

## JSP 3-3

### JSP 3-2

## 데이터베이스 & 오라클 개요

### 데이터베이스란?

체계적으로 구성된 데이터의 집합

- 정보 저장, 관리, 검색, 수정 등을 효율적으로 수행
- 일반적으로 **구조화된 방식**(예: 표 형태)으로 데이터를 저장
- 대규모 데이터 처리에 유리하며 일관성과 정확성 유지

## DBMS (Database Management System)

DB를 조작 및 관리하는 소프트웨어

- 응용 프로그램은 DBMS를 통해 DB에 접근
- 데이터 중복 방지, 통합된 데이터 접근 가능
- 응용 로직과 데이터 관리가 **분리**되어 효율성 증가

응용 프로그램  $\longleftrightarrow$  DBMS  $\longleftrightarrow$  Database

## 관계형 데이터베이스 (RDB) & SQL

### RDBMS

- 관계(Relational) 기반으로 구성된 데이터베이스 관리 시스템

### SQL (Structured Query Language)

- 관계형 데이터베이스에서 **데이터 조작/관리용 언어**

### SQL 문 종류

## 1 DQL (Data Query Language)

- `SELECT` : 데이터 검색

## 2 DML (Data Manipulation Language)

- `INSERT` , `UPDATE` , `DELETE`
- 트랜잭션 대상 → `COMMIT` / `ROLLBACK` 가능

## 3 DDL (Data Definition Language)

- `CREATE` , `ALTER` , `DROP` , `TRUNCATE` , `RENAME`
- DB 객체 구조 정의

## 4 DCL (Data Control Language)

- `GRANT` , `REVOKE` : 사용자 권한 제어

## 5 TCL (Transaction Control Language)

- `COMMIT` , `ROLLBACK` , `SAVEPOINT`
- 트랜잭션 관리



## 오라클 관리자 계정 및 권한

계정/권한	설명
SYS	DB 최고 관리자 (모든 권한)
SYSTEM	백업/복구 제외 모든 관리 가능
SYSOPER	SYSTEM이 가진 운영 권한
SYSDBA	DB 생성/삭제 포함한 최상위 권한

---

## 관계형 DB 구성요소

- **스키마**: DB 구조
- **테이블**: 데이터 저장 단위
- **컬럼**: 테이블 내 속성
- **레코드**: 행(row), 데이터 한 건

