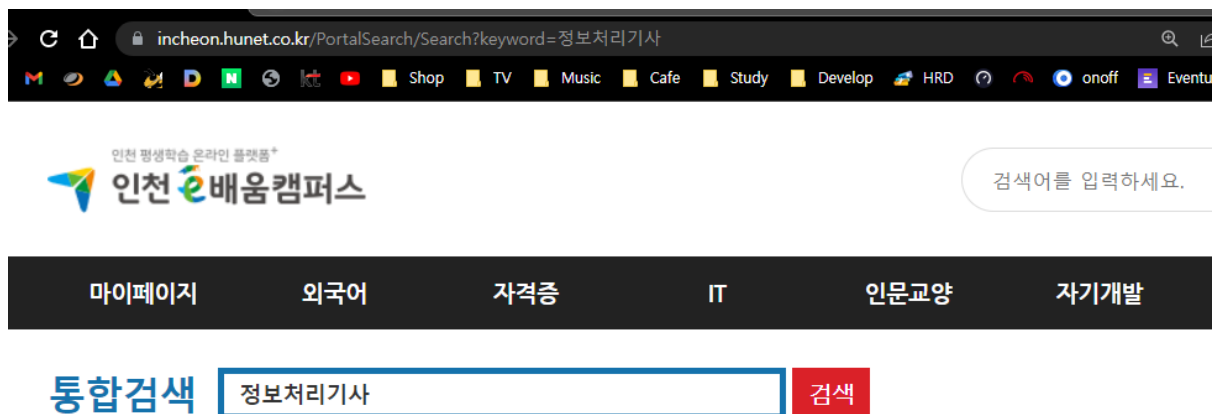


Class 1

정보처리기사 무료 동영상 사이트

인천e배움캠퍼스

<https://incheon.hunet.co.kr/PortalSearch/Search?keyword=정보처리기사>



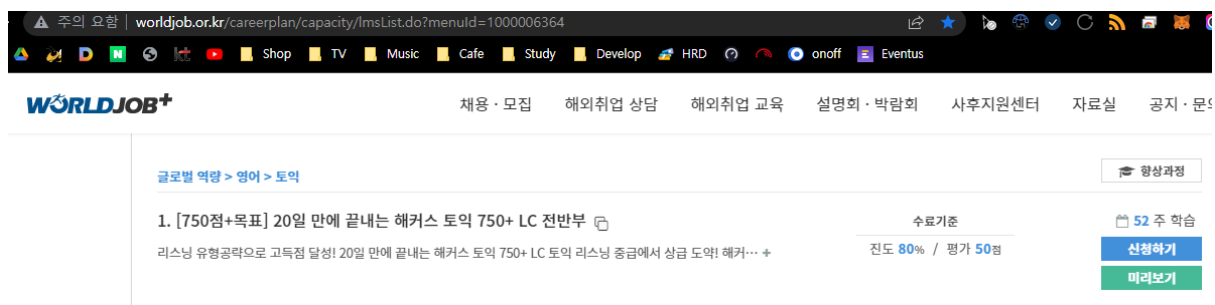
토익 무료 동영상 사이트

월드잡플러스

한국산업인력공단 운영, 해외취업, 해외진출정보, 해외채용공고, K-Move스쿨, 해외취업정착지원금 등

<http://www.worldjob.or.kr/careerplan/capacity/lmsList.do?menuId=1000006364>

WORLDJOB+



암호 jesus

```

alter session set nls_date_format='RR/MM/DD';

drop table emp;
drop table dept;

CREATE TABLE DEPT
  (DEPTNO number(10),
   DNAME VARCHAR2(14),
   LOC VARCHAR2(13) );

INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW YORK');
INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS');
INSERT INTO DEPT VALUES (30, 'SALES', 'CHICAGO');
INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');

CREATE TABLE EMP (
  EMPNO          NUMBER(4) NOT NULL,
  ENAME          VARCHAR2(10),
  JOB            VARCHAR2(9),
  MGR            NUMBER(4) ,
  HIREDATE       VARCHAR2(8),
  SAL            NUMBER(7,2),
  COMM           NUMBER(7,2),
  DEPTNO         NUMBER(2) );

INSERT INTO EMP VALUES (7839, 'KING', 'PRESIDENT', NULL, '81-11-17', 5000, NULL, 10);
INSERT INTO EMP VALUES (7698, 'BLAKE', 'MANAGER', 7839, '81-05-01', 2850, NULL, 30);
INSERT INTO EMP VALUES (7782, 'CLARK', 'MANAGER', 7839, '81-05-09', 2450, NULL, 10);
INSERT INTO EMP VALUES (7566, 'JONES', 'MANAGER', 7839, '81-04-01', 2975, NULL, 20);
INSERT INTO EMP VALUES (7654, 'MARTIN', 'SALESMAN', 7698, '81-09-10', 1250, 1400, 30);
INSERT INTO EMP VALUES (7499, 'ALLEN', 'SALESMAN', 7698, '81-02-11', 1600, 300, 30);
INSERT INTO EMP VALUES (7844, 'TURNER', 'SALESMAN', 7698, '81-08-21', 1500, 0, 30);
INSERT INTO EMP VALUES (7900, 'JAMES', 'CLERK', 7698, '81-12-11', 950, NULL, 30);
INSERT INTO EMP VALUES (7521, 'WARD', 'SALESMAN', 7698, '81-02-23', 1250, 500, 30);
INSERT INTO EMP VALUES (7902, 'FORD', 'ANALYST', 7566, '81-12-11', 3000, NULL, 20);
INSERT INTO EMP VALUES (7369, 'SMITH', 'CLERK', 7902, '80-12-11', 800, NULL, 20);
INSERT INTO EMP VALUES (7788, 'SCOTT', 'ANALYST', 7566, '82-12-22', 3000, NULL, 20);
INSERT INTO EMP VALUES (7876, 'ADAMS', 'CLERK', 7788, '83-01-15', 1100, NULL, 20);
INSERT INTO EMP VALUES (7934, 'MILLER', 'CLERK', 7782, '82-01-11', 1300, NULL, 10);

commit;

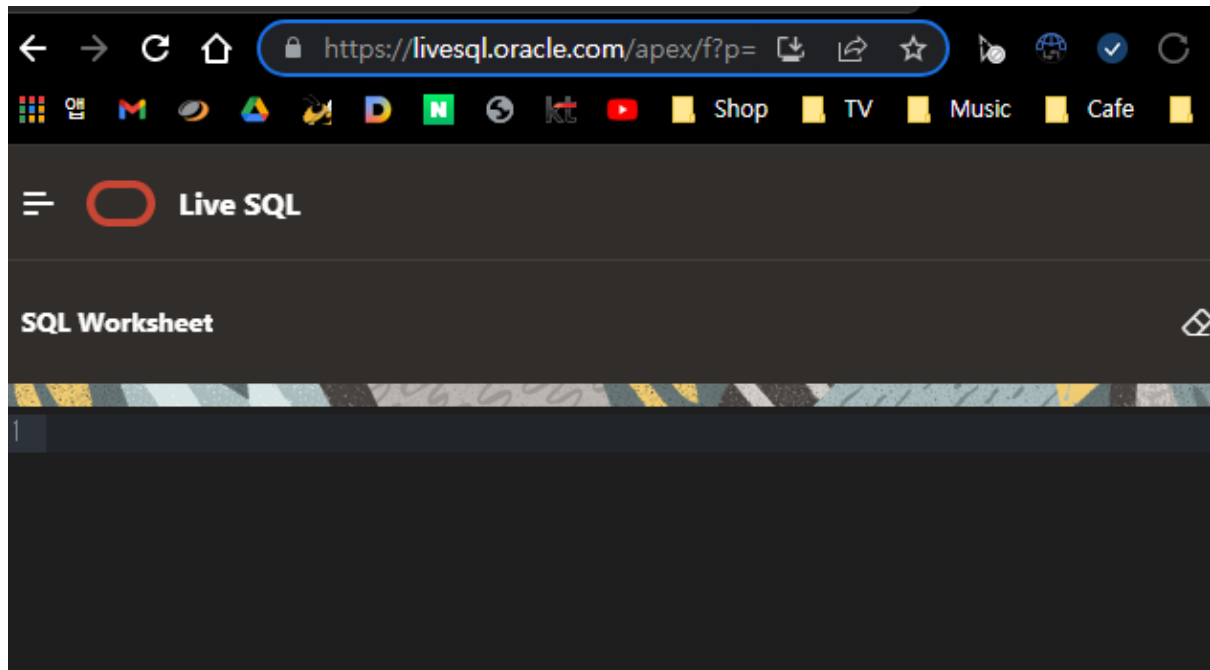
drop table salgrade;

create table salgrade
( grade   number(10),
  losal   number(10),
  hisal   number(10) );

insert into salgrade values(1,700,1200);
insert into salgrade values(2,1201,1400);
insert into salgrade values(3,1401,2000);
insert into salgrade values(4,2001,3000);
insert into salgrade values(5,3001,9999);

```

```
commit;
```



▼ DBMS ?

데이터베이스란 데이터를 저장, 검색, 관리하고 조직화하는데 사용되는 컴퓨터 시스템입니다. 데이터베이스는 일반적으로 데이터베이스 관리 시스템(Database Management System, DBMS)에 의해 관리되며, 사용자나 프로그램이 데이터에 접근할 수 있게 해줍니다.

1. 데이터베이스 유형:

데이터베이스는 크게 두 가지 유형으로 분류됩니다: **관계형 데이터베이스** (Relational Database)와 NoSQL(Not Only SQL) 데이터베이스.

- 관계형 데이터베이스: 테이블(행과 열로 이루어진) 형태로 데이터를 저장하며, 데이터베이스는 일반적으로 ACID(Atomicity, Consistency, Isolation, Durability) 속성을 유지합니다. 대표적인 관계형 데이터베이스로는 MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server 등이 있습니다.

- NoSQL 데이터베이스: 관계형 데이터베이스와는 달리, NoSQL 데이터베이스는 여러 유형의 데이터 구조를 허용하며, 테이블 간의 관계를 구성하지 않습니다. NoSQL 데이터베이스는 대용량의 분산 데이터를 처리하는데 유용하며, 확장성이 높습니다. NoSQL 데이터베이스는 주로 다음 네 가지 유형으로 분류됩니다: 문서형(Document), 키-값(Key-Value), 컬럼-패밀리(Column-Family), 그래프(Graph). 대표적인 NoSQL 데이터베이스로는 MongoDB, Redis, Cassandra, Neo4j 등이 있습니다.

이 외에도 계층형 Hierarchical, 망형 Network, 객체지향형 Object-Oriented, 객체 관계형 Object-Relational 등이 있습니다.

2. 데이터베이스 구성 요소:

데이터베이스는 다음과 같은 주요 구성 요소로 이루어져 있습니다.

