## Day01_SELECT문

1) SELECT문 작성법

```
SELECT [DISTINCT] {*, column [Alias], ....}
FROM 데이블명
[WHERE condition]
[ORDER BY {column, expression} [ASC|DESC]] ;
```

2) SQL 문장 실행

- SELECT * FROM TAB; : scott이 소유하고 있는 TABLE을 전부 보여줌
- SELECT * FROM DEPT; : DEPT테이블 데이터 전체 보기
- DESC DEPT; : DEPT테이블의 구조보기
  - NOT NULL : NULL값이 있으면 안됨
  - NUMBER(4) : 부서코드는 정수 4자리
  - NUMBER(7,2) : 총자리수 7자리 중 소수점 이하는 2자리

3) 비교연산자 : 같다(=), 크거나같다(>=) 작거나같다(<=) 다르다(<> ^= !=)

- <>, !=, ^= : 모두 다르다를 의미
- SQL에서 문자열이나 날짜는 반드시 작은따음표안에 표시해야함

4) 논리연산자 : AND, OR, NOT

5) 산술표현식

- NVL(expr1, expe2)
  - expr1 : NULL값을 포함하고 있는 COLUMN이나 표현식
  - expr2 : NULL변환을 위한 목표값
  - expr1과 expr2는 같은 타입이여야 함

6) 연결연산자(||) : 열이나 문자를 다른 열에 연결하는 연산

7) SQL 연산자

- DISTINCT 필드 : 필드 중복 제거
- BETWEEN a AND b : a부터 b(a,b값 포함, a가 꼭 작은 값이여야함)
- IN(list) : list 값 중 어느 하나와 일치한다.
- LIKE pattern : 검색하고자 하는 값을 정확히 모를 경우 pattern과 일지하는지를 검색, pattern에는 다음 두가지 와일드카드 사용
  - % : 문자가 없거나, 하나 이상의 문자가 어떤 값이 와도 상관없음
  - _ : 하나의 문자가 어떤 값이 와도 상관없음
- IS NULL : NULL 값을 가짐
- NOT BETWEEN a AND b : a와 b사이에 있지 않다.(a,b값 포함X)
- NOT IN(list) : list의 값과 일치 X

- NOT LIKE pattern : 문자형태가 pattern과 일치 X
- NOT IS NULL : NULL값을 가지지 X

8) ORDER BY 절
- ASC 오름차순
- DESC 내림차순

| | ASC(오름차순) | DESC(내림차순) |
|---|---|---|
| 숫자 | 작은 값부터 정렬 | 큰 값부터 정렬 |
| 문자 | 사전 순서로 정렬 | 사전 반대 순서로 정렬 |
| 날짜 | 빠른 날짜 순서로 정렬 | 늦은 날짜 순서로 정렬 |

## Day02_JOIN

1) JOIN : 즉, 두개 이상의 테이블에 대해서 결합하여 나타낼 때 사용
   - Equi Join : 동일한 컬럼을 기준으로 조인
   - Non-Equi Join : 동일한 컬럼없이 다른 조건을 사용하여 조인
   - Outer Join : 조인 조건에 만족하지 않는 행도 나타나는 조인
   - Self Join : 한 테이블 내에서 조인.

2) CROSS JOIN : 특별한 키워드 없이 다음과 같이 FROM절에 두 개 이상의 테이블을 기술하는 것

3) EQUI JOIN : 가장 많이 사용하는 조인 방법으로서 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치되는 행을 연결하여 결과를 생성하는 조인 방법

4) NON-EQUI JOIN : 조인 조건에 특정 범위 내에 있는지를 조사하기 위해서 WHERE 절에 조인 조건을 = 연산자 이외의 비교연산자를 사용

5) SEIF JOIN : Edui Join과 같으나 하나의 테이블에 조인이 일어나는 것이 다름. 같은 테이블에 대해 두 개의 Alias를 작성함으로서 from 절에 두 개의 테이블을 사용하는 것과 같이함

6) OUTER JOIN
- 조인 조건에 만족하지 못하였더라도 해당 로우를 나타내고 싶을 때에 사용
- NULL 값이기에 배제된 행을 결과에 포함시킬 수 있으며 다음과 같이 "(+)" 기호를 조인 조건에서 정보가 부족한 칼럼 이름 뒤에 덧붙임

