



class day 1

☰ 태그	
▼	
📅 Dates	
☰ Topic	
▼ Type	
📅 날짜	@2023년 5월 8일

▼ DBMS

- DataBase이란 데이터를 저장, 검색, 관리하고 조직화하는데 사용되는 컴퓨터 시스템이다. DataBase Management System에 의해 관리되면 사용자나 프로그램이 데이터에 접근할 수 있도록 해준다.

▼ 데이터베이스 유형

- 크게 두가지 유형으로 분류된다. **관계형 데이터베이스**(Relational Database)와 NoSql(Not Only SQL)로 분류
- **관계형 데이터베이스**: 테이블(행과 열로 이루어진) 형태로 데이터 저장하며, 데이터베이스는 일반적으로 ACID(Atomicity, Consistency, Isolation, Durability) 속성 유지한다. 대표적 관계형 DB로는 MYSQL, PostgreSQL, Oracle Database, Microsoft SQL server 등이 있다.

- **NoSQL 데이터베이스**: 관계형 데이터베이스와는 달리, 여러 유형의 데이터 구조 허용하며 테이블 간 고나께 구성하지 않는다. NoSQL 데이터베이스는 대용량의 분산 데이터 처리에 유용하며 확장성이 높다. 주로 다음 네가지 유형으로 분류 된다. 문서형(Document), 키 값(Key value), 컬럼 패밀리(Column Family), 그래프(Graph). 대표적인 NoSQL DB로는 MongoDB, Redis, Cassandra, Neo4j 등이 있다.

이 외에도 계층형(Hierarchical), 망형(Network), 객체지향형(Object-Oriented), 객체관계형(Object-Relational) 등 있다.

▼ DB 구성요소

데이터베이스는 다음과 같은 주요 구성 요소로 이뤄져있다.

- **Schema**: DB의 구조와 제약 조건 정의한다. 테이블, 인덱스, 뷰, 프로시저 등 객체를 포함한다.
- **Table**: 행과 열로 이루어진 데이터 저장 구조로 각 열은 데이터 속성을, 각 행은 데이터 레코드를 나타낸다.
- **Index**: 데이터 검색 속도를 높이기 위해 사용되는 데이터 구조이다. 일종의 DB '색인'으로 특정 영리 값에 따라 레코드를 빠르게 찾을 수 있게 한다.
- **View**: DB의 가상 테이블로 실제로 데이터를 저장하지 않지만 기본 테이블로부터 데이터를 참조하고 질의를 수행한다.
- **Transaction**: DB에서 실행되는 하나의 논리적 작업 단위이다. 트랜잭션은 데이터 일관성과 동시성을 유지하기 위해 ACID 속성을 따른다.
- **Stored Procedure**: DB 서버에서 실행 되는 일련의 SQL 명령어로 구성된 코드 블록이다. 저장 프로시저는 코드 재사용, 성능 향상, 보안 향상과 같은 이점 제공한다.
- **Trigger**: 특정 이벤트(INSERT, UPDATE, DELETE 등)가 발생할때 자동으로 실행되는 사용자 정의 코드이다. 트리거는 데이터의 일고나성을 유지하거나, 추가 로직을 적용하는데 사용된다.
- **Constraint**: 데이터의 무결성 유지를 위해 테이블에 적용되는 규칙이다. NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK 등이 있다.
- **DBMS**: DB를 생성하고 조작하며 고나리하는 SW이다. 사용자와 프로그램이 DB와 상호 작용 할 수 있게 한다. 관계형 DBMS 와 NOSQL DBMS로 나뉜다.

데이터베이스의 구성 요소는 위에서 설명한 것드 외에도 많은 다른 요소들이 있지만, 이들은 DB를 이해하는데 가장 중요한 기본 개념이다. 이 요소들은 서로 상호작용하며, 데이터를 저장, 관리하며 사용자와 프로그램이 필요한 데이터에 접근하고 처리할 수 있게 해준다.

