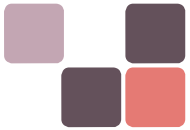
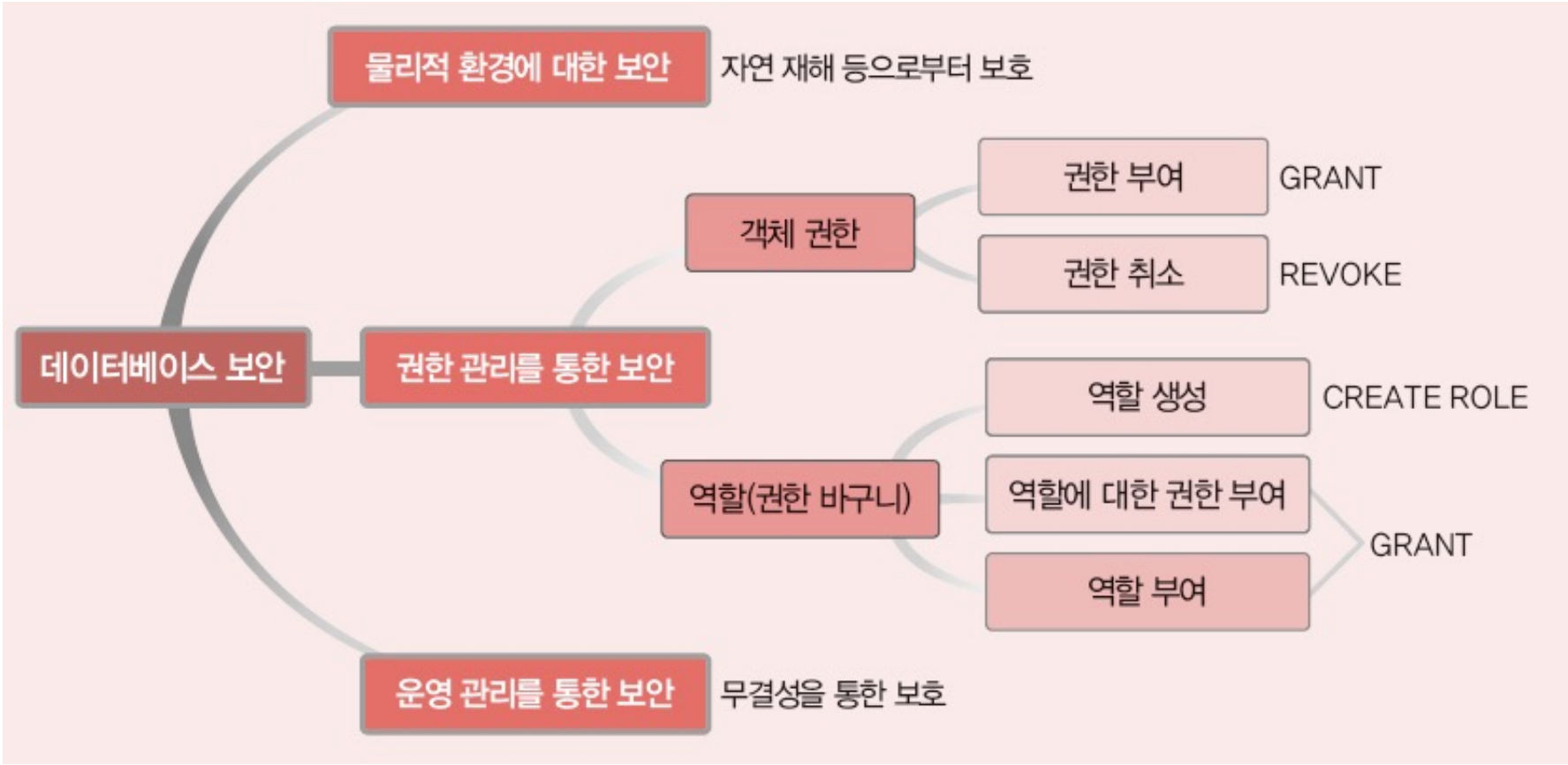


11장. 보안과 권한 관리

- 보안
- 권한 관리

학습목표



- 데이터베이스 보안의 개념과 유형을 이해한다.
- 권한을 부여하고 부여한 권한을 취소하는 방법을 익힌다.
- 역할의 개념과 필요성을 이해한다.
- 역할을 이용해 권한 관리를 수행하는 방법을 익힌다.

