

## 7주차 3차시 일괄 실행, 백업, 복원과 뷰

### 【학습목표】

1. MySQL 명령의 일괄 실행 방법을 설명할 수 있다.
2. 뷰의 개념과 장점을 알아보고, 뷰를 생성하고 삭제하는 방법을 설명할 수 있다.

### 학습내용1 : MySQL 명령의 일괄 실행

- MySQL Server 내에서 사용하는 명령어를 파일에 저장하여 일괄 처리
- 파일의 확장자는 .sql 로 지정
- 모든 명령을 파일에 저장하여 일괄처리함.
- 데이터베이스를 백업한 후 복원하기 위하여 일괄처리함.
- Windows의 명령 프롬프트 창에서 실행한다.

#### A. 테이블 생성 일괄처리

- 편집기를 사용하여 CREATE TABLE 명령을 파일에 저장
- 명령어 : mysql -u계정 -p비밀번호 데이터베이스명 < 일괄처리파일명

### 표. Customer 테이블 생성을 위한 customer.sql 파일

```
CREATE TABLE Customer (  
    account    varchar(10) NOT NULL,  
    name       varchar(10) NOT NULL,  
    grade      varchar(6) NOT NULL DEFAULT 'New',  
    credit      int DEFAULT 0,  
    address    varchar(100),  
    PRIMARY KEY ( account )  
);
```



### 그림. Customer 테이블 생성을 위한 일괄처리전

```
mysql -uroot -p itbank < customer.sql
```

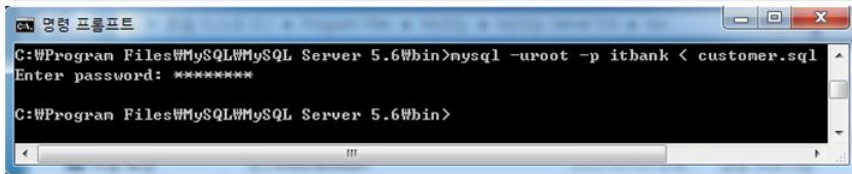


그림. Customer 테이블 생성을 위한 일괄처리

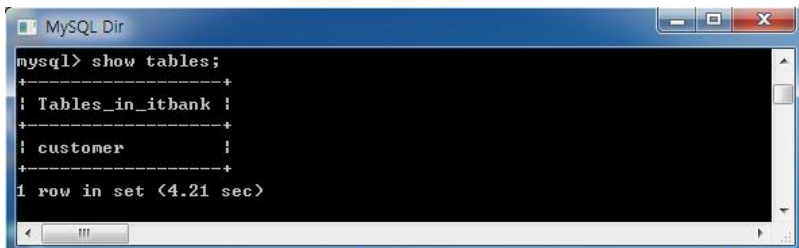


그림. Customer 테이블 생성을 위한 일괄처리후

#### B. 데이터베이스 백업

- DBMS 내의 데이터 백업은 데이터베이스 별로 한다.
- 테이블 별로 백업 할 수 없다.
- 명령어 : `mysqldump -u계정 -p비밀번호 데이터베이스명 > 백업파일명`

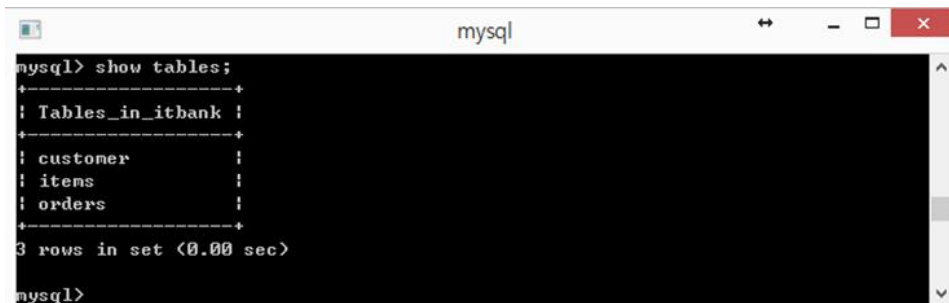
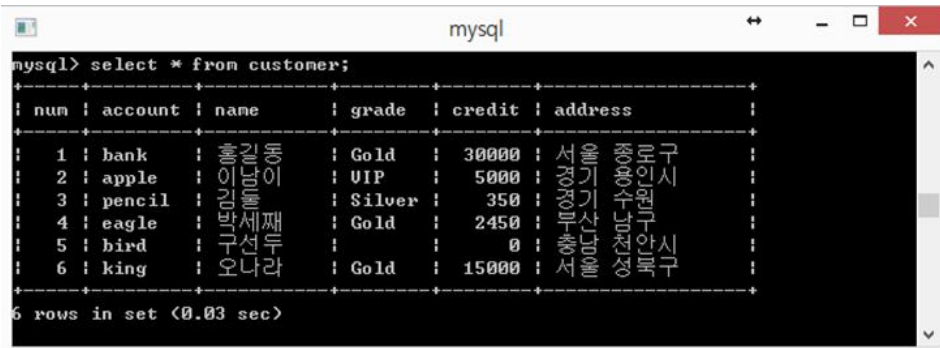


