

- 오라클

- 1> 데이터베이스 - 자료(데이터)의 의미있는 모음
- 2> 영구적 저장
- 3> 여러 프로그램에서 동시 사용 가능
- 4> 관계형
- 5> 행(1개 데이터 모음) 열(1개 행 구성 여러개 요소) 테이블
- 6> C:\Wkdigital\Woraclexe 폴더에 설치
- 7> 컴퓨터 부팅시 자동시작(설치시 설정)
- 8> 수동시작 필요시

windows메뉴-oracle 11g express edition-start database

- 9> 관계형데이터베이스 접근 언어 - sql
- 10> run sql command line(=SQL PLUS) - sql 작성, 편집, 실행
- 11> 이클립스 DATA EXPLORER 설정 사용 가능

1장 데이터베이스 개념

2장 설치

3장 전체실습-테이블, 데이터, 조회

-->hr 계정 7개 테이블 데이터 저장 상태

5장 sql developer toad orange visual studio code

6장

(내일)4장 데이터베이스 구성 - 데이터베이스 모델링 - 정규화 - 8장 테이블 생성

DQL=조회 data query language	SELECT
DDL 테이블	
DML 데이터	
TCL 트랜잭션관리	

산술	+ - * /
비교	> >= < <= != =
논리	not and or
목록	in(...,...,...)
범위	between a and b
유사패턴	like '%'
==> 문자데이터만 가능	like ' _ '

null비교연산자	is null is not null
-----------	------------------------

select 컬럼명 as 별칭 , sysdate, upper('aaa') as 별칭, +/-/*

*

from 테이블명

where 컬럼명 연산자 값

group by

having

order by 컬럼명 인덱스 별칭 asc생략가능|desc 기준컬럼에 따라 정렬

subquery

select.. (select.....);

select rownum, salary

from (select salary from employees order by salary desc)

where rownum <= 3;

- Susan (이름 사원 1명) 과 같은 부서 사원의 부서코드, 이름 조회

select department_id, first_name

from employees

where department_id = (select department_id from employees where first_name='Susan');

- William (이름 사원 2명) 과 같은 부서 사원의 부서코드, 이름 조회

select department_id, first_name

from employees

where department_id in (select department_id from employees where first_name='William');

- Susan (이름 사원 1명) 과 같거나 더 많은 급여를 받는 사원의 부서코드, 이름 조회

select department_id, first_name

from employees

where salary >= (select salary from employees where first_name='Susan');

- William (이름 사원 2명) 과 같거나 더 많은 급여를 받는 사원의 부서코드, 이름 조회
사원의 부서코드, 이름 조회

```
select department_id, first_name
```

```
from employees
```

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

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

```
select
```

```
from
```

```
where
```

```
group by
```

```
having
```

```
order by
```

그룹함수

100	1개
200	
4300	
sum	총합- 숫자
avg	평균 - 숫자
count	갯수 - 숫자 , 문자, 날짜 null 값 제외 count(*)
max	최대값 - 숫자, 문자, 날짜
min	최소값 - 숫자, 문자, 날짜
stdev	표준편차-숫자
variance	분산-숫자

```
select count(department_id) as 부서갯수, count(salary) 급여갯수,  
count(*) as 모든사원수 from employees;
```

- employees 테이블 급여 총합 조회
- employees 테이블 급여 평균 조회
- employees 테이블 급여 갯수 조회

- employees 테이블 최대급여 조회

- employees 테이블 최소급여 조회

```
select max(salary), min(salary) from employees;
```

```
select max(first_name), min(first_name) from employees;
```

```
select max(hire_date), min(hire_date) from employees;
```

- 사원이름, 최대급여 조회

(그룹함수 조회 select 절에 다른 컬럼 기술 불가.