01 보안

❖ 데이터베이스 보안의 목표

- 조직에서 허가한 사용자만 데이터베이스에 접근할 수 있도록 통제하여 보안을 유지하는 것

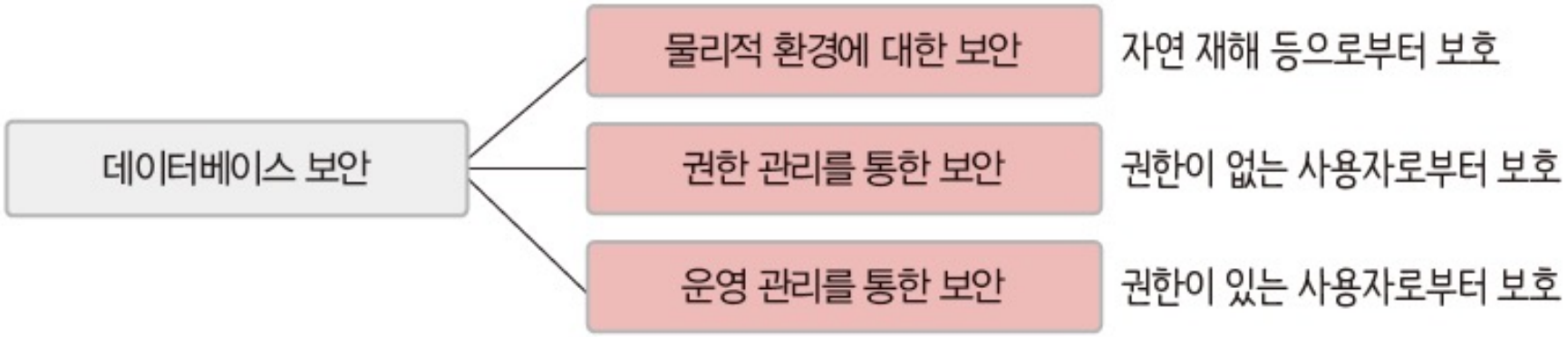
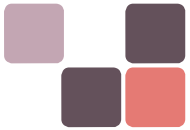


그림 11-1 데이터베이스 보안의 유형

01 보안



❖ 데이터베이스 보안

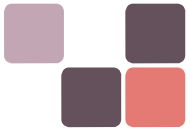
- 물리적 환경에 대한 보안
  - 자연 재해처럼 데이터베이스에 물리적 손실을 발생시키는 위험으로부터 데이터베이스를 보호
- 권한 관리를 통한 보안
  - 접근이 허락된 사용자만 권한 내에서 데이터베이스를 사용하도록 보호
  - 계정이 발급된 사용자만 데이터베이스에 접근할 수 있도록 통제하고, 사용자별로 사용 범위와 수행 가능한 작업 내용을 제한
- 운영 관리를 통한 보안
  - 접근이 허락된 사용자가 권한 내에서 데이터베이스를 사용하는 동안 데이터 무결성을 유지하도록 제약조건을 정의하고 위반하지 않도록 통제

01 보안

❖ 데이터베이스 보안

그림 11-2 보안과 무결성 유지

02 권한 관리



❖ 권한 관리의 개념

- 접근 제어(access control)
  - 계정이 발급된 사용자가 로그인에 성공했을 경우에만 데이터베이스에 접근 허용
  - 사용자 계정 관리는 데이터베이스 관리자가 담당
- 각 사용자는 허용된 권한 내에서만 데이터베이스를 사용
  - 로그인에 성공한 사용자도 데이터베이스 사용 범위와 수행 가능한 작업이 제한됨
    - 보안을 위한 데이터 단위는 데이터베이스 전체부터 특정 테이블의 특정 행과 열 위치에 있는 특정 데이터 값까지 다양함
- 데이터베이스의 모든 객체는 객체를 생성한 사용자만 사용 권한을 가짐
  - 데이터베이스 객체의 소유자는 필요에 따라 SQL 문을 이용해 다른 사용자에게 사용 권한을 부여하거나 취소할 수 있음

