

[스마트 웹 & 앱 과정]

데이터베이스 구현

포트폴리오

김문선

2019.06.24

목차

1. 데이터베이스와 테이블 생성 및 테이블 내용 입력	-----	3
2. VIEW 생성	-----	4
3. 계정 생성 및 권한 설정	-----	5
4. sample1 계정으로 접속 및 VIEW 활용	-----	6
5. 권한 회수 및 확인, sample2 계정 삭제	-----	7
6. 데이터베이스 백업 및 삭제	-----	8
7. 임의의 데이터베이스 생성과 복원	-----	9

1. 데이터베이스를 생성하고 아래의 테이블을 생성한다.

(데이터베이스와 테이블은 임의로 설정한다. 이해를 돕기위해 테이블 이름을 CONAN으로 가정)

- 제약키를 알맞게 사용했다.
- 자료형을 알맞게 지정했다.
- 테이블의 구조를 파악할 수 있다.

[쿼리]

* CONAN 테이블 생성!!

```
create table CONAN (  
    ID int(20) not null primary key,  
    NAME char(30) ,  
    Height int(10),  
    TEL char(50),  
    ADDR char(30),  
    생년월일 date  
);
```

* CONAN 테이블 구조확인하기!!

```
desc CONAN;
```

* CONAN 테이블에 내용 입력!!

```
insert into CONAN values (1, '코난', 180, '011-1234-5678', '오사카', '2000-01-03');  
insert into CONAN values (2, '유미란', 175, '019-3333-4444', '도쿄', '1999-03-07');
```

[실행화면]

* “select”를 활용해 테이블 안에 내용이 잘 들어갔는지 확인!!

	ID	NAME	Height	TEL	ADDR	생년월일
▶	1	코난	180	011-1234-5678	오사카	2000-01-03
	2	유미란	175	019-3333-4444	도쿄	1999-03-07

* desc를 통한 “CONAN” 테이블 구조 확인!

Field	Type	Null	Key	Default	Extra
ID	int(20)	NO	PRI	NULL	
NAME	char(30)	YES		NULL	
Height	int(10)	YES		NULL	
TEL	char(50)	YES		NULL	
ADDR	char(30)	YES		NULL	
생년월일	date	YES		NULL	

[설명]

ID 칼럼은 사람마다 중복이 되지않도록 하기위해 기본키로 지정했으며 사람 식별이 가능해야하기때

문에 not null 을 설정하여 null값이 들어가지않게 했습니다. **NAME**은 이름 길이가 긴 사람들을 위해서 문자형 30 으로 지정하였으며 **Height**는 사람의 키이기 때문에 int 형으로 지정했습니다. **TEL**은 전화번호이지만 중간에 '-' 삽입이 되므로 문자형으로 지정했습니다. **ADDR**은 주소로 문자형이며 '시'까지 입력하는 것으로 보아 길이를 10까지만 지정해줄까하다가 30으로 지정해주었습니다. **생년월일**은 날짜이므로 date 형식으로 지정했습니다.

2. CONAN 테이블에서 일부 정보만 조회하여 VIEW를 생성한다.

- VIEW를 생성한 기준과 목적을 설명할 수 있다.
- 목적에 맞게 VIEW의 이름을 만들 수 있다.

[쿼리]

```
create view birthparty as
select NAME, monthname(생년월일) as '생일 월',
       year(current_date())-year(생년월일)+1 as '몇번째 생일',
       생년월일 from CONAN;
```

[실행화면]

NAME	생일 월	몇번째 생일	생년월일
코난	January	20	2000-01-03
유미란	March	21	1999-03-07

[설명]

- view를 생성한 기준과 목적

- :: 코난이라는 만화에서 코난은 어린 아이 몸으로 돌아가 검은조직을 피해서 유미란이랑 한가정에서 살게 됩니다. 원래 코난이랑 유미란은 커플 사이입니다. 특별편 만화에서 코난이 미란 생일을 까먹어서 공수부로 코난이 맞을 뻔한 에피소드가 떠올라서 해당 뷰를 작성하게 되었습니다. 생년월일을 깜빡 잊을 수 없게, 자기 나이를 알면 유미란 나이를 알 수도 있지만 몇번째 생일인지(자기 나이)도 알 수 있게 작성해봤습니다. 공수부로 맞지 않게. 어린아이 몸으로 돌아간 코난이 불쌍해보이므로 작성해봤습니다.

3. sample1과 sample2 계정을 생성하고 2번에서 생성한 VIEW에 대해 일부 권한을 부여한다.

- insert 문을 이용하여 권한을 설정할 수 있다.

- grant 문을 이용하여 권한을 설정할 수 있다.

