

DB (ORACLE)



DATABASE 란

- ❖ 영구적으로 보관해야 할 데이터를 체계적으로 저장하는 데이터 보관 솔루션
- ❖ 관계형 DBMS 의 종류
 - ORACLE / MYSQL / MSSQL / MariaDB / PostgreSQL

설치

Oracle Database 11g 릴리스 2 설치 프로그램 - 데이터베이스 설치 중 - 단계 4/8

일반 설치 구성

- 보안 갱신 구성
- 설치 옵션
- 시스템 클래스
- 기본 설치**
- 필요 조건 검사
- 요약
- 제품 설치
- 완료

기본 구성으로 전체 데이터베이스 설치를 수행하십시오.

Oracle Base(S): C:\app\Administrator

소프트웨어 위치(L): C:\app\Administrator\product11.2.0\dbhome_1

데이터베이스 파일 위치(D): C:\app\Administrator\oradata

데이터베이스 버전(E): Enterprise Edition(3.34GB)

문자 집합(C): 기본값 (KO16MSWIN949)

전역 데이터베이스 이름(G): orcl

관리 비밀번호(P): *****

비밀번호 확인(C): *****

메시지(M):
 관리 비밀번호(P):[INS-30011] 입력한 ADMIN 비밀번호는 Oracle 권장 표준을 따르지 않습니다.

기본 설정

```
관리자: C:\Windows\system32\cmd.exe
C:\Users\Administrator>sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on 일 8월 15일 2011
Copyright (c) 1982, 2010, Oracle. All rights reserved.

다음에 접속됨:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0
With the Partitioning, OLAP, Data Mining and Real Application
SQL> create user ora1 identified by ora1;

사용자가 생성되었습니다.

SQL> grant dba to ora1;

권한이 부여되었습니다.

SQL> exit
```

Sql [테이블 생성]

- ❖ 쿼리를 이용해서 테이블 생성 하도록 하자.(java 계정으로 만들자)

```
SQL> create table person(  
2  num number,  
3  name varchar2(10),  
4  birth varchar2(10),  
5  tel varchar2(20),  
6  primary key(num));
```

Table created.

- ❖ 생성한 table의 column 확인

```
SQL> desc person;  
Name          Null?         Type  
-----  
  
NUM            NOT NULL     NUMBER  
NAME                               VARCHAR2(10)  
BIRTH                               VARCHAR2(10)  
TEL                               VARCHAR2(20)
```

DB를 구성하는 자료형

- ❖ 정수 자료형 : NUMBER(소수점 표현가능. NUMBER(5), NUMBER(5,2) 등)
- ❖ 실수 자료형 : BINARY_FLOAT, BINARY_DOUBLE
- ❖ 문자 자료형
 - ❖ CHAR : 고정 길이 문자열, 공간보다 입력데이터가 짧으면 공백 처리,
 - ❖ VARCHAR2 : 가변 길이 문자열, 문자열 크기만큼 공간 확보.
 - ❖ LOB타입 : CLOB, NCLOB 등(대용량의 텍스트를 저장할 때 사용된다)
 - ❖ 날짜 자료형 : DATE 등(yy/mm/dd가 기본형태)
- ❖ **LOB(Large Object) 데이터 형은 텍스트, 이미지, 동영상, 저장하기 위한 데이터형**
- ❖ **BLOB는 Binary, 이미지, 동영상, MP3등**
- ❖ **CLOB는 e-BOOK과 같은 대용량의 텍스트, 단일 바이트 인코딩**
- ❖ **NCLOB는 대용량의 텍스트, 멀티 바이트 인코딩(한글)**

SQL 기본

❖ SQL 기본 명령은 CRUD

- Create : Insert
- Read : Select
- Update : update
- Delete : Delete,

Insert query

❖ Insert 명령은 테이블에 데이터를 삽입하는 명령

❖ 기본 문법

```
INSERT INTO table명( 컬럼1, 컬럼2...) VALUES ( 값1 , 값2...)
```

```
INSERT INTO table명 VALUES ( 값1 , 값2...)
```

```
INSERT INTO person VALUES(1000, '김변수', '900101', '01012341234');
```

```
INSERT INTO person (num, name, birth, tel)  
VALUES(1001, '이상수', '200101', '01022223333');
```


