**[Day 1209]**

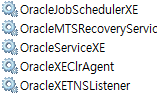
[참고 사이트]

<https://wikidocs.net/>

**Oracle DB 설치 [Oracle Database 11g Express Edition version]설치**

1. \*필수\* 설치전 java EE (html버전) 실행시켜서, tomcat server를 start하여 port 8080을 선점해둔다.
2. Oracle DB를 설치한다 [http port가 8081로 나와야 정상]

오라클 지울때는 services.msc와 registry에 잇는 oracle들 모두삭제해줘야함

이런거잇으면 삭제

1. 하기 내용 맞는지 체크후 설치진행

Destination Folder: D:\oracleXE\

Oracle Home: D:\oracleXE\app\oracle\product\11.2.0\server\

Oracle Base:D:\oracleXE\

Port for 'Oracle Database Listener': 1521

Port for 'Oracle Services for Microsoft Transaction Server': 2030

Port for 'Oracle HTTP Listener': 8081

1. RUN SQL Command Line 실행. 노란색 부분 입력

SQL\*Plus: Release 11.2.0.2.0 Production on 수 12월 9 10:53:54 2020

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> connect sys/oracle as sysdba

Connected.

SQL> alter user hr identified by hr account unlock;

User altered.

SQL> connect hr/hr

Connected.

SQL> select tname from tab;

1. [Start DB]exe 실행
2. 자동 실행 되지 않도록 아래 두개를 수동으로 변경해주기





1. Stop/start bat 파일 편집

* D:\oracleXE\app\oracle\product\11.2.0\server\bin 디렉토리내의
* [stopDB.bat] – 편집 – 아래내용 넣고

