

# sql class 1

## [DBMS]

- 데이터베이스 : 데이터를 저장, 검색, 관리, 조직화할 때 사용하는 컴퓨터 시스템
- DBMS(Database Management System) : DB 관리 시스템

## [데이터베이스 유형]

1. 관계형 데이터베이스(Relational Database)
  - 테이블 형태로 데이터 저장
  - SQL을 사용해 데이터에 접근
  - Oracle, MySQL, MariaDB, etc...
2. NoSQL 데이터베이스(Not Only SQL)
3. 기타 : 계층형, 망형, 객체지향형, 객체관계형 등

## [데이터베이스 구성요소]

1. 테이블
  - 데이터베이스의 최소 단위
  - 행과 열 구조로 데이터 저장
    - 열 : 데이터의 속성
    - 행 : 데이터의 레코드
2. 뷰(View)
  - 데이터베이스의 가상테이블
    - ex) 학생 테이블과 성적 테이블을 끌어와 하나의 테이블처럼 사용
3. 제약 조건
  - 데이터 무결성 유지를 위해 테이블에 적용하는 규칙

- NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK 등

#### 4. 데이터베이스 관리시스템(DBMS)

- 데이터베이스를 생성, 조작, 관리하는 소프트웨어

### [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]

#### 1. SQL(Structured Query Language)

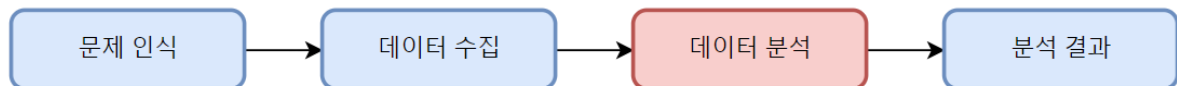
- **관계형** DBMS에서 데이터를 검색, 수정, 추가, 삭제하기 위해 사용하는 표준 프로그래밍 언어
- 구조화된 질의 언어

#### 2. 수행 작업

- 데이터 검색(조회) : SELECT 문
- 데이터 삽입 : INSERT 문
- 데이터 수정 : UPDATE 문

- 데이터 삭제 : DELETE 문
- 기타 : DROP, CREATE, ALTER 등

## [데이터 분석과정]



- 데이터 수집
  - **전처리**(pre-processing) : 데이터를 수집하고 가공하는 과정  
전체 데이터 분석 과정의 약 80% 차지
- 정형 데이터 / 비정형 데이터
  - 정형 데이터 : 틀이 잡힌 데이터. ex) 엑셀
  - 비정형 데이터 : 틀이 없는 다양하고 방대한 데이터. ex) 트위터, 음악, 그림

## [ SELECT ]

1. 데이터 조회, 가장 많이 사용되는 구문
2. 기본형

SELECT 열이름 FROM 테이블이름 ;

SELECT 열이름 1, 열이름2  
FROM 테이블이름;



## 주의사항

- 1) SQL 구문은 대문자로 사용할 것(관용적 표현, 소문자도 사용은 가능)
- 2) 줄바꿈, 들여쓰기는 특별한 제한이 없다
- 3) 마지막은 세미콜론(;)으로 마쳐야만 한다

## ▼ 실습예제

### [실습1] 전체열 조회

```
SELECT *  
FROM hr.employees;  
  
-- * : 전체열 조회  
-- employees 테이블에서 모든 전체 데이터를 보여줌
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	07/05/21	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	05/06/25	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	06/02/05	IT_PROG	4800	-	103	60

### [실습2] 줄바꿈, 들여쓰기, 일부열 조회

```
SELECT first_name, last_name  
FROM hr.employees;  
  
-- hr. : 온라인 상에서만 사용, local 환경에선 필요 X
```

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

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

## [ ORDER BY ]

### 1. 정렬 조회

### 2. 기본형

SELECT \*

FROM 테이블이름

**ORDER BY 열이름 ASC or DESC**

;



ASC : 오름차순 (예전이 먼저)

DESC : 내림차순(최근이 먼저)

### ▼ 실습예제

[실습3] 정렬 조회(내림차순)

