21.05.18

오라클 DB설치하는 방법

<https://www.oracle.com/database/technologies/xe-downloads.html>에서 윈도우 버전 설치

🡪설치한 후에 cmd-sqlplus-id : system, pw : 1234입력했을 때 떠야 정상

(계정 비밀번호를 잃어 버렸을 땐 🡪 <https://m.blog.naver.com/baekmg1988/221298569322)-->>내용 상에서 sys를 system으로 바꿔서!

**alter session set container = XEPDB1;  
(만질 필요가 없음)  
-18c에만 있는 세션개념(예전 버전에는 없음)  
-세션 당 계정을 만들어야 한다.  
-세션 안에 계정들이 존재하는 거임**

**# create user java1(id) identified by 1234(pw);  
-java1이 아이디, 비밀번호 1234로 DB 사용자 계정 생성**

#grant create session to java1;

#grant connect to java1;

#grant resource to java1;

🡪java1 계정에 필요한 권한들을 부여한다

\*show user : 현재 로그인한 계정을 보여줌

\*quit : sqlplus 나가기

#connect java1**(id)**/1234**(pw)**@localhost:1521**(포트번호)**/XEPDB1; : 만든 계정으로 접속할 수 있게 함  
connect java1/1234@localhost:1521/XEPDB1;

#alter user hr account unlock;  
-연습용 hr계정 잠금해제하기  
-오라클의 기본 연습용 계정

#exit(DB를 종료한다)

Sqldeveloper(GUI) 다운 받기

<https://www.oracle.com/tools/downloads/sqldev-downloads.html>에서 윈도우 버전으로 다운

**\*SQL로 Git 사용하기!!**

<https://git-scm.com/>에서 다운로드 받기(그리고 쭉쭉 next 해주면 됨 설치 뒤에 마지막에 Launch Git Bash를 체크하고 finish

그 후에 <https://desktop.github.com/>에서 다운 그리고 github(xoghkscc, 9513574!As)로 로그인

File-new local repo 후에 수정 후에는 Repository에서 푸쉬를 하면 됨(기존에 있는 레포를 추가하려면 local)

21.05.20

#GUI 환경에서 오라클 DB를 편하게 다루기 위해서 SQL Developer를 설치한다

#Oracle 접속

텍스트이(가) 표시된 사진

자동 생성된 설명

#관리자 계정 로그인

(비번 1234)

**#컨트롤 + 엔터 : 실행**

\*계정 비밀번호를 바꾸거나 잃어버렸을 때는 관리자 계정으로 들어가서 (alter user 계정 identified by 비밀번호;)을 치고 컨트롤+엔터를 누르면 됨

(sql 여러 줄 주석은 /\* \*/, 한 줄 주석 --)

\*단축키

새 행 삽입 : 쉬프트+엔터

선택된 줄 복사 : 컨트롤+쉬프트+D

자동완성 : 컨트롤 + 스페이스

자동정렬 : 컨트롤 + F7

**#데이터베이스란?**

* 데이터를 효율적으로 저장하기 위한 데이터 저장 전문 프로그램
* 파일 시스템의 많은 문제점과 한계점을 극복하기 위해 만든 프로그램  
  (보안성, 일관성, 저장효율 등…)

**# 파일 시스템의 문제점**

* 데이터 불일치가 발생할 수 있음  
  (ex : 헬스장 회원 정보.txt / 수영장 회원 정보.txt /PT회원 정보.txt 중 하나만 수정되는 경우)
* 다수 사용자를 위한 동시 제공이 불가능하다
* 중복 데이터를 필요 이상으로 많이 저장하게 될 수 있다.
* 파일 수정 도중 시스템에 문제가 발생했을 경우에 복구하는 기능이 없음

위 문제가 발생하지 않는 간단한 기능 구현에는 파일 시스템을 이용하는 것도 좋다 (ex : 환경설정)

**#DBMS(DataBase Management System)**

* 데이터베이스의 데이터를 관리하기 위해 필요한 프로그램
* 데이터베이스는 데이터를 보관하는 장소이고, 데이터 추가/수정/삭제/검색은 DBMS가 하는 작업이다.

**#RDBMS(Relational DataBase Management System)**

* 데이터간의 관계를 이용하여 데이터의 중복을 최소화 하는 **DBMS**
* 질의문(Query)으로 데이터를 관리한다., 쿼리문은 국제 표준을 따른다

**# 테이블(Table)**

* 관계형 데이터베이스는 데이터를 표 형태로 저장한다
* 필드 : 속성(attribute), 컬럼(column), 한 열(세로)에 저장될 데이터들의 특징을 저장하는 곳
* 레코드 : 튜플(tuple), 행(가로, row), 한 행에 저장되는 관련된 데이터들

**#NO-SQL**

* 관계형 데이터베이스의 문제점을 회피하기 위해 존재하는 DBMS
* 데이터의 중복을 해결하지 않는 대신 속도가 빠르다.

