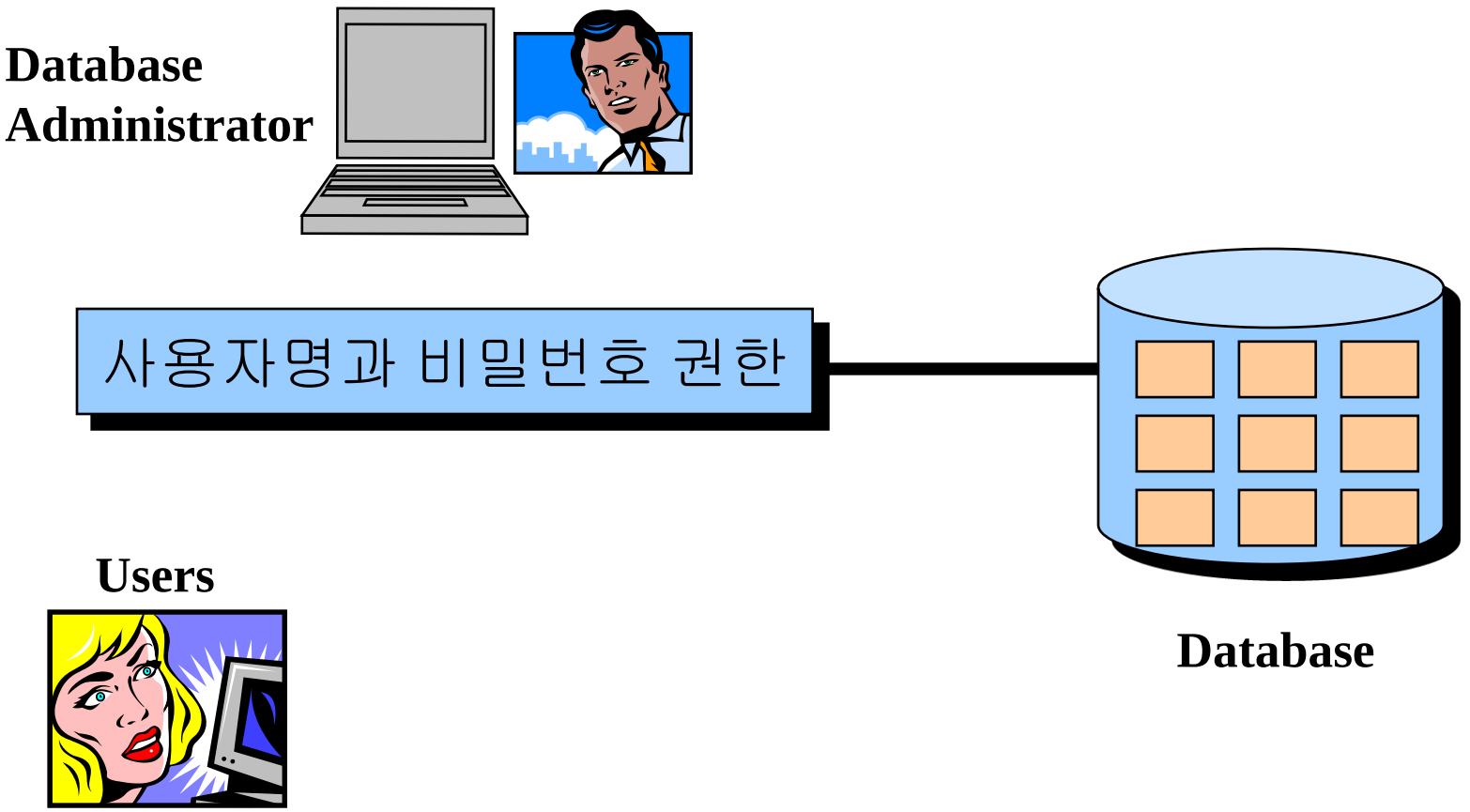
제 1장. 사용자 접근 제어

제 1장. 사용자 접근 제어

1) 목 적
2) 사용자 접근 제어
3) 사용자 생성
4) 권한(Privilege)
5) 시스템 권한(SYSTEM Privilege)
6) 객체 권한(OBJECT Privilege)
7) 롤(Role)
8) 요 약

2) 사용자 접근 제어



2-1) 데이터베이스 액세스와 보안

- 다중 사용자 환경에서는 **데이터베이스 액세스**와 **보안 유지**가 요구됨.

1) 데이터베이스 액세스

- 데이터베이스 액세스 제어.
- 데이터베이스에서 **특정 객체에 대한 액세스** 제공.
- 오라클 **데이터 사전**으로 주어지고 받는 **privilege 확인**.
- 데이터베이스 **객체**에 대한 **동의어 생성**.

2) 데이터베이스 보안

- 시스템 보안

  - **사용자명**과 **비밀번호, 사용자에게 할당된 디스크 공백, 사용자에 의해 허용된 시스템 작업** 같은 **시스템 수준**에서의 **데이터베이스 액세스와 사용**을 설명함.

