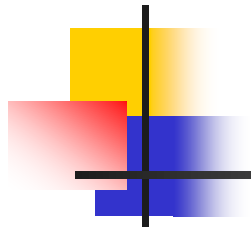


데이터베이스 프로그래밍

(소프트웨어 개발 트랙)



제 1부 고급 SQL

프로젝트 데이터베이스 개요

오라클 개요



프로젝트 데이터베이스 개요

- 수강신청 시스템
 - 요구사항 분석
 - 개념적 데이터 모델
 - 수강신청 시스템 데이터베이스 생성
 - 수강신청 어플리케이션
- 학습하게 될 기술



요구사항 분석 : 수강신청 시스템



사용사례 명세 : 로그인

- 사용 사례명 : 로그인

- 액터 : 학생

- 선행조건 : 없음

- 주요 흐름

1. 학생은 학번과 패스워드를 입력한다.
2. 시스템은 수강 신청을 가능하게 한다.
이때 시스템은 학번이 맞는지(E-1), 패스워드가 맞는지 (E-2) 확인한다.

- 예외 흐름

- E-1

1. 시스템은 학번이 맞는지 검사한다.
2. 시스템은 학생에게 학번이 맞지 않음을 알린다.

- E-2

1. 시스템은 패스워드가 맞는지 검사한다.
2. 시스템은 학생에게 패스워드가 맞지 않음을 알린다.



사용사례 명세 : 사용자 정보 수정

- 사용 사례명 : 사용자 정보 수정
 - 액터 : 학생
 - 선행조건 : 로그인
 - 주요 흐름
 1. 시스템은 로그인한 사용자 정보(주소, 비밀번호)를 보여준다.
 2. 학생은 사용자 정보를 수정한다.
이때, 시스템은 비밀번호가 맞는지 검사한다. (E-1)
 - 예외 흐름
 - E-1
 1. 시스템은 비밀번호가 4자리 이상이고,
공란이 포함되어 있지 않은지 검사한다.
 2. 비밀번호가 4자리 미만이거나 공란이 포함되어 있으면,
시스템은 수정이 불가능함을 알린다.



사용사례 명세 : 수강신청 입력(1)

■ 사용 사례명 : 수강신청 입력

- 액터 : 학생
- 선행조건 : 로그인
- 주요 흐름

1. 시스템은 아직 수강 신청하지 않은 과목들을 학생에게 보여준다.
2. 학생은 수강신청 하고자 하는 과목을 선택한다.
3. 시스템은 선택된 과목을 수강 신청된 것으로 등록한다.
이때, 시스템은 최대학점을 초과했는지(E-1),
동일한 과목을 신청했는지 (E-2),
해당 과목에 대한 수강신청 인원이 초과되었는지(E-3),
동일한 시간의 다른 과목이 이미 수강신청 되었는지(E-4)
를 검사한다.



사용사례 명세 : 수강신청 입력(2)

- 예외 흐름

- E-1

1. 시스템은 선택된 과목에 따라, 총 수강신청 과목의 총 학점이 18학점이 초과되는지 검사한다.
2. 18학점을 초과하면, 시스템은 수강신청이 될 수 없음을 알린다.

- E-2

1. 시스템은 선택된 과목이 이미 수강 신청되어 있는 과목인지 검사한다.
2. 이미 수강 신청되어 있는 과목인 경우, 시스템은 수강신청이 될 수 없음을 알린다.



사용사례 명세 : 수강신청 입력(3)

- 예외 흐름

- E-3

1. 시스템은 해당 과목에 대한 수강신청 인원이 초과되었는지 검사한다.
2. 수강신청 인원이 초과된 과목인 경우, 시스템은 수강신청이 될 수 없음을 알린다.

- E-4

1. 시스템은 동일한 시간의 다른 과목이 이미 수강 신청 되었는 지 검사한다.
2. 동일한 시간의 다른 과목이 이미 수강 신청되어 있는 경우, 시스템은 수강신청이 될 수 없음을 알린다.



사용사례 명세 : 수강신청 삭제

■ 사용 사례명 : 수강신청 삭제

- 액터 : 학생

- 선행조건 : 로그인

- 주요 흐름

1. 시스템은 수강신청 년도와 학기를 얻는다. 이 때, 수강 신청 년도와 학기는 현재 날짜가 11월, 12월인 경우는 다음 년도 1학기인 것으로 하고, 1월~4월인 경우는 현재 년도 1학기로 하며, 5월~10월은 현재 년도 2학기인 것으로 한다.
2. 시스템은 수강신청 년도와 학기에 해당하는 수강신청 내역을 학생에게 보여준다.
3. 학생은 삭제하고자 하는 과목을 선택한다.
4. 시스템은 선택된 과목을 수강신청 내역에서 삭제한다.



사용사례 명세 : 수강신청 조회

- 사용 사례명 : 수강신청 조회

- 액터 : 학생

- 선행조건 : 로그인

- 주요 흐름

1. 사용자는 조회하고자 하는 수강신청 년도와 학기를 입력한다.
사용자가 수강신청 년도와 학기를 입력하지 않는 경우는,
시스템은 현재 날짜를 기준으로 수강신청 년도와 학기를 얻어낸다
2. 시스템은 조회하고자 하는 수강신청 년도와 학기에 해당하는
수강신청 내역을 학생에게 보여준다.
3. 시스템은 해당 년도와 학기에 해당하는 수강신청 총 과목수와
총 학점을 학생에게 보여준다.