\*(좌측 접속에서 우클릭-워크시트 열기 누르면 새로운 코드 작성 판이 나타남)

**#SELECT 컬럼명 FROM 테이블명;**

* 데이터를 조회하는 쿼리문
* 컬럼명에 \*를 쓰는 것은 모든 컬럼을 뜻한다.
* **쿼리문도 대소문자를 구분하지 않는다.**
* **컬럼명/테이블명은 대소문자를 구분하지 않는다.**
* **데이터는 대소문자를 구분한다 (\*중요)**
* 컬럼명 as 바꿀 컬럼명이름🡪을 하면 출력될 때 바뀐 컬럼명 이름으로 출력 가능  
  **텍스트이(가) 표시된 사진

  자동 생성된 설명**

\*Select \* from **tab**(or **tabs**) : 어떤 테이블들이 있는지 확인할 수 있다.

\* \*를 적으면 모든 컬럼을 조회하고, 컬럼명을 적으면 해당 컬럼만 조회한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**# NUMBER(n), NUMBER(n,m)**

* 숫자 데이터만 저장할 수 있는 컬럼
* 숫자 하나만 적혀있으면 정수를 저장, 숫자가 두개 적혀있으면 실수를 저장
* NUMBER(10) : 10자리 정수를 저장하는 컬럼이 된다
* **NUMBER(10, 4) : 6자리 정수와 소수점 4자리를 저장하는 컬럼이 된다**

**# VARCHAR2(n)**

* 가변 길이 문자 데이터를 저장하는 컬럼 타입
* 저장되는 데이터의 크기에 따라 알맞은 공간만 사용한다
* VARCHAR2(10)으로 설정된 컬럼의 데이터로 2글자가 들어오면 2바이트만 사용한다.
* 가변적인 데이터를 저장할 때는 데이터가 절약되지만 크기 계산이 필요해 속도가 약간 느리다.

**# CHAR(n)**

* 고정 길이 문자 데이터를 저장하는 컬럼 타입
* 설정된 기링보다 적은 양의 데이터가 들어오더라도 설정된 길이를 모두 차지한다
* CHAR(10)으로 설정된 컬럼의 데이터로 2글자가 들어오면 10바이트를 모두 사용한다.
* 데이터가 낭비될 수는 있지만 불필요한 크기 계산이 없기 때문에 속도가 더 빠르다(VARCHAR2보다 더 빠르다)

**# DATE**

* 날짜 및 시간을 저장하는 컬럼 타입

\*산술 연산자를 이용해 계산된 값 출력하기  
select last\_name, salary+500 from employees;

\*산술 연산할 때 null값이 있는 경우  
🡪해당 컬럼에서 null값이 나왔을 때 대체값을 사용한다.  
select last\_name, salary \* (1 + nvl(commission\_pct, 0)) from employees;  
🡪commision\_pct에는 일부 NULL값이 존재하는데 이때 그 NULL값을 0으로 대체한다는 의미

