

-SQL(Structured Query Language) : database를 활용하는 언어, 비 절차적 언어

Data Base(SQL)

※ORACLE의 Database 18c Ex.Edi 설치 및 SQL 설정

- 1.sql 실행
- 2.사용자 생성
-CREATE USER (아이디)
-IDENTIFIED BY (비번);
- 3.system에서 dba를 내 아이디에 권한부여
-grant dba to (아이디);
- 4.확인
-show user
5.
-connect (아이디)

SQL 기본 단어

```
select * from dba_user; // 유저 정보 확인  
select * from product_component_version; // DB정보확인  
conn system // cd system이랑 같은 맥락
```

SQL Plus

SQL*Plus: Release 18.0.0.0.0 - Production on 화 5월 26 13:07:59 2020
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

사용자명 입력: system

비밀번호 입력:

마지막 성공한 로그인 시간: 화 5월 26 2020 13:04:23 +09:00

다음에 접속됨:

Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> show user

USER은 "SYSTEM"입니다

SQL> create user c##ora_user identified by yeo;

사용자가 생성되었습니다.

SQL> grant dba to c##ora_user;

권한이 부여되었습니다.

SQL> show user

USER은 "SYSTEM"입니다

SQL> connect c##ora_user

비밀번호 입력:

연결되었습니다.

※JAVA SE 11 (LTS(보장됨)) 설치 및 설정

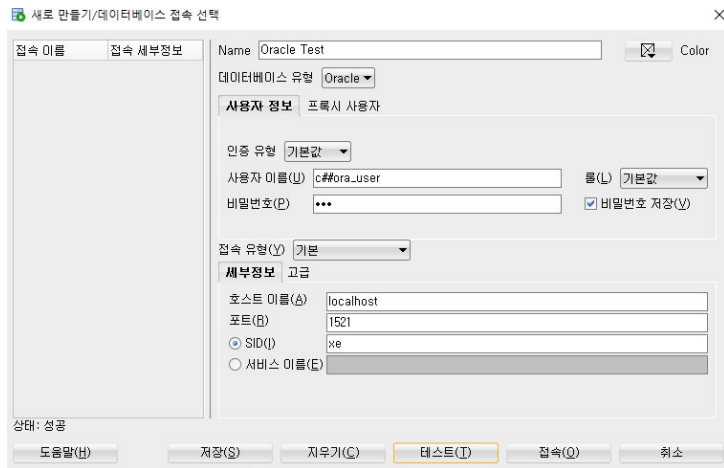
1. 환경설정

-고급 시스템 설정 환경변수

-시스템 변수->path->새로만들기->변수이름 JAVA_HOME->변수값에 디렉토리 찾아보기 클릭 (ex : C:\Program Files\Java\jdk-11.0.7) -> PATH 편집 -> %JAVA_HOME%\bin 추가 -> cmd 재부팅 후 java -version으로 확인

※SQL Developer 설치 및 설정

1. 바탕화면에 압출폴기
2. Developer 접속
3. 왼쪽상단 +
4. 테스트 후 저장



※SQL 접속법

```
C:\Users\kkitri>sqlplus
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on 화 5월 26 13:50:10 2020  
Version 18.4.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
사용자명 입력: c##ora_user
```

```
비밀번호 입력:
```

```
마지막 성공한 로그인 시간: 화 5월 26 2020 13:49:24 +09:00
```

```
다음에 접속됨:
```

```
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production  
Version 18.4.0.0.0
```

```
SQL> show user
```

```
USER은 "C##ORA_USER"입니다
```

```
SQL>
```

※오라클 sql developer 실행법

1.table 6개 생성

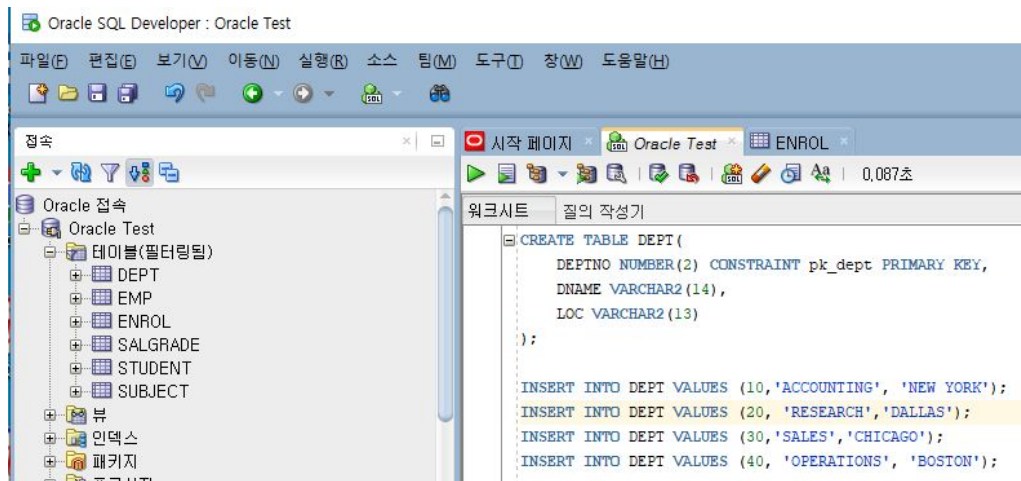
```
CREATE TABLE DEPT(  
    DEPTNO NUMBER(2) CONSTRAINT pk_dept PRIMARY KEY,  
    DNAME VARCHAR2(14),  
    LOC VARCHAR2(13)  
);
```

