



Oracle 1강

양 명 속

[now4ever7@gmail.com]



목차

- DBMS 란
- 오라클 설치
- 데이터 조회하기 - **select**



DBMS



데이터베이스란

■ 데이터베이스(Database, DB)

- 여러 응용시스템들이 공유할 수 있도록 통합 저장된 데이터의 집합
- 중복을 최소화하고, 다수의 사용자가 공유할 수 있도록 데이터들을 모아둔 시스템
- 데이터베이스는 혼자서 사용하는 시스템이 아니므로 여러 명의 사용자나 시스템간에 **상호 공유가 가능**해야 함
- 단순히 데이터가 모여 있는 것을 의미하는 것이 아니라 사용자 요구에 따라 **필요한 데이터를 신속하게 검색**할 수 있어야 하고 **데이터를 추가, 삭제, 수정**할 수 있도록 구성된 **데이터의 집합체**
- **관련된 데이터를 모아 이를 통합하여 관리할 수 있도록 하는 것**
 - 쇼핑몰사이트의 회원관리, 일반회사의 인사관리, 물품관리 등



데이터베이스의 정의

기업의 데이터를 저장하고
다수의 사용자가 동시에 사용하
더라도 데이터의 정합성과 속도
가 보장되는 것이 데이터베이스

integrated

- 중복이 배제된 통합된 데이터의 집합

stored

- 컴퓨터화되어 저장된 데이터

operational

- 조직의 기능을 수행하는데 반드시 유지해야 할 운영 데이터

shared

- 한 조직에 있는 여러 응용 시스템들이 공동으로 소유하고 유지하며 이용하는 공용데이터



데이터베이스 구축 목적

- 도서관
 - 회원, 책, 반납일, 연체료 등의 목록을 관리
- 회사
 - 사원, 부서, 급여 등에 대한 정보를 저장해야 함
- 관리하고자 하는 정보(Data)를 저장하기 위해서 데이터베이스를 구축함
- 정보 저장의 고전적인 방법 - 서류철을 관리하거나, 캐비닛에 문서들을 정리
 - => 발전된 정보 관리 방법 - 데이터베이스로 조직화된 정보 모음



데이터베이스 관리시스템

- 데이터베이스 관리시스템 (DataBase Management System, DBMS)
 - 모든 응용프로그램들이 데이터베이스를 공유할 수 있도록 관리해 주고 데이터베이스를 유지하기 위한 일련의 소프트웨어 시스템
 - 데이터를 적절하고 효율적으로 관리하기 위한 체계적인 시스템
 - 예) Oracle, MS SQL Server, mysql, informix, sybase 등



DBMS 기능

- [1] 정의기능 (생성)

- 논리적 구조 정의 : 테이블 등의 논리적 구조를 정의(오라클에서 볼 수 있는 것)
- 물리적 구조 정의: 실제 파일 (물리적으로 실제 파일을 생성)
- 논리적/물리적 구조 매핑 : 물리적으로 만든 파일과 우리가 볼 수 있는 것(오라클을 통해서 논리적으로 보는 것)을 연결

- [2] 조작기능

- 데이터를 검색, 삽입, 수정, 삭제 할 수 있는 기능
- Select, insert, update, delete

- [3] 제어기능

- 데이터의 정확성, 안정성을 유지하는 기능(무결성,보안)

데이터베이스와 테이블의 관계

데이터베이스



데이터베이스라는 큰 통안에 정보가 들어있는 다수의 테이블이 존재함

- 테이블 : 여러 개의 레코드를 담고 있는 논리적인 구조를 말함
- 데이터베이스에 사용되는 데이터들은 정해진 형식과 규칙대로 저장되어야 함.
- 이들 데이터를 저장하기 위한 기본 형식을 제공하는 것이 바로 테이블임
- 테이블은 데이터를 저장하고 추가, 삭제, 수정 하는 등의 기본적인 데이터 관리 기능을 함



데이터베이스의 예

쇼핑몰 데이터베이스

회원 테이블

아이디	이름	주민번호	주소
Hong	홍길동	880101-1112222	사당동
Lee	이재석	800205-1234141	홍제동
Kim	김창원	790608-1552212	연희동
Park	박길자	900609-2224565	상도동

제품 테이블

제품코드	제품명	가격	제조회사
C001	캐논익서스	350000	캐논
C002	후지파인픽스	420000	후지
C003	니콘쿨픽스	370000	니콘
C004	삼성케녹스	320000	삼성

데이터베이스

데이터베이스의 예

대학(학사 정보 시스템)에서는 데이터베이스에 학생들에 관하여 **신상 정보, 수강 과목, 성적** 등을 기록하고, 각 학과에 개설되어 있는 과목들에 관한 정보를 유지하고, 교수에 관해서 **신상 정보, 담당 과목, 급여 정보**를 유지한다.

항공기 예약 시스템에서는 여행사를 통해 항공기 좌석을 예약하면 모든 **예약 정보**가 데이터베이스에 기록된다.

예) 학생 테이블

name	no	address	tel	dept
홍길동	160001	사당동	010-100-2007	컴퓨터
이재석	160002	홍제동	011-300-1212	정보통신
김창원	160003	연희동	010-266-4550	영어영문
박길자	160004	상도동	011-290-1145	경영
이혜영	160005	신정동	019-316-4788	경제



데이터베이스의 특징

■ 데이터베이스의 특징

- 데이터베이스는 **데이터의 대규모 저장소**로서, 여러 부서에 속하는 여러 사용자에게 의해 동시에 사용됨
- 모든 데이터가 **중복을 최소화하면서 통합됨**
- 데이터베이스는 한 조직체의 운영 데이터뿐만 아니라 그 데이터에 관한 설명(**데이터베이스 스키마, 전체적인 데이터베이스 구조**)까지 포함.
- 효율적으로 접근이 가능하고 **질의**를 할 수 있음
- 데이터의 독립성 - 프로그램과 데이터 간의 독립성이 제공됨
- **데이터의 무결성** - 데이터베이스 안의 데이터는 오류가 있어서는 안됨



데이터베이스 장점/단점

■ 장점

- 데이터 중복 최소화
- 데이터 공유
- 무결성 유지
- 일관성 유지
- 데이터의 보안 보장

■ 단점

- 운영비 증대 : 많은 시스템 자원 요구
- 자료 처리의 복잡화 : 고급 프로그래밍 요구
- 복잡한 예비와 회복 : 장애 발생 대비를 위한 작업 필요
- 시스템의 취약성 : 시스템의 성능에 따라 DBMS 성능이 좌우됨

관계형 데이터베이스(RDBMS)

■ 관계형 데이터모델

- 1970년 중반 E.F.Codd 에 의해 관계형 데이터 모델 개념이 제안됨, RDBMS의 시초.
- 단순하면서도 모델링이 쉽다는 장점
- 관계형 데이터 모델은 실체(Entity),속성(Attribute),관계(Relationship)로 구성된 ER Diagram으로 표현

■ 관계형 데이터베이스(Relational-Database)

- 가장 대표적인 데이터베이스 시스템
- 2차원 수평적 구조(친숙하고 이해하기 쉬운 구조)
 - 데이터들을 2차원 구조를 가진 테이블 형태로 저장
- 데이터의 무결성, 트랜잭션 처리 등 데이터베이스 관리 시스템으로써의 기본적인 기능면에서 뛰어난 성능
- 질의어를 사용한 데이터 접근 방법
 - 질의어 - 일정한 패턴이 있으며 이 패턴에만 맞게 질의하고자 하는 조건들을 나열하면 찾고자 하는 정보를 검색해 줌



관계형 데이터베이스

■ ※ 데이터베이스의 구조

■ 필드(Field) – 레코드(Record) – 테이블(Table)

- 필드 : 테이블의 열에 해당되는 것 (컬럼)
- 레코드 : 필드들의 집합체, 테이블의 각 행(row)
- 인덱스 : 데이터베이스에서 데이터를 신속하게 찾을 수 있도록 정렬된 목록, 키
- 테이블 : 여러 개의 레코드를 담고 있는 논리적인 구조를 말함
 - 데이터베이스에 사용되는 데이터들은 정해진 형식과 규칙대로 저장되어야 함. 이들 데이터를 저장하기 위한 기본 형식을 제공하는 것이 바로 테이블임
 - 테이블은 데이터를 저장하고 추가, 삭제, 수정 하는 등의 기본적인 데이터 관리 기능을 함



테이블이란

- 테이블
 - 어느 특정한 주제와 목적으로 만들어지는 하나의 집합
 - 데이터베이스의 기본단위 - 테이블
 - 모든 자료는 테이블에 등록이 되고, 우리는 테이블로부터 원하는 자료를 꺼내볼 수 있다
 - 내가 알고 싶어하는 자료의 기본이 테이블
- 데이터베이스라는 큰 통안에 정보가 들어있는 다수의 테이블이 존재함
- 우리가 원하는 자료를 찾아보려면
 - 1) 데이터베이스에 접속하고
 - 2) 테이블을 찾아봐야 함

데이터베이스와 테이블의 관계



데이터베이스라는 큰 통안에
정보가 들어있는 다수의 테이블이
존재함

테이블(Table)

테이블

필드(column, 열)

아이디	이름	주민번호	주소
Hong	홍길동	880101-1112222	사당동
Lee	이재석	800205-1234141	홍제동
Kim	김창원	790608-1552212	연희동 레코드(row, 행)
Park	박길자	900609-2224565	상도동



관계형 데이터베이스

■ 관계내용 정의

- 관계란? 관계는 업무적인 **연관성**이다.
- 관계는 두 테이블 사이에 존재하며, 관계를 맺고 있는 두 테이블 중 반드시 하나는 부모 테이블이고 하나는 자식 테이블이다.
- 부모 테이블의 기본키(Primary key)는 자식 테이블에 외래키(Foreign Key)로 전이 되어진다.