▼ DBMS 종류

DBMS	제작사	작동 운영체제	최신 버전	기타
MySQL	Oracle	Unix, Linux, Windows, Mac	8.0	오픈 소스(무료), 상용
MariaDB	MariaDB	Unix, Linux, Windows	10.6	오픈 소스(무료), MySQL 초기 개발자들이 독립해서 만듦
PostgreSQL	PostgreSQL	Unix, Linux, Windows, Mac	12	오픈 소스(무료)
Oracle	Oracle	Unix, Linux, Windows	18c	상용 시장 점유율 1위
SQL Server	Microsoft	Windows	2019	주로 중/대형급 시장에서 사용
DB2	IBM	Unix, Linux, Windows	11	메인프레임 시장 점유율 1위
Access	Microsoft	Windows	2019	PC용
SQLite	SQLite	Android, iOS	3	모바일 전용, 오픈 소스(무료)

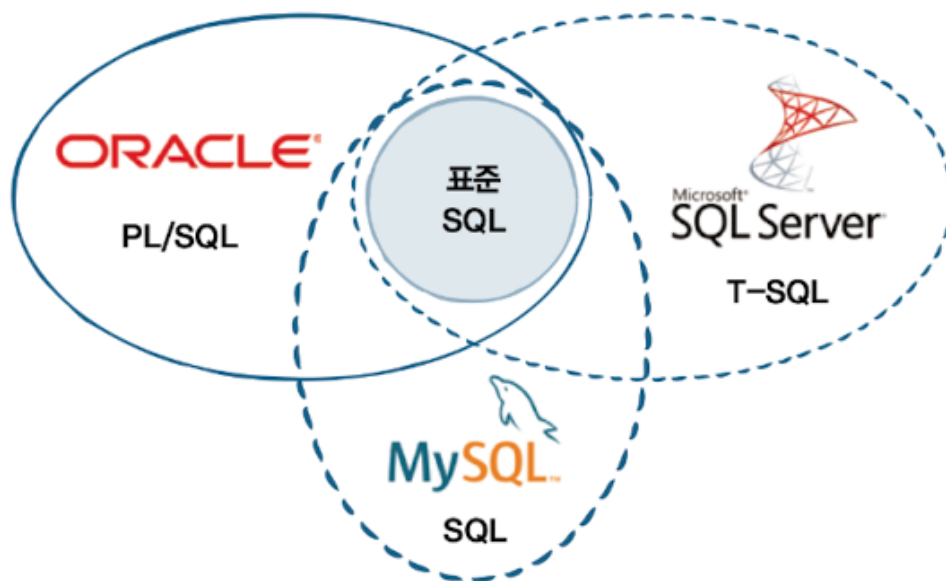
▼ SQL?

SQL(Structure Query Language)는 관계형 **DBMS**에서 데이터를 검색, 수정, 추가, 및 삭제하기 위해 사용되는 **표준 프로그래밍 언어**이다. 관계형 DB에서 데이터 조작 및 관리를 위한 명령어와 구문을 제공한다. SQL은 다양한 RDBMS 제품(ex: Oracle, MYSQL, PostgreSQL, Microsoft SQL sever 등)에서 사용되면 ANSI(미국 국제 표준 협회)및 ISO(국제 표준화 기구)에 의해 표준화 되었다.

SQL은 주로 다음과 같은 종류의 작업을 수행한다.

1. **데이터 검색(조회)**: SELECT문을 사용해 DB의 특정 테이블에서 원하는 데이터를 검색하거나 조회할 수 있다. 필터링, 정렬, 그룹화 결합 등 다양한 옵션을 사용해 원하는 결과 얻을 수 있다.
2. **데이터 삽입**: INSERT문을 사용하여 테이블에 새로운 레코드(행)를 추가 할 수 있다. 각 열에 대한 값을 지정하거나, 다른 테이블로부터 값을 복사할 수도 있다.