- 데이터 보안

  - 데이터베이스 **객체에 대한 액세스와 사용, 객체에 대해서 사용자가 할 수 있는 작업**을 설명함.

3) 사용자 생성(CREATE USER)

❖ DBA는 **CREATE USER 문장**을 사용하여 사용자를 생성.

- 이때 사용자는 어떠한 권한도 가지지 않음.
- DBA는 사용자 생성 후, 사용자에게 여러 권한을 부여함.

❖ 사용자 생성의 일반적인 형태

```
CREATE USER user_name IDENTIFIED [BY password | EXTERNALLY]
[DEFAULT      TABLESPACE     tablespace_name]
[TEMPORARY  TABLESPACE     tablespace_name]
[QUOTA {integer [ K | M ] | UNLIMITIED } ON tablespace_name
  [QUOTA {integer [ K | M ] | UNLIMITIED } ON tablespace_name ] …]
[PASSWORD EXPIRE ]
[ACCOUNT { LOCK | UNLOCK } ]
[PROFILE    { profile_name | DEFAULT } ]
```

참고1) 사용자 생성 키워드

❖ **IDENTIFIED BY password** 키워드
- 사용자가 **데이터베이스 인증**되도록 지정.

❖ **IDENTIFIED EXTERNALLY** 키워드
- 사용자가 **운영체제 인증**되도록 지정.

❖ **QUOTA** 키워드
- **테이블스페이스내에 사용자가 소유한 오브젝트에 할당되는 최대 공간**을 지정.

❖ **UNLIMITED** 키워드
- 사용자가 소유한 오브젝트가 **테이블스페이스에서 사용 가능한 공간을 전부 사용**할 수 있도록 지정.
- DEFAULT 값은 사용자는 어떠한 테이블스페이스에도 할당을 받지 못함.

❖ **PASSWORD EXPIRE** 키워드
- 사용자가 SQL*PLUS를 사용하여 데이타베이스에 로그인할 때, **암호를 재설정**하도록 지정.
- 사용자가 **데이타베이스에 의해 인증될 경우**에만 적합한 옵션임.

❖ **ACCOUNT LOCK/UNLOCK** 키워드
- **사용자 계정을 명시적으로 잠그거나 풀 때 사용**하는 키워드.
- DEFAULT 값은 **UNLOCK** 임.

❖ **PROFILE** 키워드
- 자원 사용을 제어하고 사용자에게 사용되는 암호제어 처리방식을 지정(**프로파일 지정**).

3-1) 사용자 생성의 예

```
SQL> CREATE USER peter IDENTIFIED BY my1stson
  2> DEFAULT       TABLESPACE    data01
  3> TEMPORARY TABLESPACE    temp
  4> QUOTA  15M   ON  data01
  5> PASSWORD EXPIRE;
```

```
SQL> CREATE USR scott IDENTIFIED BY tiger;
User created.
```

3-2) 사용자 생성 지침 사항

- 초기에는 **표준암호**를 선택하고, **O/S 인증**은 되도록 **삼가**할 것.
- 사용자가 **암호를 재설정**하도록 하려면, **EXPIRE** 키워드를 사용.
- 항상 **임시 테이블스페이스(TEMPORARY TABLESPACE)**를 **할당**할 것.
- 소수 사용자에게로 할당량을 제한할 것.
- **QUOTA UNLIMITED**는 주의 깊게 사용할 것.
- 사용자에게 **접속**하는 방법과 **암호를 변경**하는 방법을 교육할 것.

3-3) 사용자 변경(ALTER USER)

❖ 사용자 계정이 생성되었을 때, 각 사용자는 **DBA가 초기화한 비밀번호**를 갖게 됨.

❖ 사용자는 **ALTER USER** 문장을 사용하여, **암호**를 **변경**하고 **계정**을 **잠글** 수 있음.

❖**사용자**가 **로그인한 상태**라면 **암호 변경, 암호 만료, 계정 잠금**은

현재의 세션에는 영향을 주지 않고, **다음 세션**에만 **유효**함.

- 사용자가 암호를 잃어버려서 암호를 재설정하고자 할 경우
- 시스템에 의해 잠겨진 사용자 계정을 풀 경우
- 계정을 명시적으로 잠글 경우
- 수동으로 암호를 만료되게 하거나, 사용자 암호를 재설정할 경우

```
ALTER USER username [IDENTIFIED {BY password | EXTERNALLY } ]
[PASSWORD EXPIRE]
[ACCOUNT { LOCK | UNLOCK }];
```

```
SQL> ALTER USER peter IDENTIFIED BY hisgrandpa;
    2> PASSWORD EXPIRE;
User altered.
```

```
SQL> ALTER USER scott IDENTIFIED BY lion;
User altered.
```

4) 권 한(Privilege)

❖ 권 한(Privilege)

- 특정 **SQL**문장을 실행하기 위한 권한.
- 권한은 사용자**(User) or** 롤**(Role)**에게 부여함.