단 group by 뒤 기술 컬럼 제외)

```
select first_name, max(salary) from employees;
```

107 1

```
select first_name, salary from employees
```

```
where salary = (select max(salary) from employees)
```

```
or salary = (select min(salary) from employees);
```

GROUP BY

HAVING

ROLLUP,CUBE

- 사원 80부서원 급여 총합 조회

```
select sum(salary)
```

```
from employees
```

```
where department_id=80;
```

- 부서별로 급여 총합 조회(부서 배정 안된 사원 (department_id null) 제외)

```
select department_id, sum(salary)
```

```
from employees
```

```
where department_id is not null
```

```
group by department_id;
```

- 부서별로 급여 총합 조회(부서 배정 안된 직원 (department_id null) 제외)

```
select department_id as 부서, job_id as 직종, sum(salary) 부서직종별급여총합
from employees
where department_id is not null
group by department_id, job_id
order by department_id;
```

10 전산직 xxx

10 회계직 xxx

- 부서별로 급여 총합 조회하되 부서별급여총합이 50000 이상인 부서의 결과만 조회

```
select department_id, sum(salary)
from employees
group by department_id
having sum(salary) >= 50000;
```

(1,. from -> 2.where(일반 조건식) -> 3. group by --> 4.having(그룹함수 조건식) --> 5.select

- 부서별로 급여 총합 조회하되 **사원의 급여가 5000 미만은 제외**하고

부서별급여총합이 50000 이상인 부서의 결과만 조회

```
select department_id, sum(salary)
from employees
where salary >= 5000
group by department_id
having sum(salary) >= 50000
order by sum(salary) desc, department_id;
```

- rollup

```
select department_id as 부서, job_id as 직종, sum(salary) 부서직종별급여총합
from employees
where department_id is not null
group by rollup(department_id, job_id);
```

부서 직종

10	aaa	10번	aaa	직종 총합
10	bbb	10번	bbb	직종 총합
10		10 번	부서(aaa + bbb)	총합

```
select department_id as 부서, job_id as 직종, sum(salary) 부서직종별급여총합
from employees
where department_id is not null
group by cube(department_id, job_id);
```

부서 직종

10	aaa	10번	aaa	직종 총합
10	bbb	10번	bbb	직종 총합
10		10 번	부서(aaa + bbb)	총합
20	aaa			
	aaa		aaa	직종 (10 + 20) 총합

214p-223p with cte --> 8장 view 동일 개념

224p – 6장 마지막 --> 테이블생성

DML 데이터 저장 – 수정 – 삭제

7장

데이터형식 오라클 31개

표준SQL+종속SQL

문자 VARCHAR2	CHAR VARCHAR2 -> 영문 -1 바이트 한글 - 3 바이트 CHAR(50) --> 'ABC' --> [ABC+47바이트고정] VARCHAR2(50) --> 'ABC' ==> [ABC] NCHAR NVARCHAR2 -> 유니코드 2바이트 한글 '데이터'-->대소문자구분
정수 NUMBER(8)	BINARY_INT INT NUMBER(8) NUMBER(8,0)
실수 NUMBER(8,2)	BINARY_FLOAT FLOAT NUMBER(8,2) 정수6.소수2
날짜 date	초 표현 -> DATE 1/1000초 표현 ->TIMESTAMP TIMESTAMPXXXXXX
대용량/기타	CLOB – 1TB 문자열 대용량 데이터 웹서버(자바)- 네트워크->DB BLOB - 1TB 바이너리 대용량 데이터 BFILE BIN

SQL --> 1개문장 실행

PL/SQL --> 반복, 조건, 변수선언X

일반함수

SELECT SYSDATE

FROM DUAL;

==> 날짜(RR/MM/DD)

dual 테이블; ==> 가상 임시 테이블

함수 결과 조회 사용 . 1행

SELECT SYSDATE FROM dual;

