**FIT5148 Distributed and big data processing**

**As1**

**Luyang Ye 26610000**

**Tasks1-5 are in project named ConferenceApp**

**Task6 is in ConferenceWeb**

**Task1**

**1. SQL code**

**Create Conference table**

CREATE TABLE CONFERENCE

(

con\_id INTEGER NOT NULL ,

con\_name VARCHAR2 (20 CHAR) NOT NULL ,

con\_year INTEGER NOT NULL ,

start\_date DATE NOT NULL ,

end\_date DATE NOT NULL ,

country VARCHAR2 (10 CHAR) NOT NULL ,

city VARCHAR2 (10 CHAR) NOT NULL ,

venue VARCHAR2 (30 CHAR) NOT NULL ,

email VARCHAR2 (30 CHAR) NOT NULL

) ;

ALTER TABLE CONFERENCE ADD CONSTRAINT CONFERENCE\_PK PRIMARY KEY ( con\_id ) ;

**Create Author table**

CREATE TABLE AUTHOR

(

author\_id INTEGER NOT NULL ,

author\_fname VARCHAR2 (10 CHAR) NOT NULL ,

author\_lname VARCHAR2 (10) NOT NULL ,

author\_affiliation VARCHAR2 (10 CHAR) NOT NULL ,

author\_country VARCHAR2 (10 CHAR) NOT NULL ,

author\_email VARCHAR2 (30 CHAR) NOT NULL ,

author\_number INTEGER NOT NULL

) ;

ALTER TABLE AUTHOR ADD CONSTRAINT AUTHOR\_PK PRIMARY KEY ( author\_id ) ;

**Create Award table**

CREATE TABLE AWARD

(

award\_price INTEGER NOT NULL ,

TRACK\_track\_id INTEGER NOT NULL ,

PAPER\_paper\_id INTEGER NOT NULL

) ;

CREATE UNIQUE INDEX AWARD\_\_IDX ON AWARD

(

TRACK\_track\_id ASC

)

;

CREATE UNIQUE INDEX AWARD\_\_IDXv1 ON AWARD

(

PAPER\_paper\_id ASC

);

ALTER TABLE AWARD ADD CONSTRAINT AWARD\_PK PRIMARY KEY ( TRACK\_track\_id, PAPER\_paper\_id ) ;

**Create Paper table**

CREATE TABLE PAPER

(

paper\_id INTEGER NOT NULL ,

paper\_title VARCHAR2 (100 CHAR) NOT NULL ,

paper\_abstract VARCHAR2 (200 CHAR) NOT NULL ,

paper\_type VARCHAR2 (30 CHAR) NOT NULL ,

submit\_date DATE NOT NULL ,

TRACK\_track\_id INTEGER NOT NULL

) ;

ALTER TABLE PAPER ADD CONSTRAINT PAPER\_PK PRIMARY KEY ( paper\_id ) ;

**Create Pc Member table**

CREATE TABLE PC\_MEMBER

(

mem\_id INTEGER NOT NULL ,

mem\_fname VARCHAR2 (10) NOT NULL ,

mem\_lname VARCHAR2 (10 CHAR) NOT NULL ,

mem\_title VARCHAR2 (10 CHAR) NOT NULL ,

mem\_position VARCHAR2 (10 CHAR) NOT NULL ,

mem\_affiliation VARCHAR2 (10) NOT NULL ,

mem\_email VARCHAR2 (30 CHAR) NOT NULL ,

TRACK\_track\_id INTEGER NOT NULL

) ;

ALTER TABLE PC\_MEMBER ADD CONSTRAINT PC\_MEMBER\_PK PRIMARY KEY ( mem\_id ) ;

**Create Review table**

CREATE TABLE REVIEW

(

PC\_MEMBER\_mem\_id INTEGER NOT NULL ,

PAPER\_paper\_id INTEGER NOT NULL ,

review\_date DATE NOT NULL ,

recommendation VARCHAR2 (50 CHAR) NOT NULL ,

due\_date DATE NOT NULL ,

review\_comment VARCHAR2 (100 CHAR)

) ;

ALTER TABLE REVIEW ADD CONSTRAINT REVIEW\_PK PRIMARY KEY ( PC\_MEMBER\_mem\_id, PAPER\_paper\_id ) ;

**Create Submission table**

CREATE TABLE SUBMISSION

(

PAPER\_paper\_id INTEGER NOT NULL ,

AUTHOR\_author\_id INTEGER NOT NULL ,

AUTHOR\_order INTEGER NOT NULL ,

co\_author CHAR (1) NOT NULL

) ;

ALTER TABLE SUBMISSION ADD CONSTRAINT SUBMISSION\_PK PRIMARY KEY ( PAPER\_paper\_id, AUTHOR\_author\_id ) ;

CREATE TABLE TRACK

(

track\_id INTEGER NOT NULL ,

CONFERENCE\_con\_id INTEGER NOT NULL ,

track\_description VARCHAR2 (100 CHAR)

) ;

ALTER TABLE TRACK ADD CONSTRAINT TRACK\_PK PRIMARY KEY ( track\_id ) ;

**Create Foreign Key for each table**

ALTER TABLE AWARD ADD CONSTRAINT AWARD\_PAPER\_FK FOREIGN KEY ( PAPER\_paper\_id ) REFERENCES PAPER ( paper\_id ) ;

ALTER TABLE AWARD ADD CONSTRAINT AWARD\_TRACK\_FK FOREIGN KEY ( TRACK\_track\_id ) REFERENCES TRACK ( track\_id ) ;

ALTER TABLE PAPER ADD CONSTRAINT PAPER\_TRACK\_FK FOREIGN KEY ( TRACK\_track\_id ) REFERENCES TRACK ( track\_id ) ;

ALTER TABLE PC\_MEMBER ADD CONSTRAINT PC\_MEMBER\_TRACK\_FK FOREIGN KEY ( TRACK\_track\_id ) REFERENCES TRACK ( track\_id ) ;

ALTER TABLE REVIEW ADD CONSTRAINT REVIEW\_PAPER\_FK FOREIGN KEY ( PAPER\_paper\_id ) REFERENCES PAPER ( paper\_id ) ;

ALTER TABLE REVIEW ADD CONSTRAINT REVIEW\_PC\_MEMBER\_FK FOREIGN KEY ( PC\_MEMBER\_mem\_id ) REFERENCES PC\_MEMBER ( mem\_id ) ;

ALTER TABLE SUBMISSION ADD CONSTRAINT SUBMISSION\_AUTHOR\_FK FOREIGN KEY ( AUTHOR\_author\_id ) REFERENCES AUTHOR ( author\_id ) ;

ALTER TABLE SUBMISSION ADD CONSTRAINT SUBMISSION\_PAPER\_FK FOREIGN KEY ( PAPER\_paper\_id ) REFERENCES PAPER ( paper\_id ) ;

**Create check on recommendation table**

ALTER TABLE REVIEW ADD CONSTRAINT recommendation\_constraint CHECK ( recommendation IN ('accept', 'weak accept', 'borderline','weak reject','reject')) ;

**Create check on paper table**

ALTER TABLE PAPER ADD CONSTRAINT paper\_constraint CHECK ( PAPER\_TYPE IN ('full paper', 'Research-in-Progress papers', 'posters')) ;

**Because each paper is reviewed less or equal to 3 times, I create a trigger to control it.**

DELIMITER //

CREATE OR REPLACE TRIGGER create\_review

Before

INSERT ON REVIEW

FOR EACH ROW

DECLARE countvalue number;

BEGIN

IF INSERTING THEN

SELECT COUNT(\*) INTO countvalue FROM REVIEW@FIT5148B C

WHERE C.PAPER\_PAPER\_ID = :new.PAPER\_PAPER\_ID;

IF countvalue = 3

THEN RAISE\_APPLICATION\_ERROR(-20002, 'THE REVIEW NUMBER OF THIS PAPER IS ENOUGH');

END IF;

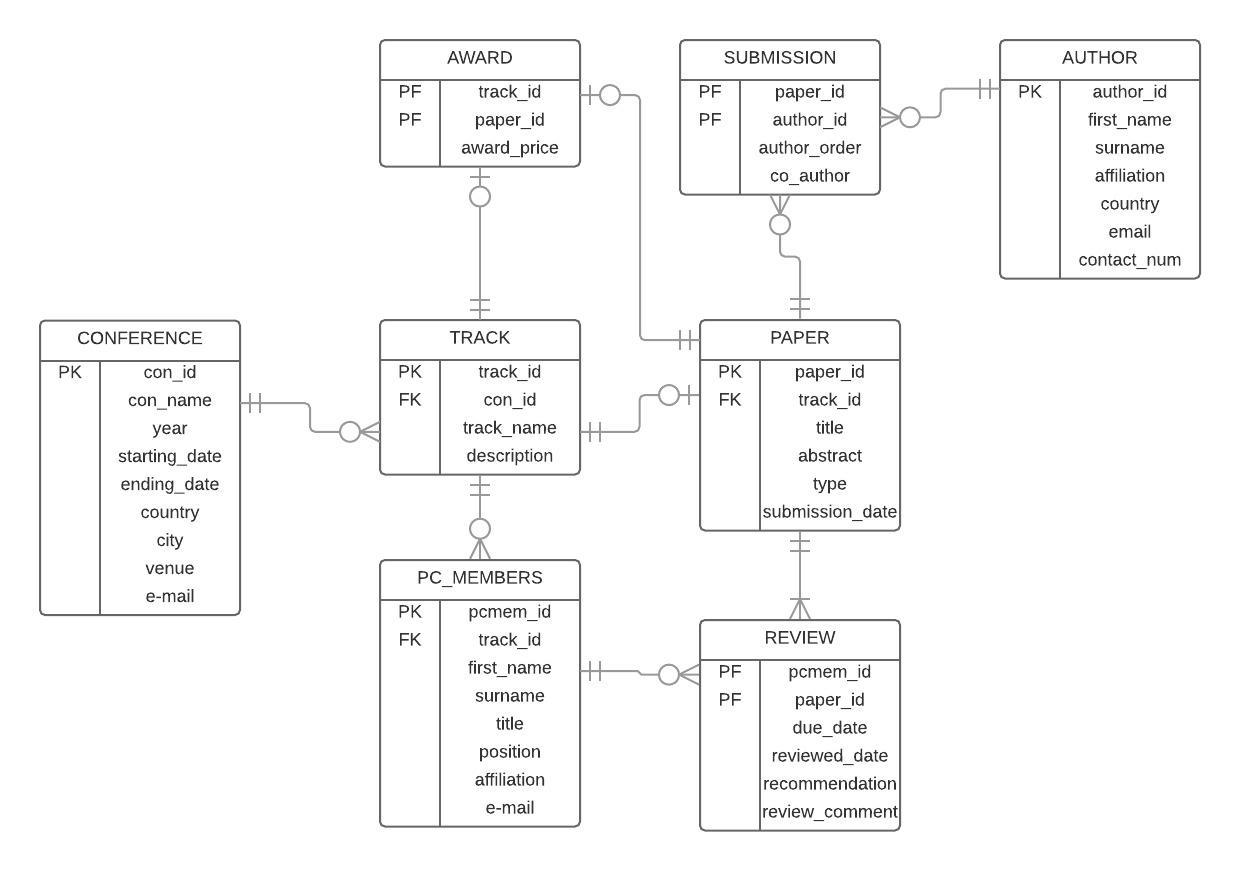
END IF;

END;

/

**2.Conference table is created in FIT5148A. Rest tables are created in FIT5148B**

**3.ER diagram**



In submission table, co\_author is a Boolean value, 1 means the author in this submission record is the corresponding author, 0 means not.

**4. Create index and sequence**

**Create index on Paper by attribute paper\_title**

CREATE INDEX paper\_index ON paper(paper\_title);

**Create Sequence on each table**

CREATE SEQUENCE paper\_id

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE author\_id

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE mem\_id

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE track\_id

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE con\_id

START WITH 1

INCREMENT BY 1;

**5.Population of meaningful data**

**Insert data into Conference table**

insert into CONFERENCE values (con\_id.nextval,'Internet development',2010,'03-JUL-2010','03-AUG-2010','USA','New York','South Cross','internetcumm@gmail.com');

insert into CONFERENCE values (con\_id.nextval,'Modern Art',2014,'01-MAY-2014','03-MAY-2014',’USA’,'Chicago','8TH ARR',chicagoart@gmail.com');

insert into CONFERENCE values (con\_id.nextval,'OCEANS ',2016,'03-APR-2016','13-APR-2016','CHINA','Shanghai','Binjiang Avenue','oceans@yahoo.com'); insert into CONFERENCE values (con\_id.nextval,'Global Warming',2014,'05-Jan-2014','15-Jan-2014','CHINA','Beijing','DONG SI','globalwarm@163.com');

insert into CONFERENCE values (con\_id.nextval,'Wireless Tech',2016,'22-JUN-2016','29-JUN-2016','USA','Kansas','Main Ave','wireless16@gmail.com');

**Inser data into Author table**

insert into AUTHOR values(author\_id.nextval,'Frank','Gab','Yale','UK','frankg@yale.edu','470700693');

insert into AUTHOR values(author\_id.nextval,'Nicol','Smith','UCL','USA','nsmith@gmail.com','488359333');

insert into AUTHOR values(author\_id.nextval,'Alice','Lee','IBM','USA','alicelee@ibm.org','488380633');

insert into AUTHOR values(author\_id.nextval,'Chong','Lin','Peking Uni','CHINA','linchong@pku.edu.cn','1353844959');

insert into AUTHOR values(author\_id.nextval,'Kusimi','Haku','Tokyo Uni','JAPAN','hakuk@yahoo.jp','83798734');

insert into AUTHOR values(author\_id.nextval,'Lucy','White','Monash Uni','AUS','lwhite@monash.edu','466849885');

insert into AUTHOR values(author\_id.nextval,'Leo','Torres','KCL','SPAIN','leotorres@kcl.edu','788707893');

insert into AUTHOR values(author\_id.nextval,'Nancy','Lopez','Yale','UK','nlopez@yale.edu','470778678');

insert into AUTHOR values(author\_id.nextval,'XiaoRan','Deng','Peking Uni','CHINA','dengxr@pku.edu.cn','1503422234');

insert into AUTHOR values(author\_id.nextval,'Jack','Irwan','Yale','UK','jirwan@yale.edu','470767868');

insert into AUTHOR values(author\_id.nextval,'Mike','Jason','UCL','USA','mjason@gmail.com','484569363');

insert into AUTHOR values(author\_id.nextval,'David','Cook','UCL','CANADA','ddcook@gmail.com','463459365');

insert into AUTHOR values(author\_id.nextval,'Lon','Kin','Peking Uni','CHINA','kinlon@pku.edu.cn','1503434564');

**Insert data into Track table**

insert into TRACK values(track\_id.nextval,6,'Ocean Resource and Development');

insert into TRACK values(track\_id.nextval,3,'Big Data Research');

insert into TRACK values(track\_id.nextval,3,'Mobile and Distribute');

insert into TRACK values(track\_id.nextval,8,'New Tech Share');

insert into TRACK values(track\_id.nextval,10,'Security');

**Insert data into Paper table**

insert into paper values(paper\_id.nextval,'The Future of Ocean Resource Development','With the new technology for oceans exploitation...','Research-in-Progress papers','03-JAN-2016','1');

insert into paper values(paper\_id.nextval,'Significance and Challenges of Big Data Research','In recent years, the rapid development of Internet, Internet of Things, and Cloud Computing have led to the explosive growth of data in almost every industry and business area...','full paper','03-MAY-2010','2');

insert into paper values(paper\_id.nextval,'The Indian Ocean nodule field geology and resource potential','The paper includes a synthesis of research findings on the structure and evolution of the Central Indian Ocean Basin and its ferromanganese deposits...','full paper','20-JAN-2016','1');

insert into paper values(paper\_id.nextval,'Decomposing social in emerging big data research','This study conducts a social network analysis of international co-authorship relationships in “big data” science and analyzes semantic networks','full paper','23-JUN-2010','2');

insert into paper values(paper\_id.nextval,'Complex event process infrastructure in mobile distributed system','With the growing success of the mobile computing devices, such as laptop computer, third generation mobile phone, personal digital assistant...','full paper','22-FEB-2010','3');

insert into paper values(paper\_id.nextval,'Three-Phase Algorithms for Task Scheduling in Distributed Mobile System','A distributed mobile DSP system consists of a group of mobile devices with different computing powers.','Research-in-Progress papers','04-JAN-2010','3');

insert into paper values(paper\_id.nextval,'CO2 capture, transport and industrial applications','Carbon dioxide (CO2) capture and storage (CCS) is the one advanced technology that conventional power generation cannot do without. CCS technology reduces the carbon...','full paper','20-DEC-2013','4');

insert into paper values(paper\_id.nextval,'Carbon dioxide recovery and utilization','Carbon Dioxide Recovery and Utilization is a complete and informative resource on the carbon dioxide sources and market at the European Union level...','full paper','11-DEC-2013','4');

insert into paper values(paper\_id.nextval,'Wireless Network Security: Building on Erasures','Proceedings of the IEEE.Description: One of the most widely-known techniques for securing a message...','full paper','03-APR-2016','5');

insert into paper values(paper\_id.nextval,'Signal-flow-based analysis of security protocols','Security protocols operating over wireless channels can incur significant communication costs (e.g., energy, delay), ...','full paper','05-APR-2016','5');