• 외래키(Foreign key)

테이블들 간에 관계를 맺고 있을 때 자신의 개체(테이블)내에서 키의 역할을 하지는 않지만, 다른 개체(테이블)의 기본키와 동일한 값을 가지고 있어서 **다른 개체(테이블)을 참조할 때 사용하는 필드**

사원번호	이름	부서	직위	생일	거주지	전화번호	성명	관계	직장명	생년월일
A-101	홍길동	영업부	사원	1980-07-03	서울시 동작구	02-1111-1111	홍영국	부	일등교역	1950-02-22
A-101	홍길동	영업부	사원	1980-07-03	서울시 동작구	02-1111-1111	김영자	모	가사	1952-09-17
A-101	홍길동	영업부	사원	1980-07-03	서울시 동작구	02-1111-1111	홍복동	남동생	대학생	1988-11-04
A-102	김이정	영업부	대리	1977-04-11	서울시 성북구	02-2222-2222	전태균	남편	대성산업	1972-06-25
A-102	김이정	영업부	대리	1977-04-11	서울시 성북구	02-2222-2222	전오성	아들	초등학생	2000-12-10
A-102	김이정	영업부	대리	1977-04-11	서울시 성북구	02-2222-2222	전이경	딸	유치원	2002-05-05
A-103	방승주	영업부	사원	1985-10-27	서울시 영등포구	02-3333-3333	방영택	부	경진테크	1954-08-19
A-103	방승주	영업부	사원	1985-10-27	서울시 영등포구	02-3333-3333	이문옥	모	가사	1955-07-23
A-103	방승주	영업부	사원	1985-10-27	서울시 영등포구	02-3333-3333	방연주	언니	서일항공	1983-01-02

사원, 가족관계가 하나의 테이블에 있는 경우
=> 데이터가 중복됨

사원 테이블

사원번호	이름	부서	직위	생일	거주지	전화번호
A-101	홍길동	영업부	사원	1980-07-03	서울시 동작구	02-1111-1111
A-102	김이정	영업부	대리	1977-04-11	서울시 성북구	02-2222-2222
A-103	방승주	영업부	사원	1985-10-27	서울시 영등포구	02-3333-3333
A-104	최철수	영업부	사원	1982-01-14	서울시 은평구	02-4444-4444
A-105	조성국	영업부	과장	1972-05-06	서울시 광진구	02-5555-5555
A-106	이영희	영업부	사원	1981-09-13	서울시 마포구	02-6666-6666
A-107	배용진	영업부	사원	1984-08-21	서울시 서대문구	02-7777-7777
A-108	구봉성	영업부	사원	1987-12-09	서울시 강남구	02-8888-8888
A-109	강남진	영업부	대리	1978-03-15	서울시 동대문구	02-9999-9999
A-110	공현주	영업부	사원	1983-11-28	서울시 금천구	02-1111-2222

사원번호	성명	관계	직장명	생년월일
A-101	홍영국	부	일등교역	1950-02-22
A-101	김영자	모	가사	1952-09-17
A-101	홍복동	남동생	대학생	1988-11-04
A-102	전태균	남편	대성산업	1972-06-25
A-102	전오성	아들	초등학생	2000-12-10
A-102	전이경	딸	유치원	2002-05-05
A-103	방영택	부	경진테크	1954-08-19
A-103	이문옥	모	가사	1955-07-23
A-103	방연주	언니	서일항공	1983-01-02

가족관계 테이블

사원번호	이름	부서	직위	생일	거주지	전화번호
A-101	홍길동	영업부	사원	1980-07-03	서울시 동작구	02-1111-1111
A-102	김이정	영업부	대리	1977-04-11	서울시 성북구	02-2222-2222
A-103	방승주	영업부	사원	1985-10-27	서울시 영등포구	02-3333-3333
A-104	최철수	영업부	사원	1982-01-14	서울시 은평구	02-4444-4444
A-105	조성국	영업부	과장	1972-05-06	서울시 광진구	02-5555-5555
A-106	이영희	영업부	사원	1981-09-13	서울시 마포구	02-6666-6666
A-107	배용진	영업부	사원	1984-08-21	서울시 서대문구	02-7777-7777
A-108	구봉성	영업부	사원	1987-12-09	서울시 강남구	02-8888-8888
A-109	강남진	영업부	대리	1978-03-15	서울시 동대문구	02-9999-9999
A-110	공현주	영업부	사원	1983-11-28	서울시 금천구	02-1111-2222

사원 테이블

기본키(Primary key)
- 하나의 행을 다른 행
과 유일하게 구별할 수
있도록 해 주는 열

거래처 테이블

사원번호	담당관리자	거래처명	거래처담당	거래개시일	전화번호	거래처주소
A-101	홍길동	일성산업	김인석	2000-03-01	02-1234-0123	서울시 구로구
A-101	홍길동	한국물산	배동선	2002-07-18	02-1234-1234	서울시 강북구
A-102	김이정	동성기획	이효선	2001-10-28	02-1234-2345	서울시 용산구
A-102	김이정	백영전자	곽한별	2000-02-11	02-1234-3456	서울시 송파구
A-103	방승주	도국건설	박성익	2000-06-25	02-1234-4567	서울시 은평구
A-103	방승주	정일제약	조한성	2001-09-09	02-1234-5678	서울시 동대문구
A-104	최철수	성진건설	장철신	2003-05-14	02-1234-6789	서울시 양천구
A-104	최철수	선일기업	안병만	2002-11-27	02-1234-7890	서울시 강서구
A-105	조성국	가나건설	정서희	2001-08-03	02-1234-8901	서울시 서초구

관계형 데이터베이스

- 고객 테이블

아이디	이름	주소	전화번호
hong	홍길동	사당동	310-1122
lee	이재석	홍제동	250-6985
kim	김창원	연희동	470-1258
park	박길자	상도동	277-8952

PK(기본키)

- 구매 테이블

아이디	구매한 상품	가격	수량
hong	캐논 익서스	300000	1
hong	노트북	2500000	1
lee	삼성 케녹스	320000	1
lee	청바지	120000	1
lee	운동화	50000	2
park	디오스 냉장고	1600000	1
park	삼성 케녹스	320000	1

FK(외래키)

다른 개체(테이블)을 참조할 때 사용하는 필드

관계형 데이터베이스

- 학생 테이블

학번	이름	주소	학과
160001	홍길동	사당동	컴퓨터
160002	이재석	홍제동	정보통신
160003	김창원	연희동	영어영문
160004	박길자	상도동	경영
160005	이혜영	신정동	경제

- 과목 테이블

과목번호	과목명
K	국어
M	수학
E	영어
H	역사

- 수강 테이블

학번	일련번호	과목번호
160001	1	K
160001	2	M
160002	1	K
160002	2	E
160003	1	M
160003	2	E
160003	3	H
160004	1	K
160005	1	E



오라클 소개

- 오라클(Oracle) : 가장 대표적인 관계형 DBMS 중의 하나
- 1977년 래리 앨리슨이 설립
 - 래리 앨리슨이 참여했던 데이터베이스 관련 프로젝트의 암호명이 오라클
- 1983년 회사 이름을 Oracle Corporation 으로 변경
- 오라클이란 이름은 회사 이름인 동시에 DBMS 제품 이름임
- 이후 지속적으로 기능을 향상시켜 제품을 출시
- 버전 8부터 버전번호 뒤에 i (internet)가 붙음
 - 인터넷 개발과 플랫폼으로써 개발된 데이터베이스
- 버전 10부터 버전번호 뒤에 g(grid)가 붙음
 - 그리드(grid) 특성을 추가함.
 - 여러 대의 컴퓨터를 마치 하나의 커다란 컴퓨터인양 사용하는 것
- 2018년 현재 버전 18c 까지 출시됨
 - oracle 8i, 9i, 10g, 11g, 12c, 18c, 19c



오라클 Edition 별 특징

- Standard Edition (SE)
 - CPU 4개 이하, RAC 기능 사용 불가
- Enterprise Edition (EE)
 - 상용으로 사용할 경우 가장 널리 사용됨
 - CPU 4개 이상, 모든 기능 사용 가능
- Standard Edition One
 - 10g에서 최초 등장, CPU 2개 이하
- Express Edition (XE)
 - Free.
 - 호스트 당 1개 CPU, 1GB 메모리, 최대저장용량 4GB.
- Personal Edition
 - 개인용 에디션, 오직 1명의 사용자만 등록하여 사용
- Oracle Database Like (모바일용)



