

- 오라클 데이터베이스

자바 프로그래밍 언어-컴퓨터 다목적 프로그램 제작

데이터들 --> 영구적 파일/데이터베이스 저장

파일 텍스트들

1> 데이터 분리 기준 없다

2> 데이터 타입 기준 없다

3> 특정 위치 데이터 접근 불가능

4> 중복 데이터 가능

5> 데이터 현재 상태 잘못된 표현

a 자바프로그램    b 파이썬프로그램

100 김사원|56000.99

200|김수정|56000.99==> 수정 / 삭제

300|김사원|56000.99

100|박대리|90000.99

퇴사자정보

200|김수정|56000.99

데이터베이스

현실세계 의미있는 데이터 모음

학생 = (학번 이름 전공 학교명 학년 성적)

회사원 = (사번 이름 부서 회사명 직급 급여)

프로그래밍 언어- 자바 , c, html, python

데이터베이스 표현 방법

1> 계층형

2> 네트워크형

===>

3> 관계형 / 객체관계형 데이터베이스 제품

oracle / mysql / ms sqlserver / db2 / .....

4> 관계형 (relational DB-RDB)

데이터 관계를 행과 열의 테이블 구조 표현

열(COLUMN) – 데이터의 구성요소 1개 표현단위

행 – 1개 데이터 = ROW = RECORD

학생 테이블

학번 정수 3자리 중복X	이름 문자 10자리	성적 실수 5자리(2자리)
100	이학생	4.5
100	XXXX	XXXX

테이블 생성-삭제

데이터 1개 ROW 저장-삭제

데이터들 조회

함수들=자바메소드

--> 관계형 데이터베이스 활용 문법 언어

SQL

1> RDB 제품 종류에 관계없이 사용 "표준" SQL

+

2> 각 RDB 독자적 SQL

==> 1장

- 설치

- oracle 11g express edition

- 무료, 크기제한, 1개만 사용

enterprise / standard edition -->

데이터베이스 (xe)	system schema= 계정 (super user)	테이블들
	hr 이미 생성 잠김	테이블들 조회 실습
	계정 생성	테이블들 생성 데이터 저장

- 삭제

제어판-프로그램및기능- oracle 11g xe ..제거

- SQL 입력 실행 툴

1> RUN SQL COMMNAND LINE 실행

2> SQL DEVELOPER / ORANGE / TOAD

/ ECLIPSE DATA EXPLORER 기능

-- sql – 오라클 독자적 sql

4글자 축약 / 대소문자 구분 없다(단 암호 대소문자 구분한다)

SQL> connect system/

SQL> conn system/암호

SQL> alter user hr identified by hr account unlock;

SQL> disconnect

SQL> disconn

SQL> connect hr/hr

hr 테이블 실습 가능

SQL> select \* from tab;

SQL> disconnect

- SQL 종류

테이블 생성 – 변경 – 삭제

====> 데이터 저장 구조 정의 언어

-----> Data Definition lang.—DDL

학번--> ID NUMBER(5)

테이블에 데이터 저장-수정-삭제

====> 데이터 조작 언어

--> DATA MANIPULATION LANG.-->DML

계정 생성 – DB 접속 허용 SQL

--> SYSTEM 계정

DATA CONTROLL LANG.--.DCL

트랜잭션 제어 언어

## TRANSACTION CONTROLL LANG.--.TCL

### DATA QUERY LANG--> DQL --> 조회

ddl	create table..... alter table drop table
dml	insert .... update delete
dcl	grant, revoke 새로운 계정 생성시 사용
tcl	commit rollback
dql	select

hr 8개 테이블 조회 실습

conn hr/hr;

select \* from tab;--> 테이블 목록 조회

- 문법

desc 테이블명;

--> 테이블 컬럼명 타입 갯수

select 조회컬럼 from 테이블명;

- employees 테이블에서 first\_name 열 조회

select first\_name , last\_name from employees;

**select \* from employees;**

- 급여 컬럼 – salary

**select first\_name, salary from employees;**

- 급여 12배 = 연봉 조회

**select first\_name, salary, salary \* 12 from employees;**

- 급여 12배 = 연봉 조회

**select first\_name, salary, salary \* 12 from employees;**

실제컬럼명을 조회 임시 변경 – alias

**select first\_name as 이름 , salary as 월급 ,  
salary \* 12 as 연봉 from employees;**

**salary + commission\_pct**

**==> 숫자타입**

- employees 107명 사원 정보 저장'

