5.2 - ONLINE EXAMINATION

CONSOLE APPLICATION

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sqlda.h>
//EXEC SQL include sqla.h;
sqlda_t *sqlda1;
sqlda_t *sqlda2;
EXEC SQL BEGIN DECLARE SECTION;
char search[20000],query[20000],x[20000],username[200],name[100];
char c,*ans;
const char *password;
int i=0,intval;
long long int longlongval;
EXEC SQL END DECLARE SECTION;
EXEC SQL WHENEVER SQLERROR CALL print_sqlca();

void print_sqlca()
{
   fprintf(stderr, "==== sqlca ====\n");
   fprintf(stderr, "sqlcode: %ld\n", sqlca.sqlcode);
   fprintf(stderr, "sqlerrm.sqlerrml: %d\n", sqlca.sqlerrm.sqlerrml);
   fprintf(stderr, "sqlerrm.sqlerrmc: %s\n", sqlca.sqlerrm.sqlerrmc);
   fprintf(stderr, "sqlerrd: %ld %ld %ld %ld %ld %ld\n",
sqlca.sqlerrd[0],sqlca.sqlerrd[1],sqlca.sqlerrd[2],
                                          sqlca.sqlerrd[3],sqlca.sqlerrd[4],sqlca.sqlerrd[5]);
   fprintf(stderr, "sqlwarn: %d %d %d %d %d %d %d %d\n", sqlca.sqlwarn[0],
sqlca.sqlwarn[1], sqlca.sqlwarn[2],
                                          sqlca.sqlwarn[3], sqlca.sqlwarn[4], sqlca.sqlwarn[5],
                                          sqlca.sqlwarn[6], sqlca.sqlwarn[7]);
   fprintf(stderr, "sqlstate: %5s\n", sqlca.sqlstate);
   fprintf(stderr, "===============\n");
}
void ExecuteQuerry()
{
         printf("Input Your Query :\n");
```

```
    scanf(" %[^\n]",query);
    EXEC SQL PREPARE eq FROM :query;
    EXEC SQL DECLARE cur1 CURSOR FOR eq;
    EXEC SQL OPEN cur1;
    while(sqlca.sqlcode==0)
    {
        EXEC SQL FETCH NEXT FROM cur1 INTO DESCRIPTOR sqlda1;
        sqlda_t *cur_sqlda;
        for (cur_sqlda = sqlda1;cur_sqlda != NULL;cur_sqlda = cur_sqlda->desc_next)
        {
            int i;
      char name_buf[1024];
      char var_buf[1024];

        if(sqlca.sqlcode != 0)
            break;
        for (i = 0; i < cur_sqlda->sqld; i++)
            {
            sqlvar_t v = cur_sqlda->sqlvar[i];
            char *sqldata = v.sqldata;
            short sqllen  = v.sqllen;
            strncpy(name_buf, v.sqlname.data, v.sqlname.length);
        name_buf[v.sqlname.length] = '\0';

         switch (v.sqltype)
         {
          case ECPGt_char:
           memset(&var_buf, 0, sizeof(var_buf));
           memcpy(&var_buf, sqldata, (sizeof(var_buf)<=sqllen ? sizeof(var_buf)-1 :
sqllen) );
           break;

          case ECPGt_int: /* integer */
           memcpy(&intval, sqldata, sqllen);
           snprintf(var_buf, sizeof(var_buf), "%d", intval);
           break;

          case ECPGt_long_long: /* bigint */
           memcpy(&longlongval, sqldata, sqllen);
           snprintf(var_buf, sizeof(var_buf), "%lld", longlongval);
           break;

          default:
```

```
                                {
                                    int i;
                                    memset(var_buf, 0, sizeof(var_buf));
                                    for (i = 0; i < sqllen; i++)
                                    {
                                        char tmpbuf[16];
                                        snprintf(tmpbuf, sizeof(tmpbuf), "%02x ", (unsigned char) sqldata[i]);
                                        strncat(var_buf, tmpbuf, sizeof(var_buf));
                                    }
                                }
                                    break;
                            }

                            printf("%s = %s\n", name_buf, var_buf);
                    }

                        printf("\n");
                }

            }
            EXEC SQL CLOSE cur1;
            EXEC SQL COMMIT;
}

void UpdateQuerry()
{
            printf("Input Your Query : \n");
            scanf(" %[^\n]",query);
            //printf("%s\n",query);
            EXEC SQL PREPARE uq FROM :query;
            EXEC SQL EXECUTE uq;
            if(sqlca.sqlcode==0)
            {
                        fprintf(stderr, "Successful\n");
            }
            EXEC SQL COMMIT;
}
int main()
{
            printf("USERNAME : ");
            scanf(" %[^\n]",username);
            getchar();
            password=getpass("PASSWORD : ");
```

```
        strcat(name,username);
        strcat(name,"@10.100.71.21");
   EXEC SQL CONNECT TO :name USER :username USING :password;
        if(sqlca.sqlcode==0)
                fprintf(stderr, "Connection Successful\n");
        else
                exit(1);
        printf("Set your search path :\n");
        scanf(" %[^\n]",search);
        EXEC SQL PREPARE search_path FROM :search;
        EXEC SQL EXECUTE search_path;
        if(sqlca.sqlcode==0)
                fprintf(stderr,"Search_path Successfully set\n");
        else
        {
                fprintf(stderr,"No such path available\n");
                exit(1);
        }
        while(1)
        {
                printf("Press 1 for INSERT,UPDATE and DELETE Querry.\nPress 2 for SELECT
querry.\nPress anything else to exit.\n");
                scanf(" %c",&c);
                if(c=='1')
                {
                        UpdateQuerry();
                }
                else if(c=='2')
                {
                        ExecuteQuerry();
                }
                else
                        break;
        }
   EXEC SQL DISCONNECT ALL;
   return 0;
}
```