Oracle 11g Express 설치

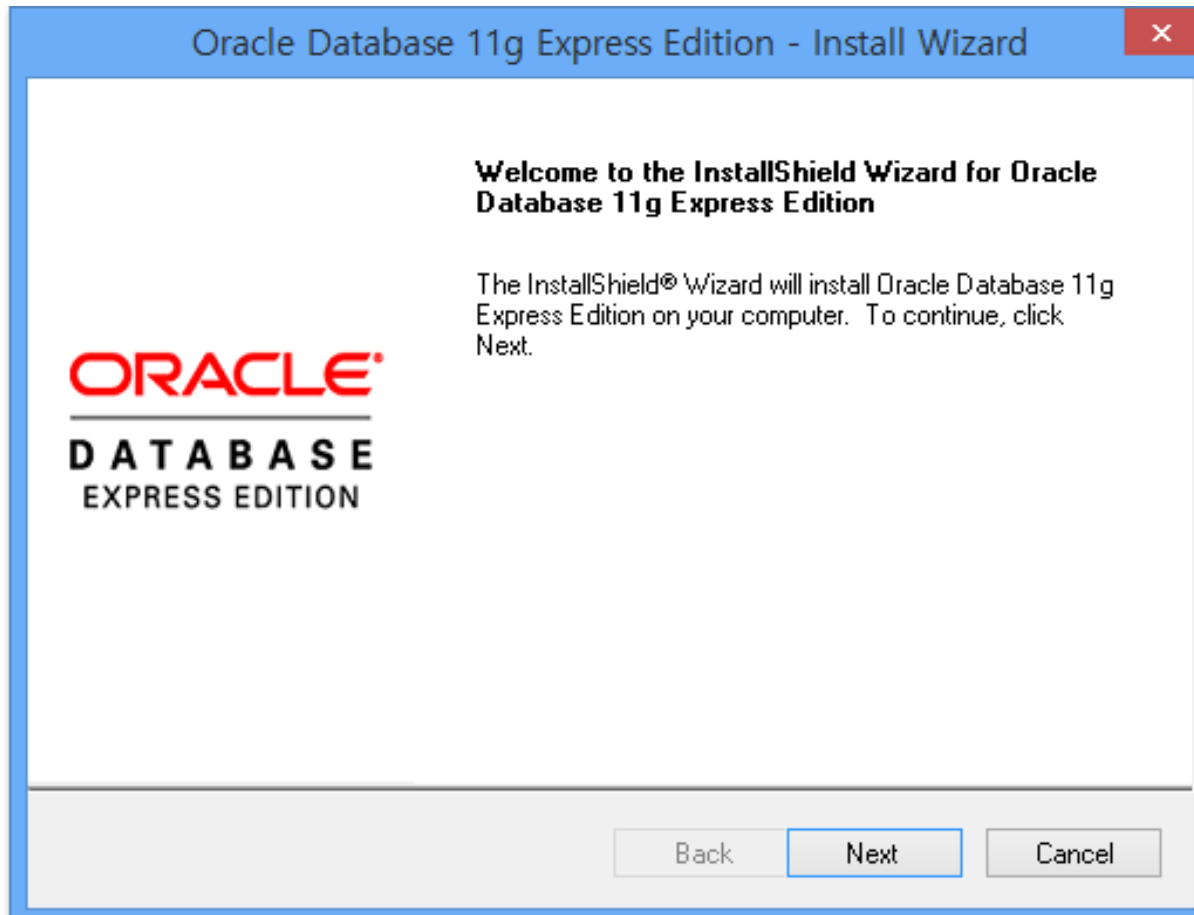


Oracle 11g Express 설치

- 익스프레스 에디션(Express Edition)
 - 무료로 사용할 수 있는 오라클 데이터베이스를 제공함
- 상용 제품에 비해 아래와 같은 성능의 제약이 있다
 - 유저 데이터베이스 공간 11GB까지 사용
 - 물리적 메모리 1GB까지 사용
 - 서버당 CPU 1개까지 사용
- 다운로드를 위해 아래의 사이트에 접속
 - <https://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/xe-prior-releases-5172097.html>
- SqlDeveloper
 - <http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

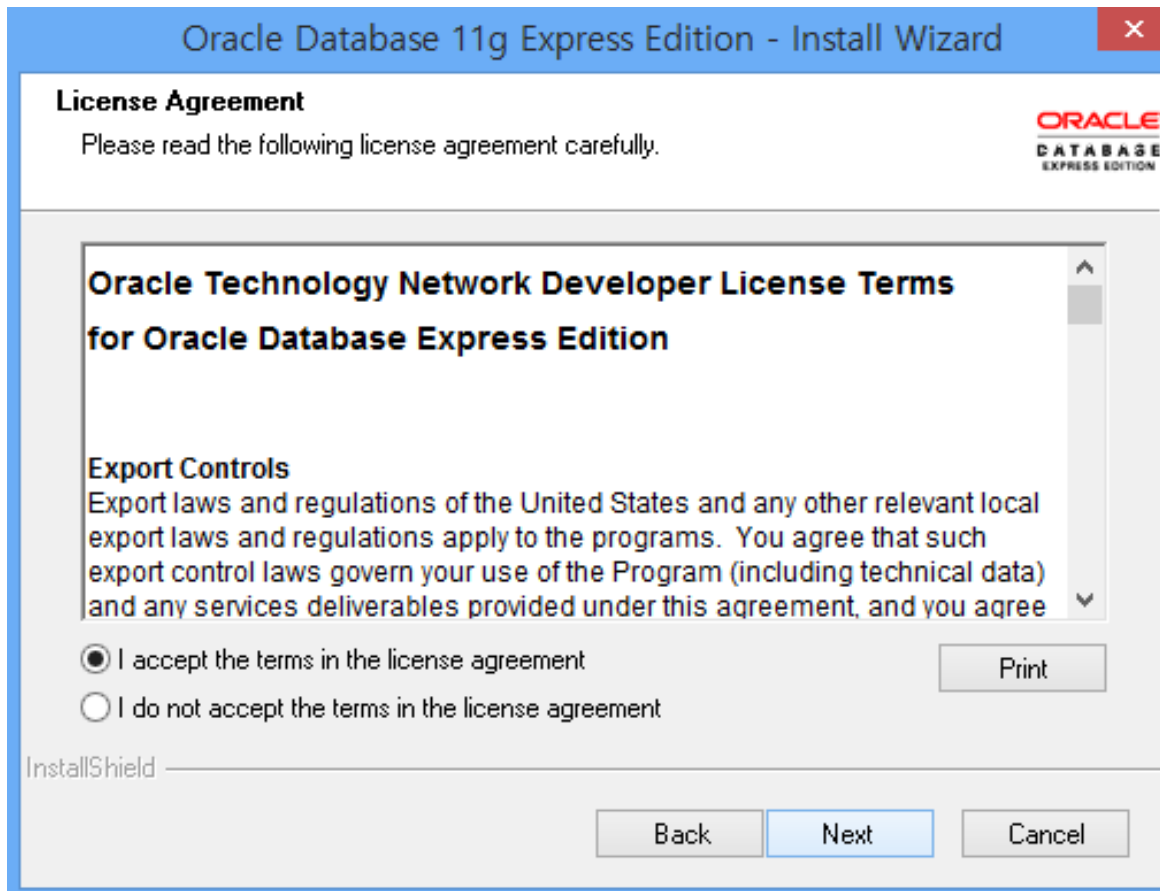
Oracle 11g Express 설치

- 다운로드 후 실행하면 아래와 같이 설치 시작 화면을 볼 수 있다



Oracle 11g Express 설치

- 라이선스 내용에 동의한다



The screenshot shows the 'Oracle Database 11g Express Edition - Install Wizard' window. The title bar is blue with a close button. The main content area is white and contains the 'License Agreement' section. It includes the Oracle logo and the text 'Please read the following license agreement carefully.' Below this is a scrollable text box containing the 'Oracle Technology Network Developer License Terms for Oracle Database Express Edition'. The text in the scroll box includes 'Export Controls' and a paragraph about export laws. At the bottom of the scroll box are two radio buttons: 'I accept the terms in the license agreement' (which is selected) and 'I do not accept the terms in the license agreement'. To the right of the radio buttons is a 'Print' button. At the bottom of the window are three buttons: 'Back', 'Next' (which is highlighted in blue), and 'Cancel'. The 'InstallShield' logo is visible in the bottom left corner.

Oracle Database 11g Express Edition - Install Wizard

License Agreement

Please read the following license agreement carefully.

ORACLE
DATABASE
EXPRESS EDITION

**Oracle Technology Network Developer License Terms
for Oracle Database Express Edition**

Export Controls
Export laws and regulations of the United States and any other relevant local
export laws and regulations apply to the programs. You agree that such
export control laws govern your use of the Program (including technical data)
and any services deliverables provided under this agreement, and you agree

☒ I accept the terms in the license agreement ☐ I do not accept the terms in the license agreement

Print

InstallShield

Back Next Cancel

Oracle 11g Express 설치

- 디폴트 설치 위치는 c:\oraclexe 인데 변경가능함

Oracle Database 11g Express Edition - Install Wizard

Choose Destination Location

Select folder where setup will install files.

Setup will install Oracle Database 11g Express Edition in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

☒ Oracle Database 11g Express Edition 631124 K

Destination Folder
C:\oraclexe\ Browse...

Space Required on C: 631124 K
Space Available on C: 10716444 K

InstallShield

Back Next Cancel

Oracle 11g Express 설치

- port 번호 확인

The screenshot shows the 'Specify Database Ports' window of the Oracle Database 11g Express Edition - Install Wizard. The window has a blue title bar with the text 'Oracle Database 11g Express Edition - Install Wizard' and a close button. The main content area is white with a blue border. It contains the text 'Specify Database Ports' and the Oracle logo. Below this, it says 'Enter the ports for Database services that are not currently in use.' There are three input fields: 'TNS Port' with the value '1521', 'MTS Port' with the value '2030', and 'HTTP Port' with the value '8080'. At the bottom, there are three buttons: 'Back', 'Next', and 'Cancel'. The 'Next' button is highlighted with a blue border. The 'InstallShield' logo is visible in the bottom left corner.

Oracle Database 11g Express Edition - Install Wizard

Specify Database Ports

ORACLE
DATABASE
EXPRESS EDITION

Enter the ports for Database services that are not currently in use.