그림. itbank 데이터베이스에 있는 테이블

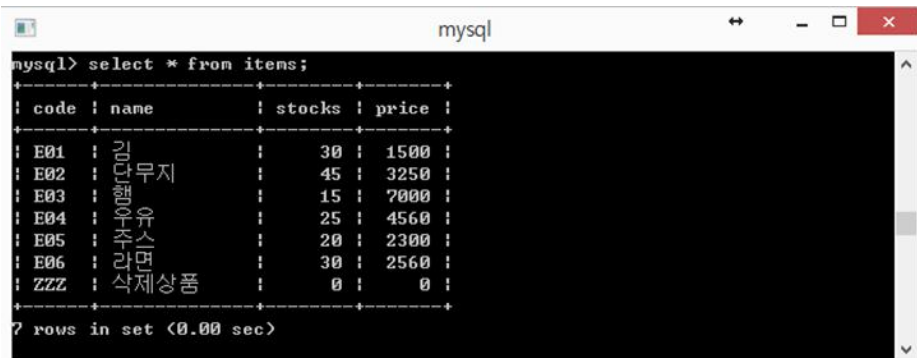


```
mysql> select * from customer;
```

num	account	name	grade	credit	address
1	bank	홍길동	Gold	30000	서울특별시 강남구
2	apple	이밤이	VIP	5000	경기 용인시
3	pencil	김돌	Silver	350	충청남도 천안시
4	eagle	박세채	Gold	2450	부산광역시
5	bird	구선두		0	충남 천안시
6	king	오나라	Gold	15000	서울특별시

6 rows in set (0.03 sec)

그림. customer 테이블에 있는 레코드

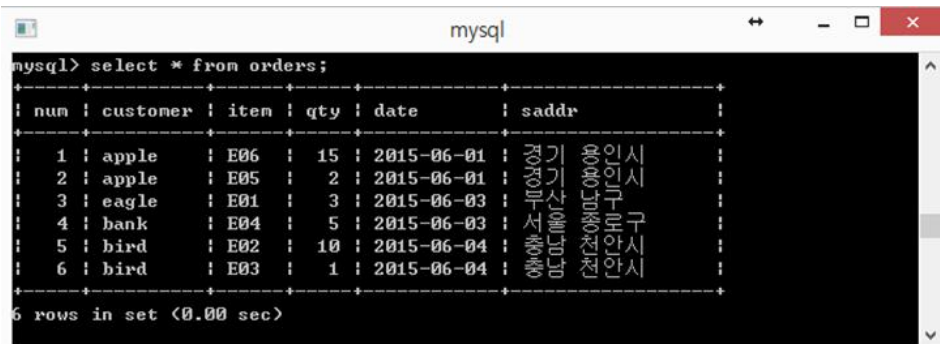


```
mysql> select * from items;
```

code	name	stocks	price
E01	김 단무지	30	1500
E02	햄	45	3250
E03	우유	15	7000
E04	추스	25	4560
E05	라면	20	2300
E06	식제상품	30	2560
ZZZ		0	0

7 rows in set (0.00 sec)