❖ 데이타베이스 관리자(DBA)

- 데이타베이스와 객체에 대한 액세스를 사용자에게 부여하는 능력을 가진 상급 사용자.

❖ 스키마(Schema)

- 테이블, 뷰, 시퀀스, 인덱스 같은 객체 모음.
- 데이타베이스 사용자에 의해 소유되고, 사용자와 동일 이름을 가짐.

4-1) 권한(Privilege)의 2종류

1) 시스템 권한 (SYSTEM Privilege)

- 사용자의 데이터베이스 액세스 권한.
- 사용자가 데이터베이스에 특별한 작업을 수행하는 것을 가능하게 함.
- 대략 **80**개 정도의 시스템 권한이 있음.

2) 객체 권한 (OBJECT Privilege)

- 데이타베이스 내의 객체의 내용을 조작하기 위한 권한.
- 사용자가 특정 객체에 접근하고, 조작하는 것을 가능하게 함.

5) 시스템 권한(SYSTEM Privilege)

❖ 사용자와 롤에 대해 80개 이상의 권한이 있음.
❖ **시스템 권한**은 대개 데이타베이스 관리자**(DBA)**에 의해 제공됨.
❖ DBA는 상급의 시스템 권한이 있음(**주된 DBA 권한**)

- 새로운 사용자 생성(**CREATE USER** 명령)
- 사용자 제거(**DROP USER** 명령)
- 테이블 제거(**DROP ANY TABLE** 명령)
- 테이블 백업(**BACKUP ANY TABLE** 명령)

❖ 사용자 or 롤에게 권한 부여와 철회하는 DDL 명령

- **GRANT** 명령 : 사용자, 사용자 그룹에게 권한 부여.
- **REVOKE** 명령 : 권한 철회.

❖ **ANY** 키워드

- '**모든 스키마에 대해 권한을 가짐**'을 의미
- **ANY 권한**을 가진 자는 **접두어 USER와 ALL을 제외**한 **모든 딕셔너리 테이블**과 PUBLIC에게 부여된 권한 상의 **모든 뷰**에 접근할 수 있음.

참고1) 시스템 권한 예

범 주	예
테이블 스페이스	CREATE TABLESPACE ALTER TABLESPACE DROP TABLESPACE UNLIMITED TABLESPACE
테이블	CREATE TABLE CREATE ANY TABLE ALTER ANY TABLE DROP ANY TABLE SELECT ANY TABLE UPDATE ANY TABLE DELETE ANY TABLE
인덱스	CREATE ANY INDEX ALTER ANY INDEX DROP ANY INDEX
세션	CREATE SESSION ALTER SESSION RESTRICTED SESSION

❖ **CREATE INDEX 권한은 없음.**

❖ **CREATE TABLE, CREATE PROCEDURE, CREATE CLUSTER** 같은 권한은 **해당 오브젝트를 삭제하는 권한도 포함함.**

❖ **CREATE TABLE**은 **CREATE INDEX와 ALALYZE명령을 포함함.**

❖ 테이블을 잘라버리려면(**Truncate**), **DROP ANY TABLE**권한이 필요함.

❖ **전체 시스템 권한 목록**를 보려면, **SYSTEM_PRIVILEGE_MAP** 뷰를 질의함.

<시스템 권한 분류>

❖ **전 시스템**에 걸친 작업을 가능하게 해주는 권한.
- 예) **CREATE SESSION, CREATE TABLESPACE**

❖ **사용자 자신의 스키마** 내의 오브젝트에 대한 관리를 가능하게 해주는 권한.
- 예) **CREATE TABLE**

❖ **모든 스키마**에 있는 오브젝트에 대한 관리를 가능하게 해주는 권한.
- 예) **CREATE ANY TABLE**

5-1) 시스템 권한 부여(GRANT문)

- 권한(Privilege)은 데이터베이스 수준에서 사용자가 할 수 있는 것이 무엇인지를 결정함.

❖ 일단 **사용자가 생성**되었다면,
DBA는 **사용자에 대한 특정 시스템 권한**을 **부여할** 수 있음.

```
GRANT {system_priv | role}    [, {system_priv | role}] …
TO        {user | role | PUBLIC} [, {user | role | PUBLIC}] …
[WITH ADMIN OPTION];
```

```
SQL> GRANT create table, create sequence, create view TO scott;
Grant succeeded.
```

- **PUBLIC** 키워드

  - 모든 사용자에게 시스템 권한을 부여함.

- **WITH ADMIN OPTION** 키워드

  - 사용자가 **부여 받은 권한 or 롤**을 **다른 사용자 or 롤에게 부여**할 수 있도록 함.