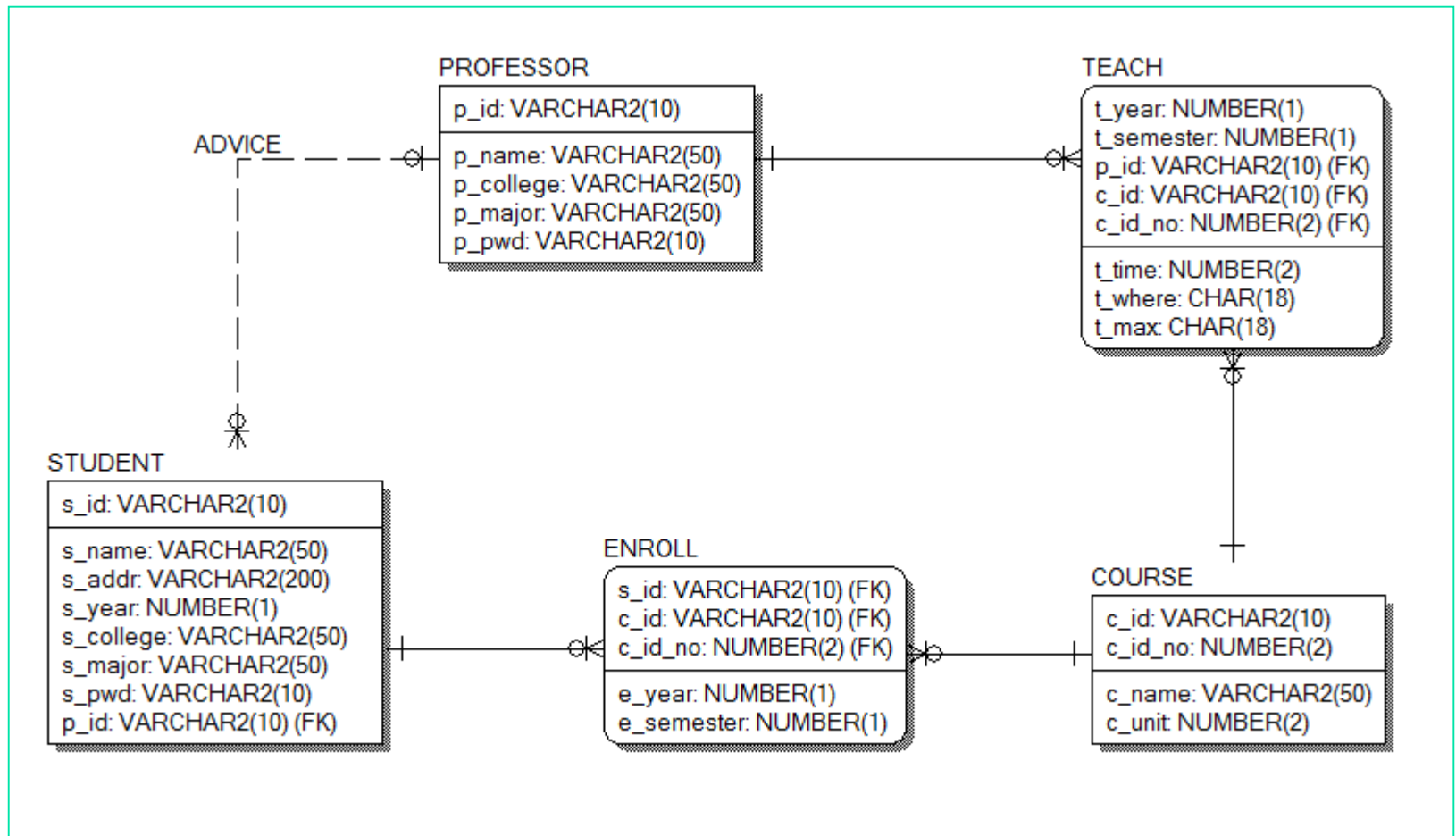
사용사례 명세 : 로그아웃

- 사용 사례명 : 로그아웃
 - 액터 : 학생
 - 선행조건 : 로그인
 - 주요 흐름
 1. 학생은 로그아웃을 선택한다.
 2. 시스템은 수강신청을 불가능하게 한다.



개념적 데이터 모델

ERD



엔터티 / 관계명	의 미
STUDENT	학생
COURSE	과목
ENROLL	등록하다
PROFESSOR	교수
TEACH	강의하다



ERD - STUDENT 엔터티

◆ STUDENT 엔터티

속성	의미
s_id	학번
s_name	이름
s_addr	주소
s_year	학년
s_college	단과대학
s_major	전공
s_pwd	패스워드

◆ COURSE 엔터티

속성	의미
c_id	과목 번호
c_id_no	분반
c_name	과목명
c_unit	학점

ERD – PROFESSOR 엔터티

◆ PROFESSOR 엔터티

속성	의미
p_id	교수 번호
p_name	교수 이름
p_college	단과대학
p_major	전공
p_pwd	패스워드

◆ ENROLL 엔터티

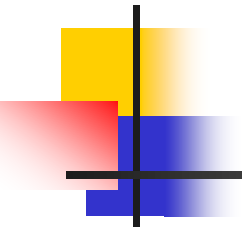
속성	의미
s_id	학번
c_id	과목번호
c_id_no	분반
e_year	등록 년도
e_semester	등록 학기



ERD – TEACH 엔터티

◆ TEACH 엔터티

속성	의미
p_id	교수 번호
c_id	과목 번호
c_id_no	분반
t_year	강의 년도
t_semester	강의 학기
t_time	강의 시간
t_where	강의실
t_max	최대 인원



수강신청 시스템 데이터베이스 생성



DDL – STUDENT 테이블 생성

```
CREATE TABLE student
(
    s_id          VARCHAR2(10),
    s_name        VARCHAR2(50) not null,
    s_addr        VARCHAR2(200),
    s_year        NUMBER(1) not null,
    s_college     VARCHAR2(50) not null,
    s_major       VARCHAR2(50) not null,
    s_pwd         VARCHAR2(10) not null,
    CONSTRAINT s_pk PRIMARY KEY (s_id)
);
```



DDL – COURSE 테이블 생성

```
CREATE TABLE course
(
  c_id          VARCHAR2(10),
  c_id_no       NUMBER(1),
  c_name        VARCHAR2(50),
  c_unit        NUMBER(1),
  CONSTRAINT c_pk PRIMARY KEY (c_id, c_id_no)
);
```



DDL – PROFESSOR 테이블 생성

```
CREATE TABLE professor
(
  p_id VARCHAR2(10),
  p_name  VARCHAR2(50) not null,
  p_college  VARCHAR2(50) not null,
  p_major    VARCHAR2(50) not null,
  p_pwd      VARCHAR2(10) not null,
  CONSTRAINT p_pk PRIMARY KEY (p_id)
);
```



DDL – ENROLL 테이블 생성

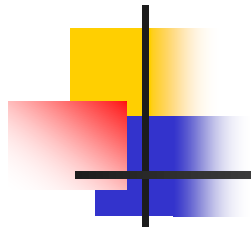
```
CREATE TABLE enroll
(
    s_id          VARCHAR2(10),
    c_id          VARCHAR2(10),
    c_id_no       NUMBER(1),
    e_year        NUMBER(1),
    e_semester    NUMBER(1),

    CONSTRAINT e_pk PRIMARY KEY (s_id, c_id, c_id_no),
    CONSTRAINT e_c_id_fk FOREIGN KEY (c_id, c_id_no)
    REFERENCES course (c_id, c_id_no)
);
```