```
SELECT *
FROM hr.employees
ORDER BY employee_id DESC;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
206	William	Gietz	WGIETZ	515.123.8181	02/06/07	AC_ACCOUNT	8300	-	205	110
205	Shelley	Higgins	SHIGGINS	515.123.8080	02/06/07	AC_MGR	12008	-	101	110
204	Hermann	Baer	HBAER	515.123.8888	02/06/07	PR_REP	10000	-	101	70
203	Susan	Mavris	SMAVRIS	515.123.7777	02/06/07	HR_REP	6500	-	101	40
202	Pat	Fay	PFAY	603.123.6666	05/08/17	MK_REP	6000	-	201	20
201	Michael	Hartstein	MHARTSTE	515.123.5555	04/02/17	MK_MAN	13000	-	100	20

## [ DISTINCT ]

- 중복값 제거 후 고유값만 출력

### ▼ 실습예제

[실습4] DISTINCT

```

1 SELECT job_id
2 FROM hr.employees;

```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
AD_VP
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_MGR
HR_REP

BEFORE

```

1 SELECT DISTINCT job_id
2 FROM hr.employees;

```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
MK_MAN
MK_REP
PR_REP

AFTER

## [ ALIAS / AS ]

1. 별명(별칭) 사용
2. 출력할 때만 임시로 열 이름을 바꿔서 보여줌 (실제 열 이름 변경 X)
3. AS
  - 이름 바꿀 때 사용
  - 생략 가능(반드시 넣어야 할 때도 있음)

- 특히 빈칸/특수기호/대소문자 등을 넣을 때는 반드시 ""를 사용

## ▼ 실습예제

### [실습5]

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

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

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

성	이   름
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida
Amit	Banda
Elizabeth	Bates
Sarah	Bell
David	Bernstein

### [실습6] 전체 열 이름 별명 적용

```
SELECT employee_id AS 사번,
       first_name AS 성,
       last_name AS 이름,
       email AS "메일 주소",
       phone_number AS 전화번호,
       hire_date AS 입사일,
```



```

    job_id AS 직무ID,
    salary AS 연봉,
    commission_pct AS 성과급지급률,
    manager_ID AS "관리자 ID",
    department_id AS 부서ID

FROM hr.employees;

```

사번	성	이름	메일 주소	전화번호	입사일	직무ID	연봉	성과급지급률	관리자 ID	부서ID
100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	07/05/21	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	05/06/25	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	06/02/05	IT_PROG	4800	-	103	60

## [ 연결연산자/버티컬 바(II) ]

- 열 값들을 하나로 출력

### ▼ 실습예제

[실습7] 공백 포함 문자 연결

```

SELECT first_name || ' ' || last_name
FROM hr.employees;

```

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

### [실습8]

```
SELECT first_name || ' ' || last_name,
       email || '@naver.com'
FROM hr.employees;
```

FIRST_NAME    ' '    LAST_NAME	EMAIL    '@NAVER.COM'
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
Sarah Bell	SBELL@naver.com
David Bernstein	DBERNSTE@naver.com

## [ 산술연산 ]

- 열 값 계산하여 출력 ( + - / \* 등 )

### ▼ 실습예제

#### [실습9]

```
SELECT employee_id 사원번호,
       salary 급여,
       salary+500 추가급여,
       salary-100 인하급여,
       salary*1.1 보너스
FROM hr.employees;
```

사원번호	급여	추가급여	인하급여	보너스
100	24000	24500	23900	26400
101	17000	17500	16900	18700
102	17000	17500	16900	18700
103	9000	9500	8900	9900
104	6000	6500	5900	6600
105	4800	5300	4700	5280
106	4800	5300	4700	5280
107	4200	4700	4100	4620
108	12008	12508	11908	13208.8
109	9000	9500	8900	9900
110	8200	8700	8100	9020

## [ WHERE 조건절 ]

1. 반드시 FROM 다음에 사용
2. 기본형

SELECT 열이름  
FROM 테이블이름

**WHERE 조건 ;**

### ▼ 실습예제

[실습10]