- 스키마(Schema): 데이터베이스의 구조와 제약 조건을 정의합니다. 테이블, 인덱스, 뷰, 프로시저 등의 객체를 포함합니다.
- 테이블(Table): 행과 열로 이루어진 데이터 저장 구조로, 각 열은 데이터의 속성을, 각 행은 데이터의 레코드를 나타냅니다.
- 인덱스(Index): 데이터 검색 속도를 높이기 위해 사용되는 데이터 구조입니다. 일종의 데이터베이스 '색인'으로, 특정 열의 값에 따라 레코드를 빠르게 찾을 수 있게 합니다.
- 뷰(View): 데이터베이스의 가상 테이블로, 실제로 데이터를 저장하지 않지만 기본 테이블로부터 데이터를 참조하고 질의를 수행합니다. 뷰는 데이터 접근을 단순화하거나, 보안 목적으로 특정 데이터에 대한 접근을 제한하는 데 사용됩니다.
- 트랜잭션(Transaction): 데이터베이스에서 실행되는 하나의 논리적 작업 단위입니다. 트랜잭션은 데이터의 일관성과 동시성을 유지하기 위해 ACID 속성을 따릅니다.
- 저장 프로시저(Stored Procedure): 데이터베이스 서버에서 실행되는 일련의 SQL 명령어로 구성된 코드 블록입니다. 저장 프로시저는 코드 재사용, 성능 향상 및 보안 향상과 같은 이점을 제공합니다.
- 트리거(Triiger): 특정 이벤트(INSERT, UPDATE, DELETE 등)가 발생할 때 자동으로 실행되는 사용자 정의 코드입니다. 트리거는 데이터의 일관성을 유지하거나, 추가 로직을 적용하는 데 사용됩니다.
- 제약 조건(Constraint): 데이터의 무결성을 유지하기 위해 테이블에 적용되는 규칙입니다. NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK 등이 있습니다.

- 데이터베이스 관리 시스템(DBMS): 데이터베이스를 생성하고, 조작하며, 관리하는 소프트웨어입니다. 사용자와 프로그램이 데이터베이스와 상호 작용할 수 있게 합니다. 관계형 DBMS와 NoSQL DBMS로 나뉩니다.

데이터베이스의 구성 요소는 위에서 설명한 것들 외에도 많은 다른 요소들이 있지만, 이들은 데이터베이스를 이해하는 데 중요한 기본 개념입니다. 이러한 요소들은 서로 상호 작용하며, 데이터를 저장하고 관리하며, 사용자와 프로그램이 필요한 데이터에 접근하고 처리할 수 있게 해줍니다.

▼ 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)

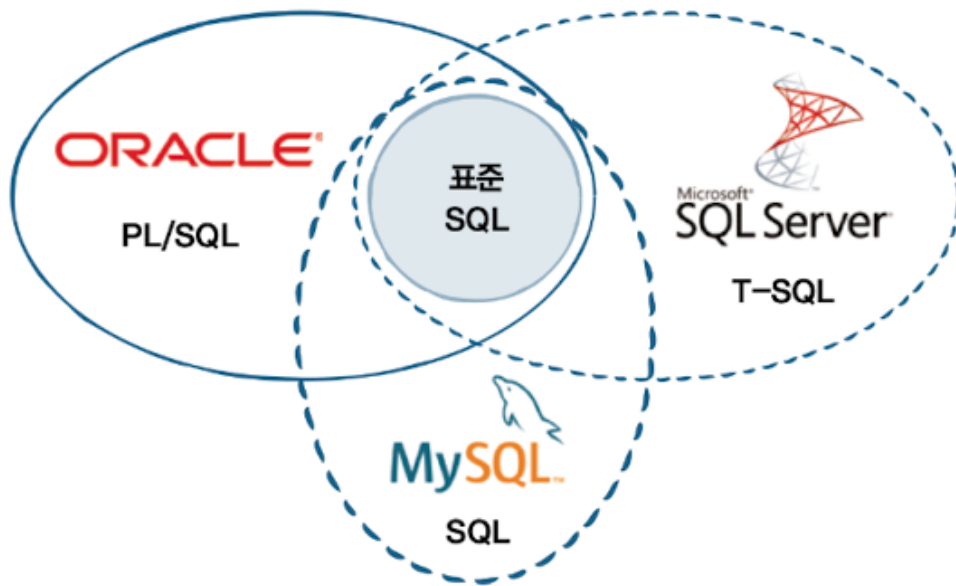
▼ SQL ?

SQL(Structured Query Language)은 관계형 데이터베이스 관리 시스템(RDBMS)에서 데이터를 검색, 수정, 추가 및 삭제하기 위해 사용되는 표준 프로그래밍 언어입니다. **관계형 데이터베이스**에서 데이터를 조작하고 관리하기 위한 명령어와 구문을 제공합니다. SQL은 다양한 RDBMS 제품(예: Oracle, MySQL, PostgreSQL, Microsoft SQL

Server 등)에서 사용되며, ANSI(미국 국립 표준 협회) 및 ISO(국제 표준화 기구)에 의해 표준화되었습니다.

SQL은 주로 다음과 같은 종류의 작업을 수행합니다:

1. 데이터 검색(조회): SELECT 문을 사용하여 데이터베이스의 특정 테이블에서 원하는 데이터를 검색하거나 조회할 수 있습니다. 필터링, 정렬, 그룹화, 결합 등 다양한 옵션을 사용하여 원하는 결과를 얻을 수 있습니다.
2. 데이터 삽입: INSERT 문을 사용하여 테이블에 새로운 레코드(행)를 추가할 수 있습니다. 각 열에 대한 값을 지정하거나, 다른 테이블로부터 값을 복사할 수도 있습니다.
3. 데이터 수정: UPDATE 문을 사용하여 테이블의 기존 레코드(행)를 수정할 수 있습니다. 조건을 지정하여 특정 레코드만 수정할 수도 있습니다.
4. 데이터 삭제: DELETE 문을 사용하여 테이블에서 레코드(행)를 삭제할 수 있습니다. 조건을 지정하여 특정 레코드만 삭제할 수도 있습니다.
5. 테이블 및 데이터베이스 구조 조작: CREATE, ALTER, DROP 문을 사용하여 데이터베이스, 테이블, 인덱스, 뷰 등의 구조를 생성, 수정, 삭제할 수 있습니다.
6. 트랜잭션 제어: 트랜잭션은 데이터베이스의 논리적 작업 단위로, COMMIT, ROLLBACK, SAVEPOINT 등의 명령어를 사용하여 트랜잭션을 제어하고 데이터의 일관성을 유지할 수 있습니다.
7. 데이터베이스 접근 제어: GRANT, REVOKE 명령어를 사용하여 사용자에게 데이터베이스 객체에 대한 권한을 부여하거나 취소할 수 있습니다. 이를 통해 데이터의 보안을 유지할 수 있습니다.



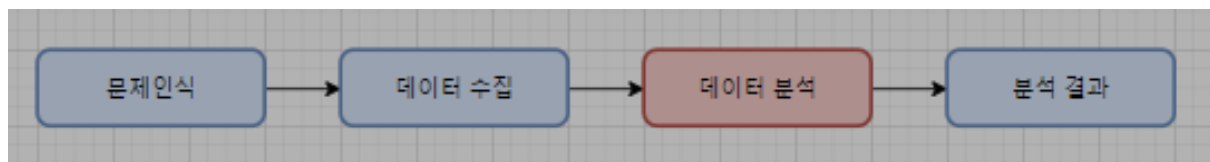
▼ Key Point

- **데이터베이스**는 데이터의 집합이며, **DBMS**는 데이터베이스를 운영/관리하는 프로그램을 말합니다.
- **테이블**은 데이터베이스의 최소 단위로, 하나 이상의 열과 행으로 구성되어 있습니다.
- **SQL**은 데이터베이스를 구축, 관리하고 활용하기 위해서 사용되는 언어입니다.

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



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

정형 데이터 와 비정형 데이터

정형 데이터 : 틀이 잡힌 데이터 엑셀 등등

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 (관용적인 표현, 소문자도 사용가능)
- 2 줄바꿈은 특별히 제한하는 것이 없다.
- 3 마지막에 ; 세미콜론으로 마쳐야만 한다.

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

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande

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

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande

전체열 조회

```
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

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

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ade

정렬 조회 ORDER BY

SELECT *

FROM 테이블이름

ORDER BY 열이름 ASC or DESC;

ASC 오름차순(예전이 먼저) DESC 내림차순 (최근이 먼저)

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
206	William	Gietz	WGIEZT	515.123.8181	02/06/2003
205	Shelley	Higgins	SHIGGINS	515.123.8080	02/06/2003
204	Hermann	Baer	HBAER	515.123.8888	02/06/2003
203	Susan	Mavris	SMAVRIS	515.123.7777	02/06/2003
202	Pat	Fay	PFAY	603.123.6666	05/08/2003

중복 값 제거하고 고유값만 출력 DISTINCT


```

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

```

성	이름
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer

```

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

```

성	이름
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer

열 값들을 하나로 출력하기 || 버티컬 바 / 연결연산자

```

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

```

FIRST_NAME LAST_NAME
EllenAbel
SundarAnde
MozheAtkinson
DavidAustin
HermannBaer

```

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

```

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

```

1 SELECT first_name||' '||last_name,
2        email||'@naver.com'
3 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