[쿼리]

* 'sample1' 계정을 생성하고 **select** 권한만 부여한다.

```
grant select on portfolio.birthparty to sample1@localhost identified by '1234';
show grants for sample1@localhost;
flush privileges;
```

* 'sample2' 계정을 생성하고 **select, update, delete** 권한만 부여한다.

```
insert into user (host,user,password,ssl_type,ssl_cipher,x509_issuer,x509_subject)
values ("localhost","sample2",password("1234"),'',','','');
flush privileges;
/* 권한 설정 */
insert into db values ("localhost","portfolio","sample2","y","n","y","y",
"n","n","n","n","n","n","n","n","n","n","n","n","n","n","n","n","n","n","n","n");
```

[실행화면]

* “**show grants for sample1@localhost;**” 명령어로 **sample1** 계정 권한 확인

Grants for sample1@localhost
GRANT USAGE ON *.* TO 'sample1'@'localhost' IDENTIFIED BY PASSWORD '*A4B6157319038724E3560894F7...
GRANT SELECT ON `portfolio`.`birthparty` TO 'sample1'@'localhost'

* “**show grants for sample2@localhost;**” 명령어로 **sample2** 계정 권한 확인

Grants for sample2@localhost
GRANT USAGE ON *.* TO 'sample2'@'localhost' IDENTIFIED BY PASSWORD '*A4B6157319038724E3560894F7F932C8886EBFCF'
GRANT SELECT, UPDATE, DELETE ON `portfolio`.`*` TO 'sample2'@'localhost'

[설명]

:: sample1 계정을 grant로 사용하여 권한부여를 했으며 sample2 계정은 insert로 사용하여 권한을 부여했습니다.

4. sample1 계정으로 MySQL에 접속하여 VIEW를 활용한다.

- sample1 계정으로 MySQL에 접속할 수 있다.
- sample1 계정에 허용되지 않은 권한을 사용하였을 때의 결과를 확인할 수 있다.

[쿼리]

* sample1 계정으로 MySQL 접속!!
> *mysql -u sample1 -p*

[실행화면]

* 'sample1' 계정으로 mysql 접속!

```
C:\Users\User>mysql -u sample1 -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 5.6.44-log MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

* 'sample1' 계정에 부여되지 않은 update문 조회시 권한에러 메시지 표출됨.

```
mysql> select * from birthparty;
+-----+-----+-----+-----+
| NAME | 생일 월 | 몇 번째 생일 | 생년월일 |
+-----+-----+-----+-----+
| 코난 | January | 20 | 2000-01-03 |
| 유미란 | March | 21 | 1999-03-07 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> update birthparty set name = '코난' where name='올레';
ERROR 1142 (42000): UPDATE command denied to user 'sample1'@'localhost' for table 'birthparty'
mysql>
```

[설명]

'mysql -u sample1 -p' 명령어로 sample1 계정으로 접속했습니다. 그리고 sample1 계정은 'select' 조회 권한만 부여되어있으므로, 다른 명령어 'update', 'delete' 등 다른 명령어 실행 시 권한관련 오류가 발생합니다.

5. sample1의 일부 권한을 회수하고 적용된 내용을 확인한다. 또한 sample2 계정을 삭제한다.
- 권한 회수 및 내용 확인에 필요한 쿼리를 모두 사용할 수 있다.
 - 계정을 삭제하고 유효한 계정을 조회할 수 있다.

[쿼리]

- * sample1 계정에 있는 select 권한을 회수합니다.

```
revoke select on portfolio.birthparty from sample1@localhost;
```

- * show grants 명령어를 통해 권한이 회수되었는지 확인합니다.

```
show grants for sample1@localhost;
```

- * sample2 계정을 삭제합니다.

```
drop user sample2@localhost;
```

- * 유효한 계정이 무엇이 있는지 확인합니다.

```
select * from user;
```

[실행화면]

- * sample1 계정에 대한 권한 확인

(권한 회수 전...)

```
Grants for sample1@localhost
GRANT USAGE ON *.* TO 'sample1'@'localhost' IDENTIFIED BY PASSWORD '*A4B6157319038724E3560894F7...'
GRANT SELECT ON `portfolio`.`birthparty` TO 'sample1'@'localhost'
```

(권한 회수 후...)

```
Grants for sample1@localhost
GRANT USAGE ON *.* TO 'sample1'@'localhost' IDENTIFIED BY PASSWORD '*A4B6157319038724E3560894F7...'
```

- * sample2 계정 삭제

(sample2 계정 삭제 전...)

Host	User	Password	Select_priv	Insert_priv	Update_priv	Delete_priv
localhost	moonsun	*A4B6157319038724E3560894F7F...	N	N	N	N
localhost	sample2	*A4B6157319038724E3560894F7F...	N	N	N	N
localhost	sample1	*A4B6157319038724E3560894F7F...	N	N	N	N

(sample2 계정 삭제 후...)

Host	User	Password	Select_priv	Insert_priv	Update_priv	Delete_priv
localhost	parttimePerson	*89C6B530AA78695E257E55D63...	N	N	N	N
localhost	moonsun	*A4B6157319038724E3560894F7F...	N	N	N	N
localhost	sample1	*A4B6157319038724E3560894F7F...	N	N	N	N

[설명]

‘sample1’ 계정에 있던 select 권한을 회수하고 나니 usage 내용만 남겨진 것이 확인됩니다. drop 명령어를 사용하여 ‘sample2’ 계정을 삭제했으며, “select * from user;” 명령어를 사용하여 유효한 계정을 확인했습니다.

6. CONAN 테이블이 속한 데이터베이스를 백업하고 삭제한다.

- 데이터베이스 백업 시 경로를 올바르게 지정할 수 있다.
- 백업된 폴더의 위치를 확인할 수 있다.
- 데이터베이스를 삭제할 수 있다.

[쿼리]