Insert query

❖ 총 5명의 데이터를 테이블에 입력

Select query

❖ Select : 데이터베이스 데이터를 조회하는 명령문

❖ 기본 문법

SELECT 컬럼명1, 컬럼명2 FROM 테이블명;

- Ex) select name from person;
- Ex) select name, birth from person;

1. SQL문은 대/소문자를 구분하지 않습니다.
2. SQL문은 한 줄 또는 여러 줄에 입력 가능합니다.
3. 대부분은 약어를 사용할 수 없습니다.
4. 가독성을 높이기 위해 들여쓰기를 권장합니다.
5. SQL문은 세미콜론(;)으로 끝나야 합니다.

Select query

- ❖ Where 절을 이용하여 원하는 값을 검색할 수 있다.
 - `Select * from person where name='김개똥';`
- ❖ num이 1002번 이상을 모두 조회 하시오.
- ❖ 테이블 조회 : `select * from tab;`
- ❖ 화면 설정 : `set linesize 300`

Delete query

- ❖ 아래 커리를 이용해 데이터를 삭제해 보자
- ❖ **delete from person** : 테이블 내의 모든 데이터가 다 삭제
 - 주의 하도록 하자.

```
SQL> delete from person;  
  
2 rows deleted.  
  
SQL> select * from person;  
  
no rows selected
```

- ❖ 등록
 - INSERT INTO person VALUES(1000, '김변수', '900101', '010-1234-1234');
 - INSERT INTO person VALUES(1001, '이상수', '910101', '010-1111-1234');
- ❖ **delete from person where name = '김개똥'** :

Update query

- ❖ update를 알아보도록 하자.
- ❖ 모든 데이터가 삭제 됐으니 다시 2-3명을 저장 후 update에 대해서 알아보도록 하자.

❖ 추가

- ❖ INSERT INTO person VALUES(1000, '김변수', '900101', '010-1234-1234');
- ❖ INSERT INTO person VALUES(1001, '이상수', '910101', '010-1111-1234');

❖ 확인

- ❖ select * from person;

Update query

❖ Where 절을 입력하지 않으면 모든 컬럼이 변경되므로 **주의**합니다.

❖ **update person set name='개똥이';** (데이터 수정)

```
SQL> update person set name='개똥이';
```

```
2 rows updated.
```

❖ **Select * from person;** (변경된 내용 확인)

```
SQL> select * from person;
```

NUM	NAME	BIRTH	TEL
1001	개똥이	111	010
1000	개똥이	111	111

❖ 모든값이 변경 되니 **주의** 하도록 하자.

Update query

- ❖ Where절을 넣어서 num 1000은 고길동 으로, 1001은 김말이 로 변경 하시오

```
SQL> update person set name='고길동' where num=1000;
```

```
1 row updated.
```

```
SQL> update person set name='김말이' where num=1001;
```

```
1 row updated.
```

- ❖ 변경후 데이터 확인

```
SQL> select * from person;
```

NUM	NAME	BIRTH	TEL
1001	김말이	111	010
1000	고길동	111	111

Commit

- ❖ 이제 commit와 rollback를 알아보고 문제를 풀도록 하자.
- ❖ COMMIT : 작업했던 내용을 확정
- ❖ 현재 작업한 내용을 commit를 진행을 해서 완전 저장 할 때 사용
- ❖ 이전에 했던 작업을 먼저 commit를 진행하자.

```
SQL> commit;
```

```
Commit complete.
```


Rollback

- ❖ Rollback : 복원
- ❖ 데이터의 수정할 때 값들을 잘못 수정을 하게 된다면, 이전으로 돌릴 수 있다.
- ❖ 현재 저장되어 있는 데이터 중에서 **고길동 -> 홍길동** 변경 후 데이터 확인
- ❖ 변경 전 내용

```
SQL> select * from person;
```

NUM	NAME	BIRTH	TEL
1001	김말이	111	010
1000	고길동	111	111

Rollback

❖ 내용 변경 하기 (고길동 -> 홍길동)

```
SQL> update person set name='홍길동' where name='고길동';
```

```
1 row updated.
```