참고1) 시스템 권한 부여 지침 사항

- **시스템 권한**을 **부여**하려면 **WITH ADMIN OPTION** 권한을 부여 받아야 함.
- WITH ADMIN OPTION과 함께 권한을 부여 받은 사용자는
  데이타베이스의 모든 사용자 or 롤에게
  시스템 권한 or 롤을 WITH ADMIN OPTION과 함께 다시 부여할 수 있음.
- **GRANT ANY ROLE** 시스템 권한을 가진 사용자는
  데이타베이스의 모든 롤을 **부여할** 수 있음.
- **애플리케이션 개발자**는 다음 **시스템 권한**을 가질 수 있음.
  - **CREATE SESSION**
  - **CREATE TABLE**
  - **CREATE SEQUENCE**
  - **CREATE VIEW**
  - **CREATE PROCEDURE**

5-2) 시스템 권한 철회(REVOKE문)

REVOKE { system_priv | role} [, {system_priv | role}]…
FROM { user | role | **PUBLIC** } [, {user | role | PUBLIC}]…

REVOKE create table, create session FORM scott;

❖ REVOKE 명령은 GRANT명령으로 직접 부여했던 권한만을 철회할 수 있음.

❖ WITH ADMIN OPTION을 사용했는지의 여부와 상관없이,
시스템 권한이 철회될 때는 **연쇄 효과(CASCADE) 가 없음.**

5-3) 시스템 권한 제한사항

❖ Oracle8의 딕셔너리 보호 방식

- **07_DICTIONARY_ACCESSIBILITY = FALSE**
- 허가 받지 않은 사용자가 딕셔너리 오브젝트에 접근하는 것을 막아줌.
- **딕셔너리 오브젝트에 접근**하는 것은 시스템 권한 **SYSDBA와 SYSOPER를 가진 사용자**로 제한됨.
- 다른 스키마의 오브젝트에 접근할 수 있는 시스템 권한일지라도 **딕셔너리 오브젝트에 접근할 수 없음.**
- **SELECT ANY TABLE 시스템 권한**은 SYS 스키마를 제외한 **다른 스키마의 뷰와 테이블에 접근 가능.**

❖ Oracle7의 딕셔너리 보호 방식

- **07_DICTIONARY_ACCESSIBILITY = TRUE**
- **SELECT ANY TABLE 시스템 권한**을 가진 사용자도
  SYS 스키마의 오브젝트인 **딕셔너리 오브젝트에 접근 가능** 함.

6) 객체 권한(OBJECT Privilege)

- 객체 권한은 객체의 타입에 따라 다양함.
- 소유자는 객체에 대한 모든 권한을 가짐.
- 사용자는 자동적으로

  자신의 스키마에 포함된 스키마 객체에 대한 모든 객체 권한을 가짐.
- 소유자는 사용자 자신의 객체에 대한 특정 권한을 제공할 수 있음.

  (WITH GRANT OPTION)

6-1) 객체 유형과 부여 가능한 객체 권한

Object Privilege	Table	View	Sequence	Procedure
ALTER	V		V	
DELETE	V	V		
EXECUTE				V
INDEX	V			
INSERT	V	V		
REFERENCES	V			
SELECT	V	V	V	
UPDATE	V	V		

6-2) 객체 권한 부여(GRANT문)

```
GRANT { object_priv [ (column_list)] [, object_priv [(column_list)] ] … | ALL [PRIVILEGES] }
ON        [schema.]object
TO        { user | role | PUBLIC } [, { user | role | PUBLIC } ] …
[WITH GRANT OPTION] ;
```

- **ALL** 키워드

  - WITH GRANT OPTION 으로 부여된 오브젝트에 대한 모든 권한 부여.

- **PUBLIC** 키워드

  - 모든 사용자에게 객체 권한 부여함.

- **WITH GRANT OPTION** 키워드

  - 부여 받은 사용자가 다른 사용자 or 롤(Role)에게 오브젝트 권한을 부여할 수 있음.

참고1) 객체 권한 부여 지침 사항

- **객체 소유자**는 자동적으로,
  자신의 스키마에 포함된 스키마 객체에 대한 모든 객체 권한을 가짐
- **권한을 부여**하려면, **오브젝트**가 **자신의 스키마에 존재**하거나,
  WITH GRANT OPTION 으로 객체 권한을 부여 받아야 함.
- 기본적으로 **소유한 오브젝트**에 대한 **모든 권한이 자동적으로 획득됨.**
- **보안**을 고려해야 한다면,
  자신의 오브젝트에 대한 권한을 다른 사용자에게 부여할 때 주의할 것.
- WITH GRANT OPTION은 **롤(Role)에 권한을 부여할 때는 사용할 수 없음.**

6-2) 객체 권한 부여의 예(1)

1) EMP 테이블에 대한 질의 권한 부여.

```
SQL> GRANT    select
  2>  ON          emp
  3>  TO          sue, rich;
Grant succeeded.
```

2) 사용자와 롤에게 지정 열을 갱신하기 위한 권한 부여

```
SQL> GRANT    update  (dname, loc)
  2>  ON          dept
  3>  TO          scott, manager;
Grant succeeded.
```

6-3) 객체 권한 부여의 예(2)

3) **권한을 전달**하기 위한 사용자 권한 제공

```
SQL> GRANT    select, insert
  2>  ON          dept
  3>  TO          scott
  4>  WITH GRANT OPTIN;
Grant succeeded.
```

