**《数据库原理和应用》**

**课程设计报告**

**Course Project of Database**

**Principles and Applications**

**学 号：191564103**

**中文姓名：纳坦**

**英文姓名：NATAN**

**南京航空航天大学**

**国际教育学院**

**Experiment 1:** SQL data definition and data insertion

**1 .**

**CREATE TABLE S**

SQL>Create table s(sno number(10) not null,sname varchar2(10) not null,sgender varchar2(1),sage

number(2),sdept varchar2(10), CONSTRAINT S\_SNO\_PK PRIMARY KEY(sno), CONSTRAINT s\_sno\_fk FOREIGN KEY (sno) REFERENCES sc(sno));

Table created.

SQL> desc s;

Name Null? Type

----------------------------------------- -------- ----------------------------

SNO NOT NULL NUMBER(10)

SNAME NOT NULL VARCHAR2(10)

SGENDER VARCHAR2(1)

SAGE NUMBER(2)

SDEPT VARCHAR2(10)

SQL> commit;

Commit complete.

**CREATE TABLE SC**

SQL> Create table sc(sno number(10 )not null, cno number(10) not null,grade number(3));

Table created.

SQL> desc sc;

Name Null? Type

----------------------------------------- -------- ----------------------------

SNO NOT NULL NUMBER(10)

CNO NOT NULL NUMBER(10)

GRADE NUMBER(3)

SQL> commit;

Commit complete.

**CREATE TABLE C**

SQL> Create table c(cno number(10) not null, cname varchar2(10) not null ,cpno varchar2(10),credit varchar2(10), CONSTRAINT C\_CNO\_PK PRIMARY KEY(cno));

Table created.

SQL> desc c;

Name Null? Type

----------------------------------------- -------- ----------------------------

CNO NOT NULL NUMBER(10)

CNAME NOT NULL VARCHAR2(10)

CPNO VARCHAR2(10)

CREDIT VARCHAR2(10)

SQL> commit ;

Commit complete.

**2**.

**ALTER TABLE SC**

SQL> alter table sc

modify (CONSTRAINT sc\_cno\_fk FOREIGN KEY(cno) REFERENCES c(cno));

Table altered.

SQL> commit ;

Commit complete.

**DROPPING TABLE S**

SQL> drop table s;

Table dropped.

SQL> desc s;

ERROR:

ORA-04043: object s does not exist

**INSERTING date of registration in s**

SQL> alter table s add (dateofreg DATE DEFAULT SYSDATE);

Table altered.

SQL> commit;

Commit complete.

SQL> desc s;

Name Null? Type

----------------------------------------- -------- ----------------------------

SNO NOT NULL NUMBER(10)

SNAME NOT NULL VARCHAR2(10)

SGENDER VARCHAR2(1)

SAGE NUMBER(2)

SDEPT VARCHAR2(10)

DATEOFREG DATE

**DROOPING DATEOFREGFROM table s**

SQL> alter table s

2 drop(dateofreg);

Table altered.

SQL> commit;

Commit complete.

SQL> desc s;

Name Null? Type

----------------------------------------- -------- ----------------------------

SNO NOT NULL NUMBER(10)

SNAME NOT NULL VARCHAR2(10)

SGENDER VARCHAR2(1)

SAGE NUMBER(2)

SDEPT VARCHAR2(10)

**CREATING INDEX**

SQL> create index index\_s on s(sno,sdept);

Index created.

**DROPPING INDEX**

SQL> drop index index\_s;

Index dropped.

**INSERTTING INTO TABLE c**

**SQL> insert into c values('4','data str','','4');**

**1 row created.**

SQL> Insert into c

Values('&cno','&cname','&cpno','&credit');

Enter value for cno: 1

Enter value for cname: data base

Enter value for cpno:

Enter value for credit: 4

old 2: Values('&cno','&cname','&cpno','&credit')

new 2: Values('1','data base','','4')

SQL> Insert into c

Values('&cno','&cname','&cpno','&credit');

Enter value for cno: 2

Enter value for cname: networking

Enter value for cpno:

Enter value for credit: 3.5

old 2: Values('&cno','&cname','&cpno','&credit')

new 2: Values('2','networking','','3.5')

1 row created.

SQL> Insert into c

Values('&cno','&cname','&cpno','&credit');

Enter value for cno: 3

Enter value for cname: AI

Enter value for cpno:

Enter value for credit: 3

old 2: Values('&cno','&cname','&cpno','&credit')

new 2: Values('3','AI','','3')

1 row created.

SQL> commit;

Commit complete.

**INSERTTING INTO TABLE SC**

SQL> Insert into sc

Values('&sno','&cno','&grade');

Enter value for sno: 191564101

Enter value for cno: 1

Enter value for grade: 90

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564101','1','90')

1 row created.

SQL> Insert into sc

2 Values('&sno','&cno','&grade');

Enter value for sno: 191564102

Enter value for cno: 1

Enter value for grade: 85

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564102','1','85')

1 row created.

SQL> Insert into sc

Values('&sno','&cno','&grade');

Enter value for sno: 191564101

Enter value for cno: 2

Enter value for grade: 96

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564101','2','96')

1 row created.

SQL> Insert into sc

Values('&sno','&cno','&grade');

Enter value for sno: 191564102

Enter value for cno: 2

Enter value for grade: 89

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564102','2','89')

1 row created.

SQL> commit;

Commit complete.

SQL> Insert into sc

2 Values('&sno','&cno','&grade');

Enter value for sno: 191564105

Enter value for cno: 4

Enter value for grade: 67

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564105','4','67')

1 row created.

SQL> Insert into sc

2 Values('&sno','&cno','&grade');

Enter value for sno: 191564103

Enter value for cno: 3

Enter value for grade: 76

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564103','3','76')

1 row created.