## Day03_단일행함수

1) 단일행함수(INPUT1 → OUTPUT1) : 오직 단일 행에서만 적용 가능, 행 별로 하나의 결과를 리턴(문자, 숫자, 날짜 형 변환 함수 등)

2) 숫자 함수 : 숫자를 처리하는 함수

1. ABS(x) : 절대값
2. COS(x) : COSINE 값을 반환
3. EXP(x) : e(2.71828183……)의 x승
4. FLOOR(x) : 소수점 아래를 버림
5. LOG(x) : log값을 반환
6. POWER(m,n) : m의 n승
7. SIGN(n) : n<0이면 -1, n=0이면 0, n>0이면 1을 반환
8. SIN(x) : SINE값 반환
9. TAN(x) : tangent값 반환
10. ROUND(x) : 특정 자릿수에서 반올림
11. ROUND(데이터, 반올림할 소수점 자리수) : 반올림할 자리수가 음수인 경우 -1은 십단위. -2는 백단위
12. TRUNC(x, n):n으로 지정한 자리수 이하를 버림
13. MOD(x, n) : x를 n으로 나눈 나머지값

3) 문자처리함수

1. LOWER(str) : 소문자로
2. UPPER(str) : 대문자로
3. INITCAP(str) : 첫 글자만 대문자로 나머지 글자는 소문자로
4. CONCAT(str1, str2) : 문자 연결
5. SUBSTR(str, 시작할위치, 추출할갯수) : 문자를 잘라 추출 (한글은 1byte), 시작할 위치는 인덱스 아님, 시작할 위치가 음수이면 끝에서부터 자리수를 센다.
6. SUBSTRB(str, 시작할위치,추출할갯수) : 문자를 잘라 추출 (한글은 2byte)
7. LENGTH(str) : 문자길이 (한글은 1byte)
8. LENGTHB(str) : 문자길이 (한글은 2~3byte)
9. INSTR(str, 찾을글자, 시작위치, 몇번째발견 검색된 횟수) str에서 찾을글자를 몇번째 발견하는지
10. LPAD, RPAD : 입력 받은 문자열과 기호를 정렬하여 특정 길이의 문자열로 반환한다
11. TRIM : 잘라내고 남은 문자를 표시한다
12. CONVERT : CHAR SET을 변환 SELECT CONVERT(ENAME,'US7ASCII','UTF8') FROM EMP;
13. CHR : ASCII코드 값으로 SELECT CHR(65) FROM DUAL;
14. ASCII : ASCII코드값을 문자로
15. REPLACE : 문자열에서 특정문자를 변경

4) 날짜함수
1. SYSDATE : 시스템 저장된 현재 날짜를 반환.
2. MONTHS_BETWEEN : 두 날짜 사이가 몇 개월인지를 반환.
3. ADD_MONTHS : 특정 날짜에 개월 수를 더함.
4. NEXT_DAY : 특정 날짜에서 최초로 도래하는 인자로 받은 요일의 날짜를 반환.
5. LAST_DAY : 해당 달의 마지막 날짜를 반환.
6. ROUND : 인자로 받은 날짜를 특정 기준으로 반올림.
7. TRUNC : 인자로 받은 날짜를 특정 기준으로 버림.

5) 형변환함수 : 숫자, 문자, 날짜의 데이터 형을 다른 데이터형으로 변환해야 하는 경우에 사용
1. TO_CHAR : 날짜형 혹은 숫자형을 문자형으로 변환
- TO_CHAR(날짜데이터, '출력형식');
- TO_CHAT(숫자데이터, '출력형식');

2. TO_DATE : 문자형을 날짜형으로 변환
- TO_DATE('문자', 'FORMAT');

3. TO_NUMBER : 문자형을 숫자형으로 변환

6) NVL 함수 : NULL을 다른 값으로 변환

7) DECODE 함수 : 프로그램 언어에서 가장 많이 사용되는 switch case 문과 같은 기능
- 형식 : DECODE (*표현식, 조건1, 결과1, 조건2, 결과2, 조건3, 결과3, 기본결과n* )