**Insert data into Submission table**

insert into submission values(2,4,1,1);

insert into submission values(2,9,2,0);

insert into submission values(3,1,1,1);

insert into submission values(3,3,2,0);

insert into submission values(5,2,1,1);

insert into submission values(8,11,1,1);

insert into submission values(8,12,2,0);

insert into submission values(10,6,1,1);

insert into submission values(9,5,1,1);

insert into submission values(13,5,1,1);

insert into submission values(15,11,1,1);

insert into submission values(15,2,2,0);

insert into submission values(13,13,1,1);

insert into submission values(20,10,1,1);

**Insert data into award table**

insert into award values('10000','1','2');

insert into award values('5000','2','3');

insert into award values('24000','3','15');

insert into award values('5000','4','8');

insert into award values('7000','5','10');

**Insert data into Pc Member table**

insert into pc\_member values(mem\_id.nextval,'Robert','Murray','Dr','researcher','MIT','mrobert@gmail.com',3);

insert into pc\_member values(mem\_id.nextval,'Amanda','Ryan','Prof','lecturer','MIT','amaryan@gmail.com',3);

insert into pc\_member values(mem\_id.nextval,'Joe','Ramer','Prof','lecturer','Havard','jramer32@gmail.com',3);

insert into pc\_member values(mem\_id.nextval,'Yasaku','Sanso','Dr','researcher','Tokyo Uni','sanso333@yahoo.jp',1);

insert into pc\_member values(mem\_id.nextval,'Billy','Logan','Prof','lecturer','Cambridge','blogan@cambridge.edu',2);

**Insert data into Review table**

insert into review values(1,15,'24-FEB-2010','accept','15-MAR-2010','Really good view about a new type of...');

insert into review values(2,15,'25-FEB-2010','accept','15-MAR-2010','A new idea about...');

insert into review values(3,15,'24-FEB-2010','accept','15-MAR-2010','Great work...');

insert into review values(4,13,'25-JAN-2016','weak reject','27-FEB-2016','Unprovable idea...');

insert into review values(5,5,'27-JUN-2010','accept','12-JUL-2010','GOOD,no doubt to be the...');

**Task 2**

**create Oracle Stored Procedure**

DELIMITER //

create or replace procedure updateVenue(

p\_con\_id in conference.con\_id%type,

p\_venue in conference.venue%type)

is

begin

update conference

set

venue = p\_venue

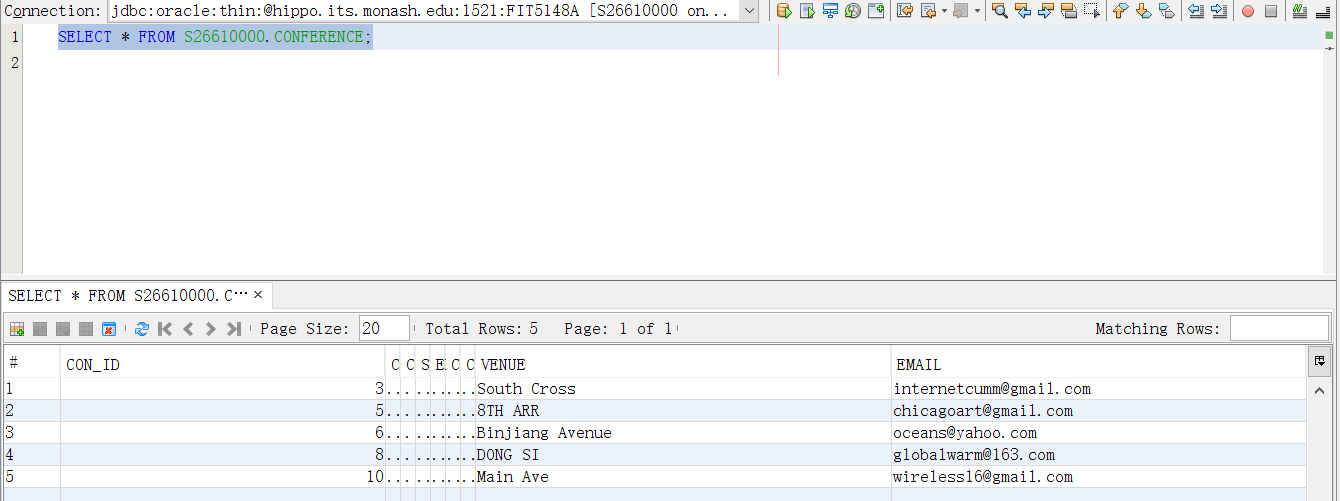
where con\_id = p\_con\_id;

END;

/

**Before updating venue:**

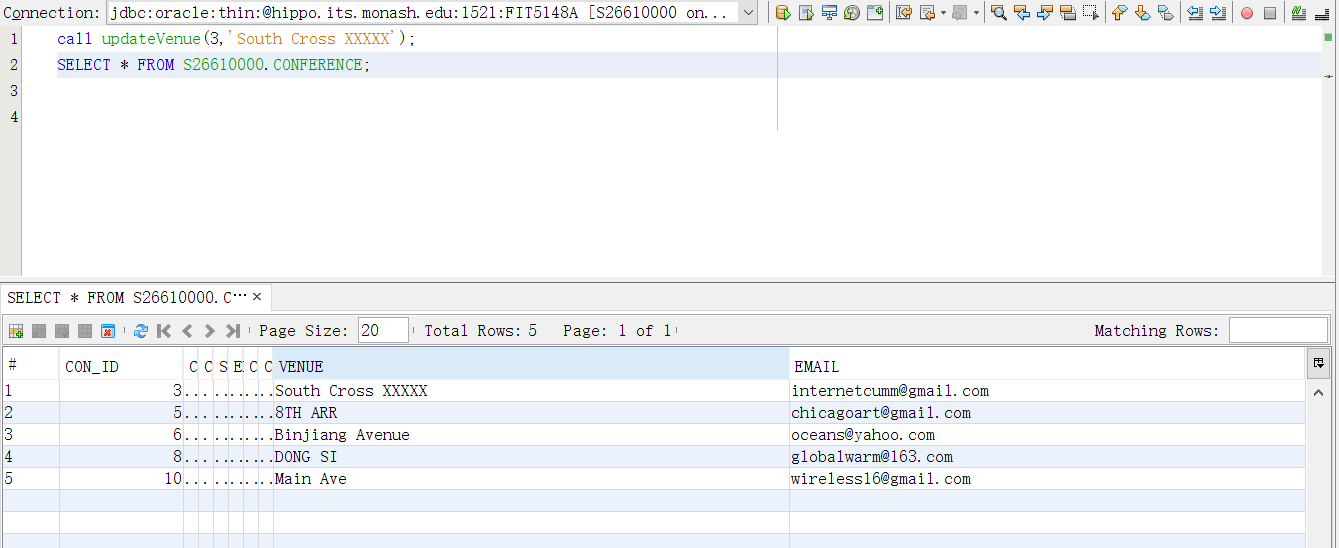
SELECT \* FROM S26610000.CONFERENCE;



**Use updateVenue procedure**

call updateVenue(3,'South Cross XXXXX');

SELECT \* FROM S26610000.CONFERENCE;



**Task 3.**

**When update track or insert a new record of track, it is necessary to make sure the conference id is already existed.**

DELIMITER //

CREATE OR REPLACE TRIGGER change\_track

Before

UPDATE OR INSERT ON TRACK

FOR EACH ROW

DECLARE countvalue number;

BEGIN

IF INSERTING OR UPDATING THEN

SELECT COUNT(\*) INTO countvalue FROM CONFERENCE@FIT5148A C

WHERE C.CON\_ID = :new.CONFERENCE\_CON\_ID;

IF countvalue = 0

THEN RAISE\_APPLICATION\_ERROR(-20002, 'THE VALUE OF FOREIGN KEY SHOULD EQUAL TO THE VALUE OF PRIMARY KEY');

END IF;

END IF;

END;

/

**When delete a conference record, users need to make sure there is no record in track table is related to this conference.**

DELIMITER //

CREATE OR REPLACE TRIGGER change\_conference

Before

DELETE ON CONFERENCE

FOR EACH ROW

DECLARE countvalue number;

BEGIN

IF DELETING THEN

SELECT COUNT(\*) INTO countvalue FROM TRACK@FIT5148B C

WHERE C.CONFERENCE\_CON\_ID = :old.CON\_ID;

IF countvalue > 0

THEN RAISE\_APPLICATION\_ERROR(-20002, 'THE VALUE OF FOREIGN KEY SHOULD EQUAL TO THE VALUE OF PRIMARY KEY');

END IF;

END IF;

END;

/

**Task 4**

**a) Horizontal fragmentation**

**Assumption**:

All of the conferences in the database are held in two countries, China or USA. So it is necessary to separate CONFERENCE table into two fragments to make it easier for users to search the conferences they want to know about.

List of applications:

Application1: Find the information about each conference which is held in USA

Application2: Find the information about each conference which is held in China

For application1 :

P1 : (LOC=”USA”)

~~P2：not(LOC=”USA”)~~ equals LOC=”CHINA”

For application2:

P2: (LOC=”CHINA”)

~~P3: not(LOC=”CHINA”)~~ equals LOC=”USA”

elimination:

~~P3: (LOC=”USA”) ∧ (LOC=”CHINA”)~~

~~P4: not(LOC=”USA”) ∧ not(LOC=”CHINA”)~~

Minterm fragments left after elimination

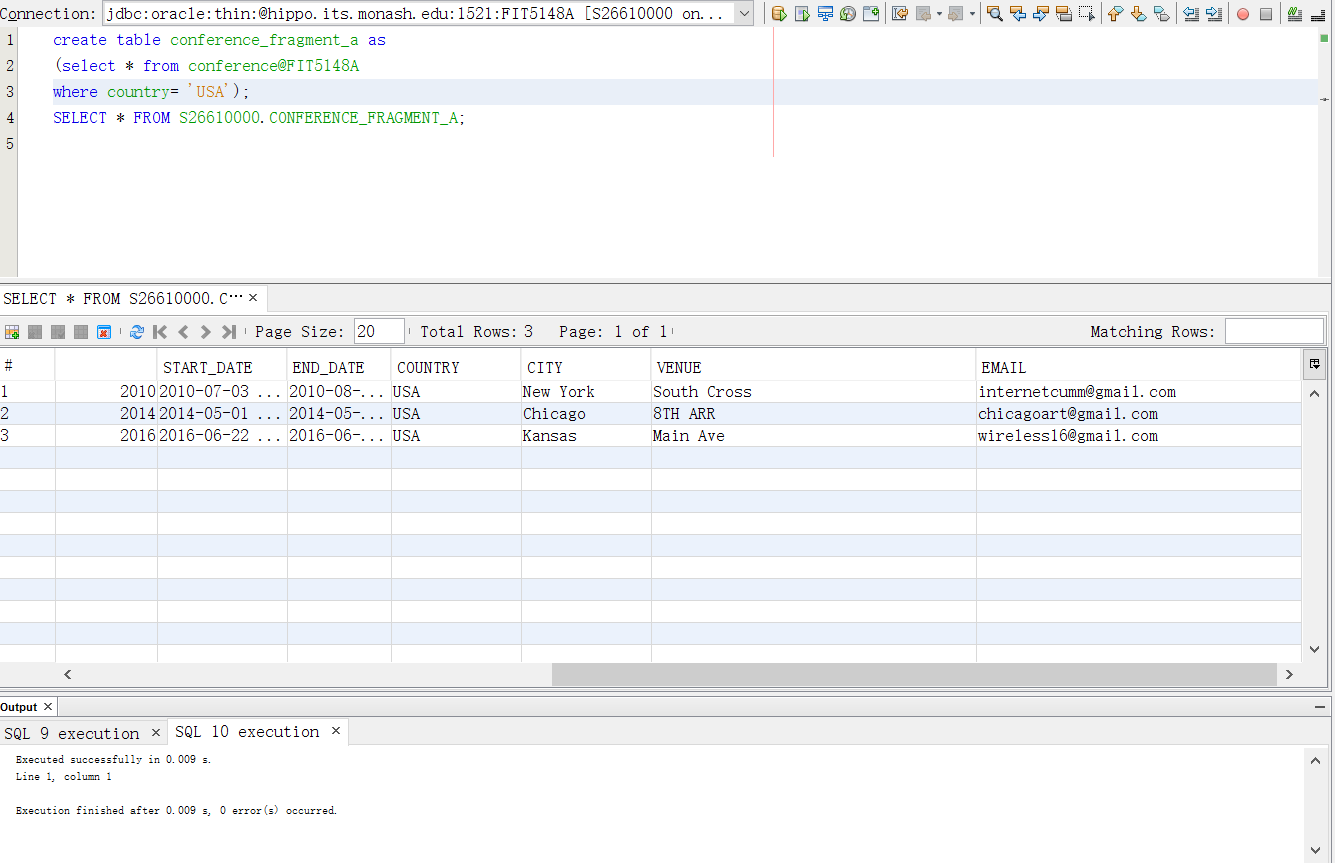
m1: (LOC=”USA”)

m2: (LOC=”CHINA”)

**At FIT5148A**

create table conference\_fragment\_a as

(select \* from conference@FIT5148A

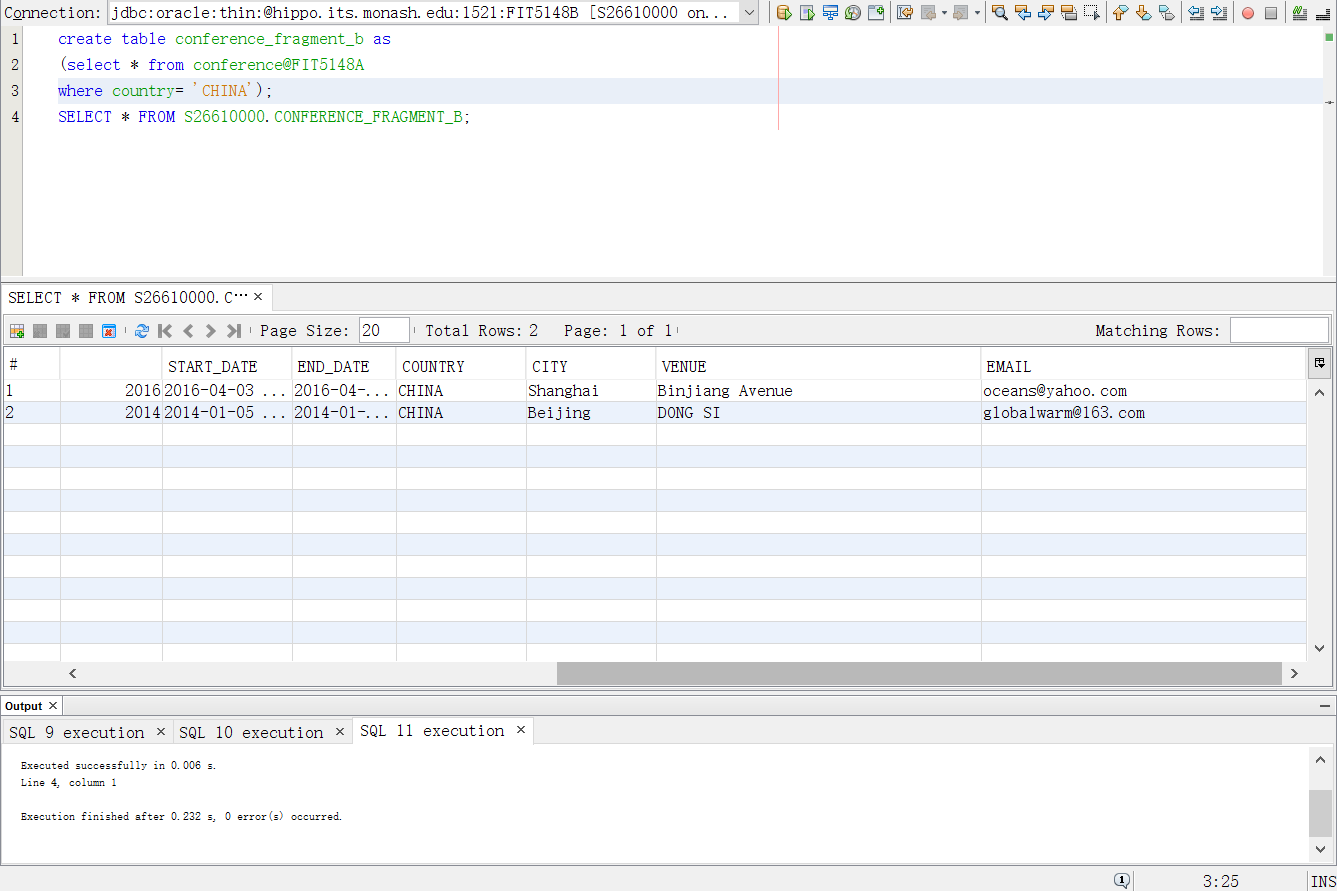
where country = 'USA');

**At FIT5148B**

create table conference\_fragment\_b as

(select \* from conference@FIT5148A

where country= 'CHINA');



**b) Vertical fragmentation**

In this conference system, PAPER table is the most vital table because users will search in this table a lot to find the information about the paper they are interested in. Different users may focus on different kind of information about the paper, So that’s why I choose paper table to do the vertical fragmentation.