SQL> Insert into sc

2 Values('&sno','&cno','&grade');

Enter value for sno: 191564103

Enter value for cno: 2

Enter value for grade: 98

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564103','2','98')

1 row created.

SQL> insert into sc

2 values('191564101','4','78');

1 row created.

SQL> insert into sc

2 values('191564102','3','88');

1 row created.

SQL>Insert into sc

2 Values('&sno','&cno','&grade');

Enter value for sno: 191564101

Enter value for cno: 3

Enter value for grade: 87

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564101','3','87')

SQL> Insert into sc

2 Values('&sno','&cno','&grade');

Enter value for sno: 191564104

Enter value for cno: 1

Enter value for grade: 55

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564104','1','55')

1 row created.

SQL> Insert into sc

2 Values('&sno','&cno','&grade');

Enter value for sno: 191564104

Enter value for cno: 2

Enter value for grade: 89

old 2: Values('&sno','&cno','&grade')

new 2: Values('191564104','2','89')

1 row created.

INSERTING INTO TABLE s

SQL> Insert into s

Values('&sno','&sname','&sgender', '&sage','&sdept');

Enter value for sno: 191564105

Enter value for sname: abel

Enter value for sgender: m

Enter value for sage: 19

Enter value for sdept: language

old 2: Values('&sno','&sname','&sgender', '&sage','&sdept')

new 2: Values('191564105','abel','m', '19','language')

1 row created.

SQL> Insert into s

Values('&sno','&sname','&sgender', '&sage','&sdept');

Enter value for sno: 191564101

Enter value for sname: nicho

Enter value for sgender: M

Enter value for sage: 19

Enter value for sdept: computer

old 2: Values('&sno','&sname','&sgender', '&sage','&sdept')

new 2: Values('191564101','nicho','M', '19','computer')

1 row created.

SQL> Insert into s

2 Values('&sno','&sname','&sgender', '&sage','&sdept');

Enter value for sno: 191564102

Enter value for sname: natii

Enter value for sgender: M

Enter value for sage: 20

Enter value for sdept: computer

old 2: Values('&sno','&sname','&sgender', '&sage','&sdept')

new 2: Values('191564102','natii','M', '20','computer')

1 row created.

SQL> commit

2 ;

Commit complete.

SQL> Insert into s

2 Values('&sno','&sname','&sgender', '&sage','&sdept');

Enter value for sno: 191564103

Enter value for sname: henry

Enter value for sgender: f

Enter value for sage: 20

Enter value for sdept: maths

old 2: Values('&sno','&sname','&sgender', '&sage','&sdept')

new 2: Values('191564103','henry','f', '20','maths')

1 row created.

SQL> Insert into s

2 Values('&sno','&sname','&sgender', '&sage','&sdept');

Enter value for sno: 191564104

Enter value for sname: tsegaye

Enter value for sgender: m

Enter value for sage: 21

Enter value for sdept: maths

old 2: Values('&sno','&sname','&sgender', '&sage','&sdept')

new 2: Values('191564104','tsegaye','m', '21','maths')

1 row created.

SQL> commit;

Commit complete.

**Experiment 2 :** data queries

1. **Find the SIDs and names of all students who have enrolled a number 1 course**

SQL> select sno,sname

2 from s

3 where sno in(select sno from sc where cno=1);

SNO SNAME

---------- ----------

191564101 nicho

191564102 natii

1. **Find the SIDs and names of all students who have enrolled a data structure course**

SQL> select sno,sname

from s

where sno in(select sno from sc where cno=(select cno from c where cname='data str'));

SNO SNAME

---------- ----------

191564101 nicho

1. **Find the SIDs and names of all students who have not enrolled a number 1 course.**

SQL> SELECT Sno, Sname

2 FROM S

3 WHERE Sno not in

4 (SELECT Sno

5 FROM SC

6 WHERE Cno = '1');

SNO SNAME

---------- ----------

191564104 tsegaye

191564103 henry

191564105 abel

1. **Find the names of all students who have enrolled all courses.**

SQL> select s1.sname,s1.sno

from s s1 inner join sc s2 on s1.sno=s2.sno

group by s1.sno,s1.sname

having count(\*)=(select count(\*) from c);

SNAME SNO

---------- ----------

nicho 191564101

1. **Find the SIDs and average grade of all students who have passed all courses except for a number 1**

SQL> SELECT sno, AVG(grade)

2 FROM sc

3 WHERE grade >= 60 AND sno in ( SELECT sno

4 FROM sc

5 WHERE cno != 1)

6 GROUP BY sno

7 ORDER BY AVG(grade) DESC;

SNO AVG(GRADE)

---------- ----------

191564104 98

191564102 96.3333333

191564103 96

191564101 89.6

191564105 74

1. **Find the name of student who has enrolled a database course with the second highest score.**

SQL> select sname from s where sno=(select sno

from sc

where grade=(select max(grade) from sc

where cno=(select cno from c where cname='data base')AND

grade <(select max(grade) from sc

where cno=(select cno from c where cname='data base'))));

SNAME

----------

natii

1. **Find the names of all students who have enrolled at least 3 courses with 3 credits and whose score of each course enrolled is 80 or more.**

SQL> SELECT Sname FROM S

WHERE Sno IN(SELECT Sno FROM SC

WHERE Cno IN (SELECT Cno FROM C

WHERE Credit>=3) AND Grade>=80

GROUP BY Sno HAVING COUNT(\*)>=3);

SNAME

----------

natii

nicho

1. **Find the SIDs of students whose number of courses taken is unique.**

SQL> SELECT Sno FROM SC

GROUP BY Sno HAVING COUNT(\*)=1;

SNO

----------

191564105

1. **Use SELECT statement to do the queries of various kinds of WHERE conditions.**

SQL> select \* from c where credit >3;

CNO CNAME CPNO CREDIT

---------- ---------- ---------- ----------

4 data str 4

1 data base 4

2 networking 3.5

SQL> select sname from s where sname like 'n%';

SNAME

----------

nicho

natii

SQL> select \* from c,sc where cname='data base' and sc.cno=c.cno;

CNO CNAME CPNO CREDIT SNO CNO GRADE

---------- ---------- ---------- ---------- ---------- ---------- ----------

1 data base 4 191564101 1 90

1 data base 4 191564102 1 85

1 data base 4 191564104 1 55

**Experiment 3:**  modification and deletion of data

1. **All grades except for null value in a number 1 course are to be increased by 10 percent**.

