

## 1장

- 데이터베이스 - 의미있는 데이터 모음
- 데이터베이스 관리시스템 - 저장 조회 관리 --> DBMS
- 관계형 데이터베이스 관리시스템 --> RDBMS
- > 데이터 구조를 테이블 형태 표현(행과 열)
- **오라클 11G XE** ENTERPIRSE / STANDARD / EXPRESS EDITION설치, MYSQL

## 2장

- 오라클 설치 / 제거(제어판-프로그램 제거-오라클 제거)
- **RUN SQL COMMAND LINE 실행-**

CONNECT system/system 엔터

alter user hr identified by hr account unlock; 엔터

connect hr/hr 엔터

select \* from tab; 엔터

**exit** 엔터

- SQL --> RDB 언어
- **RUN SQL COMMAND LINE 실행** + 별도설치가능 (SQL DEVELOPER, ORANGE, TOAD) + **ECLIPSE** (자바, 웹 실행, **SQL 실행**)

## 3장 데이터베이스 설계 / 데이터베이스 백업 / 복원

## 4장 데이터베이스 테이블 구조 모델링

## 5장 SQL DEVELOPER 추가 설치 사용법

## 6장 179P - PL/SQL기본-SQL 문법 설명

sql 문법

1> db 언어

2> 대소문자 구분하지 않는다

3> 단, 계정의 암호 대소문자 구분한다

4> 데이터값은 대소문자 구분한다

5> 오라클 문자열 데이터값 - '문자열값'

6> 클래스 생성 x , main x.

7> sql 문법 문장 1개 - 실행

#### sql 종류

DATA QUERY LANGUAGE-DQL	select ~	조회
DATA DEFINITION LANGUAGE-DDL	create table ~ alter table ~ drop table ~	테이블 구조 정의 테이블 구조 변경 테이블 삭제
DATA MANIPULATION LANGUAGE-DML	insert update delete	테이블 내에 데이터 저장 테이블 내에 데이터 수정 테이블 내에 데이터 삭제
TRANSACTION CONTROL LANGUAGE -TCL	commit rollback	연속 sql 실행 완료 / 취소 트랜잭션 처리
DATA CONTROL LANGUAGE -DCL ( DBA 만 사용 명령어 <b>SYSTEM</b> 계정-DBA HR-일반계정	<b>grant</b> <b>revoke</b>	계정 특정 권한 부여 계정 특정 권한 부여 취소

#### 오라클 설치

내부 2개 존재 - SYSTEM / hr (잠금해제) / test

테이블 정의 - 데이터 저장 계정 생성-권한 부여

```

connect system/system;
create 사용자명 test identified by 암호;
grant resource, connect to 사용자명;
connect 사용자명;
show user;
disconnect;

```

#### - DQL – 조회 – SELECT

```
select 컬럼명 from 테이블명
```

학생1명데이터=(학번 이름 전공 학년)  
 학생2명데이터=(학번 이름 전공 학년)  
 .....  
 학생n명데이터=(학번 이름 전공 학년)

#### 학생테이블

학번컬럼(COLUMN-열)	이름컬럼	전공컬럼	학년컬럼
ROW-행-레코드 100	김학생	it	4
200	'''	'''	''

```

select 이름컬럼, 전공컬럼, 학년컬럼, 학번컬럼
from 학생테이블

```

```

select 이름컬럼
from 학생테이블

```

```

select *
from 학생테이블

```

SELECT 11개컬럼 FROM EMPLOYEES ;

DESC EMPLOYEES;

edit; ==> 직전 명령 편집 - 저장 - 닫기 - / 엔터 - 실행

select tname from tab;

- employees 테이블에서 first\_name 열 조회

select first\_name , last\_name , employee\_id from employees;

- 급여 컬럼 - salary 조회

select salary from employees;

- 급여 12배 = 연봉 컬럼 없다(연봉 연산식: + - \* / ) 조회

select salary, salary\*12 from employees;

- 실제 컬럼명을 조회 임시 변경 - alias

select salary 월봉, salary\*12 연봉 from employees;

==>

select \* , 컬럼명 , as 별칭, + - \*/

from 테이블명

- 모든 사원의 직종코드 조회

select job\_id from employees; ==> 107 개 동일 직종코드 여러번

예) IT\_PROG --> IT PROGRAMMER직종

- 직종코드 종류별 1개 조회(동일 직종코드 1번 조회)

select DISTINCT job\_id from employees; ==> ???

