

SELECT (DML 또는 DQL): 조회

 0_1_데이터베이스개요.pdf 535 kB

DB 개요

데이터 : 하나하나의 값들

정보 : 데이터들을 가지고 의미를 부여하는 것.

DataBase : 한 조직에 필요한 정보를 여러 응용시스템에서 공유할 수 있도록 논리적으로 연관된 데이터를 모으고 중복되는 데이터를 최소화하여 구조적으로 통합/저장해놓은 것.

- 공용 데이터
- 통합 데이터
- 저장 데이터
- 운영 데이터

특징

- 실시간 접근성
- 계속적인 변환
- 동시 공유
- 내용에 따른 참조

DBMS

DB에서 데이터 추출 . 조작, 정의 ,제어등을 할 수 있게 해주는 데이터베이스 전용 관리 프로그램.

- 데이터 추출 : 데이터 조회
- 데이터 삽입 , 수정 , 삭제 : 데이터 조작
- 데이터 정의의 구조를 정의 : 도서관 책장이 세로로 만들지 가로로 만들지 같은 느낌
- 데이터 제어 :

DBMS 사용 이점

- 데이터 독립화
- 데이터 중복 최소화 , 데이터 무결성 보장
- 데이터의 보안 향상
- 관리 편의성 향상.

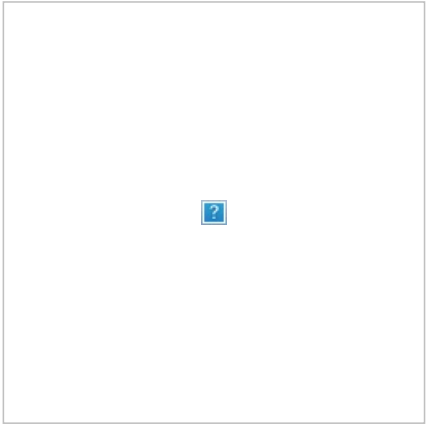
sys as sysdba : sys라는 계정이 관리자 권한으로 접속함.

DB를 편하게 사용하기 위해 Oracle이라는 DBMS를 사용하는데 이 DBMS를 사용하기 위한게 .SQLPLUS이다

CLI : 커맨드 라인

GUI : 그래픽

접속방법이다



일종의 자동 로그인기능.

SQL은 -- 이 주석의 역할을 한다
다만 범위 주석의 경우에는 /**/ 으로 이클립스와 동일하다.

 01_DML(SELECT).pdf

640 kB

- 1. 행
- 2. 컬럼
- 3. 기본키
- 4. 외래키
- 5. 값이 없음
- 6. 컬럼 값



오늘 배운 것 REMIND

-- SQL 실행 방법 : SQL문에 커서를 두거나, 블록처리 후 CTRL + ENTER

```
SELECT USERNAME FROM DBA_USERS;
```

-- 21C 버전은 일반 사용자를 구분할 때 ID 앞에 C##을 붙여야만 하는데

-- 이를 무시하고 11g와 호환되는 SQL문을 작성하게 하는 SQL을 수행.

```
ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE;
```

-생성한 사용자 계정 권한 부여

```
GRANT CONNECT, RESOURCE TO BDH;
```

--CONNECT : DB 연결 권한 ROLE

--RESOURCE : DB 기본 객체 생성 권한 ROLE

ROLE이란? 역할

- 객체(테이블 등)가 생성 될 수 있는 공간 할당량 지정/

```
ALTER USER bdh DEFAULT TABLESPACE SYSTEM QUOTA UNLIMITED ON SYSTEM
```



SQL 의 SELECT : 조회

데이터를 조회하면 조건에 맞는 행들이 조회된다
이렇게 조회된 행들의 집합은 **RESULT SET**이라고 한다.

RESULT SET

- 0개 이상의 행이 포함된다
- 조건에 맞는 행이 없을 수도 있기 때문에 0개 이상이다.

오라클에서 '*' 별은 ALL을 의미한다.



