

# MariaDB 이벤트 복사 트러블슈팅 정리

daemon\_bybit\_Ticker → daemon\_bybit\_coin\_1m  
2026-03-05

## 1. 문제 요약

원장 테이블(daemon\_bybit\_Ticker)의 데이터를 MariaDB 이벤트로 누적 테이블(daemon\_bybit\_coin\_1m)에 복사하려 했으나 데이터가 들어오지 않는 문제 발생.

## 2. 원인 분석 과정

### 2-1. 테이블 구조 문제 (해결)

처음에 두 테이블 구조가 달랐음:

- 2번 테이블에만 있던 컬럼 7개 삭제: deliveryTime, deliveryFeeRate, predictedDeliveryPrice, basisRate, value, coin, balance
- 1번 테이블에만 있던 컬럼 2개 추가: korean\_name, day\_of\_week
- 타입 불일치 컬럼들: postOnlyMaxOrderQty, prevPrice1h 등 varchar → decimal 정리

### 2-2. VIRTUAL 컬럼 문제 (해결)

이벤트 정의문에 자동 생성 컬럼 포함 시 에러 발생:

- updated\_at: ON UPDATE CURRENT\_TIMESTAMP() 자동 컬럼 → INSERT 목록에서 제외
- day\_of\_week: VIRTUAL GENERATED 컬럼 → INSERT 목록에서 제외

### 2-3. 인덱스 문제 (해결)

복사 테이블에 symbol 단독 UNIQUE 인덱스가 있어 같은 symbol 재삽입 시 충돌 발생.

- 해결: 테이블 재생성, idx\_symbol + idx\_time 일반 인덱스로 변경 (누적 구조)

### 2-4. 진짜 원인: mPnL 컬럼 (핵심)

이벤트가 ENABLED 상태인데 LAST\_EXECUTED가 NULL → 한 번도 실행 안 됨.

원인: mPnL 컬럼값이 빈 문자열로 저장되어 있고, strict mode(STRICT\_TRANS\_TABLES)가 이를 에러로 처리해 INSERT 전체를 막음.

배경:

- 바이비트 recent-trade API(4번 주소)가 mPnL 값을 빈 문자열 "로 반환
- mPnL은 개인 포지션 손익 → 공개 API로는 값이 안 옴 (계정 API 키 필요)
- 데몬 코드에서 `$$["mPnL"] ??` 로 빈 문자열 저장
- 원장은 varchar(50) 이라 빈 문자열 저장 가능
- 복사 테이블은 decimal(30,16) → strict mode가 빈 문자열을 에러로 차단

## 3. 최종 해결책

이벤트 정의문에서 mPnL을 NULLIF로 변환:

```
NULLIF(mPnL, '') -- 빈 문자열을 NULL로 변환하여 decimal 컬럼에 안전하게 삽입
```

## 4. 교훈 및 주의사항

### 4-1. 이벤트 디버깅 방법

- SHOW EVENTS → LAST\_EXECUTED가 NULL이면 정의문 자체에 에러
- 이벤트 정의문을 SQL 탭에서 직접 실행해서 에러 확인

- 컬럼 그룹별로 나눠서 어느 컬럼이 문제인지 추적

## 4-2. strict mode 주의

- MariaDB STRICT\_TRANS\_TABLES: 빈 문자열을 숫자 컬럼에 넣으면 전체 INSERT 실패
- 원장이 varchar인 컬럼을 decimal 테이블로 복사할 때 반드시 NULLIF(col, "") 처리
- 특히 외부 API에서 값이 안 올 수 있는 컬럼은 반드시 확인

## 4-3. VIRTUAL/자동 컬럼

- VIRTUAL GENERATED 컬럼은 INSERT 목록에서 반드시 제외
- ON UPDATE CURRENT\_TIMESTAMP() 컬럼도 INSERT 목록에서 제외

## 4-4. 바이비트 API mPnL

- mPnL(미실현 손익)은 개인 포지션 데이터 → 공개 API로 수집 불가
- 계정 API 키 없이는 항상 빈 값 반환
- 데몬 코드에서 " 대신 null로 저장하는 것이 근본 해결책

## 5. 최종 이벤트 정의문

```
INSERT INTO daemon_bybit_coin_1m (
  symbol, korean_name, `time`,
  status, baseCoin, quoteCoin, settleCoin, optionsType,
  launchTime, priceScale,
  minLeverage, maxLeverage, leverageStep,
  minPrice, maxPrice, tickSize,
  maxOrderQty, minOrderQty, qtyStep, postOnlyMaxOrderQty,
  unifiedMarginTrade, fundingInterval, copyTrading,
  upperLimitPrice, lowerLimitPrice,
  lastPrice, indexPrice, markPrice,
  prevPrice24h, price24hPcnt, highPrice24h, lowPrice24h, prevPricel1h,
  openInterest, openInterestValue,
  turnover24h, volume24h,
  fundingRate, nextFundingTime,
  ask1Size, bid1Price, ask1Price, bid1Size,
  execId, price, size, side, isAdlTrade, mPnL, isBlockTrade,
  start, `open`, high, low, `close`, volume, turnover,
  m_close, i_close, p_close,
  buyRatio, sellRatio, period,
  riskId, isLowestRisk, maintenanceMargin, initialMargin, `limit`
)
SELECT
  symbol, korean_name,
  UNIX_TIMESTAMP(NOW(3))*1000,
  status, baseCoin, quoteCoin, settleCoin, optionsType,
  launchTime, priceScale,
  minLeverage, maxLeverage, leverageStep,
  minPrice, maxPrice, tickSize,
  maxOrderQty, minOrderQty, qtyStep, postOnlyMaxOrderQty,
  unifiedMarginTrade, fundingInterval, copyTrading,
  upperLimitPrice, lowerLimitPrice,
  lastPrice, indexPrice, markPrice,
  prevPrice24h, price24hPcnt, highPrice24h, lowPrice24h, prevPricel1h,
  openInterest, openInterestValue,
  turnover24h, volume24h,
```

```
fundingRate, nextFundingTime,  
ask1Size, bid1Price, ask1Price, bid1Size,  
execId, price, size, side, isAdlTrade,  
NULLIF(mPnL, ''), -- 핵심: 빈 문자열 → NULL 변환  
isBlockTrade,  
start, `open`, high, low, `close`, volume, turnover,  
m_close, i_close, p_close,  
buyRatio, sellRatio, period,  
riskId, isLowestRisk, maintenanceMargin, initialMargin, `limit`  
FROM daemon_bybit_Ticker;
```