- 이름 대문자 변경 조회

(자바) ELLEN<-----"Ellen".toUpperCase();

select upper(first\_name) from employees;

select 컬럼정보

from 테이블명-->테이블 "모든" 레코드 조회

where 레코드 조건식 만족

- employees테이블 급여 salary 10000 이상인 사원의

이름과 급여 조회

select first\_name, salary from employees

where salary >= 10000;

- employees테이블 급여 salary 10000 이상이고 15000 이하인 사

원의 이름과 급여 조회

select first\_name, salary from employees

where salary >= 10000 and salary <= 15000;

select first\_name, salary from employees

where salary between 10000 and 15000;

-employees 테이블에서 사번(EMPLOYEE\_ID 컬럼) 100 , 120, 200, 300, 105, 205, 305 사원의 사번 조회

```
select employee_id from employees
where employee_id in (100, 120, 200, 300, 105, 205, 305)
==> employee_id = 100 or ....
```

where 컬럼명 연산자 값;  
where salary >= a + 10000;

컬럼명 >= 'a' and 컬럼명 <='c'  
숫자 = 대소비교  
문자 = 사전 나열 순서(앞 작다 뒤 크다)  
날짜 = 오래 작다 / 최근 크다

비교연산자	> >= < <= = !=(<>)
산술연산자	+ - * / (나머지함수제공)
논리연산자	and or not
범위연산자	컬럼명 between A and B
목록연산자	컬럼명 in (값1, 값2, ...)
유사패턴연산자	like , %, _ % : 0개 이상의 모든 글자 _ : 1개의 모든 글자 _____
null비교연산자	is null is not null where first_name is null;

- 02/06/07 입사일자(hire\_date)

```
select hire_date from employees
```

```
where hire_date = '02/06/07';
```

날짜와 문자 - ' '

```
select sysdate from dual;-> 오라클 현재시각 함수
```

'21/08/12' --> 오라클 기본 날짜시각형식 - 'rr/mm/dd'

'99/01/01' -->

rr -> 0-49 - 2000년대

50-99 - 1900년대

```
select to_char(sysdate , 'yyyy-mm-dd hh24:mi:ss') from dual;
```

- 02년도 입사일자(hire\_date)

```
select hire_date from employees
```

```
where hire_date >= '02/01/01'
```

```
and hire_date <= '02/12/31';
```

문자는 사전나열순이라고 하셨는데 한글이나 다른 언어도 가능한가요?

숫자 < 영문자 대문자 < 영문자 소문자 < 특수문자 < 한글이나 다  
국어 < 특수문자들....

```
- select first_name from employees;
```

- 이름에 'ex' 문자 포함 사원의 이름 조회

```
select first_name from employees
```

```
where first_name like '%ex%';
```

% -> 0개 이상의 아무 문자

like

????ex?????

sql 대소문자 구분하지 않는다

암호나 데이터문자값 대소문자 구분한다

- 이름에 'Ex' 문자 시작 사원의 이름 조회

```
select first_name from employees
```

```
where first_name like 'Ex%';
```

- 이름에 'ex' 문자 종료되는 사원의 이름 조회

```
select first_name from employees
```

```
where first_name like '%ex';
```

LIKE '%A%'

LIKE 'A%'

LIKE '%A'

- 이름에 'e' 포함하되 2번째 있는 경우 이름 조회

Aexxxxxx

BeXXXXX

ce

```
select first_name from employees
```

```
where first_name like '_e%'
```