02 권한 관리

❖ 권한 관리의 개념

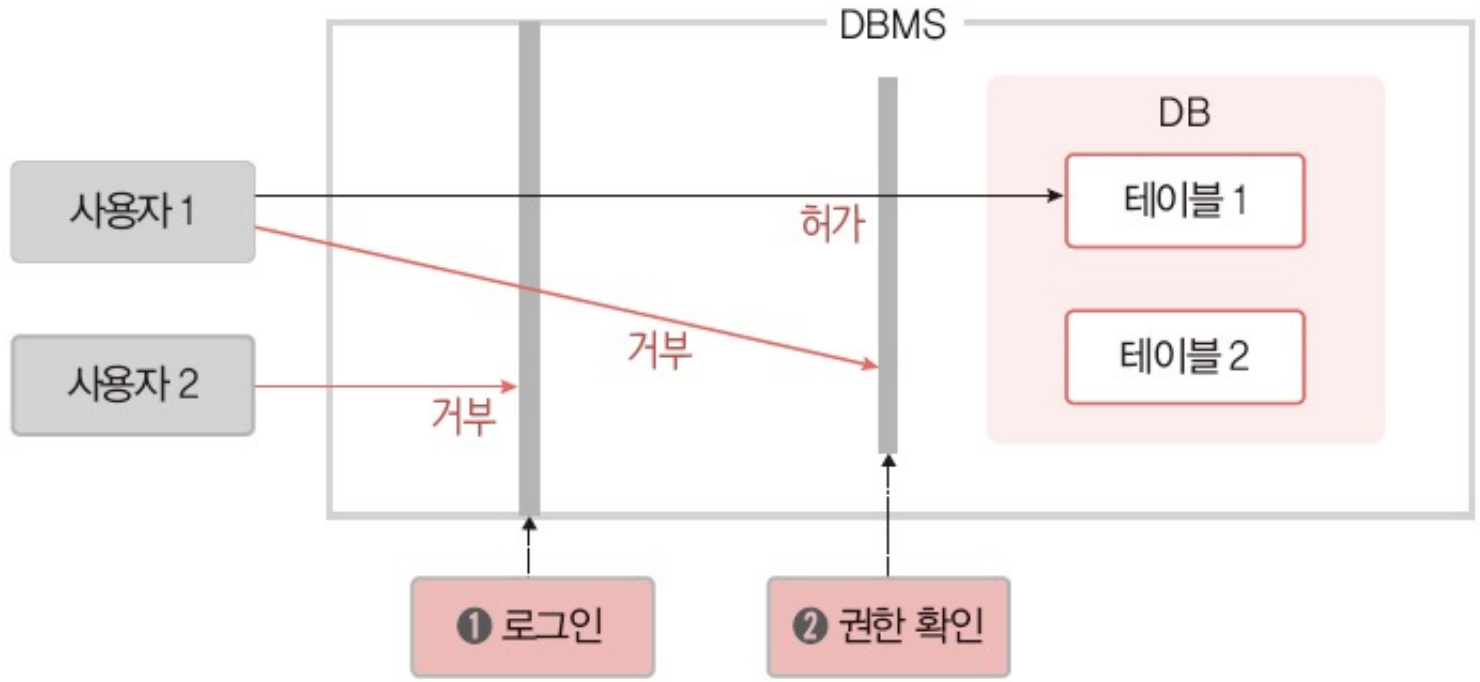


그림 11-3 로그인과 데이터베이스 접근 권한

02 권한 관리

❖ 권한 관리를 통한 보안

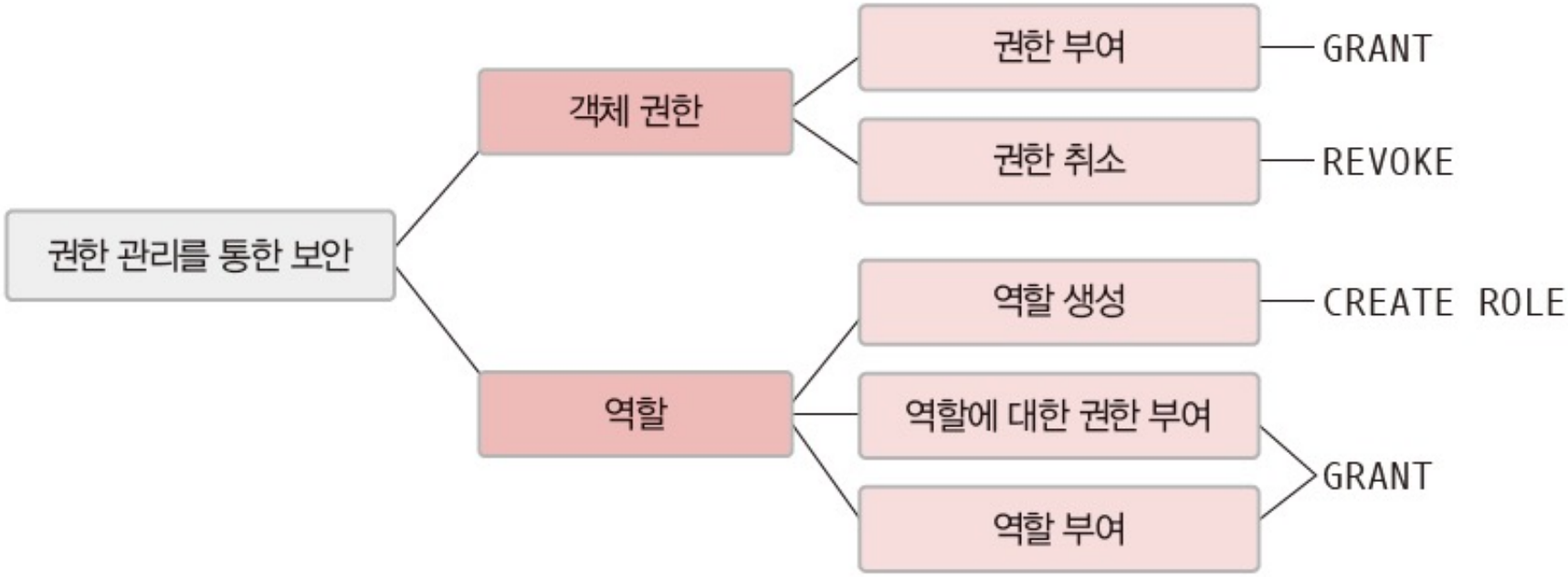
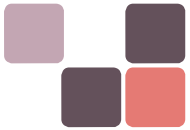


그림 11-4 권한 관리를 통한 보안

02 권한 관리



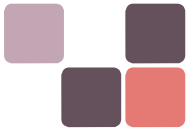
❖ 객체 권한 부여 : GRANT 문

- 객체의 소유자가 다른 사용자에게 객체에 대한 사용 권한을 부여

```
GRANT 권한 ON 객체 TO 사용자 [WITH GRANT OPTION];
```

- 부여 가능한 주요 권한
  - INSERT, DELETE, UPDATE, SELECT, REFERENCES
    - REFERENCES : 외래키 제약조건을 정의할 수 있는 권한
    - UPDATE나 SELECT는 테이블의 일부 속성에 대한 권한 부여도 가능
  - 여러 권한을 한 번에 부여하는 것도 가능
- 기본적으로 GRANT 문으로 부여받은 권한은 다른 사용자에게 부여할 수 없음

