

2017년 1학기 데이터베이스	개 발 완 료 보 고 서	문서번호	-
		페이지번호	1/4 페이지
		작 성 자	임지훈, 권도혁
		작성일자	2017. 06. 01.

프로젝트 명	영화예매시스템 데이터베이스 구축 프로젝트		
개요	<p>일반 사용자가 사용하는 기능과 관리자가 사용하는 기능을 별도로 구분하여 구현한 영화 예매 시스템과 데이터베이스를 구축하였다.</p> <p>일반 사용자에서는 사용자에게 해당되는 테이블 내용을 보여주는 전체출력 및 요구사항명세서에서 나열한 5가지의 검색기능과 나머지 각 1개씩의 삽입, 삭제, 수정기능을 구현하였다.</p> <p>관리자에서는 관리자에게 해당되는 테이블 내용을 보여주는 전체출력 및 요구사항명세서에서 나열한 9가지의 검색기능과 나머지 각 1개씩의 삽입, 삭제, 수정기능을 구현하였다.</p>		
개발기간	2017/03/25 ~ 2017/06/01		
개발 완료일자	2017/06/01		
프로젝트 팀	학번/이름	13010975 / 권도혁	
	학번/이름	13010979 / 임지훈	
산출물 목록 (소스 코드 포함)	<p>1 - 요구사항명세서 작성 (블랙보드로 제출 완성)</p> <p>2 - 스키마정의서 (블랙보드로 제출 완성)</p> <p>3 - 중간 소스코드 (리눅스 계정으로 제출 완성)</p> <p>4 - 개발완료보고서 및 최종 소스코드(리눅스 계정으로 제출 완성)</p>		
기타	<p>사용자모드에서 조건부 검색기능은 5개, 관리자모드에서 조건부 검색기능은 9개이며, 내포질의와 그룹바이/해빙을 이용한 검색은 각각 3개씩으로 구현했습니다. 코드에서 주석부분에 한글은 리눅스에서 볼 경우 깨져서 보일 수 있습니다.</p>		

* 아래에 대해서 자유 형식으로 상세하게 작성한다.

1. 프로그램의 상세한 구조

A. 메인메뉴

처음에 프로그램을 실행시키면 메인 메뉴가 출력된다. 메인메뉴에서 숫자 1을 입력하면 유저모드로 실행이 되며 2를 누르면 관리자모드로 실행이 된다. 3번을 누르면 데이터베이스 내 테이블의 기존에 있던 내용이 삭제되며 처음 상태에 입력되어져 있던 내용들로 다시 채워진다. (초기화진행)

0을 누르면 프로그램이 종료된다. (모든 메뉴에서 0번은 종료를 한다.) 구조를 보면 다음과 같다.

a) 메인 메뉴

- a-1) 유저모드
- a-2) 관리자모드
- a-3) 데이터베이스 리셋
- a-4) 종료

B. 유저모드 (메인메뉴의 a-1에 해당함)

유저모드의 메뉴에서 1번은 유저에 대한 테이블 내용을 검색할 수 있다. 1번 메뉴를 눌러서 들어가게 되면 여기서 1번은 유저테이블의 전체 출력, 2번은 사용자가 예매한 예매정보에 대한 정보출력, 3번은 사용자가 지금까지 봐온 영화의 장르에 대한 카운팅 값을 보여준다. 4번은 사용자의 등급 정보에 대해서 검색 할 수 있으며, 5번은 사용자가 쿠키를 주문한 내역을 출력한다. 6번은 사용자의 이메일을 입력하면 이벤트에 응모한 내역이 출력된다. 0번을 누르면 이전 메뉴로 돌아간다.

다시 유저모드로 돌아가서 유저모드에서의 메인메뉴의 2번을 누르면 Insert data 기능을 이용할 수 있다. 이는 유저가 이벤트를 응모하는 기능을 구현한 것이다. 방법은 처음에 사용자id를 입력하고 응모하고 싶은 영화 관리번호를 누르면 응모가 완료되며 당첨여부는 30%의 확률로 응모하는 순간 바로 결정된다. 유저모드에서 3번은 사용자가 영화를 예매한 예매정보를 삭제하는 것으로 티켓을 예매했다가 취소하는 기능을 구현한 것이다. 방법은 자신의 예매관리번호를 입력하면 삭제된다. 모든 단계에서도 그렇지만 영어를 입력할 때 꼭 대소문자를 확인해서 입력하여야 한다.

유저모드에서 4번은 자신이 예매한 정보를 수정하는 기능이다. 티켓을 예매했을 때 영화나 시간을 바꾸는 기능을 구현한 것으로, 처음에 자신의 티켓 관리번호를 입력하고 그 다음에는 자신이 보고 싶은 영화를 다시 입력한다. 이때 입력하는 것은 영화관리번호이다. 그 다음에는 상영시간을 선택하고 이어서 영화관과 좌석을 선택한다. 마지막으로 1번은 카드 2번은 돈인데 결제수단을 선택하면 끝이 난다.

마찬가지로 0번을 누르면 이전 단계로 돌아간다.

b) 사용자메뉴

b-1) 검색기능

- b-1-1) 전체출력
- b-1-2) 예매정보 출력
- b-1-3) 유저가 본 영화 장르 카운팅정보 출력 - [그룹바이 사용](#)
- b-1-4) 유저의 등급 출력
- b-1-5) 유저의 쿠키 구매내역 출력
- b-1-6) 유저의 이벤트 응모내역 출력(이메일을 입력받음) - [내포질의 사용](#)
- b-1-0) 이전단계로 돌아가기

b-2) 삽입기능

b-3) 삭제기능

b-4) 수정기능

b-0) 이전단계로 돌아가기

C. 관리자모드 (메인메뉴의 a-2에 해당함)

관리자모드 메뉴에서 1번은 검색기능이며 1번을 눌러서 들어가면 1개의 전체출력과 9개의 조건부 출력이 있다. 1번은 전체출력, 2번은 영화제목을 입력하면 해당하는 영화의 모든 정보 출력, 3번은 전체/남성/여성예매율을 4번은 입력한 등급의 회원정보를 5번은 입력한 금액 이상의 할인을 받은 회원정보와 총 할인금액을 6번은 영화이름을 입력하면 해당 영화의 리뷰를 볼 수 있으며 7번은 유저가 쿠키를 주문한 총 금액을 볼 수 있고, 8번은 쿠키의 타입을 입력하면 해당 타입에 속해있는 쿠키 정보를 볼 수 있다. 9번은 이벤트에 당첨된 회원(T)과 탈락된 회원(F)의 정보를 볼 수 있으며, 10번은 쿠폰번호를 입력하면 해당 쿠폰을 가지고 있는 회원의 아이디와 해당 쿠폰을 소유하고 있는 장수를 확인 할 수 있다. 0번은 이전단계로 간다.

다시 관리자모드로 돌아와서 2번은 영화정보를 입력하는 기능이다. 등록하고 싶은 영화관리번호를 입력하고, 영화이름, 감독이름, 배우이름, 영화장르, 개봉일자를 순서대로 입력을 하면 데이터가 db에 저장된다. 3번은 영화정보를 삭제하는 기능이다. 삭제하고 싶은 영화관리번호를 입력하면 해당 영화가 db에서 삭제된다. 4번은 영화정보를 수정하는 기능으로서 수정하고 싶은 영화관리번호를 입력하고 수정하고 싶은 영화 이름, 감독이름, 배우이름, 장르, 개봉일을 다시 입력하면 수정이 완료된다.

늘 같지만 관리자모드에서도 0번을 누르면 이전단계로 돌아간다.

c) 관리자메뉴

c-1) 검색기능

c-1-1) 전체출력

c-1-2) 영화정보 출력

c-1-3) 예매율 정보 출력 - [내포질의 사용](#)

c-1-4) 회원 등급 출력

c-1-5) 회원의 총 할인금액 출력 - [그룹바이 사용](#)

c-1-6) 영화에 대한 회원의 후기 출력 - [내포질의 사용](#)

c-1-7) 회원의 쿠키주문한 총 금액이 출력 - [그룹바이 사용](#)

c-1-8) 쿠키타입 입력시 해당 쿠키정보 출력

c-1-9) 이벤트에 당첨 또는 실패한 회원 정보를 출력

c-1-10) 쿠폰번호 입력시 해당 쿠폰을 가지고 있는 회원아이디 출력

c-1-0) 이전단계로 돌아가기

c-2) 삽입기능

c-3) 삭제기능

c-4) 수정기능

c-0) 이전단계로 돌아가기

2. 프로그래밍에서 사용된 데이터 구조 (struct 구조체 등)

: 문자열에 관해서는 일반적으로 c언어에서는 제공되지 않지만 proc에서 특별하게 제공되는 VARCHAR 형을 이용하였다. 이는 c언어로 변환 시 문자의 내용을 저장하는 arr과 길이를 저장하는 len이라는 2개의 필드를 가진 구조체로 변환이 된다. 그 외에는 구조체를 사용하지 않았다.

3. 프로그램 실행 방법

: prog1.pc 파일과 makefile이 있을텐데 Shell 프로그램 창에 make를 입력하고 엔터를 누르면 컴파일 이 되면서 c파일과 실행파일이 생성될 것이다.

그 후 ./prog1 이라고 입력을 하면 파일이 실행된다.

prog1 프로그램 목록이나 기능은 '프로그램의 상세한 구조'를 보면 된다 .

4. 주요 기능

- : 프로그램의 상세한 구조에 주요 기능을 소개했으나 주요기능에 대해서 간략하게 다시 설명한다면 우선 메인메뉴에 관리자모드와 사용자모드를 선택할 수 있고, 데이터베이스를 초기화 할 수 있다. 관리자모드와 사용자모드 각각 검색/삭제/수정/삽입 기능이 있으며 관리자모드는 검색기능이 전체 테이블 검색을 제외한 조건부 검색이 9개, 사용자모드는 마찬가지로 전체 테이블 검색을 제외한 조건부 검색이 5개이다.
- 사용자모드 검색에서는 내포질의와 그룹바이/해빙 기능을 사용한게 각각 1개씩이며, 관리자모드 검색에서는 각각 2개씩이다.
- 어느 메뉴에서나 뒤로가기 혹은 종료하기 버튼은 0번이다.
- 각각의 기능에 대한 자세한 설명은 '프로그램의 상세한 구조'를 참고하면 될 것 같다.

5. 후기

- : 대학교 1학년때 이런 저런 프로그램을 만들면서 데이터베이스는 사실 웬만한 프로그램을 개발한다면 꼭 건드려봤어야 했던 기능이다. mysql을 따로 공부하면서 데이터베이스, 테이블도 만들어보고, 여러 가지 검색/삽입/삭제/수정도 해봤지만 심도있게 다루지는 않았다. 단순한 값을 저장하고 꺼내쓰는게 전부였는데, 이번 데이터베이스 강의를 들으면서, 좀더 강력한 검색기능에 대해서 많이 배우게 됐고, 이론적인부분에서 그치는게 아닌 PROC와 SQL_PLUS를 통한 영화예매시스템을 구현하면서 실습적인 부분도 많이 배운것에 대해서 많은 도움이 됐다고 생각한다.

※ 여백 부족으로 줄이 밀리게 되면 주석처리한 내용이 다음줄로 바뀌면서 아래 코드를 가져다 실행시 오류가 발생 할 수 있습니다.