**\*select distinct : 중복되는 내용이 여러 개 있어도 한번만 출력된다.  
🡪중복되는 직업은 한번만 출력**

**#SELECT 컬럼명 FROM 테이블명 WHERE 조건절;**

* SELECT문에 WHERE절을 추가하여 해당 조건을 만족하는 데이터만 조회할 수 있다  
  🡪월급이 15000달러인 사람들만 조회

**#오라클의 비교 연산자들**

* = : 같다
* <, >, <=, >= : 비교
* !=, <>, ^= : 다르다

\*오라클의 문자열 리터럴은 ‘’만 사용한다  
🡪성이 ‘Steven’인 사람들만 조사

\*날짜 타입도 가볍게 비교가 가능하다  
🡪입사일이 적힌 날짜보다 적힌 데이터들을 조회

\*and, or, not  
  

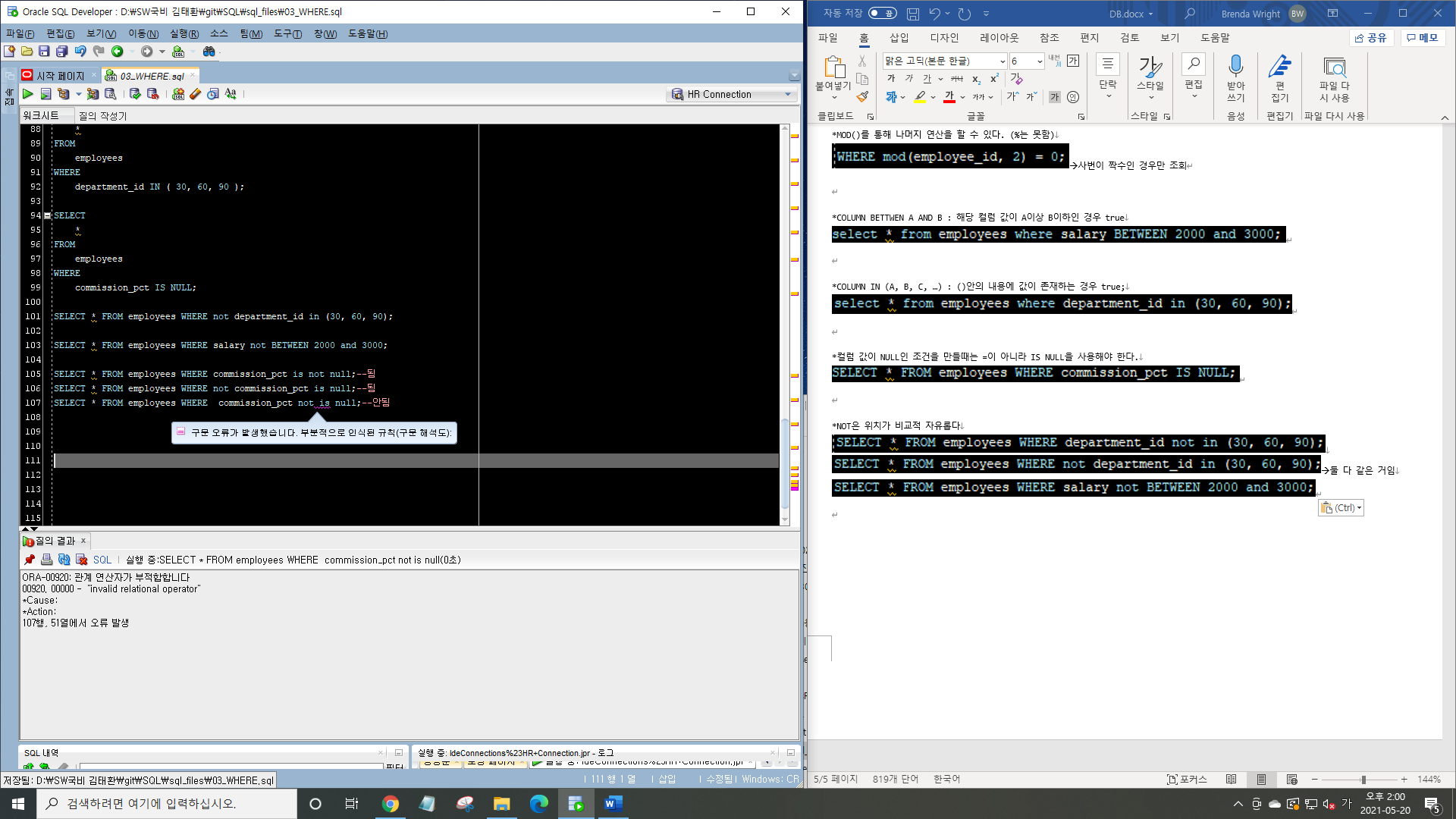

\*MOD()를 통해 나머지 연산을 할 수 있다. (%는 못함)  
🡪사번이 짝수인 경우만 조회

\*COLUMN BETTWEN A AND B : 해당 컬럼 값이 A이상 B이하인 경우 true  


\*COLUMN IN (A, B, C, …) : ()안의 내용에 값이 존재하는 경우 true;  


\*컬럼 값이 NULL인 조건을 만들때는 =이 아니라 IS NULL을 사용해야 한다.  


\*NOT은 위치가 비교적 자유롭다  
  
🡪둘 다 같은 거임  

**#LIKE**

* **데이터의 일부분으로 원하는 내용을 검색할 수 있다.**
* 문자 타입과 날짜 타입에 사용할 수 있다
* % : 길이 제한 없이 아무 문자열이나 와도 되는 와일드카드  
  (심지어 없는 것도 가능)
* \_ : 하나의 아무 문자가 반드시 와야하는 와일드카드  
     
  텍스트, 벽, 실내이(가) 표시된 사진

  자동 생성된 설명🡪마지막꺼는 유용하게 쓸 듯  
  텍스트, 화면, 검은색이(가) 표시된 사진

  자동 생성된 설명

\*UNION :합집합(중복은 알아서 제거됨)🡪단 같은 테이블끼리 해야함(단 같은 열을 지니고 있으면 다른 테이블도 가능)

텍스트, 오렌지, 실내, 화면이(가) 표시된 사진

자동 생성된 설명

\*UNION ALL : 합집합(중복 제거 안함)  
텍스트, 오렌지, 설정, 스크린샷이(가) 표시된 사진

자동 생성된 설명

\*INTERSECT : 교집합  
텍스트, 실내, 오렌지, 설정이(가) 표시된 사진

자동 생성된 설명

\*MINUM : 차집합  
텍스트, 오렌지, 화면, 설정이(가) 표시된 사진

자동 생성된 설명  
🡪즉 10~40 – 30~50이므로 10~30이라는 의미(30제외)

텍스트이(가) 표시된 사진

자동 생성된 설명

**🡪이렇게 만들면 타 컴퓨터와 연동된 서버 가능(단 기준이 되는 컴퓨터는 항상 켜놔야 접속이 가능함)**