3. **데이터 수정**: UPDATE문을 사용해 테이블의 기존 레코드(행)를 수정할 수 있다. 조건을 지정해 특정 레코드만 삭제할 수 있다.
4. **데이터 삭제**: DELETE문 사용해 테이블에서 레코드(행)를 삭제할 수 있다. 조건을 지정해 데이터베이스, 테이블, 인덱스, 뷰 등의 구조를 생성, 수정, 삭제 가능하다.
5. **테이블 및 DB구조 조작**: CREATE, ALTER, DROP문을 사용해 데이터베이스, 테이블, 인덱스, 뷰 등의 구조를 생성, 수정, 삭제할 수 있다.
6. **트랜잭션 제어**: 트랜잭션은 데이터베이스의 논리적 작업 단위로 COMMIT, ROLLBACK, SAVEPOINT 등의 명령어를 사용해 트랜잭션을 제어하고 데이터의 일관성을 유지할 수 있다.
7. **DB접근 제어**: GRANT, REVOKE 명령어를 사용해 사용자에게 데이터베이스 객체에 대한 권한을 부여하거나 취소할 수 있다. 이를 통해 데이터의 보안 유지 가능하다.



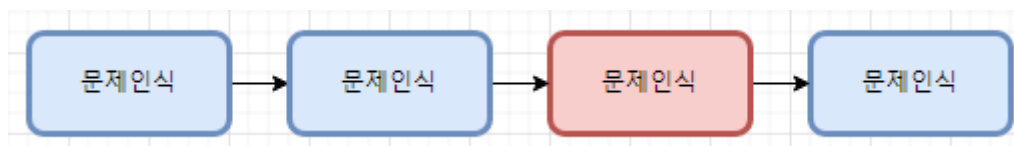
▼ Key Point

- **데이터베이스**는 데이터의 집합이며, DBMS는 데이터베이스를 운영/관리하는 프로그램을 가리킨다.
- **테이블**은 데이터베이스의 최소 단위로 하나 이상의 열과 행으로 구성되어 있다.
- **SQL**은 데이터베이스를 구축, 관리하고 활용하기 위해 사용되는 언어이다.

데이터베이스	Database	DB	데이터의 저장소
데이터베이스 관리 시스템	Database Management System	DBMS	데이터베이스를 운영/관리하는 소프트웨어
구조화된 질의 언어	Structured Query Language	SQL	관계형 DBMS에서 사용되는 언어
관계형 데이터베이스 관리 시스템	Relational Database Management System	RDBMS	테이블이라는 최소 단위로 구성된 DBMS
테이블	table		표 형태로 구성된 2차원 구조, 열과 행으로 이루어져 있으며 RDBMS의 핵심 개체
표준 SQL	Standard SQL		국제표준화기구에서 지정하는 SQL의 표준, 대부분의 DBMS 회사가 표준 SQL을 준수함

▼ DataBase

데이터 분석과정



그림그리는 툴

> app.diagrams.net

데이터를 수집하고 가공하는 과정을 특히 전처리라고 하며 데이터 전체 분석 과정의 약 80% 차지

= Pre - processing

▼ 정형 데이터와 비정형 데이터

정형 데이터: 틀이 잡힌 데이터, 엑셀 등

비정형 데이터: 틀이 없는 다양하고 방대한 데이터, 트위터, 음악, 그림 등

▼ 구문

*SELECT, 데이터 조회가장 많이 사용되는 구문

• 기본형

SELECT 열이름1, 열이름2

FROM 테이블이름;