<최종 소스코드>

```
/* File Name : Program for Movie Theater
   Memeber : 13010979(Lim jihun), 13010975(Gwon dohyeok)
   Description : Connect to Oracle DB, then make over 14 tables
*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <memory.h>

/* Declare Part */
EXEC SQL BEGIN DECLARE SECTION;
char user_id[20];
char passwd[20];
int int_a;
int int_b;
int int_c;
int int_d;
VARCHAR var20_input[20];
VARCHAR var20_a[20];
VARCHAR var20_b[20];
VARCHAR var20_c[20];
VARCHAR var20_d[20];
VARCHAR var20_e[20];
VARCHAR var20_f[20];
VARCHAR var20_g[20];
VARCHAR var20_h[20];

VARCHAR lvar20_a[20];
VARCHAR lvar20_b[20];
VARCHAR lvar20_c[20];
VARCHAR lvar20_d[20];
VARCHAR lvar20_e[20];
VARCHAR lvar20_f[20];
VARCHAR lvar20_g[20];
VARCHAR lvar20_h[20];

VARCHAR var1_a[1];
VARCHAR var3_a[3];
VARCHAR var1_input[1];
VARCHAR emp_name[20];
int emp_dept;
```

```
EXEC SQL END DECLARE SECTION;
```

```
/* Include Part */
```

```
EXEC SQL INCLUDE SQLCA;
```

```
/* Declare Program Function */
```

```
int get_menu_num();      // get mode number (mode 1 -> USER | mode 2 -> ADMIN)
```

```
void reset_data();      // initialize data at beginning
```

```
void get_data_user();    //show all data for user mode (mode -> )
```

```
void get_data_admin();   //show all data for admin mode (mode -> )
```

```
void user_mode_menu();   //show user mode menu
```

```
void admin_mode_menu();  //show admin mode menu
```

```
void user_mode_show();   // Show all data for user
```

```
void user_mode_insert(); // Insert data for user
```

```
void user_mode_delete(); // Delete data for user
```

```
void user_mode_update(); // Update data for user
```

```
void admin_mode_show();  // Show all data for admin
```

```
void admin_mode_insert(); // Insert data for admin
```

```
void admin_mode_delete(); // Delete data for admin
```

```
void admin_mode_update(); // Update data for admin
```

```
void insert_all_data();  // Insert data all tables
```

```
void delete_all_data();  // Delete data all tables
```

```
void admin_mode_show_menu(); // Show admin show menu
```

```
void user_mode_show_menu(); // Show user show menu
```

```
void connect_db();       // Connect DATABASE
```

```
void variable_initialize();
```

```
void lvariable_initialize();
```

```
/* admin show menu */
```

```
void admin_show_movie(); // Show movie information
```

```
void admin_show_rating(); // Show movie rating
```

```
void admin_show_grade(); // Show user grade
```

```
void admin_show_sale(); // Show sale price
```

```
void admin_show_review(); // Show review and stars about movie
```

```
void admin_show_price(); // Show price
```

```
void admin_show_cookie(); //Show cookie
```

```
void admin_show_event(); // Show event
```

```
void admin_show_coupon(); //Show coupon
```

```
/* user show menu */
```

```
void user_show_book(); // Show book info
```

```
void user_show_genre(); // Show genre info
```

```
void user_show_grade(); // Show grade info
```

```
void user_show_order(); // Show order info
```

```
void user_show_event(); // Show event info
```

```

main()
{
    //Declare variables
    int mode = 0;
    int m_exit = 0; //check loop main?
    /* Connect Part */
    strcpy(user_id, "s13010979"); // DB ID
    strcpy(passwd, "gns315824"); // DB PW
    //Connect DB
    EXEC SQL CONNECT : user_id IDENTIFIED BY : passwd;
    if (sqlca.sqlcode < 0) // Fail access to DB
    {
        printf("Wn%s", sqlca.sqlerrm.sqlerrmc);
        EXEC SQL ROLLBACK WORK RELEASE;
        exit(1);
    }
    else // Success access to DB
    {
        printf("WnStatus -> Connected to Oracle.WnWn");

        //Start Menu (Loop start)
        while (!m_exit)
        {
            // 1) Choose one of user-mode(mode->1) or admin-mode(mode->2)
            mode = get_menu_num();
            switch (mode)
            {
                case 1: // USER MODE
                    user_mode_menu();
                    break;
                case 2: // ADMIN MODE
                    admin_mode_menu();
                    break;
                case 3: // RESET DATA
                    reset_data();
                    break;
                case 0: // EXIT
                    //EXIT Loof
                    m_exit = 1; //TERMINATE
                    break;
                default:
                    printf("Miss mode number! Recheck mode number.WnWn");
                    break;
            }
        }
    }
}

```

```

    }
    EXEC SQL COMMIT WORK RELEASE; //Disconnect DB
}
exit(0);
}

void reset_data()
{
    /* Reset data from all database(tables) */
    //Delete all data from tables
    delete_all_data();
    //Insert all data from tables
    insert_all_data();
    printf("\nCOMPLETE RESET DATA!!\n\n");
    EXEC SQL COMMIT WORK;
}

// 1) 사용자의 상태 전환 : 일반사용자 - 관리자
int get_menu_num()
{
    int _mode = 0;
    while (1)
    {
        printf("\n*Choose mode of User mode or Admin mode & Enter number!\n");
        printf("*****\n");
        printf("* 1 -> USER MODE      *\n");
        printf("*****\n");
        printf("* 2 -> ADMIN MODE       *\n");
        printf("*****\n");
        printf("* 3 -> RESET DATABASE *\n");
        printf("*****\n");
        printf("* 0 -> EXIT           *\n");
        printf("*****\n");
        printf("Input number >>> ");
        scanf("%d", &_mode);
        printf("\n");
        if (_mode == 1 || _mode == 2 || _mode == 3 || _mode == 0)
        {
            return _mode;
        }
        else
        {
            printf("You got the wrong number! \n\n");
            return _mode;
        }
    }
}

```



```

        break;
    }
}

// 2) 일반사용자 메뉴
void user_mode_menu()
{
    int _mode = -1;
    int u_exit = 0;
    while(!u_exit){
        printf("***** USER MODE *****\n\n");
        printf("Select sub menu!\n");
        printf("1 -> Show several data from table for USER\n");
        printf("2 -> Insert data\n");
        printf("3 -> Delete data\n");
        printf("4 -> Update(Change) data\n");
        printf("0 -> Go back & Reselect mode\n\n");
        printf("number >>> ");
        scanf("%d", &_mode);
        printf("\n");
        switch (_mode){
            case 0: // SWITCH MODE (Go back)
                u_exit = 1;
                break;
            case 1: // Show all data
                user_mode_show_menu();
                //user_mode_show();
                break;
            case 2: // Insert data
                user_mode_insert(); // insert Event information (User Application)
                break;
            case 3: // Delete data
                user_mode_delete(); // delete ticket information
                break;
            case 4: // Update data
                user_mode_update(); // update ticket information
                break;
            default:
                printf("Miss mode! Recheck mode number.\n");
                _mode = -1;
                break;
        }
    }
}

```

```
}
```

```
void user_mode_show_menu()
```

```
{
```

```
    int _mode = -1;
```

```
    int u_exit = 0;
```

```
    while(!u_exit){
```

```
        printf("***** USER MODE - SHOW *****WnWn");
```

```
        printf("Select sub menu!Wn");
```

```
        printf("1 -> Show several data from table for USERWn");
```

```
        printf("2 -> Show Book data from table for USERWn");
```

```
        printf("3 -> Show Genre data from table for USERWn");
```

```
        printf("4 -> Show Grade data from table for USERWn");
```

```
        printf("5 -> Show Cookie Order data from table for USERWn");
```

```
        printf("6 -> Show Event data from table for USERWn");
```

```
        printf("0 -> Go back & Reselect modeWnWn");
```

```
        printf("number >>> ");
```

```
        scanf("%d", &_mode);
```

```
        printf("Wn");
```

```
    switch (_mode){
```

```
        case 0: // SWITCH MODE (Go back)
```

```
            u_exit = 1;
```

```
            break;
```

```
        case 1: // Show all data
```

```
            user_mode_show();
```

```
            break;
```

case 2: // User id를 입력하면 그 사람이 예매한 예매내용(회원id, 영화제목, 결제방법)이 출력된다.

```
            user_show_book();
```

```
            break;
```

case 3: // User id를 입력하면 그 사람이 본 영화의 장르를 카운트한 정보를 볼 수 있다. (FINDER)

```
            user_show_genre();
```

```
            break;
```

```
        case 4: // User id를 입력하면 그 사람의 등급과 해당 설명(ex)을 볼 수 있다. (GRADE)
```

```
            user_show_grade();
```

```
            break;
```

case 5: // User id를 입력하면 그 사람의 간식 주문 내역(간식종류, 간식이름, 결제금액)을 볼 수 있다. (M_ORDER와 COOKIE 조인)

```
            user_show_order();
```

```
            break;
```

case 6: // 이메일을 입력하면 이벤트 당첨여부와 시사회 영화의 이름, 시사회 일시를 알 수 있다. (USER와 EVENT 내포질의)

```
            user_show_event();
```

```
            break;
```

```

        default:
            printf("Miss mode! Recheck mode number.\n");
            _mode = -1;
            break;
    }

}

}

void admin_mode_show_menu()
{
    int _mode = -1;
    int u_exit = 0;
    while(!u_exit){
        printf("***** ADMIN MODE - SHOW *****\n\n");
        printf("Select sub menu!\n");
        printf("1 -> Show all data from table for ADMIN\n");
        printf("2 -> Show Movie data from table for ADMIN\n");
        printf("3 -> Show Rating data from table for ADMIN\n");
        printf("4 -> Show Grade data from table for ADMIN\n");
        printf("5 -> Show Saled price data from table for ADMIN\n");
        printf("6 -> Show Review data from table for ADMIN\n");
        printf("7 -> Show Price data from table for ADMIN\n");
        printf("8 -> Show Cookie data from table for ADMIN\n");
        printf("9 -> Show Event data from table for ADMIN\n");
        printf("10 -> Show Coupon data from table for ADMIN\n");
        printf("0 -> Go back & Reselect mode\n\n");
        printf("number >>> ");
        scanf("%d", &_mode);
        printf("\n");
        switch (_mode){
            case 0: // SWITCH MODE (Go back)
                u_exit = 1;
                break;
            case 1: // Show all data
                admin_mode_show();
                break;
            case 2: // 영화이름 입력받아서 해당 영화 정보 보여주기
                admin_show_movie();
                break;
            case 3: // 영화이름 입력받아서 영화관리번호 알아낸다음 예매율 테이블 출력하기(내포질
의)
                admin_show_rating();
                break;

```

```

        case 4: // 입력한 등급에 해당하는 회원정보 출력
            admin_show_grade();
            break;
        case 5: // 회원ID로 그룹바이해서 특정금액을 입력받아 그 이상보다 할인받은 회원ID와
총 할인금액 출력
            admin_show_sale();
            break;
        case 6: // 영화이름 검색 시 영화관리번호를 알아낸다음 해당 영화의 평점과 후기를 출력
(내포질의)
            admin_show_review();
            break;
        case 7: // 특정금액을 입력받고, 입력받은 금액보다 이상의 주문을 한 회원id와 총 결제금
액 출력
            admin_show_price();
            break;
        case 8: // 쿠키 종류를 입력시 해당 쿠키 이름과 가격 출력
            admin_show_cookie();
            break;
        case 9: // 이벤트에 당첨된 회원id, 영화정보(영화관리번호, 영화이름, 영화상영일자) 보여
주기
            admin_show_event();
            break;
        case 10: // 쿠폰 종류 입력시 해당 쿠폰을 가지고 있는 회원id와 쿠폰, 회원이름을 출력
            admin_show_coupon();
            break;

        default:
            printf("Miss mode! Recheck mode number.\n");
            _mode = -1;
            break;
    }

}

}

// 3) 관리자 메뉴
void admin_mode_menu()
{
    int _mode = -1;
    int a_exit = 0;
    while(!a_exit){
        printf("***** ADMIN MODE *****\n\n");
        printf("Select sub menu!\n");
        printf("1 -> Show all data from table for ADMIN\n");
        printf("2 -> Insert new movie data\n");

```

```

printf("3 -> Delete movie data\n");
printf("4 -> Update(Change) movie data\n");
printf("0 -> Go back & Reselect mode\n\n");
printf("number >>> ");
scanf("%d", &_mode);
printf("\n");

```

```

switch (_mode){
    case 0: // SWITCH MODE (Go back)
        a_exit = 1;
        break;
    case 1: // Show all data
        admin_mode_show_menu();
        //admin_mode_show();
        break;
    case 2: // Insert data
        admin_mode_insert(); //insert movie information
        break;
    case 3: // Delete data
        admin_mode_delete(); //delete movie information
        break;
    case 4: // Update data
        admin_mode_update(); //update movie information
        break;
    default:
        printf("Miss mode! Recheck mode number.\n");
        break;
}

```

```

}

```

```

}

```

```

// Show all data at user mode

```

```

void user_mode_show()

```

```

{

```

```

    variable_initialize();

```

```

    printf("*****TABLES CONTENTS FOR USER*****\n\n");

```

```

    //Cursor for COOKIE table

```

```

    //1. CURSOR - DECLARE

```

```

    EXEC SQL DECLARE cookie_cursor CURSOR FOR

```

```

        SELECT COO_NO, COO_TYPE, COO_NAME, COO_PRICE FROM COOKIE;

```

```

    //WHERE deptno = :dept_number;

