

sql class 1

DBMS-데이터베이스 관리 시스템

-데이터를 저장,검색,관리하고 조직화하는데 사용되는 컴퓨터 시스템

1.데이터 베이스의 유형

1)관계형 데이터베이스

-테이블 형태로 데이터를 저장.

-SQL을 사용하여 데이터에 접근.

ex)MySQL,MariaDB,Oracle.....

2)NoSQL 데이터베이스

-여러 유형의 데이터 구조를 허용하며,테이블 간의 관계를 구성하지 않음

-대용량의 분산 데이터를 처리하는데 유용.

2.데이터베이스의 구성 요소

1)스키마

-데이터베이스의 구조와 제약 조건을 정의

2)테이블

-행과 열로 이루어진 데이터 저장구조.

-열:데이터의 속성

-행:데이터의 레코드

3)인덱스

-데이터 검색 속도를 높이기 위해 사용되는 데이터 구조

4)뷰

-실제로 데이터를 저장하지 않지만 기본 테이블로부터 데이터를 가져와 접근을 단순화 하거나 보안 목적으로 특정 데이터에 대한 접근을 제한 하는데 사용됨

5)제약조건

-데이터의 무결성을 유지하기 위해 테이블에 적용되는 규칙

ex)NOT NULL,CHECK,UNIQUE....

SQL(Structured Query Language)

-관계형 데이터베이스 관리 시스템(RDBMS)에서 데이터를 검색,수정,추가 및 삭제하기 위해 사용되는 표준 프로그래밍 언어.

-관계형 데이터 베이스에서 데이터를 조작하고 관리하기 위한 명령어와 구문 제공

1.데이터 검색

-SELECT문을 사용하여 특정테이블에서 원하는 데이터를 검색하거나 조 회할 수 있음

-필터링,정렬,그룹화,결합 등 다양한 옵션을 사용하여 원하는 결과를 얻을 수 있다.

2.데이터삽입

-INSERT문을 사용하여 테이블에 새로운 레코드(행)를 추가할 수 있다.

-각 열에 대한 값을 지정하거나,다른 테이블로부터 값을 복사할 수도 있다.

3.데이터 수정

-UPDATE 문을 사용하여 테이블의 기존 레코드(행)을 수정할 수 있다.

4.데이터 삭제

-DELETE 문을 사용하여 테이블에서 레코드(행)를 삭제할 수 있다.

5.테이블 및 데이터베이스 구조조작

-CREATE,ALTER,DROP문을 사용하여 데이터베이스,테이블,인덱스,뷰 등의 구조를 생성,수정,삭제할 수 있다.

6.트랜잭션 제어

-논리적 작업 단위

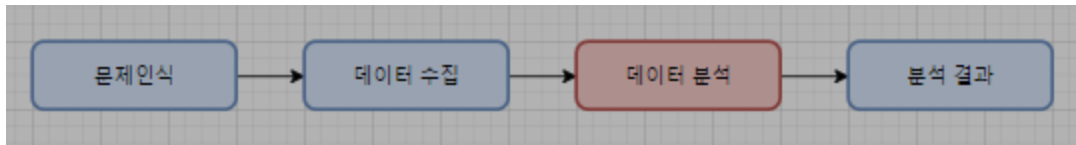
-COMMIT,ROLLBACK,SAVEPOINT등의 명령어로 데이터의 일관성을 유지.

7.데이터베이스 접근 제어

-GRANT,REVOKE명령어를 사용하여 사용자에게 데이터베이스 객체에 대한 권한을 부여하거나 취소할 수 있다.

-데이터 보안 유지.

데이터 분석과정



데이터를 수집하고 가공하는 과정을 특히 전처리라고 하며 데이터 전체 분석 과정의 약 80%를 차지한다. **pre-processing**

정형 데이터와 비정형 데이터

1.정형 데이터

-틀이 잡힌 데이터 엑셀 등등

2.비정형 데이터

-틀이 없는 다양하고 방대한 데이터,트위터,음악,그림 등등...

SELECT,데이터 조회

가장 많이 사용되는 구문

기본형

SELECT 열이름 FROM 테이블이름;

SELECT 열이름1,열이름2..

FROM 테이블이름;

SELECT 열이름1,열이름2..