4) 모든 사용자가 Alice의 DEPT 테이블의 데이터를 질의 하도록 허용.

```
SQL> GRANT     select
  2>  ON          alice.dept
  3>  TO          PUBLIC;
Grant succeeded.
```

6-4) 부여된 권한 확인

데이터 사전 테이블	설 명
ROLE_SYS_PRIVS	롤에게 부여된 시스템 권한.
ROLE_TAB_PRIVS	롤에게 부여된 테이블 권한.
USER_ROLE_PRIVS	사용자에 의해 액세스 가능한 롤.
USER_TAB_PRIVS_MADE	사용자가 부여된 객체 권한.
USER_TAB_PRIVS_RECD	사용자에게 부여된 객체 권한.
USER_COL_PRIVS_MADE	사용자가 객체의 열에 대해 부여한 객체 권한.
USER_COL_PRIVS_RECD	특정 열에 대해 사용자에게 부여된 객체 권한.

6-5) 객체 권한 철회(REVOKE문)

❖ 다른 사용자에 대해 부여된 **권한을 철회**하기 위해 **REVOKE 문장**을 사용함.

❖ WITH GRANT OPTION으로 부여된 객체 권한을 철회하면, 연쇄적으로 철회됨

- **연쇄 효과(CASCADE) 있음.**

❖ 권한 부여자(Grantor)는 자신이 부여했던 사용자로부터만 권한을 철회할 수 있음.

```
REVOKE   {object_priv [, privilege] … | ALL [PRIVILEGE] }
ON       [schema.]object
FROM     { user | role | PUBLIC } [, { user | role | PUBLIC } ] …
[CASCADE CONSTRAINTS];
```

- **ALL 키워드**
  - 사용자에게 부여된 모든 객체 권한 철회.
- **CASCADE CONSTRAINT 키워드**
  - REFERENCES 또는 ALL 권한 철회 시, **관련 참조 무결성 제약 조건을 삭제함.**

6-6) 객체 권한 철회의 예

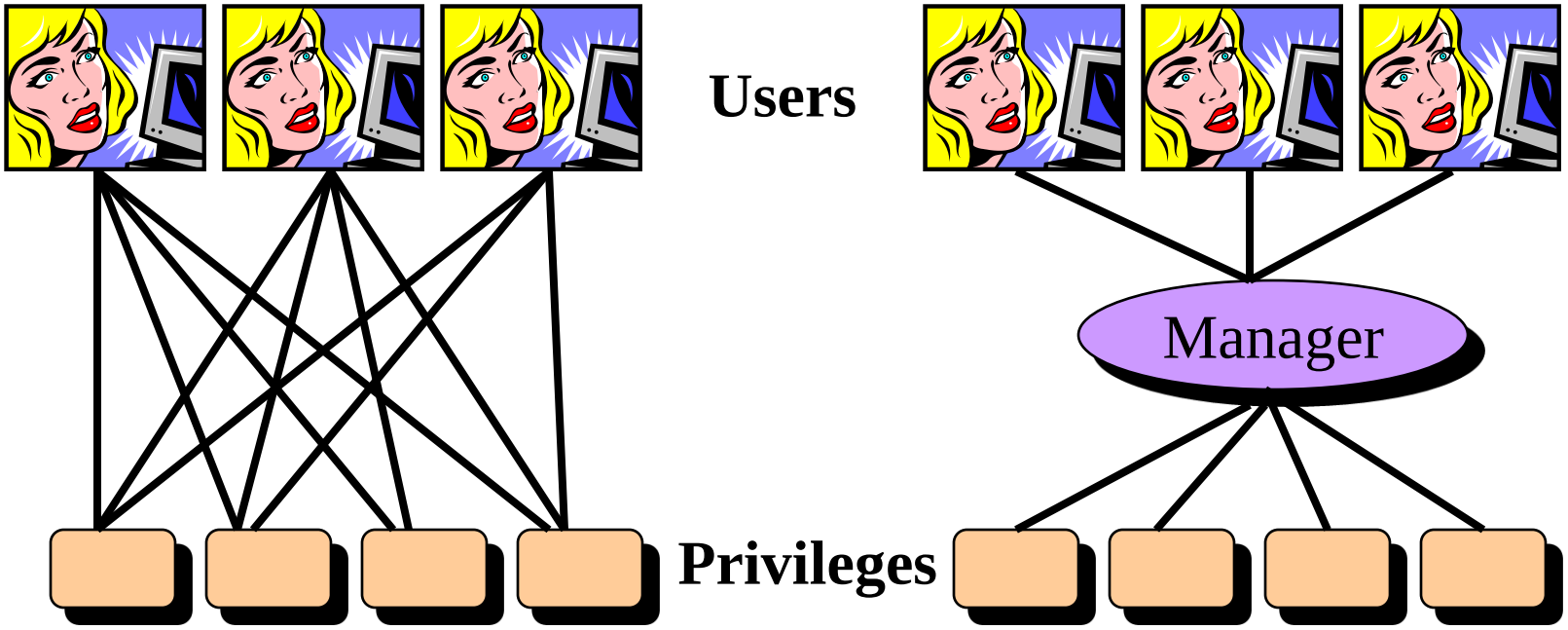
❖ 사용자 Alice로서, DEPT 테이블에 대해

사용자 Scott에게 주어진 SELECT와 INSERT권한을 철회함.

```
SQL> REVOKE   select, insert
  2> ON           dept
  3> FROM       scott;
Revoke succeeded.
```

