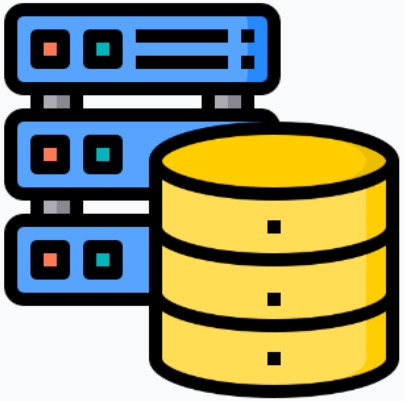
2장. SQL – DDL, DML

데이터 베이스 구성요소



SELECT – 자료 검색

SELECT 문

SELECT 열이름 (or 별칭)

FROM 테이블 이름

1. 열(칼럼)을 조회하고 정렬하는 방법

전체 검색 및 특정 열 검색

```
1  -- 테이블 전체 조회
2  SELECT * FROM employees;
3
4  --1. 열을 조회하고 정렬하는 방법
5  -- 특정 열만 조회
6  SELECT employee_id, first_name, last_r
7  FROM employees;
8
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1	100	Steven	King
2	101	Neena	Kochhar
3	102	Lex	De Haan
4	145	John	Russell
5	146	Karen	Partners
6	201	Michael	Hartstein
7	108	Nancy	Greenberg
8	205	Shelley	Higgins

SELECT – 자료 검색

별칭(이름) 만들기

```
-- AS를 사용하여 별칭 만들기(생략 가능)
SELECT employee_id AS 사원번호, first_name 이름, las
FROM employees;
```

사원번호	이름	성
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan
103	Alexander	Hunold
104	Bruce	Ernst
105	David	Austin

ORDER BY를 이용한 정렬

```
-- 정렬 : ORDER BY - ASC(오름차순), DESC(내림차순)
SELECT employee_id, first_name, last_name, salary
FROM employees ORDER BY salary DESC;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
100	Steven	King	24000
101	Neena	Kochhar	17000
102	Lex	De Haan	17000
145	John	Russell	14000
146	Karen	Partners	13500

SELECT – 자료 검색

```
-- 데이터 값 계산하기 : 산술 연산자
SELECT employee_id, salary, salary+500, salary-100
FROM employees;

SELECT employee_id AS 사원번호, salary AS 급여, salary+500 AS 추가급여,
FROM employees;
```

	사원번호	급여	추가급여	인하급여
1	100	24000	24500	23900
2	101	17000	17500	16900
3	102	17000	17500	16900
4	103	9000	9500	8900
5	104	6000	6500	5900
6	105	4800	5300	4700

```
-- 데이터 값 연결하기 : 연결 연산자 '||'
SELECT employee_id, first_name ||' '||last_name
FROM employees;

SELECT employee_id, first_name ||'@'||'gmail.com'
FROM employees;
```

SQL | 50개의 행이 인출됨(0.003초)

	EMPLOYEE_ID	NAME
1	100	Steven King
2	101	Neena Kochhar
3	102	Lex De Haan
4	103	Alexander Hunold
5	104	Bruce Ernst
6	105	David Austin

DISTINCT 를 이용한 중복값 제거

```
SELECT DISTINCT job_id, department_id
FROM employees;
```

SQL | 인출된 모든 행: 20(0.002초)

	JOB_ID	DEPARTMENT_ID
1	IT_PROG	60
2	FI_MGR	100
3	PU_MAN	30
4	ST_MAN	50
5	SA_REP	80
6	AD_VP	90
7	AD_ASST	10

SELECT – 자료 검색

2. 행의 데이터 값 조회 방법 – WHERE 조건절 이용

조건과 일치하는 값 검색하기(비교 연산)

```
SELECT *
FROM employees
--WHERE first_name = 'David';
WHERE employee_id <> 105;
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
1	105	David	Austin	DAUSTIN
2	151	David	Bernstein	DBERNSTE
3	165	David	Lee	DLEE

특정 문자가 포함된 데이터 검색 – LIKE 연산자

```
SELECT *
FROM employees
WHERE job_id LIKE 'AD%';
```

EMPLOYEE_ID	JOB_ID	MANAGER_ID
109	FI_ACCOUNT	108
110	FI_ACCOUNT	108
111	FI_ACCOUNT	108
112	FI_ACCOUNT	108
113	FI_ACCOUNT	108
206	AC_ACCOUNT	205

NULL 검색