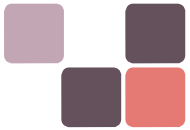
02 권한 관리



❖ 객체 권한 부여 : GRANT 문

- PUBLIC
  - 모든 사용자에게 권한을 똑같이 부여하고 싶다면 특정 사용자를 지정하는 대신 PUBLIC 키워드를 이용하여 작성
- WITH GRANT OPTION
  - 사용자가 자신이 부여받은 권한을 다른 사용자에게도 부여할 수 있도록 함

02 권한 관리



❖ 객체 권한 부여 : GRANT 문

예제 11-1

고객 테이블에 대한 검색 권한을 사용자 Hong에게 부여해보자.

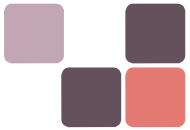
```
▶▶ GRANT SELECT ON 고객 TO Hong;
```

예제 11-2

고객 테이블에 대한 삽입과 삭제 권한을 모든 사용자에게 부여해보자.

```
▶▶ GRANT INSERT, DELETE ON 고객 TO PUBLIC;
```

02 권한 관리



❖ 객체 권한 부여 : GRANT 문

예제 11-3

고객 테이블을 구성하는 속성 중 등급과 적립금 속성에 대한 수정 권한을 사용자 Park에게 부여해보자.