7) 롤(Role)

- **롤(Role)**은 **사용자에 대해 부여**될 수 있는 **관련 권한(Privilege)의 그룹**임.
- 롤(Role)은 권한 관리를 쉽고 간편하게 해줌.
- 한 사용자가 여러 롤을 액세스할 수 있고, 다른 여러 사용자에게 동일한 롤을 지정할 수 있음.



7-1) Role의 특성

❖ 시스템 권한을 부여하고 철회할 때 사용되는 것과 동일한 명령으로 사용자에게 **부여(GRANT)**하고, **철회(REVOKE)**할 수 있음.

❖ **롤(Role) 자신을 제외**한 모든 **사용자(User)와 롤(Role)에게 부여**할 수 있음.

❖ **롤(Role)**은 **원형적으로 부여할 수 없음.**

- 예). **Role_A가 Role_B에게 GRANT**하였다면, **Role_B는 Role_A에게 GRANT할 수 없음.**

❖ **시스템 권한**과 **오브젝트 권한**으로 이루어짐.

❖ 각 사용자가 부여 받은 **롤(Role)을 enable/disable** 할 수 있음.

❖ **암호를 요구**하도록 할 수 있음.

❖ **각 롤(Role) 이름은** 기존의 사용자 이름과 롤 이름과는 다른 **유일한 것**이어야 함.

❖ 누군가가 소유하는 것도 아니고, 어느 스키마에 속하는 것도 아님.

❖ **데이터 딕셔너리**에 설명되어져 있음.

7-2) Role의 장점

❖ 권한 부여 작업 감소

- 권한들을 하나의 롤(Role)에 부여한 후, 그 롤(Role)을 각 사용자에 부여함으로 권한 관리가 간단해 짐.

❖ 동적 권한 관리

- 롤(Role)과 관련된 권한이 수정되면, 해당 롤(Role)을 부여받은 모든 사용자는 자동적으로 즉시 수정된 권한을 갖게 됨.

❖ 권한의 선택적 사용

- 롤(Role)은 enable/disable으로 되어, 임시로 권한을 ON/OFF으로 할 수 있음.
- 롤(Role)을 enable하는 것은 사용자가 그 롤(Role)을 부여 받았는지 검증하는데 사용될 수 있음.

❖ OS를 통해 부여

- 운영체제 명령, 유틸리티으로 데이타베이스의 사용자에게 롤(Role)을 할당하는데 사용될 수 있음.

❖ 연쇄적으로 철회하지 않음.

- 연쇄(CASCADE) 철회를 발생하지 않고 오브젝트 권한을 철회할 수 있음**(WITH ADMIN OPTION).**

❖ 향상된 성능

- 롤(Role)을 disable하여 실행 중 검증해야 할 권한을 줄일 수 있음.
- 롤(Role)을 사용하면 데이터 딕셔너리에 저장된 부여에 대한 정보가 줄어 듬.

7-3) Role의 생성(CREATE ROLE)

❖ 롤(Role)을 생성하는 일반적인 형태

CREATE ROLE role_name
[NOT IDENTIFIED | **IDENTIFIED** {**BY password** | EXTERNALLY}]

- NOT IDENTIFIED 키워드
  - 롤(Role)을 enable할 때 아무런 검증도 필요하지 않음을 의미.
- IDENTIFIED BY password 키워드
  - 롤(Role)을 enable할 때 데이타베이스 검증이 필요함을 의미.
- EXTERNALLY 키워드
  - 롤(Role)을 enable하기 전에 외부 서비스(운영체제)에 의해 인증을 받아야 함을 의미.
- ORACLE SECURITY SERVER
  - OEM의 롤(Role)과 사용자(User)를 중점적으로 설정하도록 해주는 보안 제품.

참고1) Role의 생성

❖ manager라는 이름의 Role 생성

```
SQL> CREATE ROLE manager;
Role created.
```

❖ manager 롤에게 create table, create view 권한 할당

```
SQL> GRANT create table, create view TO manager;
Grant succeeded.
```

참고2) Role 생성시 지침 사항

- **롤(Role)**이 **임무를 수행하는데 필요한 권한**을 포함하고 있기 때문에,
  롤(Role) 이름은 보통 **응용프로그램 작업 or 직무 이름**임.