```

```

//2. CURSOR - OPEN
EXEC SQL OPEN cookie_cursor;
printf("*COOKIE TABLE Wn(COO_NO | COO_TYPE | COO_NAME | COO_PRICE)WnWn");
//3. CURSOR - FETCH
while (1)
{
    EXEC SQL FETCH cookie_cursor INTO : var20_a, : var20_b, : var20_c, : int_a;
    if (sqlca.sqlcode == 1403)
    {
        //printf("No Data Found !Wn");

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %dWn", var20_a.arr, var20_b.arr, var20_c.arr, int_a);
    variable_initialize();
}

//4. CURSOR - CLOSE
EXEC SQL CLOSE cookie_cursor;

//Cursor for BOOK table
EXEC SQL DECLARE book_cursor CURSOR FOR
    SELECT BO_NO,
    BO_USERID, BO_TICKETNO, BO_DATE, BO_NUM, BO_PAY FROM BOOK;

EXEC SQL OPEN book_cursor;
printf("*BOOK TABLE Wn(bo_no | bo_userId | bo_ticketNo | bo_date | bo_num |
bo_pay)WnWn");
while (1)
{
    EXEC SQL FETCH book_cursor INTO : var20_a, : var20_b, : var20_c, : var20_d, : int_a, :
var20_e;
    if (sqlca.sqlcode == 1403)
    {
        //printf("BOOK table - not yet dataWnWn");

printf("-----
-----Wn");

```

```

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %s | %d | %sWn", var20_a.arr, var20_b.arr, var20_c.arr,
var20_d.arr, int_a, var20_e.arr);
    variable_initialize();
}
EXEC SQL CLOSE book_cursor;

//Cursor for TICKET table
EXEC SQL DECLARE ticket_cursor CURSOR FOR
    SELECT ti_no,
           ti_movieNo, ti_timeView, ti_movieName, ti_userId, ti_thType, ti_seatNo, ti_movieGenre
                                           F R O M
TICKET;

EXEC SQL OPEN ticket_cursor;
printf("*TICKET TABLE Wn(ti_no | ti_movieNo | ti_timeView | ti_movieName | ti_userId |
ti_thType | ti_seatNo | ti_movieGenre)WnWn");
while (1)
{
    EXEC SQL FETCH ticket_cursor INTO : var20_a, : var20_b, : var20_c, : var20_d, :
var20_e, : var20_f, : var20_g, : var20_h;
    if (sqlca.sqlcode == 1403)
    {
        //printf("TICKET table - not yet dataWn");

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %s | %s | %s | %s | %sWn", var20_a.arr, var20_b.arr,
var20_c.arr, var20_d.arr, var20_e.arr, var20_f.arr, var20_g.arr, var20_h.arr);
    variable_initialize();
}
EXEC SQL CLOSE ticket_cursor;

```

```

//Cursor for REVIEW table
EXEC SQL DECLARE review_cursor CURSOR FOR
    SELECT re_no,
           re_movieNo, re_userId, re_stars, re_review
                                FROM REVIEW;

EXEC SQL OPEN review_cursor;
printf("*REVIEW TABLE Wn(re_no | re_movieNo | re_userId | re_stars | re_review)WnWn");
while (1)
{
    EXEC SQL FETCH review_cursor INTO : var20_a, : var20_b, : var20_c, : int_a, : var20_d;
    if (sqlca.sqlcode == 1403)
    {
        // printf("REVIEW table - not yet dataWn");

printf("-----Wn");
        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %d | %sWn", var20_a.arr, var20_b.arr, var20_c.arr, int_a,
var20_d.arr);
    variable_initialize();
}
EXEC SQL CLOSE review_cursor;

//Cursor for FINDER table
EXEC SQL DECLARE finder_cursor CURSOR FOR
    SELECT fi_no,
           fi_userId, fi_SF, fi_ro, fi_dr, fi_ho, fi_gr
                                FROM FINDER;

EXEC SQL OPEN finder_cursor;
printf("*FINDER TABLE Wn( fi_no | fi_userId | fi_tiNo | fi_SF | fi_ro | fi_dr | fi_ho | fi_gr
)WnWn");
while (1)
{
    EXEC SQL FETCH finder_cursor INTO : var20_a, : var20_b, : int_a, : int_b, : int_c, :
int_d, : var20_c;
    if (sqlca.sqlcode == 1403)

```



```

        {
            // printf("FINDER table – not yet dataWn");

printf("-----
-----Wn");

            break;
        }
        else if (sqlca.sqlcode != 0)
        {
            printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
            break;
        }
        printf("%s | %s | %d | %d | %d | %d | %sWn", var20_a.arr, var20_b.arr, int_a, int_b,
int_c, int_d, var20_c.arr);
        variable_initialize();
    }
    EXEC SQL CLOSE finder_cursor;

//Cursor for GRADE table
EXEC SQL DECLARE grade_cursor CURSOR FOR
    SELECT gr_grade,
        gr_userId, gr_grEx
            FROM GRADE;

EXEC SQL OPEN grade_cursor;
printf("*GRADE TABLE Wn(gr_grade | gr_userId | gr_grEx)WnWn");
while (1)
{
    EXEC SQL FETCH grade_cursor INTO : var20_a, : var20_b, : var20_c;
    if (sqlca.sqlcode == 1403)
    {
        //printf("GRADE table – not yet dataWn");

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %sWn", var20_a.arr, var20_b.arr, var20_c.arr);
    variable_initialize();
}

```

```

EXEC SQL CLOSE grade_cursor;

//Cursor for EVENT table
EXEC SQL DECLARE event_cursor CURSOR FOR
    SELECT ev_no,
           ev_userId, ev_movieNO, ev_Mdate, ev_check
           FROM EVENT;

EXEC SQL OPEN event_cursor;
printf("*EVENT TABLE Wn(ev_no | ev_userId | ev_movieNO | ev_Mdate | ev_check)WnWn");
while (1)
{
    EXEC SQL FETCH event_cursor INTO : var20_a, : var20_b, : var20_c, : var20_d, : var3_a;
    if (sqlca.sqlcode == 1403)
    {
        //printf("EVENT table - not yet dataWn");

printf("-----Wn");
        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %s | %sWn", var20_a.arr, var20_b.arr, var20_c.arr, var20_d.arr,
var3_a.arr);
    variable_initialize();
}
EXEC SQL CLOSE event_cursor;

//Cursor for COUPON table
EXEC SQL DECLARE coupon_cursor CURSOR FOR
    SELECT cou_no,
           cou_userId, cou_type, cou_num
           FROM COUPON;

EXEC SQL OPEN coupon_cursor;
printf("*COUPON TABLE Wn(cou_no | cou_userId | cou_type | cou_num)WnWn");
while (1)
{
    EXEC SQL FETCH coupon_cursor INTO : var20_a, : var20_b, : var20_c, : int_a;
    if (sqlca.sqlcode == 1403)
    {

```

```

        // printf("COUPON table - not yet dataWn");

printf("-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %dWn", var20_a.arr, var20_b.arr, var20_c.arr, int_a);
    variable_initialize();
}
EXEC SQL CLOSE coupon_cursor;

} //End function

// Insert data at user mode
void user_mode_insert()
{
    /* 이벤트 신청하기 (User Application)
        1. 회원id 선택
        2. 시사회 영화 선택 - M12 or M13
        3. 이벤트번호와 시사회 일시는 영화선택에 따라서 자동으로 넣어준다.
        4. 당첨여부는 랜덤으로 뽑아서 기록한다.
    */
    variable_initialize();
    //1 회원id 선택
    printf("*****USER MODE - INPUT*****Wn");
    //Show User ids
    //Cursor for M_USER table
    EXEC SQL DECLARE muser_cursor3 CURSOR FOR
        SELECT us_id
        FROM M_USER;

    EXEC SQL OPEN muser_cursor3;
    printf("*User Id List*Wn");
    while (1)
    {
        EXEC SQL FETCH muser_cursor3 INTO : lvar20_a;
        if (sqlca.sqlcode == 1403)
        {
            //printf("M_USER table - not yet dataWn");
            printf("-----Wn");

```

```

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf("Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s Wn", lvar20_a.arr);
    lvariable_initialize();
}
EXEC SQL CLOSE muser_cursor3;
/////

printf("Select User ID >>> "); scanf("%s", var20_a.arr); var20_a.len = strlen(var20_a.arr);

//시사회 영화 선택 M12 or M13
//M12와 M13 정보 보여주기
//Cursor for MOVIE table
EXEC SQL DECLARE movie_cursor3 CURSOR FOR
    SELECT mo_no, mo_name, mo_director, mo_actor, mo_genre, mo_Rdate
    FROM MOVIE
    WHERE MO_NO IN ('M12', 'M13');

EXEC SQL OPEN movie_cursor3;
printf("*MOVIE List Wn(mo_no | mo_name | mo_director | mo_actor | mo_mo_genre |
mo_Rdate)WnWn");
while (1)
{
    EXEC SQL FETCH movie_cursor3 INTO : lvar20_a, : lvar20_b, : lvar20_c, : lvar20_d, :
lvar20_e, : lvar20_f;
    if (sqlca.sqlcode == 1403)
    {
        // printf("MOVIE table - not yet dataWn");

printf("-----Wn");
        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %s | %s | %sWn", lvar20_a.arr, lvar20_b.arr, lvar20_c.arr,
lvar20_d.arr, lvar20_e.arr, lvar20_f.arr);

```

```

        lvariable_initialize();
    }
EXEC SQL CLOSE movie_cursor3;
//랜덤생성 (결과값을 var1_a 에 담는다.) - 30% 의 당첨률
int rand_num = 1 + rand()/10; //1부터 10 사이의 수를 추측한다.
if (rand_num % 3 == 0) {
    //당첨
    strcpy(var1_a.arr, "T"); var1_a.len = strlen(var1_a.arr);
}else {
    //실패
    strcpy(var1_a.arr, "F"); var1_a.len = strlen(var1_a.arr);
}
// 입력받기
int num_input = 1;
printf("----Info : 1 -> Select M12 | 2 -> Select M13---\n");
printf("Input number 1 or 2 >>> ");
scanf("%d", &num_input);
if (num_input == 1){
    // Select M12
    EXEC SQL INSERT INTO EVENT VALUES ('E1', :var20_a, 'M12', '2018-02-03', :var1_a);
} else if (num_input == 2){
    // Select M13
    EXEC SQL INSERT INTO EVENT VALUES ('E2', :var20_a, 'M13', '2018-02-15', :var1_a);
} else {
    // Select M12
    printf("You choose the wrong number. It will be basically selected at M12 Movie\n");
    EXEC SQL INSERT INTO EVENT VALUES ('E1', :var20_a, 'M12', '2018-02-03', :var1_a);
}

//2 - 데이터베이스에 추가시키기(꼭 변수들 초기화 시킨 다음에 해야 함)
//EXEC SQL INSERT INTO MOVIE VALUES (:var20_a, :var20_b, :var20_c, :var20_d, :var20_e,
:var20_f);

//3 - 입력한 데이터 다시 한번 확인
printf("\n**Your event application has been successfully received!**\n");
variable_initialize();
}

// Delete data at user mode
void user_mode_delete()
{
    // 1. 예매한 예매번호를 입력받음
    printf("*** USER MODE - DELETE ***\n");
    printf("Input BOOK NUMBER ex)B3 >>> "); scanf("%s", var20_input.arr); var20_input.len =
strlen(var20_input.arr);

```

// 2. 입력받은 예매번호로 예매(BOOK) 정보 삭제 (삭제하기 전 티켓번호 저장)

//Cursor for BOOK table

```
EXEC SQL DECLARE book_cursor3 CURSOR FOR
    SELECT bo_ticketNo
    FROM BOOK
    WHERE BO_NO = :var20_input;
```

EXEC SQL OPEN book_cursor3;

while (1)

```
{
    EXEC SQL FETCH book_cursor3 INTO : var20_a;
    if (sqlca.sqlcode == 1403)
    {
        //printf("BOOK table - not yet dataWnWn");
```

```
printf("-----
-----Wn");
```

```
        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
}
```

}

EXEC SQL CLOSE book_cursor3;

EXEC SQL DELETE FROM BOOK WHERE BO_NO = :var20_input;

EXEC SQL DELETE FROM TICKET WHERE TI_NO = :var20_a;

// 3. 영화정보 삭제 다시 한번 확인

```
printf("Wn**BOOK data which you input is successfully Delete in Database !**Wn");
variable_initialize();
```

}

// Update data at user mode

void user_mode_update()

{

// 1. 수정하고 싶은 티켓번호 입력

```
printf("*** USER MODE - UPDATE **Wn");
```

```
printf("Input Ticket number ex)TICKET4 >>> "); scanf("%s", var20_input.arr); var20_input.len
= strlen(var20_input.arr);
```