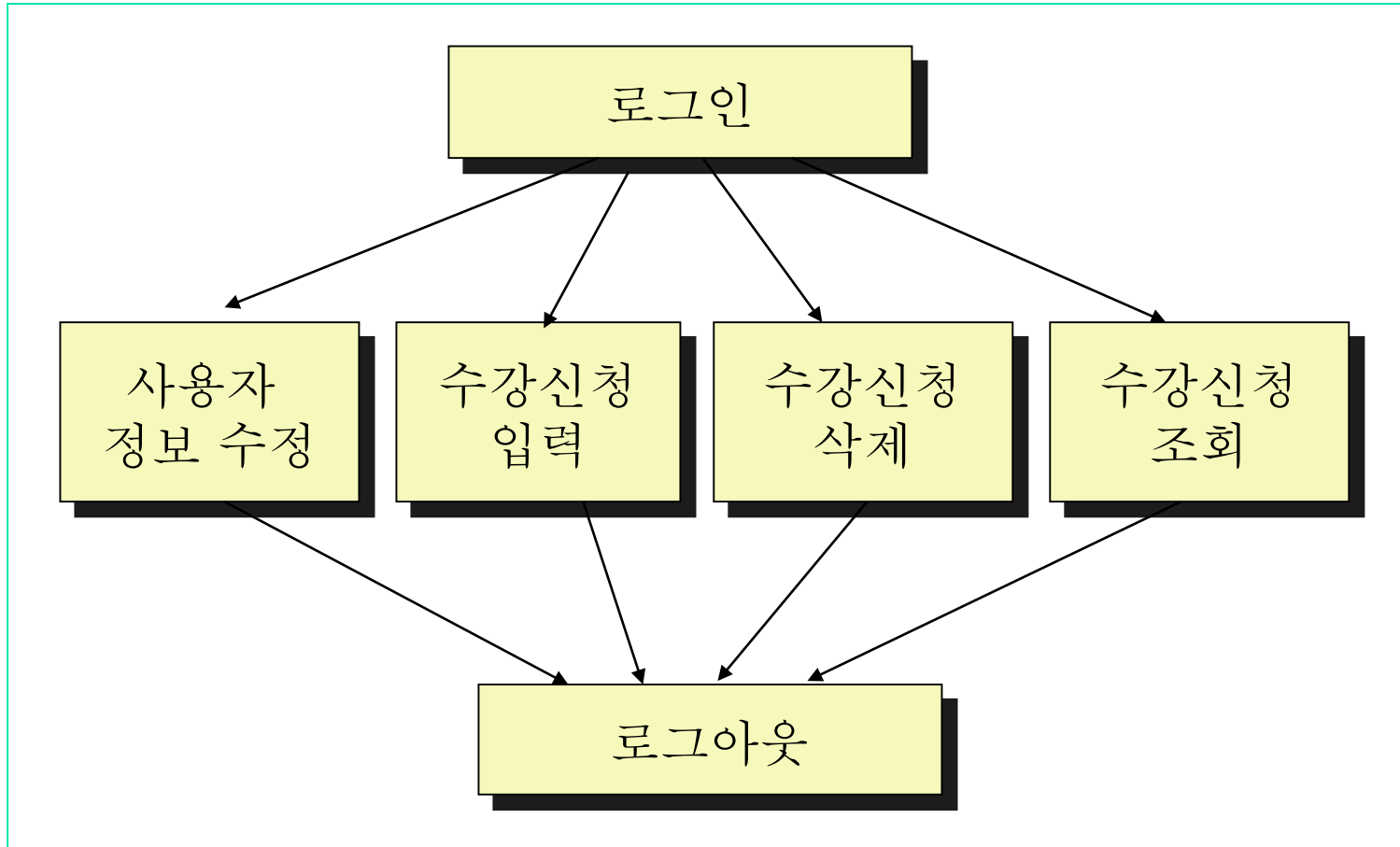
DDL – TEACH 테이블 생성

```
CREATE TABLE teach
(
  p_id          VARCHAR2(10),
  c_id          VARCHAR2(10),
  c_id_no       NUMBER(1),
  t_year        NUMBER(1),
  t_semester    NUMBER(1),
  t_time        NUMBER(1),
  t_where       VARCHAR2(50),
  t_max         NUMBER(2),
  CONSTRAINT t_pk PRIMARY KEY (p_id, c_id, c_id_no, t_year, t_semester ),
  CONSTRAINT t_c_id_fk FOREIGN KEY (c_id, c_id_no)
    REFERENCES course (c_id, c_id_no)
);
```