**Four queries:**

q1: select paper\_title from paper where paper\_id = value;

q2: select track\_id from paper where paper\_title = value;

q3: select paper\_abstract, paper\_type from paper where paper\_title = value;

q4: select submit\_date from paper where track\_id = value;

**Usage Matrix**

Because paper\_id is the primary key so it has to be in every fragment, that’s why it is unnecessary to put it into usage matrix

A1: paper\_title

A2: paper\_abstract

A3: paper\_type

A4: submit\_time

A5: track\_id

A1 A2 A3 A4 A5

q1 1 0 0 0 0

q2 1 0 0 0 1

q3 1 1 1 0 0

q4 0 0 0 1 1

**Access Frequency Matrix**

S1 S2 sum

q1 5 20 25

q2 30 15 45

q3 15 20 35

q4 0 5 5

**Attribute Affinity Matrix**

A1 A2 A3 A4 A5

A1 105 35 35 0 45

A2 35 35 35 0 0

A3 35 35 35 0 0

A4 0 0 0 5 5

A5 45 0 0 5 50

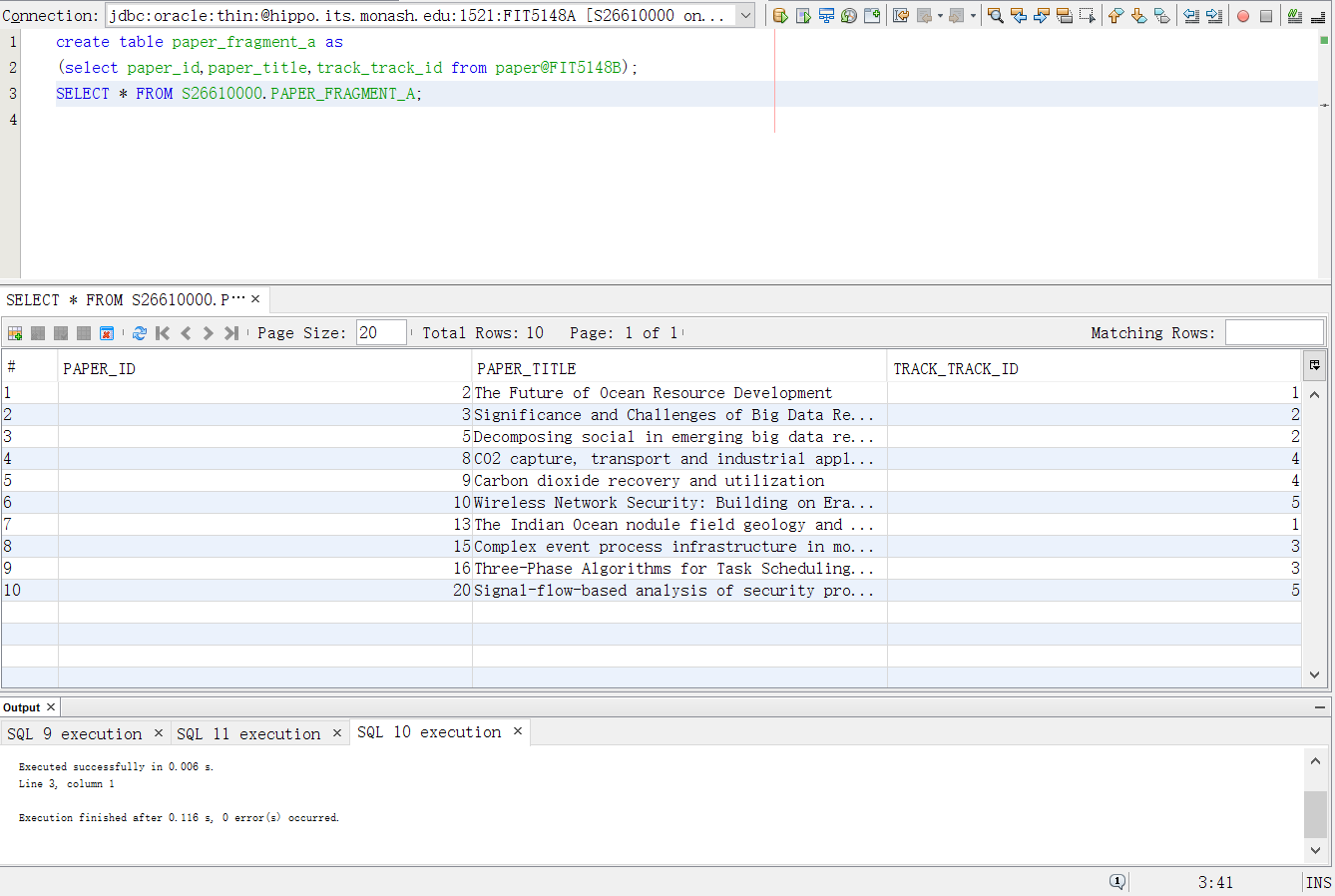
According to the result of affinity matrix, paper\_title and track\_id plus PK paper\_id should be in one site. Paper\_ abstract, paper\_type and submit\_time plus PK are in the other site

**At FIT5148A**

create table paper\_fragment\_a as

(select paper\_id,paper\_title,track\_track\_id from paper@FIT5148B);

SELECT \* FROM S26610000.PAPER\_FRAGMENT\_A;

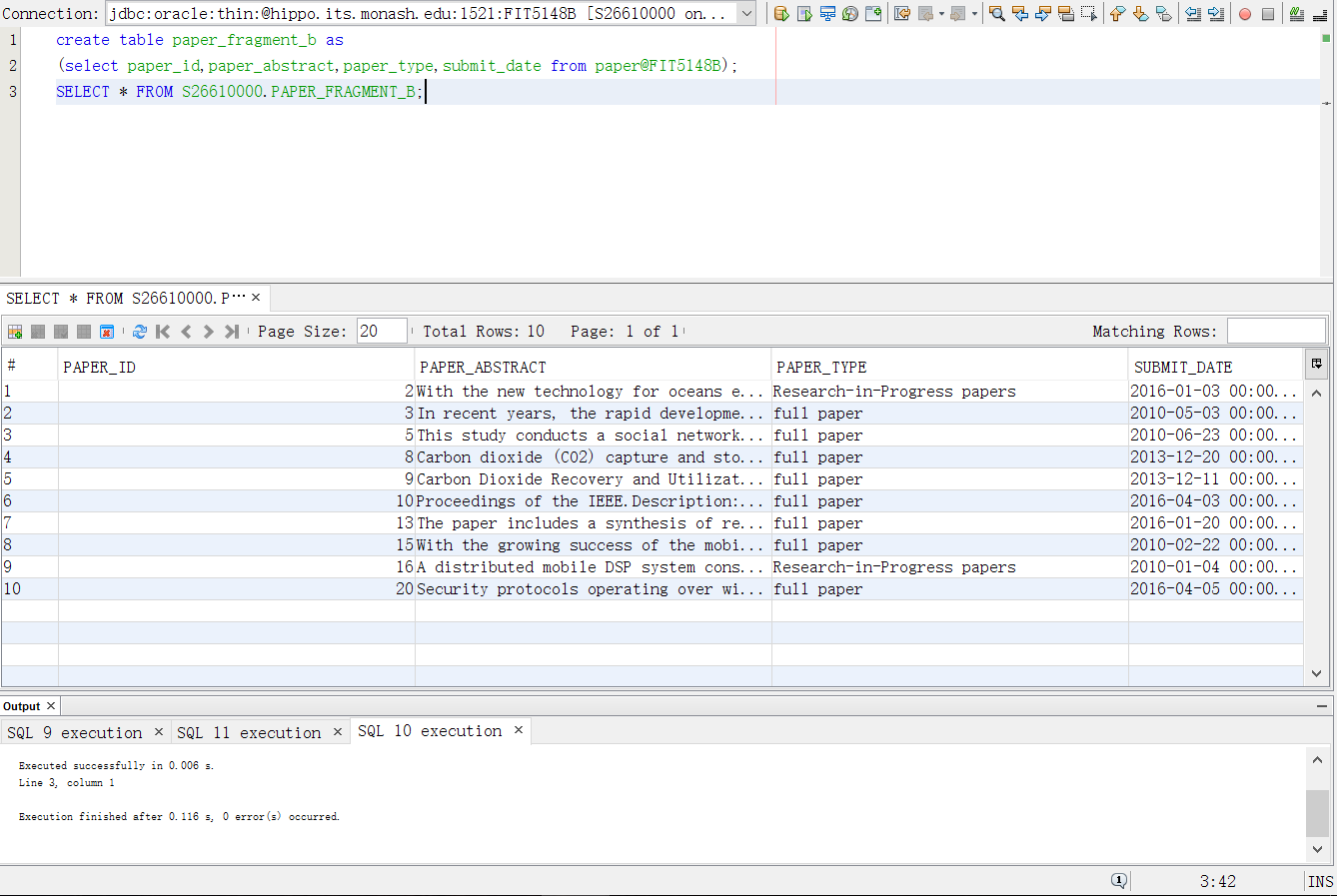


**At FIT5148B**

create table paper\_fragment\_b as

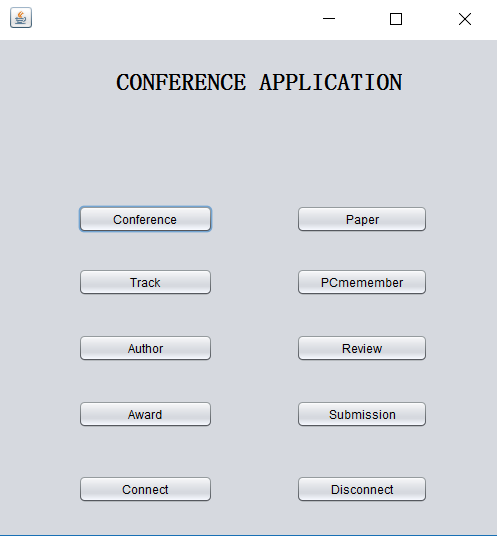
(select paper\_id,paper\_abstract,paper\_type,submit\_date from paper@FIT5148B);

SELECT \* FROM S26610000.PAPER\_FRAGMENT\_B;



**Task 5.**

**Main Frame**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

ConferenceFrame ConferenceFrame = new ConferenceFrame();

ConferenceFrame.setVisible(true);

ConferenceFrame.setDefaultCloseOperation(ConferenceFrame.DISPOSE\_ON\_CLOSE);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

TrackFrame TrackFrame = new TrackFrame();

TrackFrame.setVisible(true);

TrackFrame.setDefaultCloseOperation(TrackFrame.DISPOSE\_ON\_CLOSE);

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

AuthorFrame AuthorFrame = new AuthorFrame();

AuthorFrame.setVisible(true);

AuthorFrame.setDefaultCloseOperation(AuthorFrame.DISPOSE\_ON\_CLOSE);

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

AwardFrame AwardFrame = new AwardFrame();

AwardFrame.setVisible(true);

AwardFrame.setDefaultCloseOperation(AwardFrame.DISPOSE\_ON\_CLOSE);

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

PaperFrame PaperFrame = new PaperFrame();

PaperFrame.setVisible(true);

PaperFrame.setDefaultCloseOperation(PaperFrame.DISPOSE\_ON\_CLOSE);

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

PcmemeberFrame PcmemeberFrame = new PcmemeberFrame();

PcmemeberFrame.setVisible(true);

PcmemeberFrame.setDefaultCloseOperation(PcmemeberFrame.DISPOSE\_ON\_CLOSE);

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

ReviewFrame ReviewFrame = new ReviewFrame();

ReviewFrame.setVisible(true);

ReviewFrame.setDefaultCloseOperation(ReviewFrame.DISPOSE\_ON\_CLOSE);

}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

SubmissionFrame SubmissionFrame = new SubmissionFrame();

SubmissionFrame.setVisible(true);

SubmissionFrame.setDefaultCloseOperation(SubmissionFrame.DISPOSE\_ON\_CLOSE);

}

private void closeActionPerformed(java.awt.event.ActionEvent evt) {

try{

conn1.close();

System.out.println("Connection A closed");

conn2.close();

System.out.println("Connection B closed");

} catch(SQLException ex){

System.out.println("error in closing connection");

} // TODO add your handling code here:

}

private void startActionPerformed(java.awt.event.ActionEvent evt) {

try {

DriverManager.registerDriver (new OracleDriver());

conn1 = DriverManager.getConnection ("jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148a", "S26610000", "student");

System.out.println("Connected to Database A");

conn2 = DriverManager.getConnection ("jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b", "S26610000", "student");

System.out.println("Connected to Database B");

}

catch(SQLException f)

{

System.out.println("error in connection");

}

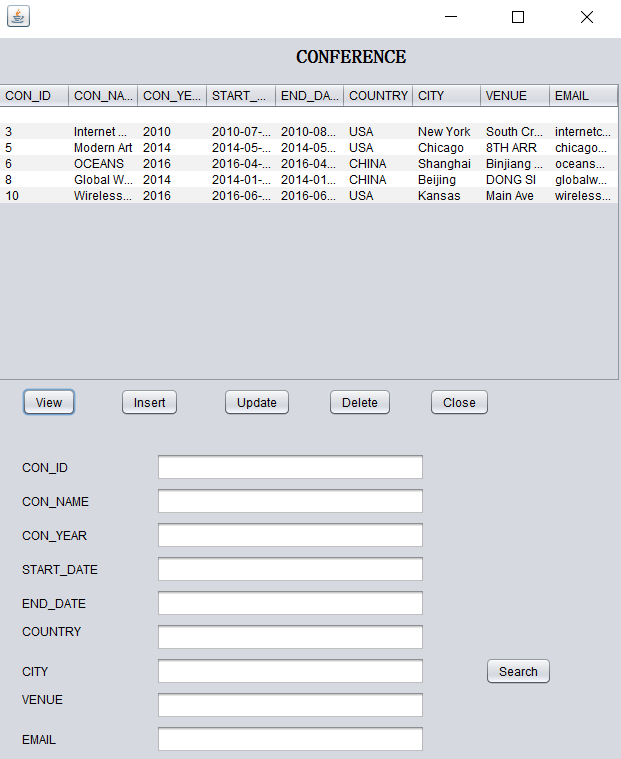
}

{

}

**Conference Frame**

**View**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

dtm.setRowCount(0);

stmt = ConferenceApp.conn1.createStatement();

rset = stmt.executeQuery("select \* from conference");

ResultSetMetaData mdata = rset.getMetaData();

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

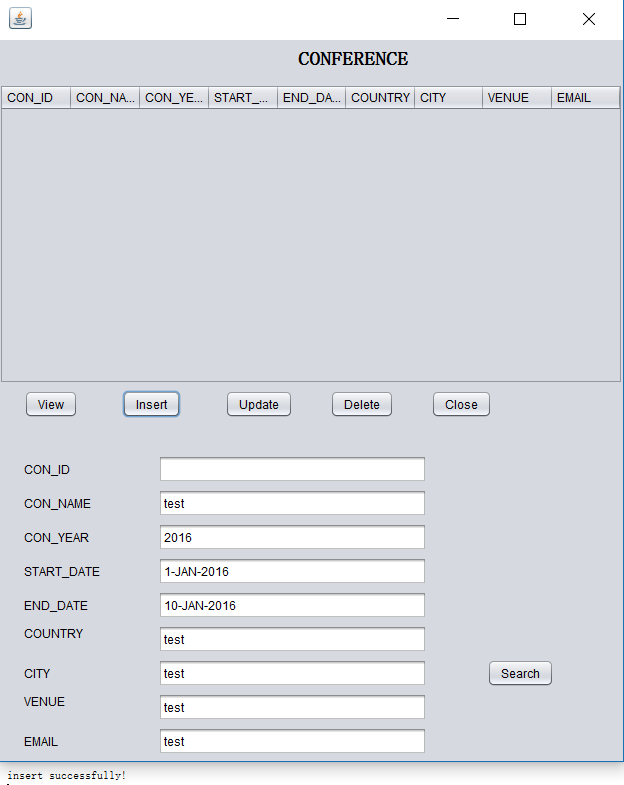
} catch (SQLException f) {

System.out.println(f.getMessage());

}

}

**Insert**



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

String conname = jTextField2.getText();