// 2. 영화정보 보여주고 영화관리번호 입력받기 (영화 선택하기)

```

printf("*** MOVIE MENU ***\n");
//Cursor for MOVIE table
EXEC SQL DECLARE movie_cursor4 CURSOR FOR
    SELECT mo_no, mo_name, mo_director, mo_actor, mo_genre, mo_Rdate
    FROM MOVIE;

EXEC SQL OPEN movie_cursor4;
printf("*MOVIE TABLE \n(mo_no | mo_name | mo_director | mo_actor | mo_genre |
mo_Rdate)\n\n");
while (1)
{
    EXEC SQL FETCH movie_cursor4 INTO : lvar20_a, : lvar20_b, : lvar20_c, : lvar20_d, :
lvar20_e, : lvar20_f;
    if (sqlca.sqlcode == 1403)
    {
        // printf("MOVIE table - not yet data\n");

printf("-----\n\n");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %d\n%s\n", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %s | %s | %s\n\n", lvar20_a.arr, lvar20_b.arr, lvar20_c.arr,
lvar20_d.arr, lvar20_e.arr, lvar20_f.arr);
    lvariable_initialize();
}
EXEC SQL CLOSE movie_cursor4;

printf("Input Movie manage number ex)M1 >>> "); scanf("%s", var20_a.arr); var20_a.len =
strlen(var20_a.arr);
// 3. 상영일정정보 보여주고 상영일정번호 입력받기 (날짜선택)
printf("*** MOVIE TIME VIEW ***\n");
//Cursor for TIMETABLE table
EXEC SQL DECLARE timetable_cursor3 CURSOR FOR
    SELECT ti_no,
    ti_movieNo, ti_time
    FROM TIMETABLE;

EXEC SQL OPEN timetable_cursor3;
printf("*TIMETABLE TABLE \n(ti_no | ti_movieNo | ti_time)\n\n");

```

```

while (1)
{
    EXEC SQL FETCH timetable_cursor3 INTO : lvar20_a, : lvar20_b, : lvar20_c;
    if (sqlca.sqlcode == 1403)
    {
        //printf("TIMETABLE table - not yet dataWn");

printf("-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %sWnWn", lvar20_a.arr, lvar20_b.arr, lvar20_c.arr);
    lvariable_initialize();
}
EXEC SQL CLOSE timetable_cursor3;

printf("Input Movie Time view ex)TIME1 >>> "); scanf("%s", var20_b.arr); var20_b.len =
strlen(var20_b.arr);
// 4. 영화관 정보 보여주고 영화관번호 입력받기 (영화관 선택, 좌석선택)
printf("*** MOVIE THEATER ***Wn");
//Cursor for THEATER table
EXEC SQL DECLARE theater_cursor3 CURSOR FOR
    SELECT th_type, th_seatRange, th_seatNum
    FROM THEATER;

EXEC SQL OPEN theater_cursor3;
printf("*THEATER TABLE Wn(th_type | th_seatRange | th_seatNum)WnWn");
while (1)
{
    EXEC SQL FETCH theater_cursor3 INTO : lvar20_a, : lvar20_b, : int_a;
    if (sqlca.sqlcode == 1403)
    {
        //printf("THEATER table - not yet dataWn");

printf("-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)

```



```

    {
        printf(" Oracle Error Detected!!: %d\n%s\n", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %d\n\n", lvar20_a.arr, lvar20_b.arr, int_a);
    lvariable_initialize();
}
EXEC SQL CLOSE timetable_cursor3;

```

```

    printf("Input Theater type    ex)NORMAL_A    >>> "); scanf("%s", var20_c.arr); var20_c.len =
strlen(var20_c.arr);
    printf("Input Theater seat    ex)B30    >>> "); scanf("%s", var20_d.arr); var20_d.len =
strlen(var20_d.arr);
    // 5. 결제방법 선택 1 -> 현금, 2 -> 카드
    printf("*** PAY METHOD ***\n");
    printf("---INFO : 1 -> CARD | 2 -> MONEY---\n");
    printf("Input Number 1 OR 2    >>> ");
    //memset(var20_a.arr, 0x00, sizeof(var20_a));
    int input_num;
    scanf("%d", &input_num);

    if (input_num == 1){strcpy(var20_e.arr, "CARD"); var20_e.len = strlen(var20_e.arr);}
    else if (input_num == 2){strcpy(var20_e.arr, "MONEY"); var20_e.len = strlen(var20_e.arr);}
    else {
        printf("You choose the wrong number! It will be saved as MONEY.\n\n");
        strcpy(var20_e.arr, "MONEY"); var20_e.len = strlen(var20_e.arr);
    }
}

```

```

//영화 이름 찾기
EXEC SQL SELECT mo_name, mo_genre INTO :var20_f, :var20_g FROM MOVIE WHERE
MO_NO = :var20_a;
EXEC SQL SELECT BO_NO INTO :var20_h FROM BOOK WHERE BO_TICKETNO =
:var20_input;
// 8. 티켓(TICKET) DB에 정보 수정
EXEC SQL UPDATE TICKET SET TI_MOVIENO=:var20_a, TI_TIMEVIEW=:var20_b,
TI_MOVIENAME=:var20_f, TI_THTYPE=:var20_c, TI_SEATNO=:var20_d, TI_MOVIEGENRE=:var20_g
WHERE TI_NO =:var20_input;
// 9. 예매(BOOK) DB에 정보 수정(결제방법 수정)
EXEC SQL UPDATE BOOK SET BO_PAY=:var20_e
WHERE BO_NO =:var20_h;

//3 - 수정한 데이터 다시 한번 확인
printf("\n**Ticket data is successfully Update in Database !**\n");

```

```

    printf("=> %s, %s, %s, %s, %s, %s, %s, %sWnWn", var20_input.arr, var20_a.arr, var20_b.arr,
var20_c.arr, var20_d.arr, var20_e.arr, var20_f.arr, var20_g.arr);

    variable_initialize();
}

// Show all data at admin mode
void admin_mode_show()
{

    variable_initialize();
    printf("*****TABLES CONTENTS FOR ADMIN*****WnWn");

    //Cursor for MOVIE table
    EXEC SQL DECLARE movie_cursor CURSOR FOR
        SELECT mo_no,
            mo_name, mo_director, mo_actor, mo_genre, mo_Rdate
                                FROM MOVIE;

    EXEC SQL OPEN movie_cursor;
    printf("*MOVIE TABLE Wn(mo_no | mo_name | mo_director | mo_actor | mo_genre |
mo_Rdate)WnWn");
    while (1)
    {
        EXEC SQL FETCH movie_cursor INTO : var20_a, : var20_b, : var20_c, : var20_d, :
var20_e, : var20_f;
        if (sqlca.sqlcode == 1403)
        {
            // printf("MOVIE table - not yet dataWn");

printf("-----Wn");
            break;
        }
        else if (sqlca.sqlcode != 0)
        {
            printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
            break;
        }
        printf("%s | %s | %s | %s | %s | %sWn", var20_a.arr, var20_b.arr, var20_c.arr,
var20_d.arr, var20_e.arr, var20_f.arr);
        variable_initialize();
    }
    EXEC SQL CLOSE movie_cursor;

```

```

//Cursor for RATING table
EXEC SQL DECLARE rating_cursor CURSOR FOR
    SELECT ra_no,
           ra_movieNo, ra_totalR, ra_maleR, ra_femaleR
           FROM RATING;

EXEC SQL OPEN rating_cursor;
printf("*RATING TABLE Wn(ra_no | ra_movieNo | ra_totalR | ra_maleR | ra_femaleR)WnWn");
while (1)
{
    EXEC SQL FETCH rating_cursor INTO : var20_a, : var20_b, : int_a, : int_b, : int_c;
    if (sqlca.sqlcode == 1403)
    {
        //printf("RATING table - not yet dataWnWn");

printf("-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %d | %d | %d Wn", var20_a.arr, var20_b.arr, int_a, int_b, int_c);
    variable_initialize();
}
EXEC SQL CLOSE rating_cursor;

//Cursor for TIMETABLE table
EXEC SQL DECLARE timetable_cursor CURSOR FOR
    SELECT ti_no,
           ti_movieNo, ti_time
           FROM TIMETABLE;

EXEC SQL OPEN timetable_cursor;
printf("*TIMETABLE TABLE Wn(ti_no | ti_movieNo | ti_time)WnWn");
while (1)
{
    EXEC SQL FETCH timetable_cursor INTO : var20_a, : var20_b, : var20_c;
    if (sqlca.sqlcode == 1403)
    {
        //printf("TIMETABLE table - not yet dataWn");

printf("-----Wn");

```

```

-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %sWn", var20_a.arr, var20_b.arr, var20_c.arr);
    variable_initialize();
}
EXEC SQL CLOSE timetable_cursor;

//Cursor for THEATER table
EXEC SQL DECLARE theater_cursor CURSOR FOR
    SELECT th_type,
           th_seatRange, th_seatNum
           FROM THEATER;

EXEC SQL OPEN theater_cursor;
printf("*THEATER TABLE Wn(th_type | th_seatRange | th_seatNum)WnWn");
while (1)
{
    EXEC SQL FETCH theater_cursor INTO : var20_a, : var20_b, : int_a;
    if (sqlca.sqlcode == 1403)
    {
        //printf("THEATER table - not yet dataWn");

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %dWn", var20_a.arr, var20_b.arr, int_a);
    variable_initialize();
}
EXEC SQL CLOSE timetable_cursor;

//Cursor for COUPONINFO table
EXEC SQL DECLARE couponIF_cursor CURSOR FOR
    SELECT couIF_type,

```

```

        coulF_benefit, coulF_ex
            FROM COUPONINFO;

EXEC SQL OPEN couponIF_cursor;
printf("*COUPONINFO TABLE Wn(coulF_type | coulF_benefit | coulF_ex)WnWn");
while (1)
{
    EXEC SQL FETCH couponIF_cursor INTO : var20_a, : var20_b, : var20_c;
    if (sqlca.sqlcode == 1403)
    {
        //printf("COUPONINFO table - not yet dataWn");

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf("Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %sWn", var20_a.arr, var20_b.arr, var20_c.arr);
    variable_initialize();
}
EXEC SQL CLOSE couponIF_cursor;

//Cursor for M_USER table
EXEC SQL DECLARE muser_cursor CURSOR FOR
    SELECT us_id,
        us_name, us_pw, us_email, us_sex
            FROM M_USER;

EXEC SQL OPEN muser_cursor;
printf("*M_USER TABLE Wn( us_id | us_name | us_pw | us_email | us_sex )WnWn");
while (1)
{
    EXEC SQL FETCH muser_cursor INTO : var20_a, : var20_b, : var20_c, : var20_d, :
var20_e;
    if (sqlca.sqlcode == 1403)
    {
        //printf("M_USER table - not yet dataWn");

printf("-----
-----Wn");

        break;

```

```

    }
    else if (sqlca.sqlcode != 0)
    {
        printf("Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %s | %sWn", var20_a.arr, var20_b.arr, var20_c.arr, var20_d.arr,
var20_e.arr);
    variable_initialize();
}
EXEC SQL CLOSE muser_cursor;

//Cursor for SALEINFO table
EXEC SQL DECLARE saleIF_cursor CURSOR FOR
    SELECT sa_no,
        sa_userId, sa_coutype, sa_addSale, sa_addSMoney, sa_totalSMoney
                                FROM SALEINFO;

EXEC SQL OPEN saleIF_cursor;
printf("*SALEINFO TABLE Wn(sa_no | sa_userId | sa_coutype | sa_addSale | sa_addSMoney
| sa_totalSMoney)WnWn");
while (1)
{
    EXEC SQL FETCH saleIF_cursor INTO : var20_a, : var20_b, : var20_c, : var20_d, : int_a, :
int_b;
    if (sqlca.sqlcode == 1403)
    {
        //printf("SALEINFO table – not yet dataWn");

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf("Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %s | %d | %dWn", var20_a.arr, var20_b.arr, var20_c.arr,
var20_d.arr, int_a, int_b);
    variable_initialize();
}
EXEC SQL CLOSE saleIF_cursor;

//Cursor for M_ORDER table

