



Chapter 09

데이터베이스 관리와 보안

MySQL로 배우는 데이터베이스 개론과 실습

목차

01

데이터베이스 관리

02

데이터베이스 보안

03

백업과 복원

학습목표

- ❖ 데이터베이스 관리의 중요성을 이해하고 데이터베이스 관리 업무에는 무엇이 있는지 알아본다.
- ❖ DBMS 또는 특정 데이터에 대한 사용자 권한 제한 방법에 대해 알아본다.
- ❖ 데이터베이스 장애 시 백업 데이터를 이용하여 복원하는 방법을 알아본다.

01 데이터베이스 관리

1. 데이터베이스 관리의 중요성
2. 데이터베이스 관리 업무
3. 데이터베이스 관리 기본 명령



1. 데이터베이스 관리의 중요성

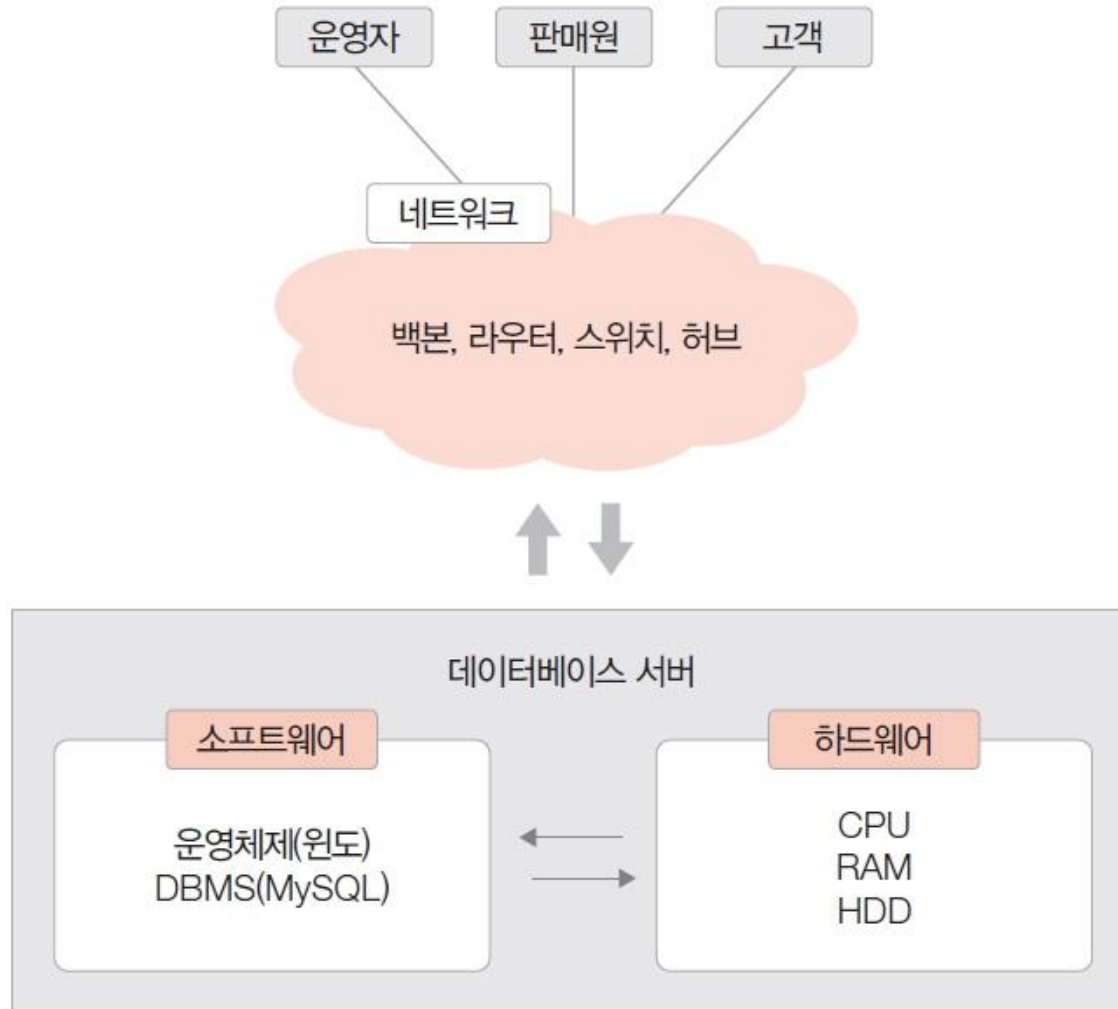


그림 9-1 데이터베이스 운영 환경

2. 데이터베이스 관리 업무

- DBA가 데이터베이스 관리를 위해 수행하는 일반적인 업무

- 서비스 관리
- 장애 대처
- 사용자 관리 및 권한 관리
- 사용자 데이터베이스 관리
- 인덱스 관리
- 점검 및 모니터링
- 백업과 복원
- 시스템 데이터베이스 관리
- 데이터베이스 저장 공간 관리

그림 9-2 데이터베이스 관리 업무

※ 자세한 내용은 책을 참고합니다.