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

Query Result x											
SQL   All Rows Fetched: 1 in 0.016 seconds											
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(null)	(null)	

### [실습11-1]

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

-- 오류 발생 -> 작은 따옴표 사용해야함
```

Query Result x											
SQL   Executing: SELECT * FROM employees WHERE last_name = "King" in 0 seconds											
ORA-00904: "King": invalid identifier											
00904. 00000 - "%s: invalid identifier"											
*Cause:											
*Action:											
Error at Line: 3 Column: 19											

### [실습11-2]

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

Query Result x											
SQL   All Rows Fetched: 2 in 0.005 seconds											
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	156	Janette	King	JKING	011.44.1345.429268	30-JAN-04	SA_REP	10000			
2	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000			

### [실습11-3]

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

-- 대소문자 구분 -> 조회 결과 X

Query Result x

SQL | All Rows Fetched: 0 in 0.002 seconds

EMPLOYEE...	FIRST_NA...	LAST_NAME	EMAIL	PHONE_N...	HIRE_DATE	JOB_
-------------	-------------	-----------	-------	------------	-----------	------

\*\* SELECT는 select로 대소문자를 구분하지 않지만, 찾을 값에서는 구분함(King - king 구분)

[실습12]

```
SELECT *
FROM employees
WHERE salary >= 10000;
```

Query Result x

SQL | All Rows Fetched: 19 in 0.012 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION
1	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000	(
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000	(
4	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-02	FI_MGR	12008	(
5	114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-02	PU_MAN	11000	(
6	145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-04	SA_MAN	14000	
7	146	Karen	Partners	KPARTNER	011.44.1344.467268	05-JAN-05	SA_MAN	13500	
8	147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-MAR-05	SA_MAN	12000	
9	148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	15-OCT-07	SA_MAN	11000	
10	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-08	SA_MAN	10500	

### 3. 복잡한 조건 적용하기

- BETWEEN 1 AND 4
- IN(1, 2, 3, 4)

- LIKE '서울%' : '서울'로 시작하는 값 모두 조회. ex) 서울시 구로구, 서울시 강남구
- LIKE '%도' : '도'로 끝나는 값 모두 조회. ex) 경기도, 제주도
- LIKE '%@%' : '@'를 포함하는 값 모두 조회. 이메일 조회에 주로 사용
- IS NULL : NULL 찾기

## ▼ 실습예제

### [실습13] BETWEEN

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

Query Result x

All Rows Fetched: 18 in 0.031 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	12008
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

### [실습14] IN

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

-- = 사용하지 않도록 주의

Query Result x								
All Rows Fetched: 7 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	150	Peter	Tucker	PTUCKER	011.44.1344.129268	30-JAN-05	SA_REP	10000
4	156	Janette	King	JKING	011.44.1345.429268	30-JAN-04	SA_REP	10000
5	169	Harrison	Bloom	HBLOOM	011.44.1343.829268	23-MAR-06	SA_REP	10000
6	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	MK_MAN	13000
7	204	Hermann	Baer	HBAER	515.123.8888	07-JUN-02	PR_REP	10000

### [실습15] LIKE, %

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

Query Result x								
All Rows Fetched: 4 in 0.006 seconds								
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST	4000
2	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD PRES	24000
3	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000
4	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000

### [실습15-2] LIKE, \_(언더바)

```
SELECT *
FROM employees
WHERE job_id LIKE 'AD__';

/*
_ : 글자의 개수
*/
```

Query Result x								
All Rows Fetched: 2 in 0.007 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



## [실습16] IS NULL