그림. items 테이블에 있는 레코드



```
mysql> select * from orders;
```

num	customer	item	qty	date	saddr
1	apple	E06	15	2015-06-01	경기 용인시
2	apple	E05	2	2015-06-01	경기 용인시
3	eagle	E01	3	2015-06-03	부산광역시
4	bank	E04	5	2015-06-03	서울특별시 강남구
5	bird	E02	10	2015-06-04	충남 천안시
6	bird	E03	1	2015-06-04	충남 천안시

6 rows in set (0.00 sec)

그림. orders 테이블에 있는 레코드

### C. 백업된 파일 살펴보기

- 복원할 데이터베이스에는 복원할 동일한 이름의 테이블이 있어도 된다.  
. 테이블이 있을 경우, 테이블이 자동 삭제되고 백업된 데이터로 테이블이 생성됨

#### 표. 백업된 itbank 데이터베이스 파일

```
DROP TABLE IF EXISTS `customer`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `customer` (
  `num` int(11) NOT NULL AUTO_INCREMENT,
  `account` varchar(10) NOT NULL,
  `name` varchar(10) NOT NULL,
  `grade` varchar(6) NOT NULL DEFAULT 'New',
  `credit` int(11) DEFAULT '0',
  `address` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`num`),
  UNIQUE KEY `account` (`account`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

#### 표. 백업된 itbank 데이터베이스 파일

```
LOCK TABLES `customer` WRITE;
/*!40000 ALTER TABLE `customer` DISABLE KEYS */;
INSERT INTO `customer` VALUES
(1,'bank','홍길동','Gold',30000,'서울 종로구'),
(2,'apple','이남이','VIP',5000,'경기 용인시'),
(3,'pencil','김둘','Silver',350,'경기 수원'),
(4,'eagle','박세째','Gold',2450,'부산 남구'),
(5,'bird','구선두','',0,'충남 천안시'),
(6,'king','오나라','Gold',15000,'서울 성북구');
/*!40000 ALTER TABLE `customer` ENABLE KEYS */;
UNLOCK TABLES;
```

#### [참고] LOCK TABLES

- 형식

#### 표. LOCK TABLES 형식

```
LOCK TABLES 테이블명 Lock_type
[ , 테이블명 Lock_type ] ...

Lock_type :
  READ [LOCAL] | WRITE

UNLOCK TABLES
```

- 테이블을 사용할 동안 다른 사람의 접근을 제한한다.
- READ  
. LOCK 한 사람이 테이블을 읽을 수 만 있고 기록할 수는 없다.  
. 다른 사람은 테이블을 읽거나 쓰기를 할 수 없다.

- READ LOCAL
  - . LOCK한 사람은 읽기만 가능하다.
  - . 다른 사람은 쓰기와 읽기가 가능하다.
- WRITE
  - . LOCK한 사람만 읽고 쓸수 있다.
  - . 다른 사람은 읽고 쓰기가 불가능하다.
- UNLOCK TABLES
  - . LOCK 한 테이블을 해제한다.

```
mysql> LOCK TABLES customer READ;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into customer values ( '', 'good', '도돌이', 'New', 0, '제주 서귀포시' );
ERROR 1099 (HY000): Table 'customer' was locked with a READ lock and can't be updated

mysql> select * from customer;
```

num	account	name	grade	credit	address
1	bank	홍길동	Gold	30000	서울 종로구
2	apple	이남이	VIP	5000	경기 용인시
3	pencil	김돌	Silver	350	경기 수원시
4	eagle	박세재	Gold	2450	부산 남구
5	bird	구선두		0	충남 천안시
6	king	오나라	Gold	15000	서울 성동구
100	oil	오오구	New	0	충남 서산