열값 계산하여 출력하기 산술연산 + - * /

```
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

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

사원번호	급여	추가급여	인하급여	보너스
100	24000	24500	23900	26400

Windows 11 on VMware

WORKSTATION



Create a New
Virtual Machine

What type of configuration do you want?

☒ Typical (recommended)
Create a Workstation 17.x virtual machine in a few easy steps.

☐ Custom (advanced)
Create a virtual machine with advanced options, such as a SCSI controller, virtual disk type and compatibility with older VMware products.

< Back Next >

New Virtual Machine Wizard

Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

☐ Installer disc:

DVD RW 드라이브 (E:)

☒ Installer disc image file (iso):

C:\Users\WSIST\Downloads\Win11_22H2_English_

Browse...

Windows 11 x64 detected.

☐ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

Help < Back Next > Cancel

Your evaluation period ends in 30

New Virtual Machine Wizard

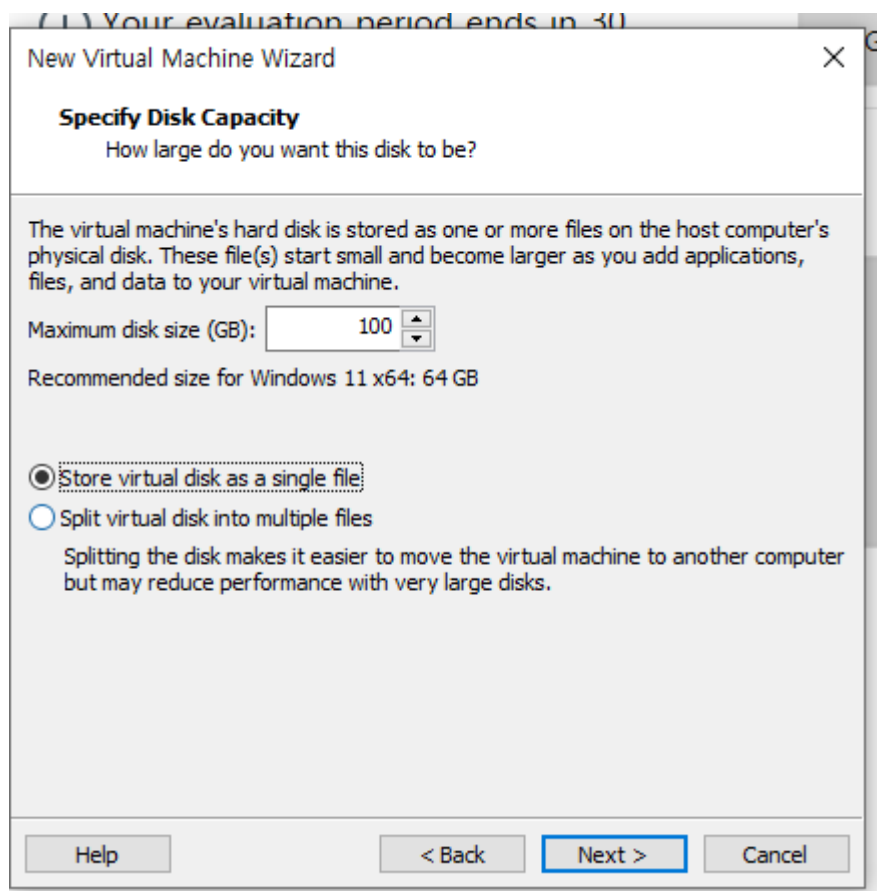