String conyear = jTextField3.getText();

String startdate= jTextField4.getText();

String enddate = jTextField5.getText();

String country = jTextField6.getText();

String city = jTextField7.getText();

String venue = jTextField8.getText();

String email = jTextField9.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn1.createStatement();

rset = stmt.executeQuery(

"insert into conference values(con\_id.nextval,'"+conname+"','"+conyear+"','"+startdate+"','"+enddate+"','"+country+"','"+city+"','"+venue+"','"+email+"')");

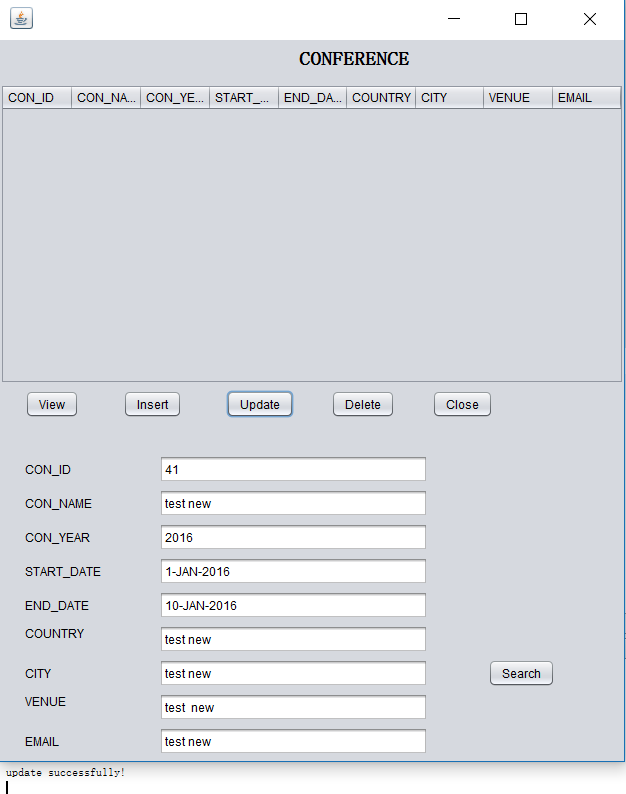
System.out.println("insert successfully!");

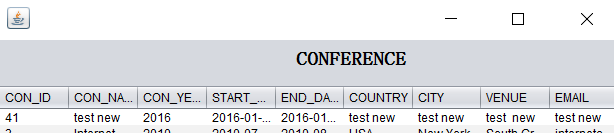
} catch (SQLException f) {

System.out.println(f.getMessage());

} }

**update**





rivate void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

int conid = Integer.parseInt(jTextField1.getText());

String conname = jTextField2.getText();

String conyear = jTextField3.getText();

String startdate= jTextField4.getText();

String enddate = jTextField5.getText();

String country = jTextField6.getText();

String city = jTextField7.getText();

String venue = jTextField8.getText();

String email = jTextField9.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn1.createStatement();

rset = stmt.executeQuery(

"update conference set con\_name='"+conname+"',con\_year='"+conyear+"',start\_date='"+startdate+"',end\_date='"+enddate+"',"

+ "country='"+country+"',city='"+city+"',venue='"+venue+"',email='"+email+"'where con\_id="+conid);

System.out.println("update successfully!");

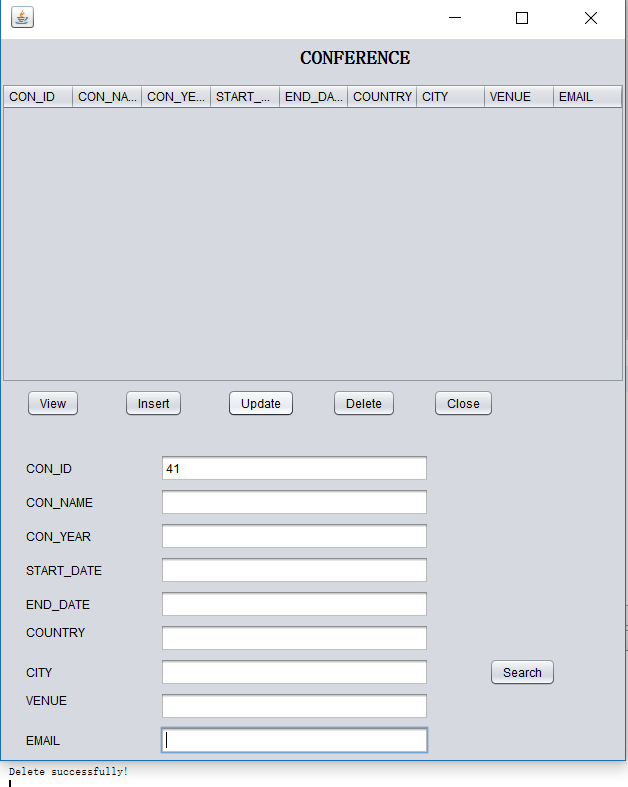
} catch (SQLException f) {

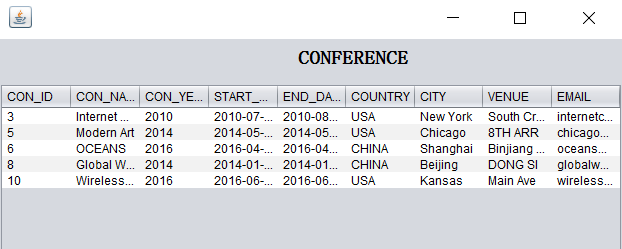
System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Delete**





private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int conid = Integer.parseInt(jTextField1.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn1.createStatement();

rset = stmt.executeQuery(

"delete from conference where con\_id ="+ conid);

System.out.println("Delete successfully!");

}

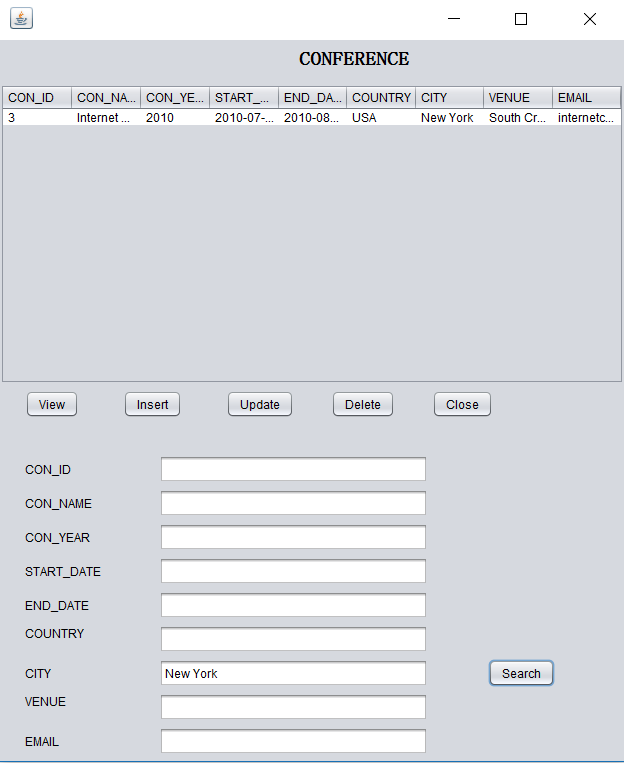
catch (SQLException f) {

System.out.println(f.getMessage());

}

}

**Search by city**



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String search = jTextField7.getText();

try {

stmt = ConferenceApp.conn1.createStatement();

rset = stmt.executeQuery

("select \* from conference where city = '" + search+"'");

System.out.println("search successful");

ResultSetMetaData mdata = rset.getMetaData();

dtm.setRowCount(0);

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

} catch (SQLException f) {

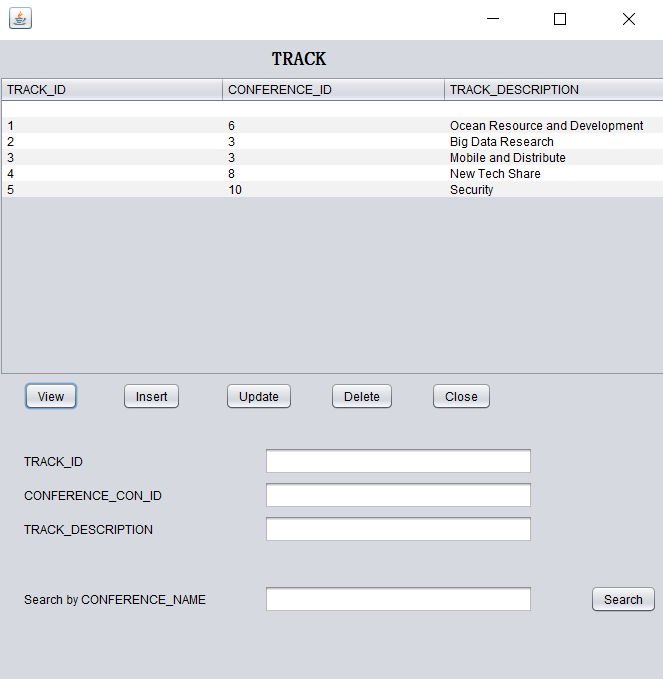
System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Track Frame**

**View**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

dtm.setRowCount(0);

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery("select \* from track");

ResultSetMetaData mdata = rset.getMetaData();

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

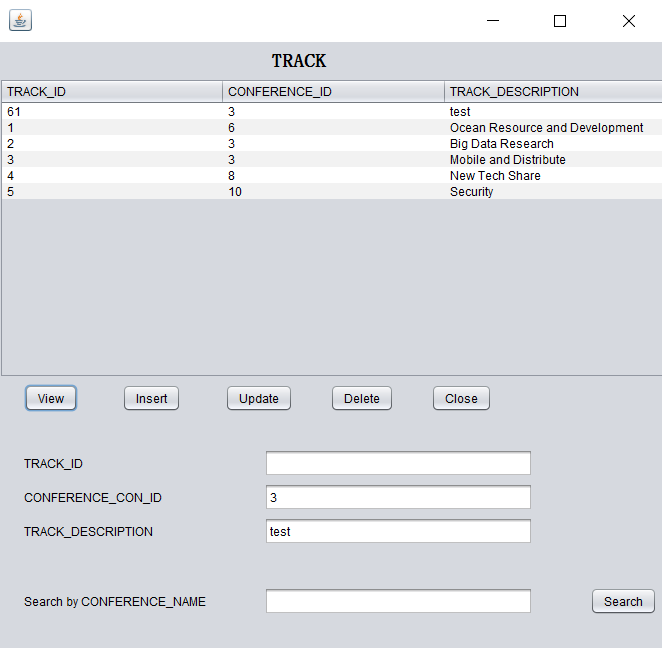
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Insert**



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

String conference\_con\_id = jTextField2.getText();

String track\_description = jTextField3.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"insert into track values(track\_id.nextval,'"+conference\_con\_id+"','"+track\_description+"')");

System.out.println("insert successfully!");

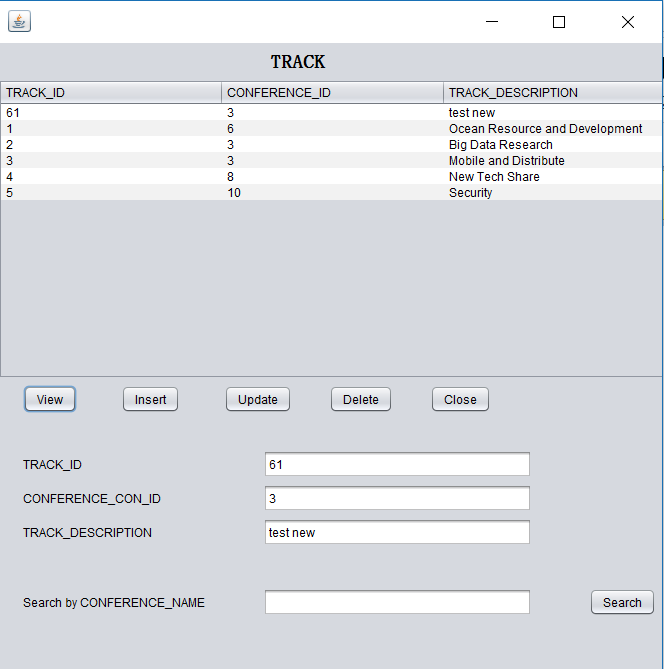
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Update**



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

int trackid = Integer.parseInt(jTextField1.getText());

String conference\_con\_id = jTextField2.getText();

String track\_description = jTextField3.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"update track set conference\_con\_id='"+conference\_con\_id+"',track\_description='"+track\_description+"' where track\_id="+trackid);

System.out.println("update successfully!");

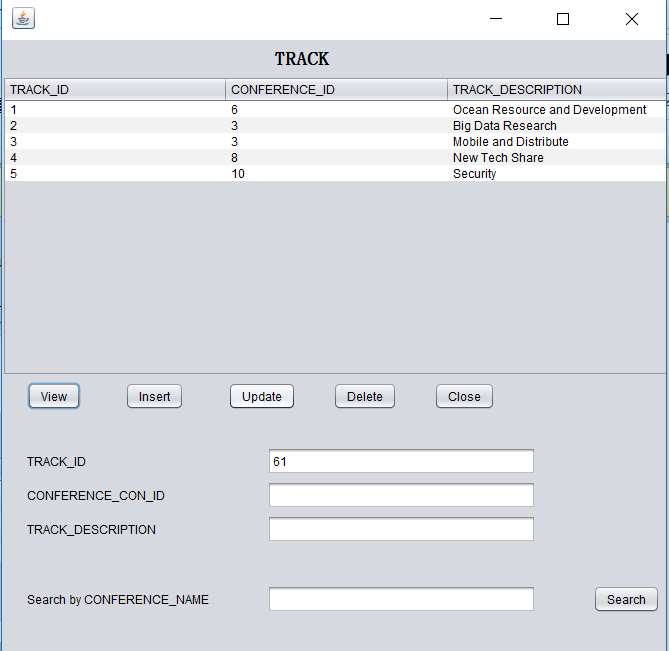
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Delete**



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int trackid = Integer.parseInt(jTextField1.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"delete from track where track\_id ="+ trackid);

System.out.println("Delete successfully!");

}

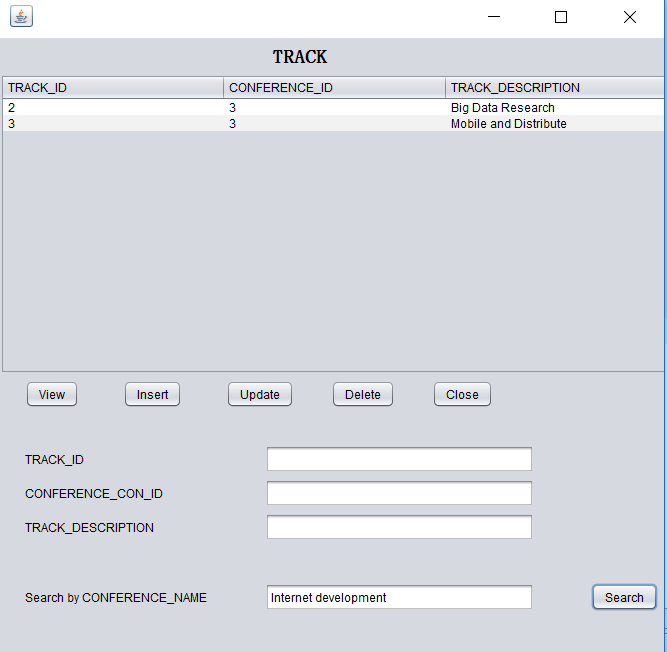
catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Search by conference’s name**



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String search = jTextField7.getText();

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery

("select \* from track t join conference@FIT5148A c on t.conference\_con\_id=c.con\_id where c.con\_name = '" + search+"'");

System.out.println("search successful");

ResultSetMetaData mdata = rset.getMetaData();

dtm.setRowCount(0);

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

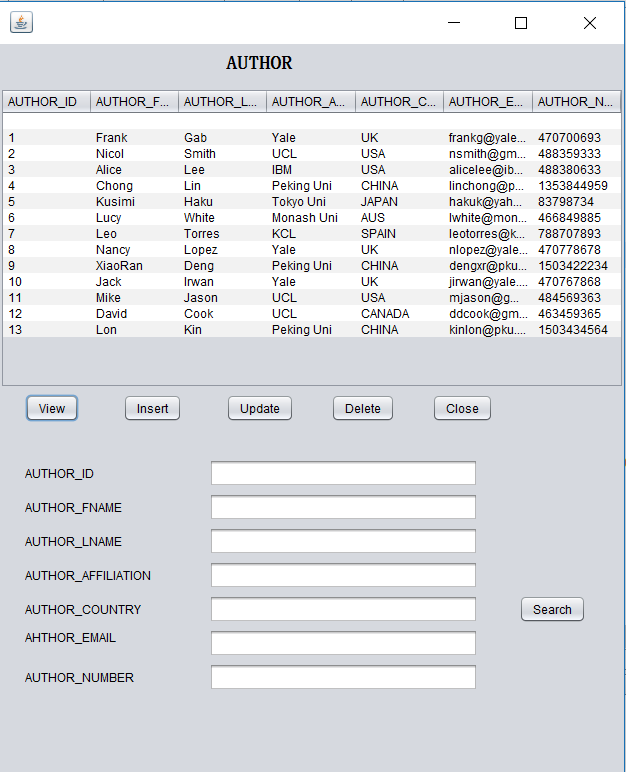
}

dtm.addRow(rowData);