SELECT 열이름1, 열이름2

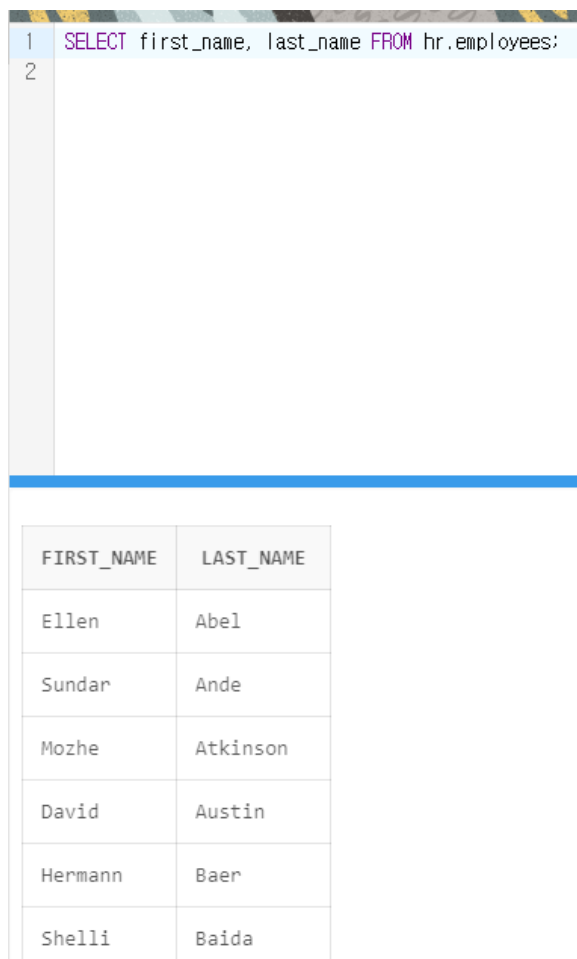
FROM 테이블이름;

○ 주의사항

- 1) SQL 구문은 대문자로 사용하자 (관용적 표현, 소문자로 사용 가능)
- 2) 줄바꿈은 특별히 제한하는 것이 없다.
- 3) 마지막에 ; 세미콜론으로 마쳐야만 한다.

▼ 실행 이미지

*hr.employee는 온라인이기에 앞에 붙임



```
1 SELECT first_name, last_name FROM hr.employees;
2
```

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida

1	SELECT first_name, last_name
2	FROM hr.employees;
3	

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Neena	Kochhar

전체열 조회

1	SELECT *
2	FROM hr.employees;
3	

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
101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000

employees 테이블에서 모든(*) 전체 데이터를 보여달라는 의미

일부열 조회

```
1 v 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

- 정렬 조회 ORDER BY

SELECT *

FROM 테이블 이름

ORDER BY 열이름 ASC or DESC;

▼ 실행 이미지

ASC 오름차순(예전이 먼저) , DESC 내림차순(최근이 먼저)

*적지 않으면 오름차순으로 출력

1	SELECT *
2	FROM hr.employees
3	ORDER BY employee_id DESC ;
4	

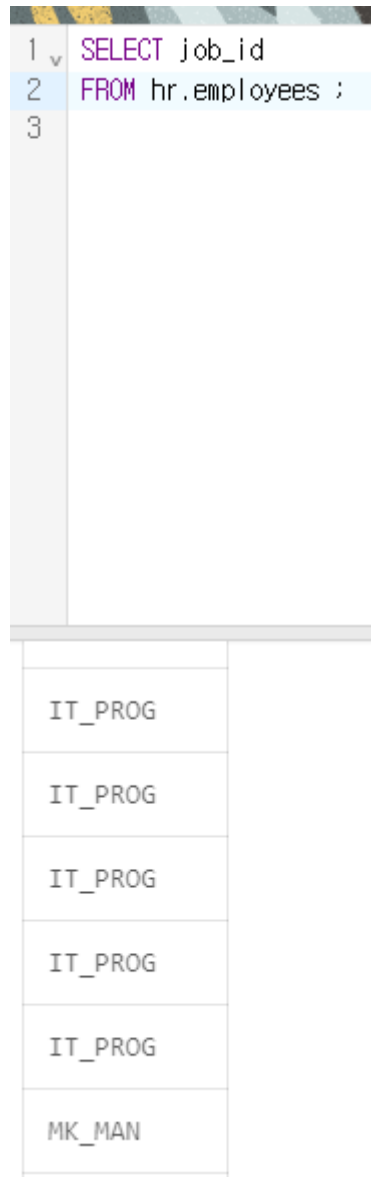
EMPLOYEE_ID	FIRST_NAME	LAST_NAME
206	William	Gietz
205	Shelley	Higgins
204	Hermann	Baer
203	Susan	May
202	Pat	Fay
201	Michael	Hart

- 중복 값 제거하고 고유 값 만 출력 DISTINCT

SELECT DISTINCT 열 이름

FROM 테이블 이름 ;

▼ 실행 이미지



The image shows a screenshot of a SQL query editor. The query is: `SELECT job_id FROM hr.employees ;`. The results are displayed in a table below the query. The table has one column, `job_id`, and contains six rows of data. The first five rows are `IT_PROG` and the last row is `MK_MAN`. This demonstrates the effect of the `DISTINCT` keyword, which removes duplicate values.

job_id
IT_PROG
IT_PROG
IT_PROG
IT_PROG
IT_PROG
MK_MAN

Before

1	SELECT DISTINCT job_id
2	FROM hr.employees ;
3	

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PREP
AD_VP
FI_ACCOUNT

after

- 별명(별칭) 사용하기 ALIAS/AS

SELECT 열 이름 AS 지정 이름

FROM 테이블 이름 ;

- 별명은 열이름을 임시로 보여줄때만 바꾸어 표현한다
- 즉 원래 열이름 자체는 바꾸지 않는다.
- 이름을 바꿀때 AS를 사용하며 생략 가능하다.(반드시 넣을때도 있음)

- 특히 빈칸/특수기호/대소문자 등 넣을 시 반드시 "" 를 사용해야한다.

▼ 실행 이미지



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

성	이름
Ellen	Abel
Sundar	Ande

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

성	이름
Ellen	Abel
Sundar	Ande

- 열 값들을 하나로 출력하기 || 버티컬 바 / 연결연산자

SELECT 열 이름 || 열 이름

FROM 테이블 이름 ;

▼ 실행 이미지

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

FIRST_NAME ' ' LAST_NAME
Ellen Abel
Sundar Ande
Mozhe Atkinson