TNS Port

MTS Port

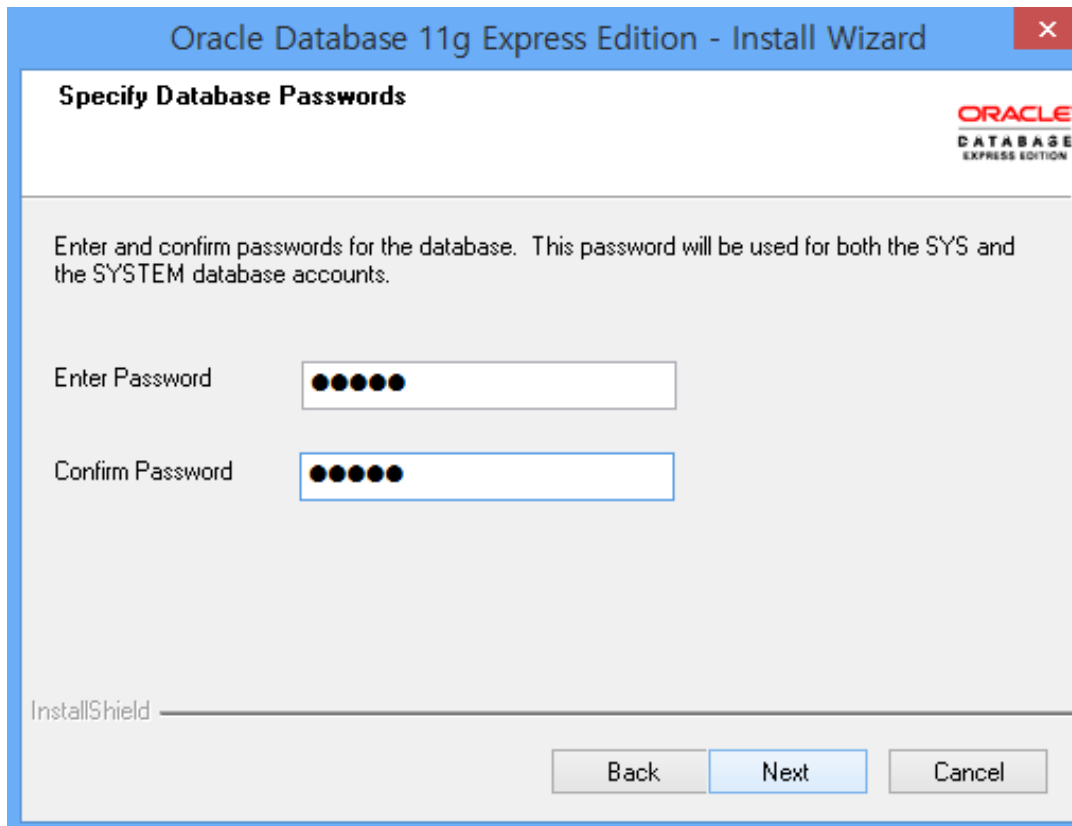
HTTP Port

InstallShield

Back Next Cancel

Oracle 11g Express 설치

- 설치 후 SYS로 접속할 때 사용할 관리자용 비밀번호를 설정한다



Oracle Database 11g Express Edition - Install Wizard

Specify Database Passwords

Enter and confirm passwords for the database. This password will be used for both the SYS and the SYSTEM database accounts.

Enter Password

Confirm Password

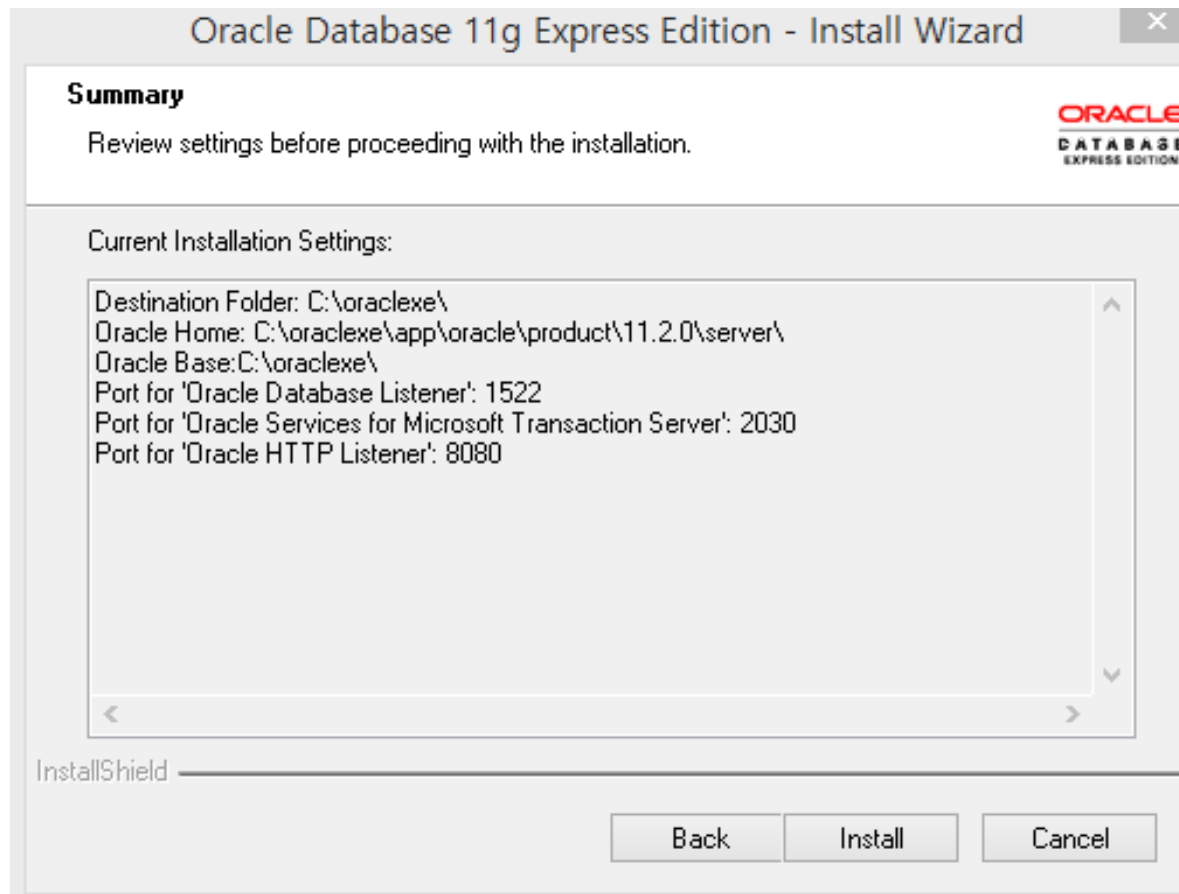
InstallShield

Back Next Cancel

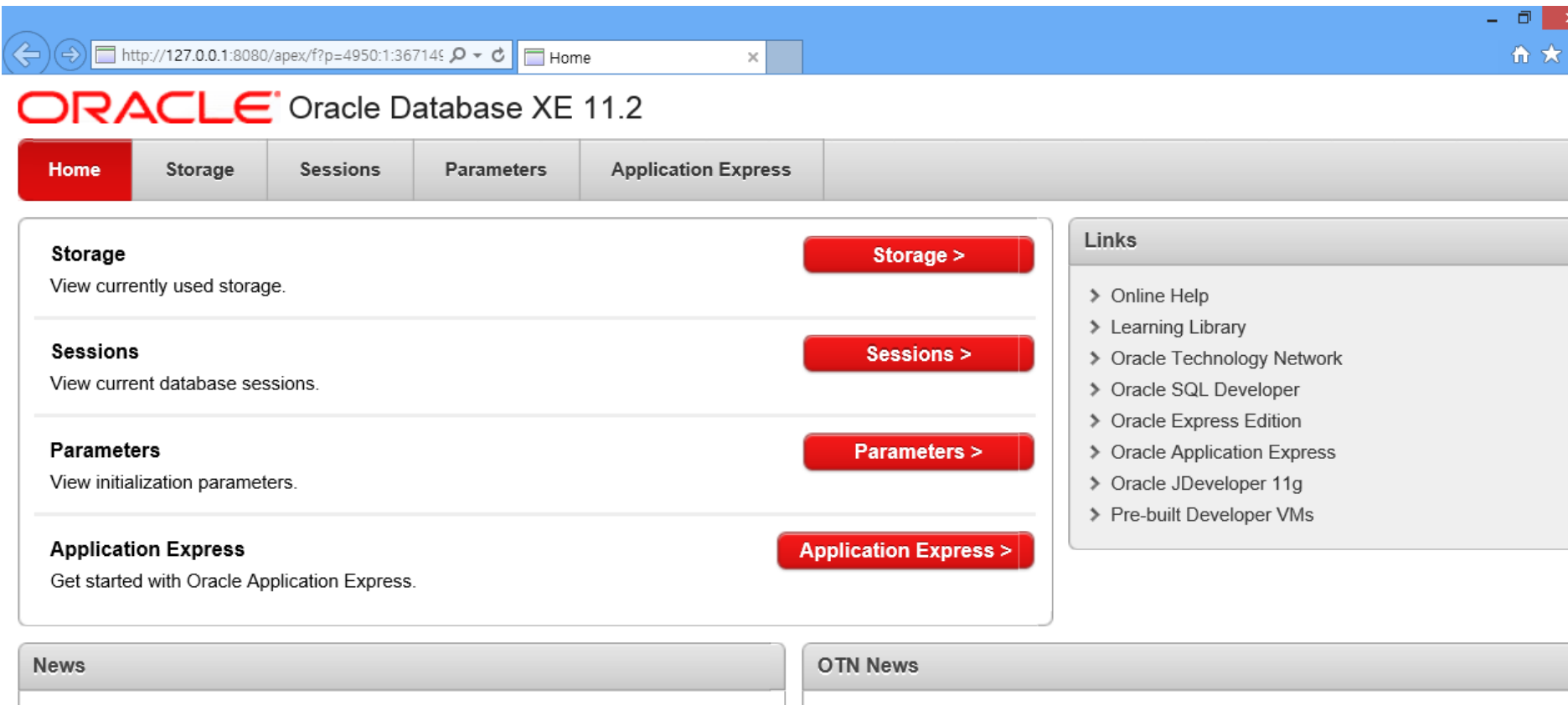
123\$

Oracle 11g Express 설치

- 설치와 관련된 설정 내용을 확인



- 설치가 완료되면 바탕화면에 생성된 "Get Started With Oracle Database" 아이콘을 클릭하여 아래와 같이 접속 화면이 나오면 설치가 정상적으로 진행된 것임





설정

- hr 사용자 잠금 해제
 - C:\W>sqlplus / as sysdba
 - SQL>alter user hr account unlock;
 - SQL>alter user hr identified by hr123;
 - hr123으로 hr의 비밀번호 변경
 - SQL>conn hr/hr123;
 - Connected. 정상 연결 확인