SQL> UPDATE SC

2 SET grade = grade \* (1 +0.1)

3 WHERE grade not in(select grade from sc where cno = 1 AND grade is NULL);

12 rows updated.

SQL> commit;

Commit complete.

1. **Delete all records enrolled at data structure course in the SC table.**

SQL> Delete from sc

Where cno=(select cno from c where cname=’data str’);

2 rows deleted.

1. **Delete all records in the SC and S table whose student number is 191564103**

SQL> delete from sc

2 where sno =191564103;

2 rows deleted.

SQL> delete from s

2 where sno = 191564103;

1 row deleted.

**Experiment 4:** the operations of view.

1. **Define a view of male students whose attributes include SID, student name, course name and grade enrolled**.

SQL> conn system

Enter password:

Connected.

SQL> GRANT CREATE VIEW TO SCOTT;

Grant succeeded.

SQL> CONNECT scott

Enter password:

Connected.

SQL> SQL> CREATE VIEW view1(Sno,Sname,Cname,Grade)

2 AS SELECT S.Sno, Sname, Cname, Grade

3 FROM S, SC, C

4 WHERE S.Sno=SC.Sno AND

5 Sgender='m' AND C.Cno=SC.Cno;

View created.

SQL> commit;

Commit complete.

SQL> select \* from view1;

SNO SNAME CNAME GRADE

---------- ---------- ---------- ----------

191564101 nicho data base 99

191564102 natii data base 94

191564105 abel data str 74

191564101 nicho data base 61

191564104 tsegaye networking 98

191564101 nicho networking 106

191564102 natii networking 98

191564101 nicho data str 86

191564102 natii AI 97

191564101 nicho AI 96

10 rows selected.

1. **Find the SIDs and names of all students in the previous view with average grade 80 and more.**

SQL> Select sno ,avg(grade)

From view1

Group by sno

Having avg(grade)>=80;

SNO AVG(GRADE)

---------- ----------

191564101 89.6

191564102 96.3333333

191564104 98

**Experiment 5:** library function and access control.

1. **Calculate the number of courses taken and average grade for each student whose grade is not null.**

SQL> SELECT sno,COUNT(\*), AVG(Grade) FROM SC

WHERE Grade IS NOT NULL GROUP BY SNO;

SNO COUNT(\*) AVG(GRADE)

---------- ---------- ----------

191564102 3 96.3333333

191564103 2 96

191564105 1 74

191564104 1 98

191564101 5 89.6

1. **Using the GRANT statement, grant various privileges on the base tables of S, SC and C to other users.**

SQL> conn system

Enter password:

Connected.

SQL> GRANT CREATE VIEW TO SCOTT;

Grant succeeded.

SQL> GRANT SELECT, UPDATE, INSERT

2 ON s TO "NATAN";

Grant succeeded.

SQL> GRANT SELECT, UPDATE, INSERT

ON C TO "NATAN";

Grant succeeded.

SQL> GRANT SELECT, UPDATE, INSERT

ON SC TO "NATAN";

Grant succeeded.

SQL> GRANT select on s to natan;

Grant succeeded.

SQL> grant update on s to natan;

Grant succeeded.

SQL> grant select on s to public;

Grant succeeded.

SQL> grant update (sname),insert (sid, sname) on sa to natan;

Grant succeeded.

3**. After successful completion of the experiment, withdraw the base tables and views having been created.**

SQL>DROP TABLE SC;

Table dropped.

SQL>DROP TABLE C;

Table dropped.

SQL>DROP TABLE S;

Table dropped.

SQL> drop view view1;

View dropped.

**Experiment 6:** comprehensive experiment: the implementation of a small management information system.

**Description**: This is simple student management system made with pro\*c it contains 4 tables it need’s windows 7 and visual studio. (Deleting, displaying, modifying and creating database with pro\*c)

**CODE PART**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <sqlca.h>

//global declaration

int sa1,sc1,c1;

void welcome();

void workmenu();

void connectuser();

void resiterstudent();

void display();

void menu();

void searchoption();

void search();

void insert();

void createtable();

void detele();

void modify();

void studentresult();

void deletestudentinfo();

void deltescore();

void deletecourse();

void menu3();

void menu4();

void connectuser1();

void main()

{

system(" color F1");

welcome();

sa1=0;

sc1=0;

c1=0;

system("cls");

workmenu();

}

void connectuser()

{

//declaration

char username[10],server[10],password[10];

printf("user name:");

scanf("%s",username);

printf("password:");

scanf("%s",password);

printf("server:");

gets(server);

system("cls");

EXEC SQL CONNECT :username IDENTIFIED BY :password USING :server;

if(sqlca.sqlcode==0)

printf("Connected\n" );

else

{

printf("wrong user name or password\n");

printf("%.\*s\n",sqlca.sqlerrm.sqlerrml,sqlca.sqlerrm.sqlerrmc);

connectuser();

}

}

int menu\_select()