```
1 SELECT first_name || last_name  
2 FROM hr.employees;
```

FIRST_NAME LAST_NAME

EllenAbel

SundarAde

MozheAtkinson

1	SELECT first_name ' ' last_name,
2	email '@ inc.com'
3	
4	FROM hr.employees;

FIRST_NAME ' ' LAST_NAME	EMAIL '@INC.COM'
Ellen Abel	EABEL@ inc.com
Sundar Aude	SANDE@ inc.com
Mozhe Atkinson	MATKINSO@ inc.com

- 열 값 계산하여 출력하기 산술연산 ... + -*/

SELECT 열 이름 + 100

FROM 테이블 이름 ;

*기존 값이 바뀌는 것이 아님

▼ 실행 이미지


```
1 ✓ SELECT employee_id, salary, salary+500, salary-100, salary*1.1|
2 FROM hr.employees;
```

EMPLOYEE_ID	SALARY	SALARY+500	SALARY-100	SALARY*1.1
100	24000	24500	23900	26400
101	17000	17500	16900	18700
102	17000	17500	16900	18700

```

1  SELECT employee_id 사원번호,
2     salary 급여,
3     salary+500 "추가 급여",
4     salary-100 "인하 급여",
5     salary*1.1 보너스
6  FROM hr.employees;

```

사원번호	급여	추가 급여	인하 급여	보너스
100	24000	24500	23900	26400
101	17000	17500	16900	18700
102	17000	17500	16900	18700

- AS없이 명칭 지정 사용가능

- **WHERE 조건절**

SELECT 열 이름 1, 열 이름 2

FROM 테이블 이름

WHERE 조건;

- WHERE 절은 반드시 from 다음에 사용한다

▼ 실행 이미지

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

의 결과 x

SQL | 인출된 모든 행: 1(0,011초)

	EMPLOYEE_ID	FIRST_NAME	LAST.
1	100	Steven	King

SELECT는 select로 대소문자를 상관없지만, 찾을 값(King)은 대소문자 구별한다.(King/king)