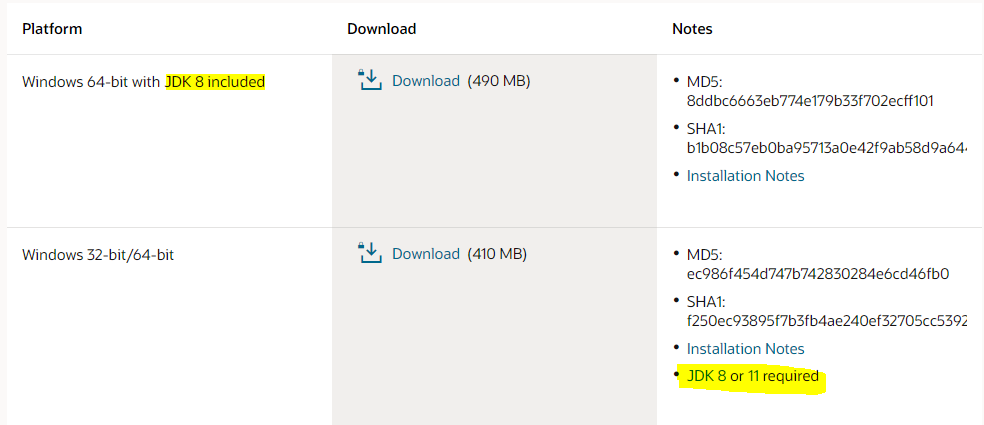
net stop OracleServiceXE

net stop OracleXETNSListener

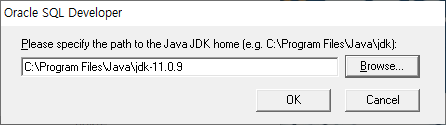
* 저장

**[SQL Developer 19.2.1 download]**

1. Oracle 홈페이지 [SQL developer download]검색
2. 둘 중에 필요한 걸로 다운로드 (이미 JDK 8이나 11 깔려있으면 아래꺼 다운로드)



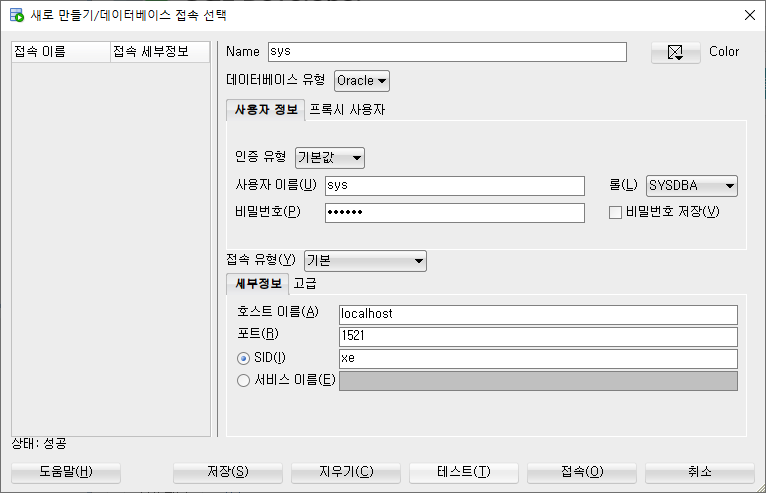
1. 경로 선택해주고 ok



**[ORACLE SQL DEVELOPER 시작하기]**

1. Start DB 실행
2. 시스템 관리자 생성하기

[접속]-[ + ] 클릭 새로만들기



1. 테스트 해보고, 성공하면, 저장. -> 접속
2. 유저 생성하기

[sys]창에 아래 입력

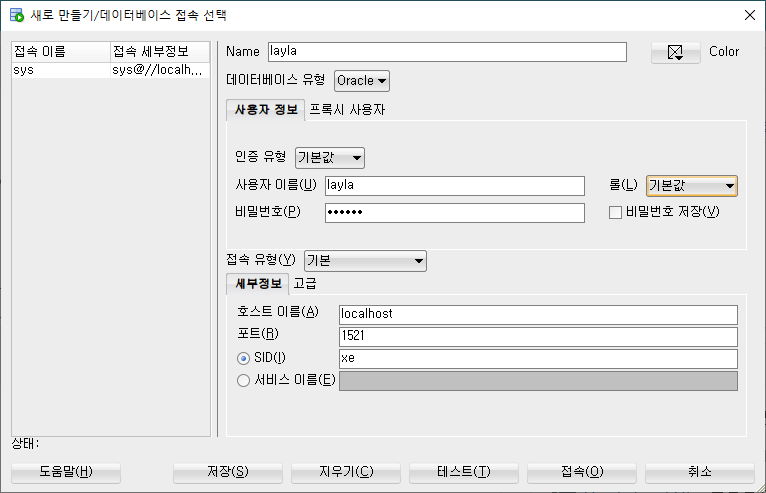
CREATE user layla IDENTIFIED by oracle;

유저 아이디 Layla, 비밀번호 oracle

GRANT connect, resource, dba to layla;

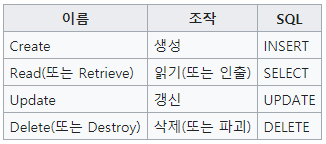
접속권한, 리소스권한, DBA권한을 layla에게 부여

1. [새로만들기]-생성한 유저로 접속



**[CRUD]**

<https://ko.wikipedia.org/wiki/CRUD>



**[DML] 중요! 문제많이나옴.**

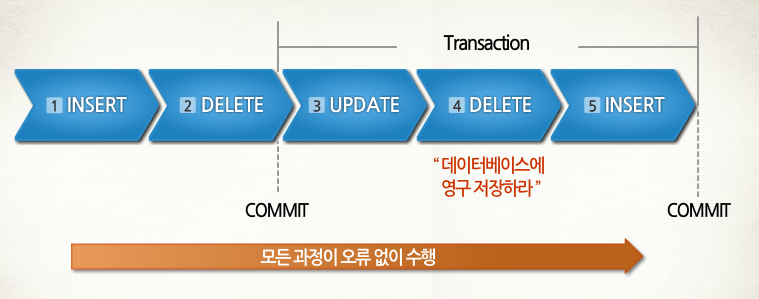
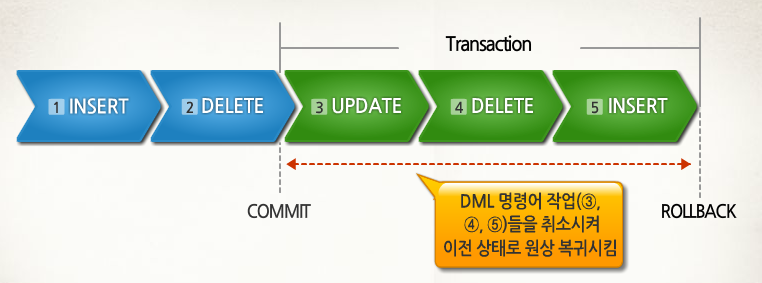
데이터 조작 언어 (Data Manipulation Language, DML)