3. 데이터베이스 관리 기본 명령

❖ SHOW DATABASES 명령

- DB에 어떤 데이터베이스가 있는지 보여줌
- 기본적으로 mysql, sys 등 관리용 데이터베이스를 확인할 수 있음

```
SHOW DATABASES;
```

Database
information_schema
madangdb
mysql
performance_schema
sakila
sys
world

3. 데이터베이스 관리 기본 명령

❖ USE 명령

- 사용할 데이터베이스를 지정하는 명령어

```
USE mysql; -- 데이터베이스 mysql을 사용
```

❖ SHOW TABLES 명령

- 사용할 수 있는 데이터베이스 목록을 보여줌

```
SHOW TABLES;
```

tables_in_mysql
servers
slave_master_info
slave_relay_log_info
slave_worker_info
slow_log
tables_priv
time_zone
time_zone_leap_second
time_zone_name
time_zone_transition
time_zone_transition_type
user

3. 데이터베이스 관리 기본 명령

❖ DESC 명령

- 테이블의 구조를 살펴보는 명령어

DESC mysql.User

Field	Type	Null	Key	Default	Extra
Host	char(60)	NO	PRI		
User	char(32)	NO	PRI		
Select_priv	enum('N','Y')	NO		N	
Insert_priv	enum('N','Y')	NO		N	
Update_priv	enum('N','Y')	NO		N	
Delete_priv	enum('N','Y')	NO		N	
Create_priv	enum('N','Y')	NO		N	
Drop_priv	enum('N','Y')	NO		N	
Reload_priv	enum('N','Y')	NO		N	
Shutdown_priv	enum('N','Y')	NO		N	
Process_priv	enum('N','Y')	NO		N	
File_priv	enum('N','Y')	NO		N	
Grant_priv	enum('N','Y')	NO		N	

3. 데이터베이스 관리 기본 명령

❖ SELECT * FROM mysql.User 명령

- 사용자 계정 테이블인 User의 내용을 확인하고 사용자 계정을 살펴봄
- 권한이 있는 관리자(root)가 mysql 데이터베이스의 User 테이블을 조회할 수 있음

```
SELECT * FROM mysql.User;
```

Host	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv
localhost	hr	N	N	N	N	N	N
localhost	madang	N	N	N	N	N	N
localhost	mysql.infoschema	Y	N	N	N	N	N
localhost	mysql.session	N	N	N	N	N	N
localhost	mysql.sys	N	N	N	N	N	N
localhost	root	Y	Y	Y	Y	Y	Y
localhost	scott	N	N	N	N	N	N

02 데이터베이스 보안

1. 로그인 사용자 관리
2. 권한 관리
3. 권한 관리 실습



데이터베이스 접근 권한

- 일반적으로 DBMS는 [그림 9-3]과 같이 ❶ 로그인 단계에서 DBMS 접근을 제한하는 로그인 사용자 관리와 ❷ 로그인한 사용자별로 데이터베이스 및 테이블에 대한 사용을 제한하는 권한 관리의 기능을 제공함