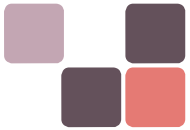
```
▶▶ GRANT UPDATE(등급, 적립금) ON 고객 TO Park;
```

예제 11-4

고객 테이블에 대한 검색 권한을 WITH GRANT OPTION을 포함하여 사용자 Lee에게 부여해보자.

```
▶▶ GRANT SELECT ON 고객 TO Lee WITH GRANT OPTION;
```

02 권한 관리



❖ 시스템 권한 부여 : GRANT 문

- 시스템 권한은 데이터베이스 관리자가 부여함
  - 시스템 권한 : 데이터베이스 관리와 관련된 작업에 대한 권한
    - CREATE TABLE, CREATE VIEW 등 데이터 정의어(DDL)와 관련된 권한들
- 시스템 권한을 부여할 때는 객체를 지정할 필요가 없음

예제 11-5

테이블을 생성할 수 있는 시스템 권한을 사용자 Song에게 부여해보자.

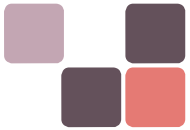
```
▶▶ GRANT CREATE TABLE TO Song;
```

예제 11-6

뷰를 생성할 수 있는 시스템 권한을 사용자 Shin에게 부여해보자.

```
▶▶ GRANT CREATE VIEW TO Shin;
```

02 권한 관리



❖ 객체 권한 취소 : REVOKE 문

- 객체 소유자가 다른 사용자에게 부여한 객체의 사용 권한을 취소

```
REVOKE 권한 ON 객체 FROM 사용자 CASCADE ¦ RESTRICT;
```

- 사용자 A가 사용자 B에게, 사용자 B는 사용자 C에게 같은 권한을 부여한 경우
  - CASCADE 옵션
    - 권한을 취소할 사용자 A가 B뿐 아니라 C가 부여받은 권한도 연쇄적으로 함께 취소
  - RESTRICT 옵션
    - 권한을 취소할 사용자 A가 C가 부여받은 권한은 취소하지 않도록 함