<https://brownbears.tistory.com/180>



[COMMIT과 ROLLBACK]

<https://wikidocs.net/4096>

* **COMMIT**
  + 모든 작업을 정상적으로 처리하겠다고 확정하는 명령어이다.
  + 트랜젝션의 처리 과정을 데이터베이스에 반영하기 위해서, **변경된 내용을 모두 영구 저장한다.**
  + COMMIT 수행하면, **하나의 트랜젝션 과정을 종료하게 된다.**
  + TRANSACTION(INSERT, UPDATE, DELETE)작업 내용을 실제 DB에 저장한다.
  + 이전 데이터가 완전히 UPDATE된다.
  + 모든 사용자가 변경한 데이터의 결과를 볼 수 있다. 
* **ROLLBACK**
  + 작업 중 문제가 발생했을 때, 트랜젝션의 처리 과정에서 발생한 변경 사항을 취소하고, 트랜젝션 과정을 종료시킨다.
  + 트랜젝션으로 인한 **하나의 묶음 처리가 시작되기 이전의 상태로 되돌린다.**
  + TRANSACTION(INSERT, UPDATE, DELETE)작업 내용을 취소한다.
  + 이전 COMMIT한 곳까지만 복구한다.
* 트랜젝션 작업 중 하나라도 문제가 발생하면, 모든 작업을 취소해야 하기 때문에 하나의 논리적인 작업 단위로 구성해 놓아야 한다.
  + 문제가 발생하면, **논리적인 작업의 단위를 모두 취소**해 버리면 되기 때문이다. 

**[ORACLE DB DATATYPE총정리]**

https://coding-factory.tistory.com/416

데이터 타입이란 컬럼이 저장되는 데이터 유형을 말합니다. 오라클에서 기본적으로 제공하는 기본 데이터 타입과 이 기본데이터 타입을 혼합하여 만들 수 있는 사용자 정의 데이터 타입으로 구분할 수 있습니다. 기본 데이터 타입은 문자형, 실수, 소수, 자료형 등의 여러 데이터를 식별하는 타입입니다.

문자 데이터 타입

|  |  |
| --- | --- |
| 데이터타입 | 설명 |
| CHAR(n) | 고정길이 문자 / 최대 2000byte / 디폴트 값은 1byte |
| VARCHAR2(n) | 가변길이 문자 / 최대 4000BYTE / 디폴트 값은 1byte |
| NCHAR(n) | 고정길이 유니코드 문자(다국어 입력가능) / 최대 2000byte / 디폴트 값은 1byte |
| NVARCHAR(n) | 가변길이 유니코드 문자(다국어 입력가능) / 최대 2000byte / 디폴트 값은 1byte |
| LONG | 최대 2GB 크기의 가변길이 문자형 |
| CLOB | 대용량 텍스트 데이터 타입(최대 4Gbyte) |
| NCLOB | 대용량 텍스트 유니코드 데이터 타입(최대 4Gbyte) |

가변길이란 실제 입력된 데이터 길이에 따라서 크기가 변하는 것을 의미합니다.

ex) 예를들어 가변형 문자형 데이터타입인 Varchar2에 10이라는 데이터 공간을 할당하고 abc라는 데이터를 입력했다고 치면 그 데이터는 10이라는 데이터 공간을 할당했기에 10byte까지만 데이터를 입력받을 수 있고 3글자만 입력되었으므로 그 데이터공간은 3byte가 됩니다. 반대로 고정형 문자형 데이터타입인 CHAR에 10이라는 데이터 공간을 할당하고 마찬가지로 abc라는 데이터를 입력했다고 치면 그 데이터는 3글자만 입력되었지만 데이터 공간은 10byte가 됩니다. 남 / 여 같이 1글자만 들어올 수 있는것이 아니라면 대부분 VARCHAR2를 사용합니다.