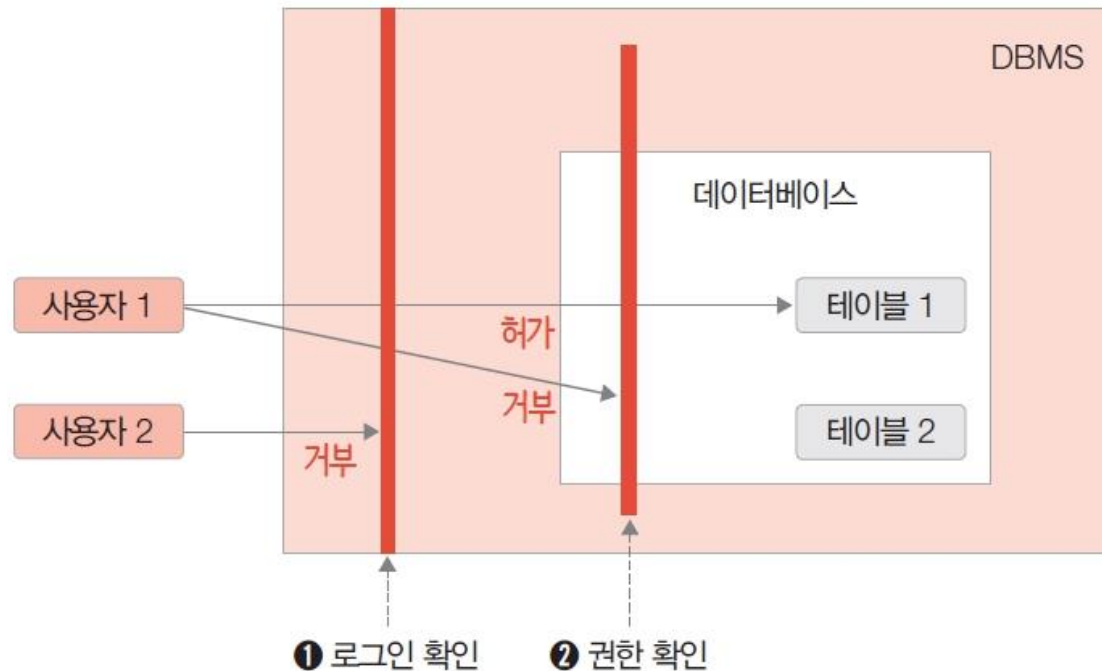


그림 9-3 데이터베이스 접근 권한

1. 로그인 사용자 관리

❖ 사용자 계정 생성 - CREATE USER

- 사용자 계정을 생성하는 명령으로, 계정 생성 권한을 가진 관리자r(oot)가 사용 가능

```
CREATE USER [사용자이름] IDENTIFIED BY [비밀번호];
```

- localhost에서 접속하는 사용자 생성
 - MySQL은 사용자 생성 시 사용자 이름과 접속 가능한 호스트 이름을 같이 표시
 - 사용자 이름은 'user@hostname'과 같은 형태가 됨

```
CREATE USER madang@localhost IDENTIFIED BY 'madang';
```

- 외부에서 접속하는 사용자 생성
 - 외부에서 호스트에 상관없이 접속 가능한 madang 사용자는 다음과 같이 생성함

```
CREATE USER madang@ '%' IDENTIFIED BY 'madang';
```

1. 로그인 사용자 관리

- 특정 사이트에서 접속하는 사용자 생성
 - 특정 사이트인 happy.md.kr에서 접속 가능한 madang 사용자 계정은 다음과 같이 생성함

```
CREATE USER madang@'%.happy.md.kr' IDENTIFIED BY 'madang';
```

- User 테이블을 조회하여 madang 사용자가 생성되었는지 확인함

```
SELECT * FROM mysql.User WHERE User LIKE 'madang';
```

Host	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv
localhost	madang	N	N	N	N	N	N

❖ 사용자 계정 삭제-DROP USER

- 사용자 계정을 삭제하는 명령의 형식
- mdguest@localhost 사용자를 삭제하는 예

```
DROP USER [사용자이름];
```

```
DROP USER mdguest@localhost;
```

2. 권한 관리

- MySQL에서 madang, mdguest 각각의 사용자가 접속하면 [그림 9-4]와 같이 세션이 생성되는데, 각 세션은 사용자와 연결을 유지하면서 명령을 처리함

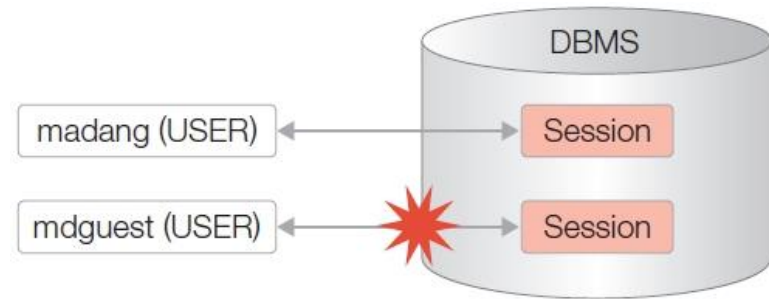


그림 9-4 사용자 계정과 세션의 관계

- 소유한 개체에 대한 사용 권한을 관리하기 위한 명령을 DCL이라고 함
- 대표적인 DCL 문에는 권한을 허가하기 위한 GRANT 문과 권한을 취소하기 위한 REVOKE 문이 있음

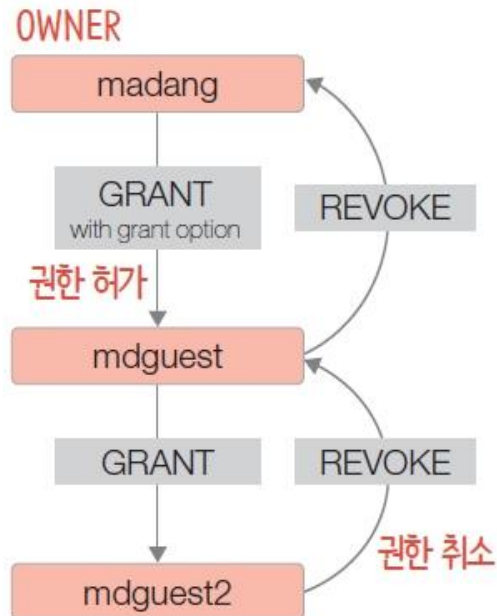


그림 9-5 사용자와 GRANT 문/REVOKE 문의 관계

2. 권한 관리

❖ 권한 허가 – GRANT

- 객체를 생성한 소유자가 대상 객체에 대한 권한을 다른 사용자에게 허가하는 명령

```
GRANT 권한 [(컬럼리스트)]  
[ON 객체] TO {사용자|롤}  
[WITH GRANT OPTION]
```