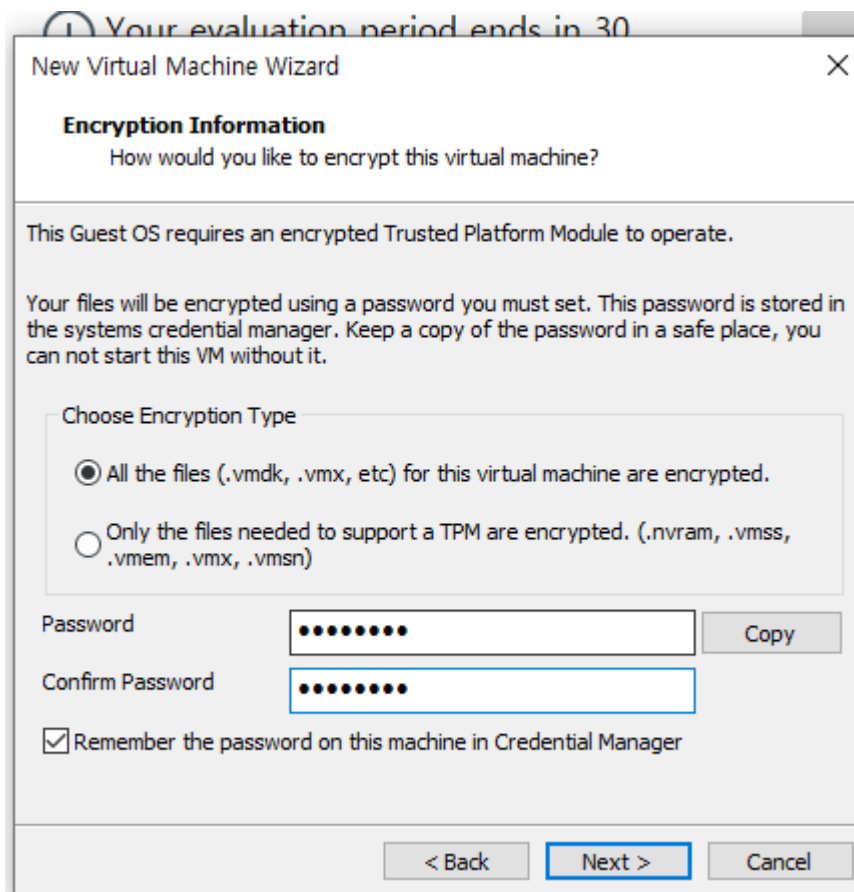
Name the Virtual Machine
What name would you like to use for this virtual machine?

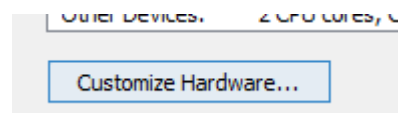
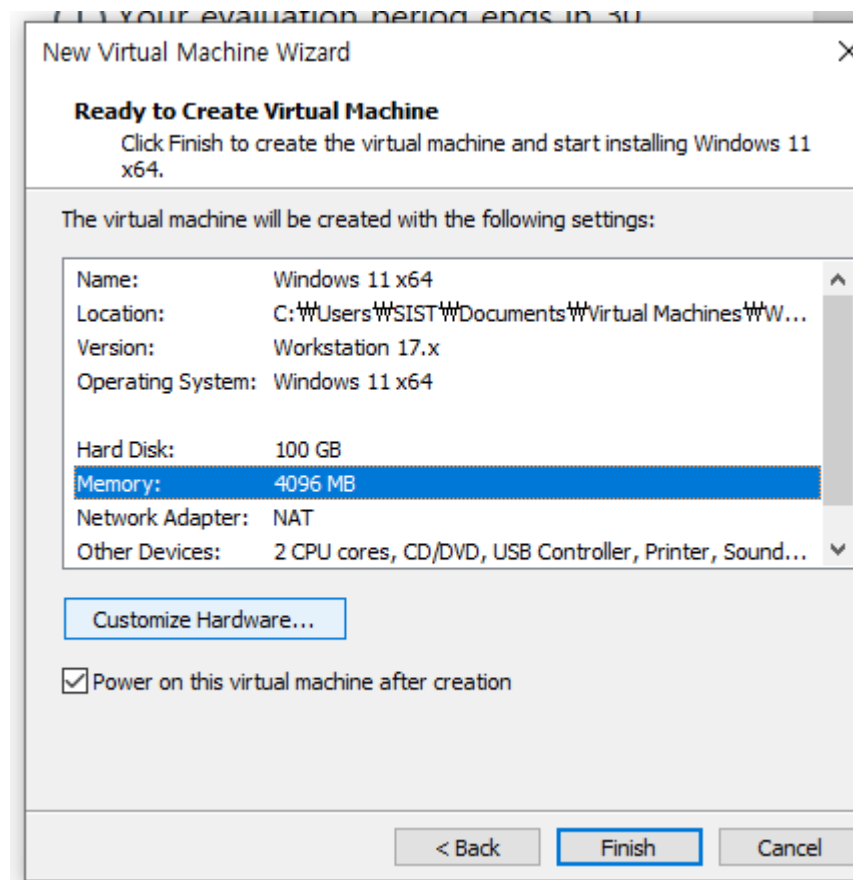
Virtual machine name:

Location:

The default location can be changed at Edit > Preferences.

< Back **Next >** Cancel





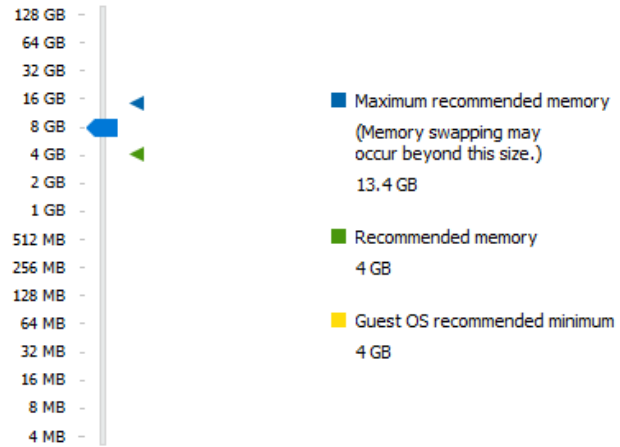
Hardware

Device	Summary
Memory	4 GB
Processors	2
New CD/DVD (SATA)	Using file C:\Users\WSIST...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: MB



Hardware

Device	Summary
Memory	8 GB
Processors	4
New CD/DVD (SATA)	Using file C:\Users\WSIST...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Processors

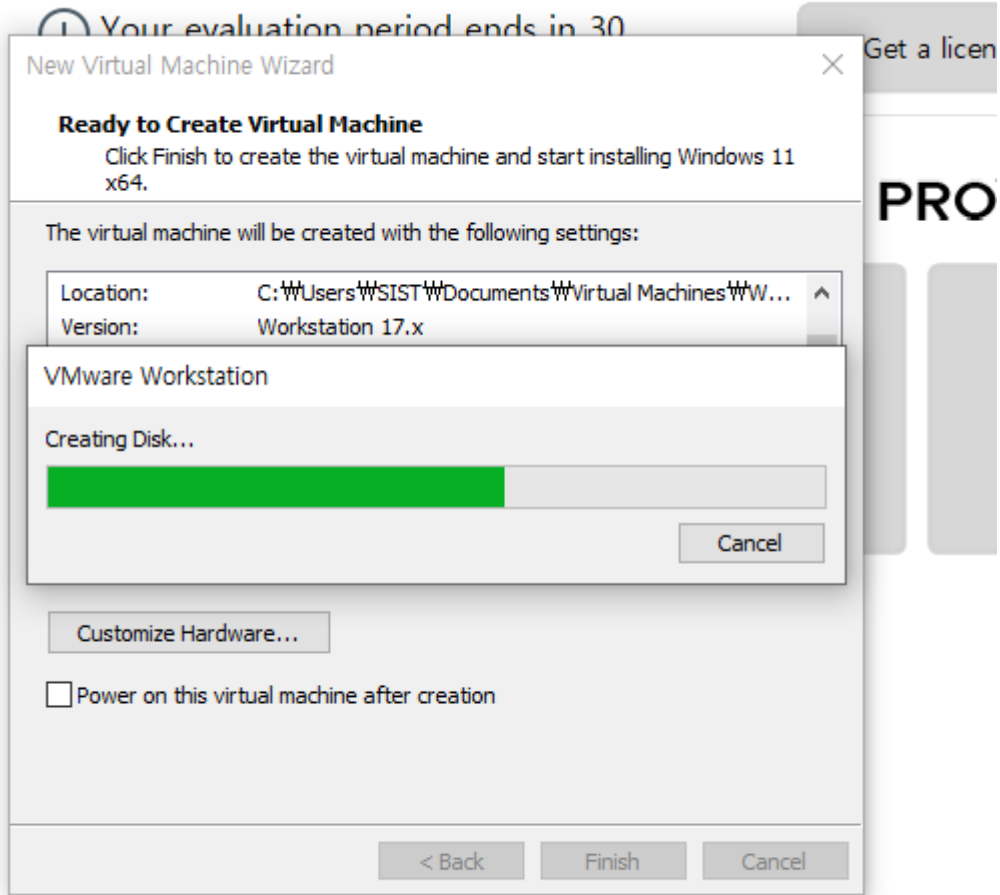
Number of processors:

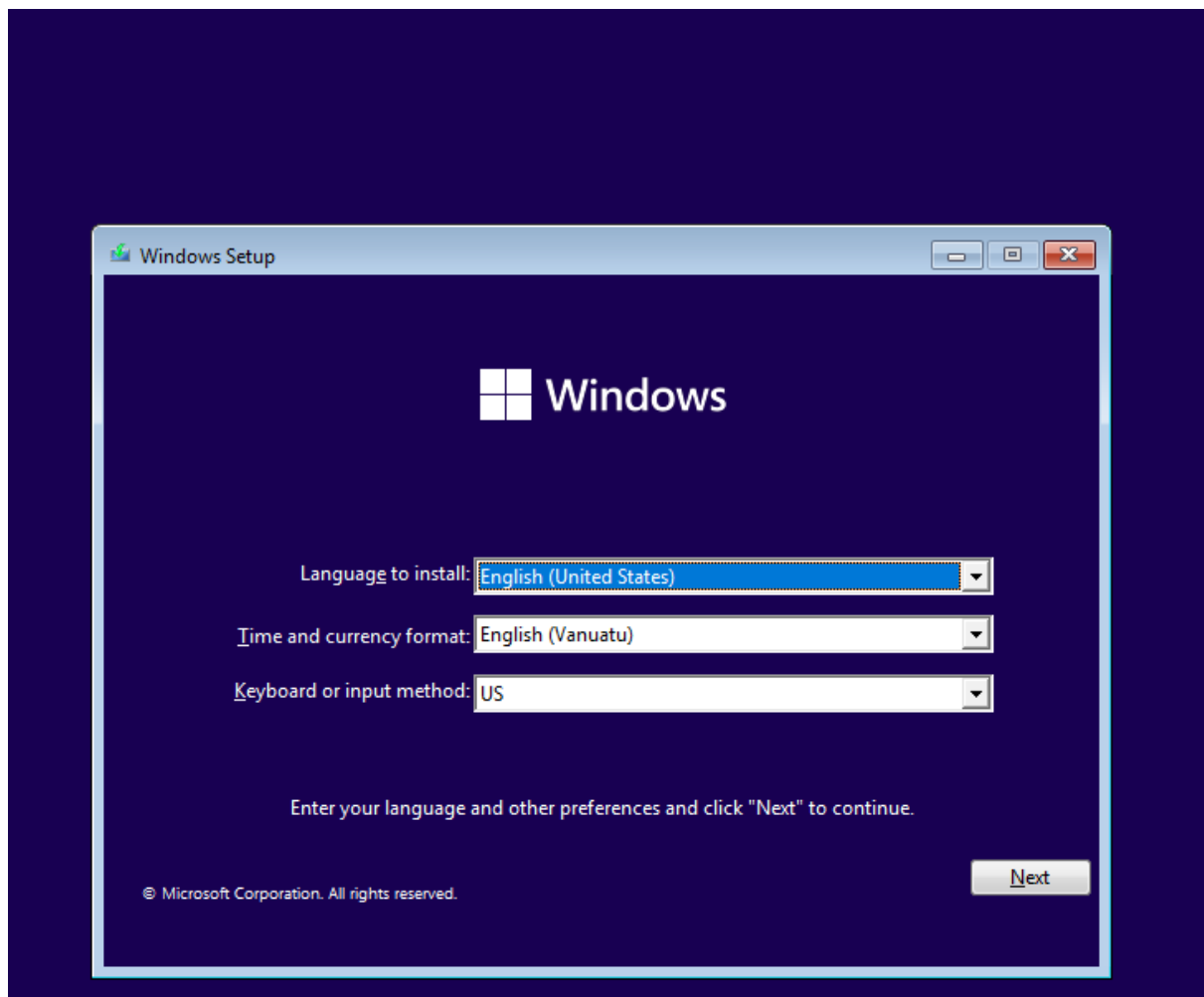
Number of cores per processor:

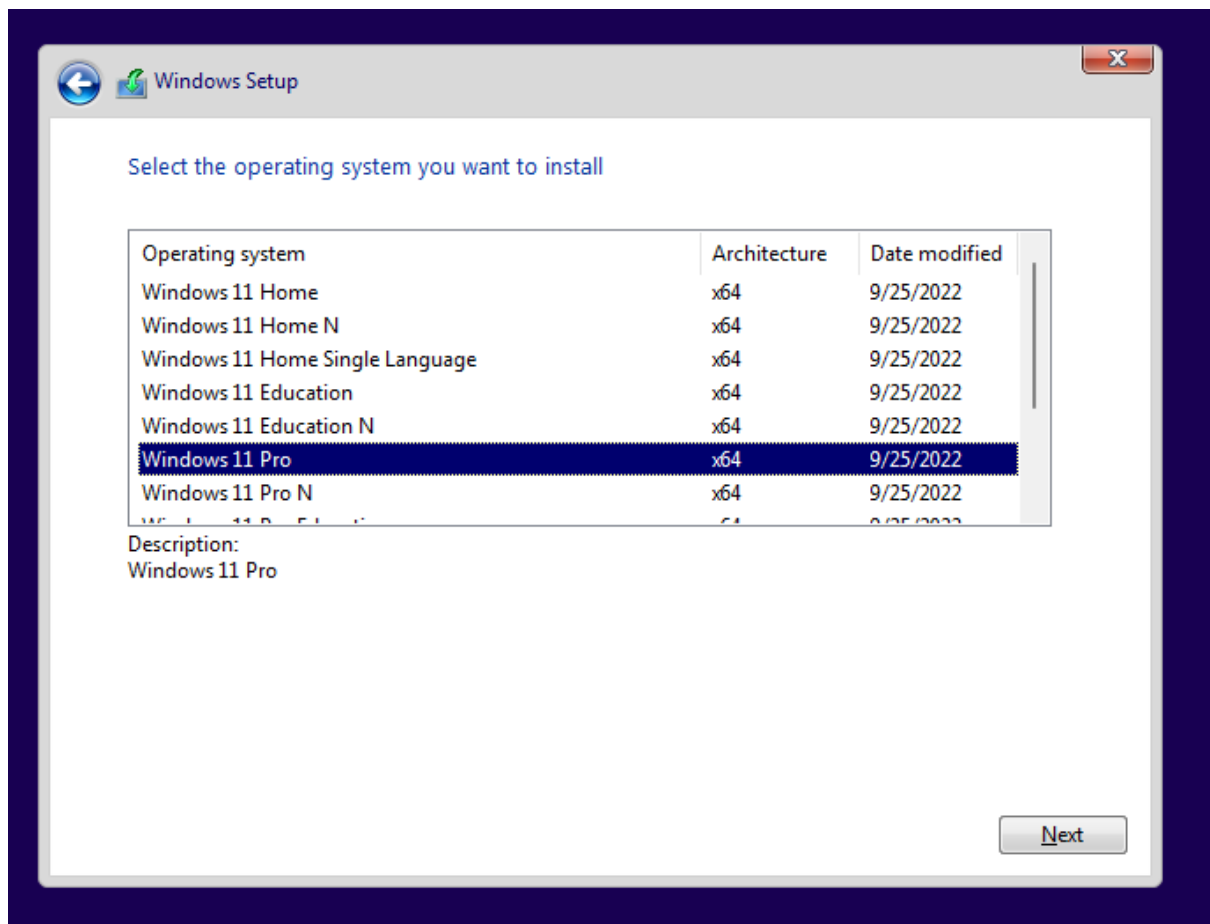
Total processor cores: 4

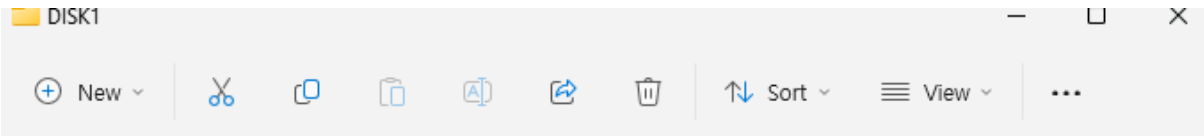
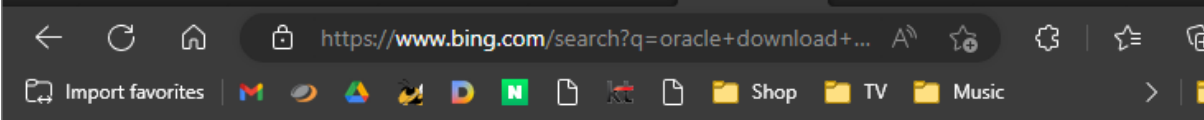
Virtualization engine

