# Mysql / MariaDB

- description : MariaDB, Mysql 관련 자료 정리
- author : 주레피
- email : dhan@repia.com
- lastupdate : 2020-02-08

## 주의 사항

- 대소문자 주의하는 것에 대해 고민하자

## FAQ

사용자 패스워드 변경하기

## 시작/종료

CentOS7

```
// 상태 보기
[mysql@dev tmp]$ systemctl status mariadb

// 시스템 부팅시 자동 시작
[mysql@dev tmp]$ systemctl enable mariadb

// MariaDB 재시작
[mysql@dev tmp]$ systemctl restart mariadb
```

## 백업, Dump

MariaDB 백업설정 및 데이터 이관
mysqldump 사용법
mysqldump 옵션
[Project :: Server] MariaDB 백업설정 및 데이터 이관

## 사용자 관리(생성, 조회, 수정, 삭제)

## Mysql DB

user table 사용자 추가

```
INSERT INTO mysql.`user`
(Host, `User`, Password, Select_priv, Insert_priv, Update_priv, Delete_priv,
Create_priv, Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv,
Grant_priv, References_priv, Index_priv, Alter_priv, Show_db_priv,
Super_priv, Create_tmp_table_priv, Lock_tables_priv, Execute_priv,
Repl_slave_priv, Repl_client_priv, Create_view_priv, Show_view_priv,
Create_routine_priv, Alter_routine_priv, Create_user_priv, Event_priv,
Trigger_priv, Create_tablespace_priv, ssl_type, ssl_cipher, x509_issuer,
x509_subject, max_questions, max_updates, max_connections,
max_user_connections, plugin, authentication_string, password_expired,
is_role)
VALUES('%', '__user__', password('__password__'), 'Y', 'Y', 'Y', 'Y', 'Y',
'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', '', '', '', '', 0, 0, 0, 0, '',
'', 'N', 'N')
;
```

db table 사용자 추가

```
INSERT INTO mysql.db
(Host, Db, `User`, Select_priv, Insert_priv, Update_priv, Delete_priv,
Create_priv, Drop_priv, Grant_priv, References_priv, Index_priv, Alter_priv,
Create_tmp_table_priv, Lock_tables_priv, Create_view_priv, Show_view_priv,
Create_routine_priv, Alter_routine_priv, Execute_priv, Event_priv,
Trigger_priv)
VALUES('%', '__db__', '__user__', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y',
'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
```

적용하기

```
FLUSH privileges;
```

## Maria DB

사용자 생성

```
MariaDB [(none)]> create user '${User }'@'${Host }' identified by
'${Password }';
Query OK, 0 rows affected (0.00 sec)
```

EX : create user 홍길동@'127.0.0.1' identified by '홍길동12';
Host : 허용 IP
Tip : 허용 IP는 보통 3가지를 등록한다.

- 홍길동@'127.0.0.1'
- 홍길동@'localhost'
- 홍길동@'내 IP' | 외부로 나가는 회사 IP

## 사용자 권한 변경(${DB }내 모든 테이블)

```
MariaDB [(none)]> grant all privileges on ${DB }.* to '${User }'@'${Host }';
Query OK, 0 rows affected (0.00 sec)
```

EX : grant all privileges on repia.* to '홍길동'@'localhost';
Tip : grant all - 모든권한

- all 대신 select, update, create 등이 있다.

```
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Tip : 권한 바꾸고 쿼리로 데이터를 수정하고 등록했던 모든 것을 메모리에 올리는 개념. 마지막에 꼭 해주는 것이 좋다.

링크

- 개발노트 :: MySQL 사용자 권한 확인 및 추가
- MariaDB 10.4 부터 변경된 user 테이블

## LOCK, UNLOCK