1. 각 **응용 프로그램 작업을 위한 롤(Application Role)** 을 생성 함.
   예). 롤(Role) 이름은 응용프로그램 작업에 대응 됨 : Benefits, Payroll
2. 응용 프로그램 롤(Role)에 작업을 수행하는데 필요한 **권한(Privilege)을 할당**.
3. 각 유형의 **사용자를 위한 롤(User Role)**을 생성함.
   예). 롤(Role) 이름은 직무 이름에 대응됨 : pay_clerk
4. **사용자 롤(User Role)**에 **응용프로그램 롤(Application Role)**을 부여함.
5. **사용자**에게 **사용자 롤과 응용프로그램 롤**을 부여함.

7-4) 오라클 미리 정의된 롤(Role)

다음 롤(Role)들은 오라클 데이타베이스에서 자동적으로 정의됨.

롤(Role) 이름	설 명
CONNECT	이전 버전과의 호환을 위해서 제공됨.
RESOURCE	
DBA	모든 시스템 권한과 WITH ADMIN OPTION
EXP_FULL_DATABASE	DB를 Export할 권한
IMP_FULL_DATABASE	DB를 Import할 권한
DELETE_CATALOG_ROLE	Data Dictionary 테이블에 대한 DELETE 권한
EXECUTE_CATALOG_ROLE	Data Dictionary 패키지에 대한 EXECUTE 권한
SELECT_CATALOG_ROLE	Data Dictionary 테이블에 대한 SELECT 권한

7-5) Role의 수정(ALTER ROLE)

- 롤(Role)의 **인증 방식을 변경**하기 위해 수정될 수 있음.

❖ Role 수정의 일반적인 형태

```
ALTER ROLE role_name
[NOT IDENTIFIED | IDENTIFIED {BY password | EXTERNALLY} ]
```

❖ hr_clerk 이름의 Role 생성

```
CREATE ROLE hr_clerk;
```

❖ hr_clerk의 Role을 수정(인증 방법의 수정)

```
ALTER ROLE hr_clerk IDENTIFIED BY commission;
```

```
ALTER ROLE hr_clerk IDENTIFIED EXTERNALLY;
```

```
ALTER ROLE hr_clerk NOT IDENTIFIED;
```

7-6) Role의 권한 할당과 부여(GRANT)

- 사용자에게 롤(Role)을 부여하려면, 권한 부여시와 같은 **GRANT 명령**을 사용함.

❖ 롤(Role) 지정의 일반적인 형태

```
GRANT role_name [, role_name] …
TO        { user | role | PUBLIC } [, {user | role | PUBLIC} ]…
[WITH ADMIN OPTION]
```

❖ manager라는 이름의 Role 생성

```
SQL> CREATE ROLE manager;
```

❖ manager 롤에게 create table, create view 권한 할당

```
SQL> GRANT create table, create view TO manager;
```

❖ BLAKE와 CLARK에게 manager 롤(Role) 부여

```
SQL> GRANT manager TO BLAKE, CLARK;
```

7-7) DEFAULT ROLE 지정(ALTER USER)

- 사용자가 세션을 시작하면 기본 롤(DEFAULT ROLE)만 enable 됨.
- **DEFAULT ROLE**은 사용자가 로그인할 때, 자동적으로 enable되는 할당된 롤(Role)의 부분집합 임.
- 디폴트로 사용자에게 할당된 모든 롤(Role)이 사용자 로그인 시에 enable 됨.
- DEFAULT ROLE이 되려면 반드시 사전에 부여되어 있어야 함.
- 따라서, CREATE USER 명령으로 DEFAULT ROLE을 설정할 수 없음.
- 암호로 인증되는 롤(Role)의 경우, **DEFAULT ROLE**으로 만들 때에는 암호가 필요하지 않음.

❖ DEFAULT Role을 생성하는 일반적인 형태

ALTER USER user_name
DEFAULT ROLE { role, [, role] … | **ALL** [**EXCEPT** role [, role] …| **NONE**]

- **ALL** 키워드
  - 사용자에게 부여된 모든 롤을 DEFAULT ROLE으로 만듬.
- **ALL EXCEPT** 키워드
  - 사용자에게 부여된 롤 중 EXCEPT절에 나열된 것을 제외한 모든 롤을 DEFAULT ROLE으로 만듬.
- **NONE** 키워드
  - DEFAULT ROLE을 아무것도 부여하지 않음.
  - 로그인시 사용자가 갖는 권한은 사용자에게 직접 할당된 권한뿐임.

참고1) DEFAULT ROLE(기본 롤) 지정 예

❖ SCOTT 사용자에게 **hr_clerk, sales_clerk**의 직무 롤을 기본 롤로 지정.

```
ALTER USER scott DEFAULT ROLE hr_clerk, sales_clerk;
```

❖ SCOTT 사용자에게 **부여된 모든 롤**을 기본 롤로 지정.

```
ALTER USER scott DEFAULT ROLE ALL;
```

❖ SCOTT 사용자에게 부여된 롤 중,
hr_clerk 직무 롤을 제외한 모든 롤을 기본 롤로 지정.

```
ALTER USER scott DEFAULT ROLE ALL EXCEPT hr_clerk;
```

❖ SCOTT 사용자에게 기본 롤로 **아무것도 지정하지 않음.**

```
ALTER USER scott DEFAULT ROLE NONE;
```

7-8) Role의 Enable과 Disable

❖ **사용자**가 **세션을 시작(로그인)**하면 **DEFAULT ROLE만 enable됨.**

❖ 사용자로부터 임시로 롤(Role)을 철회하려면 그 롤을 disable 함.