오라클 설치

- 설정파일 확인하기

- listener.ora
- tnsnames.ora
- C:\Woraclexe\Wapp\Woracle\Wproduct\W11.2.0\server\Wnetwork\WADMIN\Wlistener.ora

- SERVICE_NAME, HOST, PORT 확인

(HOST = 192.168.0.15)(PORT = 1521))

(SERVICE_NAME = XE)

(HOST = 202-00)(PORT = 1521))

(SERVICE_NAME = XE)

HOST => ip나 컴퓨터
이름을 넣는다



설치 시 오류

- Universal Installer 자체의 오류.
- 9i부터 설치 오류가 많다고 보고됨.
- 오류의 종류
 - JDK 버전 오류(1.4 기준).
 - 설치 디렉토리 문제 (한글이름, 정확한 경로)
 - Enterprise Manager 문제
 - iSQL*Plus 문제
 - 윈도우 사용자 권한 문제
- 설치 오류 시, 가장 좋은 방법은 제거 후 재 설치



오라클 제거방법

- Universal Installer를 통해 제거.
- 완전히 제거가 안됨(UI 버그)
- 제거 방법
 - 서비스 정지.
 - Universal Installer를 실행해 제거.
 - 관련 파일 제거

(1) Registry

HKEY_CURRENT_USER\SOFTWARE\ORACLE

HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application\Oracle.oracle

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\OracleDBConsole

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Oracle10g_home

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\OracleService



오라클 제거방법

- (2) 환경변수 제거 (ORACLE_HOME, ORACLE_SID, TNS_NAME, PATH..)
- (3) 파일 제거



SQL Developer

- 쿼리 작성 툴
 - SQL*Plus, iSQL*Plus
 - SQL Developer
 - Toad, Orange 등
- SQL Developer
 - 오라클 사에서 직접 제작한 툴로, 무료로 사용가능.
 - 9i 이후 버전에서만 사용가능
 - <http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html?ssSourceSiteId=otnkr>
- 토드 (Toad)
 - <http://www.toadworld.com/downloads#oracle>
 - ToadForOracle_Trial_13.1.0.78_x64_En.msi

오라클 실행

- hr 유저의 계정 풀기
alter user hr account unlock;

- 암호변경
alter user hr identified by hr123;

- 설치 완료 후 실행하는 방법
 - 오라클 서비스 시작하기
 - 제어판-관리도구-서비스-OracleServiceXE - 시작
 - 또는 cmd 창에서 net start OracleServiceXE
- [1] 설치한 프로그램으로 접속
 - 시작-> 프로그램->오라클 -> 응용 프로그램 개발 -> SQL PLUS 클릭
 - 또는 Toad for Oracle
- [2] cmd 창에서 접속
 - **sqlplus conn as sysdba** 또는 **sqlplus / as sysdba**
 - 접속은 conn 명령어
 - 수정은 alter 명령어
 - 사용자 계정 수정 (hr 유저의 계정 잠그기/풀기)
 - 예) alter user hr account lock; (계정 잠그기)
 - **alter user hr account unlock; (풀기)**
 - 암호변경 : **alter user 사용자ID identified by 변경할 암호**
 - 예) **alter user hr identified by hr123;**



오라클 실행

- 오라클이 실행 중인지 확인하기
 - 관리 도구 -> 서비스
 - 서비스에 OracleServiceXE, oracleListener가 시작됨으로 되어있으면 작동 중임.



설치 후 작업

■ 사용자 계정

- sys, system – 오라클 데이터베이스 관리자
 - 오라클 데이터베이스 내에서 시스템적인 작업을 할 수 있는 DBA 권한이 자동으로 할당되어 있음
 - sys – 시스템 정보를 갖고 있는 모든 객체들(테이블, 뷰 등)에 접근할 수 있고 이러한 객체들의 소유자임
 - 데이터 디렉터리의 소유자
 - sys 사용자가 system 사용자보다 더 많은 권한을 가짐
 - system – 오라클 데이터베이스의 여러 옵션들과 툴 등에 대한 정보를 가진 객체들의 소유자
 - 모든 권한이 SYS와 같으나 데이터베이스 생성 권한 없음
- scott, hr: SAMPLE 사용자 계정



오라클 기본

- SQL기본
 - SQL은 대소문자 구분 안함
 - 데이터는 대소문자 구분
 - Data Type은 일치시켜야 함
 - sql문을 종료할 때는 ;(세미콜론)을 반드시 사용
- SQL주석
 - Single Line : --
 - Multi Line : /* */



오라클 기본 명령어

- 생성된 테이블 목록보기
 - `select * from tab;`
 - `tab` : 테이블이름(데이터베이스에 생성되어 있는 테이블정보를 가지고 있는 테이블)
- 특정 테이블의 정보를 볼 수 있는 명령어 : `describe`
 - `describe emp;` 또는 `desc emp;`
- DUAL 테이블
 - 정해진 테이블이 없이 연산을 사용하고자 할 때 `from`절에 사용할 수 있는 가상 테이블
 - `select 10*20 from dual;`
 - `desc sys.dual;`
 - `select * from dual;`
- oracle sql plus에서도 가능



SQL 소개

What Is SQL?

구조적 질의 언어(SQL)

- RDBMS를 액세스해서 데이터를 조작할 수 있는 언어

- DBMS는 데이터의 조회, 저장, 삭제와 같은 작업은 SQL을 통해 처리함

- SQL

- Structured Query Language.

- DBMS에서 사용하는 데이터 질의 및 처리 언어.

- 데이터베이스에서 데이터를 조회, 입력, 수정, 삭제, 저장하는데 사용되는 질의 언어

- SQL의 특징

- 타 언어에 비해 배우고 사용하기 쉽다.

- 구조화된 언어 : 어느 일정한 틀이나 패턴이 있다는 뜻

- 집합적, 선언적 언어

- ANSI, ISO 표준 (DBMS 종류에 상관 없이 사용 가능)

- 개발자, DBA, DB 모델링 종사자들이 필수로 알아야 하는 언어

- RDBMS와 함께 탄생, 발전

- SEQUEL (Structured English Query Language)

구조화된 언어 : 어느 일정한 틀이나 패턴이 있다는 뜻
-이러한 틀과 패턴에 맞게 조건들을 나열하기만 하면
SQL은 우리가 원하는 결과를 가져다 줌

• 질의어(Query Language)

- 데이터베이스의 데이터를 읽고, 쓰기 위한 용도
질의(Query)는 영문을 그대로 발음하여 쿼리라고
도 부름



SQL 문의 종류

- DML (Data Manipulation Language) : 데이터 조작어
 - 데이터를 조회, 저장, 삭제할 때 사용되는 SQL 문장
 - SELECT
 - INSERT, UPDATE, DELETE
 - COMMIT, ROLLBACK
- DDL (Data Definition Language) : 데이터 정의어 , 객체 조작어
 - 데이터베이스 객체들(테이블, 인덱스, 뷰, 트리거 등)을 생성하는데 사용되는 SQL 문장
 - 데이터베이스나 테이블을 생성할 목적으로 사용하는 언어
 - CREATE, DROP, ALTER, TRUNCATE
 - 삭제하는 것 수정하는 것
- DCL (Data Control Language) : 데이터 제어어
 - 데이터베이스에 있는 데이터에 접근을 제어하는 언어
 - GRANT, REVOKE

GRANT - 데이터베이스 사용자에게 사용권한을 부여하는 명령
REVOKE - 그랜트의 반대로 사용권한을 취소하는 명령



SQL 문장의 구성요소

- 예약어 (Reserved Word)
- 연산자 (Operator)
 - 단항, 이항 연산자
 - 수식, 문자, 논리, 계층형 쿼리, Set 연산자
- 의사컬럼 (Pseudocolumn) – 모조, 유령 컬럼
 - ROWNUM
 - ROWID : 각 row가 저장된 주소 값
- 함수 (Function)
 - 내장 함수
 - 사용자 정의 함수
- 표현식 (Expression)
- 조건 (Condition)



SQL 문장의 구성요소

- 표현식 (Expression)
 - select 리스트
 - WHERE, HAVING 의 조건절
 - ORDER BY, CONNECT BY, START WITH
 - INSERT문의 VALUES 절.
 - UPDATE 문의 SET 절
- 조건 (Condition)
 - 하나 이상의 표현식과 논리연산자가 결합되어, 그 결과가 TRUE, FALSE, UNKNOWN 값을 반환하는 것.



