1. **롤(ROLE)**

>다수의 사용자와 다양한 권한(privilege)을 효과적(권한 부여, 제거)으로 관리하기 위하여 서로 관련된 권한을 한 곳에 묶은 그룹

>롤이란, 사용자나 사용자 그룹이 필요로 하는 여러 관련된 권한들을 한데 묶어서 이름을 붙여 놓은 권한의 집합을 의미

**1)롤 생성 -> 2) 롤에 권한 부여 -> 3) 사용자에게 롤 부여**

텍스트이(가) 표시된 사진

자동 생성된 설명

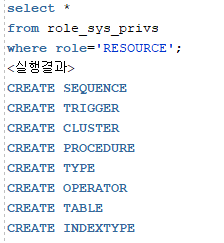
【형식】

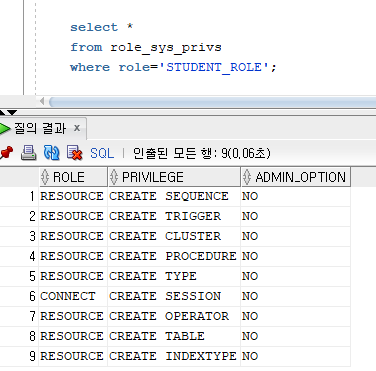
CREATE ROLE 롤이름

[NOT IDENTFIED 또는 IDENTIFIED

{BY PASSWORD 또는 EXTERNALLY 또는 GLOBALLY 또는 USING 패키지}];

* **롤에 부여된 시스템 권한이 무엇이 있는지 조회**





**롤 회수(제거)**

REVOKE 롤이름

FROM 계정명 또는 롤이름 또는 PUBLIC;

텍스트이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

**권한과 롤에 관한 정보 확인**

|  |  |
| --- | --- |
| 롤 조회에 사용되는 뷰 | |
| **view 이름** | **설명** |
| [dba\_roles](file:///C:\Ora_Help\oracle\dictionary\dba_roles.html) | 서버에 존재하는 모든 롤 조회 |
| [role\_sys\_privs](file:///C:\Ora_Help\oracle\dictionary\role_sys_privs.html) | 롤에 부여된 시스템 권한이 무엇무엇인지 조회 |
| [user\_role\_privs](file:///C:\Ora_Help\oracle\dictionary\user_role_privs.html) | 사용자에게 부여된 롤이 무엇무엇 인지를 조회 |
| [system\_privilege\_map](file:///C:\Ora_Help\oracle\dictionary\system_privilege_map.html) | 서버에 준비된 시스템 권한이 무엇무엇 인지를 조회 |
| [dba\_sys\_privs](file:///C:\Ora_Help\oracle\dictionary\dba_sys_privs.html) | 사용자와 롤에 부여된 모든 권한 정보 |
| [role\_tab\_privs](file:///C:\Ora_Help\oracle\dictionary\role_tab_privs.html) | 롤에 부여된 테이블 개체의 권한 정보 조회 |
| [role\_role\_privs](file:///C:\Ora_Help\oracle\dictionary\role_role_privs.html) | 롤에 부여된 다른 롤 조회 |
| [session\_roles](file:///C:\Ora_Help\oracle\dictionary\session_roles.html) | 현재 사용자의 세션에서 활성화 되어 있는 롤 조회 |

select \*

FROM dba\_roles; -- 관리자 권한을 가진 계정만 조회가능, SYS에서만 조회 가능

00942. 00000 - "table or view does not exist" - 오류 발생

테이블이(가) 표시된 사진

자동 생성된 설명

**SYS 계정에서… 확인…**

테이블이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**2. -- ㅂ. SID( 전역 데이터베이스 이름 )**

-- 설치된 오라클 DB의 고유한 이름

-- 오라클 무료 버전을 설치하면 자동으로 SID는 XE로 잡힘

-- 오라클 무료 버전은 1개만 설치 가능

**3. -- ㅅ. 데이터 모델**

**-- ㅈ. R+DBMS**

-- 컴퓨터에 데이터를 저장하는 방식을 정의해 놓은 개념 모델

-- 관계형 데이터 모델 사용 중

-- 관계형 데이터 모델 + DBMS = RDBMS

**4. -- ㅇ. 스키마(Schema)**

**1) DB에서 어떤 목적을 위하여 필요한 여러 개로 구성된 테이블들의 집합을 Schema라 한다**

**2) DATABASE SCHEMA(DB 스키마) ?**

-- USER A가 생성되면 자동적으로 동일한 이름의 SCHEMA A가 생성된다.

