

학습목표

1. 메타데이터
2. 캐릭터셋
3. 스토리지엔진

학습내용

- 메타데이터(METADATA)에 대해 알 수 있습니다.
- 캐릭터셋(CharacterSet)에 대해 알 수 있습니다.

사전퀴즈

1. 데이터사전(Data Dictionary)은 읽고 쓰는 것이 가능하다.

정답 : X

해설 : 데이터사전은 메타데이터를 저장하는 읽기 전용 DB이다.

2. 캐릭터셋으로 권장되는 것은 UTF-8이다.

정답 : O

해설 : 캐릭터셋(Character Set)은 문자 인코딩정보를 정하는 것이다. DB의 캐릭터셋으로 권장하는 것은 UTF-8이다.

수업

1. 메타데이터

메타데이터의 개념

- 데이터를 위한 데이터
- DB, 테이블의 스키마에 대한 정보를 저장하는 테이블
- DB명, 테이블명, 컬럼명, 사용자명, SHOW 명령어의 결과값 ...
- 메타데이터는 크게 데이터 사전, 데이터 디렉토리로 되어 있음

데이터 사전(Data Dictionary)

- MariaDB에서는 Information_schema 또는 mysql이라고 불리는 DB
- 데이터베이스의 정보 저장
- 시스템 카탈로그(System Catalog)라고도 함
- 일반적으로 읽기전용정보(Read-only)

데이터 디렉토리(Data Directory)

- DBMS의 모든 데이터가 저장되는 디렉토리(폴더)
- DB저장, 상태 및 로그저장
- 윈도우용 MariaDB에서는 C:\Program Files\MariaDB 10.x\data에 데이터가 저장됨
my ini 파일에서 데이터가 저장되는 위치를 지정할 수 있음
예) datadir="C:/Sample/data"

실습

```
-- 데이터베이스 목록 확인
SHOW DATABASES;

-- information_schema DB 사용
USE information_schema;

-- 데이터베이스에 있는 테이블 목록 확인
SHOW TABLES;

-- information_schema DB에 있는 테이블 목록 확인
SHOW TABLES IN INFORMATION_SCHEMA;

-- 테이블 상태 확인
SHOW TABLE STATUS;

-- 테이블 컬럼명 확인
SHOW COLUMNS FROM 테이블명;

-- 테이블 인덱스 확인
SHOW INDEX FROM 테이블명;

-- information_schema DB의 character_set 확인
DESCRIBE INFORMATION_SCHEMA.CHARACTER_SETS;
```

INFORMATION_SCHEMA

INFORMATION_SCHEMA DB에 있는 테이블들의 목록 확인

```
-- 방법 1
USE information_schema;
SHOW TABLES;

-- 방법 2
SHOW TABLES IN INFORMATION_SCHEMA;
```

- CHARACTER_SETS (문자코드)
- COLLATIONS
- COLLATION_CHARACTER_SET_APPLICABILITY
- COLUMNS (컬럼정보들)
- COLUMN_PRIVILEGES (컬럼에 대한 권한들)
- KEY_COLUMN_USAGE (기본키들)
- REFERENTIAL_CONSTRAINTS (외래키에 대한 제약조건들)
- ROUTINES
- SCHEMATA
- STATISTICS
- TABLES (테이블들)
- TABLE_CONSTRAINTS (테이블에 대한 제약조건들)
- TRIGGERS (트리거)

- USER_PRIVILEGES (사용자권한)
- VIEWS

2. 캐릭터셋/콜레이션

캐릭터셋(Character Set)

- 문자인코딩 정보/메타데이터의 일종
- 문자열(VARCHAR, CHAR)의 값을 저장할 때 사용되는 기본정보
- DB/테이블별로 별도 설정 가능

ASCII / ISO-8859-1	아스키계열
EUC-KR / KSC_5601	한글 완성형 계열
UTF-8 / UNICODE	유니코드계열
UTF8 (기본)	

* 참고 *

unicode에서 파생된 것이 utf8이다. 즉, 큰 의미로 utf8은 unicode라고 봐도 된다.

unicode는 보통 2바이트로 한 글자를 표현하는데, 이 점은 ASCII코드와의 호환성에서 문제를 일으킨다. 왜냐하면 ASCII 코드는 1바이트로 한 글자를 표현하기 때문이다. 그래서 ASCII코드와 unicode 사이의 호환성 문제를 해결하기 위해 unicode를 1바이트 또는 멀티바이트(1~4바이트) 형태로 표현하는 utf8을 사용한다. utf8에서 영문은 1바이트, 한글은 3바이트로 표현된다.