2.insert 값 입력

```
INSERT INTO DEPT VALUES (10,'ACCOUNTING', 'NEW YORK');  
INSERT INTO DEPT VALUES (20, 'RESEARCH','DALLAS');  
INSERT INTO DEPT VALUES (30,'SALES','CHICAGO');  
INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');
```

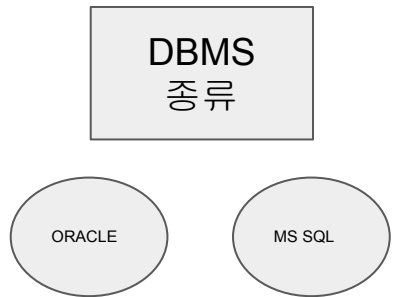
3.끝난 후에는 DB 저장을 위해 꼭

Commit; 입력(**COMMIT**을 해야 DB의메모리 -> 하드디스크)

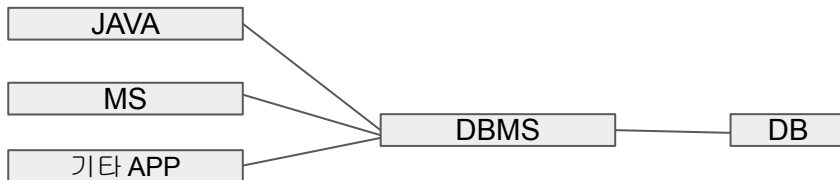


※sql++ 실행법

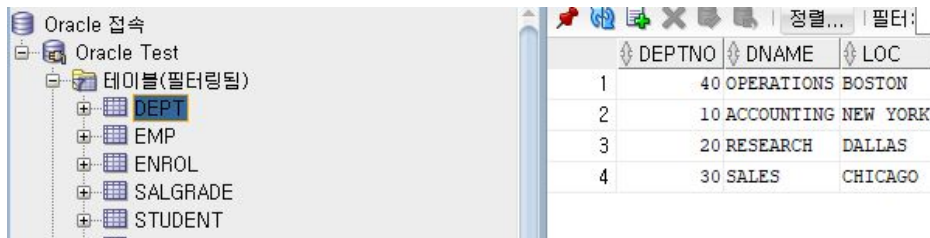
```
select * from tab; // 생성한 tab 모두 보기  
select * from dept; // dept table 보기
```



※RDBMS? 테이블 간 속성정보 교류 가능
 ※SQL란? Table에서 내가 원하는걸 뽑는 코딩 작업이 SQL
 ※DBMS란?



※SQL Developer 기본 개념



DEPT (기준 테이블)	DEPTNO(열)	DNAME	LOC
1			
2			
3			
4			

용어 설명

5 select [열 이름] // 열에 있는
것을 출력 및 함수

1 from [테이블 이름] //
테이블로부터

2 where[조건문]

3 group by

4 having

6 order by

예제 1

```
SELECT ENAME, EMPNO  
FROM EMP;
```

예제 2

```
SELECT ENAME, SAL  
FROM EMP  
WHERE EMPNO = 7499;
```

예제 3

1학년 학생들의 이름, 학번을
출력

예제 4

20153075 학번을 가진 학생의
이름, 키 몸무게, 성별을
출력하시오

예제 5

기계과 학생들의 학번, 이름,
학과명을 출력하시오

예제 8

유가인씨는 기계이며, 키는
154이다를 출력

예제 9

전체 학과를 출력(중복되는
학과명은 한개로만 출력되게)

distinct 사용

문법

WHERE stu_no **LIKE** '____20%';

WHERE stu_grade **IN** (1,2,3);

WHERE sal **NOT BETWEEN** '1000' **AND** '2900';