표 9-1 주요 객체별 지정 가능한 권한(객체 권한, Object privileges)

객체 권한 \ 객체	테이블	뷰	프로시저, 함수
ALTER	가능		
DELETE	가능	가능	
EXECUTE			가능
INDEX	가능		
INSERT	가능	가능	
REFERENCES	가능	가능	
SELECT	가능	가능	
UPDATE	가능	가능	

2. 권한 관리

❖ 권한 취소 – REVOKE

- GRANT 문으로 허가한 권한을 취소, 회수하는 명령
- 권한을 부여한 사용자는 부여한 권한의 범위 내에서 권한을 취소할 수 있음

```
REVOKE 권한 [컬럼리스트]  
[ON 객체] FROM {사용자|롤}
```

- GRANT 문이 권한 부여를 위해 'TO 사용자'로 표기했다면, REVOKE 문은 권한 취소를 위해 'FROM 사용자'로 표기함

3. 권한 관리 실습

- 실습은 Workbench에서 서로 다른 사용자 Query를 실행시켜 확인하면서 진행함

질의 9-1

(root 계정) 새로운 사용자 mdguest를 생성하시오. 비밀번호는 mdguest로 한다. 실습에서는 호스트를 localhost에서만 접속한다고 가정하고 localhost에 대한 사용자를 생성한다.

```
CREATE USER mdguest@'localhost' IDENTIFIED BY 'mdguest';  
# mdguest 사용자 생성은 User 테이블을 통해 확인한다.  
SELECT * FROM mysql.User WHERE User LIKE 'mdguest';
```

Host	User	Select_ priv	Insert_ priv	Update_ priv	Delete_ priv	Create_ priv	Drop_ priv
localhost	mdguest	N	N	N	N	N	N

3. 권한 관리 실습

질의 9-2

(root 계정) 새로운 사용자 mdguest2를 생성하시오. 비밀번호는 mdguest2로 한다.

```
CREATE USER mdguest2 IDENTIFIED BY 'mdguest2';  
# mdguest2 사용자 생성은 User 테이블을 통해 확인한다.  
SELECT * FROM mysql.User WHERE User LIKE 'mdguest2';
```

Host	User	Select_ priv	Insert_ priv	Update_ priv	Delete_ priv	Create_ priv	Drop_ priv
%	mdguest2	N	N	N	N	N	N