```
7 rows in set (0.00 sec)
```

그림. LOCK TABLES 테이블명 READ

```
mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into customer values ( '', 'good', '도돌이', 'New', 0, '제주 서귀포시' );
Query OK, 1 row affected, 1 warning (0.03 sec)
```

그림. UNLOCK TABLES

#### D. 데이터베이스 복원

- DBMS 내의 데이터 복원은 데이터베이스 별로 한다.
- 테이블 별로 복원할 수 없다.
- 복원할 데이터베이스는 DBMS내에 만들어져 있어야 한다.
- 복원할 때 사용하는 계정은 해당 데이터베이스에 접근할 권한을 가지고 있어야 한다.
- 명령어 : `mysql -u계정 -p비밀번호 데이터베이스명 < 백업파일명`

## 학습내용2 : 뷰

### 1. 뷰 (View) 개념

- 다른 테이블을 기반으로 만들어진 가상 테이블
- 데이터를 실제로 저장하지 않음
- 논리적으로만 존재하는 테이블
- 일반 테이블과 동일한 방법으로 사용
- 다른 뷰를 기반으로 새로운 뷰를 만들 수 있음
- 뷰를 통해 기본 테이블의 내용을 쉽게 검색할 수 있음
- 기본 테이블의 내용을 변화시키는 작업은 제한적임

### 2. 뷰 생성

- 명령어 : CREATE VIEW

#### 표. CREATE VIEW 형식

```
CREATE VIEW 뷰이름 [ (속성리스트) ]  
AS SELECT 문  
[WITH CHECK OPTION];
```

- 속성리스트 : 뷰를 구성하는 속성의 이름
  - . 생략 가능 - SELECT 문에서 사용한 속성의 이름을 뷰에서 사용
- SELECT 문 : 기본 테이블에 대한 SELECT 문
  - . ORDER BY를 사용할 수 없음
  - . 일반적인 SELECT 문과 동일
- WITH CHECK OPTION
  - . 생성한 뷰에 삽입이나 수정 연산을 할 때, SELECT 문에서 제시한 뷰의 정의 조건을 위반하면 수행되지 않도록 하는 제약조건을 제시
- 예제 : Customer 테이블에서 등급이 VIP이거나 Gold 인 고객의 account, name, credit 으로 구성된 뷰를 GoodCustomer 이름으로 생성하고. 뷰의 내용을 검색하시오.
  - . 기본 테이블의 데이터



```
mysql> select * from customer;
```

num	account	name	grade	credit	address
1	bank	홍길동	Gold	30000	서울 중로구
2	apple	이남이	VIP	5000	경기 용인시
3	pencil	김돌	Silver	350	경기 수원시
4	eagle	박세재	Gold	2450	부산 남구
5	bird	구선두		0	충청 천안시
6	king	오나라	Gold	15000	서울 성북구
100	oil	오오구	New	0	충청 서산시
101	good	도돌이	New	0	제주 서귀포시

```
8 rows in set (0.00 sec)
```

그림. customer 테이블

. 뷰 생성 명령어 및 테이블 내용 조회

```
CREATE VIEW GoodCustomer ( account, name, credit )
AS SELECT account, name, credit
FROM customer
WHERE grade='VIP' OR grade='Gold'
WITH CHECK OPTION ;
```

```
select * from GoodCustomer;
```

그림. GoodCustomer 뷰 생성과 조회

```
mysql> select * from GoodCustomer;
```

account	name	credit
bank	홍길동	30000
apple	이남이	5000
eagle	박세재	2450
king	오나라	15000

```
4 rows in set (0.02 sec)
```

그림. GoodCustomer 뷰의 데이터

. 뷰의 구조 보기

```
mysql> desc goodcustomer;
```

Field	Type	Null	Key	Default	Extra
account	varchar(10)	NO		NULL	
name	varchar(10)	NO		NULL	
credit	int(11)	YES		0	

```
3 rows in set (0.00 sec)
```

그림. GoodCustomer 뷰의 구조

. 뷰의 속성리스트가 SELECT 문의 속성리스트와 이름이 동일하므로 지정하지 않고 사용

```
CREATE VIEW GoodCustomer1
AS SELECT account, name, credit
FROM customer
WHERE grade='VIP' OR grade='Gold'
WITH CHECK OPTION ;
```

```
select * from GoodCustomer1;
```

그림. GoodCustomer1 뷰 생성과 조회

```
mysql> desc goodcustomer1;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| account | varchar(10) | NO   |     | NULL    |       |
| name    | varchar(10) | NO   |     | NULL    |       |
| credit  | int(11)     | YES  |     | 0       |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.03 sec)
```

그림. GoodCustomer1 뷰 구조

- 예제 : GoodCustomer 뷰에서 Credit이 10000 이상인 고객의 account와 credit로 구성된 GoodCustomerBigCredit 뷰를 생성하시오.