## Day03_그룹 함수

1) 그룹 함수 : 여러 행 또는 테이블 전체의 행에 대해 함수가 적용되어 하나의 결과 값을 가져오는 함수
- GROUP BY절을 이용하여 그룹 당 하나의 결과가 주어지도록 그룹화 할 수 있음
- HAVING절을 사용하여 그룹 함수를 가지고 조건 비교를 할 수 있음
- COUNT(*)를 제외한 모든 그룹함수는 NULL값을 고려하지 않음
- MIN,MAX 그룹함수는 모든 자료형에 대해서 사용할 수 있음

2) 그룹 함수의 종류
1. SUM : 그룹의 누적 합계
2. AVG : 그룹의 평균
3. COUNT : 그룹의 총 개수
4. MAX : 그룹의 최대값
5. MIN : 그룹의 최소값
6. STDDEV : 그룹의 표준편차
7. VARIANCE : 그룹의 분산

3) GROUP BY 절 : 어떤 컬럼 값을 기준으로 그룹함수를 적용할 경우 GROUP BY 절 뒤에 해당 컬럼을 기술, GROUP BY 절 다음에는 칼럼의 별칭을 사용할 수 없고, 반드시 칼럼명을 기술

```
SELECT 컬럼명, 그룹함수
        FROM 테이블명
          WHERE 조건(연산자)
          GROUP BY 컬럼명;
```

4) HAVING 조건

```
SELECT column, group_function
FROM table
[WHERE condition]
[GROUP BY group_by_expression]
[HAVING    group_condition]
[ORDER BY column] ;
```

1. SELECT 절에 조건을 사용하여 결과를 제한할 때는 WHERE 절을 사용하지만, 그룹의 결과를 제한할 때에는 HAVING 절을 사용.
2. HAVING 절은 GROUP BY 절 앞에 기술 가능하지만 GROUP BY 절 다음에 기술하는 것이 논리적으로 권장. HAVING 절이 SELECT 절에 있는 그룹에 적용되기 전에 그룹이 구성.

## Day04_SUB QUERY

1) 서브 쿼리 : 하나의 SQL 문장의 절 안에 포함된 또 하나의 SELECT 문장
- 서브 쿼리를 포함하고 있는 쿼리문을 메인 쿼리, 포함된 또 하나의 쿼리를 서브 쿼리
- 비교 연산자의 오른쪽에 기술하고 반드시 괄호로 둘러 쌓아야 함
- 메인 쿼리가 실행되기 이전에 한번만 실행

2) 단일행 서브쿼리 : 서브쿼리의 결과가 단일 행
- 단일 행 연산자 : =, >, >=, <, <=, <>

3) 다중행 서브쿼리 : 서브쿼리의 결과가 2행 이상의 행
복수 행 연산자 : IN, NOT IN, ANY, SOME, ALL, EXISTS
- IN : 메인 쿼리의 비교 조건('=' 연산자로 비교할 경우)이 서브 쿼리의 결과 중에서 하나라도 일치하면 참
- ANY, SOME : 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상이 일치하면 참
- ALL : 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 모든 값이 일치하면 참
- EXISTS : 메인 쿼리의 비교 조건이 서브 쿼리의 결과 중에서 만족하는 값이 하나라도 존재하면 참

## Day05_DDL,DCL,DML

1) DML : Data Manipulation Language ; 데이터 검색, 수정 등
- INSERT : 데이터베이스 객체에 데이터를 입력
- DELETE : 데이터베이스 객체에 데이터를 삭제
- UPDATE : 기존에 존재하는 데이터베이스 객체 안의 데이터수정
- SELECT : 데이터베이스 객체로부터 데이터를 검색

2) DDL : Data Definition Language ; 데이터와 그 구조를 정의
- CREATE : 데이터 베이스 객체 생성

```
CREATE TABLE table_name
(column_name   data_type  expr, ...)
```

> CHAR(SIZE) : 고정 길이 문자 데이터. VARCHAR2와 동일한 형태의 자료를 저장할 수 있고, 입력된 자료의 길이와는 상관없이 정해진 길이만큼 저장 영역 차지. 최소 크기는 1

> VARCHAR2(size) ; Up to 2000 Bytes 가변 길이 문자 데이터. 실제 입력된 문자열의 길이만큼 저장 영역을 차지. 최대 크기는 명시해야 하며, 최소 크기는 1

> NUMBER ; Internal Number Format 최고 40자리까지의 숫자를 저장. 이때 소수점이나 부호는 길이에 포함 X

> NUMBER(w) ; W자리까지의 수치로 최대 38자리까지 가능. (38자리 유효 숫자)

> NUMBER(w, d) ; W는 전체 길이, d는 소수점 이하 자릿수. 소수점은 자릿수에 포함 X.