SELECT : 모든

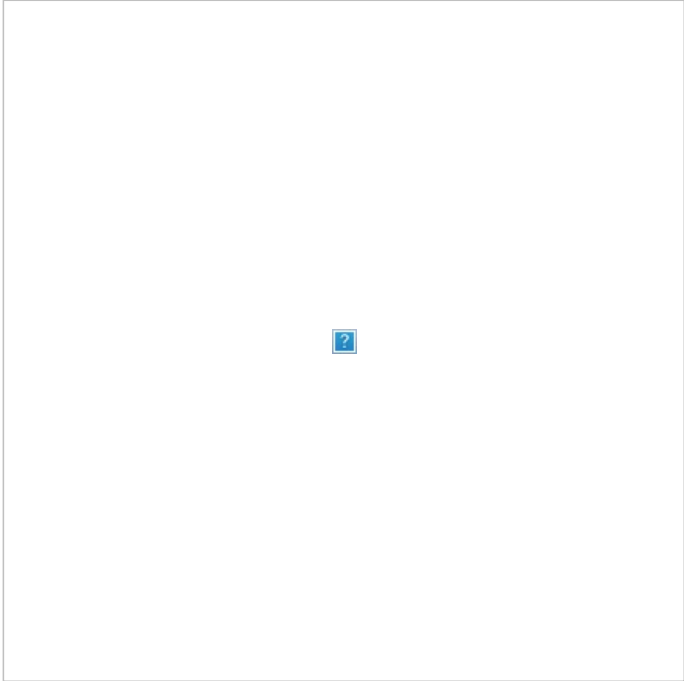
'*':ALL을 의미한다

FROM EMPLOYEE :EMPLOYEE라는 테이블에서
EMPLOYEE에 모든 정보를 조회한다.



- **EMPLOYEE**테이블에서 **EMP_NAME**이라는 칼럼을 조회한다.

하나의 칼럼을 조회하고 따옴표(,)를 이용해 여러 칼럼을 조회 할 수 있다.



위에 **SELECT**문에서는 **EMP_NO / EMP_NAME / EMAIL / HIRE_DATE**를 조회한다

칼럼값의 산술 연산이 가능하다

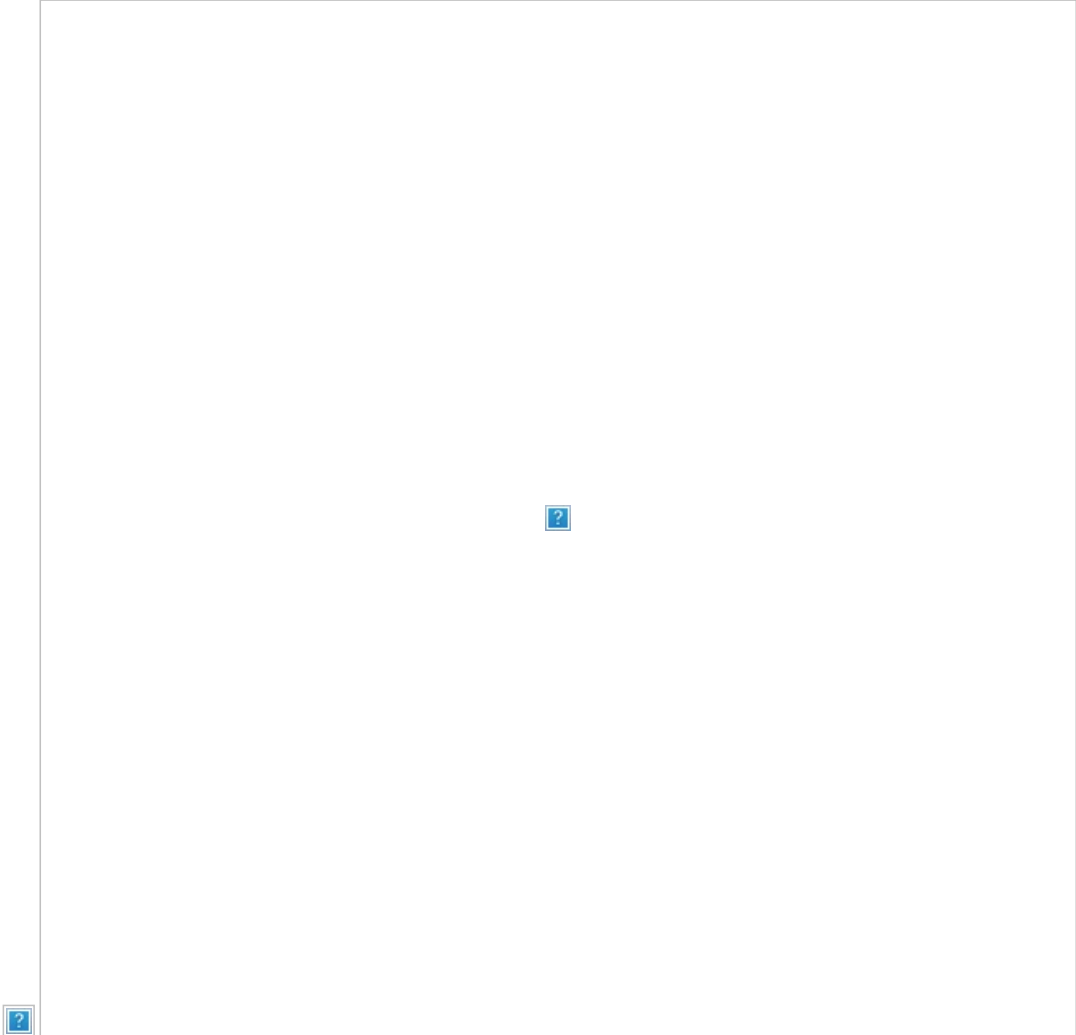
칼럼값:테이블에서 한 칸의 정보 (보라색 칸)