**\_**( 반드시 1개) , **%**(0개이상 여러개) --> 아무 글자 가능

- 이름에 'e' 포함하되 3번째 있는 경우 이름 조회

```
select first_name from employees
```

```
where first_name like '__e%'
```

- 이름에 'e' 포함하되 2개 이상 있는 경우 이름 조회

```
select first_name from employees
```

```
where first_name like '%e%e%';
```

eexxxex

xxxexxxxx

exxxexxx

xxeeexxxe

- hire\_date 컬럼에서 입사년도 (년 월 일) 02년도

```
select hire_date from employees
```

```
where hire_date >= '02/01/01'
```

```
and hire_date <= '02/12/31';
```

02%

02\_\_\_\_\_

==> like 연산자

```
select hire_date from employees
```

```
where hire_date like '02%';
```

```
select hire_date from employees
```

where hire\_date like '02\_\_\_\_\_';

- employees 테이블 커미션 commission\_pct

commission\_pct 컬럼 - 실수 - .14 ( <= 1)

commission\_pct 컬럼 null

( 출력형태 run sql command line - 공백)

다른 툴 null , '-', --

- employees 테이블 커미션 받는 사원 조회

select commission\_pct from employees

where commission\_pct is null;

select employee\_id from employees;

==> 테이블 정의 - 107 저장 순서 기본 조회

- 이름 알파벳 오름차순 순서 조회

select first\_name from employees

order by first\_name;

select first\_name from employees

order by first\_name asc;

- 이름 알파벳 내림차순 순서 조회

select first\_name from employees

order by first\_name desc;

select employee\_id from employees

order by employee\_id desc;

-이름, 급여 조회. 급여 많은 사원부터 조회하되  
급여 같으면 이름 알파벳 순서

```
select first_name, salary from employees  
order by salary desc ?????, first_name ;
```

```
select first_name, salary from employees  
order by 2 desc , 1 ;
```

```
select first_name 이름 , salary 급여 from employees  
order by 급여 desc , 이름;
```

- 커미션 많은 사원부터 조회.

```
select commission_pct from employees  
order by commission_pct desc;-->null 먼저
```

```
select commission_pct from employees  
order by commission_pct asc;-->null 마지막
```

- 커미션 많은 사원부터 조회하되 null 마지막 조회

```
select commission_pct from employees  
order by commission_pct desc nulls last;
```

- 커미션 적은 사원부터 조회하되 null 먼저 조회

```
select commission_pct from employees  
order by commission_pct nulls first;
```

[SELECT 문법]

select 컬럼명

from 테이블명

[where 조회 데이터의 조건식 ]

[group by ]

[having ]

[order by 정렬순서 컬럼명 asc/desc nulls first/last , , , ]

- rownum 과 sample

```
select employee_id from employees
where employee_id <= 104;
```

- 사번 조회 아무 사번 5개 추출-> 매번 서로 다른 결과

```
select employee_id from employees sample(5);
```

- 이름 조회 순서대로 5개 추출-> 고정

```
select first_name from employees
where rownum <= 5;
```

- 급여를 조회하되 5명 추출

```
select rownum, salary from employees
where rownum <= 5
```

- 급여 많은 사원부터 전체 사원 조회

```
select rownum, salary
from employees
order by salary desc;
```

-select 실행

1> from 절 테이블 가져온다(8개 테이블 중에서 1개 테이블 메모리)

2> select 절 실행하면 salary 컬럼 가져올 때마다 rownum 생성

1 14000

2 17000

....

3> 급여 많은 사원부터 2번 결과의 순서 변경한다

2 17000

1 14000

....