-- USER A는 SCHEMA A와 관련되어 DATABASE를 ACCESS한다. 그러므로 USER의 이름과 SCHEM A는 서로 바뀌어 쓰일 수 있다.

**-- 특정 USER와 관련된 모든 OBJECT의 모음**

-- scott 계정 생성 -> scott 계정이 사용할 수 있는 모든 OBJECT 만들어지고 이 모음을 '스키마'라고 한다.

-- emp 테이블(객체) 생성

FROM 스키마.테이블명

FROM scott.emp;

**3) 용어정리**

-- 인스턴스(instance) : 오라클 서버 -> 시작(startup)해서 종료(shutdown)할 때 까지

-- 세션(session) : 어떤 특정 사용자가 로그인해서 로그아웃 할 때 까지

-- 스키마(schema) : 특정 USER와 관련된 OBJECT(테이블 등..)의 모음

**5. user\_ , dba\_ , all\_**

SELECT \*

FROM user\_users; -- user\_ 접두사 : 현재 접속중인 USER(여기서는 SYS)가 접근할 수 있는 모든 사용자 정보 뷰(view)

FROM dba\_users; -- dba\_ 접두사 : 오라클 관리자만 사용할 수 있음 + DB 내의 모든 사용자의 데이터(정보) 뷰(view)

FROM all\_users; -- all\_ 접두사 : 모든 사용자 정보를 가져오는 view

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**6. 오라클 주석처리 방법 3가지를 적으세요.**

ㄱ. -- 한줄 주석처리

ㄴ. rem REM 한줄 주석처리

ㄷ. /\*\*/ 블럭 주석처리

**7. Oracle SQL Developer 에서 쿼리(SQL)을 실행하는 방법을 모두 적으세요.**

ㄱ. F9

ㄴ. F5

ㄷ. Ctrl + Enter

**8. 자료 사전( Data [Dictionary] ) 이란?**

메타 데이터(META DATA) == Data Dictionary의 정보

1)Data Dictionary = TABLE과 View들의 집합 -> View는 나중에 배울 예정..

2)역할 : DB의 정보를 제공하는 역할

SCOTT 계정

emp 테이블 생성

테이블이 생성되었다고 기록되어지는 곳이 자료 사전

그 안에 테이블의 생성시기, 소유자 등등이 나옴

3) DB 생성시 자동으로 SYS 계정 생성 -> SYS Schema 생성 -> 내부에 자료 사전(Data Dictionary) 생성

4) 새로 추가가 되어더라도 백업이 되어있다.

5) 자료사전은 DB 생성 후에 기본 테이블만 생성되어 있음. 따라서 DBA\_, ALL\_, USER\_로 시작하는 자료사전 View는 확인 X

6) 자료사전은 다음과 같이 시작되는 대표적인 뷰 4가지와 테이블 1가지

|  |  |
| --- | --- |
| [DBA\_](file:///C:\Ora_Help\oracle\dictionary\DBA_view.html) | DB 전체에 포함되는 모든 객체에 대한 자세한 정보 |
| [ALL\_](file:///C:\Ora_Help\oracle\dictionary\ALL_view.html) | 자신이 생성한 객체와 다른 사용자가 만든 객체중에서 자신이 볼 수 있는 정보를 제공한다. |
| [USER\_](file:///C:\Ora_Help\oracle\dictionary\USER_view.html) | 자신이 생성한 모든 객체에 대한 정보 |
| [V$\_](file:///C:\Ora_Help\oracle\dictionary\V_view.html) | DB의 성능분석/통계 정보를 제공하며 X$테이블에 대한 뷰이다. dynamic performance view는 V$fixed\_table를 조회하여, 리스트를 확인할 수 있음 |
| X$\_ | DB의 성능 분석/통계 정보를 제공하는 테이블 |

**7) 자료사전에서 관리하는 정보**  
• 데이터베이스의 물리적 구조와 객체의 논리적 구조  
• 오라클 사용자 이름과 스키마 객체 이름  
• 사용자에게 부여된 접근 권한과 롤  
• 무결성 제약조건에 대한 정보  
• 컬럼별로 지정된 기본 값  
• 스키마 객체에 할당된 공간의 크기와 사용중인 공간의 크기 정보  
• 데이터베이스 이름,버전,생성 날짜, 시작모드,인스턴스 이름정보

**9. SQL 문장 작성법**

• SQL 문장은 대소문자를 구별하지 않는다.