> DATE : BC 4712년 1월 1일~AD 4712년 12월 31일까지의 날짜

> 식별자 명명규칙 : 반드시 문자로 시작. A~Z까지 대소문자, 0~9까지의 숫자, 특수기호는 _,$,#만 포함 가능, 1~30글자까지 가능. 공백 허용 안함, 오라클에서 사용되는 예약어나 다른 객체명과 중복 불가

> 테이블의 구조만 복사하기

```
CREATE TABLE EMP06
AS
SELECT * FROM EMP WHERE 1=0;
```

- DROP : 데이터 베이스 객체를 삭제

- ALTER : 기존에 존재하는 데이터베이스 객체를 다시 정의
> ADD COLUMN 절을 사용하여 새로운 칼럼을 추가.

```
ALTER TABLE table_name
ADD (column_name data_type expr, ...);
```

> MODIFY COLUMN 절을 사용하여 기존 칼럼을 수정.

```
ALTER TABLE table_name
MODIFY (column_name data_type expr, ...);
```

> DROP COLUMN 절을 사용하여 기존 칼럼을 삭제

```
DROP TABLE table_name;
```

- TRUNCATE : 데이터베이스 객체 내용 삭제

```
TRUCATE TABLE table_name
```

- 테이블의 제약조건 : DBMS에 부적합한 데이터가 테이블에 삽입되는 것을 방지하기 위해 CONSTRAINT를 사용

- 데이터 무결성 제약조건의 종류
> NOT NULL : 이열은 NULL값을 포함하지 않음을 지정
> UNIQUE : 테이블의 모든 행에 대해 유일해야 하는 값을 가진 열 또는 열의 조합을 지정
> PRIMARY KEY : 유일하게 테이블의 각 행을 식별
> FOREIGN KEY : 열과 참조된 테이블의 열 사이의 외래키 관계를 적용하고 설정
> REFERENCES 예약어 사용: 참조 무결성 제약 조건 지정
> CHECK : 참이어야 하는 조건을 지정
> DEFAULT ; 제약조건은 아니나 쓸 수 있는 것 :

3) DCL : Data Control Language ; 데이터베이스 사용자의 권한 제어
- GRANT : 데이터 베이스 객체에 권한 부여
- REVOKE : 이미 부여된 데이터베이스 객체의 권한을 취소
- COMMIT : 트랜잭션 확정 (TCL)
- ROLLBACK : 트랜잭션 취소 (TCL)
- SAVEPOINT : 복귀지점 설정 (TCL)

## Day06_Sequence

1) Sequence : 순차적인 번호를 자동으로 생성하는 객체로 테이블과 독립적으로 생성 및 저장 가능

2) 특징

- 시퀀스에서 생성되는 번호는 유일, 기본 테이블에서 인조 Primary Key 생성시 주로 사용
- 여러 테이블에 의해 공유도 가능
- 시퀀스는 테이블과 관계없이 생성, 저장. 오라클 내부 루틴에 의해 발생 > 증가, 감소

```
CREATE SEQUENCE sequence_name
[INCREMENT BY n]  [START WITH n]
[MAXVALUE n] [MINVALUE n]
[CYCLE | NOCYCLE]
[CACHE n | NOCACHE];
```

3) 시퀀스 삭제& 수정

- 삭제 : DROP SEQUENCE sequence_name;
- 수정 : ALTER SEQUENCE sequence_name [INCREAMENT BY n] [START WITH n] [MAXVALUE n][MINVALUE n];

## Day0 7 _View

1) View : 행과 컬럼으로 구성된 가상 테이블.