```

```

EXEC SQL DECLARE morder_cursor CURSOR FOR
    SELECT or_no,
        or_cooNo, or_userId, or_price
        FROM M_ORDER;

EXEC SQL OPEN morder_cursor;
printf("M_ORDER TABLE Wn(or_no | or_cooNo | or_userId | or_price)WnWn");
while (1)
{
    EXEC SQL FETCH morder_cursor INTO : var20_a, : var20_b, : var20_c, : int_a;
    if (sqlca.sqlcode == 1403)
    {
        //printf("M_ORDER table - not yet dataWn");

printf("-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf("Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %dWn", var20_a.arr, var20_b.arr, var20_c.arr, int_a);
    variable_initialize();
}
EXEC SQL CLOSE morder_cursor;

} //end function

// Insert data at admin mode - Movie data
void admin_mode_insert()
{
    variable_initialize();
    //1 - 영화관리번호, 영화이름, 영화감독, 출연배우, 장르, 영화개봉일 입력하게 하기
    printf("*****INPUT*****Wn");
    printf("Input Movie Management number(mo_no)   ex) M14 >>> "); scanf("%s", var20_a.arr);
    var20_a.len = strlen(var20_a.arr);
    printf("Input Movie Name(mo_name) >>> "); scanf("%s", var20_b.arr); var20_b.len =
    strlen(var20_b.arr);
    printf("Input Movie Director(mo_director) >>> "); scanf("%s", var20_c.arr); var20_c.len =
    strlen(var20_c.arr);
    printf("Input Movie Actor(mo_actor) >>> "); scanf("%s", var20_d.arr); var20_d.len =
    strlen(var20_d.arr);
    printf("Input Movie genre(mo_genre) >>> "); scanf("%s", var20_e.arr); var20_e.len =

```

```

strlen(var20_e.arr);
    printf("Input Movie Release date(mo_Rdate) >>> "); scanf("%s", var20_f.arr); var20_f.len =
strlen(var20_f.arr);

    //2 - 데이터베이스에 추가시키기(꼭 변수들 초기화 시킨 다음에 해야 함)
    EXEC SQL INSERT INTO MOVIE VALUES (:var20_a, :var20_b, :var20_c, :var20_d, :var20_e,
:var20_f);

    //3 - 입력한 데이터 다시 한번 확인
    printf("Wn**New Movie data which you input is successfully Input in Database !**Wn");
    printf("=> %s, %s, %s, %s, %s, %sWnWn", var20_a.arr, var20_b.arr, var20_c.arr, var20_d.arr,
var20_e.arr, var20_f.arr);
}

// Delete data at admin mode
void admin_mode_delete()
{
    variable_initialize();
    // 1. 삭제할 영화관리번호 입력받기
    printf("*****DELETE*****Wn");
    printf("Input Movie Management number(mo_no) which you want to delete >>> ");
    scanf("%s", var20_a.arr); var20_a.len = strlen(var20_a.arr);

    // 2. 입력받은 영화관리번호로 해당 영화정보 삭제
    EXEC SQL DELETE FROM MOVIE WHERE MO_NO =: var20_a;

    // 3. 영화정보 삭제 다시 한번 확인
    printf("Wn**Movie data which you input is successfully Delete in Database !**Wn");
}

// Update data at admin mode
void admin_mode_update()
{
    variable_initialize();
    //1 - 영화관리번호, 영화이름, 영화감독, 출연배우, 장르, 영화개봉일 입력하게 하기
    printf("*****UPDATE*****Wn");
    printf("Select Movie Management number(mo_no) which you want to change ex) M14 >>> ");
    scanf("%s", var20_a.arr); var20_a.len = strlen(var20_a.arr);
    printf("Input Movie Name(mo_name) >>> "); scanf("%s", var20_b.arr); var20_b.len =
strlen(var20_b.arr);
    printf("Input Movie Director(mo_director) >>> "); scanf("%s", var20_c.arr); var20_c.len =
strlen(var20_c.arr);
    printf("Input Movie Actor(mo_actor) >>> "); scanf("%s", var20_d.arr); var20_d.len =
strlen(var20_d.arr);
    printf("Input Movie genre(mo_genre) >>> "); scanf("%s", var20_e.arr); var20_e.len =

```



```

strlen(var20_e.arr);
    printf("Input Movie Release date(mo_Rdate) >>> "); scanf("%s", var20_f.arr); var20_f.len =
strlen(var20_f.arr);

    //2 - 데이터베이스에 추가시키기(꼭 변수들 초기화 시킨 다음에 해야 함)
    EXEC SQL UPDATE MOVIE SET MO_NAME=:var20_b, MO_DIRECTOR=:var20_c,
MO_ACTOR=:var20_d, MO_GENRE=:var20_e, MO_RDATE=:var20_f
    WHERE MO_NO =:var20_a;

    //3 - 수정한 데이터 다시 한번 확인
    printf("\n**Movie data is successfully Update in Database !**\n");
    printf("=> %s, %s, %s, %s, %s, %s\n\n", var20_a.arr, var20_b.arr, var20_c.arr, var20_d.arr,
var20_e.arr, var20_f.arr);
}

// Delete all data form tables
void delete_all_data()
{
    EXEC SQL DELETE FROM FINDER;
    EXEC SQL DELETE FROM SALEINFO;
    EXEC SQL DELETE FROM BOOK;
    EXEC SQL DELETE FROM TICKET;
    EXEC SQL DELETE FROM THEATER;
    EXEC SQL DELETE FROM GRADE;
    EXEC SQL DELETE FROM M_ORDER;
    EXEC SQL DELETE FROM COOKIE;
    EXEC SQL DELETE FROM EVENT;
    EXEC SQL DELETE FROM COUPON;
    EXEC SQL DELETE FROM COUPONINFO;
    EXEC SQL DELETE FROM TIMETABLE;
    EXEC SQL DELETE FROM RATING;
    EXEC SQL DELETE FROM REVIEW;
    EXEC SQL DELETE FROM M_USER;
    EXEC SQL DELETE FROM MOVIE;
}

// Insert all data at tables
void insert_all_data()
{
    // Movie TABLE
    EXEC SQL INSERT INTO MOVIE VALUES ('M1', 'GoG2', 'James Gunn', 'Chris Pratt', 'Fantasy',
'2017-05-15');
    EXEC SQL INSERT INTO MOVIE VALUES ('M2', 'The Boss Baby', 'Tom McGrath', 'Alec
Baldwin', 'Animation', '2016-04-08');
    EXEC SQL INSERT INTO MOVIE VALUES ('M3', 'The Sheriff In Town', 'KimHyeongJu',

```

```

'LeeSeongMin', 'Drama', '2015-08-07');
EXEC SQL INSERT INTO MOVIE VALUES ('M4', 'The King Case Note', 'MoonHyonSung',
'LeeSunGuen', 'Action', '2011-05-09');
EXEC SQL INSERT INTO MOVIE VALUES ('M5', 'The Mayor', 'ParkInJae', 'ChaeMinSik',
'Drama', '2014-07-03');
EXEC SQL INSERT INTO MOVIE VALUES ('M6', 'Smurfs', 'Kelly Asbury', 'Mandy Patinkin',
'Animation', '1998-11-24');
EXEC SQL INSERT INTO MOVIE VALUES ('M7', 'Fast and Furious', 'F. Gary Gray', 'Vin
Diesel', 'Action', '2017-03-08');
EXEC SQL INSERT INTO MOVIE VALUES ('M8', 'Dancer', 'Steven Cantor', 'Sergei Polunin',
'Drama', '2012-05-30');
EXEC SQL INSERT INTO MOVIE VALUES ('M9', 'The Unknown Girl', 'Jean-Pierre', 'Adele
Haenel', 'Drama', '2014-09-16');
EXEC SQL INSERT INTO MOVIE VALUES ('M10', 'Beauty and Beast', 'Bill Condon', 'Emma
Watson', 'Fantasy', '2017-04-08');
EXEC SQL INSERT INTO MOVIE VALUES ('M11', 'SuSuPeyong', 'HongJuYeon', 'HaJeongWo',
'Drama', '2011-09-27');
EXEC SQL INSERT INTO MOVIE VALUES ('M12', 'Kuroko Basketball', 'Shunsuke Tada',
'Kensho Ono', 'Animation', '2018-02-04');
EXEC SQL INSERT INTO MOVIE VALUES ('M13', 'The Plan', 'ChaeJinSeong', 'Henry E. Brady',
'Drama', '2018-02-16');
// M_USER TABLE
EXEC SQL INSERT INTO M_USER VALUES ('apple', 'Michel', 'a1234', 'e_apple@naver.com',
'MALE');
EXEC SQL INSERT INTO M_USER VALUES ('banana', 'John', 'b1234', 'e_banana@naver.com',
'MALE');
EXEC SQL INSERT INTO M_USER VALUES ('carrot', 'Helen', 'c1234', 'e_carrot@naver.com',
'FEMALE');
EXEC SQL INSERT INTO M_USER VALUES ('dog', 'Daina', 'd1234', 'e_dog@naver.com',
'FEMALE');
EXEC SQL INSERT INTO M_USER VALUES ('egg', 'Abel', 'e1234', 'e_egg@naver.com',
'FEMALE');
EXEC SQL INSERT INTO M_USER VALUES ('frog', 'Emma', 'f1234', 'e_frog@naver.com',
'FEMALE');
EXEC SQL INSERT INTO M_USER VALUES ('grape', 'Ava', 'g1234', 'e_grape@naver.com',
'MALE');
EXEC SQL INSERT INTO M_USER VALUES ('horse', 'Emily', 'h1234', 'e_horse@naver.com',
'FEMALE');
EXEC SQL INSERT INTO M_USER VALUES ('ink', 'Zoe', 'i1234', 'e_ink@naver.com', 'FEMALE');
EXEC SQL INSERT INTO M_USER VALUES ('jun', 'James', 'j1234', 'e_jun@naver.com',
'FEMALE');
EXEC SQL INSERT INTO M_USER VALUES ('kiwi', 'Jack', 'k1234', 'e_kiwi@naver.com',
'MALE');
EXEC SQL INSERT INTO M_USER VALUES ('lemon', 'Henry', 'l1234', 'e_lemon@naver.com',
'MALE');

```

```

EXEC SQL INSERT INTO M_USER VALUES ('monkey', 'Ryan', 'm1234', 'e_monkey@naver.com',
'FEMALE');

// REVIEW TABLE
EXEC SQL INSERT INTO REVIEW VALUES ('RE1', 'M1', 'apple',8 , 'Very interesting');
EXEC SQL INSERT INTO REVIEW VALUES ('RE2', 'M11', 'banana',10 , 'OMG');
EXEC SQL INSERT INTO REVIEW VALUES ('RE3', 'M2', 'carrot',5 , 'So so');
EXEC SQL INSERT INTO REVIEW VALUES ('RE4', 'M10', 'dog', 7, 'I enjoyed it');
EXEC SQL INSERT INTO REVIEW VALUES ('RE5', 'M12', 'egg', 6, 'Not bad');
EXEC SQL INSERT INTO REVIEW VALUES ('RE6', 'M7', 'apple', 7, 'Son love it');
EXEC SQL INSERT INTO REVIEW VALUES ('RE7', 'M4', 'dog', 2, 'Too Bornnnning');
EXEC SQL INSERT INTO REVIEW VALUES ('RE8', 'M6', 'horse', 10, 'My Best Movie');
EXEC SQL INSERT INTO REVIEW VALUES ('RE9', 'M9', 'ink', 9, 'How Lovely');
EXEC SQL INSERT INTO REVIEW VALUES ('RE10', 'M13', 'lemon', 5, 'Story is bad');
EXEC SQL INSERT INTO REVIEW VALUES ('RE11', 'M8', 'banana', 6, 'Hummmmm');
EXEC SQL INSERT INTO REVIEW VALUES ('RE12', 'M5', 'carrot', 8, 'Good Actor');
EXEC SQL INSERT INTO REVIEW VALUES ('RE13', 'M3', 'banana',10 , 'Cheap at the price');

// RATING TABLE
EXEC SQL INSERT INTO RATING VALUES ('R1', 'M1',15 ,70 ,30);
EXEC SQL INSERT INTO RATING VALUES ('R2', 'M2',35 ,50 ,50);
EXEC SQL INSERT INTO RATING VALUES ('R3', 'M3',20 ,40 ,60);
EXEC SQL INSERT INTO RATING VALUES ('R4', 'M4',45 ,20 ,80);
EXEC SQL INSERT INTO RATING VALUES ('R5', 'M5',50 ,30 ,70);
EXEC SQL INSERT INTO RATING VALUES ('R6', 'M6',55 ,40 ,60);
EXEC SQL INSERT INTO RATING VALUES ('R7', 'M7',19 ,90 ,10);
EXEC SQL INSERT INTO RATING VALUES ('R8', 'M8',27 ,40 ,60);
EXEC SQL INSERT INTO RATING VALUES ('R9', 'M9',33 ,35 ,65);
EXEC SQL INSERT INTO RATING VALUES ('R10', 'M10',31 ,25 ,75);
EXEC SQL INSERT INTO RATING VALUES ('R11', 'M11',59 ,75 ,15);
EXEC SQL INSERT INTO RATING VALUES ('R12', 'M12',29 ,15 ,85);
EXEC SQL INSERT INTO RATING VALUES ('R13', 'M13',8 ,9 ,91);

// TIMETABLE TABLE
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME1', 'M1', '2017-05-17 09:00:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME2', 'M5', '2014-07-05 15:00:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME3', 'M7', '2017-03-10 17:30:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME4', 'M2', '2016-04-10 16:00:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME5', 'M8', '2012-05-30 19:00:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME6', 'M7', '2017-03-15 21:30:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME7', 'M4', '2011-05-11 07:30:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME8', 'M3', '2015-08-09 20:00:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME9', 'M13', '2017-12-20 07:00:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME10', 'M11', '2011-09-29 09:30:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME11', 'M12', '1989-02-04 10:30:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME12', 'M10', '2017-04-09 16:50:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME13', 'M6', '1998-11-26 14:20:00');
EXEC SQL INSERT INTO TIMETABLE VALUES ('TIME14', 'M9', '2014-09-18 20:40:00');