~~• SQL\*Plus에서 SQL 문장은 SQL 프롬프트에 입력되며, 이후의 Line은 줄번호가 붙는다.~~

• SQL 명령을 종료할 때는 세미콜론(;)을 반드시 붙여야 한다.

~~• 맨 마지막 명령어 1개가 SQL buffer에 저장된다.~~

• SQL 문장은 한 줄 이상일 수 있다.

• SQL 명령어를 구성하고 있는 단어 중 어디에서나 분리해도 된다.

• 한 개의 line에 한 개의 절(select, from, where) 형태로 나누어 입력하는 것을 권한다. 그러나 한 개의 단어를 두 줄로 나누어 입력해서는 안된다.

• keyword는 대문자로 입력하도록 권한다. 다른 모든 단어 즉, table 이름, column 이름은 소문자로 입력을 권한다. – 권장사항

• keyword는 단축하거나 줄로 나누어 쓸 수 없다.

• 절은 대개 줄을 나누어 쓰도록 권한다.

• 탭과 줄 넣기는 읽기 쉽게 하기 위해 사용을 권한다.

**10. PL/SQL = SQL + 절차적 언어 문법 -> SQL에 절차적 언어(Procedural Language)가 확장된 것**

SQL? 구조화된 질의 언어

**11. null값의 유무를 확인할 때 사용하는 SQL 연산자! IS [NOT] NULL**

SQL 연산자는 쿼리에 사용하는 연산자

Null은 아직 정해지지 않은 값으로 ‘’(빈문자)와 0하고는 다른 값

**IS [NOT] NULL**

• column이 완전히 비어 있으면 null임  
• ''(공백),0(zero)과는 다른 값임  
• null은 아직 정해지지 않은 값, 계산 될 수 없는 값을 의미  
• 있다, 없다의 값을 비교할 때는 IS NULL, IS NOT NULL을 사용

**12. emp 테이블에서 pay(sal+comm)가 1000 이상~ 2000 이하 받는 30부서원들만 조회하는 쿼리 작성**

조건 : ㄱ. pay 기준으로 오름차순 정렬 --ename을 기준으로 오름차순 정렬해서 출력(조회)

ㄴ. comm 이 null은 0으로 처리 ( nvl () )

**1번째 풀이 방법)**

텍스트이(가) 표시된 사진

자동 생성된 설명

WHERE **pay** BETWEEN 1000 AND 2000 AND deptno = 30 입력하면 아래와 같이 오류메시지 발생

오류 메시지:

ORA-00904: "PAY": invalid identifier -> WHERE은 별칭을 인식하지 못한다.

00904. 00000 - "%s: invalid identifier"

**SELECT절 처리순서가 WHERE이 먼저이기 때문에 SELECT에서 준 별칭을 사용할 수 없다. ORDER BY는 SELECT 다음에 처리되어서 별칭 사용 가능**

**2번째 풀이 방법) WITH절 사용**

텍스트이(가) 표시된 사진

자동 생성된 설명

**쿼리 안에 쿼리가 들어가져 있는 것을 서브쿼리(subquery)라고 한다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

**3번째 풀이 방법) 인라인뷰(inline view) 사용**

**인라인뷰(inline view) ? FROM 절 안에 있는 서브쿼리를 인라인뷰 라고 한다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

**13. 오라클 날짜에서 년도 얻어오는 방법!**

**TO\_CHAR() 와 EXTRACT()**

**차이점은 ? TO\_CHAR은 문자열을 반환, EXTRACT() 숫자형을 반환**

TO\_CHAR(날짜형) 함수는 날짜형 인자값(매개변수)을 받아 내가 원하는 값(년도, 월, 일, 요일 등)을 문자열(VARCHAR2)로 반환하는 함수

