Sports Training & Fitness Database

Goal: Track athlete training sessions and progress.

Tables:

- `athletes(athlete_id, name, sport, age)`
- `sessions(session_id, athlete_id, date, duration, type)`
- `metrics(session_id, speed, endurance, strength_score)`

Queries:

- Find average training duration per athlete.
- Identify improvements in endurance over time.
- Compare training intensity across sports.

a. Create database

```
CREATE DATABASE sports_training;
USE sports_training;
```

```
mysql> CREATE DATABASE sports_training;
Query OK, 1 row affected (0.04 sec)

mysql> USE sports_training;
Database changed
```

b. Create tables

```
-- Athlete information
CREATE TABLE athletes (
    athlete_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    sport VARCHAR(50),
    age INT
);

-- Training sessions
CREATE TABLE sessions (
    session_id INT AUTO_INCREMENT PRIMARY KEY,
    athlete_id INT,
    date DATE,
    duration INT, -- minutes
    type VARCHAR(50), -- e.g., Strength, Endurance, Speed
    FOREIGN KEY (athlete_id) REFERENCES athletes(athlete_id)
);

-- Performance metrics per session
CREATE TABLE metrics (
    metric_id INT AUTO_INCREMENT PRIMARY KEY,
    session_id INT,
    speed DECIMAL(5,2), -- km/h
    endurance DECIMAL(5,2), -- minutes sustained
    strength_score INT, -- arbitrary score
    FOREIGN KEY (session_id) REFERENCES sessions(session_id)
);
```

```
mysql> -- Athlete information
mysql> CREATE TABLE athletes (
    ->     athlete_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     sport VARCHAR(50),
    ->     age INT
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql>
mysql> -- Training sessions
mysql> CREATE TABLE sessions (
    ->     session_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     athlete_id INT,
    ->     date DATE,
    ->     duration INT, -- minutes
    ->     type VARCHAR(50), -- e.g., Strength, Endurance, Speed
    ->     FOREIGN KEY (athlete_id) REFERENCES athletes(athlete_id)
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql>
mysql> -- Performance metrics per session
mysql> CREATE TABLE metrics (
    ->     metric_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     session_id INT,
    ->     speed DECIMAL(5,2), -- km/h
    ->     endurance DECIMAL(5,2), -- minutes sustained
    ->     strength_score INT, -- arbitrary score
    ->     FOREIGN KEY (session_id) REFERENCES sessions(session_id)
    -> );
Query OK, 0 rows affected (0.10 sec)
```

c. Insert sample data

```
INSERT INTO athletes (name, sport, age)
VALUES
('Alice Johnson', 'Basketball', 22),
('Bob Smith', 'Soccer', 25),
('Charlie Lee', 'Volleyball', 20);

INSERT INTO sessions (athlete_id, date, duration, type)
VALUES
(1, '2025-11-01', 60, 'Endurance'),
(2, '2025-11-02', 45, 'Strength'),
(3, '2025-11-03', 50, 'Speed'),
(1, '2025-11-04', 70, 'Strength');

INSERT INTO metrics (session_id, speed, endurance, strength_score)
VALUES
(1, 12.5, 40.0, 70),
(2, 10.0, 30.0, 85),
(3, 14.0, 35.0, 60),
(4, 11.0, 50.0, 90);
```

```
mysql> INSERT INTO athletes (name, sport, age)
    -> VALUES
    -> ('Alice Johnson', 'Basketball', 22),
    -> ('Bob Smith', 'Soccer', 25),
    -> ('Charlie Lee', 'Volleyball', 20);
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO sessions (athlete_id, date, duration, type)
    -> VALUES
    -> (1, '2025-11-01', 60, 'Endurance'),
    -> (2, '2025-11-02', 45, 'Strength'),
    -> (3, '2025-11-03', 50, 'Speed'),
    -> (1, '2025-11-04', 70, 'Strength');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO metrics (session_id, speed, endurance, strength_score)
    -> VALUES
    -> (1, 12.5, 40.0, 70),
    -> (2, 10.0, 30.0, 85),
    -> (3, 14.0, 35.0, 60),
    -> (4, 11.0, 50.0, 90);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

d. Example queries

```
-- Average training duration per athlete
SELECT a.name, AVG(s.duration) AS avg_duration
FROM athletes a
JOIN sessions s ON a.athlete_id = s.athlete_id
GROUP BY a.name;

-- Improvements in endurance over time
SELECT a.name, s.date, m.endurance
FROM athletes a
JOIN sessions s ON a.athlete_id = s.athlete_id
JOIN metrics m ON s.session_id = m.session_id
WHERE a.name = 'Alice Johnson'
ORDER BY s.date;

-- Compare training intensity across sports
SELECT a.sport, AVG(m.strength_score) AS avg_strength
FROM athletes a
JOIN sessions s ON a.athlete_id = s.athlete_id
JOIN metrics m ON s.session_id = m.session_id
GROUP BY a.sport;
```

```
mysql> -- Average training duration per athlete
mysql> SELECT a.name, AVG(s.duration) AS avg_duration
    -> FROM athletes a
    -> JOIN sessions s ON a.athlete_id = s.athlete_id
    -> GROUP BY a.name;
+---------------+--------------+
| name          | avg_duration |
+---------------+--------------+
| Alice Johnson |      65.0000 |
| Bob Smith     |      45.0000 |
| Charlie Lee   |      50.0000 |
+---------------+--------------+
3 rows in set (0.00 sec)

mysql>
mysql> -- Improvements in endurance over time
mysql> SELECT a.name, s.date, m.endurance
    -> FROM athletes a
    -> JOIN sessions s ON a.athlete_id = s.athlete_id
    -> JOIN metrics m ON s.session_id = m.session_id
    -> WHERE a.name = 'Alice Johnson'
    -> ORDER BY s.date;
+---------------+------------+-----------+
| name          | date       | endurance |
+---------------+------------+-----------+
| Alice Johnson | 2025-11-01 |     40.00 |
| Alice Johnson | 2025-11-04 |     50.00 |
+---------------+------------+-----------+
2 rows in set (0.03 sec)
```

```
mysql>
mysql> -- Compare training intensity across sports
mysql> SELECT a.sport, AVG(m.strength_score) AS avg_strength
    -> FROM athletes a
    -> JOIN sessions s ON a.athlete_id = s.athlete_id
    -> JOIN metrics m ON s.session_id = m.session_id
    -> GROUP BY a.sport;
+------------+--------------+
| sport      | avg_strength |
+------------+--------------+
| Basketball |      80.0000 |
| Soccer     |      85.0000 |
| Volleyball |      60.0000 |
+------------+--------------+
3 rows in set (0.00 sec)
```