```
SELECT *
FROM employees
WHERE last_name = "King";
```

질의 결과 x

SQL | 실행 중:SELECT

ORA-00904: "King": invalid identifier
00904, 00000 - "%s: invalid identifier"
*Cause:
*Action:
3행, 19열에서 오류 발생

```
SELECT *
FROM employees
WHERE last_name = 'King';
```

질의 결과 x

SQL | 인출된 모든 행: 2(0,014초)

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	
1	156	Janette	King	J
2	100	Steven	King	S

- 복잡한 조건 적용하기

```
SELECT *  
FROM 테이블 이름  
WHERE 열 이름 BETWEEN 값1 AND 값2 ;
```

```
SELECT *  
FROM 테이블 이름  
WHERE 열 이름 IN (값1, 값2) ;
```

```
SELECT *  
FROM 테이블 이름  
WHERE 열 이름 IS NULL ;
```

- LIKE '%도': 도로 끝나는 값 찾기 경기도/제주도, '%@%' 포함하기만 하면 '서울%' 서울시 구로 서울시 강남
*%는 몇글자 상관없이 글자 찾음
- IS NULL null 찾기

▼ 실행 이미지

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

질의 결과 x

SQL | 인출된 모든 행: 18(0,006초)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	101 Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000
2	102 Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	17000
3	108 Nancy	Greenberg	NGREENBE	515.124.4569	02/08/17	FI_MGR	12008
4	114 Den	Raphaely	DRAPHEAL	515.127.4561	02/12/07	PU_MAN	11000
5	145 John	Russell	JRUSSEL	011.44.1344.429268	04/10/01	SA_MAN	14000
6	146 Karen	Partners	KPARTNER	011.44.1344.467268	05/01/05	SA_MAN	13500
7	147 Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	05/03/10	SA_MAN	12000
8	148 Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	07/10/15	SA_MAN	11000
9	149 Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	08/01/29	SA_MAN	10500
10	150 Peter	Tucker	PTUCKER	011.44.1344.129268	05/01/30	SA_REP	10000
11	156 Janette	King	JKING	011.44.1345.429268	04/01/30	SA_REP	10000