}

**Author Frame**

**View**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

dtm.setRowCount(0);

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery("select \* from author");

ResultSetMetaData mdata = rset.getMetaData();

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

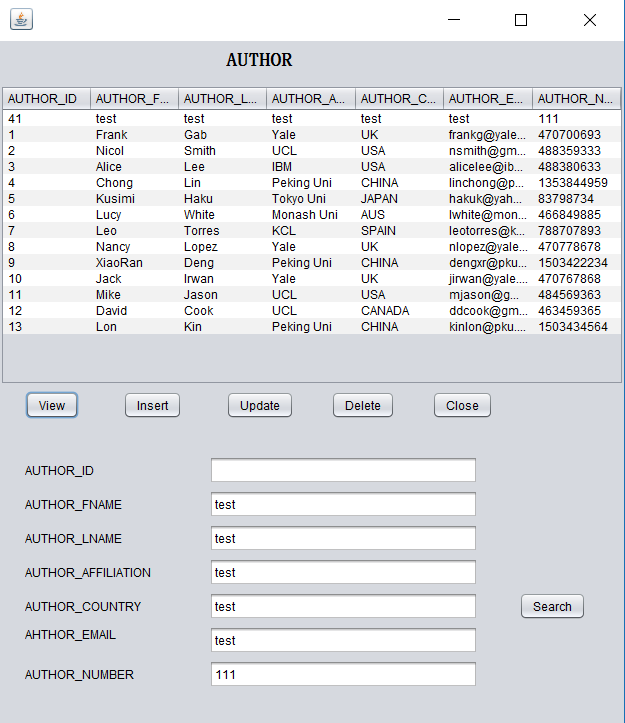
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Insert**



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

String fname = jTextField2.getText();

String lname = jTextField3.getText();

String affiliation= jTextField4.getText();

String country = jTextField5.getText();

String email = jTextField6.getText();

String number = jTextField7.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"insert into author values(author\_id.nextval,'"+fname+"','"+lname+"','"+affiliation+"','"+country+"','"+email+"','"+number+"')");

System.out.println("insert successfully!");

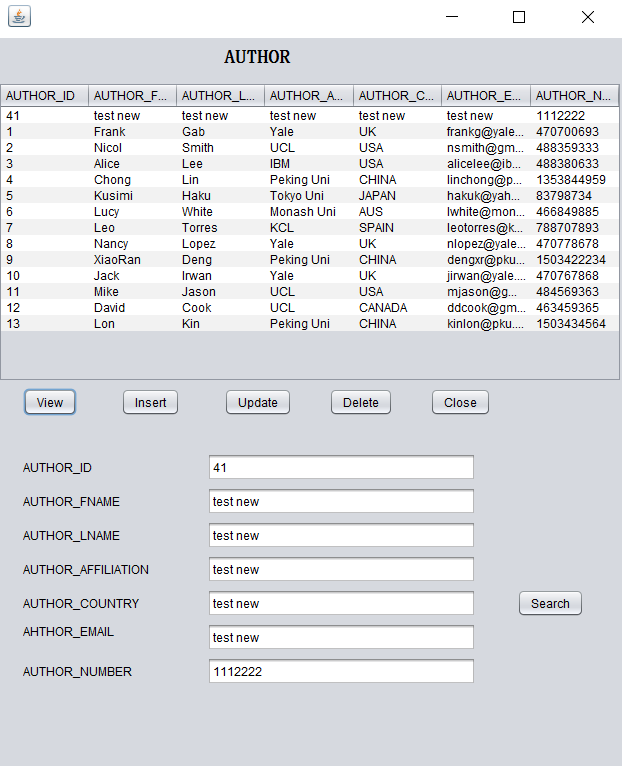
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Update**



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

int authorid = Integer.parseInt(jTextField1.getText());

String fname = jTextField2.getText();

String lname = jTextField3.getText();

String affiliation= jTextField4.getText();

String country = jTextField5.getText();

String email = jTextField6.getText();

String number = jTextField7.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"update author set author\_fname='"+fname+"',author\_lname='"+lname+"',author\_affiliation='"+affiliation+"',author\_country='"+country+"',"

+ "author\_email='"+email+"',author\_number='"+number+"'where author\_id="+authorid);

System.out.println("update successfully!");

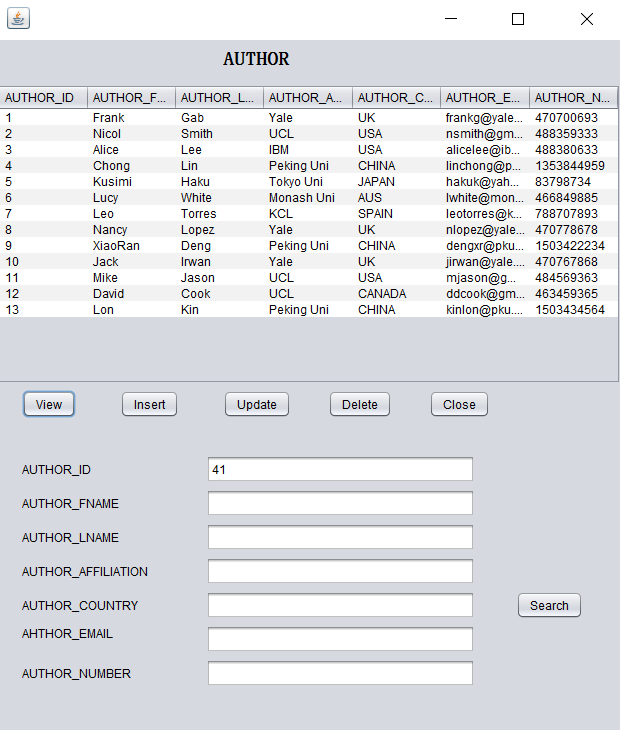
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Delete**



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int authorid = Integer.parseInt(jTextField1.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"delete from author where author\_id ="+ authorid);

System.out.println("Delete successfully!");

}

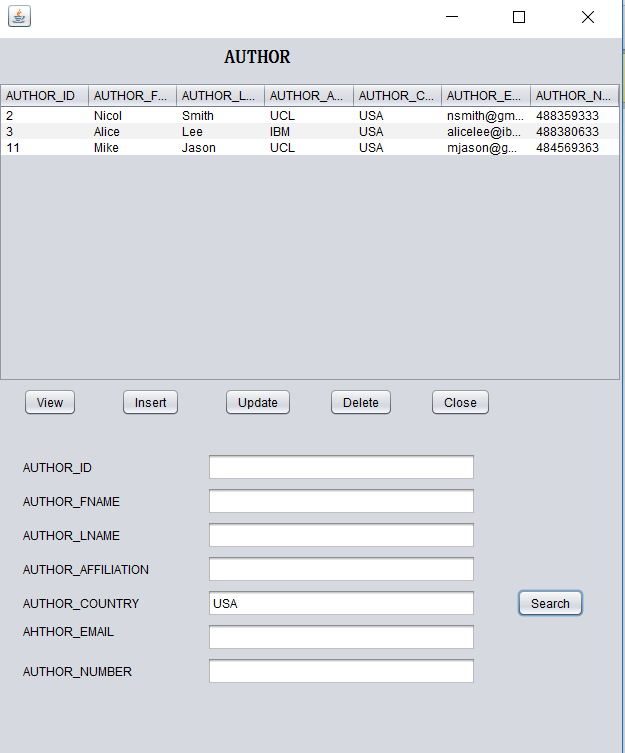
catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Search by country**



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String search = jTextField5.getText();

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery

("select \* from author where author\_country = '" + search+"'");

System.out.println("search successful");

ResultSetMetaData mdata = rset.getMetaData();

dtm.setRowCount(0);

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

} catch (SQLException f) {

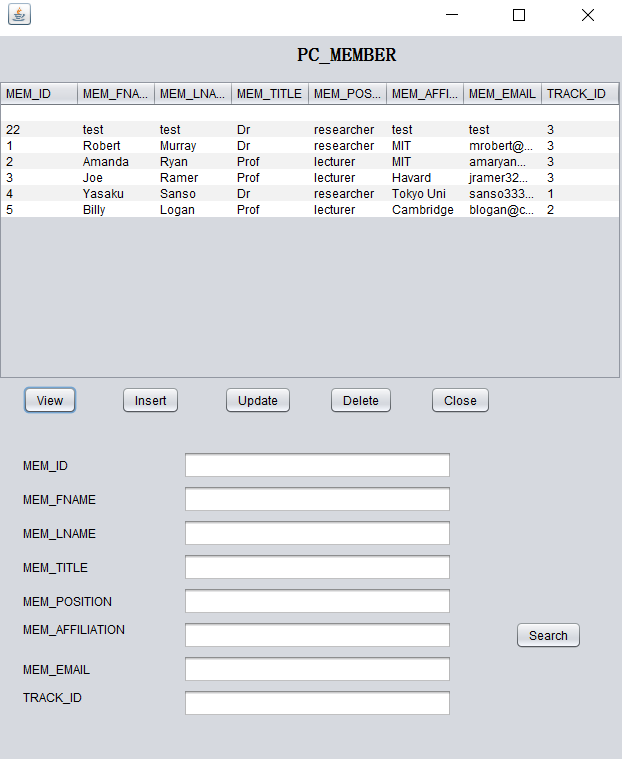
System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**PCmember Frame**

**View**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

dtm.setRowCount(0);

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery("select \* from pc\_member");

ResultSetMetaData mdata = rset.getMetaData();

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

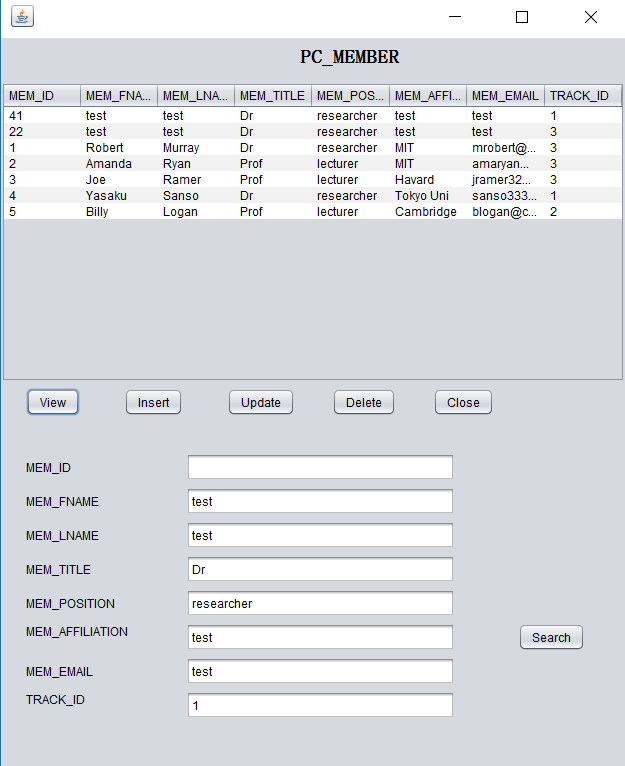
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Insert**



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

String fname = jTextField2.getText();

String lname = jTextField3.getText();

String title= jTextField4.getText();

String position = jTextField5.getText();

String affiliation = jTextField6.getText();

String email = jTextField7.getText();

String trackid = jTextField8.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"insert into pc\_member values(mem\_id.nextval,'"+fname+"','"+lname+"','"+title+"','"+position+"','"+affiliation+"','"+email+"','"+trackid+"')");

System.out.println("insert successfully!");

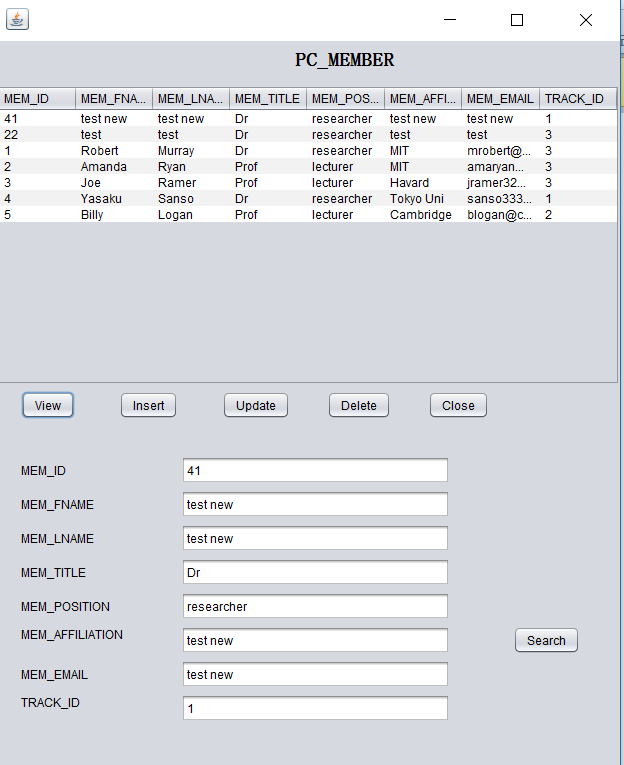
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Update**



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

int memid = Integer.parseInt(jTextField1.getText());

String fname = jTextField2.getText();

String lname = jTextField3.getText();

String title= jTextField4.getText();

String position = jTextField5.getText();

String affiliation = jTextField6.getText();

String email = jTextField7.getText();

String trackid = jTextField8.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"update pc\_member set mem\_fname='"+fname+"',mem\_lname='"+lname+"',mem\_title='"+title+"',mem\_position='"+position+"',"

+ "mem\_affiliation='"+affiliation+"',mem\_email='"+email+"',track\_track\_id='"+trackid+"' where mem\_id="+memid);

System.out.println("update successfully!");

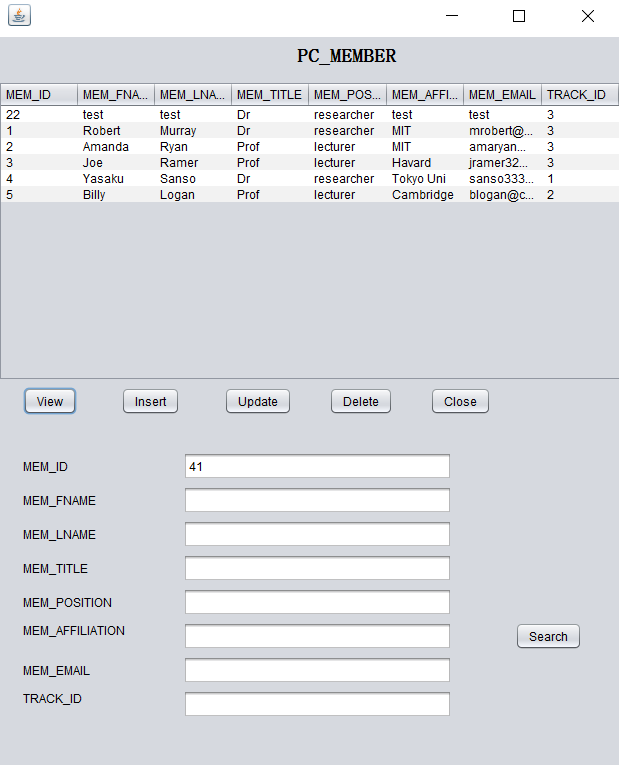
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Delete**



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int memid = Integer.parseInt(jTextField1.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"delete from pc\_member where mem\_id ="+ memid);

System.out.println("Delete successfully!");

}

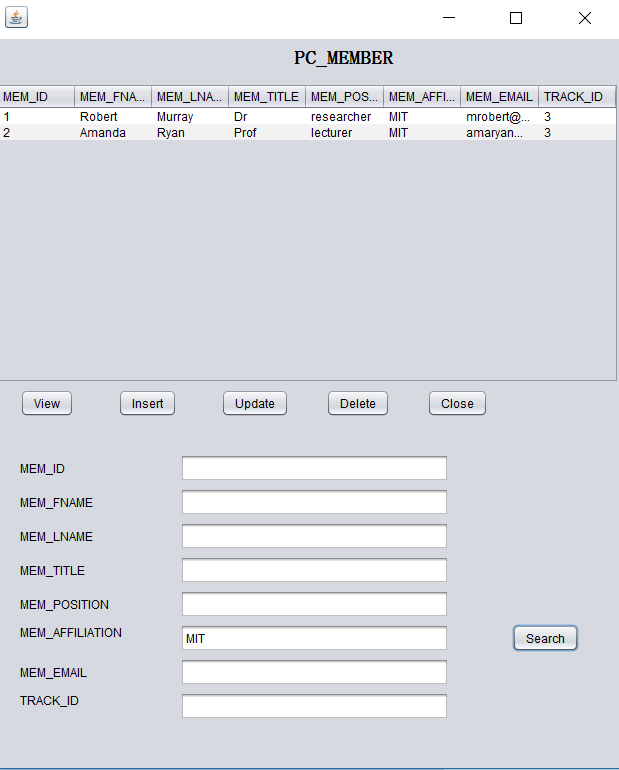
catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Search by affiliation**