- 직종코드 종류별 1개 조회(동일 직종코드 1번)

**select job\_id from employees;=>107**

**select distinct job\_id from employees;-->19**

- String : toUpperCase() ==> upper()

**select first\_name, upper(first\_name) as a from employees;**

select 컬럼명1,컬럼명2 | \* | as 별명 | distinct | +-\*/ | 함수  
from  
where  
group by  
having  
order by

- employees테이블 급여 salary 10000 이상인 사원의 이름과 급여  
조회

이름 = first\_name  
급여 = salary  
select first\_name, salary  
from employees  
where salary >= 10000 and salary <= 11000;  
where not salary >= 10000;  
where salary < 10000;

-employee\_id 컬럼 사번 50 100 150 200 250 300 인 사원 사번, 이  
름 조회

select employee\_id, first\_name  
from employees  
where employee\_id=50 or employee\_id =100  
or employee\_id =150 or employee\_id =200  
or employee\_id =250 or employee\_id =300

--->

```
select employee_id, first_name
from employees
where employee_id in(50, 100, 150, 200, 250, 300);
```

- employees 테이블에서 first\_name Jennifer

```
select first_name
from employees
where first_name = 'Jennifer'
```

자바->문자열- " " / 문자1개 -' '

오라클 -> 문자타입 - ' '

- employees 테이블에서 first\_name J 로 시작

( 암호나 " 문자데이터==> 대소문자 구분)

```
select first_name
from employees
where first_name like 'J%'
```

= 'Jennifer'

like 'J%'

- employees 테이블에서 first\_name r로 끝나는 사원 조회

```
select first_name
from employees
where first_name like '%r'
```

- employees 테이블에서 first\_name er 을 포함하는 사원 조회

```
select first_name
from employees
where first_name like '%er%'
```



% - 모든 문자, 문자 갯수 상관없다(0개이상)

- employees 테이블 job\_id 컬럼에서 manager 직종 조회

```
select job_id from employees
```

```
where job_id like '%MAN';
```

```
where job_id like '___MAN';
```

```
where job_id like '___W_MAN' escape 'W';
```

( 단 MANAGER 직종은 3글자MAN으로 끝난다)

( 단 MANAGER 직종은 2글자\_MAN으로 끝난다)

'\_anb\_def'

'가%나%다'

- employees 테이블 first\_name(이름), hire\_date 컬럼(입사일) 조회하  
되 입사년도가 2005년 이후 조회

```
select first_name, hire_date
```

```
from employees
```

```
where hire_date like '05/____'
```

```
where hire_date like '05/%'
```

```
where hire_date >= '05/01/01' and hire_date <= '05/12/31';
```

```
where hire_date between '05/01/01' and '05/12/31';
```

타입 / 데이터 형태 저장

```
desc employees;
```

date – 오라클 내장 포맷 / 포맷 변경

```
select sysdate from dual;
```

===> 현재시스템날짜시각정보

21/03/24

rr/mm/dd

rr --> 0-49 값 2000년대

50-99 값 1900년대

```
select hire_date from employees;
```

- employees 테이블 commission\_pct 컬럼 조회

```
select commission_pct
```

```
from employees
```

```
order by 1 asc;-->null 마지막
```

```
order by 1 desc ;-->null 처음
```

- employees 테이블 commission\_pct 컬럼 조회 값이 있는 데이터만 조회. null 제외

```
select commission_pct
```

```
from employees
```

```
where commission_pct is not null
```

```
order by 1 desc ;
```

```
desc employees;
```

(자바 참조형변수만 null)

(오라클 모든 타입의 데이터 값이 없으면 null -공백 -run sql command line)

(..... - null / NULL / - / # )