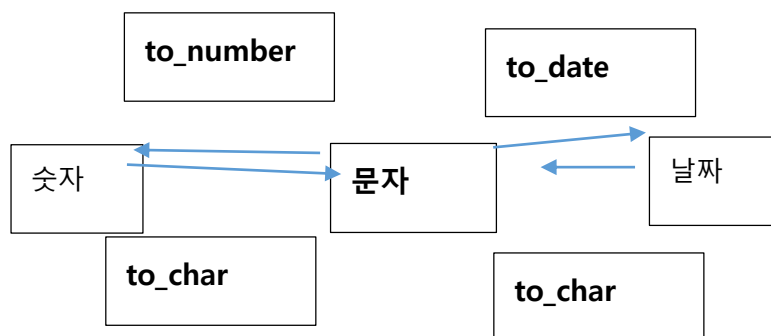
- cast

```
select sysdate from dual;
```

```
select cast( sysdate as timestamp ) from dual;
```

```
select cast( 12345.678 as number(10,2) ) from dual;
```

- to_char / to_number / to_date



```
SELECT 100+200 FROM DUAL;
```

```
SELECT '100'+'200' FROM DUAL; --> '0-9구성' 자동 숫자 변환
```

```
SELECT '123,456'+'200' FROM DUAL;
```

```
SELECT TO_NUMBER('123,456' , '999,999')+'200' FROM DUAL; --> 명시적
```

```
SELECT TO_NUMBER('$100', '$999' ) + TO_NUMBER('$200', '$999' ) FROM DUAL;
```

'123,456' ---> 숫자 변환

```
TO_NUMBER('123,456' , '999,999')--->123456
```

```
TO_NUMBER('$100', '$999' )-->100
```

```
SELECT 123456 FROM DUAL;--> 123456
```

```
SELECT TO_CHAR(123456.789, '$999,999') FROM DUAL;
```

--> \$123,456 문자 형태 변환

```
SELECT TO_CHAR(123456.789, '$999,999.99') FROM DUAL;
```

```
SELECT TO_CHAR(123456.7, '$999,999.99') FROM DUAL;
```

```
SELECT TO_CHAR(123456.7, 'L999,999.99') FROM DUAL;
```


,	, 기호
\$	\$ 기호
L	LOCALE CURRENCY - ₩
9	1자리숫자
0	1자리숫자
YY – 2000년대 YYYY RR(0-49, 50-99)	년도
MM	월
DD	일
HH HH24	시간
MI	분
SS	초
DAY	요일

SELECT SYSDATE FROM DUAL;

21/03/25

- 현재 오라클 설정 날짜형태

SELECT TABLE_NAME FROM DICT
WHERE TABLE_NAME LIKE '%NLS%';

--> NLS 포함 테이블

DESC NLS_SESSION_PARAMETERS;

SELECT * FROM NLS_SESSION_PARAMETERS
WHERE PARAMETER= 'NLS_DATE_FORMAT';

SELECT SYSDATE FROM DUAL; --> 21/03/25

SELECT SYSDATE FROM DUAL; --> 2021/03/25 13:39:11

```
SELECT TO_CHAR( SYSDATE , 'YYYY/MM/DD DAY HH24:MI:SS')  
FROM DUAL;
```

```
SELECT TO_CHAR( SYSDATE , 'fmYYYY"년도" MM"월" DD"일" HH"시" MI"분" SS"초"' )  
FROM DUAL;
```

문자-->날짜

to_date

'21/03/25' ---> 문자 자동 날짜

```
select sysdate + 1 from dual;
```

```
select to_date('21/03/25' , 'yy/mm/dd') + 1 from dual;
```

```
select sysdate + 365 from dual;
```

- 오늘로부터 5년후 날짜 조회

```
select to_char(sysdate, 'yyyy')+5 from dual;
```

- 오늘로부터 5개월후 날짜 조회

```
select to_char(sysdate, 'mm') + 5 from dual;
```

05년도 입사자 조회

```
select hire_date from employees
```

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

```
select hire_date from employees
```

```
where to_char(hire_date, 'yyyy') = '2005';
```