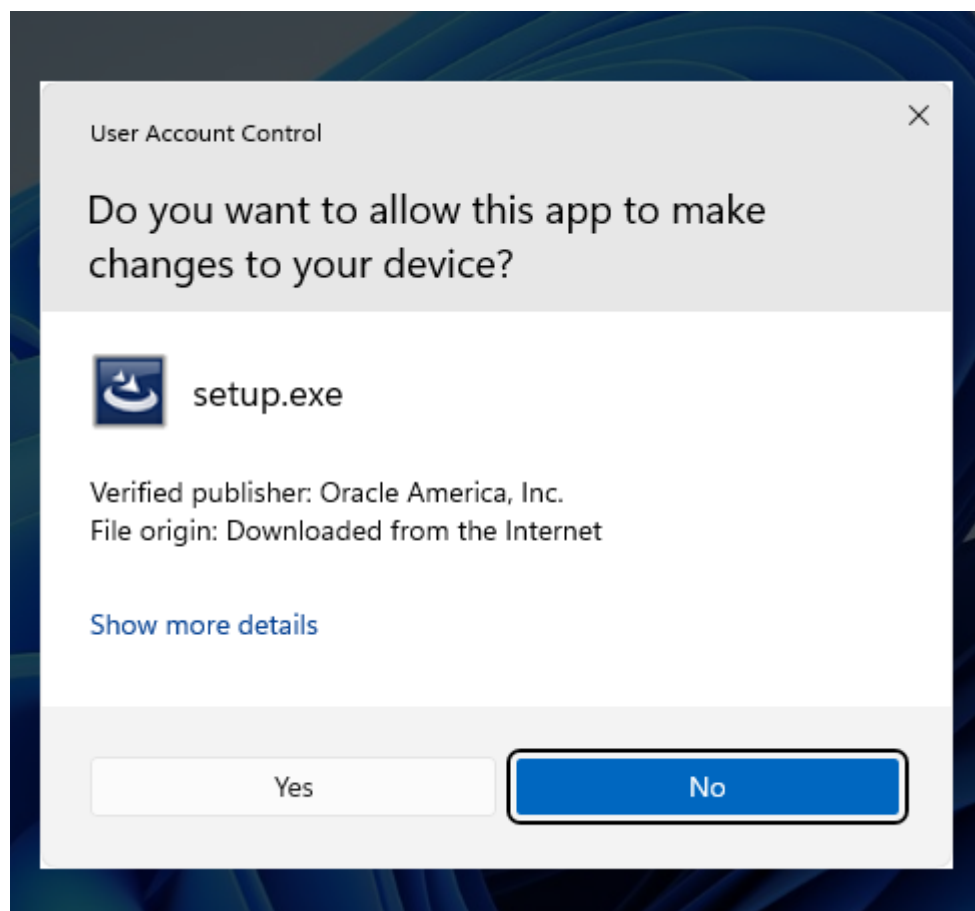
- ☐ Virtualize Intel VT-x/EPT or AMD-V/RVI
- ☐ Virtualize CPU performance counters
- ☐ Virtualize IOMMU (IO memory management unit)

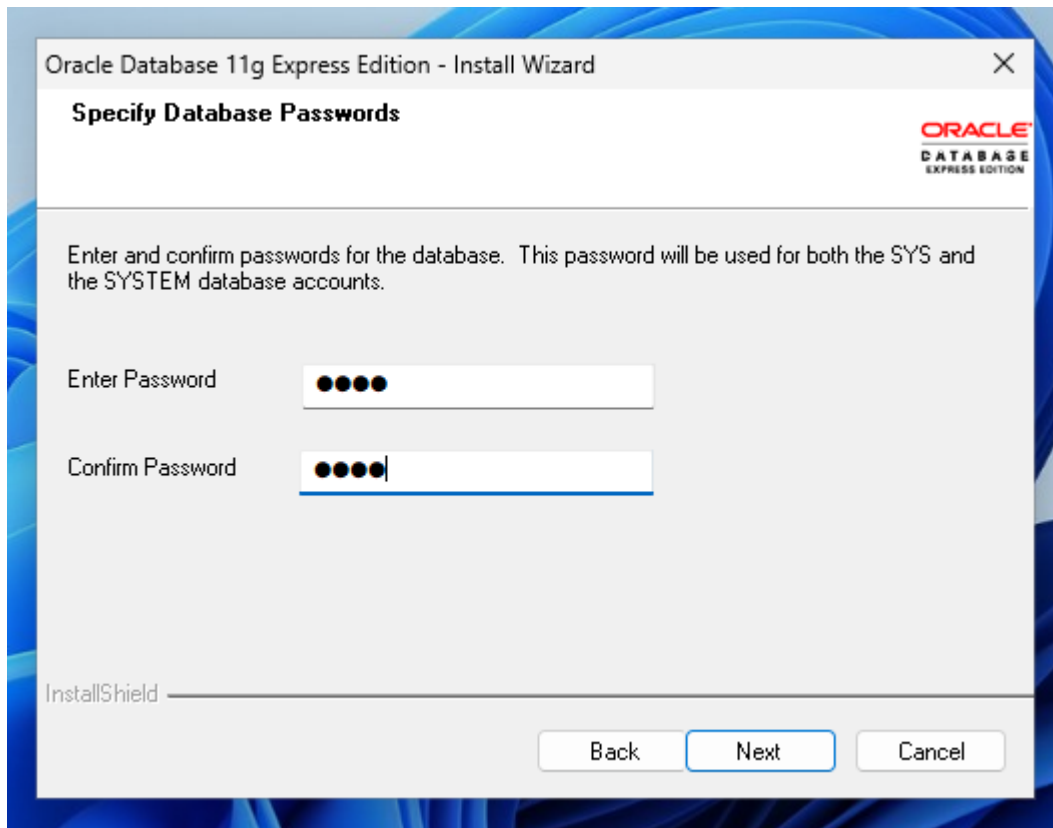




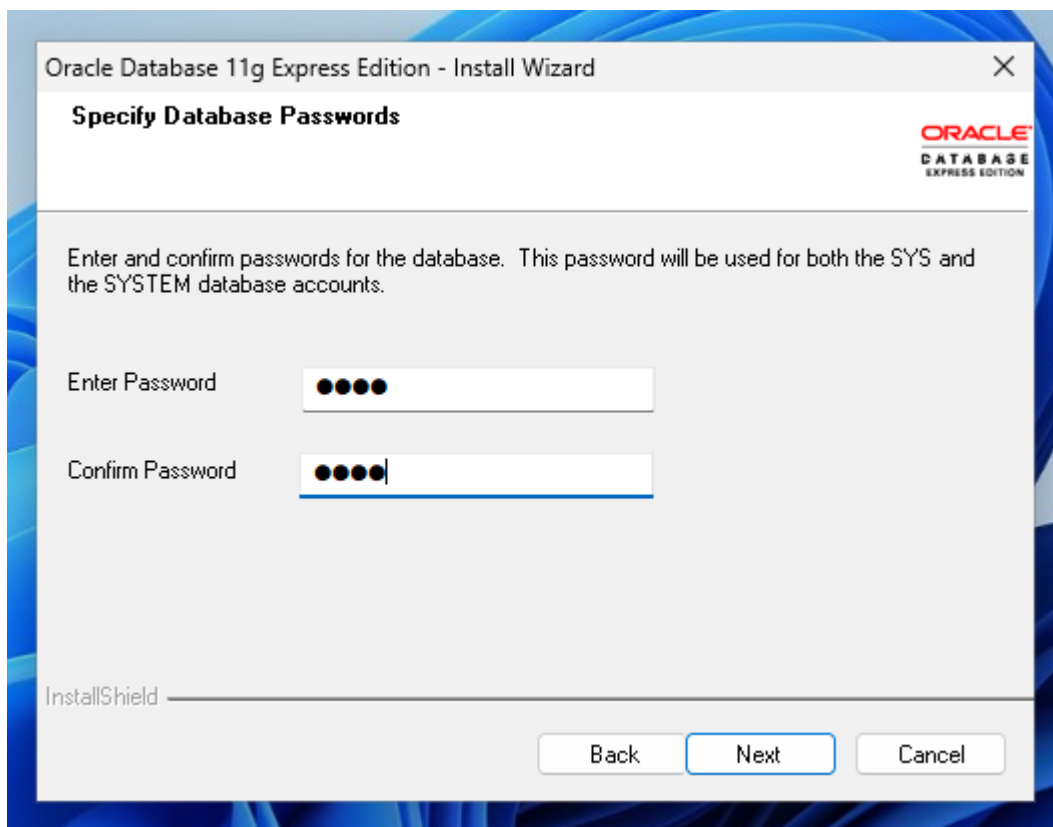




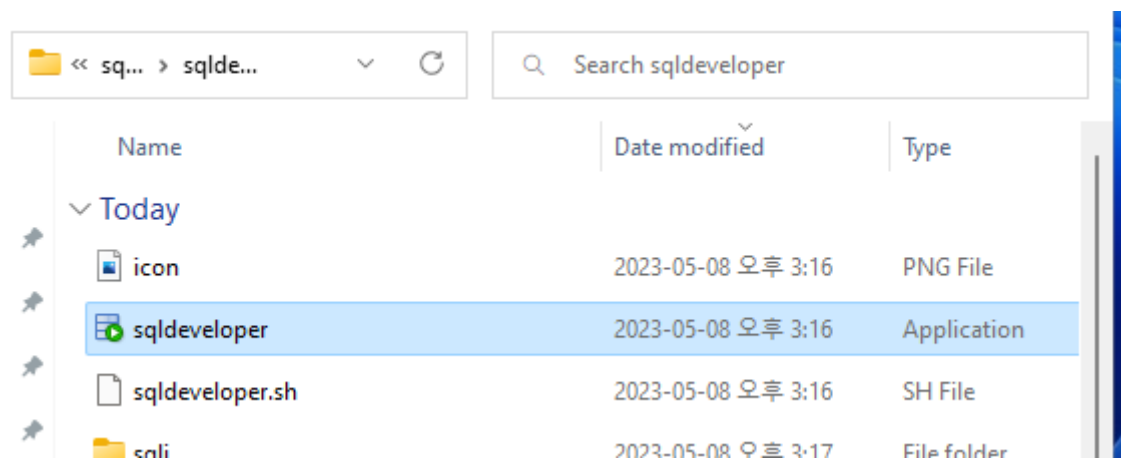
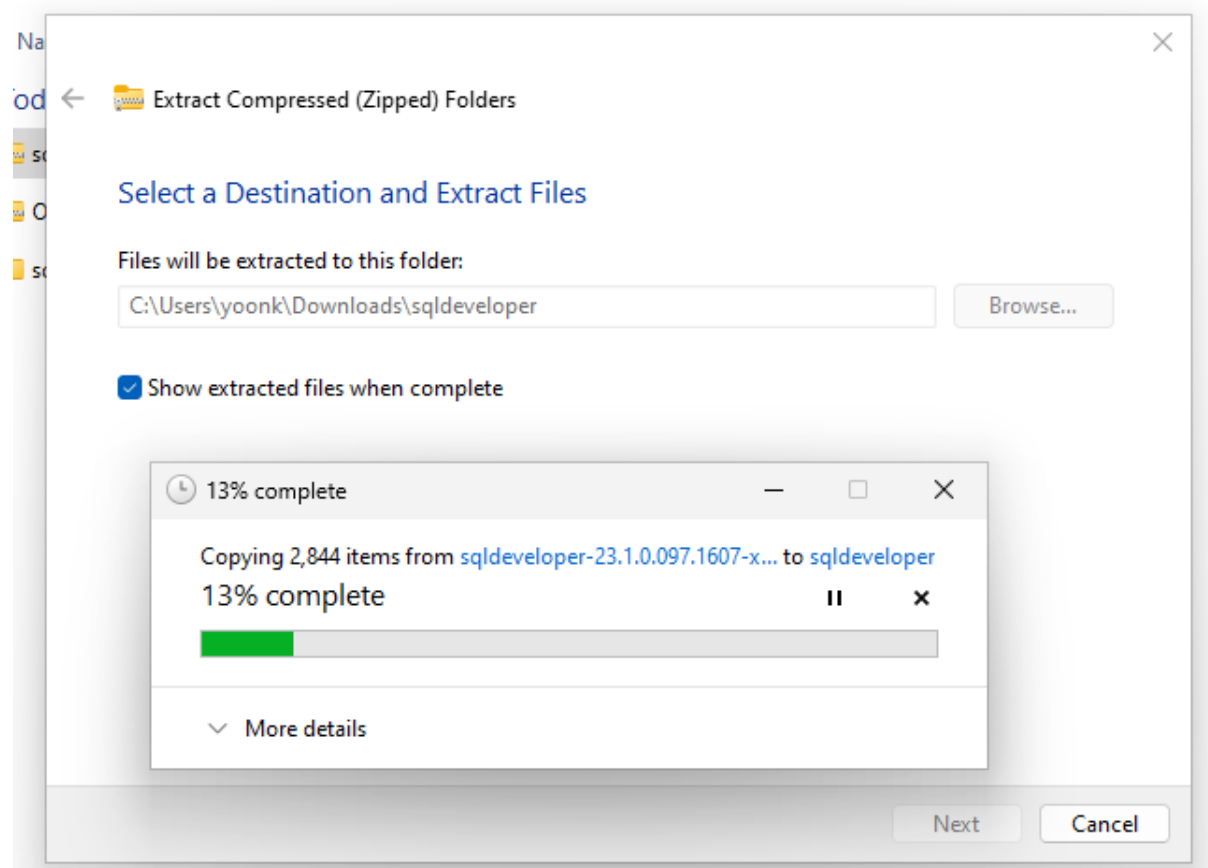


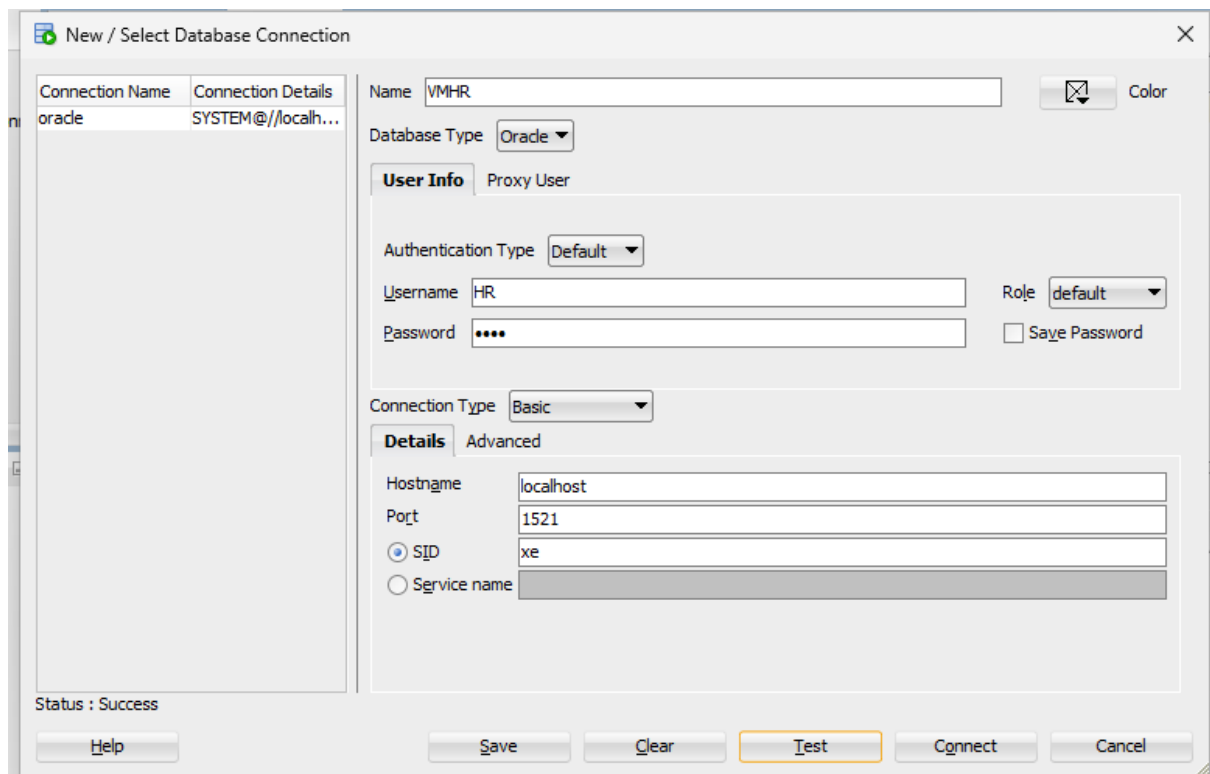


0000 비번설정

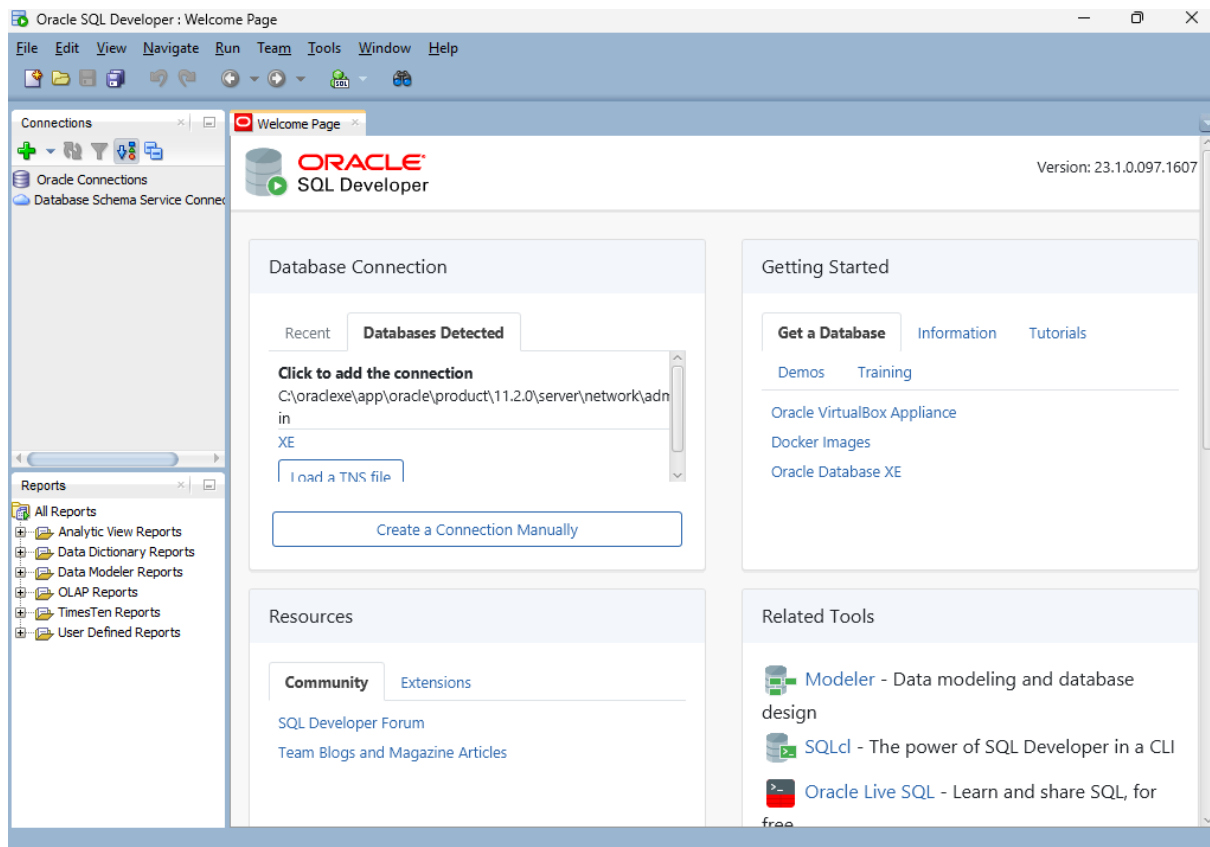


sql developer 는 설치 과정 없이 압축 만 풀어서 사용한다.





준비끝





New / Select Database Connection

Connection Name | Connection Details

Name ☐ Color

Database Type **Oracle**

User Info Proxy User

Authentication Type **Default**

Username Role **default**

Password ☐ Save Password

Connection Type **Basic**

Details Advanced

Hostname

Port


☒ SID

☐ Service name

Status :

New / Select Database Connection

Connection Name Connection Details

Name  Color

Database Type

User Info Proxy User

Authentication Type

Username Role

Password ☒ Save Password