```
SELECT employee_id, job_id, manager_id
FROM employees
--WHERE manager_id IS NULL;
WHERE manager_id IS NOT NULL;
```

EMPLOYEE_ID	JOB_ID	MANAGER_ID
100	AD_PRES	(null)

SELECT – WHERE 조건절

2. 행의 데이터 값 조회 방법 – WHERE 조건절 이용

논리 연산자 : BETWEEN~AND 와 IN

```
SELECT *
FROM employees
WHERE salary BETWEEN 10000 AND 20000;
```

EMPLOYEE_ID	FIRST_NAME	JOB_ID	SALARY
101	Neena	AD_VP	17000
102	Lex	AD_VP	17000
108	Nancy	FI_MGR	12008
114	Den	PU_MAN	11000
145	John	SA_MAN	14000

```
SELECT first_name, hire_date, job_id, salary
FROM employees
WHERE salary IN (10000, 17000, 24000);
```

FIRST_NAME	HIRE_DATE	JOB_ID	SALARY
Steven	2003/06/17	AD_PRES	24000
Neena	2005/09/21	AD_VP	17000
Lex	2001/01/13	AD_VP	17000
Peter	2005/01/30	SA_REP	10000
Janette	2004/01/30	SA_REP	10000
Harrison	2006/03/23	SA_REP	10000
Hermann	2002/06/07	PR_REP	10000

SELECT – 자료 검색

- employees 테이블 복사

```
-- employees 테이블 복사(특정 칼럼) --
CREATE TABLE t_employee AS
SELECT employee_id, first_name, last_name, salary, job_id
FROM employees;

SELECT * FROM t_employee;
```

```
-- first_name이 'David'이고, salary가 4000을 초과하는 직원
SELECT * FROM t_employee
WHERE first_name = 'David'
AND salary > 4000;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	JOB_ID
105	David	Austin	4800	IT_PROG
151	David	Bernstein	9500	SA_REP
165	David	Lee	6800	SA_REP

SELECT – 자료 검색

- 실습 문제

아래의 실행 화면처럼 salary(급여)가 7000을 초과하고, job_id(직무)가 'IT' 나 'FI'로 시작하는 직원을 검색하세요.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	JOB_ID
103	Alexander	Hunold	9000	IT_PROG
108	Nancy	Greenberg	12008	FI_MGR
109	Daniel	Faviet	9000	FI_ACCOUNT
110	John	Chen	8200	FI_ACCOUNT
111	Ismael	Sciarra	7700	FI_ACCOUNT
112	Jose Manuel	Urman	7800	FI_ACCOUNT
113	Luis	Popp	6900	FI_ACCOUNT

INSERT : 행 삽입하기

INSERT INTO 테이블 이름(열이름1, 열이름2...)

VALUES (데이터 값1, 데이터값 2, ...)

예제1. 1행 추가하기

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	271	Sample Dept	200	1700
2	270	Payroll	(null)	1700
3	260	Recruiting	(null)	1700
4	250	Retail Sales	(null)	1700
5	240	Government Sales	(null)	1700

```
-- 자료 검색
SELECT * FROM t_department
ORDER BY department_id DESC;

-- 자료 추가
INSERT INTO t_department VALUES (271, 'Sample_Dept', 200, 1700);

-- 롤백 - 명령 취소
ROLLBACK;

-- 명령 실행 완료 : 트랜잭션
COMMIT;
```

INSERT : 행 삽입하기

INSERT 후에는 COMMIT을 명령해야함.

명령을 취소 할때는 ROLLBACK 사용(단, COMMIT을 하기 전에 사용 가능)

롤백(ROLLBACK) 후 검색

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	270	Payroll	(null)	1700
2	260	Recruiting	(null)	1700
3	250	Retail Sales	(null)	1700
4	240	Government Sales	(null)	1700
5	230	IT Helpdesk	(null)	1700
6	220	NOC	(null)	1700
7	210	IT Support	(null)	1700
8	200	Operations	(null)	1700
9	190	Contracting	(null)	1700
10	180	Construction	(null)	1700

UPDATE : 행 갱신하기

UPDATE 테이블 이름

SET 열이름 1 = 데이터 값 1

[WHERE 조건식]

```
-- departments에서 department_name이 Sample_Dept인
-- manager_id를 201, location_id를 1800으로 변경하기
UPDATE departments
SET manager_id = 201,
    location_id = 1800
WHERE department_name = 'Sample_Dept';
```

