학습목표

1. INDEX

학습내용

- 30만개 이상의 샘플 DB에서 인덱스(INDEX)의 효과를 학습합니다.

사전퀴즈

1. launchpad.net/test-db는 유료 샘플 데이터이다.
정답 : X
해설 : launchpad.net/test-db는 무료 샘플 데이터입니다.

2. 인덱스의 효과는 데이터 크기에 무관하다.
정답 : X
해설 : 인덱스 효과는 데이터가 커지면 더욱 커지고 데이터가 작을 경우나 업데이트가 잦을 경우에는 미미하다.

수업

1. 대형 테이블 인덱스

- 테이블당 30만 개의 레코드(총 6개)
  https://launchpad.net/test-db
- 두 번째 다운로드 링크 클릭/링크복사
  wget https://launchpad.net/test-db/employees-db-1/1.0.6/+download/employees_db-full-1.0.6.tar.bz2
- bzip2 프로그램 설치 및 압축해제(ubuntu)
  apt-get update
  apt-get install bzip2
  bzip2 -d employees-db.bz2
  tar xvf employees-db
  cd employee-db
- 적재
  mysql -uroot -p < employees.sql

인덱스(INDEX) 종류

- B-트리 인덱스(기본)
- R-트리 인덱스
  → 공간정보 검색 (거리, 위치 등)
  예를 들어, gps의 특정 위치를 기준으로 일정 거리 안에 있는 모든 레코드들을 검색할 때 사용
  → 보통 InnoDB에서는 제공하지 않음(MySQL 5.7이상부터는 InnoDB에서도 R-트리 인덱스 지원)
  MyISAM/Aria에서는 제공함
- 해시 인덱스
  → 메모리와 같은 작은 데이터셋에 효과적
  → 테이블을 수정하는 쿼리(insert/update/delete)보다는 select 쿼리가 많은 테이블에 효과적
- 프랙탈 트리 인덱스

엔진별 지원 인덱스

DB버전별로 인덱스 지원 사항이 다르므로 사용하기 전에 확인필수

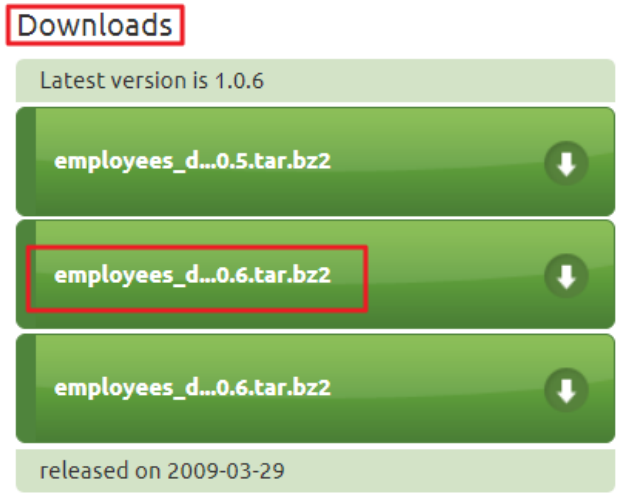
예를 들어, 기존에는 InnoDB에서는 R-Tree를 사용하지 못한다고 되어 있었지만 MySQL 5.7이상부터는 지원함

- MyISAM(Aria)
  → B-Tree, R-Tree
- InnoDB(XtraDB)
  → B-Tree
- Memory
  → B-Tree, Hash
- NDB
  → Hash, B-Tree

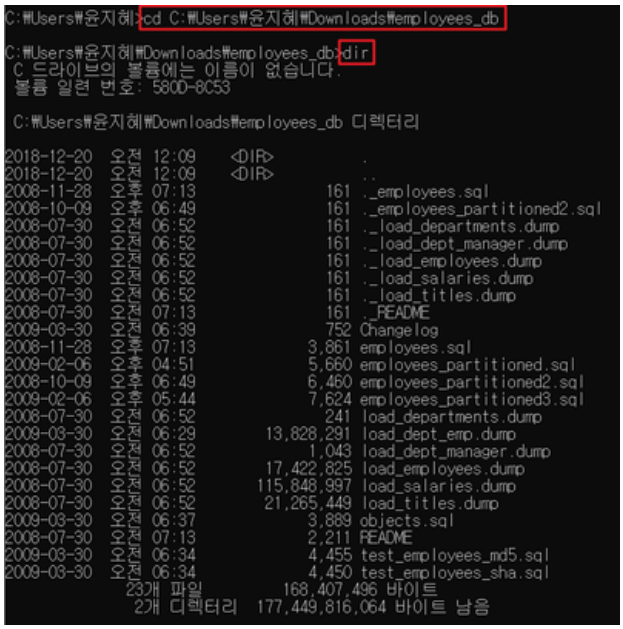
2. 실습

1) https://launchpad.net/test-db 사이트 접속

2) 두 번째 다운로드 링크 클릭하여 파일 다운로드



3) 다운로드 받은 파일 압축풀기



4) mariaDB 접속 후, employees.sql 파일 DB에 적재

```
MariaDB [(none)]> source employees.sql
Query OK, 0 rows affected, 1 warning (0.001 sec)

Query OK, 1 row affected (0.014 sec)

Database changed
+-----------------------------+
| INFO                        |
+-----------------------------+
| CREATING DATABASE STRUCTURE |
+-----------------------------+
1 row in set (0.000 sec)

Query OK, 0 rows affected, 6 warnings (0.001 sec)
```