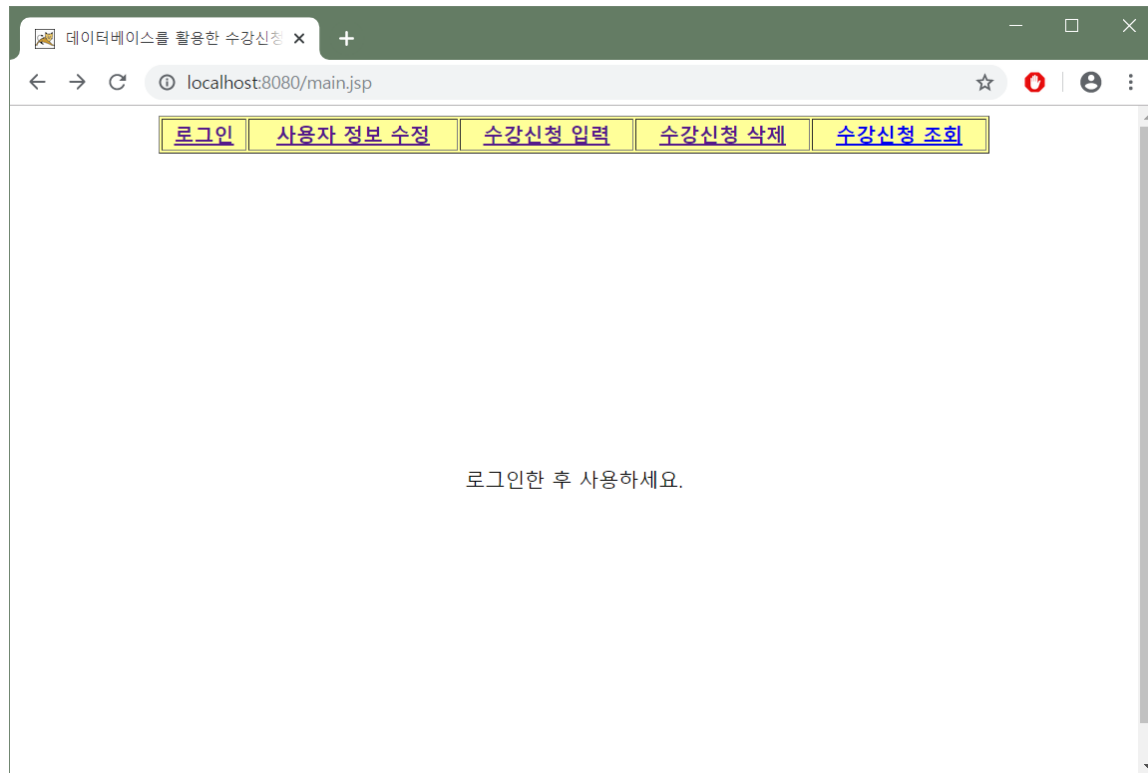



수강신청 어플리케이션

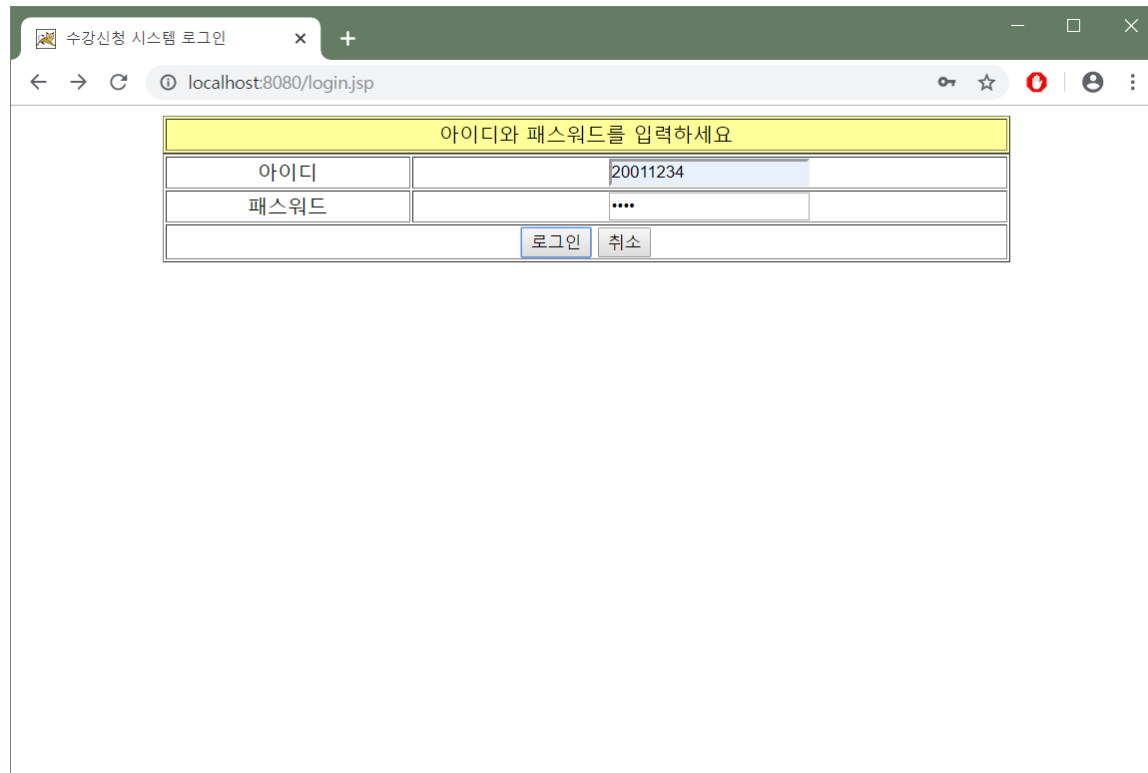
수강신청 어플리케이션 구성도



메인 화면



로그인 화면



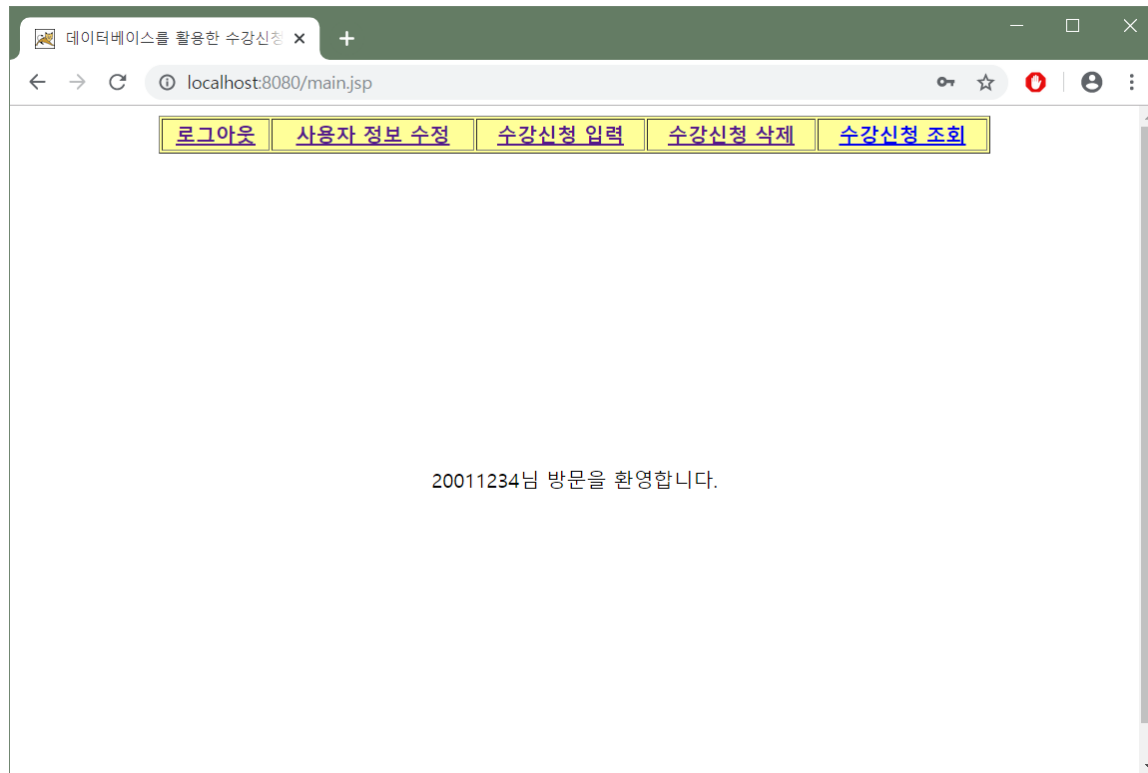
수강신청 시스템 로그인

localhost:8080/login.jsp

아이디와 패스워드를 입력하세요

아이디	20011234
패스워드
<input type="button" value="로그인"/> <input type="button" value="취소"/>	

로그인 후의 화면



사용자 정보 수정 화면

수강신청 사용자 정보 수정

← → ↻ localhost:8080/update.jsp ☆ 🔴 👤 ⋮

[로그아웃](#) [사용자 정보 수정](#) [수강신청 입력](#) [수강신청 삭제](#) [수강신청 조회](#)


주소	경기도 화성군 송산면 고정1리 540-2
패스워드
<input type="button" value="수정"/>	

수강신청 입력 화면



과목번호	분반	과목명	학점	수강신청
C800	3	데이터베이스 프로그래밍	3	신청
C900	3	객체지향 윈도우즈 프로그래밍	3	신청
M100	3	멀티미디어 개론	3	신청
M200	3	선형대수	3	신청
M300	3	그래픽 활용	3	신청
M400	3	윈도우즈 프로그래밍	3	신청
M700	3	게임 프로그래밍	3	신청

수강신청 삭제 화면



과목번호	분반	과목명	학점	수강신청
C100	3	컴퓨터 프로그래밍	3	삭제
C200	3	자료구조	3	삭제
C300	3	알고리즘	3	삭제
C400	3	데이터베이스 시스템	3	삭제
C500	3	운영체제	3	삭제