{

char i;

do{

printf("enter 1 to display all data\n");

printf("enter 2 to search \n");

printf("enter 3 to insert \n");

printf("enter 4 to delete \n");

printf("enter 5 to modify\n");

printf("enter 6 to save and close connection\n");

printf("enter 7 to reconnect\n");

printf("enter 8 to go to main menu\n");

printf("Please input index(1-7):\n");

i=getchar();

}while(i<'1' || i>'7');

return (i-'0');

}

void search(){

char id[10],name[10],email[10],adress[10],phonenumber[10],gender[2],coname[10],average[4];

int a;

searchoption();

scanf("%d",&a);

//system("pause");

printf("please enter student id\n");

scanf("%s",id);

switch(a){

case 1: {

EXEC SQL select sname ,sadress into :name, :adress from SA

where sid=:id;

printf("Name: %s Adress:%s\n ",name,adress);

system("pause");

break;

}

case 2:

{

EXEC SQL select name,email into :name, :email from SA

where upper(name)=upper(:name) AND sid=:id;

printf("Name: %s email:%s\n ",name,email);

system("pause");

break;

}

case 3:

{

EXEC SQL select name,spnumber into :name, :phonenumber from SA

where upper(name)=upper(:name) AND sid=:id;

printf("Name: %s phone number:%s\n ",name,phonenumber);

system("pause");

break;

}

case 4:{

EXEC SQL select name,gender into :name, :gender from SA

where upper(name)=upper(:name) AND sid=:id;

printf("Name: %s gender: %s\n ",name,gender);

system("pause");

break;

}

case 5:{

EXEC SQL DECLARE tmp5 CURSOR FOR

select sname,cname

from s,sc,c

where s.sid=sc.sid and c.cid=sc.cid and sid=:id;

EXEC SQL OPEN tmp5;

EXEC SQL FETCH tmp5 INTO :name,:coname;

printf("student name: %s\n",name);

printf("Course name:%s\n",coname);

EXEC SQL CLOSE tmp5;

system("pause");

break;

}

case 6:{

EXEC SQL select sid, avg(grade) into :id,:average from sc

where sid=:id

group by sid;

printf("id :%s average: %s\n",id,average);

system("pause");

break;

}

default : {printf("error");

}

}

}

void searchoption(){

printf("enter 1 to search adress of student \n");

printf("enter 2 to search email adress of student \n");

printf("enter 3 to search phonenumber of student \n");

printf("enter 4 to search gender of student \n");

printf("enter 5 to search course take by student \n");

printf("enter 6 to search average of each student \n");

}

void insert(){

int press;

char id[10],name[10],adress[10],phonenumber[10],email[10],gender[1],coid[10],grade[3],coname[10];

printf("press 1 to insert new student\n");

printf("press 2 to insert new student score\n");

printf("press 3 to insert new course\n") ;

printf("Fill the following files to insert you cannot make name and student id empty\n");

printf("please enter choice first: ");

scanf("%d",&press);

if(press==1){

sa1=sa1+1;

printf("enter name\n");

scanf("%s",name);

printf("enter id\n");

scanf("%s",id);

printf("enter address\n");

scanf("%s",adress);

printf("enter phone number\n");

scanf("%s",phonenumber);

printf("enter email\n");

scanf("%s",email);

printf("please enter gender\n");

scanf("%s",gender);

while(name==""||id==""){

printf("please enter name \n");

scanf("%s",name);

printf("please enter id\n");

scanf("%s",id);

printf("insertting succeeded");

system("pause");

}

EXEC SQL insert into SA

values (:id,:name,:adress,:email,:phonenumber,:gender);

printf("i am here ");

system("pause");

//EXEC SQL COMMIT;

// break;

}

//inserting new score

else if(press==2){

sc1=sc1+1;

printf("enter student id\n");

scanf("%s",id);

printf("enter course id\n");

scanf("%s",coid);

printf("enter grade\n");

scanf("%s",grade);

EXEC SQL insert into sc

values (:id,:coid,:grade);

//EXEC SQL COMMIT;

}

else if(press==3){

c1=c1+1;

printf("enter course id\n");

scanf("%s",coid);

printf("enter course name\n");

scanf("%s",coname);

while(coid==""||coname==""){

printf("please enter course id \n");

scanf("%s",coid);

printf("pleasecourse name\n");

scanf("%s",coname);

printf("insertting succeeded");

system("pause");

c1++;

}

EXEC SQL insert into c

values (:coid,:coname);

EXEC SQL COMMIT;

}

else {

system("cls");

printf("wrong input");

// break;

}

}

void createtable(){

EXEC SQL CREATE TABLE SA

(

SID number(10)PRIMARY KEY,

sname varchar(10) not null,

sadress VARCHAR(10) not Null,

semail CHAR(10),

spnumber NUMBER(10),

sgender varchar(1)

);

EXEC SQL CREATE TABLE SC

(

SID number(10) primary key,

CID number(10) not null,

grade number(10)

);

EXEC SQL CREATE TABLE C

(

CID number(10) primary key,

CNAME varchar2(10) not Null

);

EXEC SQL CREATE TABLE SP

(

sname varchar2(10) primary key,

spassword number(4)

);

EXEC SQL COMMIT;

}

void detele(){

int choice;

printf("press 1 to delete student data\n");

printf("press 2 to delete student score\n");

printf("press 3 to delete course \n");

scanf("%d",&choice);

switch(choice){

case 1:{

deletestudentinfo();

printf("delete complete\n");

EXEC SQL COMMIT;

system("pause");

break;

}

case 2:{

deltescore();

printf("delete complete\n");

system("pause");

EXEC SQL COMMIT;

break;

}

case 3:{

deletecourse();

printf("delete complete\n");

system("pause");