숫자형 데이터 타입

|  |  |
| --- | --- |
| 데이터 타입 | 설명 |
| NUMBER(P,S) | 가변숫자 / P (1 ~ 38, 디폴트 : 38) / S (-84 ~ 127, 디폴트 값 : 0)  / 최대 22byte |
| FLOAT(P) | NUMBER의 하위타입 / P (1~128 .디폴트 : 128) / 이진수 기준 / 최대 22byte |
| BINARY\_FLOAT | 32비트 부동소수점 수 / 최대 4byte |
| BINARY\_DOUBLE | 64비트 부동소수점 수 / 최대 8byte |

숫자 타입은 4가지 타입이 있습니다. 대부분 NUMBER형을 사용합니다. P는 소수점을 포함한 전체 자릿수를 의미하고, S는 소수점 자릿수를 의미합니다. NUMBER는, 가변숫자이므로 P와S를 입력하지 않으면 저장 데이터의 크게에 맞게 자동으로 조절됩니다.

날짜 데이터 타입

|  |  |
| --- | --- |
| 데이터타입 | 설명 |
| DATE | BC 4712년 1월 1일부터 9999년 12월 31일, 연, 월, 일, 시, 분, 초 까지 입력 가능 |
| TIMESTAMP | 연도, 월, 일, 시, 분, 초 + 밀리초까지 입력가능 |

가장 일반적으로 쓰는 데이터타입은 DATE타입입니다.

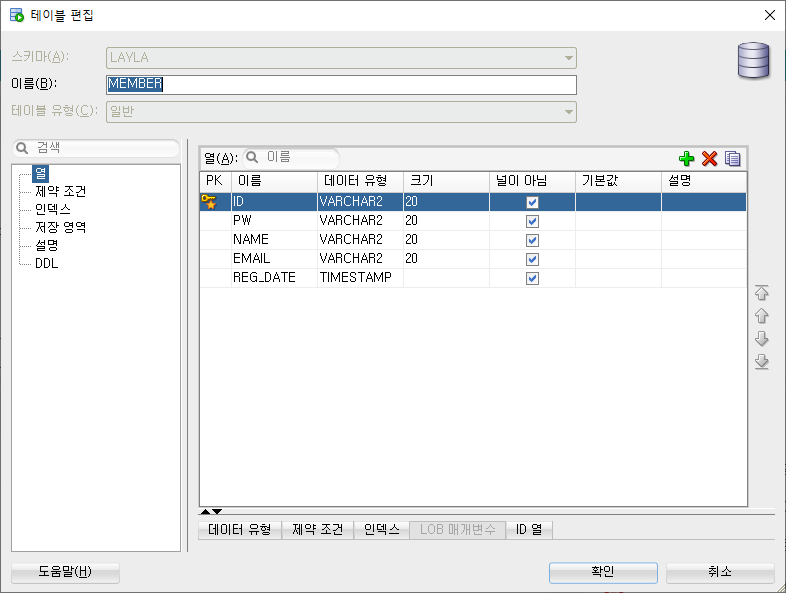
LOB 데이터 타입

|  |  |
| --- | --- |
| 데이터 타입 | 설명 |
| CLOB | 문자형 대용량 객체 고정길이와 가변길이 문자집합 지원 |
| NCLOB | 유니코드를 지원하는 문자형 대용량 객체 |
| BLOB | 이진형 대용량 객체 |
| BFILE | 대용량 이진 파일에 대한 위치,이름 저장 |

LOB이란 Large Object의 약자로 대용량 데이터를 저장할 수 있는 데이터 타입입니다. 일반적으로 그래픽, 이미지, 사운드등 비정형 데이터를 저장할때 L,OB타입을 사용합니다. 문자형 대용량 데이터는 CLOB나 NCLOB, 그래픽, 이미지, 동영상등의 대이터는 BLOB를 주로 사용합니다.