WHERE comm **IS NOT** null // Value가 없기에 등호 사용 불가

WHERE deptno **NOT IN** (10,20);

ORDER BY SAL **DESC**; // 내림차순

ORDER BY SAL ; // 오름차순

ORDER BY JOB, SAL; // JOB 오름차순, JOB의 SAL별로 오름차순

/ 함수(숫자형) 반올림, 올림, 내림, 나머지, 절대값*/*

SELECT ABS(-10) **FROM** DUAL; */*껍데기*/*

SELECT ROUND(stu_height, -1) */*소수점 기준으로 왼쪽 반올림 -1...-2...-3*/*
FROM STUDENT;

SELECT SIGN(-100) **FROM** DUAL; */*음수면 -1, 양수면+1*/*

SELECT MOD(10,2) **FROM** DUAL; */*나머지 값*/*

SELECT FLOOR(10.6) **FROM** DUAL; */*내림*/*

SELECT CEIL(10.6) **FROM** DUAL; */*올림*/*

SELECT TRUNC(10.27 ,1) **FROM** DUAL; */*소수점 기준 오른쪽 첫번째까지 나눔*/*

/ 함수(숫자형) 반올림, 올림, 내림, 나머지, 절대값*/*
SELECT ABS(-10) **FROM** DUAL; */*껍데기*/*

SELECT ROUND(stu_height, -1) */*소수점 기준으로 왼쪽 반올림 -1...-2...-3*/*
FROM STUDENT;

SELECT SIGN(-100) **FROM** DUAL; */*음수면 -1, 양수면+1*/*

SELECT MOD(10,2) **FROM** DUAL; */*나머지 값*/*

SELECT FLOOR(10.6) **FROM** DUAL; */*내림*/*

SELECT CEIL(10.6) **FROM** DUAL; */*올림*/*

SELECT TRUNC(10.27 ,1) **FROM** DUAL; */*소수점 기준 오른쪽 첫번째까지 나눔*/*

/ 함수(문자형) */*

SELECT LOWER('KOREA') **FROM** DUAL;

SELECT UPPER('korea') **FROM** DUAL;

SELECT INITCAP('korea') **FROM** DUAL; */*초기문자 대문자 변경*/*

SELECT SUBSTR('ABCDEFGF', 0, 3) **FROM** DUAL; */*0번째부터 3개 문자열 출력*/*

SELECT SUBSTR('998877-1234567', 0, 7) **FROM** DUAL; */*0이상 7미만 자리까지만 잘려옴*/*

SELECT RPAD('998877-', 14, '**') **FROM** DUAL; */*14자리까지 늘리고, 오른쪽에 *붙이기*/*

SELECT RPAD(SUBSTR('998877-1234567', 0, 8), 14, '**') **FROM** DUAL;

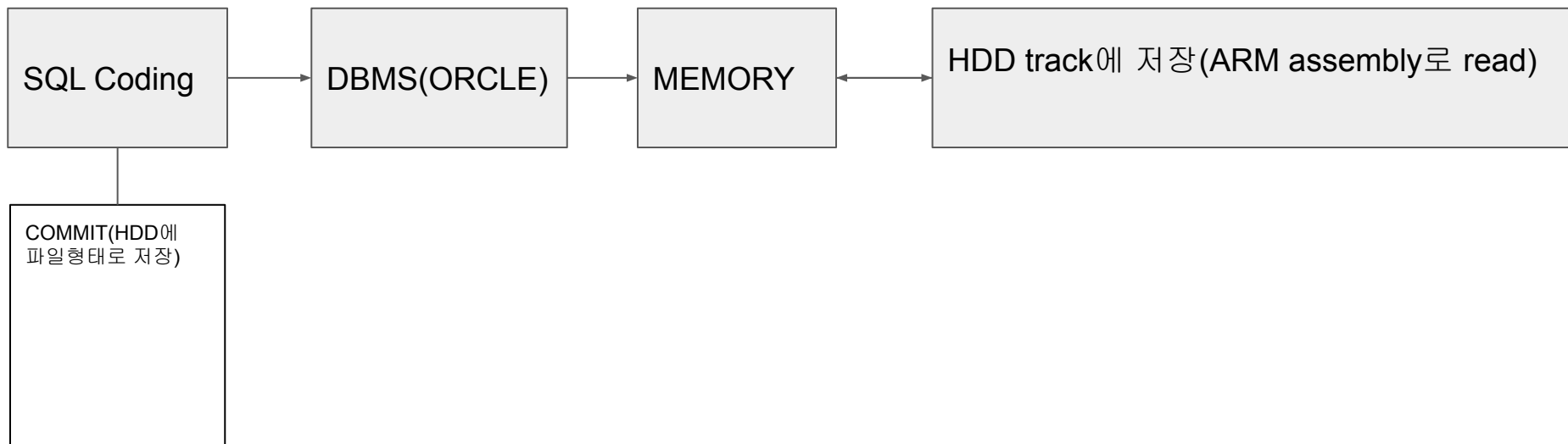
SELECT ename, TO_CHAR(hiredate, 'YY-MM-DD') */*형변환, 내가 원하는 데이터형을 만들기 위해*/*
FROM EMP;