3. 권한 관리 실습

- [질의 9-3]부터는 [그림 9-6]의 내용을 실습

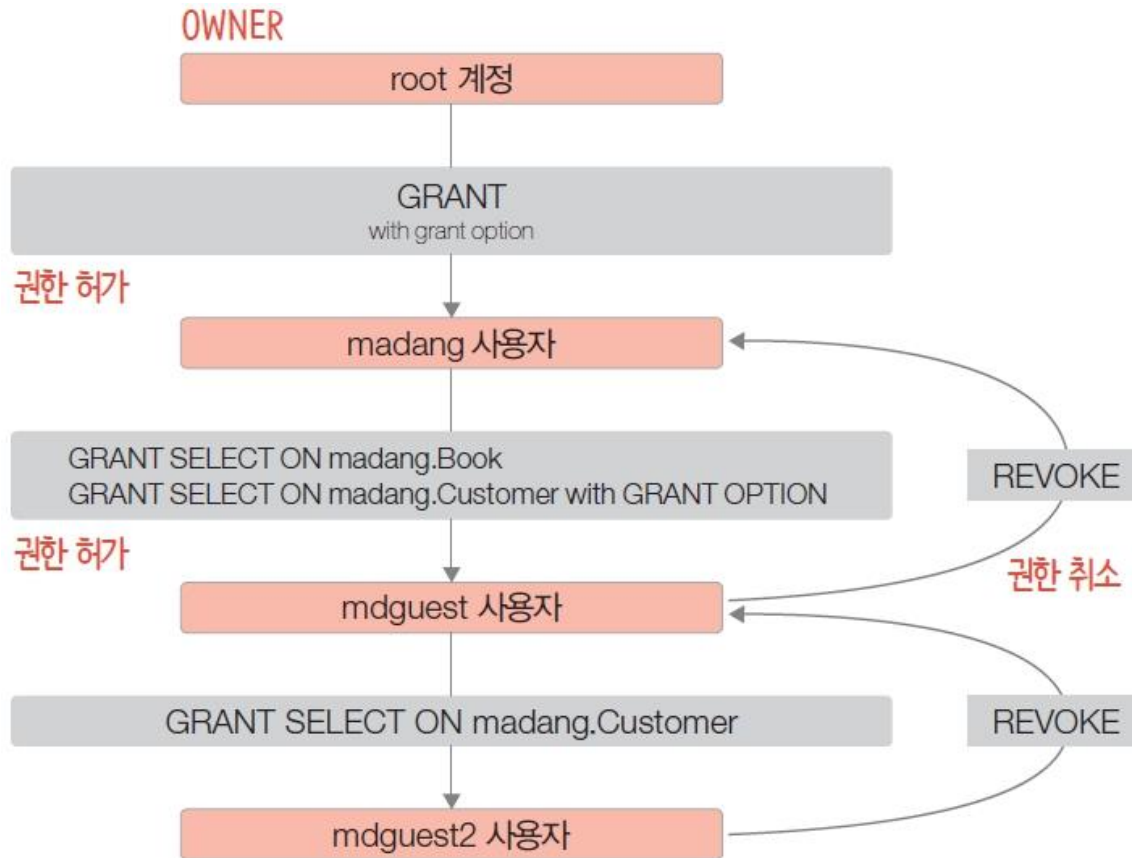


그림 9-6 GRANT 문과 REVOKE 문 실습

3. 권한 관리 실습

- 사용자는 madang, mdguest, mdguest2이며 데이터베이스는 madangdb
- madangdb에 있는 테이블에 대한 권한을 부여하고 회수하는 상황을 실습

질의 9-3

(root 계정) madang 사용자에게 데이터베이스 생성(CREATE) 및 테이블 생성 등에 관한 모든 권한을 GRANT OPTION과 함께 부여하시오. FLUSH 문으로 권한 변경 사항을 반영하고 SHOW 문으로 권한 부여를 확인하시오(이 내용은 3장에서 madang 사용자 생성 시 이미 실행해 본 바 있다).

```
GRANT ALL PRIVILEGES ON *.* TO madang@localhost WITH GRANT OPTION;  
FLUSH PRIVILEGES;  
SHOW GRANTS FOR madang@localhost;
```

Grants for madang@localhost

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, ...

GRANT APPLICATION_PASSWORD_ADMIN,AUDIT_ABORT_EXEMPT,AUDIT_ADMIN, ...

GRANT ALL PRIVILEGES ON `madangdb`. *TO `madang`@'localhost' WITH GRANT OPTION

3. 권한 관리 실습

- 이제 다시 mdguest/mdguest로 접속해 Query 창을 만들기
- 정상적으로 접속되면 지금까지 실습에 사용한 madangdb 데이터베이스의 Book 테이블을 조회해보기

```
(mdguest 계정)  
Use madangdb;  
SELECT * FROM Book;
```

Error Code: 1044. Access denied for user 'mdguest'@'localhost' to database 'madangdb'

- 오류가 발생. 실제 mdguest 사용자가 madangdb 데이터베이스에 대한 접근 권한을 가지고 있지 않기 때문임
- 다른 사용자가 권한을 갖고 있는 데이터베이스의 테이블에 접근하기 위해서는 테이블명 앞에 데이터베이스 이름을 붙여서 표시해줌

```
SELECT * FROM madangdb.Book;
```

Error Code: 1142. SELECT command denied to user 'mdguest'@'localhost' for table 'book'

3. 권한 관리 실습

질의 9-4

(madang 계정) mdguest에게 Book 테이블의 SELECT 권한을 부여하시오.

```
GRANT SELECT ON madangdb.Book TO mdguest@localhost;
```

0 row(s) affected

- 이제 mdguest 사용자는 다음 질의를 수행할 수 있음. mdguest 계정으로 접속하여 실행해보기

```
USE madangdb;  
SELECT * FROM Book;
```

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	배구 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

3. 권한 관리 실습

질의 9-5

(madang 계정) mdguest에게 Customer 테이블의 SELECT, UPDATE 권한을 WITH GRANT OPTION과 함께 부여하시오.

```
GRANT SELECT, UPDATE ON madangdb.Customer TO mdguest@localhost  
WITH GRANT OPTION;
```

0 row(s) affected

질의 9-6

(mdguest 계정) madangdb.Book 테이블과 madangdb.Customer 테이블의 SELECT 권한을 mdguest2에 부여하시오.

```
GRANT SELECT ON madangdb.Book TO mdguest2;
```

Error Code: 1142. GRANT command denied to user 'mdguest'@'localhost' for table 'book'