**[ORACLE SQL DEVELOPER로 테이블 만들기]**

1. 유저이름 우클릭 – 새테이블 – 내용입력하고 확인 누르면 자동으로 DB table을 생성해준다

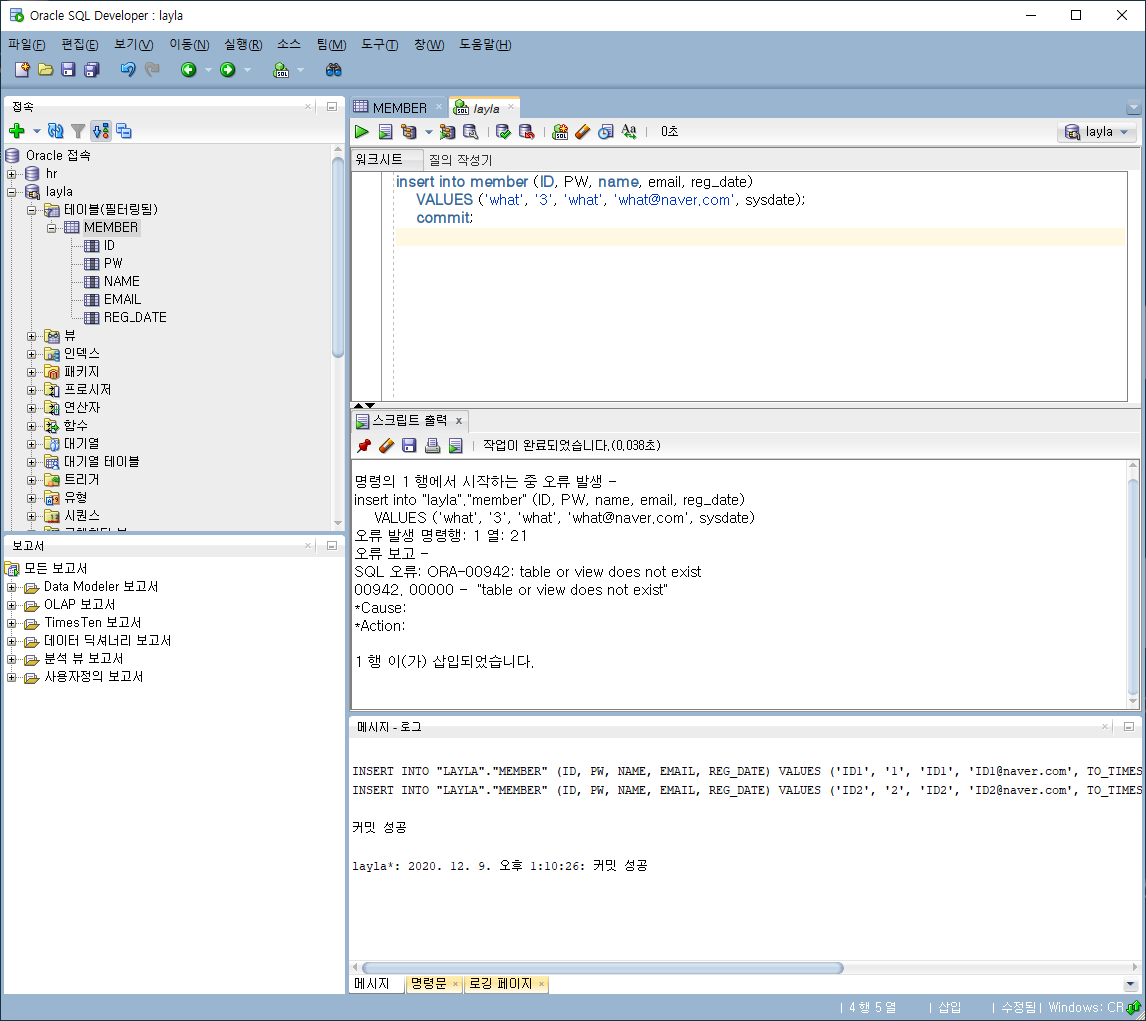


\*not null이라는건 데이터가 비면 안된다는 것

\*상기의 테이블을 dB에 직접 생성시킬 때는 아래처럼 SQL 워크시트를 열어서 직접 코드를 다 써야한다.