칼럼명에 산술 연산을 작성하면 조회시 연산이 반영되서 조회된다.



SALARY라는 월급 값을 연봉($SALARY \times 12$)으로 계산에 출력했다



입사 일수도 산술 연산이 가능하다 계산 결과 값은 (총 입사일 / 1년)으로 나눈 것

칼럼들의 별칭을 지정 할 수 있다.

- 1. 컬럼명 AS 별칭
- 2. 컬럼명 별칭
- 3. 컬럼명 AS "별칭"
- 4. 컬럼명 "별칭"

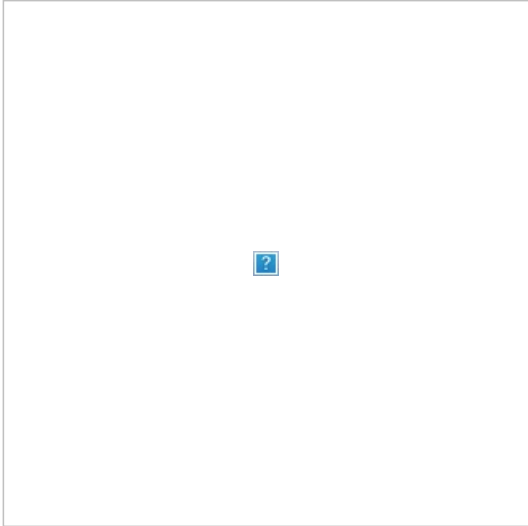
1 번은 오로지 '문자'로만 명명 가능하다.

2 번은 1번에서 AS만 생략한 것

3번은 " "쌍따옴표 안에 특수문자 띄어쓰기 문자 모두 사용가능하다

4번은 3번에서 AS만 제외하고 사용한 것.

큰 차이는 없음으로 편한 방식으로 사용하면 된다.



리터럴 : 값 그 자체
DB에서 리터럴 : 임의로 지정한 값을 기존 테이블에 원래 있었던 것 처럼 사용한다.
리터럴 표기법 : ' ' (홀따옴표)



EMPLOYEE 테이블에 원이라는 단위는 존재하지 않았지만
'원'이라는 리터럴을 AS 단위라는 별칭을 설정해주며 임의로 만들어 주었다.

테이블에서 중복된 값을 제거하고 조회하고 싶을 때는
DISTINCT을 사용한다



DECP_CODE 컬럼에서 하나의 값만 출력된다

WHERE

- 조건을 충족하는 값을 가진 행만 조회 하고자 할 때.
- 비교 연산자 : > , < , >= , <= , =(같다 한번만 사용) , != , <>(같지 않다)



- 어디 테이블인지 먼저 확인하고
- 조건을 확인하고
- 조회한다.