수강신청 조회 화면

수강신청 조회

localhost:8080/select.jsp?year=2024&semester=1

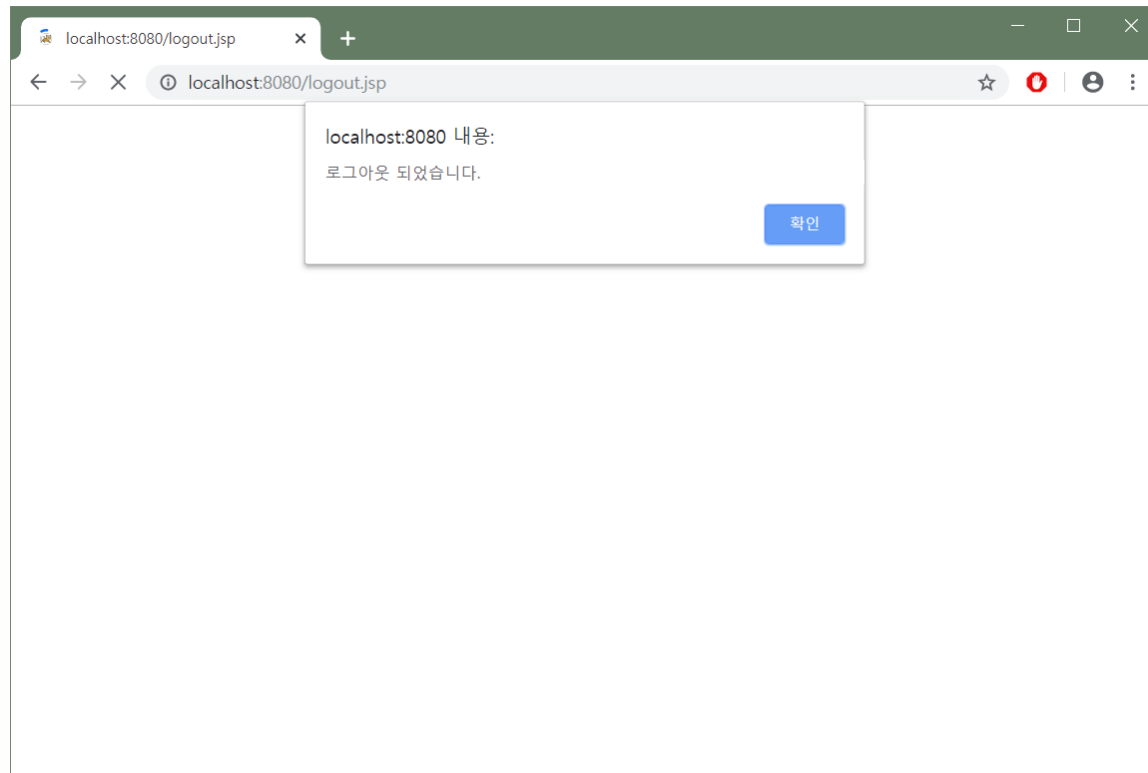
[로그아웃](#) [사용자 정보 수정](#) [수강신청 입력](#) [수강신청 삭제](#) [수강신청 조회](#)

과목번호	분반	과목명	학점
C100	3	컴퓨터 프로그래밍	3
C200	3	자료구조	3
C300	3	알고리즘	3
C400	3	데이터베이스 시스템	3
C500	3	운영체제	3

총신청과목수	5
총신청학점	15

2024 년도 1 학기 [조회](#)

로그아웃 화면





학습하게 될 기술 : 고급 SQL

- 조인
 - 조인
 - 외부 조인
 - 셀프 조인
- 서브 질의
- 무결성 제약조건
 - 무결성 제약조건 개요
 - 무결성 제약조건 생성
 - 무결성 제약조건 삭제
 - 무결성 제약조건 조회



학습하게 될 기술 : 고급 SQL

- 데이터 처리 함수
 - 데이터 처리 함수 개요
 - 문자형 함수
 - 숫자형 함수
 - 날짜형 함수
 - 변환 함수
- 뷰
 - 뷰 개요
 - 뷰 생성
 - 뷰 삭제
 - 뷰를 통한 데이터 변경



학습하게 될 기술 : 저장 프로시저

- 데이터베이스 내에서 **SQL**과 프로그래밍 언어의 통합 기법
 - 오라클 **PL/SQL**
 - 개념 및 특징
 - 문법
 - 개발 방법
 - 프로시저 및 함수
 - **PL/SQL**을 이용한 프로시저 및 함수 작성
 - 프로시저 및 함수 실행
 - 트리거
 - 트리거 개요
 - 트리거 생성 및 실행

- 자바와 표준 데이터베이스 **API(JDBC)**를 이용한 데이터베이스 연동 기술
 - JDBC의 개념
 - JSP에서 JDBC를 이용한 데이터베이스 연동
 - 저장 프로시저 활용 방법
 - 소프트웨어 컴포넌트 기술 중 하나인 자바 빈즈를 사용한 데이터베이스 어플리케이션 프로그래밍 기법
 - 데이터베이스 연결 풀링을 통한 데이터베이스 어플리케이션 프로그래밍의 효율화
 - 수강신청 어플리케이션 프로그램 완성



학습하게 될 기술 : 실전 프로젝트 예제

■ 실전 프로젝트

- 실무적인 요구사항으로부터 **ER-Win**을 활용하여 데이터베이스 모델링 수행
- 데이터베이스를 구축해 봄으로써 실전 프로젝트 경험
 - 요구사항 명세
 - 데이터베이스 모델링
 - 엔티티 관계 모델링
 - 인덱스
- *[저장 프로시저]*
- *[데이터베이스 어플리케이션 개발]*

- 오라클 개요
- SQL, SQL*Plus, PL/SQL
- SQL*PLUS의 실행
- 도움말 명령어
- Oracle DB에 대한 데이터 보기
- 셸 명령어 실행
- SQL 문 편집 명령어



오라클 (Oracle) 개요

- 오라클
 - Database/ERP와 이에 대한 솔루션을 제공하는 회사 이름
 - Oracle DBMS
 - 관계형 데이터베이스(RDB) 모델에 기초하여 만들어졌으며 이후 객체-관계형(Object-Relational) 모델 지원



SQL, SQL*Plus, PL/SQL

- SQL(**S**tructured **Q**uery **L**anguage)
 - 데이터베이스 구조를 정의하고, 데이터를 처리하기 위한 표준 언어
 - 기능에 따라 3가지로 구분
 - 데이터 정의어 (**DDL** : Data Definition Language)
 - 데이터 조작어 (**DML** : Data Manipulation Language)
 - 데이터 제어어 (**DCL** : Data Control Language)



SQL, SQL*Plus, PL/SQL

■ SQL*Plus

- 데이터베이스의 데이터를 조작하기 위한 **SQL** 명령과 **SQL**에 프로그래밍 요소를 추가한 **PL/SQL** 블록수행 가능
- **SQL**과 **PL/SQL**블록의 실행, 저장, 편집 가능
- 질의 결과를 리포트 형태로 생성
- 오라클 데이터베이스 사이에 데이터 복사 가능
- 데이터베이스 관리 가능



