데이터

사실을 나타내는 수치 / 값

데이터베이스 DB

특정 조직의 업무 처리를 위해 시스템에서 사용되는 데이터의 집합

데이터베이스 관리 시스템 DBMS

데이터의 일관성, 무결성 등의 문제가 발생하지 않도록 관리 기능을 수행하는 소프트웨어

데이터베이스 시스템 DBS

정보를 저장, 관리하여 사용자에게 제공하는 프로그램의 총칭

개념적 데이터 모델링

* + 실세계의 대상과 관련성을 데이터 모델(표기법)을 사용하여 논리적으로 추상화하는 단계

논리적 데이터 모델링

* + 데이터베이스 관리 시스템이 데이터를 운용 및 관리하는 모델에 따른 데이터베이스의 스키마를 생성하는 단계

물리적 데이터 모델링

실제 데이터베이스 관리 시스템에 맞도록 정의된 개체들을 구현

추상적으로 기술된 모든 사항이 대부분 구현

* 모델링(사전적 의미)
  + 사물의 형태를 형상화하는 일
  + 모델을 보고 닮게 만드는 일. 아파트 모델 하우스
  + 데이터 모델 : 데이터의 관계, 접근과 그 흐름에 필요한 처리 과정에 관한 추상화된 모형
  + 데이터 모델링 : 데이터 개체들을 분석하고, 개체들 사이의 연관성을 정의 하는 작업

데이터 정의 언어(DDL)

* + 데이터베이스 객체를 관리하는 SQL 명령어 집합
  + CREATE 생성, ALTER 수정, DROP 삭제

데이터 조작 언어(DML)

* + 데이터를 조작하는 SQL 명령어 집합
  + INSERT 삽입, UPDATE 수정, DELETE 삭제, SELECT 검색

데이터 제어 언어(DCL)

* + 사용자 권한을 통제하는 SQL 명령어 집합
  + GRANT 부여, REVOKE 회수
* DISTINCT로 열 중복 제거하기 (열이 한 개인 경우)

select deptno from emp ;

select distinct deptno from emp ;

정렬

select \* from emp order by sal asc ;

select \* from emp order by sal desc ;

* 직무가 job 가 CLERK 이고 부서 deptno 가 10 또는 30인 사원 정보

select \* from emp

where job = 'CLERK' **and** deptno = 10 **or** deptno = 30 ;

* 직무 job 이 SALESMAN 이 아닌 사원 출력

select \* from emp

where **not** job = 'SALESMAN' ;

select \* from emp where job != 'SALESMAN' ;

* 직무 job 이 CLERK 이거나 MANAGER 인 사원 출력

select \* from emp

where job = 'CLERK' **or** job = 'MANAGER' ;

select \* from emp

where job **in** ('CLERK' , 'MANAGER' ) ;

* 사원이름 ename 에 AM 이 포함되어 있는 사원 출력

select \* from emp

where ename **like** '%AM%' ;

* 직속상관 mgr 이 있는 사원 출력

select \* from emp

where mgr **is not null** ;

* 집합 연산자
  + UNION : 중복이 제거되는 합집합
  + UNION ALL : 중복을 허용하는 합집합
  + MINUS : 차집함
  + INTERSECT : 교집합
* select empno , ename from emp where deptno = 10
* **union**
* select empno , ename from emp where sal > 3000 ;
* select empno , ename from emp where deptno = 10
* **union all**
* select empno , ename from emp where sal > 3000 ;
* select empno , ename from emp where deptno = 10
* **minus**
* select empno , ename from emp where sal > 3000 ;
* select empno , ename from emp where deptno = 10
* **intersect**
* select empno , ename from emp where sal > 3000 ;
* UPPER(문자열) : 문자 데이터를 모두 대문자로 변환하여 반환
* LOWER(문자열) : 문자 데이터를 모두 소문자로 변환하여 반환
* INITCAP(문자열) : 첫 글자는 대문자로, 나머지 문자는 소문자로 변환 후 반환

select \* from emp

where **upper(ename) = upper('Scott')** ;

* LENGTH(문자열) : 문자열 길이

select ename , **length(ename)** from emp

where **length(ename) >= 6** ;

