**프로그래밍 기초 - 2001020216\_15v3 응용 SW 기초 기술 활용 과제**

|  |  |  |  |
| --- | --- | --- | --- |
| **과정명** | 클라우드 기반 JAVA 웹 응용 SW 엔지니어 양성과정 | | |
| **교과목** | 프로그래밍 기초 | **훈련생** | 박훈 |
| **능력단위** | 2001020216\_15v3 응용 SW 기초 기술 활용 | **훈련교사** | 이원일 |

[과제1] 리눅스 기본 명령어를 조사하여 제출한다.

# cd

체인지 디렉토리

# pwd

워킹 디렉토리

# ls

리스트

옵션:

-F 폴더 파일 구분

-G 컬러

-a 숨김파일

-h 사이즈 표시. 보기편하게

-l 리스트

-t 시간순서

# touch

파일 만들기

# mkdir

디렉토리 만들기

# cp

복사하기

옵션:

-f 강제

-i 물어보기

-v 과정 보여주기

# mv

파일 옮기기

옵션:

-f 강제

-i 물어보기

-v 과정 보여주기

# rm

리무브. 없애기

옵션:

-f 강제

-i 물어보기

-v 과정 보여주기

# cat

파일 내용보기

# head

파일 내용보기 윗부분만

# tail

파일 내용보기 아랫부부만

# find

파일 찾기

옵션:

-name 이름으로 찾기

# man

커맨드 매뉴얼보기

[과제2] 데이터베이스 종류별(객체형, 관계형 등)로 그 특징을 조사하여 제출한다.

[[[ RDB ]]]

01) RDBMS (Relational Database Management System)

| 1970년 에드거 커드가 제안한 데이터 관계형 모델에 기초하는 데이터베이스.

| 키(key)와 값(value)들의 간단한 관계를 테이블화 시킨 매우 간단한 원칙의 전산정보 데이터베이스.

| 데이터를 컬럼과 로우를 이루는 하나 이상의 테이블(또는 관계)로 정리함.

| 로우는 레코드 또는 튜플로 부른다

| 일반적으로 각 테이블(또는 관계)는 하나의 엔티티 타입을 대표.

| 로우는 그 엔티티 종류의 인스턴스를 대표.

| 컬럼은 그 인스턴스의 속성이 되는 값들을 대표.

| 오라클, MS-SQL, MySQL, PostgreSQL, CUBRID

02) ORDBMS (Oriented-Relational Database Management System)

| 객체지향 데이터베이스 모델을 가진 관계형 데이터베이스 관리 시스템.

| 개발자가 스스로 데이터 형과 메서드를 자유롭게 정의하여 DB를 개발.

| 기존의 RDBMS에 외부 소프트웨어 도구를 추가하여 ORDBMS와 비슷한 기능을 제공할 수도 있는데 이러한 외부 소프트웨어 도구를 객체 관계 매핑 시스템이라고 부름.

| ORDBMS 기술의 목표는 소프트웨어 개발자에게 문제 영역을 생각하는 수준까지 DB 설계의 추상화 수준을 높이는 것.

| PostgreSQL, 오라클

[[[ NoSQL (Not Only SQL) ]]]

01)

| 전통적인 RDB보다 덜 제한적인 일관성 모델을 이용하는 데이터의 저장 및 검색을 위한 매커니즘을 제공.

| 이런 접근에 대한 동기는 디자인의 단순화, 수평적 확장성, 세세한 통제를 포함.

| 단순 검색 및 추가 작업을 위한 매우 최적화된 키 값 저장 공간으로, 레이턴시(latency)와 스루풋(throughput)과 관련하여 상당한 성능 이익을 내는 것이 목적.

| 빅데이터와 실시간 웹 애플리케이션의 상업적 이용에 널리 사용.