산술연산자	+ - * /
비교연산자	> >= < <= !=(<>) =
논리연산자	not and or
목록연산자	in (...)
유사연산자	like % - 모든 문자, 문자 갯수 상관없다(0개) _ - 모든 문자 1개
범위연산자	between ~ and
null처리 연산자	is null is not null <b>=null (x)</b> <b>!=null(x)</b>

select

from

where

group by

having

order by

select ~ from ~ where ?;

select ~ from ~ order by ?;

select ~ from ~ where ? order by ??;

select first\_name from employees order by first\_name asc;a-z가-하

select first\_name from employees order by first\_name desc;

select first\_name, salary from employees

order by salary desc , first\_name asc;

select hire\_date from employees order by hire\_date asc;

select hire\_date from employees order by hire\_date [asc생략가능];

select hire\_date, first\_name from employees order by 1, 2 [asc생략가능];

select hire\_date as 입사일 from employees order by 입사일 asc;

order by 컬럼명 or index or alias [asc생략]|desc

- 입사일 컬럼 – hire\_date 'rr/mm/dd'

- 입사월이 6월 입사 직원 이름 급여 입사일 조회되 입사일이 가장 최근 직원부터 조회

select first\_name, salary, hire\_date

from employees

where hire\_date like '\_\_\_06%'

order by 3 desc , 2 desc;

- 최근 입사자 5명만 조회

hire\_date 입사일자 desc 정렬하여 1-5 데이터 조회

```
select hire_date from employees  
order by hire_date desc
```

- upper / sysdate / rownum

```
select rownum, hire_date from employees  
where rownum <= 5  
order by hire_date desc
```

-작성순서 select from where order by	-실행순서 from 테이블 찾는다->107 where 조건에 맞는 레코드를 찾는다-> 107 같거나 더 적은 갯 수 select 컬럼 조회한다 order by 정렬 기준 컬럼 정렬한 다=순서 뒤바뀐다
--	--

rownum – 조회하는 행번호생성함수(1~)

- subquery

1> 정렬 이후 상위 몇개 ?==> top-n QUERY

```
select rownum , hire_date  
from (select * from employees order by hire_date desc)  
where rownum <= 5;
```

```
2> desc departments;  
select * from departments;
```

Sales 부서에 근무하는 사원의 이름 조회

```
select first_name  
from employees  
where department_name='Sales'
```

```
select department_id from departments  
where department_name='Sales';
```

```
select first_name, department_id  
from employees  
where department_id=(select department_id from departments  
where department_name='Sales');
```

107 employees 테이블 100 이사원 .... 80	27 departments 테이블 10 총무부 20 인사부 ... 80 교육부
--------------------------------------	---

- 이름 Susan과 같은 부서에 근무하는 사원의 직종과 급여 조회  
employees

job\_id  
salary  
first\_name  
department\_id

```
select job_id, salary  
from employees  
where department_id in(select employee_id  
from employees  
where first_name = '????')
```

```
1.select department_id  
from employees  
where first_name = 'Susan';
```

- William과 같은 직종을 가진 사원의 부서, 급여 조회

```
select department_id, salary, job_id  
from employees  
where job_id in  
(select job_id from employees where first_name='William')
```

- Susan보다 더 급여를 많이 받거나 동일하게 받는 사원의 사번, 이름, 급여 조회

```
select employee_id, first_name, salary
from employees
where salary >=
(select salary from employees where first_name='Susan') 6500
```

- William보다 더 급여를 많이 받거나 동일하게 받는 사원의 사번, 이름, 급여 조회

```
select employee_id, first_name, salary
from employees
where salary >= all
(select salary from employees where first_name='William')
```

```
select employee_id, first_name, salary
from employees
where salary >= any
(select salary from employees where first_name='William')
```

예) (6500, 1000, 3000)

1> 모든 William의 급여와 같거나 많을 때

====> 6500 같거나 많을 때 ( >= all)

2> 1명의 William 급여와 같거나 많을 때

====> 1000 같거나 많을 때 ( >= any)



select .. (select ,,)	
단일행 리턴	= / > / >= < <= !=
다중행 리턴	= -> IN != -> NOT IN > all > any

select +-

날짜 > 대소비교, + -

select sysdate from dual;

--> '21/03/24'

sysdate+1--> 내일날짜나

sysdate - 1-->어제날짜

**sysdate함수 - hire\_date컬럼**

**현재일 - 입사일 ==> 입사 경과일수**