오라클 날짜형 자료형 : DATE, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE

【형식】

TO\_CHAR( date [,'fmt' [,'nlsparam']])

nlsparam

날짜 출력 형식의 종류 : Y, YYY

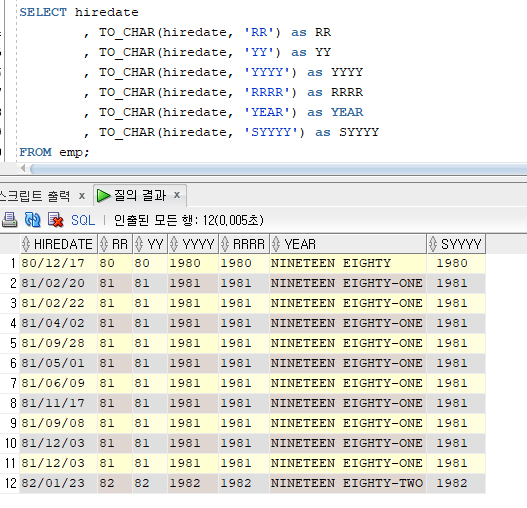
**[YY와 RR의 차이점] - Ora\_Help에서 to\_char 검색 후 읽어보기**

RR과 YY는 둘다 년도의 마지막 두자리를 출력해 주지만, 현재 system상의 세기와 나타내고자 하는 년도의 세기를 비교했을 때 출력되는 값이 다르다.

RR은 시스템상(1900년대)의 년도를 기준으로 하여 이전 50년도에서 이후 49년까지는 기준년도와 가까운 1850년도에서 1949년도까지의 값으로 표현하고,

이 범위를 벗아날 경우 다시 2100년을 기준으로 이전 50년도에서 이후 49년까지의 값을 출력한다.

YY는 무조건 system상의 년도를 따른다.



테이블이(가) 표시된 사진

자동 생성된 설명

**내가 풀어본 것..**

텍스트이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

아래 두가지로도 38번 문제를 풀 수도 있음 나중에 다시 다뤄볼 예정~

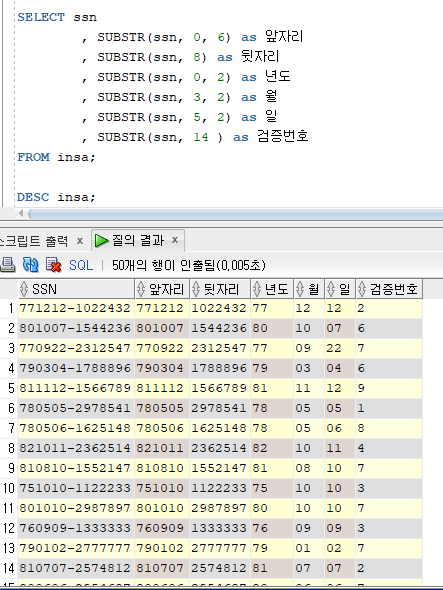
LIKE SQL 연산자

REGEXP\_LIKE() 함수

**14. 문제) insa 테이블에서 주민등록번호로 부터**

**앞자리 6자리 출력, 뒷자리 7자리 출력, 년도 2자리 출력, 월 2자리, 일 2자리 출력,**

**주민번호 마지막 검증 1자리 출력**



SUBSTR() 두번째 값을 마이너스로 주면 뒤에서부터 가져올 수 있음

>**771212-1\*\*\*\*\*\* 이런 형식으로 출력하기**

SELECT ssn

, SUBSTR(ssn, 0, 8) || '\*\*\*\*\*\*' as RRN

, CONCAT(SUBSTR(ssn, 0, 8), '\*\*\*\*\*\*') as RRN

FROM insa;

**15. 39번 문제 emp 테이블에서 직속상사(mgr)가 없는 사원의 정보를 조회**

SELECT empno, ename, mgr

FROM emp

WHERE mgr IS NULL;

**16. SQL 연산자를 적으세요.**

ㄱ. [NOT] IN()

ㄴ. [NOT] BETWEEN a AND B

ㄷ. IS [NOT] NULL

ANY, SOME, ALL -> WHERE 조건절의 서브쿼리를 사용할 때 쓰이는 SQL 연산자이다.

WHERE SOME, ANY, ALL + 서브쿼리

**수업시작!**

1. **LIKE 연산자와 REGEXP\_LIKE 연산자**

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**HR 계정**

텍스트이(가) 표시된 사진

자동 생성된 설명