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String search = jTextField6.getText();

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery

("select \* from pc\_member where mem\_affiliation = '" + search+"'");

System.out.println("search successful");

ResultSetMetaData mdata = rset.getMetaData();

dtm.setRowCount(0);

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

} catch (SQLException f) {

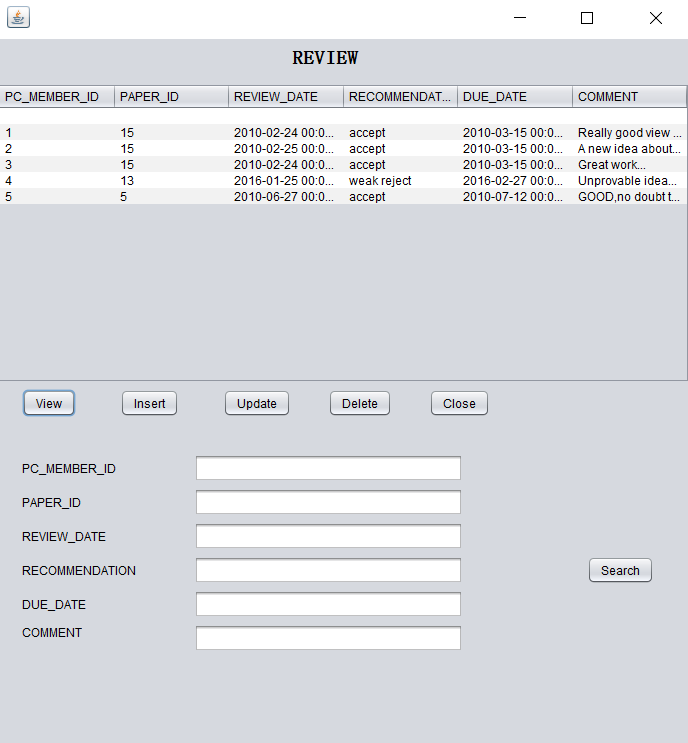
System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Review Frame**

**View**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

dtm.setRowCount(0);

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery("select \* from review");

ResultSetMetaData mdata = rset.getMetaData();

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

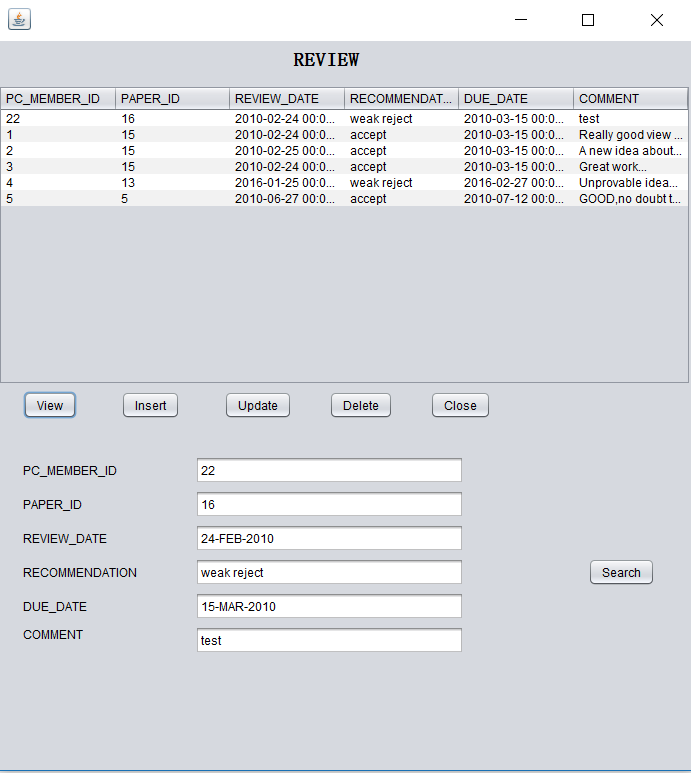
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Insert**



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

int pcid = Integer.parseInt(jTextField1.getText());

int paperid = Integer.parseInt(jTextField2.getText());

String reviewdate = jTextField3.getText();

String recommend= jTextField4.getText();

String duedate = jTextField5.getText();

String comment = jTextField6.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"insert into review values('"+pcid+"','"+paperid+"','"+reviewdate+"','"+recommend+"','"+duedate+"','"+comment+"')");

System.out.println("insert successfully!");

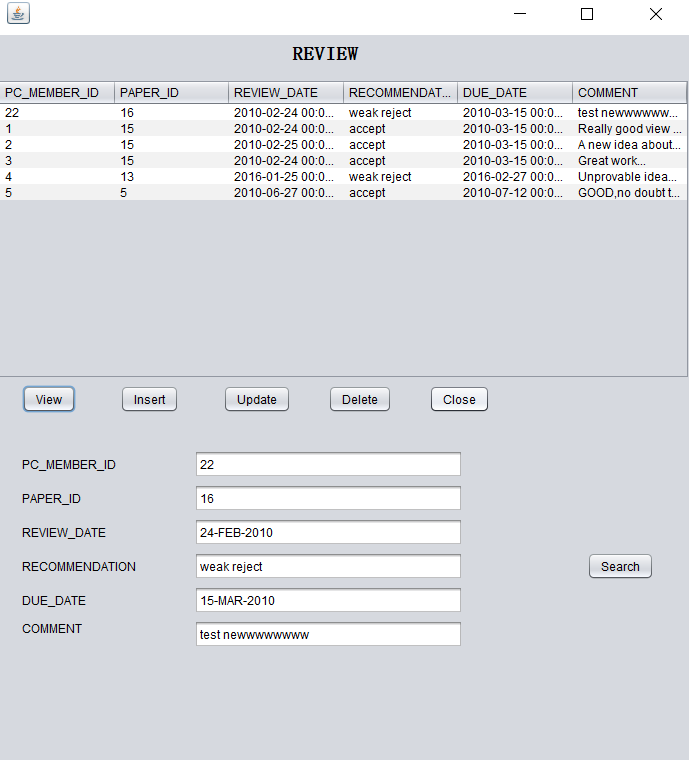
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Update**



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

int pcid = Integer.parseInt(jTextField1.getText());

int paperid = Integer.parseInt(jTextField2.getText());

String reviewdate = jTextField3.getText();

String recommend= jTextField4.getText();

String duedate = jTextField5.getText();

String comment = jTextField6.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"update review set review\_date='"+reviewdate+"',recommendation='"+recommend+"',due\_date='"+duedate+"',review\_comment='"+comment+"' where pc\_member\_mem\_id ="+ pcid +" and paper\_paper\_id="+paperid);

System.out.println("update successfully!");

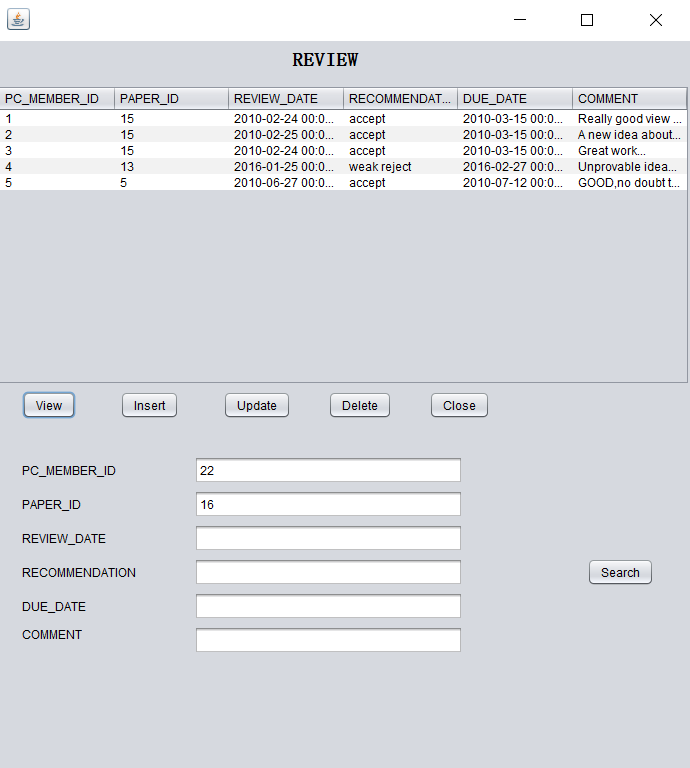
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Delete**



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int pcid = Integer.parseInt(jTextField1.getText());

int paperid = Integer.parseInt(jTextField2.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"delete from review where pc\_member\_mem\_id ="+ pcid +" and paper\_paper\_id="+paperid);

System.out.println("Delete successfully!");

}

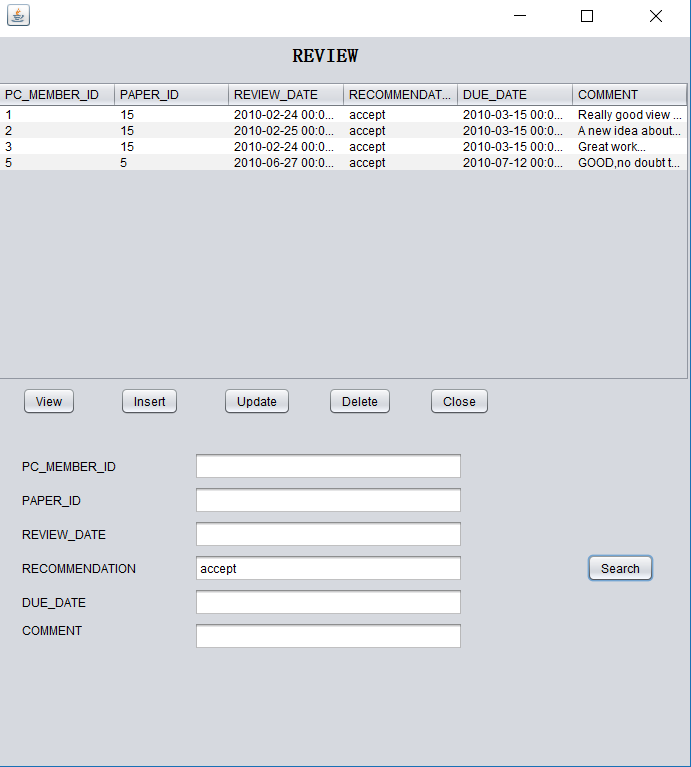
catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Search by recommendation**



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String search = jTextField4.getText();

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery

("select \* from review where recommendation = '" + search+"'");

System.out.println("search successful");

ResultSetMetaData mdata = rset.getMetaData();

dtm.setRowCount(0);

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

} catch (SQLException f) {

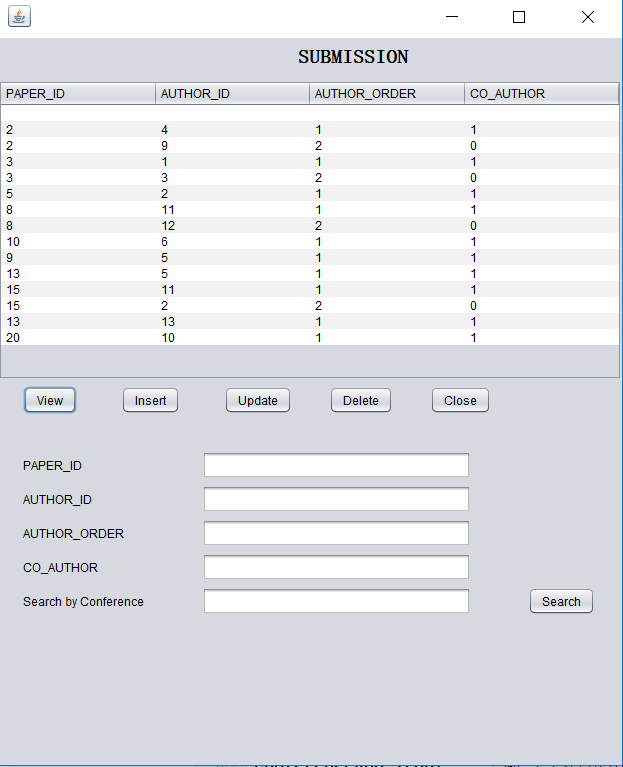
System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Submission Frame**

**View**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

dtm.setRowCount(0);

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery("select \* from submission");

ResultSetMetaData mdata = rset.getMetaData();

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

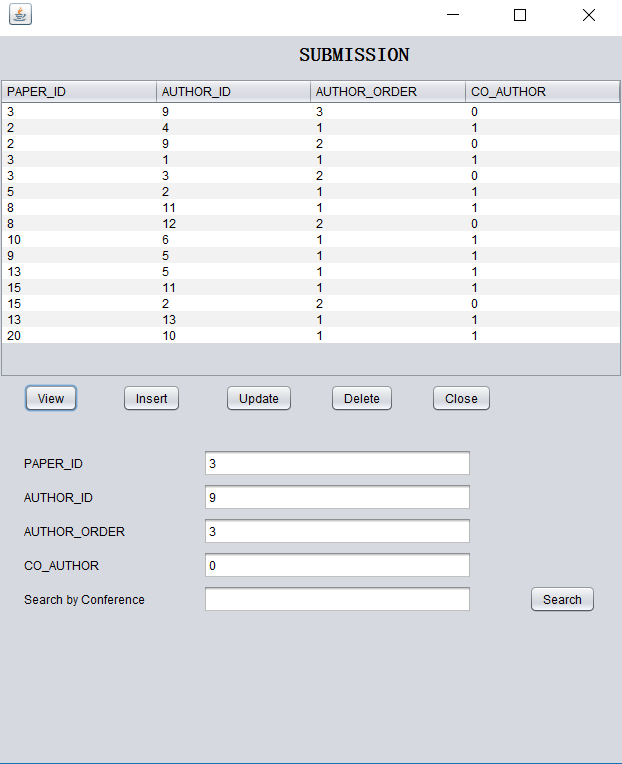
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Insert**



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

int paperid = Integer.parseInt(jTextField1.getText());

int authorid = Integer.parseInt(jTextField2.getText());

String order = jTextField3.getText();

String co\_author= jTextField4.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"insert into submission values('"+paperid+"','"+authorid+"','"+order+"','"+co\_author+"')");

System.out.println("insert successfully!");

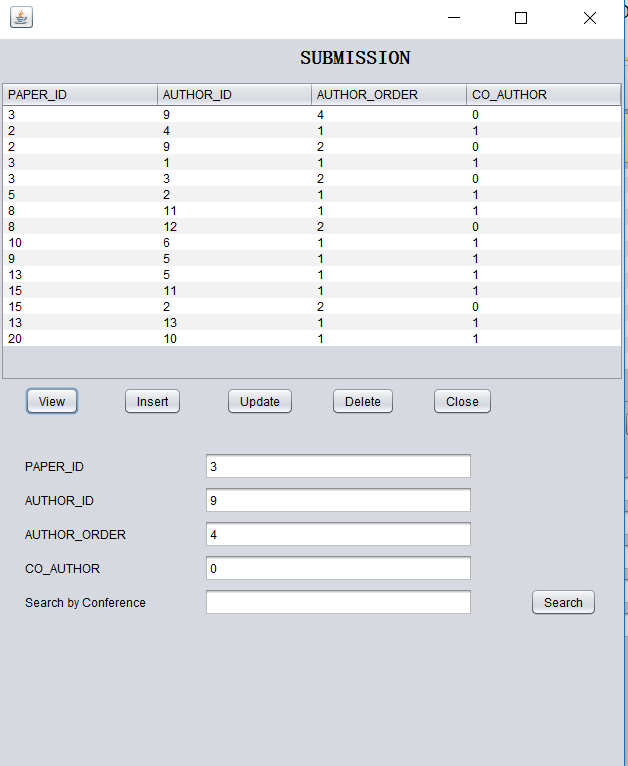
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Update**



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

int paperid = Integer.parseInt(jTextField1.getText());

int authorid = Integer.parseInt(jTextField2.getText());

String order = jTextField3.getText();

String co\_author= jTextField4.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"update submission set author\_order='"+order+"',co\_author='"+co\_author+"'where paper\_paper\_id="+paperid+" and author\_author\_id="+authorid);

System.out.println("update successfully!");

