EXPENSE TRACKER - COMPLETE SQL QUERIES GUIDE

MCA Project | All Latest Working Queries | January 2026

TABLE OF CONTENTS

---

1 DATABASE & TABLES CREATION

1.1 Create Database

```
CREATE DATABASE IF NOT EXISTS expense_tracker;
USE expense_tracker;
```

1.2 Create Users Table

```
CREATE TABLE IF NOT EXISTS users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    full_name VARCHAR(100),
    role ENUM('admin', 'user') DEFAULT 'user',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

1.3 Create Expenses Table

```
CREATE TABLE IF NOT EXISTS expenses (
    expense_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    category VARCHAR(50) NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    description TEXT,
```

```
    expense_date DATE NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE
);
```

1.4 Create Budget Table (Optional)

```
CREATE TABLE IF NOT EXISTS budget (
    budget_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    category VARCHAR(50) NOT NULL,
    limit_amount DECIMAL(10, 2) NOT NULL,
    month_year VARCHAR(7),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE
);
```

1.5 Create Categories Table (Optional)

```
CREATE TABLE IF NOT EXISTS categories (
    category_id INT AUTO_INCREMENT PRIMARY KEY,
    category_name VARCHAR(50) UNIQUE NOT NULL,
    description TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

---

2 ADMIN USER SETUP (WORKING SOLUTION)

2.1 Insert Admin User

```
INSERT INTO users (username, email, password_hash, full_name, role, created_at)
VALUES ('admin', 'admin@expense.com', '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uh
```

Login: admin / admin123

2.2 Insert Test User

```
INSERT INTO users (username, email, password_hash, full_name, role, created_at)
VALUES ('user1', 'user1@test.com', '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheW
```

Login: user1 / user123

2.3 Insert Demo User

```
INSERT INTO users (username, email, password_hash, full_name, role, created_at)
VALUES ('demo', 'demo@demo.com', '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/i
```

Login: demo / admin123

---

3 USER MANAGEMENT QUERIES

3.1 Get All Users (Admin Only)

```
SELECT user_id, username, email, full_name, role, created_at
FROM users
ORDER BY created_at DESC;
```

3.2 Get User by Username or Email (Login)

```
SELECT * FROM users
WHERE username = 'admin' OR email = 'admin@expense.com';
```

3.3 Get User by ID

```
SELECT * FROM users
WHERE user_id = 1;
```

3.4 Update User Profile

```
UPDATE users
SET full_name = 'John Doe', email = 'john@example.com', updated_at = NOW()
WHERE user_id = 2;
```

3.5 Change User Password (Hashed)

```
UPDATE users
SET password_hash = '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi'
WHERE user_id = 2;
```

3.6 Delete User (Cascades to expenses)

```
DELETE FROM users
WHERE user_id = 2;
```

3.7 Get User Count

```
SELECT COUNT(*) as total_users FROM users;
```

3.8 Get Admin Users

```
SELECT user_id, username, email
FROM users
WHERE role = 'admin';
```

---

4 EXPENSE MANAGEMENT QUERIES

4.1 Add New Expense

```
INSERT INTO expenses (user_id, category, amount, description, expense_date)
VALUES (1, 'Food', 500.00, 'Lunch at restaurant', '2026-01-09');
```

4.2 Get All Expenses for User

```
SELECT expense_id, category, amount, description, expense_date
FROM expenses
WHERE user_id = 1
ORDER BY expense_date DESC;
```

4.3 Get Expenses by Date Range

```
SELECT expense_id, category, amount, description, expense_date
FROM expenses
WHERE user_id = 1
AND expense_date BETWEEN '2026-01-01' AND '2026-01-31'
ORDER BY expense_date DESC;
```

4.4 Get Expenses by Category

```
SELECT category, SUM(amount) as total
FROM expenses
WHERE user_id = 1
GROUP BY category;
```