02 권한 관리

❖ 객체 권한 취소 : REVOKE 문

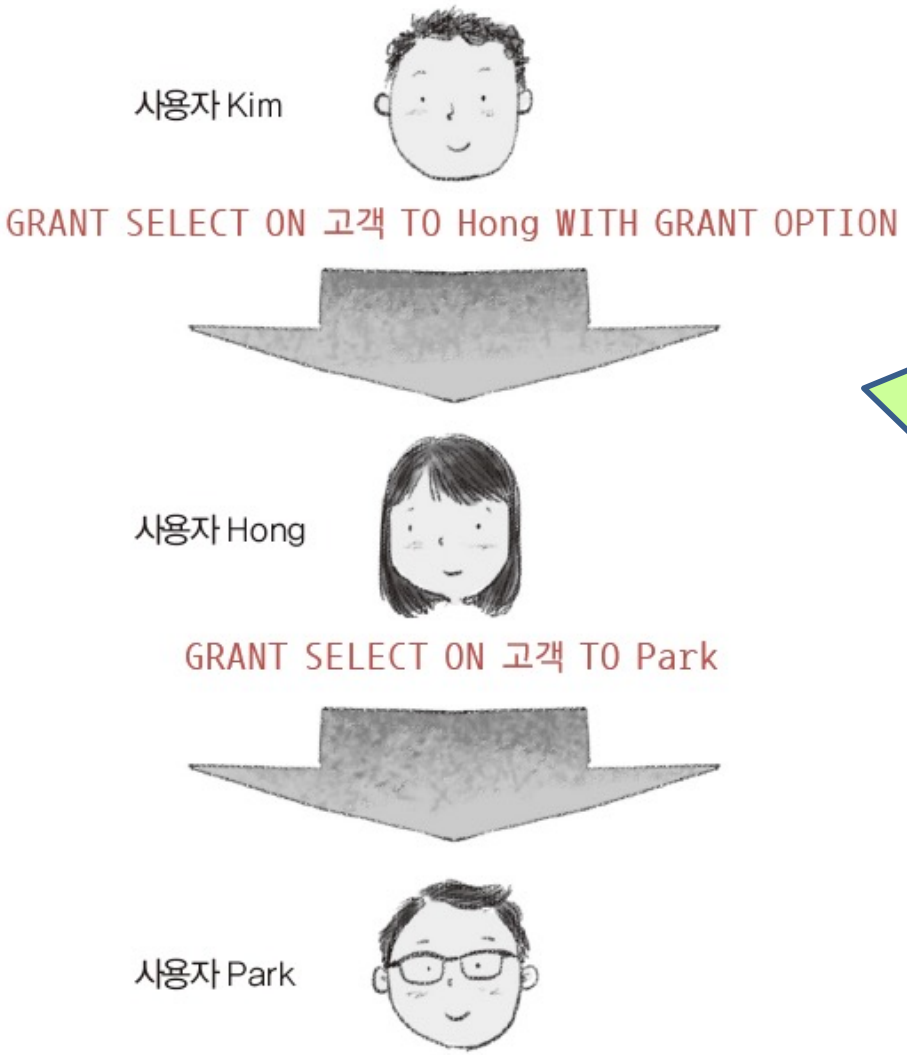


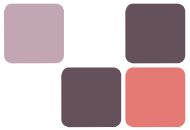
그림 11-5 세 사용자에게 권한을 부여한 예

사용자 "Kim"이 "Hong"에게
부여한 고객 테이블에 대한
검색 권한을 취소한다면
"Park"에게 부여된
검색 권한은 어떻게 처리될까?

선택 가능

(CASCADE 또는 RESTRICT)

02 권한 관리



❖ 객체 권한 취소 : REVOKE 문

예제 11-7

[그림 11-5]와 같이 권한이 부여된 상황에서, Kim이 Hong에게 부여한 고객 테이블에 대한 검색 권한을 취소하면서 Hong이 다른 사용자에게 부여한 고객 테이블에 대한 검색 권한도 함께 취소하도록 해보자.

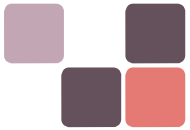
```
▶▶ REVOKE SELECT ON 고객 FROM Hong CASCADE;
```

예제 11-8

[그림 11-5]와 같이 권한이 부여된 상황에서, Hong이 다른 사용자에게 권한을 부여한 적이 없는 경우에만 Kim이 Hong에게 부여한 고객 테이블에 대한 검색 권한을 취소하는 명령문을 작성해보자.

```
▶▶ REVOKE SELECT ON 고객 FROM Hong RESTRICT;
```

02 권한 관리



❖ 시스템 권한 취소 : REVOKE 문

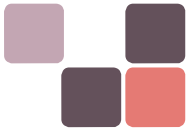
- 데이터베이스 관리자가 다른 사용자에게 부여한 시스템 권한을 취소
- 특정 객체에 대한 권한 취소가 아니므로 객체를 지정할 필요 없음

예제 11-9

Hong에게 부여한 테이블 생성 권한을 취소해보자.

```
▶▶ REVOKE CREATE TABLE FROM Hong;
```

02 권한 관리



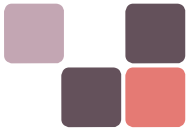
❖ 권한 목록

- 권한 부여에 관한 내용을 기록한 것
  - 사용자들에게 어떤 권한을 부여했는지, WITH GRANT OPTION을 포함하여 권한을 부여했는지 등

표 11-1 고객 테이블에 대한 각 사용자의 권한 목록

사용자 \ 권한	고객 테이블에 대한 권한
Kim	소유자
Hong	INSERT / DELETE / SELECT
Park	INSERT / DELETE / UPDATE(등급, 적립금)
Lee	INSERT / DELETE / SELECT(WITH GRANT OPTION)