Connection Type

Details Advanced

Hostname

Port


☒ SID

☐ Service name

Status : Success

New / Select Database Connection

Connection Name Connection Details

Name  Color

Database Type

User Info Proxy User

Authentication Type

Username Role

Password ☒ Save Password

Connection Type

Details Advanced

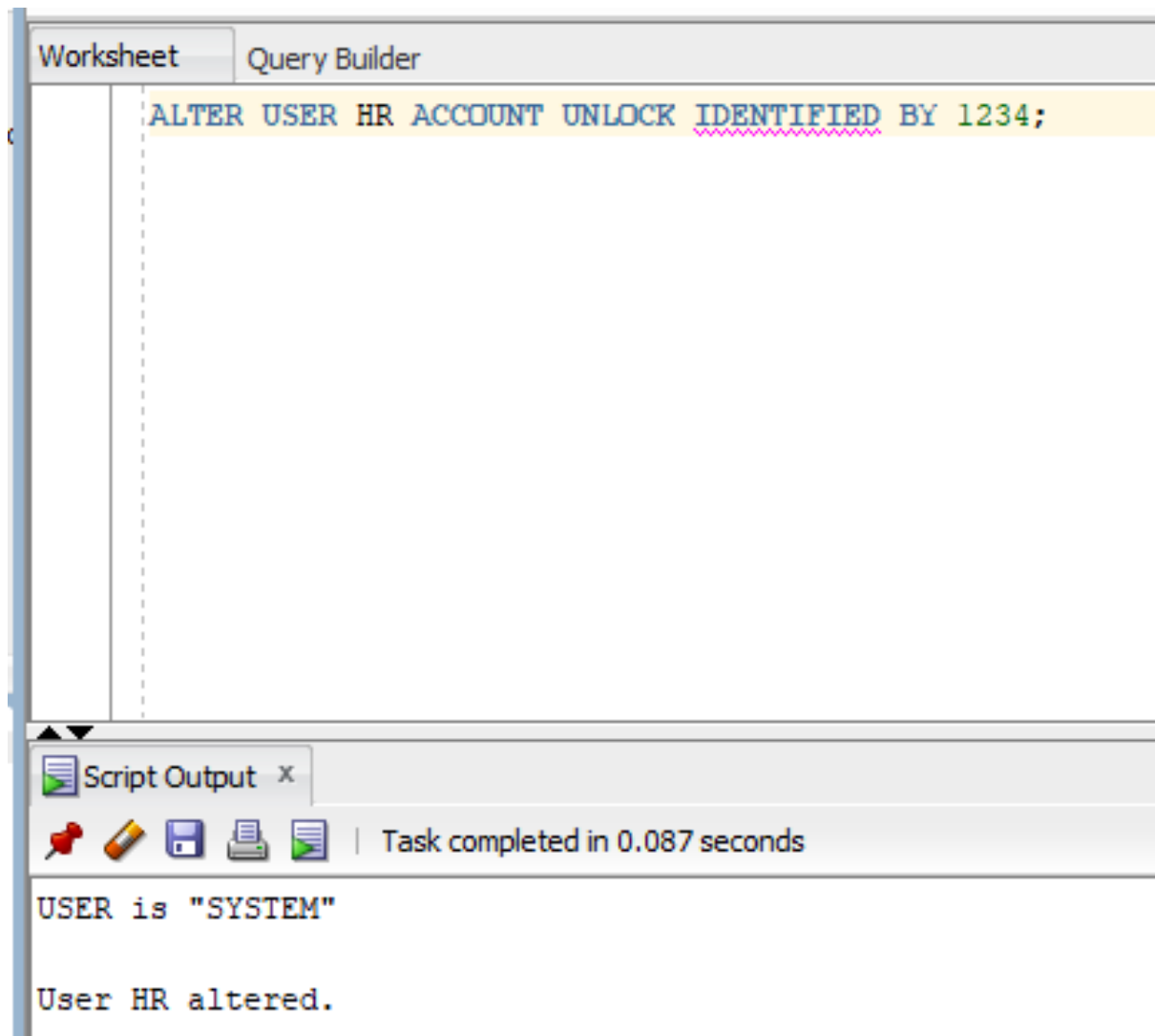
Hostname

Port

☒ SID

☐ Service name

Status : Success



New / Select Database Connection

Connection Name	Connection Details
orade	SYSTEM@//localh...

Name: VMHR Color

Database Type: Oracle

User Info Proxy User

Authentication Type: Default

Username: HR Role: default

Password: ☐ Save Password

Connection Type: Basic

Details Advanced

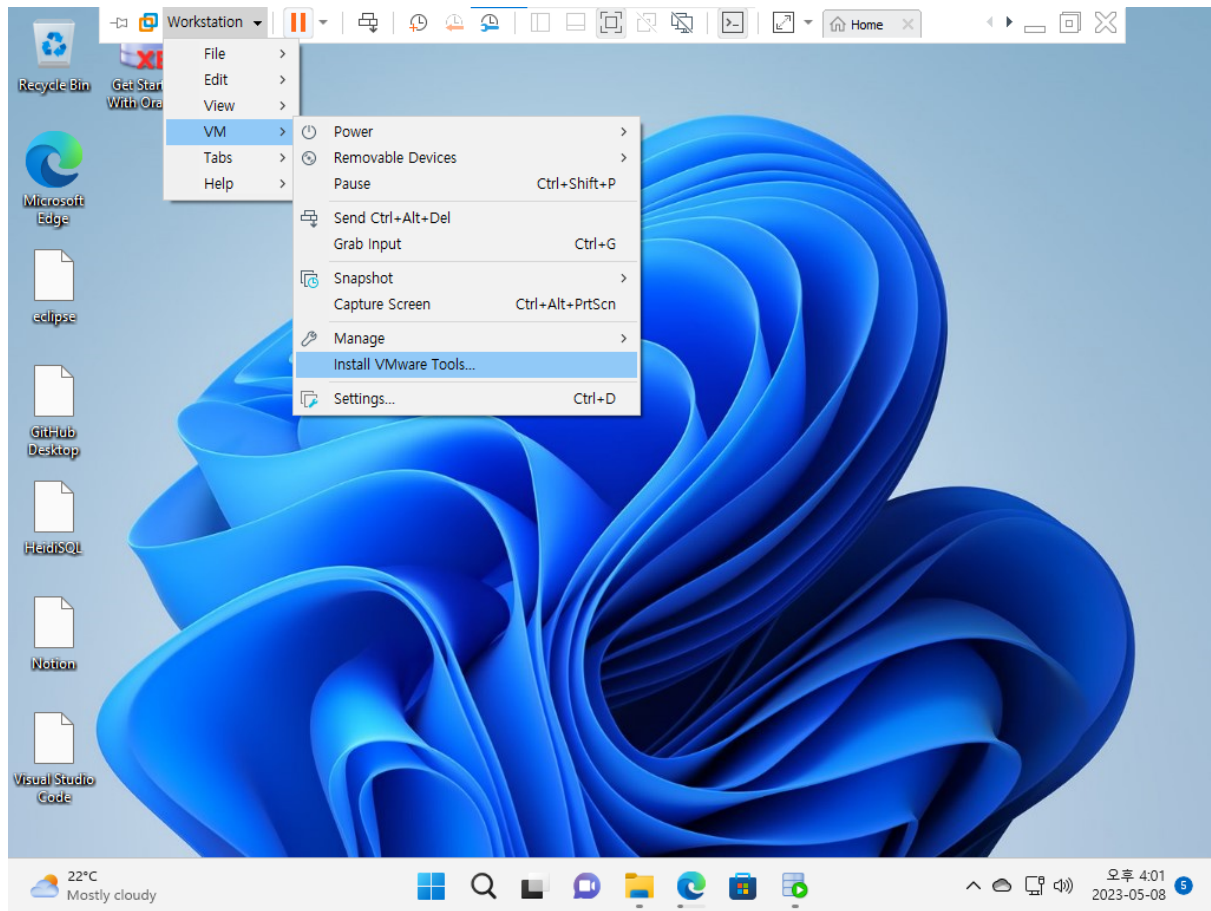
Hostname: localhost

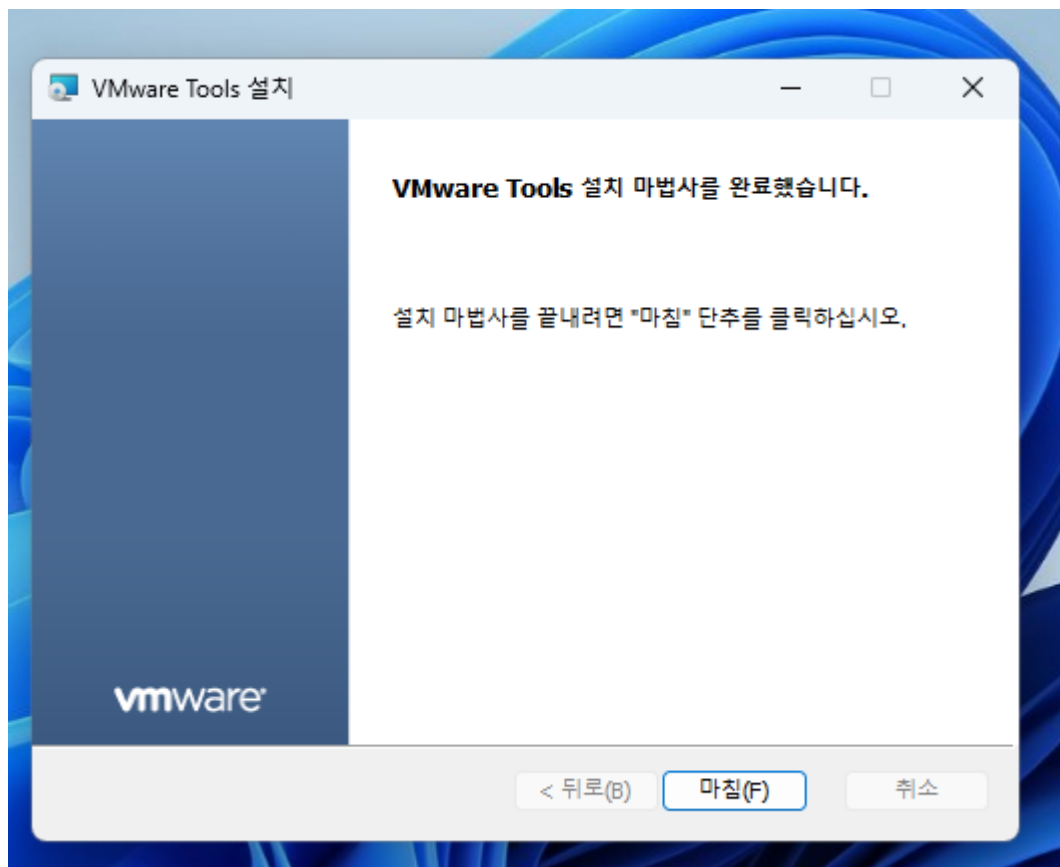
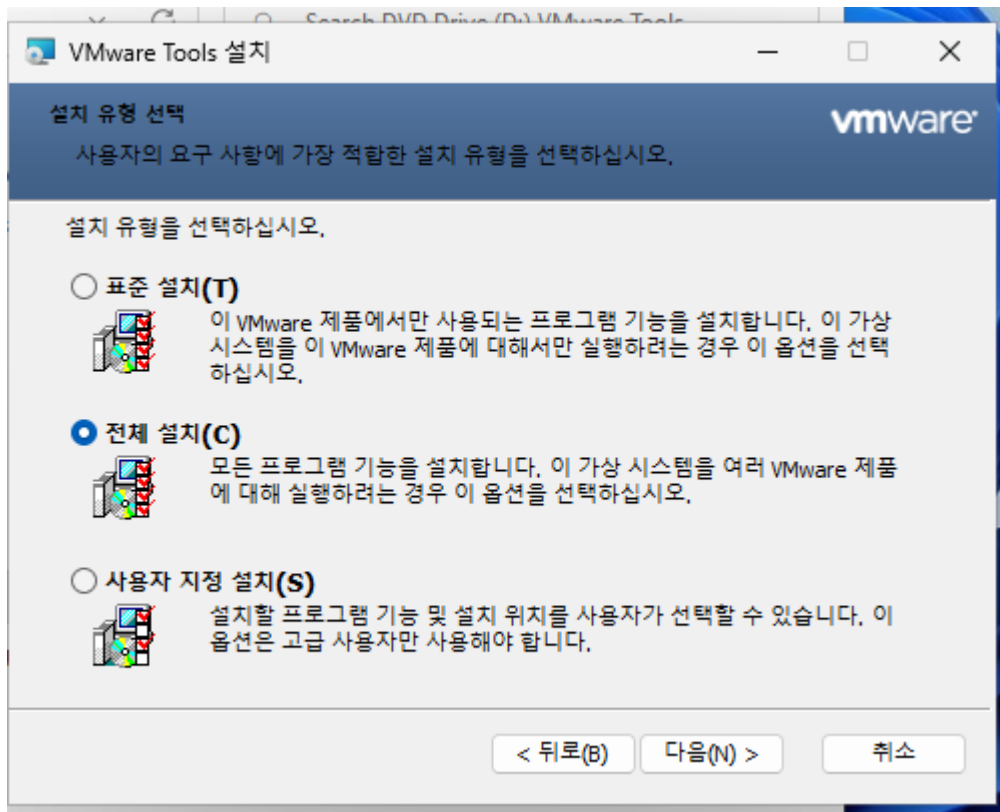
Port: 1521

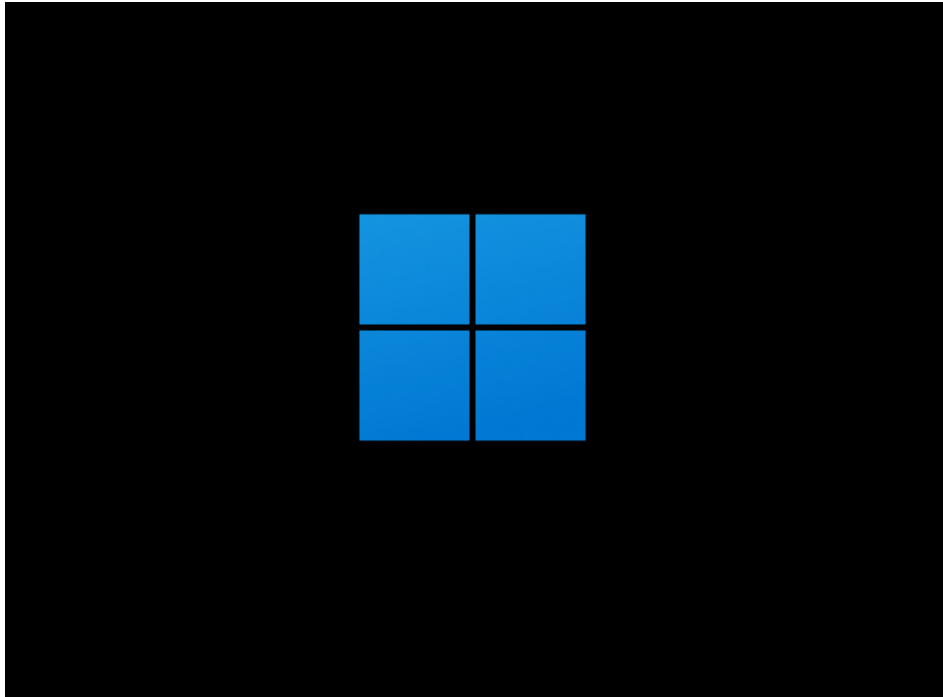
☒ SID: xe

☐ Service name:

Status : Success







WHERE 조건 절

- where 절은 반드시 from 다음에 사용한다.

SELECT 열이름1, 열이름2

FROM 테이블이름

WHERE 조건;

Worksheet Query Builder

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

Query Result x

SQL | All Rows Fetched: 1 in 0.008 seconds

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

Worksheet Query Builder

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

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

Worksheet Query Builder

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

Query Result x

SQL | All Rows Fetched: 2 in 0.009 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	CC
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	

Worksheet Query Builder

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

Query Result x

SQL | All Rows Fetched: 0 in 0.003 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	CC
--	-------------	------------	-----------	-------	--------------	-----------	--------	--------	----

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

Welcome Page x VMHR x

Worksheet Query Builder