3. 권한 관리 실습

- mdguest가 Book 테이블의 SELECT 권한을 mdguest2에게 부여할 수 있는 권한을 받지 못했기 때문에 오류가 발생함
- 그렇지만 Customer 테이블의 SELECT 권한은 WITH GRANT OPTION으로 받았기 때문에 mdguest2에게 부여해 줄 수 있음

```
GRANT SELECT ON madangdb.Customer TO mdguest2;
```

```
GRANT SELECT ON madangdb.Custo ... 0 row(s) affected
```

- mdguest2 계정으로 접속하여 다음 문장을 실행함

```
(mdguest2 계정)  
USE madangdb;  
SELECT * FROM Customer;
```

custid	name	address	phone
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	김연경	대한민국 경기도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001
5	박세리	대한민국 대전	NULL

3. 권한 관리 실습

질의 9-7

(madang 계정) mdguest에게 부여된 Book 테이블의 SELECT 권한을 취소하시오.

```
REVOKE SELECT ON madangdb.Book FROM mdguest@localhost;
```

0 row(s) affected

질의 9-8

(madang 계정) mdguest에게 부여된 Customer 테이블의 SELECT 권한을 취소하시오.

```
REVOKE SELECT ON madangdb.Customer FROM mdguest@localhost;
```

0 row(s) affected

03 백업과 복원

1. 백업의 개념
2. 백업의 종류
3. MySQL의 백업과 복원



1. 백업의 개념

❖ 백업

- 데이터베이스에서도 예상하지 못한 문제가 발생할 수 있으므로 데이터베이스를 복제하여 보관하는 작업

❖ 복원

- 장애가 발생하여 운영 중인 데이터에 손상이 발생했을 때, 이전에 복사해 둔 백업 파일을 사용하여 데이터를 원래대로 되돌려 놓는 작업
- 데이터가 손실되거나 손상된 경우, 백업 파일을 활용하여 원래 상태로 복구하는 과정

❖ 데이터베이스 시스템 운영 시 일어날 수 있는 장애

- 미디어 오류
- 사용자 오류
- 하드웨어 장애

2. 백업의 종류

- 백업은 일반적으로 전체 백업, 차등 백업, 증분 백업으로 나뉨

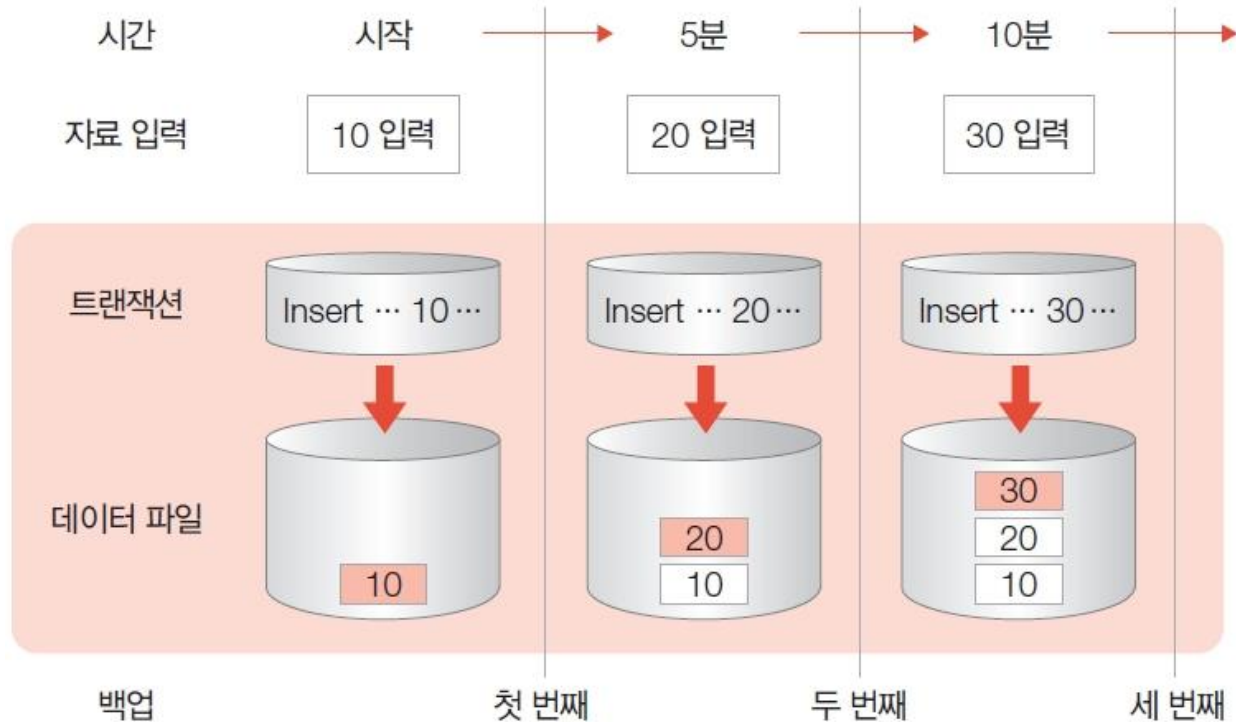


그림 9-7 데이터 파일의 입력 순서(시간순)