```
> mysqldump -u root -p portfolio > C:\portfoliodumpfile\portdump.sql
```

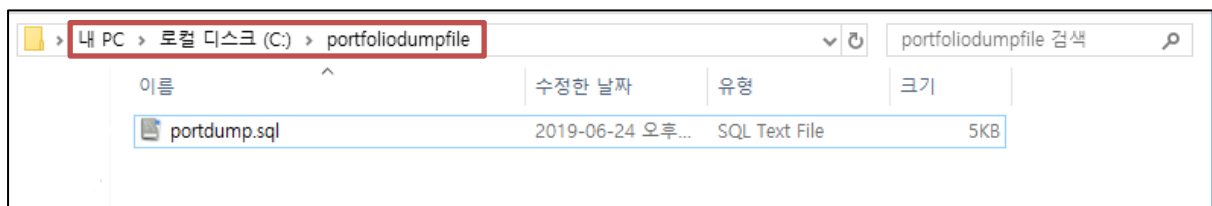
```
C:\Users\Wuser>cd C:\Program Files\MySQL\MySQL Server 5.6\bin
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqldump -u root -p portfolio > C:\portfoliodumpfile\portdump.sql
Enter password: ****
C:\Program Files\MySQL\MySQL Server 5.6\bin>
```

```
> drop databases portfolio;
```

```
drop database portfolio;
```

[실행화면]

* 지정된 경로로 데이터베이스가 백업되었는지 확인합니다.



* 'portfolio' 데이터베이스가 삭제되었는지 "show databases;" 를 통해 확인합니다.

Database
information schema
bookstore
mysql
mytest
performance schema
sakila
sample
test
world

[설명]

DB를 백업하기 위해서 'C:\Program Files\MySQL\MySQL Server 5.6\bin' cmd에서 'cd' 명령어를 사용해 해당 경로로 이동합니다. 백업 쿼리를 실행하고 지정해놓은 경로로 이동합니다. 해당 경로 (폴더)를 보면 portfolio 데이터베이스가 'portdump.sql' 파일로 백업이 된 것을 확인할 수 있습니다.

백업 후, portfolio 데이터베이스를 삭제하기 위해 'drop database portfolio;' 명령어를 사용합니다. 데이터베이스가 완전히 삭제되었는지 확인하기 위해 'show databases' 명령어를 사용하여 확인합니다.

7. 임의의 데이터베이스를 만들고 6번에서 백업한 데이터베이스를 복원한다.

- 복원을 위한 데이터베이스를 생성할 수 있다.
- 생성된 데이터베이스에 올바른 데이터베이스를 복원할 수 있다.

[쿼리]

* 6번에서 백업한 디비파일을 복원시키기 위한 DB를 생성함.

```
create database backupdb default character set utf8
                        default collate utf8_general_ci;
```

* 생성된 데이터베이스에 복원.

```
C:\Users\User>mysql -u root -p backupdb < C:\portfoliodumpfile\portdump.sql
Enter password: ****
```

[실행화면]

* 'show databases;' 명령어를 사용해 'backupdb' 생성된 부분을 확인..

Database
information schema
backupdb
bookstore
mysql
mytest
performance schema
sakila

* 'show tables;' 와 'select 문' 으로 제대로 복원되었는지 확인한 화면

<pre>mysql> use backupdb; Database changed mysql> show tables; +-----+ Tables_in_backupdb +-----+ birthparty birthparty_person conan +-----+ 3 rows in set (0.00 sec)</pre>	<pre>mysql> select * from conan; +----+-----+-----+-----+-----+-----+ ID NAME Height TEL ADDR 생년월일 +----+-----+-----+-----+-----+-----+ 1 코난 180 011-1234-5678 오사카 2000-01-03 2 유미란 175 019-3333-4444 도쿄 1999-03-07 +----+-----+-----+-----+-----+-----+ 2 rows in set (0.00 sec)</pre>
--	--

[설명]

백업한 디비를 복원하기 위해 임의의 'backupdb' 데이터베이스를 생성하고 해당 DB로 'portdump.sql' 파일을 복원합니다. 그리고 해당 백업파일이 잘 들어갔는지 show tables 와 select로 해당 내용이 들어갔는지 확인합니다.