2차원 표 형태로 저장 장소  
관리

Table: 데이터 저장 공간

name	id	....	mail
~	~	...	~

행 row : 저장하려는 하나씩 개체를 구성하는  
여러 값은 가로로 늘어뜨린 형태

열 column : 저장하려는 데이터를 대표하는 이름과 특성 정의  
저장정보의 구조형, 중복 허용 여부 등의 조건·범위 지정

## ❖ 롤(Role)

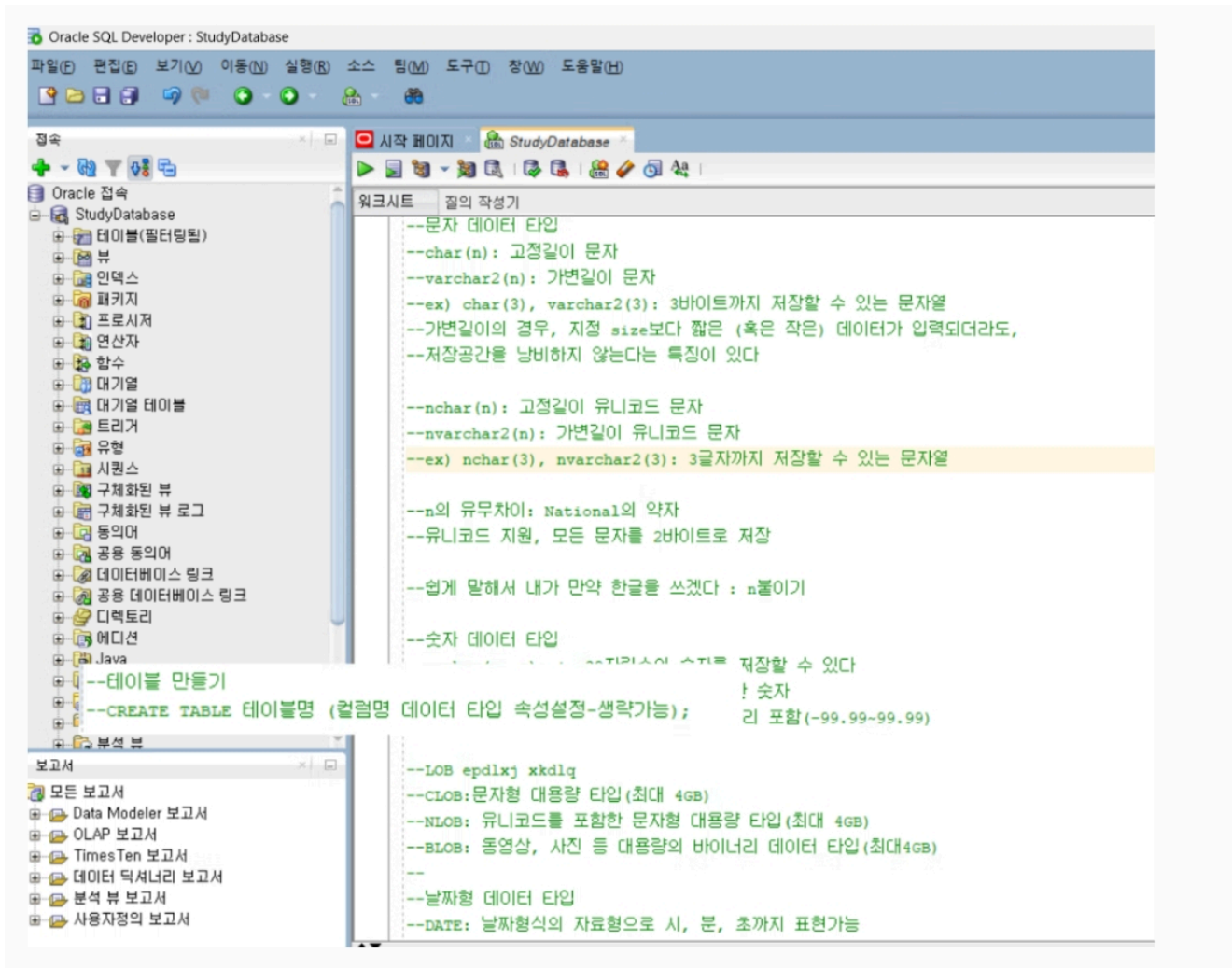
여러 사용자 권한을 그룹화하여 효율적으로 관리하는 개념

롤	설명
CONNECT	DB 접속 및 객체 생성
RESOURCE	테이블/시퀀스 생성 권한
DBA	전체 권한, 다른 사용자에게 권한 부여 가능

## abc 오라클 데이터 타입

타입	설명
CHAR(n)	고정 길이 문자 (최대 2000B)
VARCHAR2(n)	가변 길이 문자 (최대 4000B)
NCHAR(n)	유니코드 고정 문자 (다국어 지원, 최대 2000B)
NVARCHAR2(n)	유니코드 가변 문자 (최대 4000B)
NUMBER(p, s)	정수/소수 표현 (p: 전체자리, s: 소수점 자리)

## 📷 시각 자료



--SQL: 데이터베이스 질의언어

--SELECT: 검색

--WHERE: 조건

--\*: 모든 것

--INSERT: 저장

--UPDATE: 수정

--DELETE: 삭제

CREATE TABLE MEMBER (

테이블명

ID VARCHAR2 (30) PRIMARY KEY,  
PW VARCHAR2 (30) NOT NULL,  
NAME VARCHAR2 (30) NOT NULL,  
AGE VARCHAR2 (3) NOT NULL,  
EMAIL VARCHAR (40)

); 컬럼명

→ 자료형 + 조건  
생략 가능

--데이터 추가하기

--INSERT INTO 테이블명 (컬럼명1,,,,,) VALUES (데이터1, ,,,,,);

INSERT INTO MEMBER (ID, PW, NAME, AGE, EMAIL)  
VALUES('jon1024', 'jon1024', 'test', '20', 'test@naver.com');

INSERT INTO MEMBER (ID, PW, NAME, AGE, EMAIL)  
VALUES('hong', '1234', 'test', '20', 'hong@naver.com');

--데이터 수정하기

--UPDATE 테이블명 SET 수정할 컬럼명 = 변경할 값 WHERE 조건

UPDATE MEMBER SET NAME='홍길동' WHERE PW='1234';

SELECT ID, PW, NAME  
FROM MEMBER;

UPDATE MEMBER SET PW='7777' WHERE NAME='홍길동';

SELECT \* FROM MEMBER;

--데이터 삭제하기

--DELETE FROM 테이블명 WHERE 조건

DELETE FROM MEMBER WHERE NAME='test';  
commit;

-- 테이블 객체를 삭제: drop

-- DROP 테이블명;

DROP TABLE MEMBER;

SELECT \* FROM TAB;

--조건에 맞게 조회하기

--SELECT 컬럼명 FROM 테이블명 WHERE 조건식  
commit;

SELECT ID, PW, NAME  
FROM MEMBER  
WHERE NAME='test';

```

--ORDER BY: 정렬
--ASC: 오름차순
--DESC: 내림차순

--SELECT * FROM MEMBER ORDER BY ID ASC; : 아이디를 기준으로 오름차순 정렬
--SELECT * FROM MEMBER ORDER BY ID DESC; : 아이디를 기준으로 내림차순 정렬

SELECT * FROM MEMBER ORDER BY ID ASC;
SELECT * FROM MEMBER ORDER BY ID DESC;

```

```

--산술연산자
SELECT NAME, NAME || 'PLUS' AS "NAME+STR" FROM MEMBER;
--문자열 결합은 ||를 사용
--숫자는 +, -, *, 등등 사용 가능

--비교연산자
SELECT * FROM MEMBER WHERE NAME='A'; --같다
SELECT * FROM MEMBER WHERE NAME != 'A'; --같지않다 ^=도 같은 표현
SELECT * FROM MEMBER WHERE NAME > 'A'; --크다
--이외에도 우리가 알고 있는 비교 연산자 적용 가능

```

```

--논리 연산자
--AND: 두 가지 조건을 모두 만족
SELECT * FROM MEMBER WHERE NAME='A' AND ID='1111';
-- OR: 두 가지 조건 중 하나만 만족해도 가능
SELECT * FROM MEMBER WHERE NAME='A' OR ID='2222';
--NOT: 조건을 만족하지 않아야한다
SELECT * FROM MEMBER WHERE NOT NAME='A';

```