3시 10분

집계함수=다중행함수	급여 총합 SELECT SALARY FROM EMPLOYEES; 107개 레코드 급여--> 1개 결과 SUM , AVG, MAX, MIN, COUNT, STD, VARIANCE
단일행함수	SELECT UPPER(FIRST_NAME) FROM EMPLOYEES; --> 107 레코드의 이름 대문자 변경

함수명(매개변수,,,,,)

SELECT SALARY FROM EMPLOYEES;

-급여 총합

SELECT SUM(SALARY) FROM EMPLOYEES;

-급여 평균

SELECT AVG(SALARY) FROM EMPLOYEES;

-최대 급여, 최소 급여

SELECT MAX(SALARY), MIN(SALARY) FROM EMPLOYEES;

-가장 먼저 입사한 사원의 입사일자 , 가장 최근에 입사한 사원의 입사일자

SELECT MIN(HIRE\_DATE) 사장의입사일 , MAX(HIRE\_DATE) 신입사원의입사일  
FROM EMPLOYEES;

- 이름 알파벳 순서상 가장 처음, 마지막 이름

SELECT MIN(FIRST\_NAME) 처음 , MAX(FIRST\_NAME) 마지막 FROM  
EMPLOYEES;

SELECT SUM(FIRST\_NAME) FROM EMPLOYEES;

-급여 받은 사원수

SELECT COUNT(SALARY) FROM EMPLOYEES;->107

-커미션 받는 사원수

SELECT COUNT(COMMISSION\_PCT) FROM EMPLOYEES;->35

- 부서 속한 사원수

SELECT COUNT(DEPARTMENT\_ID) FROM EMPLOYEES;->106

- 모든 사원수

SELECT COUNT(\*) FROM EMPLOYEES;->107

- 이름 있는 사원수

SELECT COUNT(FIRST\_NAME) FROM EMPLOYEES;->107

SUM, AVG -> 숫자타입 컬럼만

MAX, MIN -> 모든 타입 컬럼

COUNT (컬럼명)-> 모든 타입 컬럼.

컬럼의 NOT NULL 데이터 갯수만.

COUNT (\*)-> 테이블에서 NULL 데이터 갯수도 포함.

- 급여 총합 조회

SELECT SUM(SALARY) FROM EMPLOYEES;

- 10번 부서 속한 부서원들의 급여 총합 조회

SELECT SUM(SALARY) FROM EMPLOYEES

WHERE DEPARTMENT\_ID=10;

- 부서종류 조회

SELECT DISINCT DEPARTMENT\_ID FROM EMPLOYEES;

- 부서별 급여 총합 조회

SELECT SUM(SALARY) FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID;

- 급여 총합 조회(X)

**SELECT FIRST\_NAME, SUM(SALARY) FROM EMPLOYEES;**

**==> 집계함수 SELECT 시는 다른 컬럼 같이 조회 불가능**

- 부서별 급여 총합 조회

**SELECT DEPARTMENT\_ID, SUM(SALARY) FROM EMPLOYEES  
GROUP BY DEPARTMENT\_ID;**

**==> 집계함수 SELECT 시는 다른 컬럼 같이 조회 불가능.**

**단, GROUP BY 뒤 나열 컬럼명 같이 조회 가능**

- 부서별 급여 총합 조회하되 부서코드 NULL 제외.

급여총합 많은 부서부터 순서대로 조회

**SELECT DEPARTMENT\_ID, SUM(SALARY)  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID IS NOT NULL  
GROUP BY DEPARTMENT\_ID  
ORDER BY SUM(SALARY) DESC;**

**GROUP BY DEPARTMENT\_ID, JOB\_ID**

- 부서별 급여 총합 조회하되 총합이 10만 이상 조회

**SELECT DEPARTMENT\_ID, SUM(SALARY)  
FROM EMPLOYEES  
GROUP BY DEPARTMENT\_ID  
HAVING SUM(SALARY) >= 100000;**

**FROM->WHERE->GROUP BY-> HAVING ->SELECT ->ORDER BY**

[SELECT 문법]-작성 순서

select 컬럼명

from 테이블명

[where 조회 데이터의 조건식 ]

[group by 집계함수 적용기준 컬럼명 ]

[having 집계함수 조건식( SUM AVG MAX MIN COUNT )

[order by 정렬순서 컬럼명 asc/desc nulls first/last , , , ]