SQL, SQL*Plus, PL/SQL

- PL/SQL

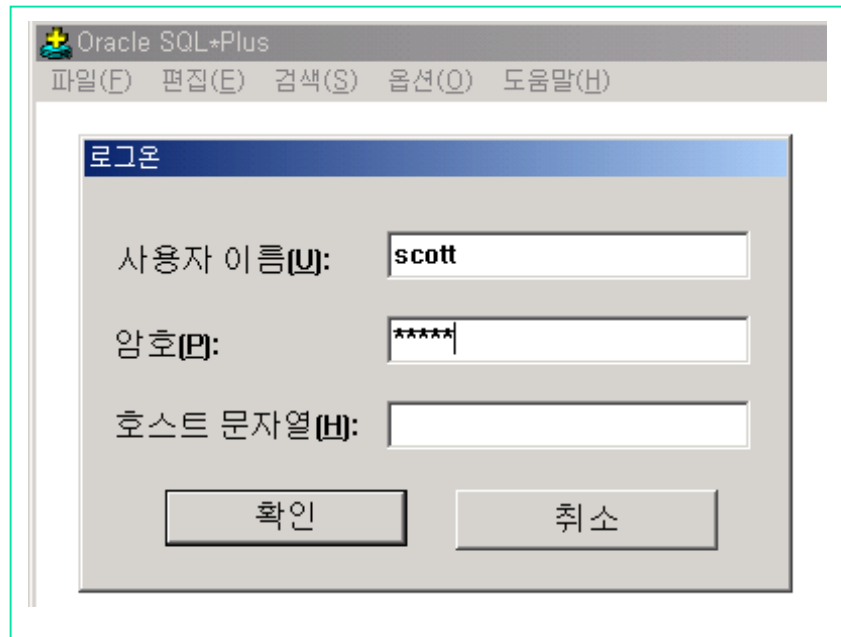
(Oracle's **P**rocedural **L**anguage extension to **SQL**)

- SQL 문장에서 변수 정의, 조건 처리, 반복 처리 등을 지원하며, 오라클 자체에 내장되어 있는 절차적 언어 (procedural language)

SQL*PLUS의 실행

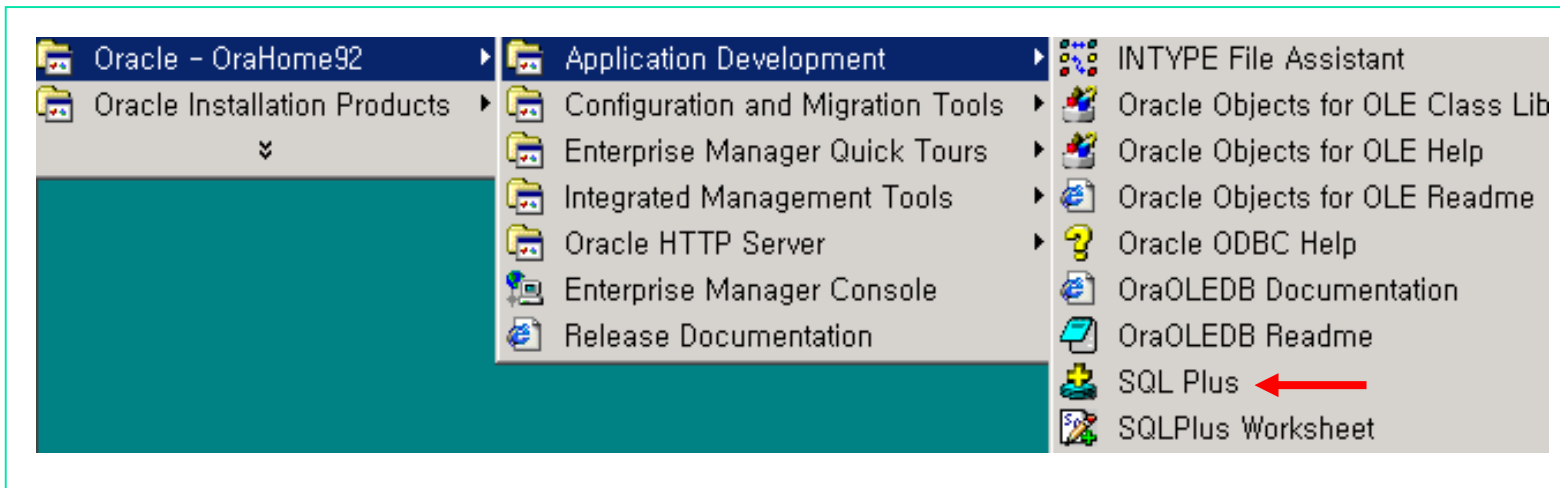
■ 실행 방법

- 시작 메뉴에서 시작하거나 윈도우의 도스 창에서 **sqlplus** 입력
- 예) 사용자 **scott**, 암호 **tiger**로 접속
 - 호스트 문자열은 로컬에서 수행하는 경우 공백



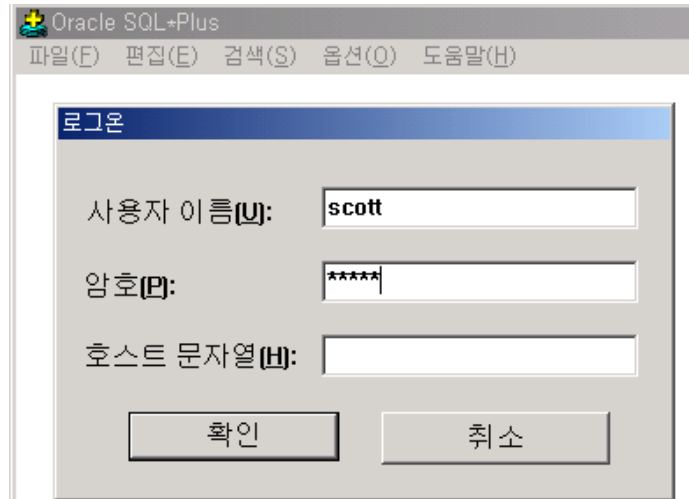
SQL*PLUS의 실행

- 방법 1
 - 윈도우 시작 메뉴를 이용한 접속



Cont.

- scott / tiger 로 접속



```
Oracle SQL*Plus
파일(F)  편집(E)  검색(S)  옵션(O)  도움말(H)

SQL*Plus: Release 9.2.0.1.0 - Production on 화 Dec 24 16:00:38 2002
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

다음에 접속됨:
Personal Oracle9i Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

→ SQL> |
```

SQL*PLUS의 실행

- 방법 2
 - 도스창에서 **sqlplus**를 입력하여 접속
 - **sqlplus** 사용자아이디 / 비밀번호

```
C:\WINNT\System32\cmd.exe - sqlplus scott/tiger
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>sqlplus scott/tiger

SQL*Plus: Release 9.2.0.1.0 - Production on 화 Dec 24 16:01:26 2002

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

다음에 접속됨:
Personal Oracle9i Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

SQL>
```


SQL*PLUS의 실행 - 종료

- 종료 방법
 - **exit**나 **quit** 명령 실행

```
C:\WINNT\System32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\W>sqlplus scott/tiger

SQL*Plus: Release 9.2.0.1.0 - Production on 화 Dec 24 16:01:26 2002

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

다음에 접속됨:
Personal Oracle9i Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production


SQL> exit
Personal Oracle9i Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production에서 분리되었습니다.

C:\W>_
```