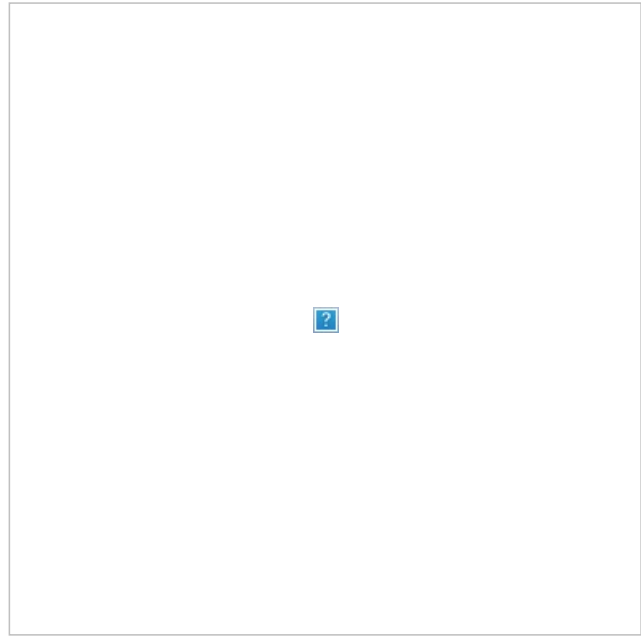
부서 번호가 D9인 직원의 사원번호, 이름, 전화번호, 부서코드 조회

논리 연산자(AND , OR)

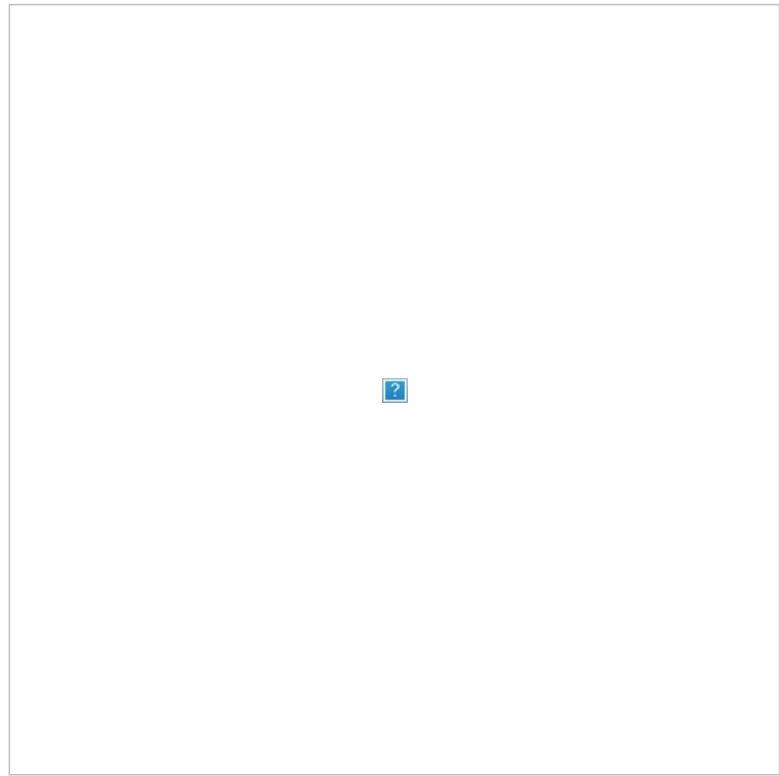


보기만 해도 바로 알것 같다.
만약 두번의 논리 연산이 필요할 경우 BETWEEN을 사용하는 경우가 나온다

칼럼명 BETWEEN A AND B : 칼럼값이 A 이상 B이하를 의미한다



NOT BETWEEN 의 경우 A미만 B초과인 경우



날짜의 논리 연산도 가능하다.



해당 구문에서는 설정값을 YYYY - MM -- NN 으로 설정 했기 때문에 저런 형식으로 타이핑 했다.