실행 순서

FROM->WHERE->GROUP BY-> HAVING ->SELECT ->ORDER BY

=====

함수, subquery 이용하면 해결 가능

224p – DML

7장 1,2절

-ORACLE의 데이터 형식

숫자 / 날짜 / 문자 / 이진수-대용량 동영상 이미지 음향

숫자	NUMBER(38) 정수 NUMBER(5) NUMBER(5, 0) 실수 NUMBER(10, 2) 정수8/소수점이하2 NUMBER(2,2) .14 ( <=1)
날짜	DATE 년 월 일 시 분 초 요일 값 저장 가능 기본적 DATE 타입 형식 – RR/MM/DD 형식/변경가능 RR 0~49 - 2000년대 50-99 -1900년대
문자	VARCHAR2(10BYTE) 'JAVA'--> 4BYTE '가나다'--> 9BYTE(한글 1개당 3BYTE)



	<p>CHAR(2) – 2BYTE 문자(알파벳 2문자)</p> <p>CHAR(10) – 10BYTE 고정 – [JAVA][6BYTE] 'JAVA' --&gt; 4BYTE</p> <p>VARCHAR2(10) – 10BYTE 최대/동적 – [JAVA] 'JAVA' --&gt; 4BYTE</p> <p>VARCHAR2(4000)- 1300 한글</p>
--	--

SELECT LENGTH('자바'), LENGTHB('자바') FROM DUAL

- 집계함수/단일행함수

집계함수	sum avg max min count	
단일행함수	문자열함수	length- 문자갯수 lengthb – 바이트수 upper / lower – 대소문자변경
	숫자함수	
	날짜함수	
	타입변환함수	
	NULL처리함수	

- 'A'-ASCII CODE

ASCII('A')

SELECT 컬럼명 , 함수명 FROM 테이블명;

select ASCII('A') from dual;

dual – 가상 테이블로 select 결과 저장 임시 테이블  
1행.

```
select ascii ('가') from dual;
```

```
select asciistr('가') from dual;
```

- employees 테이블에서 조회 first\_name salary 조회  
xxxx 사원은 급여 xxxx를 받습니다

<|| -> 문자열 결합,concat연산자>

```
select first_name || ' 사원은 급여 ' || salary || ' 를 받습니다'  
as 급여정보 from employees;
```

<concat – 문자열결합함수>

```
select concat( concat( concat( first_name , ' 사원은 급여 ') , salary ) , ' 를 받  
습니다')  
as 급여정보 from employees;
```

(자바) --> 이름변수 + " 사원은 급여 " + 급여변수 + " 를 받습니다"

instr – 특정 문자열 찾는 위치 리턴

```
select first_name from employees  
where instr(first_name, 'ex') >= 1;  
where first_name like '%ex%';
```

```
select first_name, instr(first_name, 'ex') from employees;  
==> 이름 ex 포함된 인덱스-1
```

- lower / upper / initcap -대소문자변경

```
select first_name from employees
where first_name = initcap('neena');
```

first\_name = 첫문자대문자+나머지소문자

```
select first_name from employees
where lower(first_name )= lower('NeeNa');
```

```
select first_name from employees
where upper(first_name )= upper ('NeeNa');
```

- instr('Neena', 'Ne') --> 1

- substr('Neena', 1, 2) --> 'Ne'

- 02년도 입사자 입사일 조회

```
where hire_date >= '02/01/01' and hire_date <='02/12/31';
```

```
where hire_date like '02%';
```

```
where substr(hire_date, 1, 2) = '02'
```

```
where instr(hire_date, '02') = 1
```

- trim : 제거

- pad : 추가

lpad, ltrim - 왼쪽 ? 추가, 왼쪽 공백 제거

rpadd, rtrim - 오른쪽

```
select length(trim(' java prgroam ')) from dual;
```

```
select lpad('java progrm' , 20, '-') from dual;
```

02월 입사자

## 8장

9장 이후 프로시저 함수 자바 연동 리눅스참조