도움말 명령어

■ 도움말

- **help** 명령으로 도움말을 볼 수 있다.
- 예 : help index

```
SQL> help index;   
Enter Help [topic] for help.  
  
@          COPY          PAUSE          SHUTDOWN  
@@         DEFINE       PRINT          SPOOL  
/          DEL           PROMPT         SQLPLUS  
ACCEPT     DESCRIBE     QUIT          START  
APPEND     DISCONNECT   RECOVER       STARTUP  
ARCHIVE LOG EDIT        REMARK        STORE  
ATTRIBUTE  EXECUTE      REPFOOTER     TIMING  
BREAK      EXIT        REPHEADER     TTITLE  
BTITLE     GET         RESERVED WORDS <SQL> UNDEFINE  
CHANGE     HELP        RESERVED WORDS <PL/SQL> VARIABLE  
CLEAR      HOST        RUN           WHENEVER OSERROR  
COLUMN     INPUT       SAVE          WHENEVER SQLERROR  
COMPUTE    LIST        SET  
CONNECT    PASSWORD    SHOW
```

SQL> _

- 예 : help quit

```
SQL> help quit ←
```

```
QUIT <Identical to EXIT>
```

```
----
```

Commits or rolls back all pending changes, logs out of Oracle, terminates SQL*Plus and returns control to the operating system. In iSQL*Plus, commits or rolls back all pending changes, stops processing the current iSQL*Plus script and returns focus to the Input area. In iSQL*Plus click the Logout button to log out of Oracle.

```
<EXIT|QUIT> [SUCCESS|FAILURE|WARNING|n|variable|:BindVariable]
[COMMIT|ROLLBACK]
```

```
SQL>
```



Oracle DB에 대한 데이터 보기 (1)

- Oracle DB에 대한 데이터, 즉 테이블 이름, 컬럼 이름, 사용자 이름, 데이터베이스 오브젝트의 구조 등을 보기를 원할 때
 - Data Dictionary에 대한 질의를 통해 가능
- Oracle DB 내에 있는 테이블들
 - **User tables** : 사용자에게 의해 만들어지고 유지되는 테이블의 집합
 - **Data Dictionary** : Oracle Server에 의해 만들어지고 유지되는 테이블들의 집합



Oracle DB에 대한 데이터 보기 (2)

- **DESCRIBE** : 테이블의 구조를 보기 위한 명령
 - 형식
 - `DESC table_name;`
 - `describe table_name;`
 - 예제
 - `SQL > DESC EMP;`
- 현재 사용 가능한 모든 테이블 보기
 - `SQL > select * from tab;`
 - `SQL > select * from tabs;`
 - `SQL > select table_name from tabs;`

Cont.

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
BONUS	TABLE	
DEPARTMENT	TABLE	
DEPT	TABLE	
EMP	TABLE	
FRIEND	TABLE	
PHOTOS	TABLE	
SALGRADE	TABLE	
STUDENT	TABLE	
TEST	TABLE	
VIEW_AUG_SALARY	VIEW	
VIEW_DEPT30	VIEW	
VIEW_ED	VIEW	
VIEW_EDJOIN30	VIEW	

13 개의 행이 선택되었습니다.

결과는 사용자
마다 다를수 있다.

Oracle DB에 대한 데이터 보기 (3)

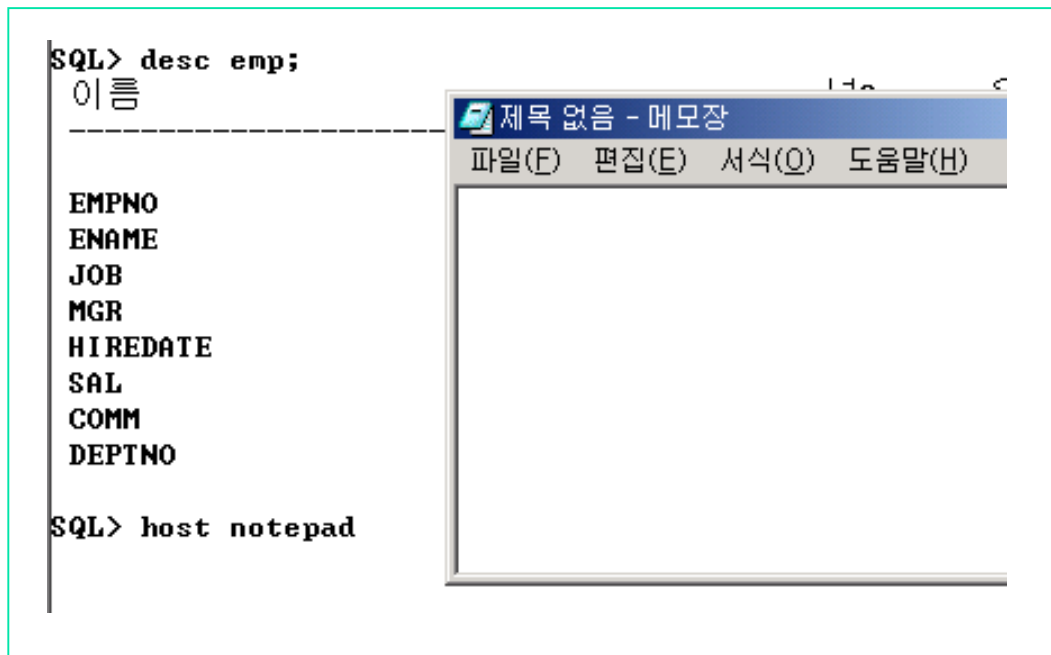
- 예 : emp 테이블 구조보기
 - SQL> desc emp;

```
SQL> desc emp;
이름
```

	널?	유형
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

셸 명령어 실행

- 셸 명령어 실행
 - sql*plus 내에서 윈도우의 프로그램 구동 가능
 - **host** 명령 이용
 - 예: SQL> **host** notepad;



The screenshot shows a SQL*Plus command window on the left and a Notepad application window on the right. The SQL*Plus window displays the output of the 'desc emp;' command, listing the columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. Below this, the command 'SQL> host notepad' is entered. The Notepad window, titled '제목 없음 - 메모장', is open and empty, demonstrating the successful execution of the 'host' command to launch an external application.

```
SQL> desc emp;  
이름  
-----  
EMPNO  
ENAME  
JOB  
MGR  
HIREDATE  
SAL  
COMM  
DEPTNO  
  
SQL> host notepad
```




SQL문 편집 명령어

- 최근 SQL문 편집
 - SQL*PLUS의 편집 명령어
 - SQL버퍼 - 사용자가 가장 최근에 입력한 SQL명령문 또는 PL/SQL 블록을 저장하는 공간

SQL문 편집 명령어

명령어	약어	목적
APPEND text	A text	마지막 행에 text 추가
CHANGE /old/new	C /old/new	한 라인에서 old를 new로 변경
CHANGE /text	C /text	한 라인에서 text 삭제
CLEAR BUFFER	CL BUFF	버퍼의 모든 내용을 지움
DEL (DEL *)	없음	현재 행 삭제
DEL n	없음	n번 행 삭제
DEL m n	없음	m번 행부터 n번 행까지 삭제
INPUT	I	하나 이상의 행 추가
INPUT text	I text	text로 구성된 행 추가
LIST	L	SQL 버퍼의 모든 행 나열
LIST n	L n or n	N번 행을 나열

SQL문 편집 명령어

- 최근 SQL문 편집 (예제 1/3)
 - 'list' 명령은 SQL버퍼의 모든 행 나열
 - 3행의 '*' 표시는 3행이 선택된 상태 의미

```
SQL> select empno, ename, job
2  from emp
3  where empno=7900;
```