4.5 Get Total Expenses for User

```
SELECT SUM(amount) as total_expenses
FROM expenses
WHERE user_id = 1;
```

4.6 Get Monthly Expense Summary

```
SELECT
    DATE_FORMAT(expense_date, '%Y-%m') as month,
    SUM(amount) as monthly_total,
    COUNT(*) as transaction_count
FROM expenses
WHERE user_id = 1
GROUP BY DATE_FORMAT(expense_date, '%Y-%m')
ORDER BY month DESC;
```

4.7 Get Top Expense Categories

```
SELECT category, COUNT(*) as frequency, SUM(amount) as total
FROM expenses
WHERE user_id = 1
GROUP BY category
ORDER BY total DESC
LIMIT 5;
```

4.8 Update Expense

```
UPDATE expenses
SET amount = 600.00, description = 'Updated lunch cost', category = 'Food'
WHERE expense_id = 1 AND user_id = 1;
```

4.9 Delete Expense

```
DELETE FROM expenses
WHERE expense_id = 1 AND user_id = 1;
```

4.10 Search Expenses by Description

```
SELECT * FROM expenses
WHERE user_id = 1
AND description LIKE '%restaurant%'
ORDER BY expense_date DESC;
```

---

5 REPORTS & ANALYSIS QUERIES

5.1 Daily Expense Report

```
SELECT
    expense_date,
    COUNT(*) as transactions,
    SUM(amount) as daily_total
FROM expenses
WHERE user_id = 1
GROUP BY expense_date
ORDER BY expense_date DESC;
```

5.2 Category Breakdown (Percentage)

```
SELECT
    category,
    SUM(amount) as total,
    ROUND(SUM(amount) / (SELECT SUM(amount) FROM expenses WHERE user_id = 1) * 100, 2) as pe
```

```
FROM expenses
WHERE user_id = 1
GROUP BY category;
```

5.3 Highest Expense Transaction

```
SELECT * FROM expenses
WHERE user_id = 1
ORDER BY amount DESC
LIMIT 1;
```

5.4 Average Monthly Spending

```
SELECT
    ROUND(AVG(monthly_total), 2) as avg_monthly_spending
FROM (
    SELECT SUM(amount) as monthly_total
    FROM expenses
    WHERE user_id = 1
    GROUP BY DATE_FORMAT(expense_date, '%Y-%m')
) as monthly_summary;
```

5.5 Expense Trend (Last 6 Months)

```
SELECT
    DATE_FORMAT(expense_date, '%Y-%m') as month,
    SUM(amount) as total
FROM expenses
WHERE user_id = 1
AND expense_date >= DATE_SUB(NOW(), INTERVAL 6 MONTH)
GROUP BY DATE_FORMAT(expense_date, '%Y-%m')
ORDER BY month;
```

5.6 Admin Dashboard - All Users Expenses

```
SELECT
    u.username,
    COUNT(e.expense_id) as total_expenses,
    SUM(e.amount) as total_amount,
    ROUND(AVG(e.amount), 2) as avg_expense
FROM users u
LEFT JOIN expenses e ON u.user_id = e.user_id
WHERE u.role = 'user'
GROUP BY u.user_id, u.username;
```

5.7 Get Platform Statistics

```
SELECT
    (SELECT COUNT(*) FROM users) as total_users,
    (SELECT COUNT(*) FROM expenses) as total_expenses,
    (SELECT SUM(amount) FROM expenses) as total_amount_tracked;
```

---

6 SECURITY & MAINTENANCE QUERIES

6.1 Backup All Data

```
# Run in terminal/command prompt
mysqldump -u root expense_tracker > backup_expense_tracker.sql
```

6.2 Restore from Backup

```
# Run in terminal/command prompt
mysql -u root expense_tracker < backup_expense_tracker.sql
```