❖ **SET ROLE** 명령과 **DBMS_SESSION.SET_ROLE 프로시저**는 롤(Role)을 enable/disable 함.

❖ SET ROLE 명령은 사용자에게 부여된 그 밖의 롤(Role)은 turn off 시킴.

❖ 롤(Role)을 enable/disable 하려면, 먼저 해당 롤(Role)이 사용자에게 부여되어 있어야 함.

❖ 암호가 포함된 롤(Role)은 암호가 SET ROLE 명령에 포함되어 있어야 함.

❖ 사용자가 다시 세션을 시작하면 다시 DEFAULT ROLE만 enable 됨.

❖ 롤(Role)은 PL/SQL 명령을 허용하는 어떤 도구나 프로그램에서 enable될 수 있음.

❖ 롤(Role)은 저장 프로시저로부터는 enable될 수 없음.

=> 저장 프로시저의 작업이 프로시저를 호출한 보안 도메인(권한 집합)을 변경할 수 있기 때문임.

❖ PL/SQL 에서는 익명의 블록 or 응용 프로그램 프로시저에서만 enable/disable될 수 있고, 저장 프로시저에서는 enable/disable될 수 없음.

참고1) Role의 enable/disable의 예

- SET ROLE 명령은 사용자에게 부여된 그 밖의 롤(Role)은 turn off 시킴.

❖ Role을 enable/disable하는 일반적인 형태

```
SET ROLE {  role [IDENTIFIED BY password]
                  [, role [IDENTIFIED BY password] …
          | ALL [EXCEPT role [, role]… ]
          | NONE        }
```

- 암호를 가진 sales_clerk 직무 롤(Role)을 활성화

```
SET ROLE sales_clerk IDENTIFIED BY commission;
```

- 암호가 없는 sales_clerk 직무 롤(Role)을 활성화

```
SET ROLE hr_clerk;
```

- sales_clerk을 제외한 현재 사용자에게 부여된 모든 롤(Role)을 활성화

```
SET ROLE ALL EXCEPT sales_clerk;
```

- 현재 사용자 세션에서 모든 롤을 비활성화

```
SET ROLE NONE;
```

7-9) Role의 철회(REVOKE)

- 롤(Role) 철회는 권한 철회와 같은 REVOKE 명령을 사용함.

❖ 롤(Role)을 제거하는 일반적인 형태

```
REVOKE  role [, role] …
FROM    { user | role | PUBLIC } [, { user | role | PUBLIC} ]…
```

- SCOTT 사용자로부터 sales_clerk 직무 롤 제거

```
REVOKE sales_clerk FORM scott;
```

- 모든 사용자로부터 hr_manager 직무 롤 제거

```
REVOKE hr_manager FORM PUBLIC;
```

7-10) Role의 제거(DROP ROLE)

- 데이타베이스로부터 **롤(Role)을 제거**하는 **DROP ROLE 명령**.
- 오라클 서버는 **롤(Role)을 삭제**할 때, 오라클 서버는
  삭제할 롤(Role)이 부여되었던 **모든 사용자와 롤(Role)으로부터 롤(Role)을 철회**한 후,
  데이터베이스로부터 제거 함.
- **WITH ADMIN OPTION**과 함께 롤(Role)을 부여 받았거나,
  DROP ANY ROLE 시스템 권한을 가지고 있어야만 **롤(Role)을 삭제**할 수 있음.

❖ 롤(Role)을 제거하는 일반적인 형태

```
DROP ROLE role_name;
```

- 데이타베이스로부터 hr_manager 직무 롤 제거

```
DROP ROLE hr_manager;
```

7-10) Role 정보 출력

롤(Role) 뷰	설 명
DBA_ROLES	데이타베이스에 존재하는 모든 롤(Role)
DBA_ROLE_PRIVS	사용자와 롤에 부여된 롤
ROLE_ROLE_PRIVS	롤에 부여된 롤
DBA_SYS_PRIVS	사용자와 롤에 부여된 시스템 권한
ROLE_SYS_RPIVS	롤에 부여된 시스템 권한
ROLE_TAB_PRIVS	롤에 부여된 테이블 권한
SESSION_ROLES	사용자가 현재 enable한 롤

8) 요 약

권 한	설 명
CREATE USER	DBA가 사용자를 생성할 것을 허용함.
GRANT	사용자가 사용자의 객체를 액세스하는 권한을 다른 사용자에게 부여하도록 함.
CREATE ROLE	DBA가 권한의 모음을 생성하는 것을 허용함.
ALTER USER	사용자가 그들의 비밀번호를 변경할 수 있도록 함.
REVOKE	사용자로부터 객체 권한을 제거함.