EXEC SQL COMMIT;

break;

}

default :{

//printf("wrong choice\n");

//system("pause");

break;

}

}

}

void modify(){

int change;

int n;

char name2[10];

char email[10];

char Phonenumber[10];

char adress[10];

char gender[2];

menu3();

printf("please enter choice\n");

scanf("%d",&n);

if(n==1){

menu4();

scanf("%d",&change);

if (change==1){

printf("input name of person to be update:");

scanf("%s",name2);

printf("please input new email:");

scanf("%s",email);

EXEC SQL update SA

set semail=:email

where upper(sname)=upper(:name2);

//EXEC SQL COMMIT;

printf("modification complete\n");

system("pause");

}

else if(change==2){

printf("input name of person to be update:");

scanf("%s",name2);

printf("please input new adress:");

scanf("%s",adress);

EXEC SQL update SA

set sadress=:adress

where upper(sname)=upper(:name2);

EXEC SQL COMMIT;

printf("modification complete\n");

system("pause");

}

else if(change==3){

printf("input name of person to be update:");

scanf("%s",name2);

printf("please input new phonenumber:");

scanf("%s",Phonenumber);

EXEC SQL update SA

set spnumber=:Phonenumber

where upper(sname)=upper(:name2);

EXEC SQL COMMIT;

printf("modification complete\n");

system("pause");

}

else if(change==4){

printf("input name of person to be update:");

scanf("%s",name2);

printf("please input new gender:");

scanf("%s",gender);

EXEC SQL update SA

set sgender=:gender

where upper(sname)=upper(:name2);

EXEC SQL COMMIT;

printf("modification complete\n");

system("pause");

}

else {

printf("input error\n");

}

}

else if(n==2){

char coname[10],grade[10],name[10];

printf("please input course name \n");

scanf("%s",coname);

printf("and student name\n");

scanf("%s",name);

printf("enter new grade:\n");

scanf("%s",grade);

EXEC SQL update Sc

set grade=:grade

where upper(cname)=upper(:coname)

and upper(sname)=upper(:name);

EXEC SQL COMMIT;

printf("modification complete\n");

system("pause");

}

else {

printf("input error\n");

}

}

void display ()

{

int g=sa1;

int m=sc1;

int b=c1;

//declaration

char dis[10], id[10],gender[1],coid[10],grade[3],coname[10], name[10],email[30],adress[10], phonenumber[12];

int a;

printf("press 1 to dispaly student adress\n");

printf("press 2 to display student grade\n");

printf("press 3 to dispaly course list\n");

scanf("%d",&a);

switch(a){

case 1:{

if(sa1==0){

printf("nothing to display\n");

}

system("pause");

EXEC SQL DECLARE tmp1 CURSOR FOR

SELECT sid,sname,sadress,semail,spnumber FROM SA;

EXEC SQL OPEN tmp1;

while(sa1!=0)

{

EXEC SQL FETCH tmp1 INTO :id,:name, :adress, :email, :phonenumber;

printf("ID: %s\n",id);

printf("Name:%s\n",name);

printf("Adress:%s\n",adress);

printf("Email:%s\n",email);

printf("Phone Number:%s\n",phonenumber);

sa1=sa1-1;

system("pause");

}

EXEC SQL CLOSE tmp1;

sa1=g;

break;

}

case 2:{

if(sc1==0){

printf("nothing to display\n");

}

system("pause");

EXEC SQL DECLARE tmp0 CURSOR FOR

SELECT SA.sid,sname,SC.cid,cname,sgrade

FROM SA,SC,C

where sa.sid=sc.sid and sc.cid=c.cid;

EXEC SQL OPEN tmp0;

while(sc1!=0)

{

EXEC SQL FETCH tmp0 INTO :id,:name, :coid,:grade,:coname;

printf("Student ID: %s\n",id);

printf("name:%s\n",name);

printf("course ID:%s\n",coid);

printf("grade:%s\n",grade);

printf("Course name:%s\n",coname);

sc1=sc1-1;

system("pause");

}

EXEC SQL CLOSE tmp0;

sc1=m;

break;

}

case 3:{

if(c1==0){

printf("nothing to display\n");

}

system("pause");

EXEC SQL DECLARE tmp2 CURSOR FOR

select cid,cname

from c;

EXEC SQL OPEN tmp2;

while(c1!=0)

{

EXEC SQL FETCH tmp2 INTO :coid,:coname;

printf("Course ID: %s\n",coid);

printf("Course name:%s\n",coname);

c1=c1-1;

system("pause");

}

EXEC SQL CLOSE tmp2;

c1=b;

break;

}

}

}

void deletestudentinfo(){

char name[10],id[10];

int n,cho;

printf("do you want to delete the whole data?if so press 1, if not press 2\n");

scanf("%d",&n);

if(n==1){

EXEC SQL delete SA;

printf("delete complete\n");

EXEC SQL COMMIT;

system("pause");

sa1=0;

}

else {

printf("do you want to use name or id to delete?\n");

printf("press 1 for id and 2 for name\n");

scanf("%d",&cho);

if(cho==1){

printf("please enter id of student to be deleted\n");

scanf("%s",id);

//EXEC SQL delete SA where upper(name)=upper(:name);

EXEC SQL COMMIT;

sa1=sa1-1;

}

else if (cho==2){

printf("please enter name of student to be deleted\n");

scanf("%s",name);

EXEC SQL delete SA where sid=:id;

sa1=sa1-1;

EXEC SQL COMMIT;

}

else {

printf( "wrong choice\n");

}

}

}

void deltescore(){

char name[10],cho[2],id[10],choice,coname[10],coid[10],grade[3],grade1[3],grade2[3],key[2];

int n ;

printf("do you want to delete the whole data?if so press 1, if not press 2\n");

scanf("%d",&n);

if(n==1){

EXEC SQL delete sc;

sc1=0;