```
ALTER USER '${User }'@'${Host }' ACCOUNT LOCK;
ALTER USER '${User }'@'${Host }' ACCOUNT UNLOCK;
```

MySQL, MariaDB의 DB, 계정, 권한 생성 및 설정
[mysql] mysql, mariaDB 계정 생성, 권한 추가
Account Locking

## 쿼리 대소문자 구분

Windows에서는 잘 되던 것이 리눅스 환경으로 배포하면 가끔 테이블을 못 찾는다는 에러가 뜬다.
`Table ${Table name} doesn't exist`
그 이유는
Windows에서는 디렉토리와 파일에 접근할 때, 대소문자를 구분하지 않지만 Linux계열은 구분한다.
즉, Linux계열에서 쿼리는 대소문자를 정확히 해서 테이블명과 컬럼명을 찾아야한다.

> 해결 방법

- mariadb를 기본 설치했다면 `/etc/my.cnf` 파일에 `lower_case_table_names = 1`를 추가해야 된다.

| value | meaing |
|---|---|
| 0 | CREATE TABLE 이나 CREATE DATABASE 실행시 디스크에 저장되는 테이블과 데이타베이스의 이름을 대소문자를 구분해서 생성한다.<br>SELECT 나 Insert 사용시에도 대소문자를 구분해서 사용해야 한다. 대소문자를 구별하는 OS 에서만 의미가 있고 Windows/Mac OS X 에 는 적용되지 않는다 |
| 1 | 테이블과 DB 이름을 소문자로 생성하며 참조시에는 소문자로 변경하여 처리한다. 기존에 대문자가 포함되어 생성한 테이블과 DB 는 문제가 될 수 있다. |
| 2 | CREATE TABLE 이나 CREATE DATABASE 실행시 디스크에 저장되는 테이블과 데이타베이스의 이름을 대소문자를 구분해서 생성한다.<br>참조시에는 소문자로 변경한다. 대소문자를 구분하지 않는 파일 시스템을 가진 OS(Mac OS X) 에서만 동작한다. |

위 표를 보면 알듯이 설정을 한 후에 데이터가 생성되어야 하기 때문에
다시 DATABASE를 지우고 새로 생성해야 한다.
생성한 다음 dump 파일을 넣어 주면 된다.

# 인덱스 관리 (생성, 조회, 삭제)

```
// 조회 (SHOW INDEX FROM 테이블명)
$> SHOW INDEX FROM HEQMS.TB_HFDMS2_STD_CMPTN_SD;
// 삭제 (ALTER TABLE 테이블명 DROP INDEX 인덱스명)
$> ALTER TABLE HEQMS.TB_HFDMS2_STD_CMPTN_SD DROP INDEX
IX_TB_HEQMS2_STD_JOB_CMPTN_SD_01;
// 생성 (CREATE INDEX 인덱스명 ON 테이블명 (필드명 리스트))
$> CREATE INDEX IX_TB_HFDMS2_STD_CMPTN_SD_01 USING BTREE ON
TB_HFDMS2_STD_CMPTN_SD (STD_NO);
$> CREATE INDEX IX_TB_HFDMS2_STD_CMPTN_SD_02 USING BTREE ON
TB_HFDMS2_STD_CMPTN_SD (STD_NO, DGNSS_ROUND);
```

> 인덱스 이름 변경은 잘 안됨

# 함수(function) 생성

프로시져, 뷰 확인 방법

```
[프로시져 목록 확인방법]
mysql> SHOW PROCEDURE STATUS

[프로시져 스크립트 확인방법]
mysql> SHOW CREATE PROCEDURE uso_Get_Mini_Evaluation

[뷰 스크립트 확인방법]
mysql> SHOW CREATE VIEW v_core_ability_examination
```

MySQL 프로시져, 뷰 확인방법

### 예제

- ID,CD 값으로 해당 NAME을 조회하는 함수
- delimiter(구문문자) 사용하여 함수생성문 안에서 문장의 구분이 쉽도록 설정한다 (필수 X)