1. 테이블 생성후 SQL worksheet에서 임의의 유저 데이터 생성해서 넣어보기



(입력)

insert into member (ID, PW, name, email, reg\_date)

VALUES ('what', '3', 'what', 'what@naver.com', sysdate); 🡪 시스템 현재 날짜

insert into 🡪 멤버에 잇는 컬럼명을 생략하고 순서대로만 입력해줘도 됨. 단 빈칸없이.

VALUES ('what', '3', 'what', 'what@naver.com', sysdate);

insert into member (ID, PW, name, reg\_date) 🡪 멤버에 null허용되는 부분이 있으면, 빼고 입력가능.

VALUES ('what2', '3', 'what1', sysdate);

commit;

(실행)

**[SQL]**  Structured Query Language: SQL

[Day 1209]

| **특수기호** | **영문** |
| --- | --- |
| \ | Back Slash |
| @ | At Sigh |
| & | Ampersand |
| ' | Aposterophe |
| ` | Grave |
| - | Hyphen |
| < | Left Angle Braket |
| > | Right Angle Braket |
| { | Left Brace |
| } | Right Brace |
| [ | Left Braket |
| ] | Right Braket |
| | | Vertical Bar, Pipe |
| \* | Asterisk |
| " | Quotation Mark |
| ! | Exclamation Point |
| # | Crosshatch |
| . | Period or Dot |
| ~ | Tilde |
| ^ | Caret, Circumflex |

==============================================1

insert into member (ID, PW, name, email, reg\_date)

VALUES ('what', '3', 'what', 'what@naver.com', sysdate);

--일반적인 패턴 (쿼리)

insert into member

VALUES ('what1', '3', 'what1', 'what@naver.com', sysdate);

--null을 허용하는 column이 있다면 아래와 같이 해당부분 생략하고 데이터입력도 가능.

insert into member (ID, PW, name, reg\_date)

VALUES ('what2', '3', 'what1', sysdate);

--멤버테이블에서 모든 데이터를 뽑아서 보여달라는 쿼리

select \* from member;

--멤버테이블에서 열거한 데이터를 뽑아서 보여달라는 쿼리

select id, pw, name, email, reg\_date from member;

--column명 뒤에 as 를 쓰면 해당 column이름을 바꿔줌. as대신 띄워 쓰기도 가능

select id as 아이디, pw 비번, name, email, reg\_date from member;

--member의 별칭을 정해줬으면 static way로 접근해줄수도 됨. m.id / m.pw 등.

select m.id, m.pw, name, email, reg\_date from member m;

-- WHERE절 : 조건을 붙여서 select하는 기능

select \*from member WHERE id='what2';

select \*from member WHERE name='ID1';

--정보 업데이트

update member

set email='update@naver.com' WHERE id='ID1';

-- 특정 정보 삭제

DELETE FROM member WHERE id='ID1';

--확정

commit;

==============================================================

--DESC:Description

desc employees;

-- 중복제거 : distinct / 정렬: order by asc(desc) - ascending/descending/default는 asc임.

select DISTINCT salary from employees

order by salary asc;

--기존데이터 가지고 추가연산해서 열추가해보기

select department\_id, first\_name, last\_name, salary, salary\*1.03 FROM employees

order by salary desc;

--조건문/정렬 써보기

select department\_id, first\_name, last\_name, salary, salary\*1.03 FROM employees

where department\_id=30

order by salary desc;

-- 두가지 열의 데이터를 합쳐서 새로운 이름으로 저장하고 표현하기 (여기서는 중간에 한칸띄우려고' ' ||을 두번씀

select first\_name, last\_name, first\_name||''||last\_name as "name" FROM employees

where last\_name='King';

==============================================================