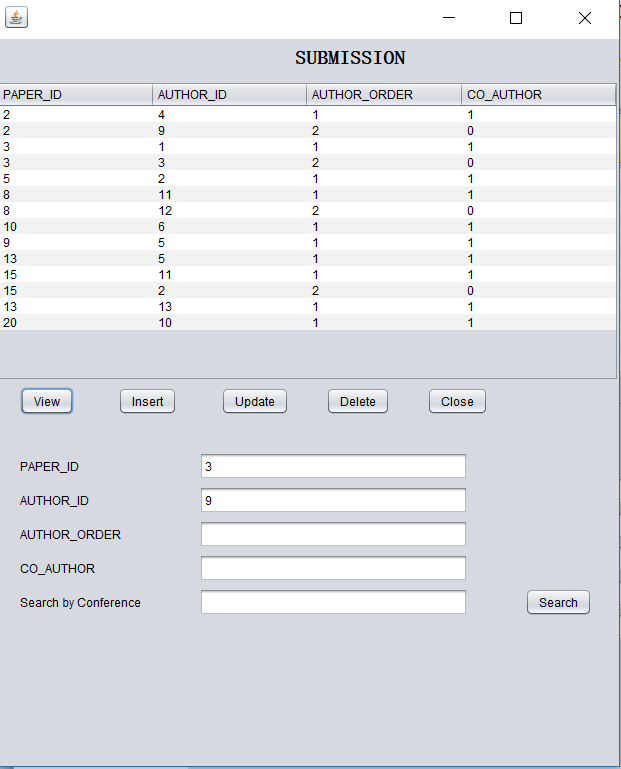
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Delete**



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int trackid = Integer.parseInt(jTextField1.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"delete from track where track\_id ="+ trackid);

System.out.println("Delete successfully!");

}

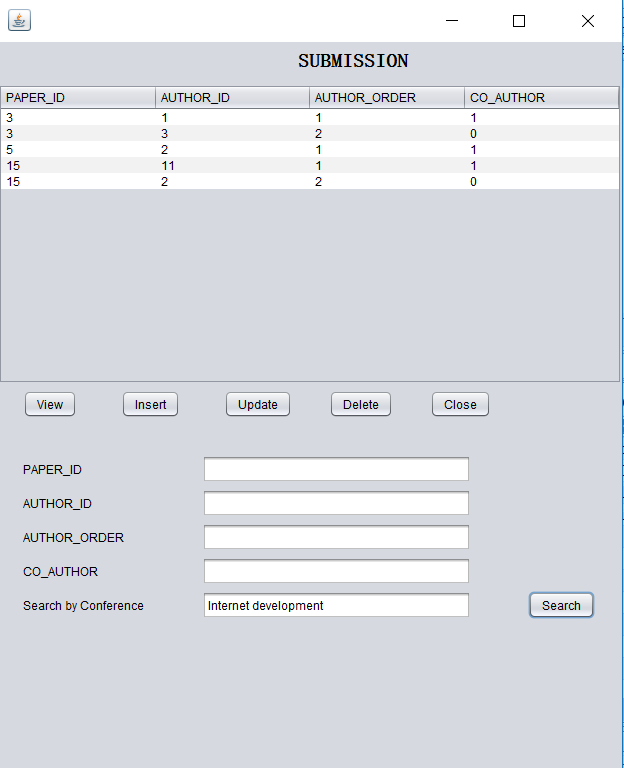
catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Search by conference name**



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String search = jTextField7.getText();

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery

("select \* from track t join conference@FIT5148A c on t.conference\_con\_id=c.con\_id where c.con\_name = '" + search+"'");

System.out.println("search successful");

ResultSetMetaData mdata = rset.getMetaData();

dtm.setRowCount(0);

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

} catch (SQLException f) {

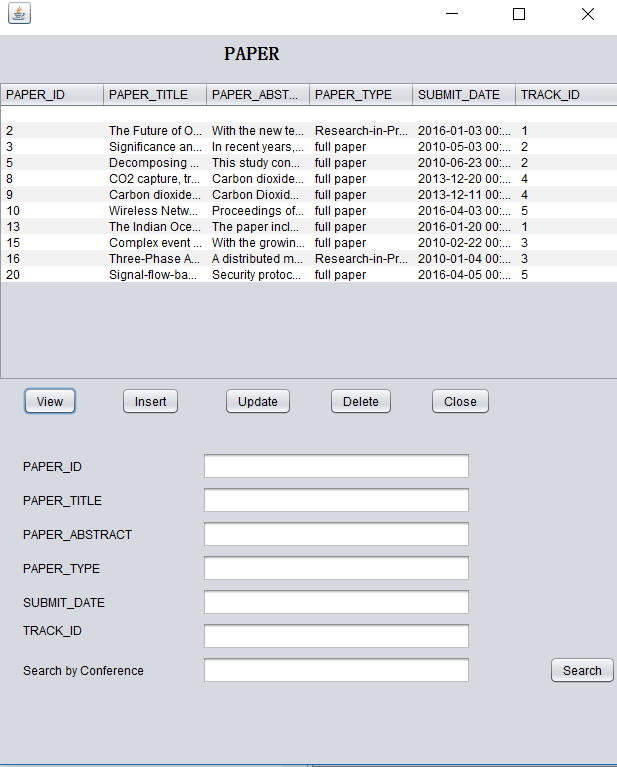
System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Paper Frame**

**View**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

dtm.setRowCount(0);

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery("select \* from paper");

ResultSetMetaData mdata = rset.getMetaData();

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

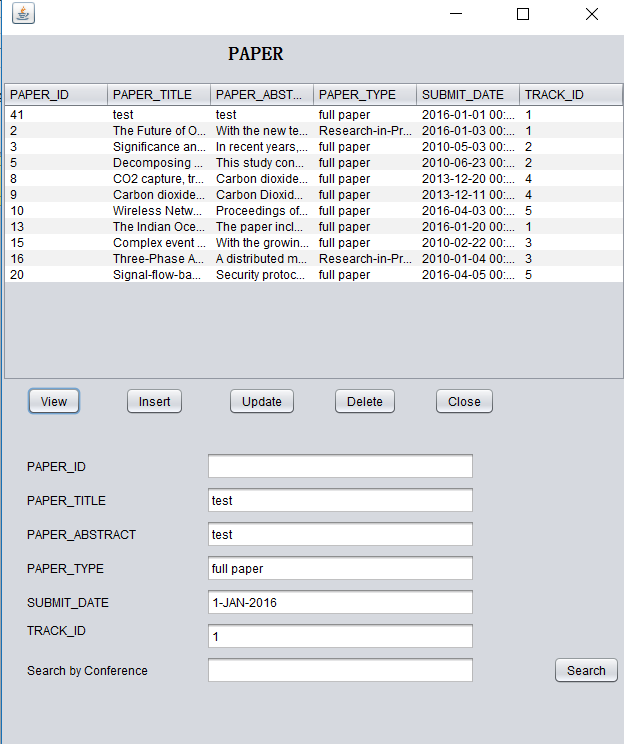
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Insert**



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

//int paperid = Integer.parseInt(jTextField1.getText());

String title = jTextField2.getText();

String paperabstract = jTextField3.getText();

String type= jTextField4.getText();

String subdate = jTextField5.getText();

String trackid = jTextField6.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"insert into paper values(paper\_id.nextval,'"+title+"','"+paperabstract+"','"+type+"','"+subdate+"','"+trackid+"')");

System.out.println("insert successfully!");

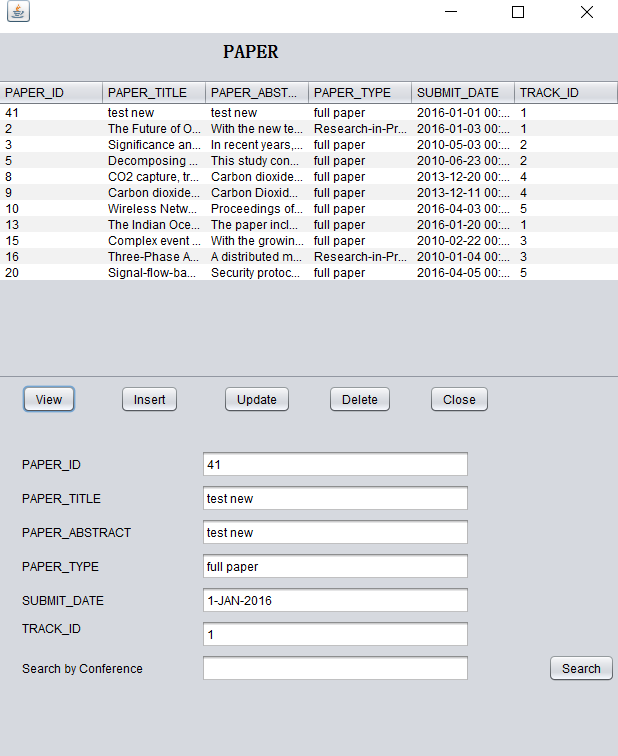
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Update**



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

int paperid = Integer.parseInt(jTextField1.getText());

String title = jTextField2.getText();

String paperabstract = jTextField3.getText();

String type= jTextField4.getText();

String subdate = jTextField5.getText();

String trackid = jTextField6.getText();

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"update paper set paper\_title='"+title+"',paper\_abstract='"+paperabstract+"',paper\_type='"+type+"',submit\_date='"+subdate+"',"

+ "track\_track\_id='"+trackid+"' where paper\_id="+paperid);

System.out.println("update successfully!");

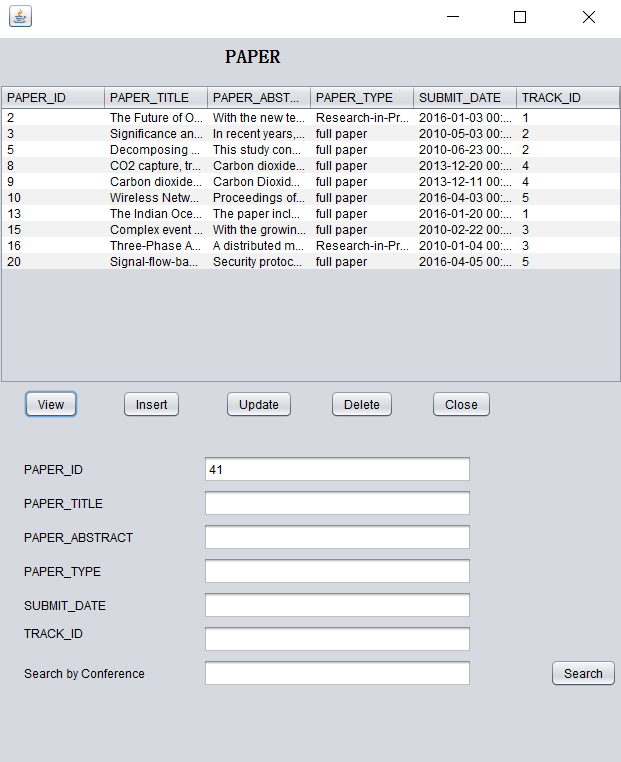
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Delete**



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int paperid = Integer.parseInt(jTextField1.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"delete from paper where paper\_id ="+ paperid);

System.out.println("Delete successfully!");

}

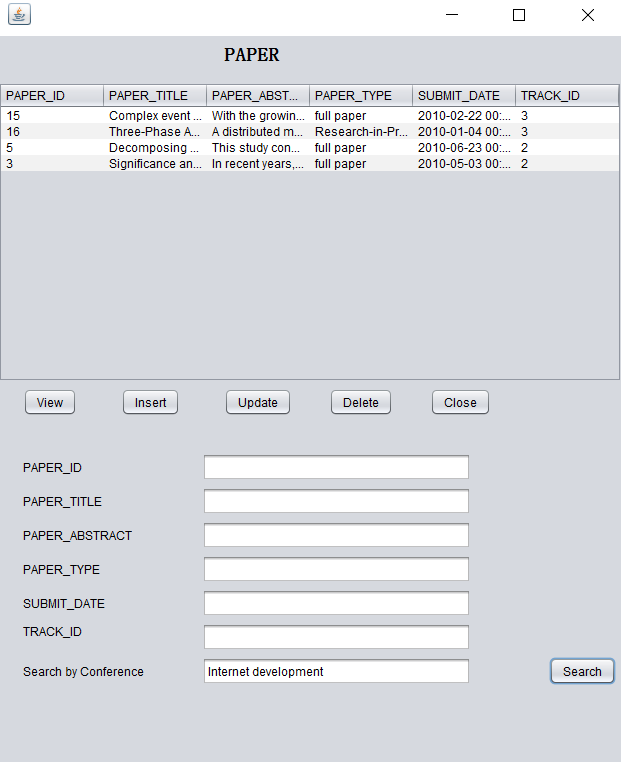
catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Search by conference name**



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String search = jTextField7.getText();

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery

("select \* from paper p join track@FIT5148B t on t.track\_id = p.track\_track\_id join conference@FIT5148A c on t.conference\_con\_id=c.con\_id where c.con\_name = '" + search+"'");

System.out.println("search successful");

ResultSetMetaData mdata = rset.getMetaData();

dtm.setRowCount(0);

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

} catch (SQLException f) {

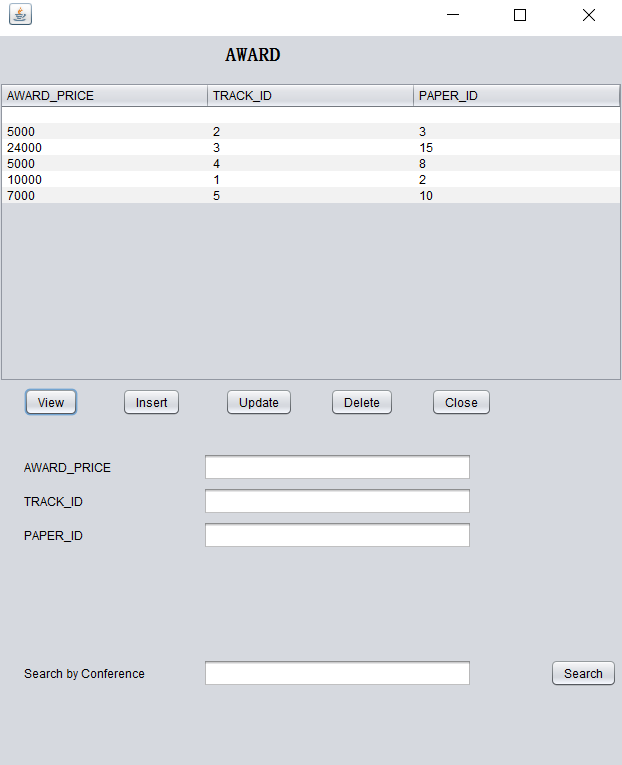
System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Award Frame**

**View**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

dtm.setRowCount(0);

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery("select \* from award");

ResultSetMetaData mdata = rset.getMetaData();

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

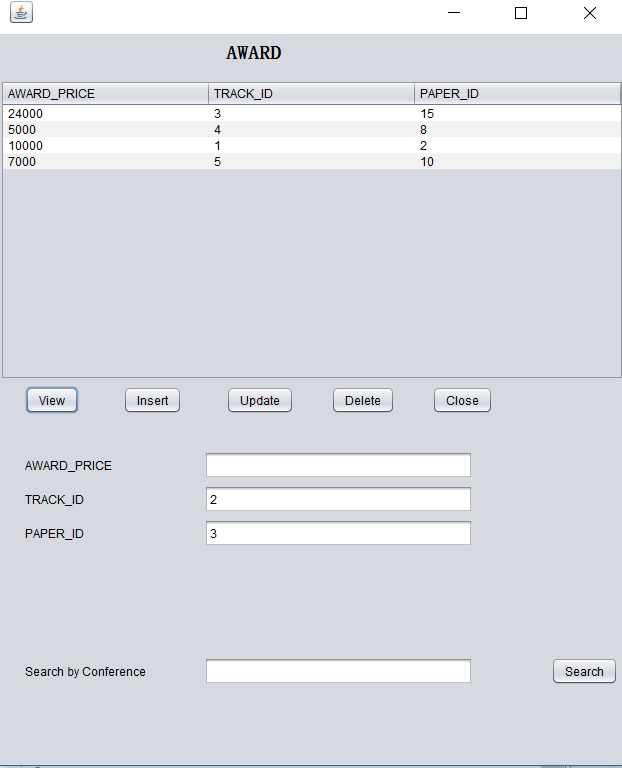
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Delete**



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

int trackid = Integer.parseInt(jTextField2.getText());

int paperid = Integer.parseInt(jTextField3.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"delete from award where track\_track\_id='"+trackid+"' and paper\_paper\_id='"+paperid+"'");

System.out.println("Delete successfully!");