```

```

// COUPONINFO TABLE
EXEC SQL INSERT INTO COUPONINFO VALUES ('GC1',500 , 'GUNJA COUPON-1');
EXEC SQL INSERT INTO COUPONINFO VALUES ('GC2',1000 , 'GUNJA COUPON-2');
EXEC SQL INSERT INTO COUPONINFO VALUES ('JC1',1500 , 'JINKWAN COUPON-1');
EXEC SQL INSERT INTO COUPONINFO VALUES ('JC2', 2000, 'JINKWAN COUPON-2');
EXEC SQL INSERT INTO COUPONINFO VALUES ('CC1',2500 , 'CHONGMU COUPON-1');
EXEC SQL INSERT INTO COUPONINFO VALUES ('CC2', 3000, 'CHONGMU COUPON-2');
EXEC SQL INSERT INTO COUPONINFO VALUES ('DC1', 3500, 'DAEYANG COUPON-1');
EXEC SQL INSERT INTO COUPONINFO VALUES ('DC2', 4000, 'DAEYANG COUPON-2');
EXEC SQL INSERT INTO COUPONINFO VALUES ('SC1',4500 , 'SEJONG COUPON-1');
EXEC SQL INSERT INTO COUPONINFO VALUES ('SC2', 5000, 'SEJONG COUPON-1');
EXEC SQL INSERT INTO COUPONINFO VALUES ('YC1', 5500, 'YULKOK COUPON-1');
EXEC SQL INSERT INTO COUPONINFO VALUES ('YC2',6000 , 'YULKOK COUPON-2');
EXEC SQL INSERT INTO COUPONINFO VALUES ('YC3',8000 , 'YULKOK COUPON-3');

// COUPON TABLE
EXEC SQL INSERT INTO COUPON VALUES ('COU1', 'apple', 'GC1', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU2', 'banana', 'GC2', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU3', 'carrot', 'JC1', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU4', 'dog', 'JC2', 2);
EXEC SQL INSERT INTO COUPON VALUES ('COU5', 'egg', 'CC1', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU6', 'frog', 'CC2', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU7', 'carrot', 'DC1', 3);
EXEC SQL INSERT INTO COUPON VALUES ('COU8', 'grape', 'DC2', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU9', 'banana', 'SC1', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU10', 'kiwi', 'SC2', 3);
EXEC SQL INSERT INTO COUPON VALUES ('COU11', 'lemon', 'YC1', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU12', 'monkey', 'YC2', 1);
EXEC SQL INSERT INTO COUPON VALUES ('COU13', 'ink', 'YC3', 2);

// EVENT TABLE
EXEC SQL INSERT INTO EVENT VALUES ('E1', 'apple', 'M12', '2018-02-03', 'T');
EXEC SQL INSERT INTO EVENT VALUES ('E1', 'banana', 'M12', '2018-02-03', 'F');
EXEC SQL INSERT INTO EVENT VALUES ('E1', 'carrot', 'M12', '2018-02-03', 'F');
EXEC SQL INSERT INTO EVENT VALUES ('E1', 'dog', 'M12', '2018-02-03', 'T');
EXEC SQL INSERT INTO EVENT VALUES ('E1', 'egg', 'M12', '2018-02-03', 'T');
EXEC SQL INSERT INTO EVENT VALUES ('E1', 'frog', 'M12', '2018-02-03', 'F');
EXEC SQL INSERT INTO EVENT VALUES ('E1', 'grape', 'M12', '2018-02-03', 'F');
EXEC SQL INSERT INTO EVENT VALUES ('E2', 'horse', 'M13', '2018-02-15', 'F');
EXEC SQL INSERT INTO EVENT VALUES ('E2', 'ink', 'M13', '2018-02-15', 'F');
EXEC SQL INSERT INTO EVENT VALUES ('E2', 'jun', 'M13', '2018-02-15', 'T');
EXEC SQL INSERT INTO EVENT VALUES ('E2', 'kiwi', 'M13', '2018-02-15', 'F');
EXEC SQL INSERT INTO EVENT VALUES ('E2', 'lemon', 'M13', '2018-02-15', 'T');
EXEC SQL INSERT INTO EVENT VALUES ('E2', 'monkey', 'M13', '2018-02-15', 'T');

// COOKIE TABLE
EXEC SQL INSERT INTO COOKIE VALUES ('C1', 'Popcorn', 'Original', 6000);
EXEC SQL INSERT INTO COOKIE VALUES ('C2', 'Popcorn', 'Caramel', 7000);

```

```

EXEC SQL INSERT INTO COOKIE VALUES ('C3', 'Popcorn', 'Onion', 7000);
EXEC SQL INSERT INTO COOKIE VALUES ('C4', 'Popcorn', 'Cheeze', 7000);
EXEC SQL INSERT INTO COOKIE VALUES ('C5', 'Fried', 'FrenchFry', 5500);
EXEC SQL INSERT INTO COOKIE VALUES ('C6', 'Fried', 'Nacho', 4500);
EXEC SQL INSERT INTO COOKIE VALUES ('C7', 'Roast', 'Grilled Squid', 5000);
EXEC SQL INSERT INTO COOKIE VALUES ('C8', 'Roast', 'Sausage', 5000);
EXEC SQL INSERT INTO COOKIE VALUES ('C9', 'Beverage', 'Coke', 2500);
EXEC SQL INSERT INTO COOKIE VALUES ('C10', 'Beverage', 'Sprite', 2500);
EXEC SQL INSERT INTO COOKIE VALUES ('C11', 'Beverage', 'Orange Juice', 2000);
EXEC SQL INSERT INTO COOKIE VALUES ('C12', 'Beverage', 'Grape Juice', 2000);
EXEC SQL INSERT INTO COOKIE VALUES ('C13', 'Beverage', 'Water', 1000);

// M_ORDER TABLE
EXEC SQL INSERT INTO M_ORDER VALUES ('O1', 'C10', 'apple',2500 );
EXEC SQL INSERT INTO M_ORDER VALUES ('O2', 'C1', 'apple',6000 );
EXEC SQL INSERT INTO M_ORDER VALUES ('O3', 'C9', 'monkey',2500 );
EXEC SQL INSERT INTO M_ORDER VALUES ('O4', 'C4', 'monkey', 7000);
EXEC SQL INSERT INTO M_ORDER VALUES ('O5', 'C3', 'grape', 7000);
EXEC SQL INSERT INTO M_ORDER VALUES ('O6', 'C10', 'grape', 2500);
EXEC SQL INSERT INTO M_ORDER VALUES ('O7', 'C8', 'ink', 5000);
EXEC SQL INSERT INTO M_ORDER VALUES ('O8', 'C9', 'ink', 2500);
EXEC SQL INSERT INTO M_ORDER VALUES ('O9', 'C2', 'carrot', 7000);
EXEC SQL INSERT INTO M_ORDER VALUES ('O10', 'C11', 'carrot', 2000);
EXEC SQL INSERT INTO M_ORDER VALUES ('O11', 'C7', 'lemon',5000 );
EXEC SQL INSERT INTO M_ORDER VALUES ('O12', 'C1', 'lemon',6000 );
EXEC SQL INSERT INTO M_ORDER VALUES ('O13', 'C9', 'lemon', 2500);

// GRADE TABLE
EXEC SQL INSERT INTO GRADE VALUES ('VVIP', 'apple', 'YC3 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('VIP', 'dog', 'SC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('VIP', 'monkey', 'SC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('GOLD', 'banana', 'DC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('GOLD', 'egg', 'DC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('GOLD', 'jun', 'DC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('SILVER', 'carrot', 'CC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('SILVER', 'horse', 'CC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('SILVER', 'ink', 'CC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('BRONZE', 'frog', 'JC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('BRONZE', 'grape', 'JC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('BRONZE', 'kiwi', 'JC2 coupon');
EXEC SQL INSERT INTO GRADE VALUES ('BRONZE', 'lemon', 'JC2 coupon');

// THEATER TABLE
EXEC SQL INSERT INTO THEATER VALUES ('NORMAL_A', 'A1~J10', 100);
EXEC SQL INSERT INTO THEATER VALUES ('NORMAL_B', 'A1~J10', 100);
EXEC SQL INSERT INTO THEATER VALUES ('NORMAL_C', 'A1~J15', 150);
EXEC SQL INSERT INTO THEATER VALUES ('NORMAL_D', 'A1~E40', 200);
EXEC SQL INSERT INTO THEATER VALUES ('NORMAL_E', 'A1~J20', 200);

```

```

EXEC SQL INSERT INTO THEATER VALUES ('SWEET_A', 'A1~E50', 100);
EXEC SQL INSERT INTO THEATER VALUES ('SWEET_B', 'A1~J30', 100);
EXEC SQL INSERT INTO THEATER VALUES ('SWEET_C', 'A1~E10', 50);
EXEC SQL INSERT INTO THEATER VALUES ('COUPLE_A', 'A1~C10', 30);
EXEC SQL INSERT INTO THEATER VALUES ('COUPLE_B', 'A1~E10',50 );
EXEC SQL INSERT INTO THEATER VALUES ('COUPLE_C', 'A1~E10', 50);
EXEC SQL INSERT INTO THEATER VALUES ('ROYAL_A', 'A1~C10',30 );
EXEC SQL INSERT INTO THEATER VALUES ('ROYAL_B', 'A1~C10',30 );

// TICKET TABLE
EXEC SQL INSERT INTO TICKET VALUES ('TICKET1', 'M1', 'TIME1', 'GoG2', 'apple',
'NORMAL_A', 'A4', 'Fantasy');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET2', 'M10', 'TIME12', 'Beauty and Beast',
'dog', 'SWEET_B', 'E22', 'Fantasy');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET3', 'M2', 'TIME4', 'The Boss Baby', 'carrot',
'COUPLE_B', 'B2', 'Animation');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET4', 'M6', 'TIME13', 'Smurfs', 'horse',
'NORMAL_D', 'D34', 'Animation');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET5', 'M11', 'TIME10', 'SuSuPeyong', 'banana',
'NORMAL_C', 'A6', 'Drama');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET6', 'M3', 'TIME8', 'The Sheriff In Town',
'banana', 'NORMAL_B', 'B30', 'Drama');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET7', 'M7', 'TIME3', 'Fast and Furious',
'apple', 'SWEET_C', 'E9', 'Action');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET8', 'M12', 'TIME11', 'Kuroko Basketball',
'egg', 'NORMAL_B', 'F9', 'Animation');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET9', 'M4', 'TIME7', 'The King Case Note',
'dog', 'ROYAL_A', 'D3', 'Action');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET10', 'M13', 'TIME9', 'The Plan', 'lemon',
'SWEET_A', 'B3', 'Drama');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET11', 'M5', 'TIME2', 'The Mayor', 'carrot',
'NORMAL_E', 'B18', 'Drama');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET12', 'M9', 'TIME14', 'The Unknown Girl',
'ink', 'ROYAL_B', 'C8', 'Drama');
EXEC SQL INSERT INTO TICKET VALUES ('TICKET13', 'M8', 'TIME5', 'Dancer', 'banana',
'COUPLE_C', 'A6', 'Drama');

// BOOK TABLE
EXEC SQL INSERT INTO BOOK VALUES ('B1', 'apple', 'TICKET1', '2017-02-05
14:30:00',1 , 'CARD');
EXEC SQL INSERT INTO BOOK VALUES ('B2', 'banana', 'TICKET5', '2017-03-16 12:00:00',1 ,
'MONEY');
EXEC SQL INSERT INTO BOOK VALUES ('B3', 'carrot', 'TICKET3', '2017-05-21 08:30:00',2 ,
'MONEY');
EXEC SQL INSERT INTO BOOK VALUES ('B4', 'dog', 'TICKET2', '2017-06-29 09:30:00',1 ,
'MONEY');
EXEC SQL INSERT INTO BOOK VALUES ('B5', 'egg', 'TICKET8', '2017-07-30 19:00:00',1 ,