02 권한 관리



❖ 역할(role)의 개념

- 여러 권한을 그룹으로 묶어 놓은 것
  - 권한들을 넣어둔 바구니

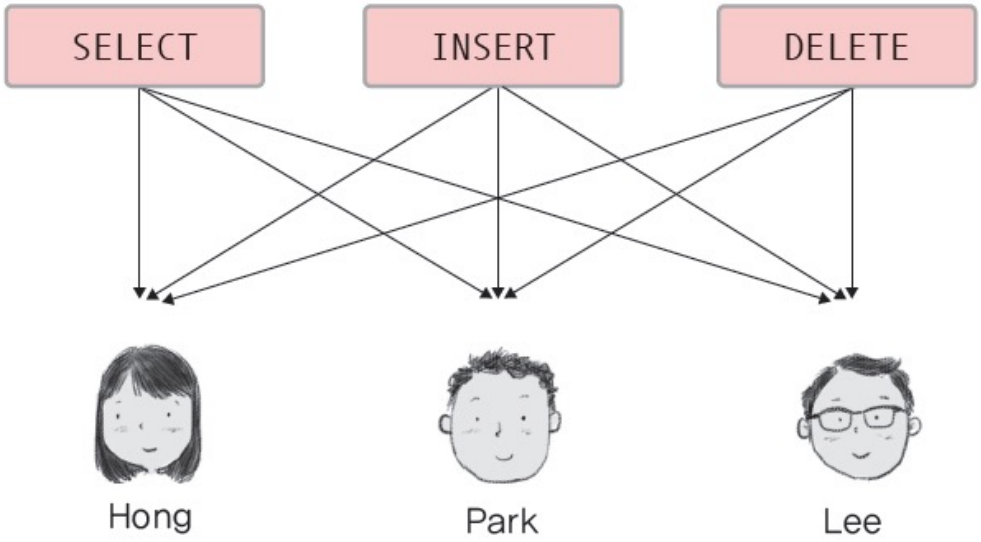
그림 11-7 권한과 역할

02 권한 관리

❖ 역할의 필요성

사용자 "Kim"이 자신의 고객 테이블에 대한 검색, 삽입, 삭제 권한을 "Hong", "Park", "Lee"에게 모두 부여하려면 작업이 번거로움

➔ **역할을 이용하면 훨씬 더 편리하게 작업할 수 있음**



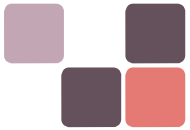

```
GRANT SELECT ON 고객 TO Hong;
GRANT INSERT ON 고객 TO Hong;
GRANT DELETE ON 고객 TO Hong;
```

```
GRANT SELECT ON 고객 TO Park;
GRANT INSERT ON 고객 TO Park;
GRANT DELETE ON 고객 TO Park;
```

```
GRANT SELECT ON 고객 TO Lee;
GRANT INSERT ON 고객 TO Lee;
GRANT DELETE ON 고객 TO Lee;
```

그림 11-6 3개의 권한을 사용자 세 명에게 부여하는 예

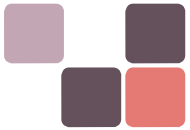
02 권한 관리



❖ 역할의 필요성

- 여러 사용자에게 동일한 권한들을 부여하고 취소하는 작업을 편리하게 수행할 수 있도록 함
  - 사용자에게 부여하고 싶은 여러 권한을 역할에 미리 넣어두고 필요할 때 역할을 부여하면 여러 권한을 한 번에 부여할 수 있음
  - 사용자에게 부여한 역할을 취소하면 한 번에 여러 권한을 취소할 수 있음
- 권한 관리가 쉬워짐
  - 새로운 권한의 추가, 기존 권한의 취소 등 역할에 변화가 생기면 해당 역할을 부여받은 모든 사용자에게 변화가 그대로 전달됨

02 권한 관리



❖ 역할 생성 : CREATE ROLE 문

- 새로운 역할의 생성은 데이터베이스 관리자가 담당

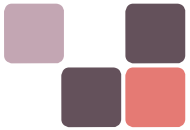
```
CREATE  ROLE  롤이름;
```

예제 11-10

role_1이라는 이름의 역할을 생성해보자.

```
▶▶ CREATE  ROLE  role_1;
```

02 권한 관리



❖ 역할에 권한 추가 : GRANT 문

- 객체와 관련된 권한을 역할에 추가하는 작업은 객체의 소유자가 담당

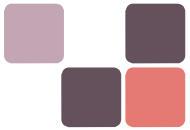
```
GRANT 권한 ON 객체 TO 롤이름;
```

예제 11-11

고객 테이블에 대한 검색·삽입·삭제 권한을 [예제 11-10]에서 생성한 role_1 역할에 넣어 보자.

```
▶▶ GRANT SELECT, INSERT, DELETE ON 고객 TO role_1;
```