```
CREATE VIEW GoodCustomerBigCredit
AS SELECT account, credit
FROM GoodCustomer
WHERE credit > 10000
WITH CHECK OPTION ;
```

```
select * from GoodCustomerBigCredit;
```

그림. GoodCustomerBigCredit 뷰 생성과 조회



```
mysql> desc goodcustomerbigcredit;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| account | varchar(10) | NO   |     | NULL    |      |
| credit  | int(11)    | YES  |     | 0       |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from goodcustomerbigcredit;
+-----+-----+
| account | credit |
+-----+-----+
| king    | 15000  |
+-----+-----+
1 row in set (0.01 sec)
```

그림. GoodCustomerBigCredit 뷰의 구조와 데이터

- 예제 : 주문 테이블에서 고객별 주문 건수로 구성된 뷰를 OrderByCustomer 라는 이름으로 생성하고, 내용을 조회하시오.

. 주문 테이블의 데이터 조회

```
mysql> select * from orders;
+-----+-----+-----+-----+-----+-----+
| num | customer | item | qty | date       | saddr |
+-----+-----+-----+-----+-----+-----+
| 1 | apple   | E06 | 15 | 2015-06-01 | 경기 용인시 |
| 2 | apple   | E05 | 2  | 2015-06-01 | 경기 용인시 |
| 3 | eagle   | E01 | 3  | 2015-06-03 | 부산 남구 |
| 4 | bank    | E04 | 5  | 2015-06-03 | 서울 중로구 |
| 5 | bird    | E02 | 10 | 2015-06-04 | 충남 천안시 |
| 6 | bird    | E03 | 1  | 2015-06-04 | 충남 천안시 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

그림. Order 테이블의 데이터

. OrderByCustomer 뷰를 생성하고 조회하기 위한 명령어

```
CREATE VIEW OrdersByCustomer ( customername, no_of_orders )
AS SELECT customer, COUNT(*)
FROM orders
GROUP BY customer;
```

```
select * from OrdersByCustomer;
```

그림. 고객별 주문의 수를 위한 OrdersByCustomer 뷰

. OrderByCustomer 뷰의 데이터 조회

```
mysql> select * from ordersbycustomer;
+-----+-----+
| customername | no_of_orders |
+-----+-----+
| apple        | 2           |
| bank         | 1           |
| bird         | 2           |
| eagle        | 1           |
+-----+-----+
4 rows in set (0.00 sec)
```

그림. OrdersByCustomer 뷰의 데이터

. OrderByCustomer 뷰의 구조

```
mysql> desc ordersbycustomer;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| customername | varchar(10) | NO   |     | NULL    |       |
| no_of_orders  | bigint(21) | NO   |     | 0        |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

그림. OrdersByCustomer 뷰의 구조

### 3. 뷰의 활용

- 뷰의 삽입, 수정, 삭제 연산
  - . 기본 테이블에서 수행됨
  - . 제한적으로 수행됨
- 변경 불가능한 뷰
  - . 기본 테이블에 있는 내용이 아니고, 집계 함수로 계산된 내용을 포함하는 뷰
  - . DISTINCT 키워드를 포함하여 정의한 뷰
  - . GROUP BY 절을 포함하여 정의한 뷰
  - . 여러 개의 테이블을 조인하여 정의한 뷰 - 변경이 불가능한 경우가 많음

```
mysql> select * from goodcustomer;
+-----+-----+-----+
| account | name      | credit |
+-----+-----+-----+
| apple   | 이남이    | 5000   |
| eagle   | 박세재    | 2450   |
| king    | 오나라    | 15000  |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> insert into goodcustomer values ( 'bank', '홍길동', 30000 );
ERROR 1369 (HY000): CHECK OPTION failed 'itbank.goodcustomer'
```

그림. GoodCustomer뷰에 데이터 삽입 거절 (CHECK OPTION 위배)



```
mysql> select * from ordersbycustomer;
+-----+-----+
| customername | no_of_orders |
+-----+-----+
| apple        | 2            |
| bank         | 1            |
| bird         | 2            |
| eagle        | 1            |
+-----+-----+
4 rows in set (0.05 sec)

mysql> insert into ordersbycustomer values ( 'apple', 3 );
ERROR 1471 (HY000): The target table ordersbycustomer of the INSERT is not insertable-into
mysql>
```

그림. OrdersByCustomer 테이블 삽입 거절 (집계함수사용)

- 뷰 삽입 (WITH CHECK OPTION)
  - . 제약 조건 : SELECT 문에서 제시한 정의 조건
  - . 정의 조건을 위반하면 수행 안 됨
  - . 예제 : 고객 테이블에서 등급(Grade)이 'VIP' 이거나 'Gold'인 고객의 계정과, 이름, 등급을 가진 GoodCustomer4 뷰를 만드시오.