```
SELECT *
FROM employees
WHERE first_name = 'David';
```

Query Result x

SQL | All Rows Fetched: 3 in 0.006 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-05	IT_PROG	4800
2	151	David	Bernstein	DBERNSTE	011.44.1344.345268	24-MAR-05	SA_REP	9500
3	165	David	Lee	DLEE	011.44.1346.529268	23-FEB-08	SA_REP	6800

Welcome Page x VMHR x

Worksheet Query Builder

```
SELECT *
FROM employees
WHERE first_name = 'david';
```

Query Result x

SQL | All Rows Fetched: 0 in 0.001 seconds

	EMPLOYEE...	FIRST_NA...	LAST_NAME	EMAIL	PHONE_N...	HIRE_DATE	JOB_ID	SAL
--	-------------	-------------	-----------	-------	------------	-----------	--------	-----

Worksheet Query Builder

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

Query Result x

SQL | All Rows Fetched: 19 in 0.003 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
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

Worksheet Query Builder

```
SELECT *
FROM employees
WHERE employee_id >= 105;
```

Query Result x

SQL | Fetched 50 rows in 0.016 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-05	IT_PROG	4800
2	106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-06	IT_PROG	4800
3	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-07	IT_PROG	4200
4	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-02	FI_MGR	12008
5	109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-02	FI_ACCOUNT	9000
6	110	John	Chen	JCHEN	515.124.4269	28-SEP-05	FI_ACCOUNT	8200
7	111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-SEP-05	FI_ACCOUNT	7700
8	112	Tina	Martins	TDMARTIN	515.124.4469	02-MAR-06	FI_ACCOUNT	6200

복잡한 조건 적용하기

BETWEEN 1 AND 4

Welcome Page x VMHR x

Worksheet Query Builder

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

Query Result x

SQL | All Rows Fetched: 18 in 0.006 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
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

IN(1,2,3,4)

Welcome Page x VMHR x

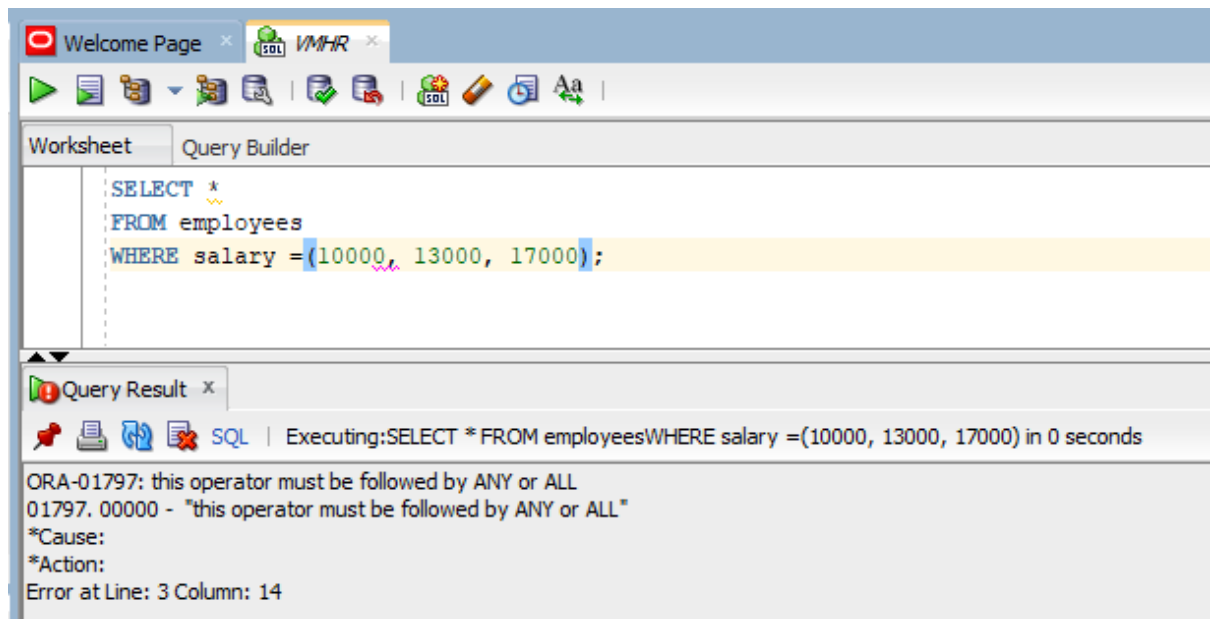
Worksheet Query Builder

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

Query Result x

SQL | All Rows Fetched: 7 in 0.005 seconds

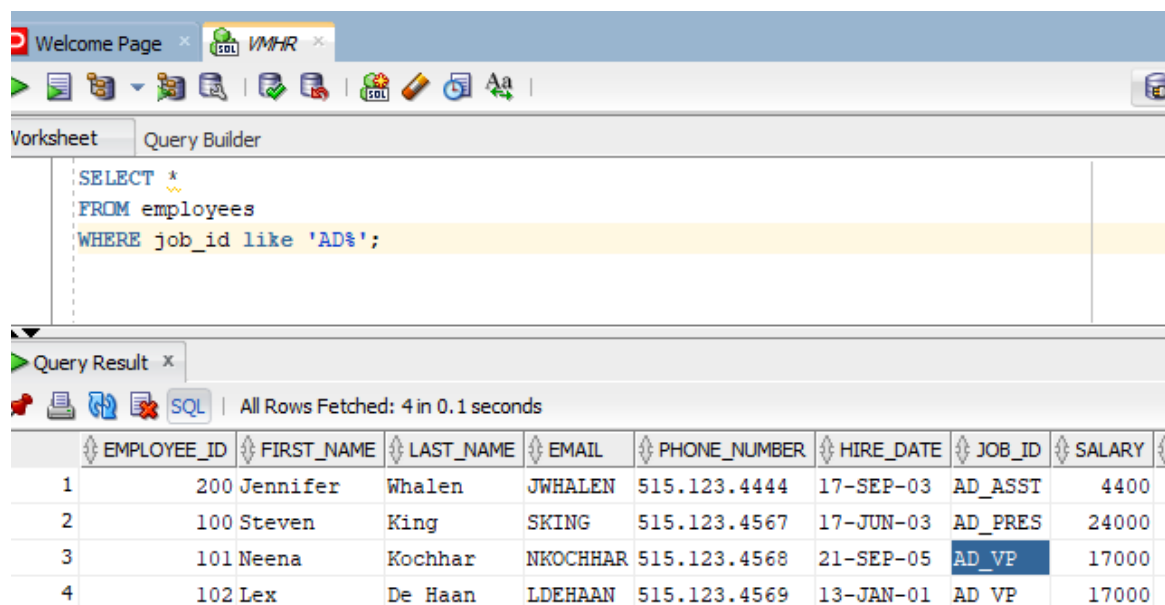
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000
2	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000
3	150	Peter	Tucker	PTUCKER	011.44.1344.129268	30-JAN-05	SA_REP	10000
4	156	Janette	King	JKING	011.44.1345.429268	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



LIKE '%도' 도로 끝나는 값다 찾기 경기도/제주도

'%@%' 포함하기만 하면

'서울%' 서울시 구로 서울시 강남



Worksheet Query Builder

```
SELECT *
FROM employees
WHERE first_name like '%a%';
```

Query Result x

SQL | Fetched 50 rows in 0.008 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	101	Neena	Kochhar	NKOCHHAR	515.123.4568
2	103	Alexander	Hunold	AHUNOLD	590.423.4567
3	105	David	Austin	DAUSTIN	590.423.4569
4	106	Valli	Pataballa	VPATABAL	590.423.4560
5	107	Diana	Lorentz	DLORENTZ	590.423.5567
6	108	Nancy	Greenberg	NGREENBE	515.124.4569
7	109	Daniel	Faviet	DFAVIET	515.124.4169
8	111	Ismael	Sciarra	ISCIARRA	515.124.4369
9	112	Jose Manuel	Urman	JMURMAN	515.124.4469

Worksheet Query Builder

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

Query Result x

SQL | All Rows Fetched: 2 in 0.009 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

IS NULL null 찾기

Welcome Page x
 VMHR x

Worksheet Query Builder

```

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

Query Result x

SQL | All Rows Fetched: 1 in 0.017 seconds

ME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DE
1	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(null)	(null)	