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	271	Sample Dept	201	1800
2	270	Payroll	(null)	1700
3	260	Recruiting	(null)	1700
4	250	Retail Sales	(null)	1700
5	240	Government Sales	(null)	1700

UPDATE : 행 갱신하기

예제2. departments 테이블을 변경하기

- 실수로 전체를 업데이트 한 경우

```
-- departments에서 department_name이 Sample_Dept인
-- manager_id를 201, location_id를 1800으로 변경하기
UPDATE departments
SET manager_id = 201,
    location_id = 1800;
--WHERE department_name = 'Sample_Dept';

ROLLBACK;
```

이전으로 되돌아감

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	271	Sample Dept	201	1800
2	270	Payroll	201	1800
3	260	Recruiting	201	1800
4	250	Retail Sales	201	1800
5	240	Government Sales	201	1800

DELETE : 행 삭제하기

DELETE [FROM] 테이블 이름

[WHERE 조건식]

[ ] – 대괄호는 생략 가능

예제1.

department_name이 'Sample Dept'행 삭제하기

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	270	Payroll	(null)	1700
2	260	Recruiting	(null)	1700
3	250	Retail Sales	(null)	1700
4	240	Government Sales	(null)	1700
5	230	IT Helpdesk	(null)	1700

DELETE : 행 삭제하기

```
21  DELETE FROM departments
22  WHERE department_name = 'Sample_Dept';
23
24  -- 전체 데이터 삭제하기
25  DELETE FROM departments;
```

스크립트 출력 x | 질의 결과 x

작업이 완료되었습니다.(0.022초)

```
명령의 25 행에서 시작하는 중 오류 발생 -
DELETE FROM departments
오류 보고 -
ORA-02292: 무결성 제약조건(HR.EMP_DEPT_FK)이 위배되었습니다- 자식 레코드가 발견되었습니다
```

INSERT : 행 삽입하기

INSERT 예제2.

```
-- ex3_1 테이블 생성 --
CREATE TABLE ex3_1(
    col1    VARCHAR2(10),
    col2    NUMBER,
    col3    DATE
);
```

```
-- 테이블(해당칼럼), 칼럼에 들어갈 값
INSERT INTO ex3_1(col1, col2, col3)
VALUES ('ABC', 10, SYSDATE);

-- 해당칼럼을 생략한 경우 --
INSERT INTO ex3_1 VALUES ('DEF', 20, SYSDATE);

-- 칼럼을 일부만 사용한 경우 --
INSERT INTO ex3_1(col1, col2)
VALUES ('GHI', 30);
```

	COL1	COL2	COL3
1	ABC	10	2020/12/26
2	DEF	20	2020/12/26
3	GHI	30	(null)

DELETE : 행 삭제하기

예제2. commit 명령어 확인하기

```
select * from tab;  -- 테이블 검색
select * from t_department;
```

```
C:₩Users₩김기용>sqlplus

SQL*Plus: Release 11.2.0.2.0 Production on 목 2월 25 08:06:06 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Enter user-name: hr
Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> select * from tab;

TNAME                                                        TABTYPE
------------------------------------------------------------ -------------
 CLUSTERID
----------
COUNTRIES                                                    TABLE

DEPARTMENTS                                                  TABLE

EMPLOYEES                                                    TABLE
```

```
SQL> select * from ex3_1;

no rows selected
```

데이터 삽입 후 검색하면 반영이 안되고, commit을 해야 반영됨

UPDATE : 행 갱신하기

예제1. ex3_1 테이블 데이터 변경하기

```
-- col2의 값을 50으로 변경 --
UPDATE ex3_1
SET col2 = 50;

-- col3의 null을 찾아 현재 날짜로 변경하기
UPDATE ex3_1
SET col3 = SYSDATE
WHERE col3 IS NULL;
```

	COL1	COL2	COL3
1	ABC	50	2020/12/26
2	DEF	50	2020/12/26
3	GHI	50	2020/12/27

```
SQL> select * from ex3_1;

COL1                 COL2 COL3
-------------------- ---------- --------
ABC                        10 21/02/25
DEF                        20 21/02/25
GHI                        30 21/02/25
```

INSERT ~ SELECT: 행 삽입하기

INSERT ~ SELECT 구문

```
-- ex3_2 테이블 생성 --
CREATE TABLE ex3_2(
    emp_id  NUMBER,
    emp_name    VARCHAR2(20)
);
```