3월 입사자 조회

```
select hire_date from employees
where hire_date like '___03%';
```

```
select hire_date from employees
where to_char(hire_date, 'MM') = '03';
```

- instr함수 - 문자열

```
select hire_date from employees
where instr(hire_date, '03') = 4;
```

- substr함수 - 문자열

```
select hire_date from employees
where substr(hire_date, 4, 2) = '03';
```

숫자나 날짜 -to_char--> 문자열

숫자나 날짜 -자동형변환--> 문자열

100 --> '100'

21/03/25'

날짜 문자 ==> ' ' ' '

정수 실수 --->

```
select to_date('21/03/25', 'yy/mm/dd')+100 from dual;
```

```
select to_date('50/03/25', 'yy/mm/dd') +100 from dual;
```

```
select TO_CHAR( to_date('50/03/25', 'yy/mm/dd') , 'yyyy' ) from dual;
```

```
select TO_CHAR( to_date('50/03/25', 'yy/mm/dd') , 'mm' ) from dual;
```

타입변환함수	CAST TO_DATE TO_CHAR TO_NUMBER
그룹함수	SUM AVG MIN MAX COUNT STDEV VARIANCE
문자데이터함수	UPPER lower initcap length lengthb substr instr

	ltrim rtrim
숫자데이터함수	mod- 나머지함수 mod(10, 3) round-반올림 round(3.6789, 0); 4 round(3.6789, 1); 3.7 round(3.6789, 2); 3.68 round(333.6789, -1) 330 round(363.6789, -2) 400 -3 -2 -1 0 .1 2 3 to_char(123456.789, '999999.9') trunc trunc(3.6789, 0); 3
날짜데이터함수	SYSDATE SYSTIMESTAMP – 1/1000초 ADD_MONTHS() MONTHS_BETWEEN()
순위함수	ROWNUM- SUBQUERY ROW_NUMBER() RANK() DENSE_RANK() ROW_NUMBER() RANK() DENSE_RANK() 순위함수() OVER(PARTITION BY 소그룹컬럼명 ORDER BY 컬럼명 ASC DESC)
NULL처리함수	NVL(SALARY, 0)

--	--

- COMMISSION_PCT 컬럼

커미션 표현 컬럼

커미션 못받는 사원 - NULL - 0

커미션 받는 사원 - .4 - 0.4

```
select first_name, nvl(commission_pct, 0) from employees;
```

```
select first_name, nvl( to_char(commission_pct), '보너스없음') as 보너스
from employees
```

```
select first_name, nvl( to_char(commission_pct), '보너스없음') "보 너 스"
from employees
```

nvl(컬럼명, null대체값)

nvl(컬럼명정수, null대체값정수)

nvl(컬럼명문자열, null대체값정수+문자열)

-급여 많은 사원부터 순위 매기자

```
SELECT FIRST_NAME AS 이름, SALARY AS 급여,
ROW_NUMBER() OVER (ORDER BY SALARY DESC ) AS 급여순위
FROM EMPLOYEES;
```

```
SELECT FIRST_NAME AS 이름, SALARY AS 급여,
RANK() OVER (ORDER BY SALARY DESC ) AS 급여순위
FROM EMPLOYEES;
```

```
SELECT FIRST_NAME AS 이름, SALARY AS 급여,
DENSE_RANK() OVER (ORDER BY SALARY DESC ) AS 급여순위
FROM EMPLOYEES;
```

```
SELECT FIRST_NAME AS 이름, SALARY AS 급여, DEPARTMENT_ID AS 부서코드,
ROW_NUMBER() OVER
(PARTITION BY DEPARTMENT_ID ORDER BY SALARY DESC )
```

AS 급여순위
FROM EMPLOYEES;

SELECT SYSDATE + 1 FROM DUAL;

SELECT TO_CHAR(SYSDATE, 'YYYY') + 1 FROM DUAL;

SELECT TO_CHAR(SYSDATE, 'MM') + 1 FROM DUAL;

-->

SELECT ADD_MONTHS(SYSDATE, 1) FROM DUAL;

SELECT SYSDATE + 30 FROM DUAL;--->(X)

3. 'Adam의 입사일은 05/11/2 이고, 급여는 7,000₩ 입니다.' 의 형식으로 직원 정보를 조회하시오.