Select – 데이터 조회하기

DESC 명령어

- 특정 테이블에 어떤 칼럼이 있는지 조회하는 명령어
- Describe의 약자

```
SCOTT>desc emp;
```

이름	널?	유형
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

select * from 테이블
where 조건

select – 데이터 조회하기

```
select [칼럼명 또는 표현식] from [테이블명, 뷰명];
```

- (1) 모든 칼럼 조회하기
 - select * from emp;
 - 여러 줄에 써도 됨 (단, SQL 키워드는 분리해서는 안됨)

```
SCOTT>select *  
      2  from emp;
```

- SQL은 대소문자 구분하지 않고 입력해도 수행됨
- 문장의 마지막 부분에는 반드시 세미콜론 (;) 입력해야 함

select - 데이터 조회하기

■ (2) 원하는 칼럼만 조회하기

- `select empno, ename from emp;`

• 조회하려는 칼럼이 여러 개일 경우, (콤마)로 구분하고 이름을 여러 개 적어 줌

■ (3) 표현식을 사용하여 출력하기

- 표현식 (literal 상수, 문자)

- 칼럼 이름 이외에 출력하기를 원하는 내용을 의미
- select 구분 뒤에 '(홀따옴표)로 묶어서 사용
- `select name, '선생님 안녕하세요' from professor;`

```
SCOTT>select name, '선생님 안녕하세요' from professor;
```

NAME	'선생님안녕하세요'
조인형	선생님 안녕하세요
박승곤	선생님 안녕하세요
송도권	선생님 안녕하세요

```
SCOTT>select empno, ename from emp;
```

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS

EMPNO	ENAME
7900	JAMES
7902	FORD
7934	MILLER

14 개의 행이 선택되었습니다.

select - 데이터 조회하기

■ (4) 칼럼 별칭 사용하여 출력하기

■ 별칭 사용 전

```
SCOTT>select studno, name from student;
```

STUDNO	NAME
9411	서진수
9412	서재수
9413	이미경

• 원래 테이블의 칼럼명이 변경된 것이 아니라 출력될 때 임시로 바꾸어서 보여주는 것
=> 이 기능을 칼럼 별칭(Column Alias) 이라고 부름

■ 별칭 사용 후 - studno, name 부분을 다른 이름으로 변경해서 출력

```
SCOTT>select studno 학번, name as "이름", profno "지도교수 번호" from student;
```

학번	이름	지도교수 번호
9411	서진수	1001
9412	서재수	2001
9413	이미경	3002

select - 데이터 조회하기

■ 칼럼 별칭 사용법

- 1. 칼럼명 뒤에 공백 주고 별명 쓰기
- 2. 칼럼명 뒤에 as 키워드 사용 후 별명 주기

■ 별명을 줄때 반드시 "(쌍따옴표)로 감싸야 하는 경우

- 별명에 공백이나 일부 특수문자(% , @ , & , | , * , + , - 등), 대소문자 구분이 필요한 경우

■ 실습>

- dept2 테이블을 사용하여 dcode를 부서#, dname을 부서명, area를 위치로 별명을 설정하여 출력하기

부서#	부서명	위치
0001	사장실	포항본사
1000	경영지원부	서울지사
1001	재무관리팀	서울지사
1002	총무팀	서울지사
1003	기술부	포항본사

select – 데이터 조회하기

- (5) DISTINCT – 중복된 값을 제거하고 출력하기
 - 중복된 데이터를 빼고 출력

```
SCOTT>select distinct deptno1 from student;
```

DEPTN01
102
201
301
101
202
103

6 개의 행이 선택되었습니다.

- distinct 키워드는 반드시 select 키워드 다음에
와야 함

```
SCOTT>select deptno1 from student;
```

DEPTN01
101
102
103
201
202
101
102
202
301
201
101
102
201
201
301
101
201
201
102
102
103

20 개의 행이 선택되었습니다.

select – 데이터 조회하기

- select deptno from emp2;

DEPTNO

0001
1000
1000
1001
1002
1003
1004
1004
1005
1005
1006
1007
1008
1008
1009
1009
1010
1010
1011

- select distinct deptno from emp2;

DEPTNO

0001
1000
1001
1002
1003
1004
1005
1006
1007
1008

select – 데이터 조회하기

distinct 사용 안 함

```
SCOTT>select deptno, job from emp;
```

DEPTNO	JOB
20	CLERK
30	SALESMAN
30	SALESMAN
20	MANAGER
30	SALESMAN
30	MANAGER
10	MANAGER
20	ANALYST
10	PRESIDENT
30	SALESMAN
20	CLERK

DEPTNO	JOB
30	CLERK
20	ANALYST
10	CLERK

14 개의 행이 선택되었습니다.

두 컬럼 distinct 사용함

```
SCOTT>select distinct deptno, job  
2 from emp;
```

DEPTNO	JOB
20	CLERK
30	SALESMAN
20	MANAGER
30	CLERK
10	PRESIDENT
30	MANAGER
10	CLERK
10	MANAGER
20	ANALYST

9 개의 행이 선택되었습니다.

- 주의사항 – distinct 키워드는 1개의 컬럼에만 적어도 모든 컬럼에 적용됨

select – 데이터 조회하기

- (6) 연결연산자로 칼럼을 붙여서 출력하기
 - 서로 다른 칼럼을 하나의 칼럼인 것처럼 연결해서 출력하는 경우
 - **연결 연산자 ||**
 - || 기호를 기준으로 왼쪽과 오른쪽 칼럼을 합쳐서 하나의 칼럼처럼 화면에 출력해 줌
 - 연결연산자는 하나만 사용할 수도 있고, 여러 개를 함께 사용할 수도 있음
 - 연결 연산자와 리터럴 문자를 함께 많이 사용

```
SCOTT>select name, position from professor;
```

NAME	POSITION
조인성	교수
박승우	교수
송도원	강사
양희재	강사
김종수	교수
김민준	교수
김나현	교수
김시원	교수
최스승	강사
최기호	교수

```
SCOTT>select name || position from professor;
```

NAME POSITION
조인성교수
박승우교수
송도원강사
양희재강사
김종수교수
김민준교수
김나현교수
김시원교수
최스승강사
최기호교수

```
SCOTT>select name || ' 교수 ' || position || ' 입니다.'
2 from professor;
```

NAME||'남은'||POSITION||'입니다'

[illegible]

학생의 키와 몸무게	
서서	72kg
이재민	64kg
김동수	52kg
박재홍	83kg
김신영	70kg
신오라	48kg
구유미	42kg
임세지	55kg
	58kg
	54kg
	72kg

- 실습 - student 테이블을 사용하여 모든 학생들이 '서진수의 키는 180 cm, 몸무게는 55kg 입니다' 와 같은 형식으로 출력되도록 리터럴 문자를 추가하고, 칼럼이름은 "학생의 키와 몸무게"라는 별명으로 출력하기

select – 데이터 조회하기

■ (7) 산술 연산자 사용하기

■ 산술연산자 : +, -, *, /

```
SQL> select ename, sal, sal+100 from emp;
```

ENAME	SAL	SAL+100
SMITH	800	900
ALLEN	1600	1700
WARD	1250	1350
JONES	2975	3075

■ 연산자 우선순위 조절 : 괄호 () 이용

```
SQL> select sal, sal+100/2, (sal+100)/2 from emp;
```

SAL	SAL+100/2	(SAL+100)/2
800	850	450
1600	1650	850
1250	1300	675
2975	3025	1537.5
1250	1300	675

• DUAL 테이블

-정해진 테이블이 없이 연산을 사용하고자 할 때 from절에 사용할 수 있는 가상 테이블

```
select 10*20 from dual;
```

```
desc sys.dual;
```

```
select * from dual;
```

select – 데이터 조회하기

- (8) where 절을 활용하여 원하는 조건만 조회하기

```
select [칼럼명 또는 표현식]
from [테이블명, 뷰명]
where 원하는 조건;
```

- emp 테이블에서 10번 부서에 근무하는 사원의 이름과 급여, 부서번호를 출력

```
SQL> select ename, sal, deptno from emp
2  where deptno = 10;
```

ENAME	SAL	DEPTNO
CLARK	2450	10
KING	5000	10
MILLER	1300	10

select – 데이터 조회하기

- emp 테이블에서 급여(sal)가 4000보다 큰 사람의 이름과 급여를 출력

```
SQL> select ename, sal from emp  
2 where sal > 4000;
```

ENAME	SAL
KING	5000

- emp 테이블에서 이름이 scott인 사람의 이름과 사원번호, 급여를 출력

```
SQL> select ename, empno, sal from emp  
2 where ename = 'SCOTT';
```

ENAME	EMPNO	SAL
SCOTT	7788	3000

```
SQL> select ename, empno, sal from emp  
2 where ename = 'scott';
```

선택된 레코드가 없습니다.

- where 절에 조건으로 주어지는 것이 **문자**나 **날짜**인 경우에는 반드시 '조건'처럼 **출따**
옴표로 둘러싸야 함
- 문자의 경우는 대소문자를 구분

select – 데이터 조회하기

■ 날짜로 조회

```
SQL> select name, pay, hiredate from professor
2  where hiredate=23-MAY-01;
where hiredate=23-MAY-01
```

×

2행에 오류:

ORA-00904: "MAY": 부적합한 식별자

```
SQL> select name, pay, hiredate from professor
2  where hiredate = 01/05/23;
where hiredate = 01/05/23
```

×

2행에 오류:

ORA-00932: 일관성 없는 데이터 유형: DATE이(가) 필요하지만 NUMBER임

```
SQL> select name, pay, hiredate from professor
2  where hiredate = '01/05/23';
```

NAME	PAY	HIREDATE
-----	-----	-----
허은	290	01/05/23

- 날짜 부분을 홑따옴표로 묶어서 검색

```
select * from professor
where hiredate='2001-09-01';
```



select – 데이터 조회하기

- Where에서 연산자 활용
 - 비교연산자 : =, !=, <, >, <=, >=
 - 논리연산자 : and, or, not
 - 범위연산자 : between
 - 목록 연산자 : in
 - 특정 패턴 검색시 사용 : like

select – 데이터 조회하기

■ (9) 다양한 연산자 활용

연산자 종류	설명
=	같은 조건을 검색
!=, <>, ^=	같지 않은 조건을 검색
>	큰 조건을 검색
>=	크거나 같은 조건을 검색
<	작은 조건을 검색
<=	작거나 같은 조건을 검색
BETWEEN a AND b	A와 B 사이에 있는 범위 값을 모두 검색
IN(a,b,c)	A이거나 B이거나 C인 조건을 검색
LIKE	특정 패턴을 가지고 있는 조건을 검색
IS NULL/ IS NOT NULL	null 값을 검색/null 이 아닌 값을 검색
A AND B	A 조건과 B 조건을 모두 만족하는 값만 검색
A OR B	A 조건이나 B 조건 중 한 가지라도 만족하는 값을 검색
NOT A	A 가 아닌 모든 조건을 검색

select – 데이터 조회하기

- 비교 연산자를 사용하여 student 테이블에서 키(height)가 180cm 보다 크거나 같은 사람 출력

```
SQL> select name, height from student
2 where height >= 180;
```

NAME	HEIGHT
서진수	180
박동호	182
임지매	182
노정호	184

- Between 연산자를 사용하여 student 테이블에서 몸무게(weight)가 60~80kg 인 사람의 이름과 체중 출력

```
SQL> select name, weight from student
2 where weight between 60 and 80;
```

NAME	WEIGHT
서진수	72
서재수	64
박동호	70
임지매	72
김지욱	70
노정호	62
안은수	63
인영민	69

8 개의 행이 선택되었습니다.