5) 데이터 베이스 목록 확인

employees 데이터베이스 선택 후, 테이블 목록 확인

```
MariaDB [employees]> show databases;
+--------------------+
| Database           |
+--------------------+
| employees          |
| information_schema |
| mysql              |
| performance_schema |
| sampledb           |
| test               |
| world              |
+--------------------+
7 rows in set (0.004 sec)

MariaDB [employees]> use employees;
Database changed
MariaDB [employees]> show tables;
+----------------------+
| Tables_in_employees  |
+----------------------+
| departments          |
| dept_emp             |
| dept_manager         |
| employees            |
| salaries             |
| titles               |
+----------------------+
6 rows in set (0.001 sec)
```

6)employees 테이블 레코드 개수 확인

```
MariaDB [employees]> select count(*) from employees;
+----------+
| count(*) |
+----------+
|   300024 |
+----------+
1 row in set (0.084 sec)
```

7) 인덱스가 생성되어 있는 컬럼(emp_no)을 사용한 검색시간과

인덱스가 생성되어 있지 않은(last_name) 컬럼을 사용한 테이블 검색시간 차이 확인

```
MariaDB [employees]> select * from employees where emp_no = 10005;
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
|  10005 | 1955-01-21 | Kyoichi    | Maliniak  | M      | 1989-09-12 |
+--------+------------+------------+-----------+--------+------------+
1 row in set (0.001 sec)
```

```
MariaDB [employees]> select * from employees where last_name='Facello';
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
|  10001 | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 |
|  10327 | 1954-04-01 | Roded      | Facello   | M      | 1987-09-18 |
| 487188 | 1959-02-20 | Richara    | Facello   | M      | 1985-08-31 |
| 488569 | 1955-11-15 | Satoru     | Facello   | M      | 1993-03-20 |
| 493549 | 1960-04-17 | Achilleas  | Facello   | F      | 1991-11-27 |
+--------+------------+------------+-----------+--------+------------+
186 rows in set (0.156 sec)
```

8) last_name 컬럼에 대해 인덱스 생성

레코드가 30만개가 넘어가기 때문에 인덱스 생성하는데 시간이 걸림

```
MariaDB [employees]> create index employees_lastname_idx on employees(last_name);
Query OK, 0 rows affected (1.838 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

9) last_name 컬럼을 사용한 테이블 검색시간 확인 (인덱스를 사용했을 때와 사용하지 않았을 때의 테이블 검색시간 비교)

인덱스를 사용하지 않았을 때 : 0.156초

인덱스를 사용했을 때 : 0.001초

```
MariaDB [employees]> select * from employees where last_name='Facello';
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
|  10001 | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 |
|  10327 | 1954-04-01 | Roded      | Facello   | M      | 1987-09-18 |
| 488569 | 1955-11-15 | Satoru     | Facello   | M      | 1993-03-20 |
| 493549 | 1960-04-17 | Achilleas  | Facello   | F      | 1991-11-27 |
+--------+------------+------------+-----------+--------+------------+
186 rows in set (0.001 sec)
```

10) 인덱스가 걸려있는 컬럼(last_name)과 인덱스가 걸려있지 않은 컬럼(first_name)을 동시에 사용하여 검색

first_name 컬럼에는 인덱스가 걸려있지 않으므로 컬럼 앞쪽에 있는 Georgi와 같은 레코드는 비교적 빠르게 찾겠지만 Mary와 같은 레코드를 찾을 때는 시간이 오래 걸릴 수 있다. 그런데 아래 결과를 보면 시간이 차이가 나지 않는 것을 볼 수 있다. 이것은 DBMS의 캐시/버퍼링 효과 때문이다. 그러므로 쿼리 연산시간을 볼 때 인덱스/캐시/버퍼링 효과를 구분해야 한다.

```
MariaDB [employees]> select * from employees where last_name='Facello' and first_name='Georgi';
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
|  10001 | 1953-09-02 | Georgi     | Facello   | M      | 1986-06-26 |
|  55649 | 1956-01-23 | Georgi     | Facello   | M      | 1988-05-04 |
+--------+------------+------------+-----------+--------+------------+
2 rows in set (0.001 sec)

MariaDB [employees]> select * from employees where last_name='Facello' and first_name='Mary';
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
| 486829 | 1953-10-11 | Mary       | Facello   | M      | 1998-02-13 |
+--------+------------+------------+-----------+--------+------------+
1 row in set (0.001 sec)
```

11) 복합인덱스(last_name, first_name) 추가

조회조건에 여러개의 컬럼이 동시에 사용되어졌을 때는 복합인덱스를 생성해야 한다.

*** 주의할 점 ***

인덱스가 항상 의미가 있는 것은 아니다. 예를 들어, 어떤 조건에 대한 쿼리를 한 번만 사용한다면 굳이 인덱스를 생성할 필요가 없다.

하지만 동일한 쿼리를 여러번 반복적으로 사용해야 할 경우에는 인덱스를 생성해야 한다.

```
MariaDB [employees]> create index employees_lastname_firstnameidx on employees(last_name, first_name);
Query OK, 0 rows affected (1.981 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

스토리지 엔진에 따른 인덱스의 차이점

전문가의견

MariaDB는 스토리지 엔진을 선택할 수 있도록 되어 있습니다.

기본 엔진 별로 사용할 수 있는 인덱스의 종류도 차이가 납니다.

B트리 인덱스는 InnoDB와 MyISAM 모두 가능하지만, 전문검색인덱스나 R트리 인덱스는 MyISAM만이 가능합니다.

버전 별로 지원여부가 모두 다르니 해당내용을 꼭 확인한 후에 사용하시기 바랍니다.