FROM 테이블이름;

주의사항

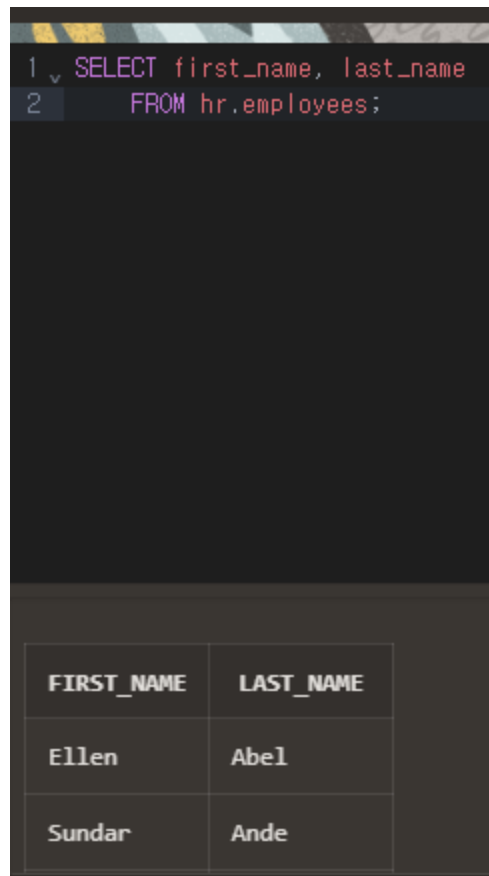
1. SQL 구문은 대문자로 사용하자.(소문자도 가능하지만 관용적 표현)
2. 줄바꿈은 특별히 제한하는 것이 없다.
3. 마지막에 ; 으로 마쳐야만 한다.

```
1 SELECT *
2 FROM hr.employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	24000

employees 테이블에서 모든(*) 전체 데이터를 보여달라는 의미

일부열 조회



The screenshot shows a SQL query being executed in a terminal. The query is: `SELECT first_name, last_name FROM hr.employees;`. Below the query, the result is displayed as a table with two columns: `FIRST_NAME` and `LAST_NAME`. The table contains two rows of data: `Ellen` and `Abel` in the first row, and `Sundar` and `Ande` in the second row.

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande

정렬 조회 ORDER BY

SELECT 열이름1,열이름2..

FROM 테이블이름

ORDER BY 열이름 DESC(내림차순 → 최근이 먼저) or ASC(오름차순)

;

중복 값 제거하고 고유값만 출력 DISTINCT

```

1 v SELECT job_id
2 FROM hr.employees;
3

```

AC_MGR
AD_ASST
AD_PRE
AD_VP
AD_VP
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT

```

1 v SELECT DISTINCT job_id
2 FROM hr.employees;
3

```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRE
AD_VP
FI_ACCOUNT
FI_MGR
HR_REP

별명(별칭) 사용하기-ALIAS/AS

- 별명은 열이름을 **임시**로 보여줄때만 바꾸어서 표현한다.
- 즉 원래 열이름 자체는 바꾸지 않는다.
- 이름을 바꿀 때 AS를 사용하며 생략도 가능하다.(반드시 넣을때도 있다.)

-특히, 빈칸/특수기호/대소문자를 넣을때는 반드시""를 사용해야만 한다.

```
SELECT first_name AS 성, last_name AS 이름
FROM hr.employees;
```

성	이름
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida
Amit	Banda
Elizabeth	Bates

```
SELECT first_name AS 성, last_name AS "이름"
FROM hr.employees;
```

성	이름
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida
Amit	Banda
Elizabeth	Bates

열 값들을 하나로 출력하기 || (버티컬 바)/연결연산

```

1 SELECT first_name||' '|| last_name
2 FROM hr.employees;
3

```

FIRST_NAME ' ' LAST_NAME
Ellen Abel
Sundar Ande
Mozhe Atkinson
David Austin
Hermann Baer
Shelli Baida
Amit Banda
Elizabeth Bates

```

, SELECT first_name||' '|| last_name AS "Name",
      email||'@naver.com' AS "Email"
FROM hr.employees;

```

Name	Email
Ellen Abel	EABEL@naver.com
Sundar Ande	SANDE@naver.com
Mozhe Atkinson	MATKINSO@naver.com
David Austin	DAUSTIN@naver.com
Hermann Baer	HBAER@naver.com
Shelli Baida	SBAIDA@naver.com
Amit Banda	ABANDA@naver.com
Elizabeth Bates	EBATES@naver.com

열값 개선하여 출력하기 산술연산 +-*/