- 이미 존재하고 있는 테이블에 제한적으로 접근하도록 하기 위함
- 물리적인 저장공간과 데이터를 가지지 않고 다른 테이블이나 뷰에서 파생된 논리적인 테이블
- 기본 테이블의 데이터가 변경되면 뷰에도 반영

2) View 장점

- 뷰를 이용한 기본 테이블의 액세스 제한을 통한 데이터에 대한 보안 기능 제공
- 기본 테이블에 영향을 주지 않을 수도 있음.
- 여러 개의 기본 테이블로 정의된 뷰가 하나의 테이블인 것처럼 인식
- 기본 테이블에 대한 복잡한 형태의 질의를 뷰로 정의하여 간단하게 표현 가능

```
CREATE [OR REPLACE] VIEW view_name
AS subquery
[WITH CHECK OPTION] -- INSERT나 UPDATE시 서브쿼리의 조건을 만족할 경우에 처리, DELETE가능
[WITH READ ONLY] -- 읽기 전용 뷰 생성
```

4) View 종류

| 단순뷰 | 복합뷰 |
| --- | --- |
| - 하나의 테이블로 구성한 뷰<br><br>- INSERT, DELETE, UPDATE와 같은 DML 명령문을 실행하여 기본 테이블의 데이터 조작 가능<br><br>- 함수나 그룹 데이터는 사용 가능<br><br>**CREATE OR REPLACE VIEW EMPv0<br>AS SELECT EMPNO, ENAME, JOB, DEPTNO FROM EMP;** | - 하나 이상의 기본 테이블로 구성한 뷰<br><br>- DML문을 제한적으로 사용<br><br>- 함수나 그룹데이터는 사용 가능<br><br>**CREATE OR REPLACE VIEW EMPv0<br>AS SELECT EMPNO, ENAME, JOB, DNAME FROM EMP E, DEPT D<br>WHERE E.DEPTNO=D.DEPTNO;** |

5) View 제한 조건

- 테이블에 NOT NULL로 만든 컬럼들이 뷰에 다 포함되어 있어야 함
- WITH READ ONLY옵션을 설정한 뷰는 갱신 불가
- WITH CHECK OPTION을 설정한 뷰는 뷰의 조건에 해당되는 데이터만 삽입, 삭제, 수정 O

## Day08_INDEX

1) INDEX : SQL 명령문의 처리 속도를 향상시키기 위해서 컬럼에 대해서 생성하는 오라클 객체

```
CREATE INDEX index_name
ON table_name (column_name);
```

| 인덱스를 사용해야 하는 경우 | 인덱스를 사용하지 말아야 하는 경우 |
| --- | --- |
| - 테이블의 행의 수가 많을 때<br><br>- WHERE 문에 해당 칼럼이 많이 사용될 때<br><br>- 검색 결과가 전체 데이터의 2~4% 정도 일 때<br><br>- JOIN에 자주 사용되는 칼럼이나 NULL을 포함하는 컬럼이 많은 경우 | - 테이블에 행의 수가 적을 때<br><br>- WHERE 문에 해당 컬럼이 자주 사용되지 않을 때<br><br>- 검색 결과가 전체 데이터의 10~15% 이상 일 때<br><br>- 입력 수정 삭제 등이 자주 일어 날 때 |

## Day09_Transaction

1) Transaction

- 데이터베이스의 Transaction : 데이터 처리의 한 단위
- 오라클의 Transaction : 오라클에서 발생하는 여러 개의 SQL 명령문들을 하나의 논리적인 작업 단위로 처리

2) COMMT

- 모든 작업들을 정상적으로 저리하겠다고 확정하는 명령어
- 트랜잭션(INSERT, UPDATE, DELETE)의 처리 과정을 데이터베이스에 모두 반영하기 위해서 변경된 내용을 모두 영구 저장
- 명령어를 수행하게 되면 하나의 트랜잭션 과정을 종료

3) ROLLBACK

- 작업 중 문제가 발생되어서 트랜잭션의 처리과정에서 발생한 변경사항을 취소하는 명령어
- 명령어를 수행하게 되면 하나의 트랜잭션 과정을 종료
- 트랜잭션(INSERT, UPDATE, DELETE) 으로 인한 하나의 묶음 처리가 시작되기 이전의 상태로 되돌림

## Day10_MYSQL

1) MYSQL 특징

- SQL에 기반을 둔 관계형 DBMS 중 하나
- Oracle, IBM, Infomix 등의 데이터베이스는 고가이지만 MySQL 은 무료(배포판)
- 리눅스, 유닉스, 윈도우 등 거의 모든 운영체제에서 사용가능
- 처리 속도가 상당히 빠르고 대용량의 데이터 처리 용이
- 설치방법이 쉽고 초보자도 익히기 쉬움.
- 보안성이 우수