```
CREATE VIEW GoodCustomer4
AS SELECT account, name, grade
FROM customer
WHERE grade='VIP' OR grade='Gold'
WITH CHECK OPTION;
```

그림. GoodCustomer4 뷰 생성

```
select * from GoodCustomer4;
desc GoodCustomer4;
```

The screenshot shows two MySQL queries and their results. The first query is 'select \* from GoodCustomer4;' which returns a table with 6 rows. The second query is 'desc GoodCustomer4;' which shows the table structure with 3 columns: account, name, and grade.

account	name	grade
apple	이남이	VIP
eagle	박세패	Gold
king	오나라	Gold
bank1	홍길동	Gold
bank	홍길동	Gold
double	리스팅	VIP

Field	Type	Null	Key	Default	Extra
account	varchar(10)	NO		NULL	
name	varchar(10)	NO		NULL	
grade	varchar(6)	NO		New	

그림. GoodCustomer4 뷰 데이터 및 구조 조회

```
select * from customer;
```

The screenshot shows the result of the query 'select \* from customer;'. It returns a table with 10 rows and 6 columns: num, account, name, grade, credit, and address.

num	account	name	grade	credit	address
2	apple	이남이	VIP	5000	경기 용인시
3	pencil	김민준	Silver	350	경기 수원시
4	eagle	박세패	Gold	2450	부산 남구
5	bird	김민준	Gold	0	충청 안성시
6	king	오나라	Gold	15000	서울 영등포구
103	bank1	홍길동	Gold	10000	서울 영등포구
104	bank2	홍길동	Silver	2000	서울 영등포구
200	oil	홍길동	New	0	충청 안성시
201	bank	홍길동	Gold	30000	서울 영등포구
202	double	리스팅	VIP	0	NULL

그림. Customer 테이블 데이터 조회

```
insert into goodcustomer4 values ( 'memo', '메모장', 'Gold' );
select * from goodcustomer4;
```

```
mysql> insert into goodcustomer4 values ( 'memo', '메모장', 'Gold' );
Query OK, 1 row affected (0.05 sec)

mysql> select * from goodcustomer4;
+-----+-----+-----+
| account | name      | grade |
+-----+-----+-----+
| apple   | 이남이    | VIP   |
| eagle   | 박세재    | Gold  |
| king    | 오나라    | Gold  |
| bank1   | 홍길동서 | Gold  |
| bank    | 홍길동    | Gold  |
| double  | 리스트    | VIP   |
| memo    | 메모장    | Gold  |
+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

그림. 뷰에 데이터 삽입후 뷰 데이터 조회

```
select * from customer;
```

```
mysql> select * from customer;
+----+-----+-----+-----+-----+-----+
| num | account | name      | grade | credit | address |
+----+-----+-----+-----+-----+-----+
| 2   | apple   | 이남이    | VIP   | 5000   | 경기 용인시 |
| 3   | pencil  | 김돌      | Silver | 350    | 경기 수원시 |
| 4   | eagle   | 박세재    | Gold   | 2450   | 부산 남구 |
| 5   | bird    | 구선두    |        | 0       | 전남 안성시 |
| 6   | king    | 오나라    | Gold   | 15000  | 서울 성북구 |
| 103 | bank1   | 홍길동서 | Gold   | 10000  | 서울 중로구 |
| 104 | bank2   | 홍길동    | Silver | 2000   | 서울 중로구 |
| 200 | oil     | 오오오    | New    | 0       | 충남 서산 |
| 201 | bank    | 홍길동    | Gold   | 30000  | 서울 중로구 |
| 202 | double  | 리스트    | VIP    | 0       | NULL |
| 204 | memo    | 메모장    | Gold   | 0       | NULL |
+----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

그림. 뷰에 데이터 삽입후 기본테이블 데이터 조회

- 기본 테이블의 데이터가 삭제되거나 변경되면 뷰의 내용도 같이 변경됨
- . 기본 테이블의 데이터를 삭제하기전 데이터