EXEC SQL COMMIT;

printf("whole database deleted\n");

system("pasue");

}

else {

printf("press 1 to delete base on student name\n");

printf("press 2 to delete base on student id\n");

printf("press 3 to delete base on course name\n");

printf("press 4 to delete base on course id\n");

printf("press 5 to delete base on grade\n");

scanf("%d",&choice);

switch(choice){

case 1:{

printf("please student name\n");

scanf("%s",name);

EXEC SQL delete sc

where sid=(select sid from SA where upper(sname)=upper(:name));

sc1=sc1-1;

EXEC SQL COMMIT;

break;

}

case 2: {

printf("please student id\n");

scanf("%s",id);

EXEC SQL delete sc

where sid=:id;

EXEC SQL COMMIT;

sc1=sc1-1;

break;

}

case 3:{

printf("please enter course name\n");

scanf("%s",coname);

EXEC SQL delete sc

where cid=(select cid from c where upper(cname)=upper(:coname));

EXEC SQL COMMIT;

sc1=sc1-1;

break;

}

case 4:{

printf("please course id\n");

scanf("%s",coid);

EXEC SQL delete sc

where cid=:coid;

EXEC SQL COMMIT;

sc1=sc1-1;

break;

}

case 5:{

printf("if you want to insert based single score press 1 else or press any key if you want to insert based on range of score\n");

if(key=="1"){

printf("please enter grade\n");

scanf("%s",grade);

EXEC SQL delete sc

where grade=:grade;

EXEC SQL COMMIT;

sc1=sc1-1;

}

else {

printf("please enter lower range\n");

scanf("%s",grade1);

printf("please enter highest range\n");

scanf("%s",grade2);

EXEC SQL delete sc

where grade between :grade1 and :grade2;

EXEC SQL COMMIT;

sc1=sc1-1;

}

break;

}

}

}

}

void deletecourse(){

char coname[10],coid[10],temp1[20],temp2[10];

int choice;

printf("press 1 to delete based on course name\n");

printf("press 2 to delete based on course id\n");

scanf("%d",&choice);

if(choice==1){

printf("please input course name");

scanf("%s",coname);

EXEC SQL select grade into:temp1 from sc

where cid=(select cid from c where upper(cname)=upper(:coname));

EXEC SQL COMMIT;

if(temp1==""){

EXEC SQL delete c

where upper(cname)=upper(:coname);

c1=c1-1;

EXEC SQL COMMIT;

}

else {

printf("cannot delete because there is grade based on this course\n");

}

}

else if(choice==2){

printf("please input course id\n");

scanf("%s",coid);

EXEC SQL select grade into:temp2 from sc

where cid=:coid;

if(temp2==""){

EXEC SQL delete c

where cid=:coid;

EXEC SQL COMMIT;

c1=c1-1;

}

else {

printf("cannot delete because there is grade based on this course\n");

}

}

}

void menu3(){

printf("Enter 1 to modify student data\n");

printf("enter 2 to modify student grade\n");

}

void menu4(){

printf("enter 1 to modify email\n");

printf("enter 2 to modify adress\n");

printf("enter 3 to modify phone number\n");

printf("enter 4 to modify gender\n");

}

void workmenu(){

int j;

int n;

int q;

while(1){

printf("press 1 for students\n ");

printf("press 2 for teachers\n");

scanf("%d",&j);

if(j==1){

system("cls");

connectuser1();

createtable();

while(1){

system("cls");

printf("press 1 to register\n ");

printf("press 2 to see result and average\n");

printf("press 3 to go back to main menu\n");

scanf("%d",&q);

switch(q){

case 1:{

system("cls");

resiterstudent();

break;

}

case 2:{

system("cls");

studentresult();

break;

}

case 3:{

system("cls");

workmenu();

break;

}

}

}}

else if(j==2){

connectuser();

createtable();

while(1)

{

system("cls");

n= menu\_select();

switch(n){

case 1:{

system("cls");

display();

break;

}

case 2:{

system("cls");

search();

break;}

case 3:{

system("cls");

insert();

break;}

case 4:{

system("cls");

detele();

break;

}

case 5:{

system("cls");

modify();

break;

}

case 6:{

system("cls");

EXEC SQL COMMIT RELEASE;

exit(0);

break;

}

case 7:{

system("cls");

connectuser();

break;

}

case 8:{

system("cls");

workmenu();

break;

}

default:{

printf("wrong input");

system("pause");

break;

}

}

}}

}}

void resiterstudent(){

char name[10],password1[4],password2[4],id[10],adress[10],email[10],phonenumber[10],gender[3];

printf("please enter name:\n");

scanf("%s",name);

printf("please enter id:\n");

scanf("%s",id);

printf("please enter password:\n");

scanf("%s",password1);

printf("please enter confirm password:\n");

scanf("%s",password2);

printf("please enter adress:\n");

scanf("%s",adress);

printf("please enter email:\n");

scanf("%s",email);

printf("please enter phone number:\n");

scanf("%s",phonenumber);

printf("please enter gender write m for male and f for female:\n");

scanf("%s",gender);

if(password1!=password2){

printf("password is not the same please re enter password");

printf("please enter password:\n");

scanf("%s",password1);

printf("please enter confirm password:\n");

scanf("%s",password2);

}

EXEC SQL insert into sp

values(:name,:password1);

EXEC SQL insert into sa

values(:id,:name,:adress,:email,:phonenumber,:gender);

printf("student registered");

system("pasue");

sa1=sa1+1;

EXEC SQL COMMIT;