2. 백업의 종류

❖ 전체 백업

- 데이터베이스 개체, 시스템 테이블, 데이터 등 데이터베이스 전체를 백업하는 것
- 백업 시점의 데이터베이스 복사본을 만들어 두는 것으로 이해하면 됨

❖ 차등 백업

- 전체 백업을 수행한 이후 변경된 데이터만 저장하는 것
- 전체 백업으로 데이터베이스의 복사본을 만든 후 그 복사본과 차이가 있는 변경 부분만 백업하는 방법

❖ 증분 백업

- 데이터베이스에서 수행한 작업을 기록하고 있는 트랜잭션 로그 파일을 저장하는 방법

3. MySQL의 백업과 복원

❖ MySQL의 백업 방법

■ 물리적 백업

- 데이터베이스를 구동하기 위해 필요한 모든 파일을 물리적으로 '복사'하는 방법
- 데이터베이스를 운영 중일 때 진행하는지 혹은 중지했을 때 진행하는지에 따라 콜드 백업과 핫 백업으로 나뉨

■ 논리적 백업

- 실제 데이터베이스를 구성하는 물리적 파일을 직접 복사하는 방법이 아니라 데이터베이스에 있는 콘텐츠(내용)를 별도의 파일로 옮겨 백업하는 방법
- 데이터를 일종의 스크립트 형태로 백업함

❖ MySQL의 논리적 백업 실습

- 우선 MySQL에서 백업 및 복구에 사용할 디렉토리를 지정한 후 다음과 같이 진행

❶ 백업 파일이 저장될 폴더를 준비. 여기서는 C:\wmadang\wbackup 폴더를 생성

❷ Workbench에서 root 계정으로 쿼리 창을 만듦. 여기서는 madangdb 데이터베이스 전체를 백업하고 복원해 보도록 함. 먼저 데이터베이스가 있는지 확인함

```
USE madangdb;  
SELECT * FROM Customer;
```

3. MySQL의 백업과 복원

- Workbench에서 [Navigator] - Administration 탭의 [MANAGEMENT] - [Data Export]를 클릭
- 그러면 [그림 9-8]과 같은 Data Export 화면이 나타남
- 다음 사항을 선택

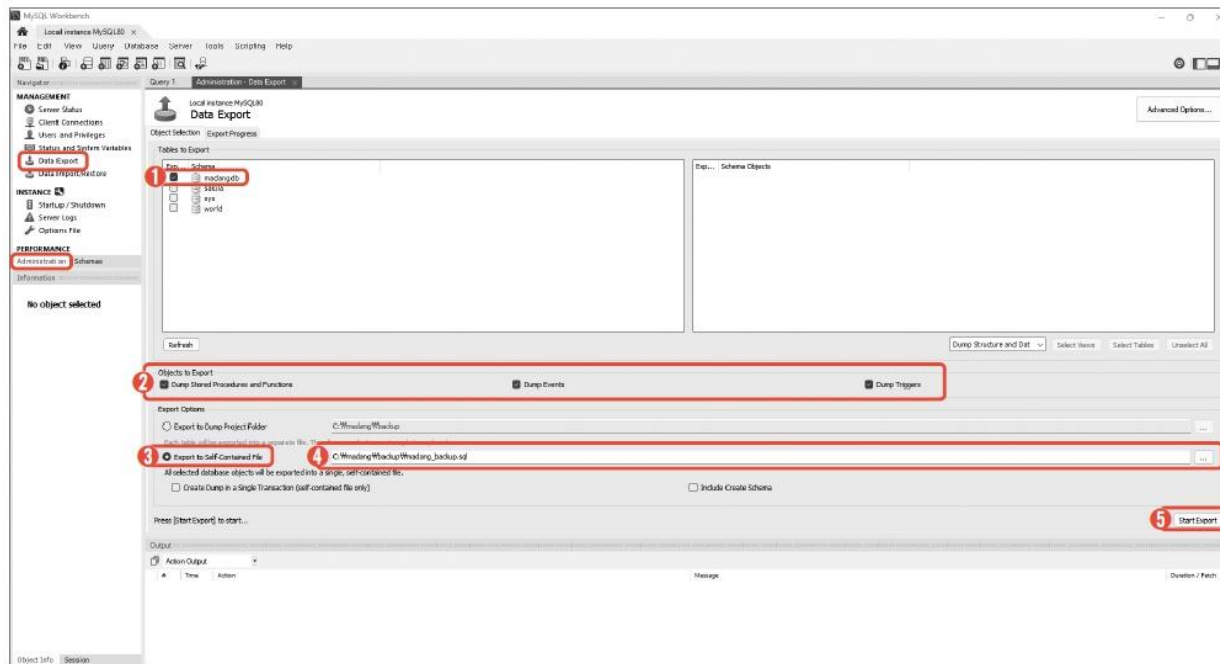


그림 9-8 데이터베이스 백업-Export 화면

③ 백업이 끝나면 백업 화면을 닫고 백업된 파일을 살펴봄. 여기서는 C:\wmailing\backup\wmailing_backup.sql 파일을 확인