직원정보

Adam의 입사일은 05/11/2 이고, 급여는 7,000₩ 입니다.

select

first_name || '의 입사일은 ' || to_char(hire_date, 'yy/mm/fmdd') || ' 이고, 급여는 ' ||
LTRIM (to_char(salary, '999,999L')) || ' 입니다.'

as 직원정보

from employees;

-입사한지 경과년수 조회

```
SELECT ROUND( (SYSDATE - HIRE_DATE) / 365 )  
FROM EMPLOYEES
```

- 입사한지 경과주수 조회

```
SELECT ROUND( (SYSDATE - HIRE_DATE) / 7 )  
FROM EMPLOYEES
```

- 입사한지 경과개월수 조회

```
SELECT MONTHS_BETWEEN(SYSDATE, HIRE_DATE) FROM EMPLOYEES
```

```
SELECT SYSDATE FROM DUAL;
```

```
SELECT SYSTIMESTAMP FROM DUAL;
```

- FIRST_NAME er 포함 조회

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

```
select first_name  
from employees  
where instr(first_name, 'er') > 0
```

```
select first_name  
from employees  
where lower(first_name) = 'jennifer';  
where upper(first_name) = 'JENNIFER';  
where initcap(first_name) = initcap('JENNIFER');
```

```
replace / translate ('이것이 자바다', '자바', '오라클')
```

```
substr('이것이 자바다', 1, 2) --> 이
```

```
instr('이것이 자바다', '이') --> 1
```

```
select length(ltrim('   aaa   ')) from dual; -> 'aaa   '
```

```
select rtrim('   aaa   ') from dual; -> '   aaa'
```

```
select ltrim('##aaa##' , '#') from dual; --->'aaa##'
```

```
select rtrim('##aaa###') from dual; --> '##aaa'
```

```
select 3456.789 , round(3456.789, 2), round(3456.789, 1), round(3456.789, 0),  
round(3456.789, -1), round(3456.789, -2)  
from dual;
```

```
select 3456.789 , trunc(3456.789, 2), trunc (3456.789, 1), trunc(3456.789, 0),  
trunc(3456.789, -1), trunc(3456.789, -2)  
from dual;
```

- employees 테이블에서 홀수 사번 조회

mod함수

```
SELECT EMPLOYEE_ID  
FROM EMPLOYEES  
WHERE MOD(EMPLOYEE_ID, 2) = 1;
```

XXXX 2로 나누어 나머지가 1이면 홀수

0이면 짝수

- employees 테이블에서 입사년도별 급여 평균 조회하되

평균은 정수로 출력. 소수점 이하 버리자.

```
SELECT SUBSTR(HIRE_DATE, 4, 2) AS 입사월,  
TRUNC( AVG(SALARY) ) AS 평균급여  
FROM EMPLOYEES  
GROUP BY SUBSTR(HIRE_DATE, 4, 2);
```



```
select row_number() over( order by salary desc), salary from employees;
```

```
select row_number() over(partition by department_id order by salary desc), salary,  
department_id from employees  
order by department_id;
```

```
select rank() over( order by salary desc), salary from employees;  
select dense_rank() over( order by salary desc), salary from employees;
```

-employees 테이블의 입사일과 급여를 조회하되 아래 형식으로 조회한다.

- employees 테이블에서 입사년도별 급여를 조회하되 아래 형식으로 조회한다.

JOIN - SELECT

8장

DDL

제약조건

DML

TCL

```
select value from nls_session_parameters  
where parameter='NLS_DATE_FORMAT';
```