```
mysql> select * from customer;
```

num	account	name	grade	credit	address
1	bank	홍길동	Gold	30000	서울 종로구
2	apple	이남이	VIP	5000	경기 용인시
3	pencil	김돌	Silver	350	경기 수원시
4	eagle	박세재	Gold	2450	부산 남구
5	bird	구선두		0	충남 천안시
6	king	오나라	Gold	15000	서울 성북구
100	oil	오오구	New	0	충남 서산
101	good	도돌이	New	0	제주 서귀포시

```
8 rows in set (0.00 sec)
```

```
mysql> select * from goodcustomer;
```

account	name	credit
bank	홍길동	30000
apple	이남이	5000
eagle	박세재	2450
king	오나라	15000

```
4 rows in set (0.00 sec)
```

그림. Customer 테이블과 GoodCustomer뷰의 데이터(삭제전)

- . customer 테이블의 bank 계정 레코드를 삭제 후

```
mysql> select * from customer;
```

num	account	name	grade	credit	address
2	apple	이남이	VIP	5000	경기 용인시
3	pencil	김돌	Silver	350	경기 수원시
4	eagle	박세재	Gold	2450	부산 남구
5	bird	구선두		0	충남 천안시
6	king	오나라	Gold	15000	서울 성북구
100	oil	오오구	New	0	충남 서산
101	good	도돌이	New	0	제주 서귀포시

```
7 rows in set (0.00 sec)
```

```
mysql> select * from goodcustomer;
```

account	name	credit
apple	이남이	5000
eagle	박세재	2450
king	오나라	15000

```
3 rows in set (0.00 sec)
```

그림. Customer 테이블과 GoodCustomer뷰의 데이터 (삭제후)



#### 4. 뷰의 장점

- 질의문을 쉽게 작성할 수 있다.
  - . 특정 조건을 만족하는 레코드들로 뷰를 미리 만들어 놓을 경우  
==> 사용자가 WHERE 절에 특정 조건을 넣지 않아도 된다.
  - . GROUP BY, 집계함수, 조인 등을 이용해 뷰를 미리 만들 경우  
==> 복잡한 질의문을 만들지 않고 SELECT절과 FROM 절만으로 데이터 검색이 가능
- 데이터의 보안 유지에 도움이 된다.
  - . 여러 사용자의 요구에 맞는 다양한 뷰를 미리 정의
  - . 사용자에게 제공된 뷰를 통해서만 데이터에 접근하도록 권한 설정  
==> 뷰에 포함되지 않은 데이터를 사용자로부터 보호할 수 있음.
- 데이터를 편리하게 관리할 수 있다.
  - . 제공된 뷰에 포함되지 않은 기본 테이블의 다른 속성은 사용자가 신경 쓸 필요가 없음
  - . 제공된 뷰와 관련이 없는 다른 테이블의 변화에도 영향을 받지 않음.

#### 뷰 삭제

- 뷰를 삭제해도 기본 테이블은 영향을 받지 않음
- 명령어 : DROP VIEW 뷰이름 CASCADE | RESTRICT;
- 삭제할 뷰를 이용해 만들어진 다른 뷰가 존재할 경우 처리 방법
  - . RESTRICT : 다른 뷰가 삭제되지 않음
  - . CASCADE : 다른 뷰도 같이 삭제

#### 【학습정리】

1. 데이터베이스의 명령어들을 파일에 저장하고, 일괄처리 실행을 하면 데이터베이스 변경을 쉽게 할 수 있다.
2. 백업 후 복원하는 과정도 일괄 처리 방법을 사용한 것이다.
3. 뷰는 데이터베이스에 있는 일반 테이블을 기반으로 만들어진 가상 테이블로 데이터를 실제로 저장하지 않고 논리적으로만 가지고 있는 테이블이다. 뷰를 사용하여 데이터의 보안을 유지하는데 도움을 받을 수 있다.