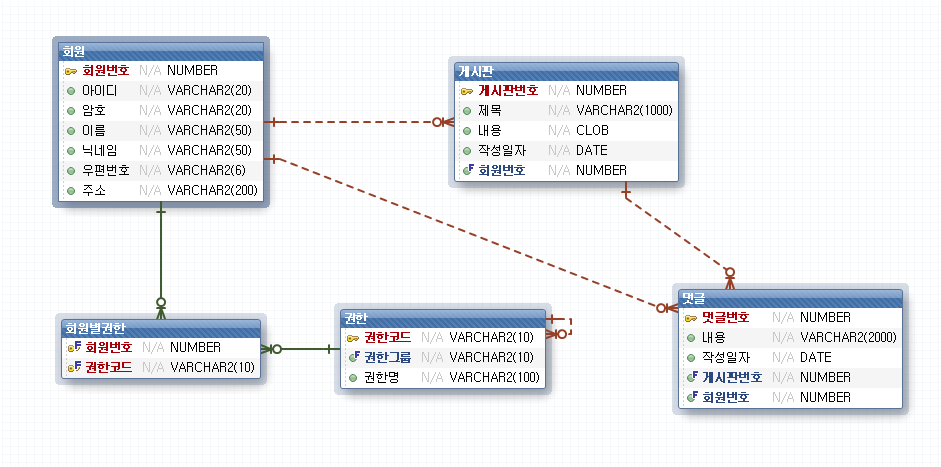
| "Not Only SQL"이라는 말은 SQL만 사용하는 것이 아니라는 말과 동시에, 쿼리 언어를 사용할 수 있다는 말이기도 함.

| MongoDB

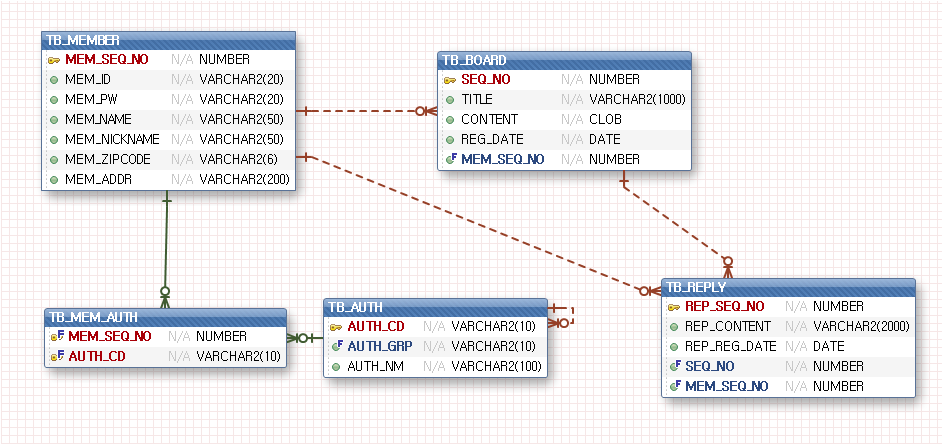
[과제3] 주어진 E-R 다이어그램을 보고 테이블을 생성하는 스크립트를 작성하여 제출한다.

아래의 논리/물리 E-R 다이어그램을 보고 테이블 생성 스크립트를 작성하시오

[논리모델]



[물리모델]



-- ERD script

-- ==========

-- ## 회원 테이블

-- \* PK : mem\_seq\_no

CREATE TABLE tb\_member (

mem\_seq\_no NUMBER,

mem\_id VARCHAR2(20),

mem\_pw VARCHAR2(20),

mem\_name VARCHAR2(50),

mem\_nickname VARCHAR2(50),

mem\_zipcode VARCHAR2(6),

mem\_addr VARCHAR2(200),

CONSTRAINT pk\_tb\_member PRIMARY KEY (mem\_seq\_no)

);

-- ## 권한 테이블

-- \* PK : auth\_cd

-- \* FK : auth\_grp

CREATE TABLE tb\_auth (

auth\_cd VARCHAR2(10),

auth\_grp VARCHAR2(10),

auth\_nm VARCHAR2(100),

CONSTRAINT pk\_tb\_auth PRIMARY KEY (auth\_cd),

CONSTRAINT fk\_tb\_auth

FOREIGN KEY (auth\_grp)

REFERENCES tb\_auth(auth\_cd)

);

-- ## 회원별 권한 테이블

-- \* PK : mem\_seq\_no, auth\_cd

-- \* FK : mem\_seq\_no, auth\_cd

CREATE TABLE tb\_mem\_auth (

mem\_seq\_no NUMBER,

auth\_cd VARCHAR2(10),

CONSTRAINT pk\_tb\_mem\_auth PRIMARY KEY (mem\_seq\_no, auth\_cd),

CONSTRAINT fk\_tb\_mem\_auth\_tb\_member

FOREIGN KEY (mem\_seq\_no)