콜레이션(Collation)

- 데이터를 정렬(문자간의 비교)할 때 사용하는 정보
- 정렬 시에 대소문자를 구분/비구분여부 설정(Case Sensitive / Insensitive)
- 한글데이터의 경우 무의미, 영문데이터를 저장할 때 주로 이슈가 되는 내용

utf8-general-ci (기본값 / 추천)

→ ci : case insensitive (대소문자 비구분)

→ utf8-general이면 대소문자 구분

utf8-unicode-ci

예제 2-1

MariaDB 기본 캐릭터셋 확인

-- 방법 1

SHOW status;

-- 방법 2

SHOW variables like '%character%';

예제 2-2

world 데이터베이스/city 테이블 캐릭터셋/콜레이션을 각각 utf8/utf8_general_ci로 설정하시오

```
-- world 데이터베이스 캐릭터셋을 utf8로, 콜레이션을 utf8_general_ci로 설정
ALTER database world character set utf8 collate utf8_general_ci;

-- world 데이터베이스 캐릭터셋 확인
SHOW CREATE DATABASE world;

-- city 테이블 캐릭터셋을 utf8로, 콜레이션을 utf8_general_ci로 설정
ALTER table city character set utf8 collate utf8_general_ci;

-- city 테이블 캐릭터셋 확인
SHOW CREATE TABLE city;
```

예제 2-3

sampleTestDB를 utf8/utf8_general_ci로 지정하고 생성하시오

```
CREATE database sampleTestDB default character set utf8 collate utf8_general_ci;
```

3. 스토리지 엔진(Storage Engine)

- 데이터베이스엔진(Database Engine)이라고도 불림
- DBMS가 데이터를 CRUD(Create/Retrieve/Update/Delete)할 때 사용하는 기본 컴포넌트
- 대표적으로 MyISAM(Aria)과 InnoDB(XtraDB)등이 있음
 - MyISAM : MySQL에서 사용
 - Aria : MyISAM에서 파생된 엔진. MariaDB에서 사용
 - InnoDB : 오라클/MySQL엔진. 오라클이 인수한 InnoDB회사의 엔진
 - XtraDB : InnoDB에서 파생된 엔진. MariaDB에서 사용
- 데이터 접근속도/안정성/트랜잭션의 지원 여부 등의 차이가 있음
- 기본값은 InnoDB

특징	MyISAM	InnoDB
구조의 복잡성	간단 (데이터모델 디자인 또한 간단)	복잡 (데이터모델 디자인 또한 복잡)
트랜잭션	없음	데이터 무결성 보장
Locking-level	테이블 단위	줄 단위
기능지원	B-Tree, R-Tree, Full-Text - Full-Text 인덱스란? 검색엔진과 유사한 방법으로 자연언어를 이용해 검색할 수 있는 특별한 인덱스로 모든 데이터 문자열의 단어를 저장한다. * 참고 * http://interconnection.tistory.com/95	B-Tree, Clustered, (R-Tree, Full-Text) - MySQL 5.7이상부터는 InnoDB도 R-Tree, Full-Text 인덱스를 지원하지만 MySQL 5.7이하 또는 MariaDB에서는 지원하지 않음
외래키지원	지원안함	지원
CRUD 속도	select 쿼리 속도가 빠름	insert, update, delete시에 빠른 속도 지원

- 테이블에 ROW COUNT를 가지고 있고, full-text 인덱스를 지원하기 때문

출처

<http://photalk.tistory.com/14>

<http://rebecca.tistory.com/14#recentComments>

예제 3-1

MariaDB 기본 스토리지 엔진 확인

* 참고 *

XA : 글로벌 트랜잭션 (여러 시스템간의 트랜잭션)

```
1 -- 쿼리 1
2 select *
3 from information_schema.engines
4 ;
5
6 -- 쿼리 2
7 show engines
8 ;
```

ENGINES (6x8)					
ENGINE	SUPPORT	COMMENT	TRANSACTIONS	XA	SAVEPOINTS
CSV	YES	Stores tables as CSV files	NO	NO	NO
MRG_MyISAM	YES	Collection of identical MyISAM tables	NO	NO	NO
MEMORY	YES	Hash based, stored in memory, useful for temporary tables	NO	NO	NO
MyISAM	YES	Non-transactional engine with good performance and small data footprint	NO	NO	NO
SEQUENCE	YES	Generated tables filled with sequential values	YES	NO	YES
InnoDB	DEFAULT	Supports transactions, row-level locking, foreign keys and encryption for tables	YES	YES	YES
Aria	YES	Crash-safe tables with MyISAM heritage	NO	NO	NO
PERFORMANCE_SCHEMA	YES	Performance Schema	NO	NO	NO

```
SELECT engine, support FROM information_schema.engines WHERE support='DEFAULT';
```

예제 3-2

기본 스토리지엔진을 변경하시오

```
-- 리부팅 시 리셋됨
SET default_storage_engine = MyISAM;

-- 영구적으로 변경하고 싶은 경우 my.ini 파일 수정
-- 파일 변경 후 MariaDB 재시작 필요
[mysqld]
default-storage-engine=MyISAM
```

스토리지 엔진에 따른 차이점

전문가의견

MariaDB는 스토리지 엔진을 선택할 수 있도록 되어 있습니다. 용도 별로 적합한 엔진을 선택하는 것이 MariaDB 중고급 사용자로 성장하는데 중요한 영향을 줍니다.

기본적인 스토리지 엔진에는 InnoDB와 MyISAM이 있습니다. 이외에도 memory, NDB 등의 엔진을 지원하고 써드파티의 스토리지 엔진도 존재합니다. 스토리지 엔진은 테이블별로 각각 지정할 수 있고 엔진 별로 트랜잭션 지원여부/동시사용성/락지원/복구가능성에 따라 다양한 차이를 보입니다. 이 외에도 파티셔닝이나 전문검색(Full-Text Search), R-트리 인덱스 적용 여부에도 스토리지 엔진과 MariaDB 버전에 따라 가능 여부가 다르니 해당 기능을 사용할 때 반드시 확인해야 합니다.