* LENGTHB(문자열) : 문자열의 바이트 수

select length('hello') , **lengthb('hello')** from dual ;

select length('안녕') , **lengthb('안녕')** from dual ;

select length('안녕하세요') , **lengthb('안녕하세요')** from dual ;

* 시스템 설정에 따라 한글은 2byte 혹은 3byte
* dual : 오라클 더미 테이블, 데이터 자장 공간이 아닌 임시 용도로 사용
* SUBSTR : 문자열 일부 추출

select job ,

**substr( job , 1 , 2 ) ,** // 첫번째부터 2개

**substr( job , 2 , 3 ) ,** // 두번째부터 3개

**substr( job , 4 ),** // 4번째부터 끝까지

**substr( job , -3 )** // 마지막 3개

from emp ;

* INSTR : 문자열 데이터 내 특정 문자 위치 찾기
  + instr ( [ 대상 문자열 데이터 ] ,  
     [ 위치를 찾으려는 부분 문자 ] ,  
     [ 위치 찾기를 시작할 대상 문자열 데이터 위치 ] ,  
     [ 시작 위치에서 찾으려는 문자가 몇 번째인지 지정] )

select **instr( 'HELLO, ORACLE!' , 'L' , 2 , 2 )**

from dual ;

* REPLACE : 특정 문자를 다른 문자로 대체
  + replace ( [ 문자열 데이터 또는 열 이름 ] ,  
     [ 찾는 문자 ] ,  
     [ 대체할 문자 ] )

select

**replace( '010-123-4567' , '-' , '/' ) ,**

**replace( '010-123-4567' , '-' , ' ' ) ,**

from dual ;

* LPAD : 남은 빈 공간을 왼쪽에 채우기
  + lpad ( [ 문자열 데이터 또는 열이름 ] ,  
     [ 데이터의 자릿수 ] ,  
     [ 빈 공강에 채울 문자 ] )

select 'Oracle' ,

**lpad( 'Oracle' , 10 , '#' ) , lpad( 'Oracle' , 10 )**

from dual ;

* RPAD : 남은 빈 공간을 오른쪽에 채우기
  + rpad ( [ 문자열 데이터 또는 열이름 ] ,  
     [ 데이터의 자릿수 ] ,  
     [ 빈 공강에 채울 문자 ] )

select 'Oracle' ,

**rpad( 'Oracle' , 10 , '#' ) , rpad( 'Oracle' , 10 )**

from dual ;

* CONCAT : 두 문자열 데이터를 합치기

select

**concat( empno , ename ) ,**

**concat( empno , concat( ':' , ename ) )**

from emp ;

* || : 문자열 연결 연산자

select

**empno || ename ,**

**empno || ':' || ename**

from emp ;

* ROUND : 반올림
* round ( [ 숫자 ] , [ 자릿수 ] )

select

round( 1234.5678 ) , round( 1234.5678 , 0 ) ,

**round( 1234.5678 , 1 ) , round( 1234.5678 , 2 )** ,

// 양수 자릿수 : 소수점 아래 표현 가능 수

**round( 1234.5678 , -1 ) , round( 1234.5678 , -2 )**

// 음수 자릿수 : 소수점 위 표현 하지 않는 수

from dual ;

* TRUNC : 버림
* trunc ( [ 숫자 ] , [ 자릿수 ] )

select

trunc( 1234.5678 ) , trunc( 1234.5678 , 0 ) ,

**trunc( 1234.5678 , 1 ) , trunc( 1234.5678 , 2 )** ,

// 양수 자릿수 : 소수점 아래 표현 가능 수

**trunc( 1234.5678 , -1 ) , trunc( 1234.5678 , -2 )**

// 음수 자릿수 : 소수점 위 표현 하지 않는 수

from dual ;

* CEIL : 지정된 숫자와 가장 가까운 큰 정수

select **ceil( 3.14 ) , ceil( -3.14 )** from dual ;

* FLOOR : 지정된 숫자와 가장 가까운 작은 정수

select **floor( 3.14 ) , floor( -3.14 )** from dual ;

* MOD : 숫자를 나눈 나머지

select **mod( 15 , 2 ) , mod( 15 , 3 ) , mod( 15 , 4 )**

from dual ;