- Between 연산자 : 두 개의 값 중에 작은 값이 먼저 오고 큰 값이 나중에 와야 함
- 두 개의 값을 다 포함하여 출력됨
- where weight >= 60 and weight <= 80 과 동일
 - => 성능면에서 더 좋다(between 연산자 사용하더라도 오라클 내부적으로 비교연산자로 변경해서 작업 수행하므로)

select – 데이터 조회하기

- In 연산자를 사용하여 student 테이블에서 101번 학과 학생과 102번 학과 학생들을 모두 출력

```
SQL> select name, deptno1 from student  
2 where deptno1 in(101, 102);
```

NAME	DEPTNO1
------	---------

서진수	101
서재수	102
김신영	101
김신지	102
김지매	101
김진욱	102
이윤나	101
김주현	102

8 개의 행이 선택되었습니다.

- where deptno1=101 or deptno1=102 과 동일
- 간편하게 in 연산자 사용

- Like 연산자를 사용하여 student 테이블에서 성이 "김"씨인 사람을 조회

```
SQL> select name from student  
2 where name like '김%';
```

NAME

김재수
김신영
김지매
김진욱
김주현

- like 와 함께 사용되는 기호 : % _
% : 글자수 제한 없고 어떤 글자가 와도 됨
_ : 글자수는 한 글자만 올 수 있고, 어떤 글자가 와도 됨

select – 데이터 조회하기

- 전체 이름 글자수가 3글자인데, 성은 '김' 이고 마지막 글자가 '호'인 사람 조회
 - where name like '김_호';
- 검색하고자 하는 문자에 '%' 나 '_' 가 있을 경우
 - Escape 문자로 표시해주어야 함 (hr 스키마에서 실행할 것)
select employee_id, last_name, job_id from employees
where job_id like '%SAW_%' ESCAPE 'W';

```
select employee_id, last_name, job_id from  
employees where job_id like '%PR*_%' ESCAPE '*';
```

```
SQL> select employee_id, last_name, job_id from employees  
2 where job_id like '%SAW_%' ESCAPE 'W';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID
145	Russell	SA_MAN
146	Partners	SA_MAN
147	Errazuriz	SA_MAN
148	Cambrault	SA_MAN
149	Zlotkey	SA_MAN
150	Tucker	SA_REP
151	Bernstein	SA_REP
152	Hall	SA_REP
153	Olsen	SA_REP
154	Cambrault	SA_REP
155	Tuvault	SA_REP
156	King	SA_REP
157	Sully	SA_REP
158	McEwen	SA_REP
159	Smith	SA_REP



select – 데이터 조회하기

- (1) is null/ is not null 연산자 활용
 - null : 오라클의 데이터 종류 중 한 가지로 어떤 값인지 모른다는 의미
 - 데이터가 없음을 의미함, 아직 정의되지 않은 미지의 값
 - 예) A라는 사람의 나이가 30살이고, B의 나이를 모른다면 'B의 나이를 null 이다' 라고 함
 - null 에는 어떤 연산을 수행해도 결과값은 항상 null 이 나옴
 - null 값은 '=' 연산을 사용할 수 없음
 - null값을 '=' 조건을 주고 검색하면 no rows selected. 라고 나오면서 값을 조회하지 못함
 - null 값을 찾고 싶을 땐 '=' 연산자를 사용할 수 없어서 'is null', null 값이 아닌 모든 값을 찾고 싶을 때는 is Not null을 사용

select – 데이터 조회하기

null 값만 출력

```
SQL> select name, bonus from professor  
2 where bonus is null;
```

NAME	BONUS
------	-------

송도권	
양선희	
김현정	
최슬기	
차범철	
전민	

6 개의 행이 선택되었습니다.

null 이 아닌 값만 출력

```
SQL> select name, bonus from professor  
2 where bonus is not null;
```

NAME	BONUS
------	-------

조이형	100
박승근	60
김영조	80
주승재	90
김도형	110
나한열	50
심스	130
박원범	50
바비	80
허은	30

10 개의 행이 선택되었습니다.

null

- 아직 정의되지 않은 미지의 값
- 데이터가 없음을 의미
- space(ASCII 32)나 0(ASCII 48)과는 다름
- null을 포함한 연산이나 비교는 의미가 없음
- 오라클의 정렬작업에서는 가장 큰 값으로 취급됨

select – 데이터 조회하기

- (2) 검색 조건이 2개 이상일 경우 조회
 - student 테이블을 사용하여 4학년 중에서 키가 170cm 이상인 사람의 이름과 학년, 키를 조회

```
SQL> select name, grade, height from student  
2 where grade=4 and height>=170;
```

NAME	GRADE	HEIGHT
서진수	4	180
서재수	4	172
김재수	4	177
박동호	4	182

select – 데이터 조회하기

- student 테이블을 사용하여 1학년이거나 또는 몸무게가 80kg 이상인 학생들의 이름과 학년, 키, 몸무게를 조회

```
SQL> select name, grade, height, weight from student  
2 where grade=1 or weight>=80;
```

NAME	GRADE	HEIGHT	WEIGHT
김재수	4	177	83
이관훈	2	175	82
이준나	1	162	48
이수미	1	175	63
이영호	1	173	69
박주현	1	179	81
여우	1	163	51

7 개의 행이 선택되었습니다.

- student 테이블을 사용하여 2학년 중에서 키가 180cm 보다 크면서 몸무게가 70kg 보다 큰 학생들의 이름과 학년, 키와 몸무게를 조회

```
SQL> select name, grade, height, weight from student  
2 where grade=2 and height>180 and weight>70;
```

NAME	GRADE	HEIGHT	WEIGHT
일지매	2	182	72

select – 데이터 조회하기

- student 테이블을 사용하여 2학년 학생 중에서 키가 180cm 보다 크거나 또는 몸무게가 70kg 보다 큰 학생들의 이름과 학년, 키, 몸무게를 조회

```
SQL> select name, grade, height, weight from student  
2 where grade=2 and (height>180 or weight>70);
```

NAME	GRADE	HEIGHT	WEIGHT
일지매	2	182	72
안광훈	2	175	82
노정호	2	184	62

and 와 or 조건이 동시에 나올 경우
우선순위 주의

- 실습> professor 테이블에서 교수들의 이름을
조회하여 성 부분에 '전'이 포함된 사람의 명단을
출력

NAME
전민
조인형
주승재

select - 데이터 조회하기

- (10) order by 절을 사용하여 출력 결과 정렬하기
 - 오름차순 정렬(기본값) asc, 내림차순 정렬 desc
 - SQL 문장의 가장 마지막에 적어야 함
- 주의사항
 - order by 구문은 SQL에서 최대한 피해야 하는 명령어 중 하나.
 - 데이터가 많은 경우 정렬을 한다는 것은 DBMS에게 큰 부담을 주는 명령어이기 때문

```
SQL> select name from professor
2 order by 1;
```

NAME

형조정열 권범권 회
도영현한비승원도스서미
김김김나바박송심양전

NAME

100

형재철기
이승범스튜디오
조주차초허

16 개의 행이 선택되었습니다.

select – 데이터 조회하기

- student 테이블을 사용하여 1학년 학생의 이름과 키를 출력. 단, **키가 작은 순서대로** 출력

```
SQL> select name, height from student
2  where grade=1;
```

NAME	HEIGHT
이민아	162
이민수	175
이민미	173
이민현	179
이민준	163

```
SQL> select name, height from student
2  where grade=1 order by height;
```

NAME	HEIGHT
이민아	162
이민준	163
이민미	173
이민수	175
이민현	179

- student 테이블을 사용하여 1학년 학생의 이름과 키, 몸무게를 출력. 단, **키는 작은 순서대로 출력하고 몸무게는 많은 사람부터** 출력

```
SQL> select name, height, weight from student
2  where grade = 1
3  order by height asc, weight desc;
```

NAME	HEIGHT	WEIGHT
이민아	162	48
이민준	163	51
이민미	173	69
이민수	175	63
이민현	179	81

- 두 개의 칼럼에 동시에 정렬이 수행되면, 우선 앞의 칼럼 기준으로 먼저 정렬한 후 뒤 칼럼의 정렬 수행
- 즉, 앞 칼럼에 같은 키가 있을 경우, 뒤의 칼럼에서 몸무게가 무거운 사람이 먼저 나옴

select – 데이터 조회하기

- student 테이블을 사용하여 1학년 학생의 이름과 생일, 키, 몸무게를 출력. 단, **생일이 빠른 사람 순서대로** 정렬

```
SQL> select name, birthday, height, weight from student
2  where grade = 1
3  order by 2;
```

NAME	BIRTHDAY	HEIGHT	WEIGHT
이수	78/01/05	175	63
이주현	78/02/23	163	51
김민	78/03/24	179	81
이민	78/08/09	173	69
이민	78/08/19	162	48

```
SQL> select name, birthday, height, weight from student
2  where grade = 1
3  order by birthday;
```

- **order by** 절에 칼럼 이름이 아닌 숫자 2 가 나왔음
- **select** 절에서 2번째 칼럼인 **birthday** 칼럼으로 정렬하라는 의미

- student 테이블을 사용하여 1학년 학생의 이름과 키를 출력. 단, **이름을 오름차순으로** 정렬