}

void studentresult(){

char name[10],password[4],id[10],sid[10],grade[4],password2[4],cid[5],average[3];

printf("please enter user name :");

scanf("%s",name);

printf("please enter password:");

scanf("%s",password);

EXEC SQL select sname,spassword into :name,:password2 from sp

where upper(sname)=upper(:name);

if(password!=password2){

printf("wrong user name or password\n");

studentresult();

}

else {

int x=sa1;

EXEC SQL DECLARE tmp10 CURSOR FOR

SELECT sid,sname,grade,cid from sa,sc

where sa.sid=sc.sid and upper(sname)=upper(:name);

EXEC SQL OPEN tmp10;

while(sa1!=0)

{

EXEC SQL FETCH tmp10 INTO :id,:name, :grade, :cid;

printf("ID: %s\n",id);

printf("Name:%s\n",name);

printf("grade:%s\n",grade);

printf("course id:%s\n",cid);

sa1=sa1-1;

system("pause");

EXEC SQL CLOSE tmp10;

}

sa1=x;

EXEC SQL select avg(grade) into average from sc,sa

where sa.sid=sc.sid and upper(sname)=upper(:name)

group by sid;

printf("Average:%s\n",average);

system("pause");

}

}

void welcome(){

printf(

" WELCOME TO INTERNATIONAL STUDENT DATA BASE 1.0\n"

);

system("pause");

}

void connectuser1()

{

//declaration

char username[10]="scott";

char password[10]="tiger";

char server[10];

printf("please enter server");

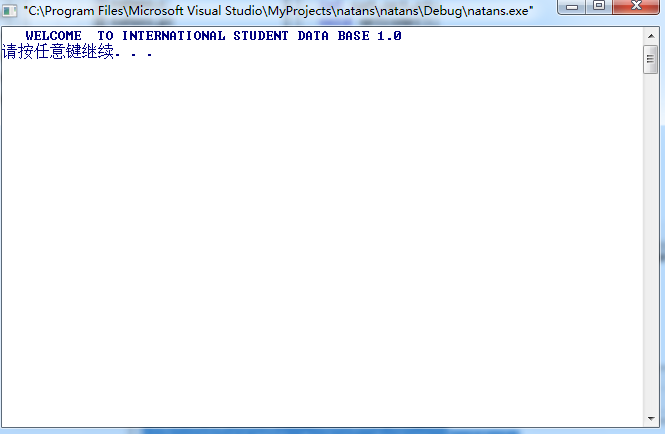
gets(server);

EXEC SQL CONNECT :username IDENTIFIED BY :password USING :server;

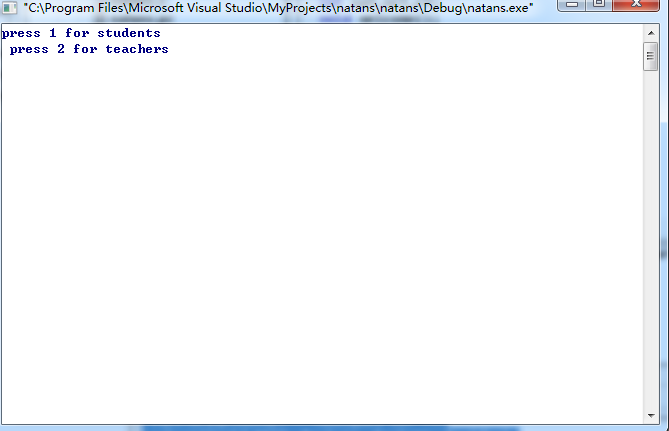
}

**INTERFACE PART**

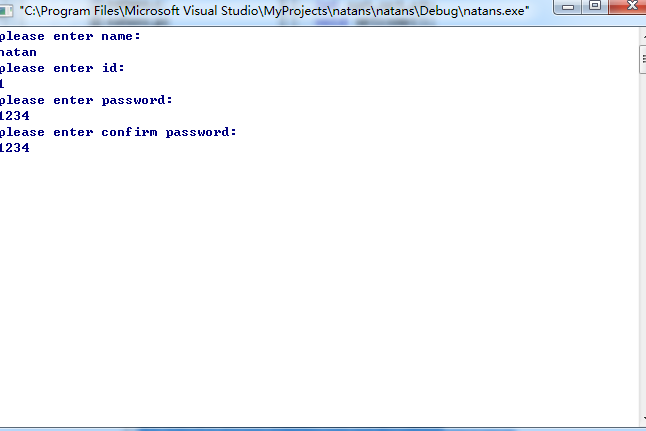
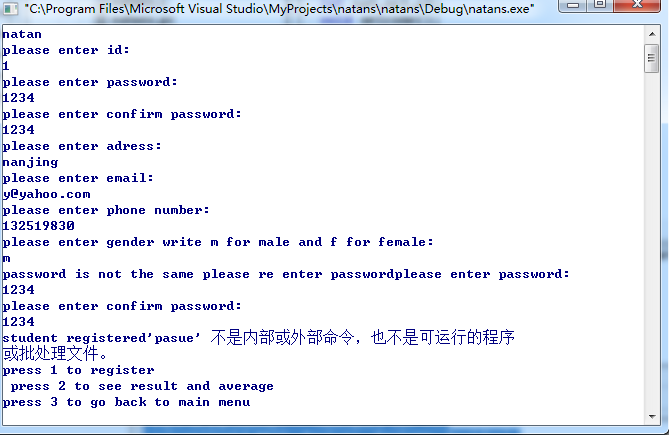
**welcoming**