02 권한 관리



❖ 역할 부여 : GRANT 문

- 역할을 사용자에게 부여하는 것은 데이터베이스 관리자가 담당

```
GRANT  롤이름  TO  사용자;
```

예제 11-12

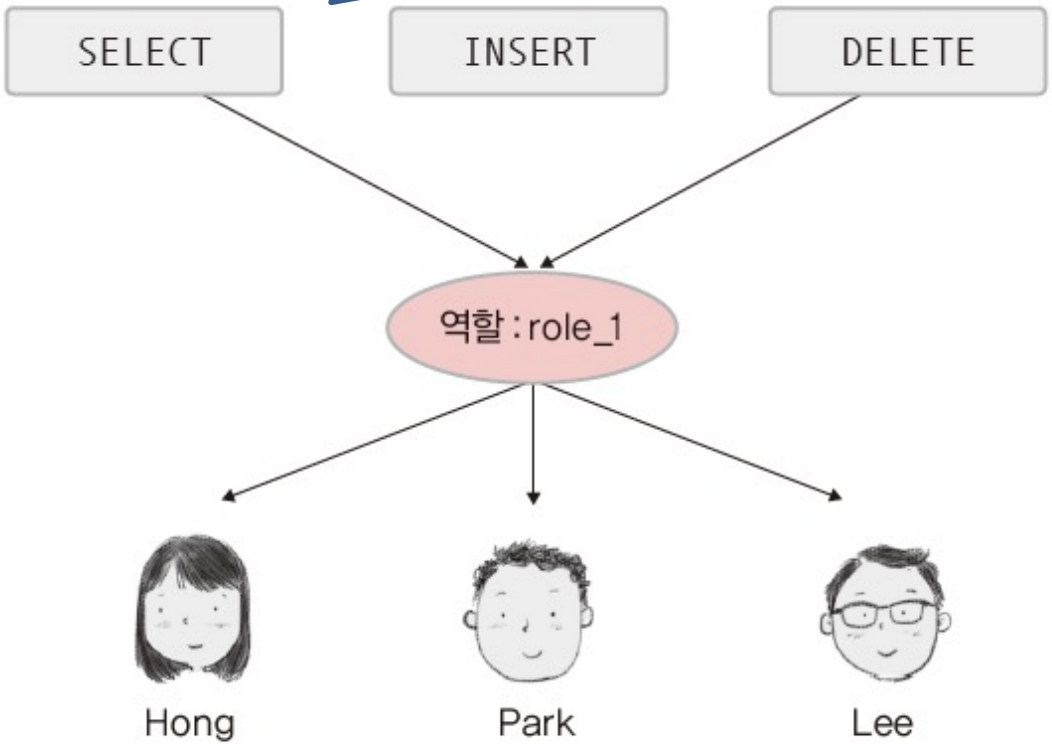
고객 테이블에 대한 검색·삽입·삭제 권한을 포함하고 있는 role_1 역할을 사용자 Hong에게 부여해보자.

```
▶▶ GRANT  role_1  TO  Hong;
```

02 권한 관리

❖ 역할을 이용한 예

역할을 이용하면 사용자 "Kim"이 자신의 고객 테이블에 대한 검색, 삽입, 삭제 권한을 "Hong", "Park", "Lee"에게 손쉽게 부여할 수 있고 새로운 권한의 추가도 간편하게 수행됨



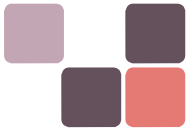

```
GRANT role_1 TO Hong;
```

```
GRANT role_1 TO Park;
```

```
GRANT role_1 TO Lee;
```

그림 11-8 역할을 이용해 3개의 권한을 세 명의 사용자에게 부여하는 예

02 권한 관리



❖ 역할 취소 : REVOKE 문

- 사용자에게 부여한 역할의 취소는 데이터베이스 관리자가 담당

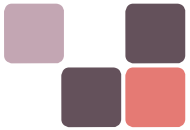
```
REVOKE  롤이름  FROM  사용자;
```

예제 11-13

사용자 Hong에게 부여한 role_1 역할을 취소해보자.

```
▶▶ REVOKE  role_1  FROM  Hong;
```

02 권한 관리



❖ 역할 제거 : DROP ROLE 문

- 역할을 제거하면 제거된 역할을 부여받은 모든 사용자에 대해 역할에 속해 있던 권한이 모두 취소됨
- 역할 제거는 데이터베이스 관리자가 담당

```
DROP  ROLE  롤이름;
```

예제 11-14

[예제 11-10]에서 생성한 role_1 역할을 제거해보자.

```
▶▶ DROP  ROLE  role_1;
```

Thank You