3. MySQL의 백업과 복원

❖ MySQL의 논리적 복원 실습

- 복원은 백업 시점에 저장된 데이터를 모두 다시 회복시키는 과정
- ❶ 복원할 파일을 확인. 앞선 실습에서 백업한 C:\madang\backup\madang_backup.sql을 사용
- ❷ 복원을 실행하려면 madangdb 데이터베이스에서 테이블을 한 개 삭제(root 계정)

```
USE madangdb;  
SET SQL_SAFE_UPDATES=0;  
DELETE FROM Orders;  
SELECT * FROM Orders;
```

orderid	custid	bookid	saleprice	orderdate
NULL	NULL	NULL	NULL	NULL

- Workbench에서 [Navigator] - Administration 탭의 [MANAGEMENT] - [Data Import/Restore]를 클릭
- [그림 9-9]와 같은 Data Import 화면이 나타남
- 이 화면에서 다음을 선택함

3. MySQL의 백업과 복원

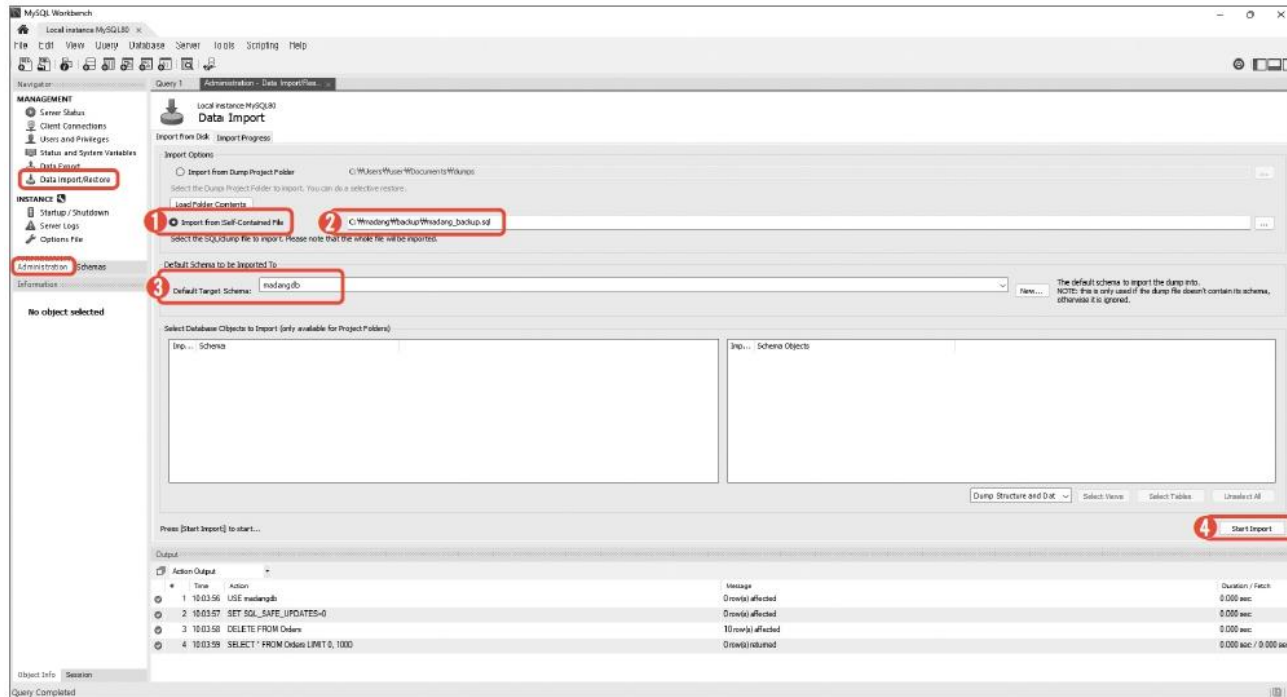


그림 9-9 데이터베이스 복원-Import 화면

③ 복원이 끝나면 복원된 자료를 확인

```
USE madangdb;  
SELECT * FROM Orders;
```

요약

1. DBA
2. DCL
3. 권한 관리
4. WITH GRANT OPTION
5. 역할(ROLE, 롤)
6. 백업, 복원
7. 백업의 종류