❖ 변경 후 데이터 확인

```
SQL> select * from person;
```

NUM	NAME	BIRTH	TEL
1001	김말이	111	010
1000	홍길동	111	111

rollback

❖ Rollback를 진행

```
SQL> rollback;
```

```
Rollback complete.
```

❖ 데이터 확인

```
SQL> select * from person;
```

NUM	NAME	BIRTH	TEL
1001	김말이	111	010
1000	고길동	111	111

Commit 과 rollback

❖ 데이터 수정. SQL> update person set name='홍길동' where name='고길동';

❖ 데이터 저장 SQL> commit;

❖ 변경 내용 확인

SQL> select * from person;

NUM	NAME	BIRTH	TEL
1001	김말이	111	010
1000	홍길동	111	111

❖ Rollback 진행 SQL> rollback;

❖ 데이터 확인

SQL> select * from person;

NUM	NAME	BIRTH	TEL
1001	김말이	111	010
1000	홍길동	111	111

Commit 과 rollback

- ❖ ROLLBACK : 작업했던 내용을 취소
- ❖ COMMIT : 작업했던 내용을 확정
- ❖ ORACLE DB에서는 기본적으로 작업했던 내용을 바로 적용하지 않는다.
- ❖ 그러므로 잘 못된 작업을 진행했을 경우에는 ROLLBACK 을 통해 되돌릴 수 있다.
- ❖ 만약 모든 작업을 문제없이 완료했다면, 확정 짓는 의미에서 COMMIT 을 사용할 수 있고, 이렇게 COMMIT 된 작업은 ROLLBACK 되지 않는다.
- ❖ ROLLBACK 은 이전 commit 시점까지 되돌린다.

ALTER QUERY

- ❖ alter table 명령문은 기존 테이블의 구조를 변경하기 위한 DDL 명령문이다. 테이블에 대한 구조 변경은 컬럼의 추가, 삭제, 컬럼의 타입이나 길이를 변경할 때 사용한다. 테이블의 구조를 변경하게 되면 기존에 저장되어 있던 데이터에 영향을 주게 된다.

- ❖ ALTER TABLE로 칼럼 추가, 수정, 삭제하기 위해서는 다음과 같은 명령어 사용
 - ADD COLUMN 절을 사용하여 새로운 칼럼을 추가한다.
 - MODIFY COLUMN 절을 사용하여 기존 칼럼을 수정한다.
 - DROP COLUMN 절을 사용하여 기존 칼럼을 삭제한다.

연습용 테이블 생성 및 alter

- ❖ 기본 테이블 하나 생성하자.

create table test_table(num number);

```
create table test_table(num number);
```

- ❖ 테이블 구조 확인

desc test_table;

```
desc test_table;
```

- ❖ test_table 테이블에 컬럼 name 추가

alter table test_table add(name varchar2(20));

```
alter table test_table add(name varchar2(20));
```

- ❖ test_table 테이블에 컬럼 name 자료형 변경

alter table test_table modify(name number);

```
alter table test_table modify(name number);
```

- ❖ modify문은 테이블에 이미 존재하는 컬럼을 변경할 수 있다. 컬럼을 변경한다는 것은 컬럼에 대해서 데이터 타입이나 크기, 기본 값들을 변경한다는 의미다.

ALTER QUERY

- ❖ 테이블의 컬럼을 삭제하는 명령어다(저장데이터 있어도 해당 컬럼 및 데이터 삭제된다.) => alter table test_table drop column name;

```
alter table test_table drop column name;
```

- ❖ 테이블의 컬럼 이름을 변경하는 명령어다(저장데이터 있어도 해당 컬럼 및 데이터 삭제된다.) => alter table test_table rename column num to numB;

```
alter table test_table rename column num to numB;
```


DROP QUERY

❖ 테이블을 삭제하는 요청입니다.

- Drop table test_table;

```
drop table test_table;
```

❖ 삭제 후 확인해 보면 에러가 발생된다. 테이블이 없기 때문에

❖ 관리자 계정으로 방금 우리가 만든 계정을 삭제해 보도록 하자.

❖ 계정 삭제 (jsp가 로그인되어 있으면 삭제 되지 않는다. **닫고 삭제**)

- drop user 아이디 cascade; (테이블이 만들어 졌기 때문에 cascade 추가)

```
SQL> drop user jsp cascade;
```

```
User dropped.
```