```
SQL> select name "이름" , height "키" from student
2  where grade = 1
3  order by 이름;
```

이름	키
김민	179
이수	175
이민	162
이민	173
이민	163

- 칼럼의 별명을 사용한 정렬



실습

- employees 테이블에서 사원아이디, 이름 - 성(예 : Steven-King), 입사일, 기본급(salary),
- 수당(salary*commission_pct), 급여(salary+수당) 조회하기
 - 모든 컬럼은 별칭을 사용한다



select – 데이터 조회하기

- (11) 집합 연산자(Set Operator)
 - 집합 – 여러 건의 데이터가 모여 있다는 뜻
 - 지금까지의 산술 연산자들 – 하나의 데이터들끼리 연산하는 것
 - 집합 연산자 – 여러 건의 데이터들(집합)을 연산하는 것
- 집합 연산자 종류

연산자 종류	내용
UNION	두 집합을 더해서 결과를 출력. 중복 값 제거하고 정렬함
UNION ALL	두 집합을 더해서 결과를 출력. 중복 값 제거 안하고 정렬 안 함
INTERSECT	두 집합의 교집합 결과를 출력. 정렬함
MINUS	두 집합의 차집합 결과를 출력. 정렬함. 쿼리의 순서 중요함



select – 데이터 조회하기

- 집합 연산자 사용할 경우 주의사항
 - 1) 두 집합의 SELECT 절에 오는 칼럼의 개수가 동일해야 함
 - 2) 두 집합의 SELECT 절에 오는 칼럼의 데이터형이 동일해야 함
 - 3) 두 집합의 칼럼명은 달라도 상관없음

select – 데이터 조회하기

- 1) Union/ Union All (두 집합을 더함)
- student 테이블과 professor 테이블을 참조하여 101번 학과에 소속되어 있는 학생과 교수들의 학번(교수는 교수번호), 이름, 학과 번호를 출력

```
SQL> SELECT studno, name, deptno1 from student
2  where deptno1=101
3  union
4  SELECT profno, name, deptno from professor
5  where deptno=101;
```

STUDNO	NAME	DEPTN01
1001	조인형	101
1002	박승곤	101
1003	송도권	101
9411	서진수	101
9511	김신영	101
9611	일지매	101
9711	이윤나	101

7 개의 행이 선택되었습니다.

```
SQL> SELECT studno, name, deptno1 from student
2  where deptno1=101
3  union all
4  SELECT profno, name, deptno from professor
5  where deptno=101;
```

STUDNO	NAME	DEPTN01
9411	서진수	101
9511	김신영	101
9611	일지매	101
9711	이윤나	101
1001	조인형	101
1002	박승곤	101
1003	송도권	101

7 개의 행이 선택되었습니다.

• union : 정렬해줌, union all : 정렬 안함

select – 데이터 조회하기

```
SQL> conn scott/tiger
```

연결되었습니다.

```
SQL> select id1, name1 from set1
2  union
3  select id2, name2 from set2;
```

ID1	NAME1
-----	-------

1	AAA
2	BBB
3	CCC

```
SQL> select id1, name1 from set1
2  union all
3  select id2, name2 from set2;
```

ID1	NAME1
-----	-------

1	AAA
1	AAA
2	BBB
2	BBB
3	CCC
3	CCC

```
create table set1(
  id1 number,
  name1 char(10)
);
```

```
create table set2(
  id2 number,
  name2 char(10),
  age number
);
```

```
insert into set1(id1, name1) values(1, 'AAA');
insert into set1(id1, name1) values(1, 'AAA');
insert into set1(id1, name1) values(2, 'BBB');
```

```
insert into set2(id2, name2, age) values(2, 'BBB',20);
insert into set2(id2, name2, age) values(3, 'CCC',15);
insert into set2(id2, name2, age) values(3, 'CCC',23);
```

```
commit;
```

select – 데이터 조회하기

- student 테이블에서 101번 학과와 201번 학과를 전공하는 학생들의 이름을 출력

```
SQL> SELECT name from student
2  where deptno1=101
3  union
4  SELECT name from student
5  where deptno2=201;
```

NAME

김신영
서진수
신은경
이윤나
일지매

```
SQL> SELECT name from student
2  where deptno1=101
3  union all
4  SELECT name from student
5  where deptno2=201;
```

NAME

서진수
김신영
일지매
이윤나
서진수
신은경

6 개의 행이 선택되었습니다.

- union : 중복되는 서진수 학생의 이름이 1회만 나옴
- union all : 서진수 학생이 2번 나옴 (복수 전공자)

```
Select name from student
where deptno1=101 or deptno2=201;
```

Union 과 동일

select – 데이터 조회하기

```
SQL> SELECT name, deptno1, deptno2 from student where deptno1=101;
```

NAME	DEPTN01	DEPTN02
서진수	101	201
김신영	101	
일지매	101	
이윤나	101	

```
SQL> SELECT name, deptno1, deptno2 from student where deptno2=201;
```

NAME	DEPTN01	DEPTN02
서진수	101	201
신은경	102	201

select – 데이터 조회하기

- 2) Intersect 연산자 사용하기
 - 두 집합의 교집합 부분을 찾아내는 집합 연산자
- student 테이블을 사용하여 101번 학과와 201번 학과를 복수 전공하는 학생들의 이름을 출력

```
SQL> SELECT name from student where deptno1=101  
2 intersect  
3 SELECT name from student where deptno2=201;
```

NAME

서진수

- deptno1 : 1전공
- deptno2 : 2전공
(복수 전공자를 구분하기 위해서 사용)

```
SQL> SELECT name from student where deptno1=101 and deptno2=201;
```

NAME

서진수

select - 데이터 조회하기

```
SQL> select id1, name1 from set1
2 intersect
3 select id2, name2 from set2;
```

ID1	NAME1
-----	-------

2	BBB
---	-----

```
SQL> select id1, name1 from set1
2 minus
3 select id2, name2 from set2;
```

ID1	NAME1
-----	-------

1	AAA
---	-----

```
SQL>
SQL> select id2, name2 from set2
2 minus
3 select id1, name1 from set1;
```

ID2	NAME2
-----	-------

3	CCC
---	-----

select – 데이터 조회하기

3) Minus 연산자 사용하기

- 큰 집합에서 작은 집합을 빼는 집합 연산자
- professor 테이블에서 전체 직원의 급여를 20% 인상하기 위한 직원 명단을 출력. 단, 직급이 전임강사인 사람들은 명단에서 제외 할 것

• minus 연산자도 결과값을 정렬함
⇒데이터 량이 많을 경우 시간이 오래 걸림
• minus 연산자 – 어떤 쿼리를 먼저 쓰는가가 아주 중요함

```
SQL> select name, position from professor
2 order by 2;
```

NAME	POSITION
전민	전임강사
양서희	전임강사
김현도	전임강사
송노권	전임강사
차범철	전임강사
주승수	전임강사
심교수	전임강사
김이교수	전임강사
노영호	전임강사
김도원	전임강사
박원범	전임강사
최슬기	조교수
나현주	조교수
김민준	조교수
이승재	조교수
홍기	조교수
최민준	조교수
박근	조교수

16 개의 행이 선택되었습니다.

```
SQL> select name, position from professor
2 minus
3 select name, position from professor
4 where position = '전임강사';
```

NAME	POSITION
김도원	정교수
김현주	정교수
나현주	정교수
박승원	정교수
박원범	정교수
심교수	정교수
김이교수	정교수
노영호	정교수
김민준	정교수
이승재	정교수
홍기	정교수
최민준	정교수
박근	정교수

11 개의 행이 선택되었습니다.

```
SQL> select name, position from professor where position != '전임강사';
```


select – 데이터 조회하기

- 집합 연산자를 사용할 때 두 개의 쿼리의 칼럼 개수가 다르거나 데이터 형이 다르면 에러 발생

```
SQL> SELECT studno, name from student
2 union
3 select profno from professor;
SELECT studno, name from student
*
```

1행에 오류:

ORA-01789: 질의 블록은 부정확한 수의 결과 열을 가지고 있습니다.

• select 절의 칼럼 개수가 다를 경우 발생하는 에러

```
SQL> SELECT studno, name from student
2 union
3 SELECT name, profno from professor;
SELECT studno, name from student
*
```

1행에 오류:

ORA-01790: 대응하는 식과 같은 데이터 유형이어야 합니다

• select 절의 칼럼의 데이터 타입이 다를 경우 발생하는 에러



실습

- product 테이블의 모든 컬럼 가져오기
- dept 테이블의 모든 컬럼 가져오기
- student 테이블에서 일부 컬럼만 가져오기
 - ID, NAME, BIRTHDAY
 - name 을 '학생 이름'으로 컬럼 제목 바꾸기



실습

- 1. professor 테이블의 모든 컬럼을 조회하는데, name 내림차순으로 조회하기
 - 조건 : position 이 '조교수' 인 것만 조회
- 2. department 테이블에서 deptno, dname, build 컬럼만 조회
 - 조건 : 학과(dname)에 '공학'이라는 단어가 들어간 학과만을 조회하기
 - 정렬 : dname 순으로 오름차순으로 정렬
- 3. emp2 테이블에서 name, emp_type, tel, pay, position 컬럼만 조회하되, position 컬럼은 컬럼제목을 '직위'로 나타내고
 - 조건 : pay가 3000만원에서 5000만원인 것들만 조회하기



실습

- 4. emp2 테이블에서 name, emp_type, tel, birthday 컬럼만 조회하되, 다음 조건에 맞는 데이터만 조회
 - 조건 : 생일(birthday)가 1980년도 인 것들만 조회하기(between 이용)
- 5. gift 테이블에서 모든 컬럼을 조회하되
 - 조건 : gname에 '세트'라는 단어가 들어간 레코드만 조회하기
- 6. emp2 테이블에서 name, position, hobby, birthday 컬럼을 조회하되
 - 조건 : position 이 null 이 아닌 것만 조회
 - 생일(birthday) 순으로 오름차순으로 정렬
- 7. emp2 테이블에서 모든 컬럼을 조회하되
 - 조건 – emp_type이 '정규직'이거나 '계약직'인 것만 조회(in 이용)
- 8. emp2 테이블에서 emp_type, position 컬럼을 조회하되
 - 중복된 행(레코드)은 제거