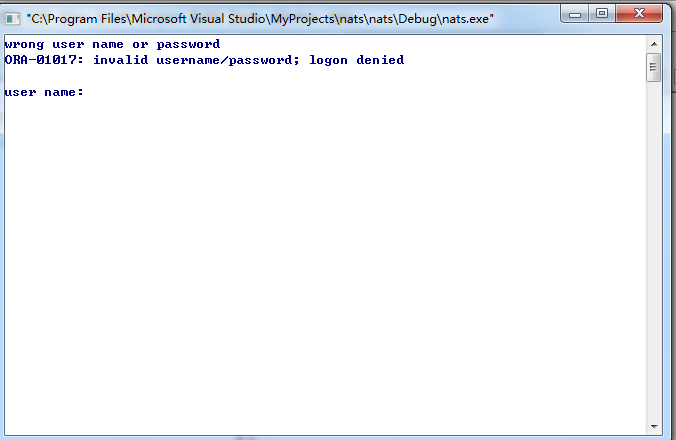
**Choice between student and teacher**

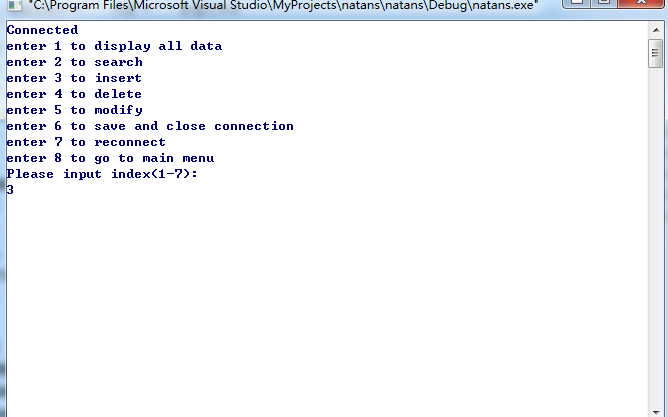


**Registering student into data base**

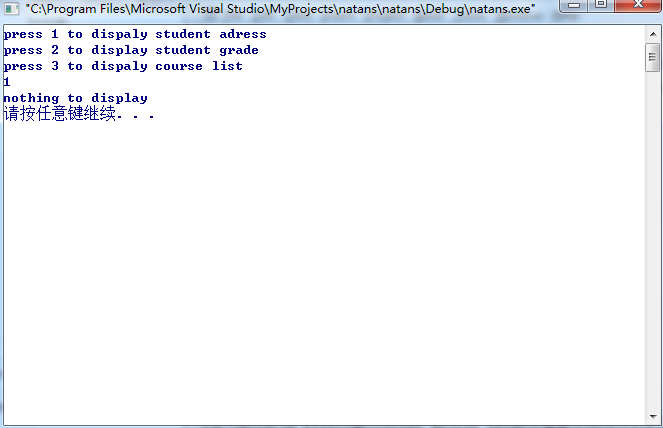
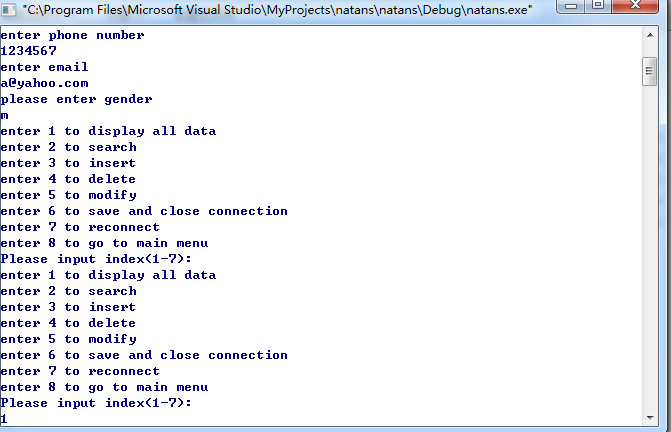
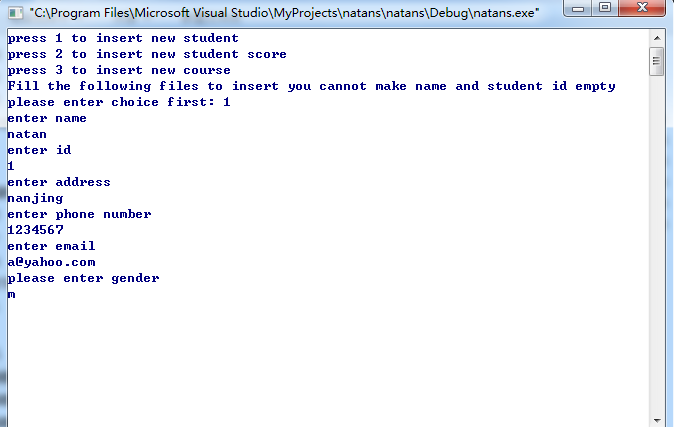
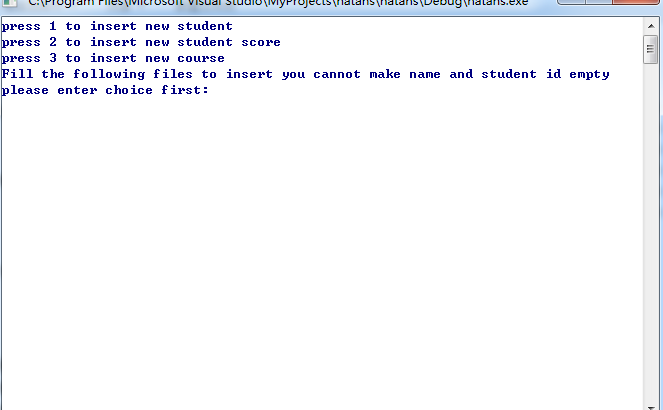


TEACHERS LOGIING IN

**Logging in with wrong password**



**Displaying before inserting empty file**

 **inserting**

Displaying inserted