CONFIGURATION GUIDES › DATABASES

Automate Daily Batch Backups of Init profiling hist Table Records to an Archive Database

Applicare 9.2 · Updated February 10, 2026 · mysql · **Pending technical review**

Step 1: Before backing up the database, create the **applicarearchive** schema, execute the applicare-related sql's and ensure all tables are created.

Step 2: Connect to the database via **MySQL Workbench**, open a new SQL query tab, and execute the following command.

NOTE: Replace the <targetDB> placeholder with actual backup database name before running the query.

```
TEXT
DELIMITER //

CREATE PROCEDURE applicarearchive.sp_BatchArchiveYesterday()
BEGIN
   DECLARE rows_affected INT DEFAULT 1;

   SET @yesterday_start = CONCAT(CURDATE() - INTERVAL 1 DAY, ' 00:00:00');
   SET @yesterday_end   = CONCAT(CURDATE() - INTERVAL 1 DAY, ' 23:59:59');

   SET @sourcedb = '<targetDB>';
   SET @backupdb  = 'applicarearchive';
   WHILE rows_affected > 0 DO

       SET @sql_query = CONCAT(
           'INSERT INTO ', @backupdb, '.int_profiling_hist ',
           'SELECT * FROM ', @sourcedb, '.int_profiling_hist ',
           'WHERE from_time BETWEEN ? AND ? ',
           'AND rl NOT IN (SELECT rl FROM ', @backupdb, '.int_profiling_hist) ',
           'LIMIT 10000'
       );
       PREPARE stmt FROM @sql_query;
       EXECUTE stmt USING @yesterday_start, @yesterday_end;
       SET rows_affected = ROW_COUNT();
       DEALLOCATE PREPARE stmt;

   END WHILE;
END //
DELIMITER ;
```

NOTE: The query above only backs up data from the int_profiling_hist table, rather than the entire database

Safe Data Duplication: This query performs a "Safe Copy" that mirrors yesterday's data into the archive without removing or altering the original records

Batch Load Management: It processes records in chunks of 10,000 to maintain low CPU usage and prevent the database from locking up during the transfer.

Automated Time Filtering: The script automatically calculates the 24-hour range for "Yesterday," ensuring only the correct historical data is captured each time it runs.

Duplicate Prevention: It uses a unique identifier check (`rl`) to ensure that even if the script runs twice, it won't create double entries in the archive.

Dynamic Table Routing: Through the use of Prepared Statements, the script can dynamically target different databases without needing manual code changes.

Step 3: Run the query below to enable the MySQL Event Scheduler and schedule the **sp_BatchArchiveYesterday** procedure to run daily at 1:00 AM.

```TEXT
SET GLOBAL event_scheduler = ON;
DROP EVENT IF EXISTS applicarearchive.daily_archive_at_1am;
CREATE EVENT applicarearchive.daily_archive_at_1am
ON SCHEDULE EVERY 1 DAY
STARTS (CURRENT_DATE + INTERVAL 1 DAY + INTERVAL 1 HOUR)
DO
 CALL applicarearchive.sp_BatchArchiveYesterday();
```

To modify the scheduled time, first execute the query below to drop the existing event, then re-create it with your desired schedule.

```TEXT
DROP EVENT IF EXISTS applicarearchive.daily_archive_at_1am;
```

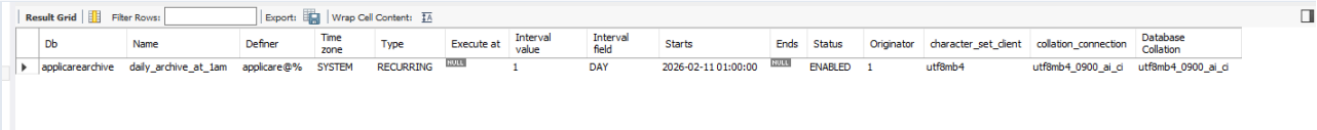
To change the time to 8:00 PM, We need to adjust the `INTERVAL` in the `STARTS` clause. Since MySQL uses a 24-hour format, 8:00 PM is 20 hours.

e.g.

```TEXT
SET GLOBAL event_scheduler = ON;
CREATE EVENT applicarearchive.daily_archive_at_8pm
ON SCHEDULE EVERY 1 DAY
STARTS (CURRENT_DATE + INTERVAL 1 DAY + INTERVAL 20 HOUR)
DO
 CALL applicarearchive.sp_BatchArchiveYesterday();
```

Step 4: Use this query to list all active events within the **applicarearchive** schema.

```
TEXT
SHOW EVENTS FROM applicarearchive;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Db	Name	Definer	Time zone	Type	Execute at	Interval value	Interval field	Starts	Ends	Status	Originator	character_set_client	collation_connection	Database Collation
applicarearchive	daily_archive_at_1am	applicare@%	SYSTEM	RECURRING	NULL	1	DAY	2026-02-11 01:00:00	NULL	ENABLED	1	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci