

Backup Database MariaDB

- **Buka Terminal**
 - Pastikan Anda memiliki akses ke terminal dan hak akses root atau pengguna dengan izin yang diperlukan.
- **Login ke MariaDB**
 - Masuk ke MariaDB menggunakan perintah berikut dan masukkan password jika diminta: **sudo mysql -u root -p**
- **Pilih Database**
 - Setelah masuk, Anda dapat melihat daftar database dengan perintah: **SHOW DATABASES;**
 - Pilih database yang ingin Anda backup.
- **Backup Database dengan mysqldump**
 - Keluar dari MariaDB dengan perintah **exit** atau **quit**.
 - Gunakan perintah **mysqldump** untuk membuat backup database. Gantilah **your_database_name** dengan nama database yang ingin Anda backup dan **backup_file.sql** dengan nama file backup yang diinginkan: **mysqldump -u root -p olahraga > backup_olahraga.sql**.
 - Masukkan password root MariaDB jika diminta.
- **Verifikasi File Backup**
 - Pastikan file backup (**backup_file.sql**) telah dibuat di direktori yang Anda tentukan.

Restore Database MariaDB

- **Buka Terminal**
 - Pastikan Anda memiliki akses ke terminal dan hak akses root atau pengguna dengan izin yang diperlukan.
- **Login ke MariaDB**
 - Masuk ke MariaDB menggunakan perintah berikut dan masukkan password jika diminta: **sudo mysql -u root -p**.
- **Buat Database Baru**
 - Jika Anda ingin merestore database ke database baru, buat database baru dengan perintah berikut. Gantilah **new_database_name** dengan nama database yang diinginkan: **CREATE DATABASE olahraga;**
 - Jika Anda ingin merestore ke database yang sudah ada, Anda dapat melewati langkah ini.
- **Keluar dari MariaDB**
 - Keluar dari MariaDB dengan perintah **exit** atau **quit**.
- **Restore Database dengan mysql**
 - Gunakan perintah **mysql** untuk merestore database dari file backup. Gantilah **new_database_name** dengan nama database yang ingin Anda restore dan **backup_file.sql** dengan nama file backup: **mysql -u root -p olahraga < backup_olahraga.sql**.
 - Masukkan password root MariaDB jika diminta.

- **Verifikasi Restore**
 - Masuk kembali ke MariaDB dan cek apakah database telah direstore dengan perintah: **USE olahraga; SHOW TABLES;**

DDL

- **Query**
 - **CREATE TABLE sports (**
 - **id INT AUTO_INCREMENT PRIMARY KEY,**
 - **name VARCHAR(255) NOT NULL,**
 - **type VARCHAR(255) NOT NULL,**
 - **created_at DATETIME,**
 - **updated_at DATETIME**
 - **);**

Create Stored Procedure

- **Create**
 - **DELIMITER //**
 -
 - **CREATE PROCEDURE CreateSport(IN sport_name VARCHAR(255), IN sport_type VARCHAR(255))**
 - **BEGIN**
 - **INSERT INTO sports (name, type, created_at, updated_at)**
 - **VALUES (sport_name, sport_type, NOW(), NOW());**
 - **END //**
 -
 - **DELIMITER ;**
 - **CALL CreateSport('Football', 'Outdoor');**
- **Read**
 - **DELIMITER //**
 -
 - **CREATE PROCEDURE ReadSports()**
 - **BEGIN**
 - **SELECT * FROM sports;**
 - **END //**
 -
 - **DELIMITER ;**
 - **CALL ReadSports();**
- **Update**
 - **DELIMITER //**
 -
 - **CREATE PROCEDURE UpdateSport(IN sport_id INT, IN sport_name VARCHAR(255), IN sport_type VARCHAR(255))**
 - **BEGIN**

- **UPDATE sports**
- **SET name = sport_name, type = sport_type, updated_at = NOW()**
- **WHERE id = sport_id;**
- **END //**
-
- **DELIMITER ;**
- **CALL UpdateSport(1, 'Soccer', 'Outdoor');**
- **Delete**
 - **DELIMITER //**
 -
 - **CREATE PROCEDURE DeleteSport(IN sport_id INT)**
 - **BEGIN**
 - **DELETE FROM sports WHERE id = sport_id;**
 - **END //**
 -
 - **DELIMITER ;**
 - **CALL DeleteSport(1);**
- **Buat Tabel histories**
 - **CREATE TABLE histories (**
 - **id INT AUTO_INCREMENT PRIMARY KEY,**
 - **sport_id INT,**
 - **action VARCHAR(50),**
 - **old_name VARCHAR(255),**
 - **new_name VARCHAR(255),**
 - **old_type VARCHAR(255),**
 - **new_type VARCHAR(255),**
 - **changed_at DATETIME**
 - **);**

Buat Trigger untuk INSERT :

DELIMITER //

CREATE TRIGGER after_sports_insert

AFTER INSERT ON sports

FOR EACH ROW

BEGIN

INSERT INTO histories (sport_id, action, new_name, new_type, changed_at)

```
VALUES (NEW.id, 'INSERT', NEW.name, NEW.type, NOW());  
END //
```

```
DELIMITER ;
```

Buat Trigger untuk UPDATE

```
DELIMITER //
```

```
CREATE TRIGGER after_sports_update
```

```
AFTER UPDATE ON sports
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO histories (sport_id, action, old_name, new_name, old_type, new_type,  
changed_at)
```

```
VALUES (OLD.id, 'UPDATE', OLD.name, NEW.name, OLD.type, NEW.type, NOW());  
END //
```

```
DELIMITER ;
```

Buat Trigger untuk DELETE

```
DELIMITER //
```

```
CREATE TRIGGER after_sports_delete
```

```
AFTER DELETE ON sports
```

```
FOR EACH ROW
```

```
BEGIN
```

```

INSERT INTO histories (sport_id, action, old_name, old_type, changed_at)

VALUES (OLD.id, 'DELETE', OLD.name, OLD.type, NOW());

END //

DELIMITER ;

Query dengan CTE:
    WITH LatestHistory AS (
        SELECT h.sport_id, h.action, h.old_name, h.new_name, h.old_type, h.new_type,
        h.changed_at,
            ROW_NUMBER() OVER (PARTITION BY h.sport_id ORDER BY
        h.changed_at DESC) AS rn
        FROM histories h
    )

    SELECT s.id, s.name, s.type, s.created_at, s.updated_at,
        lh.action, lh.old_name, lh.new_name, lh.old_type, lh.new_type, lh.changed_at
    FROM sports s
    LEFT JOIN LatestHistory lh ON s.id = lh.sport_id AND lh.rn = 1
ORDER BY s.id;

```