REFERENCES tb\_member(mem\_seq\_no),

CONSTRAINT fk\_tb\_mem\_auth\_tb\_auth

FOREIGN KEY (auth\_cd)

REFERENCES tb\_auth(auth\_cd)

);

-- ## 게시판 테이블

-- \* PK : seq\_no

-- \* FK : mem\_seq\_no

CREATE TABLE tb\_board (

seq\_no NUMBER,

title VARCHAR2(100),

content CLOB,

reg\_date DATE,

mem\_seq\_no NUMBER,

CONSTRAINT pk\_tb\_board PRIMARY KEY (seq\_no),

CONSTRAINT fk\_tb\_board\_tb\_member

FOREIGN KEY (mem\_seq\_no)

REFERENCES tb\_member(mem\_seq\_no)

);

-- ## 댓글 테이블

-- \* PK : rep\_seq\_no

-- \* FK : seq\_no, mem\_seq\_no

CREATE TABLE tb\_reply (

rep\_seq\_no NUMBER,

rep\_content VARCHAR(200),

rep\_reg\_date DATE,

seq\_no NUMBER,

mem\_seq\_no NUMBER,

CONSTRAINT pk\_tb\_reply PRIMARY KEY (rep\_seq\_no),

CONSTRAINT fk\_tb\_reply\_tb\_board

FOREIGN KEY (seq\_no)

REFERENCES tb\_board(seq\_no),

CONSTRAINT fk\_tb\_reply\_tb\_member

FOREIGN KEY (mem\_seq\_no)

REFERENCES tb\_member(mem\_seq\_no)

);

[과제4] 네트워크 계층구조에 대해 조사하여 제출한다.

!!! OSI 7 계층

OSI: Open System InterConnection

1) Physical Layer (물리)

물리적 전송 매체와 전송 신호와 관계 됨.

전송매체와 전송 신호를 액세스하기 위한 기계적, 전기적, 기능적 절차적 특성을 규정짓는 최하위 계층.

| 전기적, 기계적, 기능적, 절차 기능을 정의.

| 통신 케이블, 전송 신호 방식, 물리적 장비를 정의.

2) Data Link Layer (데이터 링크)

인접한 장치 간에 원활한 데이터의 전송을 하도록 하며, 물리적 연결(데이터 링크) 간의 신뢰성 있는 정보를 전송하는 계층의 기본 데이터 단위는 프레임.

| 동기화, 오류, 흐름 제어로 프레임(전송 데이터 구조)을 효과적으로 전송.

| 오류, 흐름을 제어함.

| 전화를 이용한 인터넷 접속 프로토콜(PPP) 기능을 담당함.

| 프레임을 동기화(BASIC 동기, HDLC 동기, SDLC 동기)함.

| 신뢰성 있고, 효율적인 데이터 전송을 함.

| 매체 액세스 제어를 한다(CSMA/CD, Token-Bus, Token-Ring).

| 전송 제어 기능을 담당

| 논리적 연결인 링크를 확립하거나 해제를 함.

3) Network Layer (네트워크)

통신 시스템 간의 경로를 선택하는 기능, 통신 트래픽의 흐름을 제어하는 기능 및 통신 중에 패킷의 분실로 재전송을 요청할 수 있는 오류 제어 기능을 수행하는 것

| 패킷 정보를 전송.

| 논리적 어드레싱.

| 정보 교환 및 중계 기능을 담당.

| 패킷을 목적지까지 전달.

| 경로 선택, 트래픽을 제어.

| 네트워크 연결을 관리.

| 체증 제어를 함.

4) Transport Layer (전송)

네트워크 종단(End Point) 시스템 간의 데이터를 일관성 있고 투명한 데이터 전송을 제공할 수 있도록 종단 간(End-to-End)에 통신을 지원.

| 종단(End-to-End) 사용자 간에 에러 복구와 흐름 제어를 제공.

| 일관성 있고 투명한 데이터 전송을 제어.

| 공유 회선을 다중화, 집중화.

| 주소를 지정.