```

SELECT employee_id,salary,
       salary+500 추가급여,
       salary*1.1 보너스
FROM hr.employees;

```

EMPLOYEE_ID	SALARY	추가급여	보너스
100	24000	24500	26400
101	17000	17500	18700
102	17000	17500	18700
103	9000	9500	9900
104	6000	6500	6600
105	4800	5300	5280
106	4800	5300	5280
107	4200	4700	4620


WHERE 조건 절

-where 절은 반드시 from 다음에 사용한다.

SELECT 열이름1, 열이름2

FROM 테이블이름

WHERE 조건;

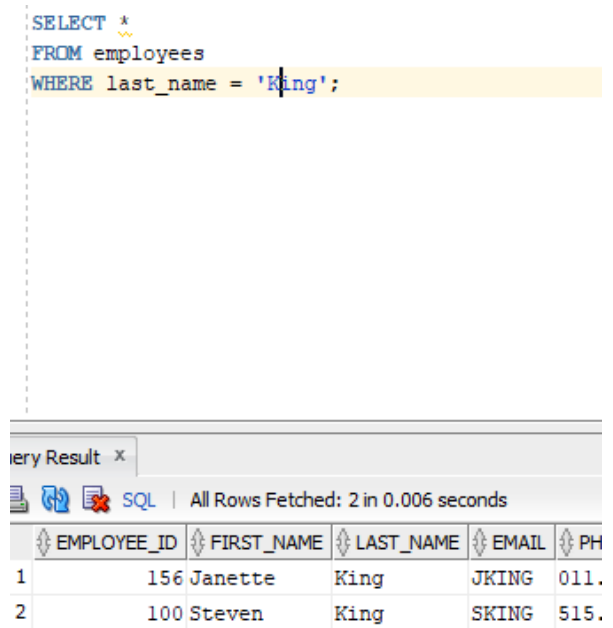
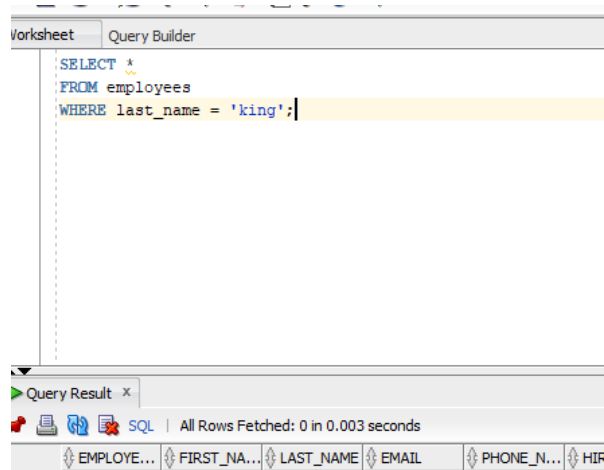


```
SELECT *
FROM employees
WHERE employee_id = 100;
```

Query Result x

SQL | All Rows Fetched: 1 in 0.017 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000



SELECT 는 select로 대소문자를 가리지는 않지만 찾을 값에서는 'king'과 'King'을 다르게 취급한다.

복잡한 조건 적용하기

BETWEEN a AND b

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

Query Result x

SQL | All Rows Fetched: 18 in 0.01 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000
2	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000
3	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-02	FI_MGR	12000
4	114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-02	PU_MAN	11000
5	145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-04	SA_MAN	14000
6	146	Karen	Partners	KPARTNER	011.44.1344.467268	05-JAN-05	SA_MAN	13500
7	147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-MAR-05	SA_MAN	12000
8	148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	15-OCT-07	SA_MAN	11000
9	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-08	SA_MAN	10500
10	150	Peter	Tucker	PTUCKER	011.44.1344.129268	30-JAN-05	SA_REP	10000
11	156	Janette	King	JKING	011.44.1345.429268	30-JAN-04	SA_REP	10000
12	162	Clara	Vishney	CVISHNEY	011.44.1346.129268	11-NOV-05	SA_REP	10500
13	168	Lisa	Ozer	LOZER	011.44.1343.929268	11-MAR-05	SA_REP	11500
14	169	Harrison	Bloom	HBLOOM	011.44.1343.829268	23-MAR-06	SA_REP	10000




IN(1,2,3,4)

Worksheet

Query Builder

```
SELECT *  
FROM employees  
WHERE salary IN(10000,13000);
```

Query Result x

 SQL | All Rows Fetched: 5 in 0.026 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	150	Peter	Tucker	PTUCKER	011.44.1344.129268	30-JAN-05	SA_REP	10000
2	156	Janette	King	JKING	011.44.1345.429268	30-JAN-04	SA_REP	10000
3	169	Harrison	Bloom	HBLOOM	011.44.1343.829268	23-MAR-06	SA_REP	10000
4	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	MK_MAN	13000
5	204	Hermann	Baer	HBAER	515.123.8888	07-JUN-02	PR REP	10000


LIKE '%도' → 도로 끝나는 모든 값 찾기 ex)경기도/제주도

LIKE '%@%' → 포함 하기만 하면

서울% → 서울로 시작하는 모든값

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

Query Result x

 All Rows Fetched: 4 in 0.008 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST
2	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES
3	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP
4	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP

_는 글자수를 뜻함 아래는 ____이므로 AD로 시작하고 뒤에 3글자

```

SELECT *
FROM employees
WHERE job_id LIKE 'AD___';

```

Query Result x

SQL | All Rows Fetched: 2 in 0.004 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP
2	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP

IS NULL → null값 찾

```

SELECT *
FROM employees
WHERE manager_id IS NULL;

```

Query Result x

SQL | All Rows Fetched: 1 in 0.006 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID
1	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(null)	(null)