```

```

'CARD');
EXEC SQL INSERT INTO BOOK VALUES ('B6', 'apple', 'TICKET7', '2017-09-02 17:30:00',3 ,
'MONEY');
EXEC SQL INSERT INTO BOOK VALUES ('B7', 'dog', 'TICKET9', '2017-10-19 13:30:00',1 ,
'MONEY');
EXEC SQL INSERT INTO BOOK VALUES ('B8', 'horse', 'TICKET4', '2017-11-22 15:00:00',2 ,
'CARD');
EXEC SQL INSERT INTO BOOK VALUES ('B9', 'ink', 'TICKET12', '2017-11-01 16:30:00',1 ,
'CARD');
EXEC SQL INSERT INTO BOOK VALUES ('B10', 'lemon', 'TICKET10', '2017-12-07 08:00:00',1
, 'MONEY');
EXEC SQL INSERT INTO BOOK VALUES ('B11', 'banana', 'TICKET6', '2017-12-19 06:30:00',3
, 'MONEY');
EXEC SQL INSERT INTO BOOK VALUES ('B12', 'carrot', 'TICKET11', '2017-12-28 19:00:00',2
, 'CARD');
EXEC SQL INSERT INTO BOOK VALUES ('B13', 'banana', 'TICKET13', '2018-01-06
13:30:00',2 , 'MONEY');
// SALEINFO TABLE
EXEC SQL INSERT INTO SALEINFO VALUES ('S1', 'apple', 'YC3', 'CARD',1000 ,9000 );
EXEC SQL INSERT INTO SALEINFO VALUES ('S2', 'banana', 'DC2', 'NO',0 ,4000 );
EXEC SQL INSERT INTO SALEINFO VALUES ('S3', 'carrot', 'CC2', 'NO', 0, 3000);
EXEC SQL INSERT INTO SALEINFO VALUES ('S4', 'dog', 'SC2', 'MORNING', 3000, 8000);
EXEC SQL INSERT INTO SALEINFO VALUES ('S5', 'egg', 'DC2', 'NO',1000 ,5000 );
EXEC SQL INSERT INTO SALEINFO VALUES ('S6', 'apple', 'GC1', 'MOBILE',2000 ,2500 );
EXEC SQL INSERT INTO SALEINFO VALUES ('S7', 'dog', 'DC1', 'MOBILE', 2000, 5500);
EXEC SQL INSERT INTO SALEINFO VALUES ('S8', 'horse', 'CC2', 'CARD', 1000, 4000);
EXEC SQL INSERT INTO SALEINFO VALUES ('S9', 'ink', 'CC2', 'CARD',1000 , 4000);
EXEC SQL INSERT INTO SALEINFO VALUES ('S10', 'lemon', 'JC2', 'NO',0 , 2000);
EXEC SQL INSERT INTO SALEINFO VALUES ('S11', 'banana', 'YC1', 'MORNING', 3000, 8500);
EXEC SQL INSERT INTO SALEINFO VALUES ('S12', 'carrot', 'SC1', 'CARD',1000 , 5500);
EXEC SQL INSERT INTO SALEINFO VALUES ('S13', 'banana', 'JC2', 'NO', 2000, 4000);
// FINDER TABLE
EXEC SQL INSERT INTO FINDER VALUES ('FD1', 'apple', 1, 0, 0, 1,'Action');
EXEC SQL INSERT INTO FINDER VALUES ('FD2', 'banana', 0, 0, 3, 0,'Drama');
EXEC SQL INSERT INTO FINDER VALUES ('FD3', 'carrot',0 , 1, 1, 0,'Drama');
EXEC SQL INSERT INTO FINDER VALUES ('FD4', 'dog', 1, 0, 0, 1,'Action');
EXEC SQL INSERT INTO FINDER VALUES ('FD5', 'egg', 0, 1, 0, 0,'Animation');
EXEC SQL INSERT INTO FINDER VALUES ('FD6', 'frog', 1, 0, 0, 1,'Action');
EXEC SQL INSERT INTO FINDER VALUES ('FD7', 'grape', 1, 0, 0, 1,'Action');
EXEC SQL INSERT INTO FINDER VALUES ('FD8', 'horse', 0, 1, 0, 0,'Animation');
EXEC SQL INSERT INTO FINDER VALUES ('FD9', 'ink', 0, 0, 1, 0,'Drama');
EXEC SQL INSERT INTO FINDER VALUES ('FD10', 'jun', 0, 0, 0, 0,'All');
EXEC SQL INSERT INTO FINDER VALUES ('FD11', 'kiwi', 0, 0, 0, 0,'All');
EXEC SQL INSERT INTO FINDER VALUES ('FD12', 'lemon', 0, 0, 1, 0,'Drama');
EXEC SQL INSERT INTO FINDER VALUES ('FD13', 'monkey', 0, 0, 0, 0,'All');

```

```
}
```

```
void variable_initialize()
{
    //변수 초기화
    int_a = 0;
    int_b = 0;
    int_c = 0;
    int_d = 0;
    memset(var20_input.arr, 0x00, sizeof(var20_input));
    memset(var20_a.arr, 0x00, sizeof(var20_a));
    memset(var20_b.arr, 0x00, sizeof(var20_b));
    memset(var20_c.arr, 0x00, sizeof(var20_c));
    memset(var20_d.arr, 0x00, sizeof(var20_d));
    memset(var20_e.arr, 0x00, sizeof(var20_e));
    memset(var20_f.arr, 0x00, sizeof(var20_f));
    memset(var20_g.arr, 0x00, sizeof(var20_g));
    memset(var20_h.arr, 0x00, sizeof(var20_h));

    memset(var1_a.arr, 0x00, sizeof(var1_a));
    memset(var3_a.arr, 0x00, sizeof(var3_a));
    memset(var1_input.arr, 0x00, sizeof(var1_input));
}
```

```
void lvariable_initialize()
{
    memset(lvar20_a.arr, 0x00, sizeof(lvar20_a));
    memset(lvar20_b.arr, 0x00, sizeof(lvar20_b));
    memset(lvar20_c.arr, 0x00, sizeof(lvar20_c));
    memset(lvar20_d.arr, 0x00, sizeof(lvar20_d));
    memset(lvar20_e.arr, 0x00, sizeof(lvar20_e));
    memset(lvar20_f.arr, 0x00, sizeof(lvar20_f));
    memset(lvar20_g.arr, 0x00, sizeof(lvar20_g));
    memset(lvar20_h.arr, 0x00, sizeof(lvar20_h));
}
```

```
void admin_show_movie() // Show movie information
{
    variable_initialize();
    // 영화이름 입력받아서 해당 영화 정보 보여주기
    printf("*****ADMIN mode - Show menu*****\n");
    printf("Input Movie name(mo_name) >>> "); scanf("%s", var20_input.arr); var20_input.len =
strlen(var20_input.arr);
    //Cursor for MOVIE table
```



```

EXEC SQL DECLARE movie_cursor1 CURSOR FOR
    SELECT mo_no, mo_name, mo_director, mo_actor, mo_genre, mo_Rdate
    FROM MOVIE
    WHERE MO_NAME = :var20_input;

EXEC SQL OPEN movie_cursor1;
printf("*MOVIE TABLE Wn(mo_no | mo_name | mo_director | mo_actor | mo_mo_genre |
mo_Rdate)WnWn");
while (1)
{
    EXEC SQL FETCH movie_cursor1 INTO : var20_a, : var20_b, : var20_c, : var20_d, :
var20_e, : var20_f;
    if (sqlca.sqlcode == 1403)
    {
        // printf("MOVIE table - not yet dataWn");

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %s | %s | %sWn", var20_a.arr, var20_b.arr, var20_c.arr,
var20_d.arr, var20_e.arr, var20_f.arr);
    variable_initialize();
}
EXEC SQL CLOSE movie_cursor1;
}

void admin_show_rating()
{
    variable_initialize();
    // 영화이름 입력받아서 해당 영화 정보 보여주기
    printf("*****ADMIN mode - Show menu*****Wn");
    printf("Input Movie name(mo_name) >>> "); scanf("%s", var20_input.arr); var20_input.len =
strlen(var20_input.arr);

    //Cursor for RATING table (내포질의)
    EXEC SQL DECLARE rating_cursor1 CURSOR FOR
        SELECT ra_no, ra_movieNo, ra_totalR, ra_maleR, ra_femaleR
        FROM RATING

```

```

        WHERE RA_MOVIENO = (SELECT MO_NO FROM MOVIE WHERE MO_NAME =
:var20_input);

EXEC SQL OPEN rating_cursor1;
printf("*RATING TABLE Wn(ra_no | ra_movieNo | ra_totalR | ra_maleR | ra_femaleR)WnWn");
while (1)
{
    EXEC SQL FETCH rating_cursor1 INTO : var20_a, : var20_b, : int_a, : int_b, : int_c;
    if (sqlca.sqlcode == 1403)
    {
        //printf("RATING table - not yet dataWnWn");

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %d %% | %d %% | %d %% Wn", var20_a.arr, var20_b.arr, int_a, int_b,
int_c);
    variable_initialize();
}
EXEC SQL CLOSE rating_cursor1;
}

void admin_show_grade() // Show user grade
{
    variable_initialize();
    // 등급 입력받아서 해당 영화 정보 보여주기
    printf("*****ADMIN mode - Show menu*****Wn");
    printf("---info : This show you User IDs which are selected grade.---Wn");
    printf("Input Grade (VVIP | VIP | GOLD | SILVER | BRONZE) >>> "); scanf("%s",
var20_input.arr); var20_input.len = strlen(var20_input.arr);

    //Cursor for GRADE table
    EXEC SQL DECLARE grade_cursor1 CURSOR FOR
    SELECT gr_grade, gr_userId
    FROM GRADE
    WHERE gr_grade = :var20_input;

    EXEC SQL OPEN grade_cursor1;
    printf("*GRADE TABLE Wn(gr_grade | gr_userId )WnWn");

```

```

while (1)
{
    EXEC SQL FETCH grade_cursor1 INTO : var20_a, : var20_b;
    if (sqlca.sqlcode == 1403)
    {
        //printf("GRADE table - not yet dataWn");

printf("-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s Wn", var20_a.arr, var20_b.arr);
    variable_initialize();
}
EXEC SQL CLOSE grade_cursor1;
}

void admin_show_sale() // Show sale price
{
    // 회원ID로 그룹바이해서 특정금액을 입력받아 그 이상보다 할인받은 회원ID와 총 할인금액 출력
    variable_initialize();
    // 등급 입력받아서 해당 영화 정보 보여주기
    printf("*****ADMIN mode - Show menu*****Wn");
    printf("---info : This show you User IDs and Total saled price which is over the price you
wrote down.---Wn");
    printf("Input Price >>> "); scanf("%d", &int_d);

    //Cursor for SALEINFO table
    EXEC SQL DECLARE saleIF_cursor1 CURSOR FOR
        SELECT sa_userId, sum(sa_totalSMoney) FROM SALEINFO
        GROUP BY SA_USERID HAVING SUM(SA_TOTALSMONEY) > :int_d;

    EXEC SQL OPEN saleIF_cursor1;
    printf("*SALEINFO TABLE Wn( sa_userId | sa_totalSMoney)WnWn");
    while (1)
    {
        EXEC SQL FETCH saleIF_cursor1 INTO : var20_a, : int_a;
        if (sqlca.sqlcode == 1403)
        {
            //printf("SALEINFO table - not yet dataWn");