5) Session Layer (세션)

응용 프로그램 간의 대화를 구성하고, 동기를 취하며 데이터 교환을 관리하기 위한 수단을 지원.

| 데이터가 전달되었으나 오류가 있는 데이터를 회복하기 위해 사용.

| 전송 방향을 결정(전이중, 반이중, 단방향).

| 데이터 교환, 데이터 관리.

| 전자 사서함 기능을 제공.

| 통신 시스템 간의 회화 기능을 관리.

| 오류가 발생하면 흐름의 마지막으로 되돌아가는 대동기점은, 데이터 교환을 대화 단위로 분할하여 오류를 확인.

| 소동기점은 대화의 중간에 삽입되며, 응용 프로그램에 따라 오류 확인이 요구되지 않을 수도 있음.

6) Presentation Layer (표현)

| 데이터의 코드를 변환.

| 데이터를 압축하여 전송.

| 정보 형식을 지정.

| 문맥을 관리.

| 구문을 검색.

| 코드릴 번역.

| 데이터를 암호화하거나 해독.

| 표현 형식을 제어.

| 데이터 표현에 존재하는 차이점들을 극복하게 함.

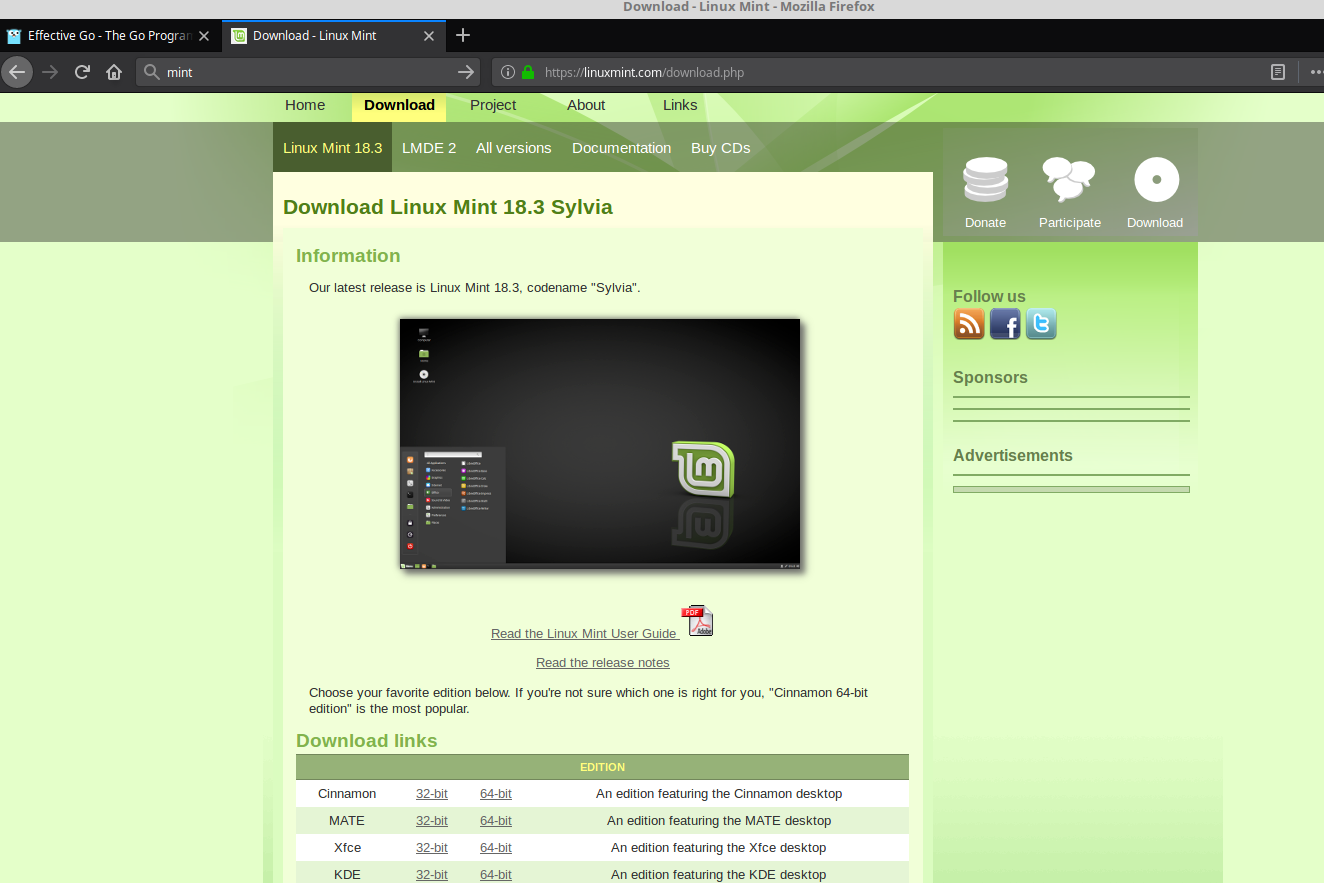
7) Application Layer (응용)