```
SELECT *  
FROM employees  
WHERE manager_id IS NULL;
```



Query Result x

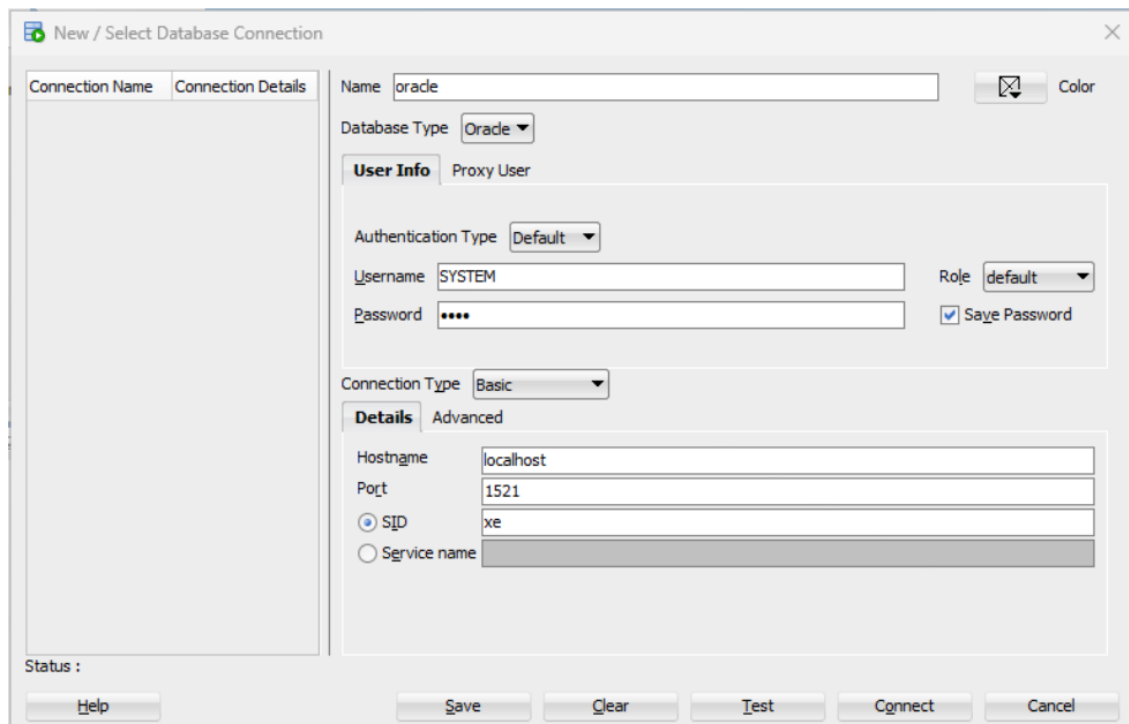
All Rows Fetched: 1 in 0.009 seconds

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

\*\* [참고] NULL값(결측치) 처리방법

- 1) NULL이 아닌 값들의 평균값으로 대체
- 2) 최빈값으로 대체

## ▼ Oracle



New / Select Database Connection

Connection Name: orade

Database Type: Oracle

User Info: Proxy User