/ NULL 활용 */*

SELECT ename, sal+comm AS 총연봉 **FROM** EMP; */* NULL은 계산 안됨 */*

SELECT ename, sal, comm, NVL2(COMM, COMM+200, 200) AS 연봉200씩추가, SAL+NVL2(COMM, COMM+200, 200) AS 추가된총연봉 */*NVL(A,B): A가 NULL이면 B로 변환 / NVL2(A,B,C): A가 NULL 아니면 B NULL이면 C*/*
FROM EMP;

DB



1. SCOTT는 입사 후, 10년 뒤 퇴직하려고 한다. 퇴직 날짜는?
2. 유가인의 주민번호는 933024-2143978 이다. 은행에서 신원확인을 위해 주민번호 앞자리를 확인하려고한다. 뒷자리를 안보이게 하는 방법은?
3. WHERE hiredate<'82/12/09' AND hiredate>'81/04/02' 다음 코드는 문자형 VS 날짜형 이다. 해당 코드가 먹힐까? (Y) ---- 이런 것이 가능하기에 몇 년을 앞서간 Tool이 SmartMaker
4. 모든 사람의 연봉을 500씩 추가해라, 현재 연봉과 총 연봉은 얼마?

※ SQL 의 DB 內 TABEL 생성 및 UPDATE

1. 생성

CREATE TABLE 생성할테이블명

(AS

SELECT *

FROM 참조테이블명

WHERE 조건);

2. 값 추가

INSERT INTO 테이블명 (생성할 열)

VALUES (열에 들어갈 값);

3. 값 업데이트

UPDATE 테이블명

SET 변수 =업데이트할 값

WHERE 조건;

※ SQL 의 DB 內 TABEL 생성 및 UPDATE

4. 제거

DELETE

FROM 테이블명

WHERE 조건;

5.컬럼 추가

ALTER TABLE 테이블명

ADD 컬럼명 타입;

6.컬럼내 데이터 변경

ALTER TABLE 테이블명

MODIFY 컬럼명 타입;

7.컬럼 삭제

ALTER TABLE 테이블명

DROP COLUMN 컬럼명

※ SQL 의 DB 內 TABEL 생성 및 UPDATE

8.테이블 데이터 삭제

DELETE FROM 테이블명

9.테이블 삭제

DROP TABLE 테이블명

10. 단순 이미지 생성

CREATE OR REPLACE VIEW 이미지명

AS

SELECT ~ ;

※ 시퀀스 제작

초기값1, 증감값 2,

최대값 20, CYCLE이 가능한

시퀀스를 만드시오.

```
CREATE SEQUENCE SEQ_NUM
```

```
START WITH 1
```

```
INCREMENT BY 2
```

```
MAXVALUE 20
```

```
CYCLE
```

```
NOCHACHE;
```

-SQL(Structured Query Language) : database를 활용하는 언어, 비 절차적 언어

JAVA

객체지향 언어(JAVA)

- 파이썬,C 등은 함축적이라 좋고, JAVA는 코딩생산성은 낮지만 여러명이 COWORK 하기엔 좋음
- JAVA는 어떤 OS에서도 사용 가능

JAVA APP(코딩)

JVM(Binary code로 바꿔줌)

OS(전기적 신호로 바꿔줌) (TR 같은것)

HW(PC)

-public static//메모리에 곧바로 올려줌 void//타입상관안함 main(String[] args//인풋) {//아웃풋} -- 컴퓨터는 main함수 부터 읽기에, 해당 코딩은 필수적

```
public static void main(String[] args){}
```

※사칙연산

```
int a = 2
```

```
System.out.println(++a + + a)
```

결과 : 5

//a에 1을 먼저 더하고 계산

```
int a = 2
```

```
System.out.println(a++ + a)
```

결과 : 3

//계산 다하고 a에 1을더함

※JAVA 활용 (조건문

API : 남의 함수 끌어다 쓰기(java api doc.에서 찾기)