EMPNO	ENAME	JOB
7900	JAMES	CLERK

```
SQL> list
1  select empno, ename, job
2  from emp
3* where empno=7900
SQL>
```

SQL문 편집 명령어

- 최근 SQL문 편집 (예제 2/3)
 - 'list 1'은 1행 선택
 - 'a, sal'은 1행의 끝에 ', sal' 추가
 - 'run'은 편집된 SQL 문 실행

```
SQL> list 1
1* select empno, ename, job
SQL> a, sal
1* select empno, ename, job, sal
SQL> run
1 select empno, ename, job, sal
2 from emp
3* where empno=7900
```

EMPNO	ENAME	JOB	SAL
7900	JAMES	CLERK	950

```
SQL>
```

SQL문 편집 명령어

- 최근 SQL문 편집 (예제 3/3)
 - 'list 3'은 3행을 선택
 - 'change /7900/7902'는 '7900'을 '7902'로 변경

```
SQL> list3  
  3* where empno=7900  
SQL> change /7900/7902  
  3* where empno=7902  
SQL> run  
  1 select empno, ename, job, sal  
  2 from emp  
  3* where empno=7902
```

EMPNO	ENAME	JOB	SAL
7902	FORD	ANALYST	3000

```
SQL> _
```

SQL문 편집 명령어

- 최근 SQL문의 재실행

- 가장 최근에 실행한 (SQL 버퍼에 기록되어 있는) SQL문을 여러 번 다시 반복해서 실행하고자 하는 경우에 사용
- 'run' 또는 '/'로 실행가능

```
SQL> run ←
  1 select empno, ename, job, sal
  2 from emp
  3* where empno=7902

      EMPNO ENAME      JOB              SAL
-----
      7902 FORD        ANALYST         3000

SQL> / ←

      EMPNO ENAME      JOB              SAL
-----
      7902 FORD        ANALYST         3000

SQL>
```



SQL문을 파일로 저장

- SQL문을 파일로 저장 실행

명 령 어	목 적
get 파일명	파일의 내용을 SQL버퍼에 저장
save 파일명	SQL 버퍼의 내용을 파일에 저장
start 파일명	파일에 저장된 내용 실행
@ 파일명	파일에 저장된 내용 실행

Cont.

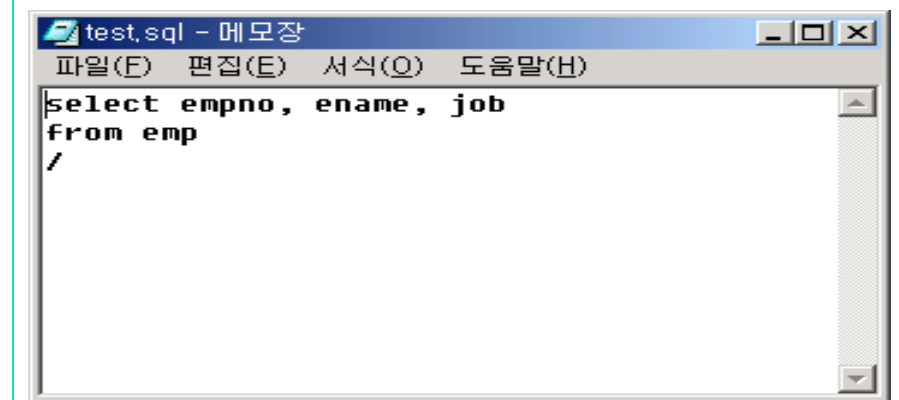
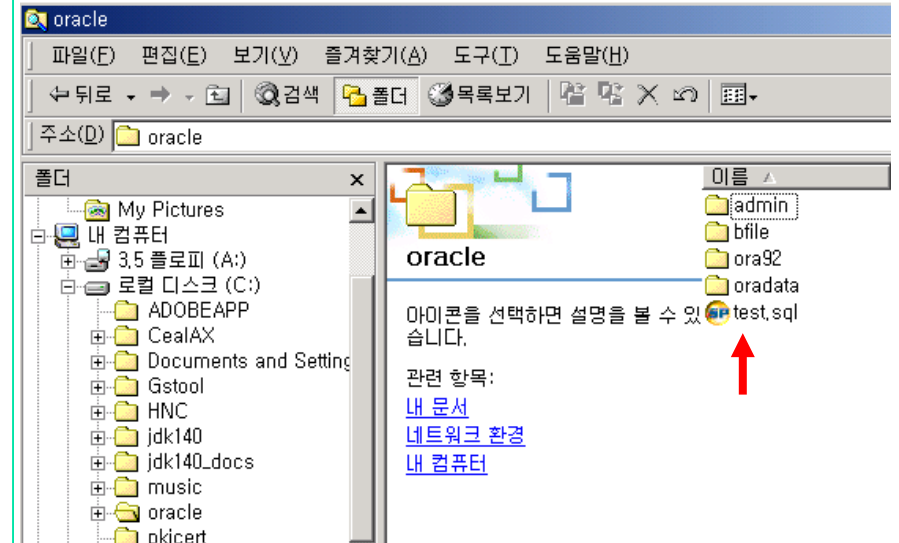
```
SQL> select empno, ename, job
2 from emp;
```

EMPNO	ENAME	JOB
7369	SMITH	CLERK
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	SALESMAN
7876	ADAMS	CLERK

EMPNO	ENAME	JOB
7900	JAMES	CLERK
7902	FORD	ANALYST
7934	MILLER	CLERK

14 개의 행이 선택되었습니다.

```
SQL> save c:\w\oracle\test.sql
file c:\w\oracle\test.sql<이>가 생성되었습니다
SQL>
```





파일로 저장된 SQL 문 실행

- 파일로 저장된 SQL 문 실행
 - ‘get’ 명령어

```
SQL> get c:\oracle\test.sql
  1  select empno, ename, job
  2* from emp
SQL> _
```

- ‘@’ 와 ‘start’ 명령어

Cont.

```
SQL> @ c:\oracle\test.sql
```

EMPNO	ENAME	JOB
7369	SMITH	CLERK
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	SALESMAN
7876	ADAMS	CLERK

EMPNO	ENAME	JOB
7900	JAMES	CLERK
7902	FORD	ANALYST
7934	MILLER	CLERK

14 개의 행이 선택되었습니다.

```
SQL> start c:\oracle\test.sql
```

EMPNO	ENAME	JOB
7369	SMITH	CLERK
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	SALESMAN
7876	ADAMS	CLERK

EMPNO	ENAME	JOB
7900	JAMES	CLERK
7902	FORD	ANALYST
7934	MILLER	CLERK

14 개의 행이 선택되었습니다.

```
SQL>
```

화면에 출력된 내용을 파일로 저장

- 화면에 출력된 내용을 파일로 저장
 - 'spool' 명령어

```
SQL> spool c:\Woracle\test.txt ←
SQL> select ename from emp where deptno=10;

ENAME
-----
CLARK
KING
MILLER

SQL> spool off ←
SQL> █
```

Cont.