6.3 View All Tables

```
SHOW TABLES;
```

6.4 View Table Structure

```
DESCRIBE users;
DESCRIBE expenses;
DESCRIBE budget;
```

6.5 View Indexes

```
SHOW INDEX FROM users;
SHOW INDEX FROM expenses;
```

6.6 Create Index for Performance

```
CREATE INDEX idx_user_expenses ON expenses(user_id);
CREATE INDEX idx_expense_date ON expenses(expense_date);
CREATE INDEX idx_username ON users(username);
```

6.7 Clear All Expenses (Dangerous!)

```
DELETE FROM expenses;
```

6.8 Clear All Users (VERY Dangerous!)

```
DELETE FROM users;
```

6.9 Reset Auto-Increment

```
ALTER TABLE expenses AUTO_INCREMENT = 1;
ALTER TABLE users AUTO_INCREMENT = 1;
```

6.10 View Database Size

```
SELECT
    table_name,
    ROUND(((data_length + index_length) / 1024 / 1024), 2) as size_mb
FROM information_schema.TABLES
WHERE table_schema = 'expense_tracker'
ORDER BY (data_length + index_length) DESC;
```

LOGIN CREDENTIALS

Username	Email	Password	Role
admin	admin@expense.com	admin123	Admin
user1	user1@test.com	user123	User
demo	demo@demo.com	admin123	User

PASSWORD HASHING INFO

Hash Algorithm: bcrypt ($2y10...) **Plaintext:** admin123 / user123 **PHP Function:** `password_verify($input, $hash)`

Example PHP Code:

```
if (password_verify('admin123', '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/ig
    // Login successful
}
```

COMMON OPERATIONS CHEAT SHEET

Operation	Query
Count user expenses	`SELECT COUNT(*) FROM expenses WHERE user_id = 1;`

Operation	Query
Total spent by user	`SELECT SUM(amount) FROM expenses WHERE user_id = 1;`
Expenses this month	`SELECT * FROM expenses WHERE user_id = 1 AND MONTH(expense_date) = MONTH(NOW());`
Most expensive item	`SELECT * FROM expenses WHERE user_id = 1 ORDER BY amount DESC LIMIT 1;`
Top expense category	`SELECT category, SUM(amount) FROM expenses WHERE user_id = 1 GROUP BY category ORDER BY SUM(amount) DESC LIMIT 1;`

IMPORTANT NOTES

1. **user_id is PRIMARY KEY** - Not `id`
2. **password_hash stores bcrypt** - Use `password_verify()` not plain comparison
3. **FOREIGN KEY CASCADE** - Deleting user deletes all their expenses
4. **Timestamps** - Auto-created on insert, auto-updated on change
5. **All amounts in DECIMAL(10,2)** - For currency precision
6. **Created: January 9, 2026** - Latest version
7. **Updated: January 11, 2026** - Verified working

SETUP INSTRUCTIONS FOR NEW PC

Step 1: Start MySQL Service

```
# Windows Command Prompt
cd C:\xampp\mysql\bin
mysql -u root
```

Step 2: Run All Database Creation Queries

Copy and paste all queries from **Section 1** (Database & Tables Creation)

Step 3: Insert Test Users

Copy and paste all queries from **Section 2** (Admin User Setup)

Step 4: Create Indexes (Optional but Recommended)

Copy queries from **Section 6.6** (Create Index for Performance)

Step 5: Verify Setup

Run query: `SELECT * FROM users;` - You should see 3 test users

VERIFICATION CHECKLIST

- ☐ Database `expense_tracker` created
- ☐ `users` table exists with 5 columns
- ☐ `expenses` table exists with 6 columns
- ☐ `budget` table created (optional)
- ☐ `categories` table created (optional)
- ☐ 3 test users inserted (admin, user1, demo)
- ☐ Can select from all tables
- ☐ Foreign key constraints working
- ☐ Indexes created for performance

Generated for MCA Project | Expense Tracker System Last Updated: January 11, 2026 IST Version: 1.0 - Production Ready