```
delimiter $$
DROP FUNCTION IF EXISTS dbname.getCodeIdByName $$
CREATE FUNCTION dbname.getCodeIdByName(findCode VARCHAR(100) ) RETURNS
VARCHAR(20)
   BEGIN
      DECLARE getName VARCHAR(20);
      SET getName = NULL;
      SELECT CODE_ID_DET  INTO getName FROM COM_CODE_DET WHERE CODE_NM_DET=
findCode;
      RETURN getName;
   END $$
delimiter ;
```

> SQL 작성후 DBeaver에서는 CTRL+ENTER(sql 실행)를 사용시 함수 적용이 안되므로 반드시 ALT + X(스크립트 실행) 를 사용하여 함수생성 결과를 확인한다.

MariaDB 업데이트

```
[mysql@dev tmp]$ mysql_upgrade -u root -p
```

## 등록된 DATA로 날짜 비교

TIMESTAMPDIFF

- 사용법
  - TIMESTAMPDIFF(단위, 날짜1, 날짜2);
- 단위
  - SECOND : 초
  - MINUTE : 분
  - HOUR : 시
  - DAY : 일
  - WEEK : 주
  - MONTH : 월
  - QUARTER : 분기
  - YEAR : 연
- 예시 (분)
  - 쿼리 : SELECT TIMESTAMPDIFF(MINUTE, '2017-03-01', '2018-03-28');
  - 결과 : 564480

## CONCAT 함수

> 예시

```
SELECT
  aei.ABILITYUNIT_EL_NM
  , group_concat(arr.RES_ROLES separator '|') AS RES_ROLES_RES
  , aei.AUE_CD
FROM
  ABILITYUNIT_INFO ai
GROUP BY aei.ABILITYUNIT_EL_NM
```

**위의 쿼리실행 시 여러 row로 있던 RES_ROLES의 값들이 한 row의 RES_ROLES_RES컬럼으로 select 된다.**

# 페이징

페이징 기본(둘다 같은 결과 나온다)

```
SELECT boardno, account, subject, content, createdate
FROM board
ORDER BY boardno DESC
LIMIT  0, 10
```

```
SELECT boardno, account, subject, content, createdate
FROM board
ORDER BY boardno DESC
LIMIT 10 OFFSET 0
```

# 테이블 복사

스키마(SCHEMA) 복사

```
CREATE TABLE ${NEW_TABLE} LIKE ${OLD_TABLE}
```

스키마(SCHEMA)와 데이터 복사

```
CREATE TABLE ${NEW_TABLE} ( SELECT * FROM ${OLD_TABLE} )
```

데이터 복사

```
INSERT INTO ${DEST_TABLE} ( SELECT * FROM ${SOURCE_TABLE} )
```

# Function & Procedures

- 유용한 사용자 정의 함수들

## Encoding

- latin1에서 utf8mb4로 변환하기

## Logging

log파일_설정

## Migration To Oracle

SQL Developer로 하면 됨 autoincrement는 자동으로 시퀀스를 생성해 줌

- MySQL to Oracle Migration

## Trouble Shooting

- TIMEZONE error
- 원격 IP로 MySQL(MariaDB)에 접속이 되지 않을 때의 해법
- |MYSQL(MariaDB)에서 외부접근이 되지않을때.(Feat. Can’t Connect To MySQL Server On ‘192.168.X.X'(10061)

## Ref

- MySQL 테이블 구조 복사 및 데이터 복사 ( Table Copy )
- MariaDB 서비스 포트 변경 (CentOS 7)
- CentOS7 MariaDB 설치
- RHEL/CentOS, Ubuntu 에 MySQL 5.6, 5.7 설치하기
- 게시판-페이징-완성 [PostIT]
- [MariaDB] root 비밀번호 분실시 초기화 방법
- MySQL root 패스워드 분실
- SQL Developer에서 MySQL 연결하기
- [MySQL] 날짜 차이 가져오기 (DATEDIFF, TIMESTAMPDIFF 함수)
- MySQL ( MariaDB ) 데이터베이스 또는 테이블의 대소문자 구분

주레피, 도봉산핵주먹, 천호동밤안개, eleven, mariadb, mysql

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다.** - 2023.12

Permanent link:
**http://125.132.25.164/dokuwiki/doku.php?id=wiki:database:mariadb&rev=1652833960**

Last update: **2022/05/18 09:32**