( IN p_id VARCHAR(10), IN p_theme VARCHAR(100), IN p_date_debut DATE, IN p_date_fin DATE, IN
p_description VARCHAR(255), IN p_cout DOUBLE, IN p_expert_id VARCHAR(10) ) BEGIN IF EXISTS (SELECT 1 FROM Experts WHERE id = p_expert_id)
THEN INSERT INTO Evenements(id, theme, date_debut, date_fin, description, cout_journalier, expert_id) VALUES (p_id, p_theme,
p_date_debut, p_date_fin, p_description, p_cout, p_expert_id); ELSE SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Erreur : L'expert
n'existe pas.'; END IF; END;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

## Fonction : Coût total des événements d'un expert donné

```
1 DELIMITER $$
2 CREATE FUNCTION cout_total_expert(p_nom VARCHAR(50), p_prenom VARCHAR(50))
3 RETURNS DOUBLE
4 DETERMINISTIC
5 BEGIN
6     DECLARE total DOUBLE;
7
8     SELECT SUM(cout_journalier * DATEDIFF(date_fin, date_debut) + 1)
9     INTO total
10    FROM Evenements e
11    JOIN Experts ex ON e.expert_id = ex.id
12    WHERE ex.nomExp = p_nom AND ex.prenomExp = p_prenom;
13
14    RETURN IFNULL(total, 0);
15 END $$
16 DELIMITER ;
17
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0162 seconde(s).)

```
CREATE FUNCTION cout_total_expert(p_nom VARCHAR(50), p_prenom VARCHAR(50)) RETURNS DOUBLE DETERMINISTIC BEGIN DECLARE total DOUBLE;
SELECT SUM(cout_journalier * DATEDIFF(date_fin, date_debut) + 1) INTO total FROM Evenements e JOIN Experts ex ON e.expert_id = ex.id
WHERE ex.nomExp = p_nom AND ex.prenomExp = p_prenom; RETURN IFNULL(total, 0); END;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

## Trigger : Interdire l'inscription en juillet de l'année en cours

```
1 DELIMITER $$
2
3 CREATE TRIGGER verif_juillet_inscription
4 BEFORE INSERT ON Inscriptions
5 FOR EACH ROW
6 BEGIN
7     IF MONTH(NEW.dateInscription) = 7 AND YEAR(NEW.dateInscription) = YEAR(CURDATE()) THEN
8         SIGNAL SQLSTATE '45000'
9         SET MESSAGE_TEXT = 'Inscription interdite en juillet de cette année.';
10    END IF;
11 END $$
12 DELIMITER ;
13
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0207 seconde(s).)

```
CREATE TRIGGER verif_juillet_inscription BEFORE INSERT ON Inscriptions FOR EACH ROW BEGIN IF MONTH(NEW.dateInscription) = 7 AND
YEAR(NEW.dateInscription) = YEAR(CURDATE()) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Inscription interdite en juillet de cette
année.'; END IF; END;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

## Tester la fonction coût total

```
1 SELECT cout_total_expert('Dupont','Jean') AS cout_total;
```

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0021 seconde(s).)

```
SELECT cout_total_expert('Dupont','Jean') AS cout_total;
```

☐ Profilage [ [Éditer en ligne](#) ] [ [Éditer](#) ] [ [Expliquer SQL](#) ] [ [Créer le code source PHP](#) ] [ [Actualiser](#) ]

☐ Tout afficher | Nombre de lignes : 25 ▼ | Filtrer les lignes:

Options supplémentaires

**cout\_total**

1001