Authentication Type: Default

Username: SYSTEM

Role: default

Password: \*\*\*\*

Save Password: ☒

Connection Type: Basic

Details: Advanced

Hostname: localhost

Port: 1521

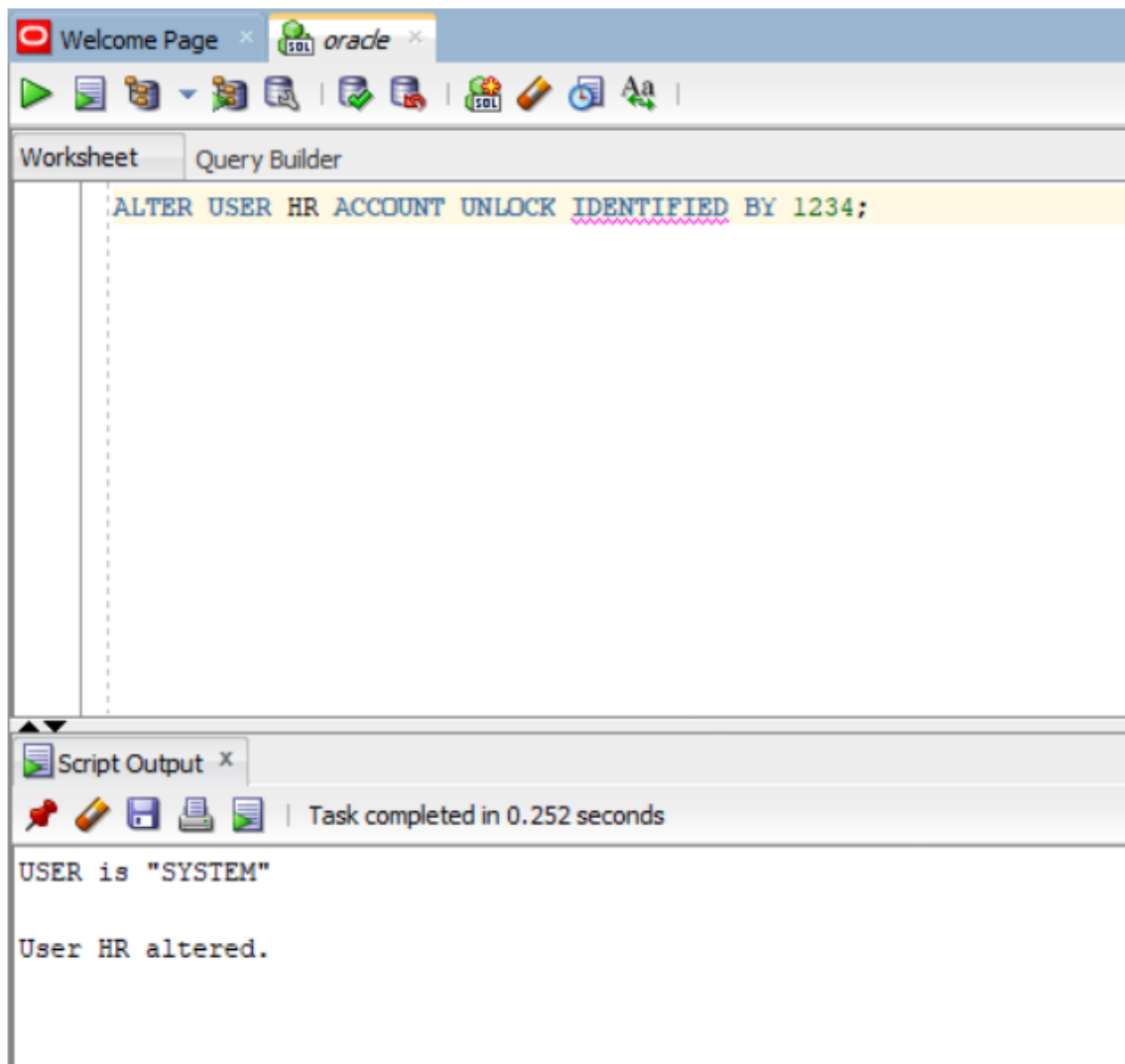
☒ SID: xe

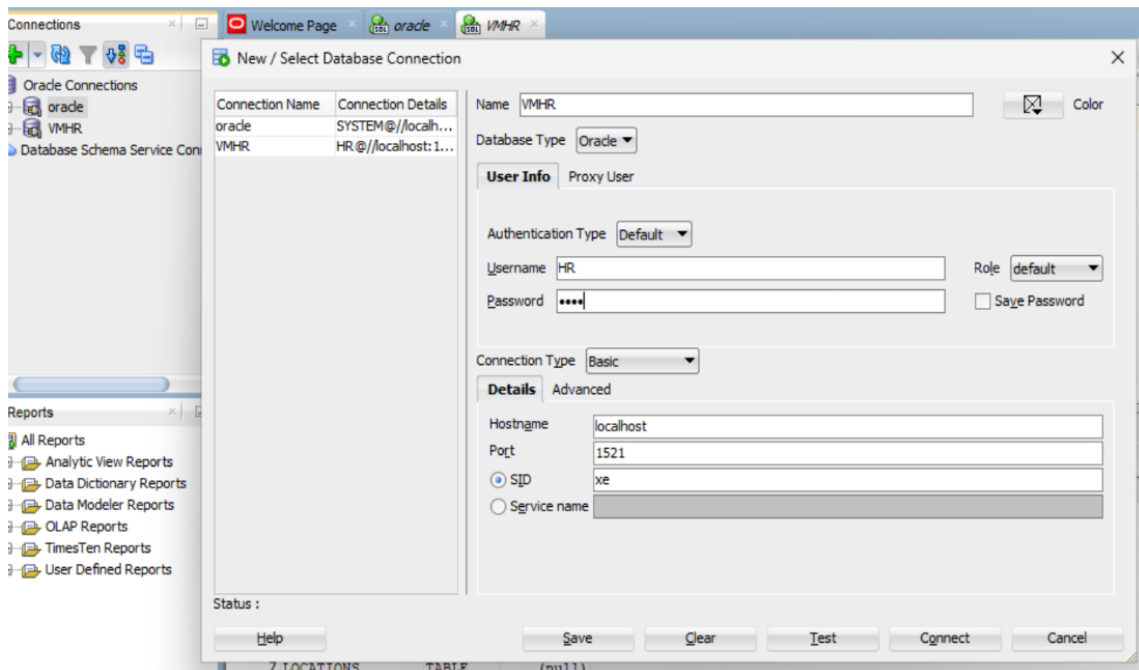
☐ Service name

Status:

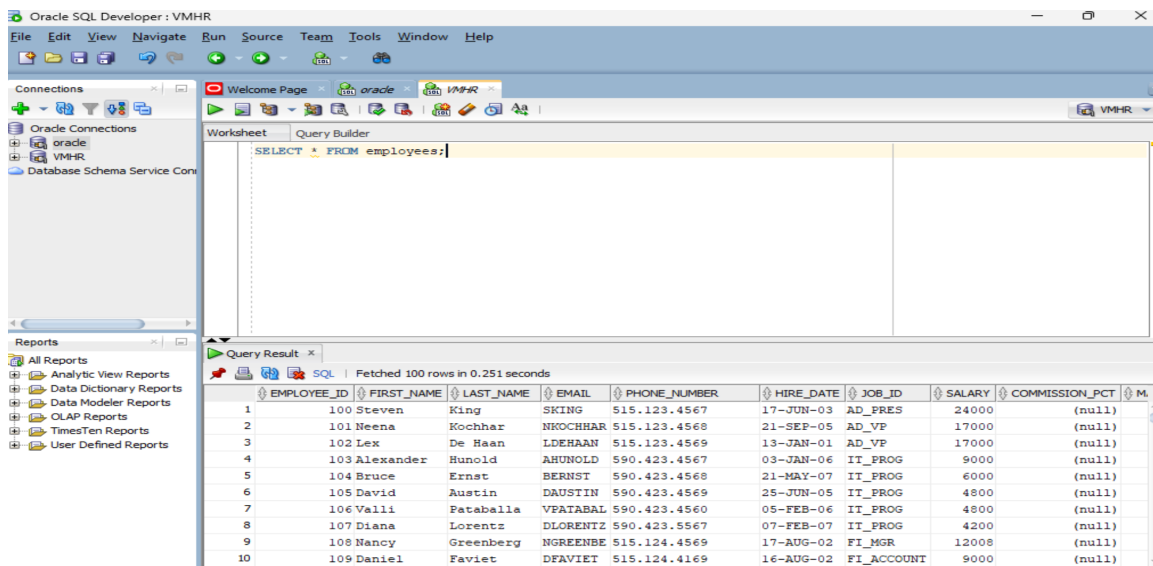
Help Save Clear Test Connect Cancel

- password : 0000
- TEST -> STATUS : SUCCESS 확인 -> CONNECT





- password : 1234
- TEST -> STATUS : SUCCESS 확인 -> CONNECT



- VM → Install VMware Tools → complete 다운

#### ▼ 기타

<https://livesql.oracle.com/> → 웹 브라우저를 통한 SQL 실습 사이트

<https://app.diagrams.net/> → Visualization 관련

<https://www.egovframe.go.kr/home/sub.do?menuNo=41> → eGovFrame  
+ VMWare PRO, Windows 11(iso), Oracle 11gR2, SQL Developer