```
SELECT *
FROM employees
WHERE salary IN(10000, 13000, 17000);
```

질의 결과 x

SQL | 인출된 모든 행: 7(0,005초)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
1	101 Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21
2	102 Lex	De Haan	LDEHAAN	515.123.4569	01/01/13
3	150 Peter	Tucker	PTUCKER	011.44.1344.129268	05/01/30
4	156 Janette	King	JKING	011.44.1345.429268	04/01/30
5	169 Harrison	Bloom	HBLOOM	011.44.1343.829268	06/03/23
6	201 Michael	Hartstein	MHARTSTE	515.123.5555	04/02/17
7	204 Hermann	Baer	HBAER	515.123.8888	02/06/07

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

질의 결과 x

SQL | 인출된 모든 행: 4(0,013초)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	200 Jennifer	Whalen	JWHALEN	515.123.4444	03/09/17	AD_ASST
2	100 Steven	King	SKING	515.123.4567	03/06/17	AD_PRES
3	101 Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP
4	102 Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP

<pre> SELECT * FROM employees WHERE job_id LIKE 'AD%'; </pre>																																															
<div>질의 결과 x</div> <div>SQL 인출된 모든 행: 4(0.003초)</div> <table> <tr> <th></th><th>EMPLOYEE_ID</th><th>FIRST_NAME</th><th>LAST_NAME</th><th>EMAIL</th><th>PHONE_NUMBER</th><th>HIRE_DATE</th><th>JOB_ID</th></tr> <tr> <td>1</td><td>200</td><td>Jennifer</td><td>Whalen</td><td>JWHALEN</td><td>515.123.4444</td><td>03/09/17</td><td>AD_ASST</td></tr> <tr> <td>2</td><td>100</td><td>Steven</td><td>King</td><td>SKING</td><td>515.123.4567</td><td>03/06/17</td><td>AD_PRES</td></tr> <tr> <td>3</td><td>101</td><td>Neena</td><td>Kochhar</td><td>NKOCHHAR</td><td>515.123.4568</td><td>05/09/21</td><td>AD_VP</td></tr> <tr> <td>4</td><td>102</td><td>Lex</td><td>De Haan</td><td>LDEHAAN</td><td>515.123.4569</td><td>01/01/13</td><td>AD_VP</td></tr> </table>									EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	1	200	Jennifer	Whalen	JWHALEN	515.123.4444	03/09/17	AD_ASST	2	100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	3	101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	4	102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID																																								
1	200	Jennifer	Whalen	JWHALEN	515.123.4444	03/09/17	AD_ASST																																								
2	100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES																																								
3	101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP																																								
4	102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP																																								

<pre> SELECT * FROM employees WHERE first_name LIKE '%a%'; </pre>																																			
<div>질의 결과 x</div> <div>SQL 50개의 행이 인출됨(0.013초)</div> <table> <tr> <th></th><th>EMPLOYEE_ID</th><th>FIRST_NAME</th><th>LAST_NAME</th></tr> <tr> <td>1</td><td>101</td><td>Neena</td><td>Kochhar</td></tr> <tr> <td>2</td><td>103</td><td>Alexander</td><td>Hunold</td></tr> <tr> <td>3</td><td>105</td><td>David</td><td>Austin</td></tr> <tr> <td>4</td><td>106</td><td>Valli</td><td>Pataballa</td></tr> <tr> <td>5</td><td>107</td><td>Diana</td><td>Lorentz</td></tr> <tr> <td>6</td><td>108</td><td>Nancy</td><td>Greenberg</td></tr> <tr> <td>7</td><td>109</td><td>Daniel</td><td>Faviet</td></tr> </table>					EMPLOYEE_ID	FIRST_NAME	LAST_NAME	1	101	Neena	Kochhar	2	103	Alexander	Hunold	3	105	David	Austin	4	106	Valli	Pataballa	5	107	Diana	Lorentz	6	108	Nancy	Greenberg	7	109	Daniel	Faviet
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME																																
1	101	Neena	Kochhar																																
2	103	Alexander	Hunold																																
3	105	David	Austin																																
4	106	Valli	Pataballa																																
5	107	Diana	Lorentz																																
6	108	Nancy	Greenberg																																
7	109	Daniel	Faviet																																

<pre> SELECT * FROM employees WHERE job_id LIKE 'AD__'; </pre>																															
<div>질의 결과 x</div> <div>SQL 인출된 모든 행: 2(0.004초)</div> <table> <tr> <th></th><th>EMPLOYEE_ID</th><th>FIRST_NAME</th><th>LAST_NAME</th><th>EMAIL</th><th>PHONE_NUMBER</th><th>HIRE_DATE</th><th>JOB_ID</th></tr> <tr> <td>1</td><td>101</td><td>Neena</td><td>Kochhar</td><td>NKOCHHAR</td><td>515.123.4568</td><td>05/09/21</td><td>AD_VP</td></tr> <tr> <td>2</td><td>102</td><td>Lex</td><td>De Haan</td><td>LDEHAAN</td><td>515.123.4569</td><td>01/01/13</td><td>AD_VP</td></tr> </table>									EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	1	101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	2	102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID																								
1	101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP																								
2	102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP																								

*언더바는 글자 갯수

SQL | 인출된 모든 행: 1(0,002초)

NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID
1	SKING	515.123.4567	03/06/17	AD_PRES	24000	(null)	(null)

새로 만들기/데이터베이스 접속 선택

접속 이름 접속 세부정보

Name Color ☐

데이터베이스 유형

사용자 정보 프록시 사용자

인증 유형

사용자 이름(U) 롤(L)

비밀번호(P) ☐ 비밀번호 저장(Y)

접속 유형(Y)

세부정보 고급

호스트 이름(A)

포트(B)

☒ SID(I)

☐ 서비스 이름(E)

상태: 성공