```

```

printf("-----
-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf("Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %dWn", var20_a.arr, int_a);
    variable_initialize();
}
EXEC SQL CLOSE saleIF_cursor1;
}

void admin_show_review() // Show review and stars about movie
{
    variable_initialize();
    // 영화이름 검색 시 영화관리번호를 알아낸다음 해당 영화의 평점과 후기를 출력(내포질의)
    printf("*****ADMIN mode - Show menu*****Wn");
    printf("---info : This show you Movie name, reviews and stars which is choosen by
you.---Wn");
    printf("Input Movie name    >>> "); scanf("%s", var20_input.arr); var20_input.len =
strlen(var20_input.arr);

    //Cursor for REVIEW table
    EXEC SQL DECLARE review_cursor1 CURSOR FOR
        SELECT re_movieNo, mo_name, re_stars, re_review
        FROM REVIEW, MOVIE
        WHERE REVIEW.RE_MOVIENO = MOVIE.MO_NO AND re_movieNo = (SELECT MO_NO
FROM MOVIE WHERE MO_NAME = :var20_input);

    EXEC SQL OPEN review_cursor1;
    printf("*REVIEW TABLE Wn(re_movieNo | mo_name | re_stars | re_review)WnWn");
    while (1)
    {
        EXEC SQL FETCH review_cursor1 INTO : var20_a, : var20_b, : int_a, : var20_c;
        if (sqlca.sqlcode == 1403)
        {
            // printf("REVIEW table - not yet dataWn");

printf("-----
-----Wn");

            break;

```

```

    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %d | %sWn", var20_a.arr, var20_b.arr, int_a, var20_c.arr);
    variable_initialize();
}
EXEC SQL CLOSE review_cursor1;

}

void admin_show_price() // Show price
{
    variable_initialize();
    // 특정금액을 입력받고, 입력받은 금액보다 이상의 주문을 한 회원id와 총 결제금액 출력
    printf("*****ADMIN mode - Show menu*****Wn");
    printf("----info : This show you User IDs and total order-price that is over than the price you
wrote down.---Wn");
    printf("Input Price >>> "); scanf("%d", &int_d);

    //Cursor for M_ORDER table
    EXEC SQL DECLARE morder_cursor1 CURSOR FOR
    SELECT or_userId, SUM(or_price)
    FROM M_ORDER
    GROUP BY or_userId HAVING SUM(or_price) > :int_d;

    EXEC SQL OPEN morder_cursor1;
    printf("*M_ORDER TABLE Wn( or_userId | total ordered price)WnWn");
    while (1)
    {
        EXEC SQL FETCH morder_cursor1 INTO : var20_a, : int_a;
        if (sqlca.sqlcode == 1403)
        {
            //printf("M_ORDER table - not yet dataWn");

printf("-----
-----Wn");

            break;
        }
        else if (sqlca.sqlcode != 0)
        {
            printf("Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);

```



```

        printf("%s | %s | %s | %d\n", var20_a.arr, var20_b.arr, var20_c.arr, int_a);
        variable_initialize();
    }

//4. CURSOR - CLOSE
EXEC SQL CLOSE cookie_cursor1;
}

void admin_show_event() // Show event
{
    int num = -1;
    variable_initialize();
    // 이벤트에 당첨된 회원id, 영화정보(영화관리번호, 영화이름, 영화상영일자) 보여주기
    printf("*****ADMIN mode - Show menu*****\n");
    printf("---info : This show you User id, Movie no, Movie name, date.---\n");
    printf("T -> Show Win a event users | F -> Show Fail a event users\n");
    printf("Input T or F >>> "); scanf("%s", var1_input.arr); var1_input.len = strlen(var1_input.arr);

    printf("input value -> %s", var1_input.arr);
    //Cursor for EVENT table
    EXEC SQL DECLARE event_cursor1 CURSOR FOR
    SELECT ev_userid, ev_movieNO, mo_name ,ev_Mdate
    FROM EVENT, MOVIE
    WHERE EVENT.EV_MOVIENO = MOVIE.MO_NO AND EV_CHECK = :var1_input;

    EXEC SQL OPEN event_cursor1;
    printf("*EVENT TABLE \n( ev_userid | ev_movieNO | movie name | ev_Mdate )\n\n");
    while (1)
    {
        EXEC SQL FETCH event_cursor1 INTO : var20_a, : var20_b, : var20_c, : var20_d;
        if (sqlca.sqlcode == 1403)
        {
            //printf("EVENT table - not yet data\n");

            printf("-----\n\n");
            break;
        }
        else if (sqlca.sqlcode != 0)
        {
            printf(" Oracle Error Detected!!: %d\n%s\n", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
            break;
        }
        printf("%s | %s | %s | %s \n", var20_a.arr, var20_b.arr, var20_c.arr, var20_d.arr);
    }
}

```

```

        variable_initialize();
    }
    EXEC SQL CLOSE event_cursor1;
}

void admin_show_coupon() //Show coupon
{
    variable_initialize();
    // 쿠폰 종류 입력시 해당 쿠폰을 가지고 있는 회원id와 쿠폰, 회원이름을 출력
    printf("*****ADMIN mode - Show menu*****Wn");
    printf("---info : This show you User id, Coupon type When you input coupon type.---Wn");
    printf("#Coupon type -> GC1 | GC2 | JC1 | JC2 | CC1 | CC2 | DC1 | DC2 | SC1 | SC2
| YC1 | YC2 | YC3Wn");
    printf("Input Coupon type >>> "); scanf("%s", var20_input.arr); var20_input.len =
strlen(var20_input.arr);

    //Cursor for COUPON table
    EXEC SQL DECLARE coupon_cursor1 CURSOR FOR
    SELECT cou_userId, cou_type, cou_num
    FROM COUPON
    WHERE cou_type = :var20_input;

    EXEC SQL OPEN coupon_cursor1;
    printf("*COUPON TABLE Wn( cou_userId | cou_type | cou_num)WnWn");
    while (1)
    {
        EXEC SQL FETCH coupon_cursor1 INTO : var20_a, : var20_b, : int_a;
        if (sqlca.sqlcode == 1403)
        {
            // printf("COUPON table - not yet dataWn");

printf("-----
-----Wn");

            break;
        }
        else if (sqlca.sqlcode != 0)
        {
            printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
            break;
        }
        printf("%s | %s | %dWn", var20_a.arr, var20_b.arr, int_a);
        variable_initialize();
    }
    EXEC SQL CLOSE coupon_cursor1;
}

```



```

{
    variable_initialize();
    // User id를 입력하면 그 사람이 본 영화의 장르를 카운트한 정보를 볼 수 있다. (FINDER, 그룹바
01 userid)
    printf("*****USER mode - Show menu*****\n");
    printf("---info : This show you User id, Movie genre When you enter the User id.---\n");
    printf("Input User id   ex) apple   >>> "); scanf("%s", var20_input.arr); var20_input.len =
strlen(var20_input.arr);

    //Cursor for FINDER table
    EXEC SQL DECLARE finder_cursor2 CURSOR FOR
        SELECT fi_userid, SUM(fi_SF), SUM(fi_ro), SUM(fi_dr), SUM(fi_ho)
        FROM FINDER
        WHERE FI_USERID = :var20_input
        GROUP BY FI_USERID HAVING COUNT(FI_USERID) >= 1;

    EXEC SQL OPEN finder_cursor2;
    printf("*FINDER TABLE \n( UserId | SF | ROMANTIC | DRAMA | HORROR )\n\n");
    while (1)
    {
        EXEC SQL FETCH finder_cursor2 INTO : var20_a, : int_a, : int_b, : int_c, : int_d;
        if (sqlca.sqlcode == 1403)
        {
            // printf("FINDER table - not yet data\n");

printf("-----\n\n");
            break;
        }
        else if (sqlca.sqlcode != 0)
        {
            printf(" Oracle Error Detected!!: %d\n%s\n", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
            break;
        }
        printf("%s | %d | %d | %d | %d \n", var20_a.arr, int_a, int_b, int_c, int_d);
        variable_initialize();
    }
    EXEC SQL CLOSE finder_cursor2;
}

void user_show_grade() // Show grade info
{
    variable_initialize();
    // User id를 입력하면 그 사람의 등급과 해당 설명(ex)을 볼 수 있다. (GRADE)
    printf("*****USER mode - Show menu*****\n");

```

```

printf("----info : This show you User id, Grade, Explain When you enter the User id.---\n");
printf("Input User id   ex)apple   >>> "); scanf("%s", var20_input.arr); var20_input.len =
strlen(var20_input.arr);

```

```

//Cursor for GRADE table
EXEC SQL DECLARE grade_cursor2 CURSOR FOR
    SELECT gr_userid, gr_grade, gr_grEx
    FROM GRADE
    WHERE GR_USERID = :var20_input;

```

```

EXEC SQL OPEN grade_cursor2;
printf("*GRADE TABLE \n(gr_userid | gr_grade | gr_grEx)\n\n");
while (1)
{
    EXEC SQL FETCH grade_cursor2 INTO : var20_a, : var20_b, : var20_c;
    if (sqlca.sqlcode == 1403)
    {
        //printf("GRADE table - not yet data\n");

```

```

printf("-----\n\n");

```

```

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf(" Oracle Error Detected!!: %d\n%s\n", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s\n", var20_a.arr, var20_b.arr, var20_c.arr);
    variable_initialize();
}
EXEC SQL CLOSE grade_cursor2;

```

```

}
void user_show_order() // Show order info
{

```

```

    variable_initialize();
    // User id를 입력하면 그 사람의 간식 주문 내역(간식종류, 간식이름, 결제금액)을 볼 수 있다.
    (M_ORDER와 COOKIE 조인)
    printf("*****USER mode - Show menu*****\n");
    printf("----info : This show you User id, Cookie type, Cookie name, total price When you
    enter the User id.---\n");
    printf("Input User id   ex)apple   >>> "); scanf("%s", var20_input.arr); var20_input.len =
    strlen(var20_input.arr);

```

```

//Cursor for M_ORDER table

```

```

EXEC SQL DECLARE morder_cursor2 CURSOR FOR
    SELECT or_userid, coo_type, coo_name, or_price
    FROM M_ORDER, COOKIE
    WHERE coo_no = or_cooNo AND or_userid = :var20_input;

EXEC SQL OPEN morder_cursor2;
printf("*M_ORDER TABLE Wn( UserId | Cookie type | Cookie name | Order price)WnWn");
while (1)
{
    EXEC SQL FETCH morder_cursor2 INTO : var20_a, : var20_b, : var20_c, : int_a;
    if (sqlca.sqlcode == 1403)
    {
        //printf("M_ORDER table - not yet dataWn");

printf("-----Wn");

        break;
    }
    else if (sqlca.sqlcode != 0)
    {
        printf("Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
        break;
    }
    printf("%s | %s | %s | %dWn", var20_a.arr, var20_b.arr, var20_c.arr, int_a);
    variable_initialize();
}
EXEC SQL CLOSE morder_cursor2;
}

void user_show_event() // Show event info
{
    variable_initialize();
    // 이메일을 입력하면 이벤트 당첨여부와 시사회 영화의 이름, 시사회 일시를 알 수 있다. (USER와
    EVENT 내포질의)
    printf("*****USER mode - Show menu*****Wn");
    printf("---info : This show you User id, User email, Movie name, Date, Result When you
    enter the E-mail Address.---Wn");
    printf("Input User email    ex)e_apple@naver.com    >>> "); scanf("%s", var20_input.arr);
    var20_input.len = strlen(var20_input.arr);

    //Cursor for EVENT table
    EXEC SQL DECLARE event_cursor2 CURSOR FOR
        SELECT ev_userid, us_email, mo_name, mo_director, ev_Mdate, ev_check
        FROM EVENT, MOVIE, M_USER
        WHERE ev_movieNO = MO_NO AND US_ID = EV_USERID AND EV_USERID = (
            SELECT US_ID FROM M_USER

```

```

        WHERE US_EMAIL = :var20_input
    );

    EXEC SQL OPEN event_cursor2;
    printf("*EVENT TABLE Wn( UserId | User email | Movie name | Movie director | Date |
Result )WnWn");
    while (1)
    {
        EXEC SQL FETCH event_cursor2 INTO : var20_a, : var20_b, : var20_c, : var20_d, :
var20_e, : var1_a;
        if (sqlca.sqlcode == 1403)
        {
            //printf("EVENT table - not yet dataWn");

printf("-----
-----Wn");

            break;
        }
        else if (sqlca.sqlcode != 0)
        {
            printf(" Oracle Error Detected!!: %dWn%sWn", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
            break;
        }
        printf("%s | %s | %s | %s | %s | %sWn", var20_a.arr, var20_b.arr, var20_c.arr,
var20_d.arr, var20_e.arr, var1_a.arr);
        variable_initialize();
    }
    EXEC SQL CLOSE event_cursor2;
}

```