	EMP_ID	EMP_NAME
1	100	King
2	101	Kochhar
3	102	De Haan
4	103	Hunold
5	104	Ernst
6	108	Greenberg
7	109	Faviet
8	110	Chen
9	111	Sciarra
10	112	Urman
11	113	Popp
12	114	Raphaely
13	120	Weiss

```
-- INSERT ~ SELECT --
-- 다른 테이블이나 뷰의 조회 결과로 나온 테이터를
-- 또 다른 테이블에 넣는 형식
INSERT INTO ex3_2(emp_id, emp_name)
SELECT employee_id, last_name
FROM employees
WHERE salary > 5000;
```

스크립트 출력 × | 질의 결과 ×

작업이 완료되었습니다.(0.028초)

58개 행 이(가) 삽입되었습니다.

의사 칼럼 – ROWNUM

의사 칼럼(Pseudo-column)이란

테이블의 칼럼처럼 동작하지만 실제로 테이블에 저장되지 않는 칼럼을 말한다.

- ROWNUM : 쿼리에서 반환되는 각 로우들에 대한 순서값을 나타낸다.

<실습 예제>

```
SELECT ROWNUM, employee_id
    FROM employees
WHERE ROWNUM < 5;

-- ROWNUM은 SELECT만 가능하다.(삭제, 수정 불가)
DELETE FROM employees WHERE ROWNUM = 4;
```

ROWNUM	EMPLOYEE_ID
1	100
2	101
3	102
4	103

0개 행 이(가) 삭제되었습니다.

의사 칼럼 – ROWNUM

- **게시판에서 ROWNUM 이용하기**

  페이지 처리를 할때 게시글이 삭제되므로 순번(SEQUENCE)으로는 계산할 수 없으므로 ROWNUM을 사용 할 수 있다.

```
SELECT ROWNUM, t_board.* FROM t_board;

SELECT ROWNUM, t_board.* FROM t_board ORDER BY regDate DESC;
```

ROWNUM	BNUM	TITLE	CONTENT
19	29	python	python
18	25	java	java
17	24	c	c
16	19	3월 마지막 주	오늘은 월요일..
15	18	주말이다	fdfdf
14	15	고양이	고양이
13	14	소	소
12	13	강아지	강아지
11	11	대나무	대나무
10	10	소나무	소나무
9	9	매화	매화
8	8	녹련	녹련

서브 쿼리

서브 쿼리

```
-- 부서 location_id가 1800인 부서 id를 가지는 사원 --
SELECT *
    FROM employees a,
         departments b
    WHERE a.department_id = b.department_id
        AND b.location_id = 1800;

-- 서브 쿼리 : 단일 (자료)행인 경우--
SELECT * FROM employees A
    WHERE A.department_id = (SELECT B.department_id
                               FROM departments B
                              WHERE B.location_id = 1800);
```

	DEPARTMENT_ID
1	20

LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	DEPARTMENT_ID_1
Hartstein	MHARTSTE	515.123.5555	04/02/17	MK_MAN	13000	(null)	100	20	20
Fay	PFAY	603.123.6666	05/08/17	MK_REP	6000	(null)	201	20	20

서브 쿼리

서브 쿼리

```
-- 서브쿼리 : 다중 자료(행)인 경우 --
SELECT * FROM employees A
    WHERE A.department_id IN (SELECT B.department_id
                                FROM departments B
                               WHERE B.location_id = 1700);
```

	DEPARTMENT_ID
1	10
2	30
3	90
4	100
5	110
6	120
7	130
8	140
9	150
10	160
11	170
12	180
13	190
14	200

LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
Whalen	JWHALEN	515.123.4444	03/09/17	AD ASST	4400	(null)	101	10
Raphaely	DRAPHEAL	515.127.4561	02/12/07	PU MAN	11000	(null)	100	30
Khoo	AKHOO	515.127.4562	03/05/18	PU CLERK	3100	(null)	114	30
Baida	SBAIDA	515.127.4563	05/12/24	PU CLERK	2900	(null)	114	30
Tobias	STOBIAS	515.127.4564	05/07/24	PU CLERK	2800	(null)	114	30
Himuro	GHIMURO	515.127.4565	06/11/15	PU CLERK	2600	(null)	114	30
Colmenares	KCOLMENA	515.127.4566	07/08/10	PU CLERK	2500	(null)	114	30
King	SKING	515.123.4567	03/06/17	AD PRES	24000	(null)	(null)	90
Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD VP	17000	(null)	100	90
De Haan	LDEHAAN	515.123.4569	01/01/13	AD VP	17000	(null)	100	90