| 사용자가 OSI 환경에 접근이 가능하도록 함.

| OSI 네트워크 환경에서 사용자에게 서비스를 제공하는 계층.

[과제5] 리눅스 시스템 설치 가이드를 작성하여 제출한다.

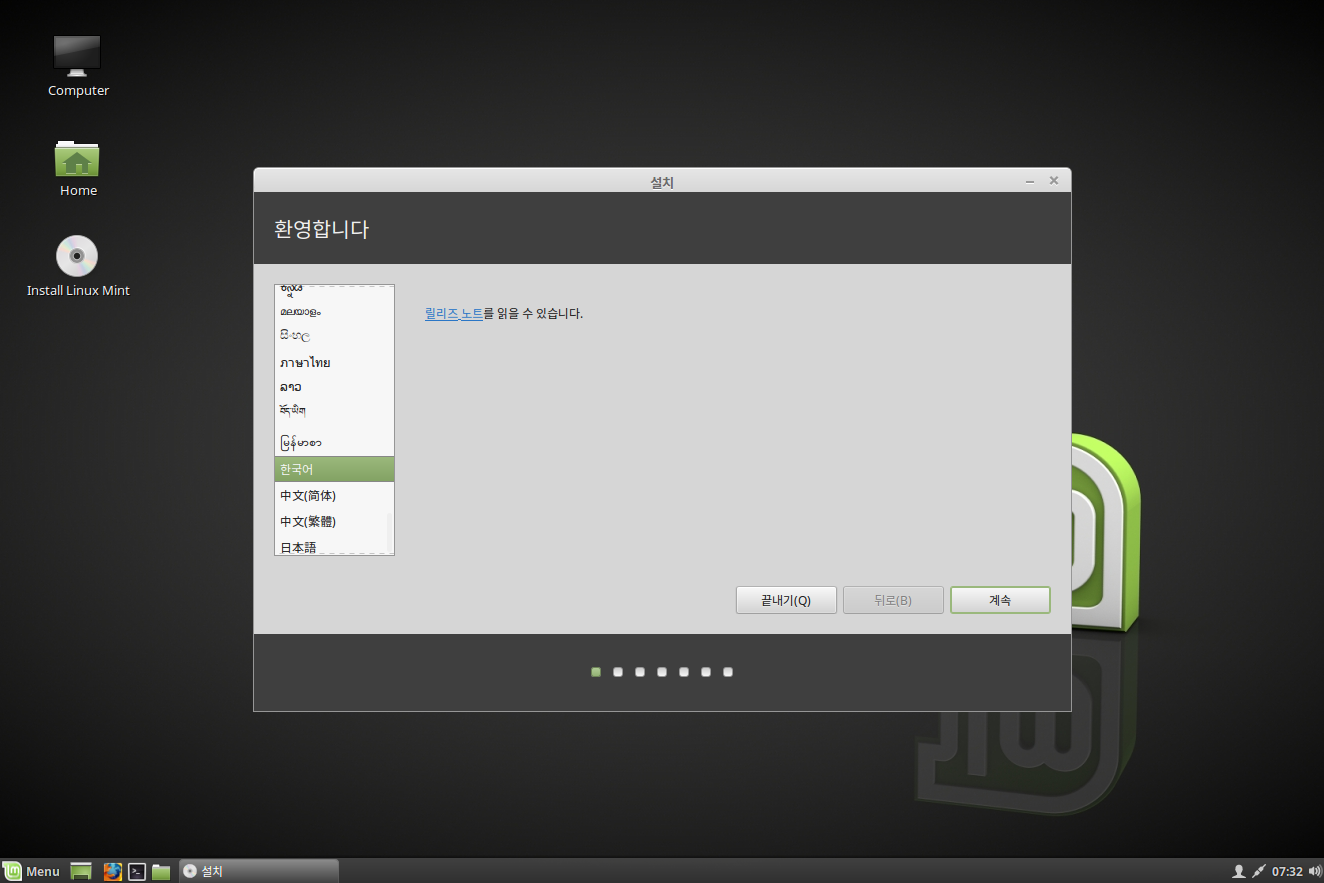