-Math.random() // 0이상 1미만 랜덤값 호출

-For 문 활용

```
For(int i = 1; i <= 10; i++) {  
    For(int j = 0; j < i; j++) {  
        System.out.println("*");  
    }  
}
```

출력물

*

**

-Input 사용(java api doc. 참조)

```
import java.util.Scanner;
```

```
[class]
```

```
[void main]
```

```
Scanner sc= new Scanner(System.in);
```

*진수 변환 예

```
String st = sc.next();// hex로 변환하기 위한 문자열 입력  
int hex = Integer.parseInt(st, 16); // String을 16진수로 변환
```

※ 369 Game

Using For, If

```
for(int i = 1; i <= 40; i++) {  
  
    if( (i%10)%3 == 0 && i%10 != 0) {  
        System.out.print("짝"+" ");  
    }  
    else {  
        System.out.print(i + " ");  
    }  
    if(i/30 == 1 && i%3 == 0 && i != 30) {  
        System.out.print("짝" + " ");  
    }  
  
}
```

※ 구구단 출력(ver. 대각선 삭제)

Using For, If

```
for(int i=1; i<10; i++) {  
    for(int j=2; j<10; j++) {  
  
        if(j-1 == i) {  
            System.out.print("    ");  
        }  
        else {  
            System.out.print(j + " * " + i + " = " + i*j + " ");  
        }  
    }  
    System.out.println("");  
}
```


※ 가위바위보 Game

Using **Math.random()**,
Double If

```
int server = (int) (Math.random()*3); //0에서 2까지 값 랜덤
int client = (int) (Math.random()*3);

System.out.println("0은 가위, 1은 바위, 2는 보 이다.");
System.out.println("컴퓨터 : " + server);
System.out.println("나 : " + client);

if(server == 0) {
    if(client == 1) {
        System.out.println("내가 이겼다");
    }
    else if(client == 2){
        System.out.println("컴퓨터가 이겼다.");
    }
    else {
        System.out.println("무승부");
    }
}

else if(server == 1) {
    if(client == 0) {
        System.out.println("컴퓨터가 이겼다");
    }
    else if(client == 2) {
        System.out.println("내가 이겼다");
    }
    else {
        System.out.println("무승부");
    }
}

else {
    if(client == 0) {
        System.out.println("내가 이겼다");
    }
    else if(client == 1) {
        System.out.println("컴퓨터가 이겼다");
    }
    else {
        System.out.println("무승부");
    }
}
```

Stack

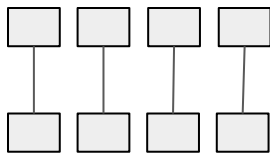
- 배열(array)
- 연결리스트(Linked List)

1. 일차원 배열

```
int arr[] = new int[7] // 7개의 주소생성 _ _ _ _ _  
for(i=0;i<7;i++){  
    arr[i] = i; // 값 배정 1 2 3 4 5 6  
}  
for(i=0;i<7;i++){  
    System.out.println(arr[i]);  
}
```

2. 이차원 배열(영상처리, 퍼즐, 머신러닝 등에 사용)

```
int [][] arr = new int[4][5]
```



arr[0] = 주소값

arr[0][0] = 값

DFS(dept first search)

-후위탐색, 중위탐색 등의 **Tree**에 해당하는 것으로, 모든 값을 다 찾는 데에 좋은 방법. 끝까지 다 찾고 **index**표시후 되돌아가 가까운 노드를 탐색함.

재귀함수

-스택에 계속 쌓이고 원하는 값이 나오면 **return**. 그 예로 **factorial**이 있음

OOP(object oriented programming)

문제를 여러개의 객체단위로 나누어 작업하는 방식
특징-클래스로 함수와 변수를 묶고, 객체를 생성하여 사용. 프로젝트를 객체단위로 나눠서 작업할 수 있기에 협업과 유지보수에 뛰어남. 절차지향은 순차적으로 처리하지만, 객체지향은 객체단위로 나누어 처리.(추상화,캡슐화,상속화,다형성)
But, 절차지향에 비해 다루기 어려움. 개발자의 능력이 중요.

개념정리

-**overflow**:메모리 가득찬 상태에서 데이터를 삽입하면 일어나는 것

-**DB**:미들웨어(**OS**와 응용프로그램의 중간역할하는 **SW**) 종류 중 하나로,
데이터들의 모임

-**DBMS**:**user**와**db** 사이에서 **user**의 요구사항에 따라
정보생성,**db**관리하는 **SW**

-**Compile**:**txt**형태 **src**를 기계어로 번역

-**transaction**:**DB**의 상태를 변환 시키기 위해, 논리기능을 수행하는
최소단위