#01 리눅스 민트 다운로드



#02 다운받은 인스톨 파일 실행



#3 Install Linux Mint 클릭



#4 설치형식 선택



#5 사용자 이름과 컴퓨터 이름 입력

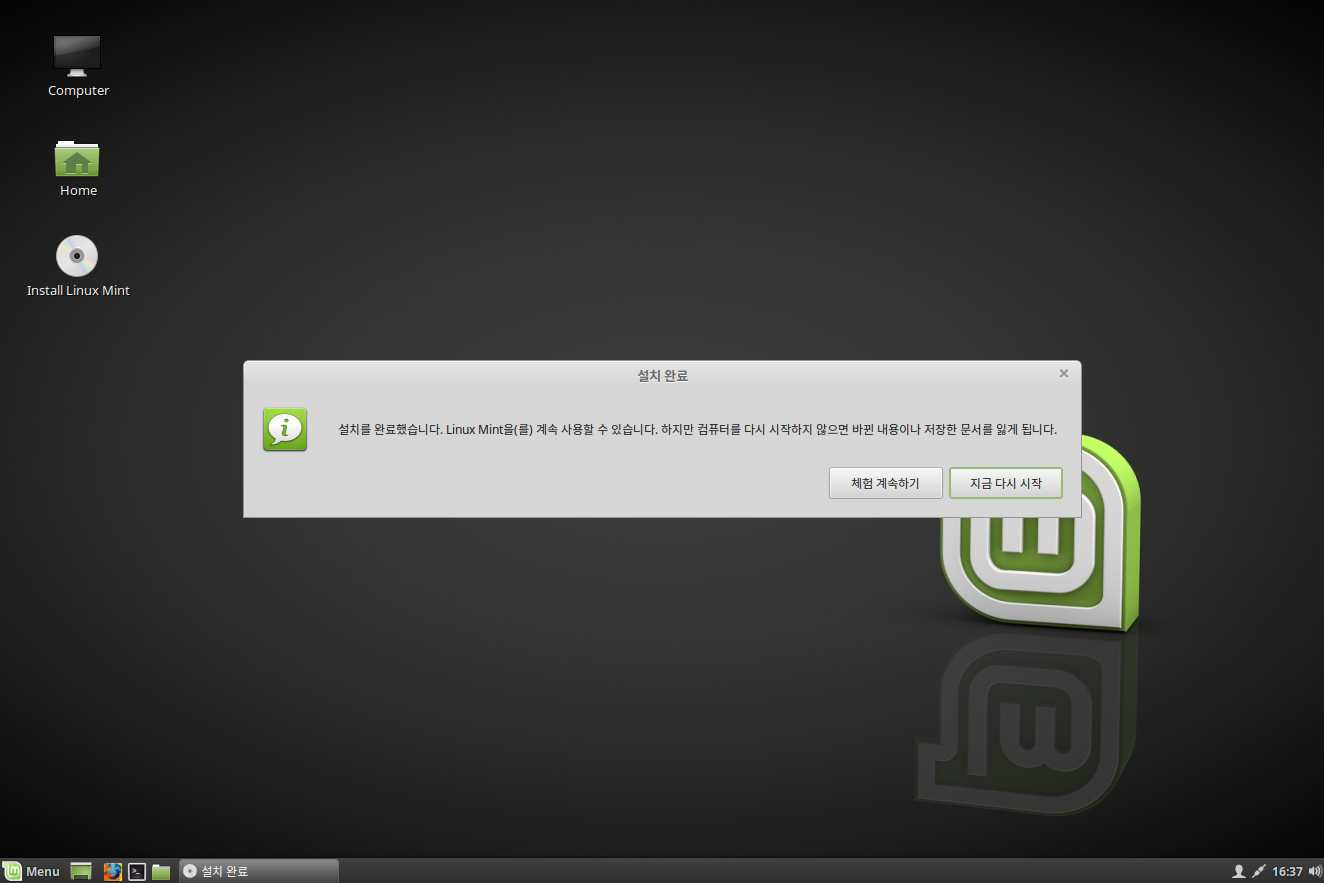


택

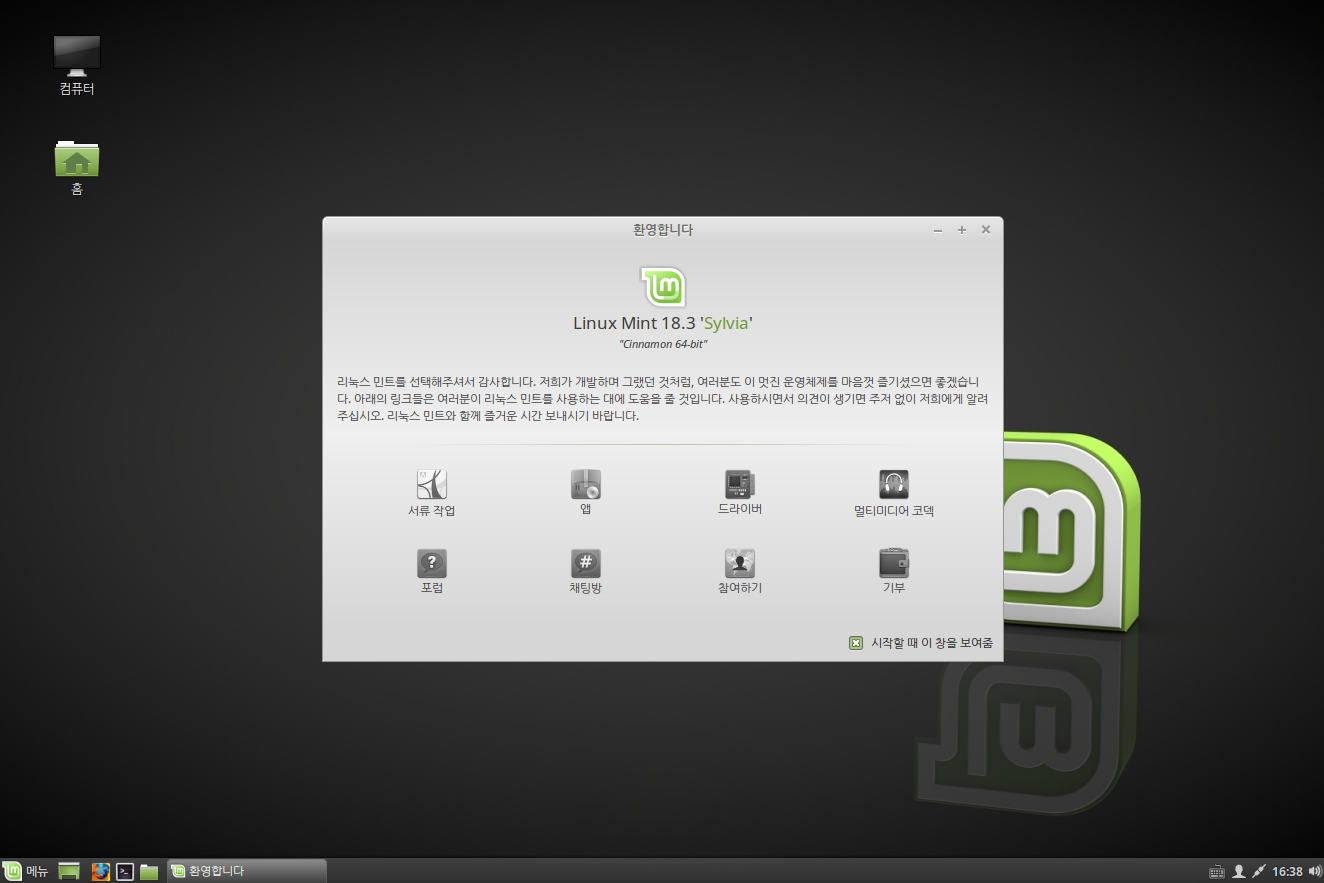
#6 설치 시작



#7 설치 완료

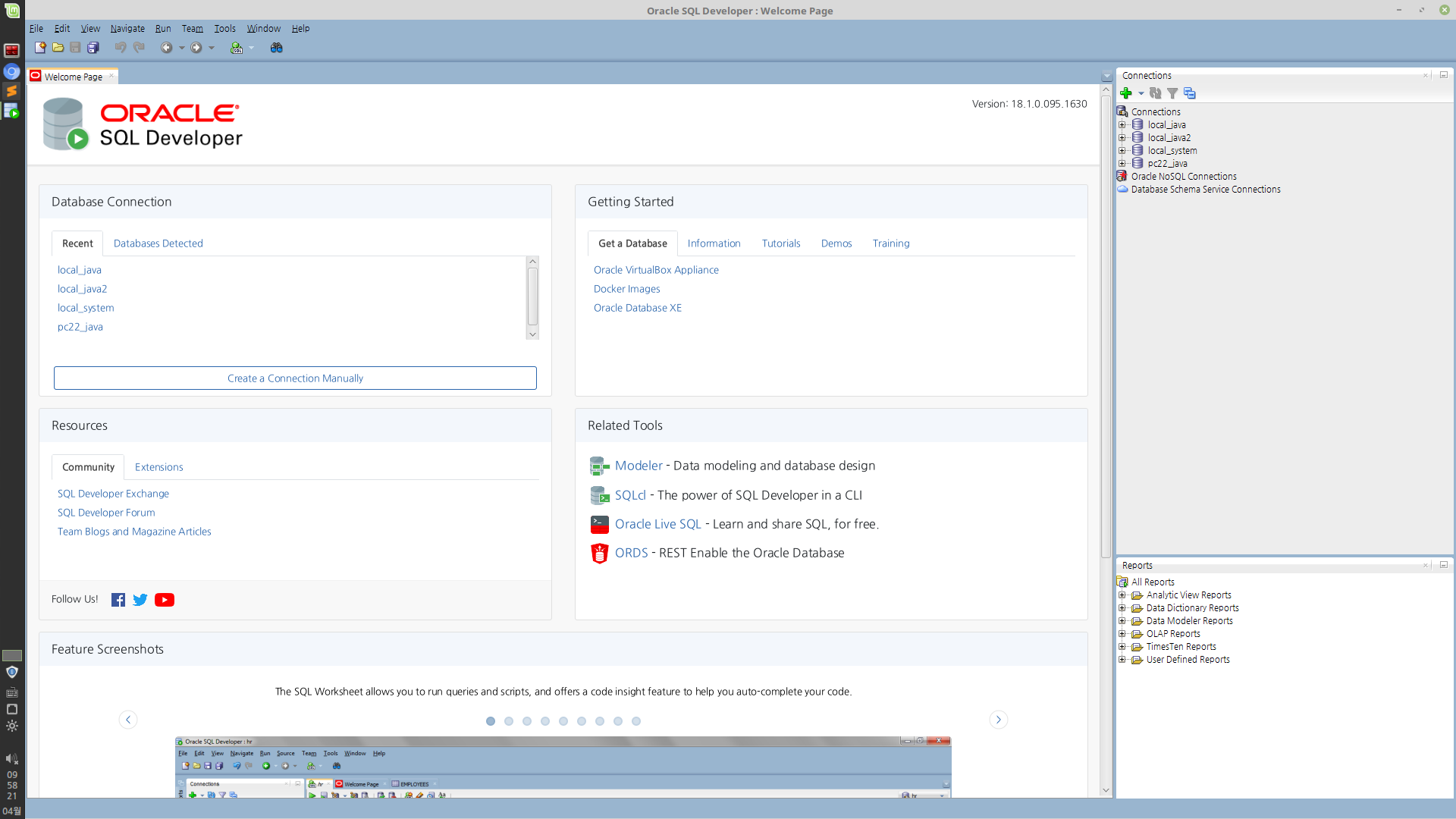


#8 리눅스 민트 시작



[과제6] 자바 기반의 개발 환경을 구축하고 해당 화면을 캡쳐해서 제출한다.

(개발환경 : DB, Java, Eclipse를 설치하고 해당 화면을 캡쳐한다.)

# DB

# Java



# Eclipse