}

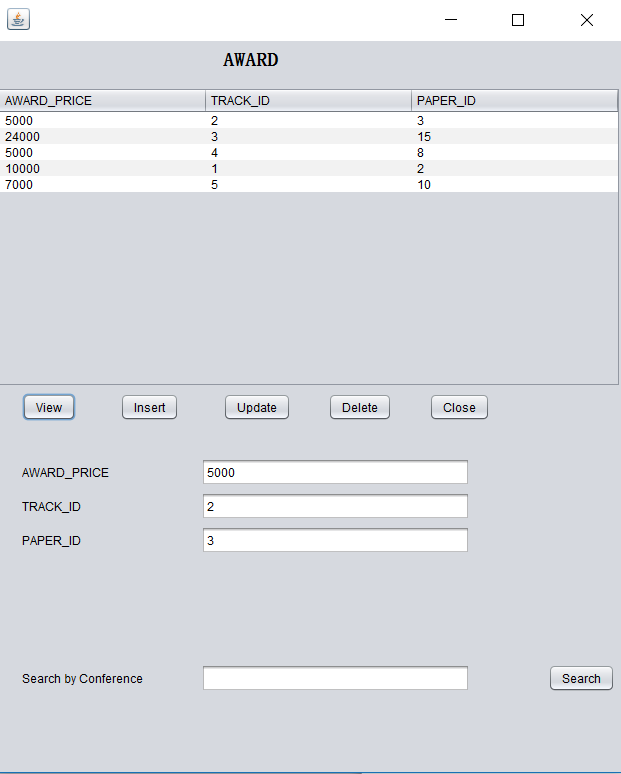
catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Insert**



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

int price = Integer.parseInt(jTextField1.getText());

int trackid = Integer.parseInt(jTextField2.getText());

int paperid = Integer.parseInt(jTextField3.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"insert into award values('"+price+"','"+trackid+"','"+paperid+"')");

System.out.println("insert successfully!");

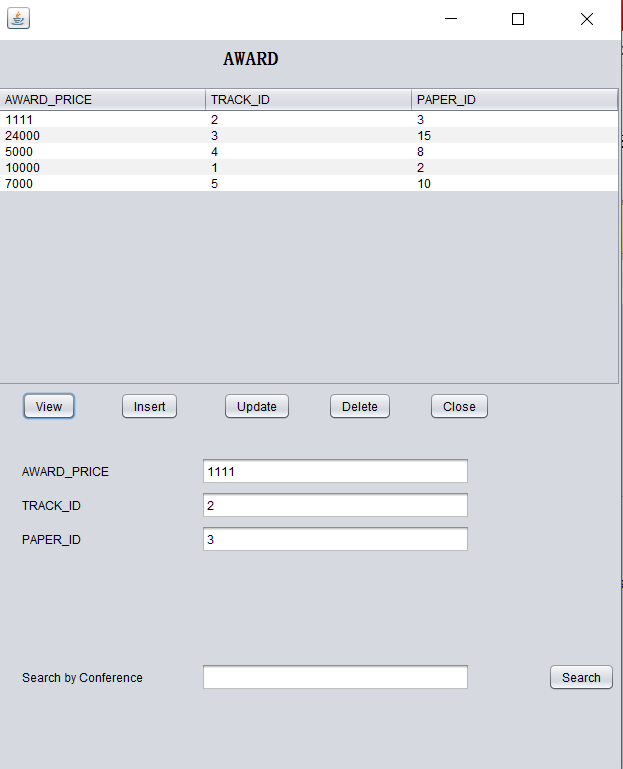
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Update**



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

int price = Integer.parseInt(jTextField1.getText());

int trackid = Integer.parseInt(jTextField2.getText());

int paperid = Integer.parseInt(jTextField3.getText());

dtm.setRowCount(0);

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery(

"update award set award\_price='"+price+"' where track\_track\_id='"+trackid+"' and paper\_paper\_id='"+paperid+"'");

System.out.println("update successfully!");

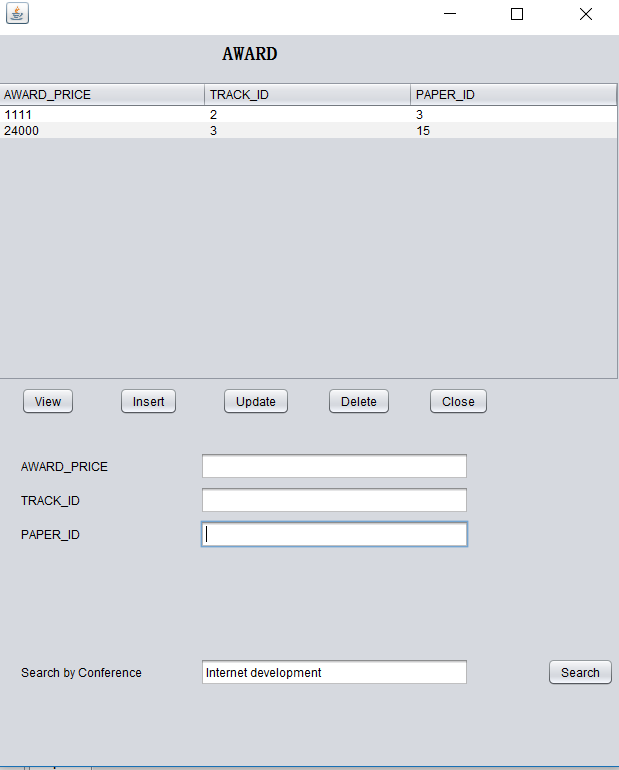
} catch (SQLException f) {

System.out.println(f.getMessage());

} // TODO add your handling code here:

}

**Search by conference name**



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String search = jTextField7.getText();

try {

stmt = ConferenceApp.conn2.createStatement();

rset = stmt.executeQuery

("select \* from award a join track@FIT5148B t on t.track\_id = a.track\_track\_id join conference@FIT5148A c on t.conference\_con\_id=c.con\_id where c.con\_name = '" + search+"'");

System.out.println("search successful");

ResultSetMetaData mdata = rset.getMetaData();

dtm.setRowCount(0);

int numberOfColumns = mdata.getColumnCount();

while (rset.next()) {

Object[] rowData = new Object[numberOfColumns];

for (int i = 0; i < rowData.length; i++) {

rowData[i] = rset.getObject(i + 1);

}

dtm.addRow(rowData);

}

} catch (SQLException f) {

System.out.println(f.getMessage());

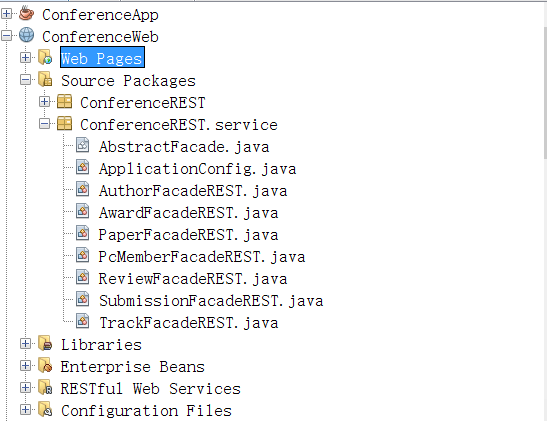
} // TODO add your handling code here:

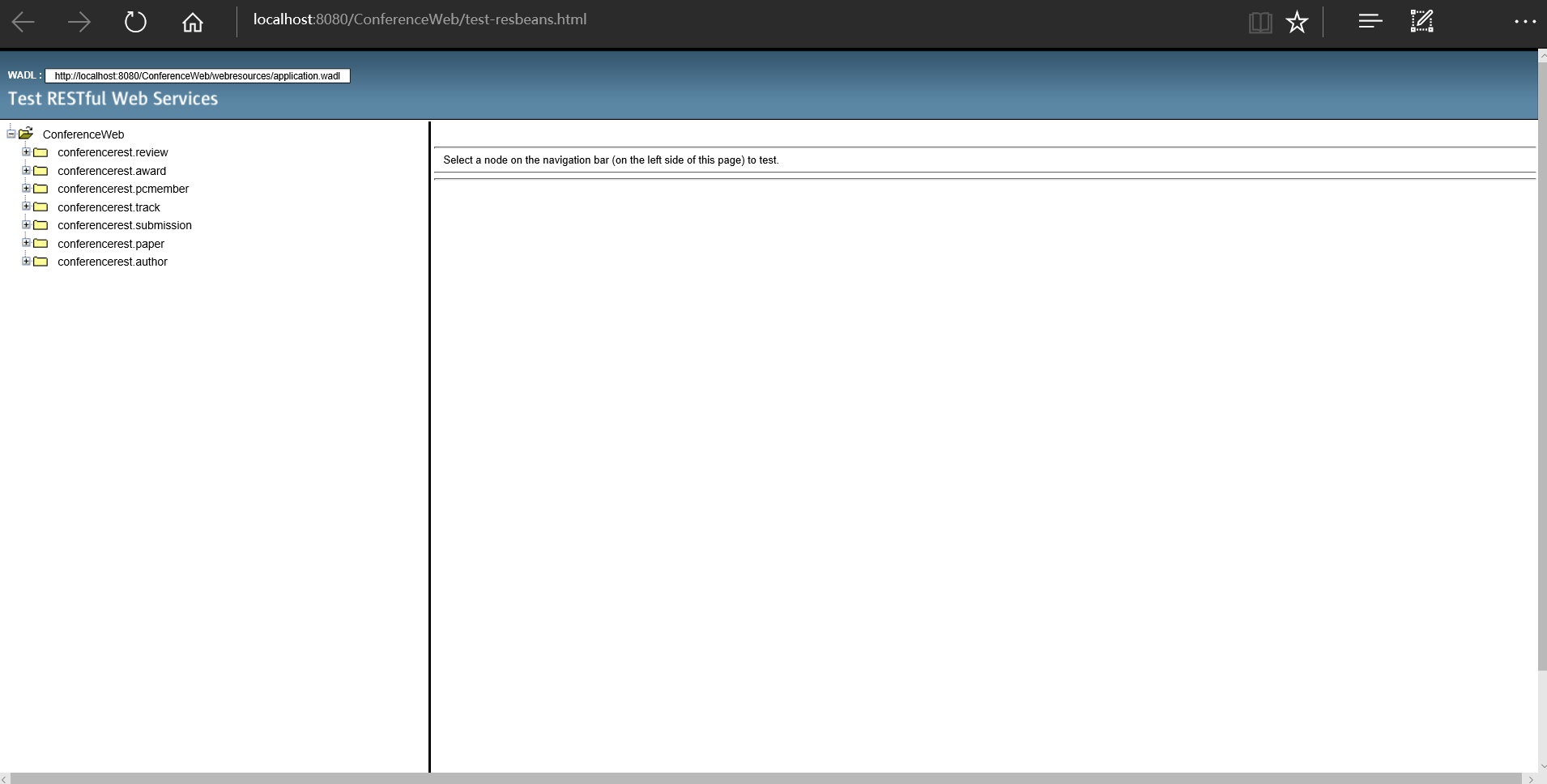
}

**Task6**

I’m also doing FIT5046Mobile and distributed computing system this semester. So I try to use RESTful service to connect from HTML web page to the databse FIT5148B

1. I create entity based on the database and RESTful service based on the entity





2. I create Javascript and index page

JavaScript:

var request = null;

var rooturl = "http://localhost:8080/ConferenceWeb/webresources/ConferenceREST.author";

function createRequest() {

try {

request = new XMLHttpRequest();

} catch (trymicrosoft) {

try {

request = new ActiveXObject("MsXML2.XMLHTTP");

} catch (othermicrosoft) {

try {

request = new ActiveXObject("Microsoft.XMLHTTP");

} catch (failed) {

request = null;

}

}

}

if (request == null)

alert("Error creating request object!");

}

function getAllAuthor()

{

document.getElementById('authroId').value = "";

document.getElementById('authorFname').value = "";

document.getElementById('authorLname').value = "";

document.getElementById('authorAffiliation').value = "";

document.getElementById('authorCountry').value = "";

document.getElementById('authorEmail').value = "";

document.getElementById('authorNumber').value = "";

createRequest();

request.onreadystatechange = handleListJSONResponse;

request.open("GET", rooturl, true);

request.send(null);

}

function getAuthor()

{

document.getElementById("output").innerHTML = "";

document.getElementById('authorFname').value = "";

document.getElementById('authorLname').value = "";

document.getElementById('authorAffiliation').value = "";

document.getElementById('authorCountry').value = "";

document.getElementById('authorEmail').value = "";

document.getElementById('authorNumber').value = "";

createRequest();

var id = document.getElementById("authorId").value;

var url = rooturl + "/" + id;

request.onreadystatechange = handleJSONResponse;

request.open("GET", url, true);

request.send(null);

}

function addAuthor()

{

document.getElementById("output").innerHTML = "";

var authorId = document.getElementById('authorId').value;

var authorFname = document.getElementById('authorFname').value;

var authorLname = document.getElementById('authorLname').value;

var authorAffiliation = document.getElementById('authorAffiliation').value;

var authorCountry = document.getElementById('authorCountry').value;

var authorEmail = document.getElementById('authorEmail').value;

var authorNumber = document.getElementById('authorNumber').value;

createRequest();

var data = JSON.stringify({"authorId": authorId.toString(),

"authorFname": authorFname.toString(),

"authorLname": authorLname.toString(),

"authorAffiliation": authorAffiliation.toString(),

"authorCountry": authorCountry.toString(),

"authorEmail": authorEmail.toString(),

"authorNumber": authorNumber.toString(),

});

request.open("POST", rooturl);

request.setRequestHeader('Content-type', 'application/json');

request.setRequestHeader("Content-Length", data.length);

request.setRequestHeader('Accept', 'application/json');

request.send(data);

document.getElementById('authorId').value = "";

document.getElementById('authorFname').value = "";

document.getElementById('authorLname').value = "";

document.getElementById('authorAffiliation').value = "";

document.getElementById('authorCountry').value = "";

document.getElementById('authorEmail').value = "";

document.getElementById('authorNumber').value = "";

}

function deleteAuthor()

{

document.getElementById("output").innerHTML = "";

createRequest();

var id = document.getElementById("authorId").value;

var url = rooturl + "/" + id;

request.open("DELETE", url, true);

request.send(null);

document.getElementById('authorId').value = "";

document.getElementById('authorFname').value = "";

document.getElementById('authorLname').value = "";

document.getElementById('authorAffiliation').value = "";

document.getElementById('authorCountry').value = "";

document.getElementById('authorEmail').value = "";

document.getElementById('authorNumber').value = "";

}

function updateAuthor()

{

document.getElementById("output").innerHTML = "";

var authorId = document.getElementById('authorId').value;

var authorFname = document.getElementById('authorFname').value;

var authorLname = document.getElementById('authorLname').value;

var authorAffiliation = document.getElementById('authorAffiliation').value;

var authorCountry = document.getElementById('authorCountry').value;

var authorEmail = document.getElementById('authorEmail').value;

var authorNumber = document.getElementById('authorNumber').value;

createRequest();

var data = JSON.stringify({"authorId": authorId.toString(),

"authorFname": authorFname.toString(),

"authorLname": authorLname.toString(),

"authorAffiliation": authorAffiliation.toString(),

"authorCountry": authorCountry.toString(),

"authorEmail": authorEmail.toString(),

"authorNumber": authorNumber.toString()});

var url = rooturl + "/" + docId;

request.open("PUT", url);

request.setRequestHeader('Content-type', 'application/json');

request.setRequestHeader("Content-Length", data.length);

request.setRequestHeader('Accept', 'application/json');

request.send(data);

document.getElementById('authorId').value = "";

document.getElementById('authorFname').value = "";

document.getElementById('authorLname').value = "";

document.getElementById('authorAffiliation').value = "";

document.getElementById('authorCountry').value = "";

document.getElementById('authorEmail').value = "";

document.getElementById('authorNumber').value = "";

}

function handleJSONResponse() {

if (request.readyState == 4 && request.status == 200)

{

var det = JSON.parse(request.responseText);

document.getElementById('authorId').value = det.authorId;

document.getElementById('authorFname').value = det.authorFname;

document.getElementById('authorLname').value = det.authorLname;

document.getElementById('authorAffiliation').value = det.authorAffiliation;

document.getElementById('authorCountry').value = det.authorCountry;

document.getElementById('authorEmail').value = det.authorEmail;

document.getElementById('authorNumber').value = det.authorNumber;

}

}

function handleListJSONResponse() {

if (request.readyState == 4 && request.status == 200)

{

var det = JSON.parse(request.responseText);

var output = "<table border = '1'>";

output += "<tr>" + "<th>authorId</th>" + "<th>authorFname</th>" + "<th>authorLname</th>"

+ "<th>authorAffiliation</th>" + "<th>authorCountry</th>" + "<th>authorEmail</th>" + "<th>authorNumber</th>"

+ "</tr>";

for (var i = 0; i < det.length; i++) {

output += "<tr>";

output += "<td>" + det[i].authorId + "</td>";

output += "<td>" + det[i].authorFname + "</td>";

output += "<td>" + det[i].authorLname + "</td>";

output += "<td>" + det[i].authorAffiliation + "</td>";

output += "<td>" + det[i].authorCountry + "</td>";

output += "<td>" + det[i].authorEmail + "</td>";

output += "<td>" + det[i].authorNumber + "</td>";

output += "</tr>";

}

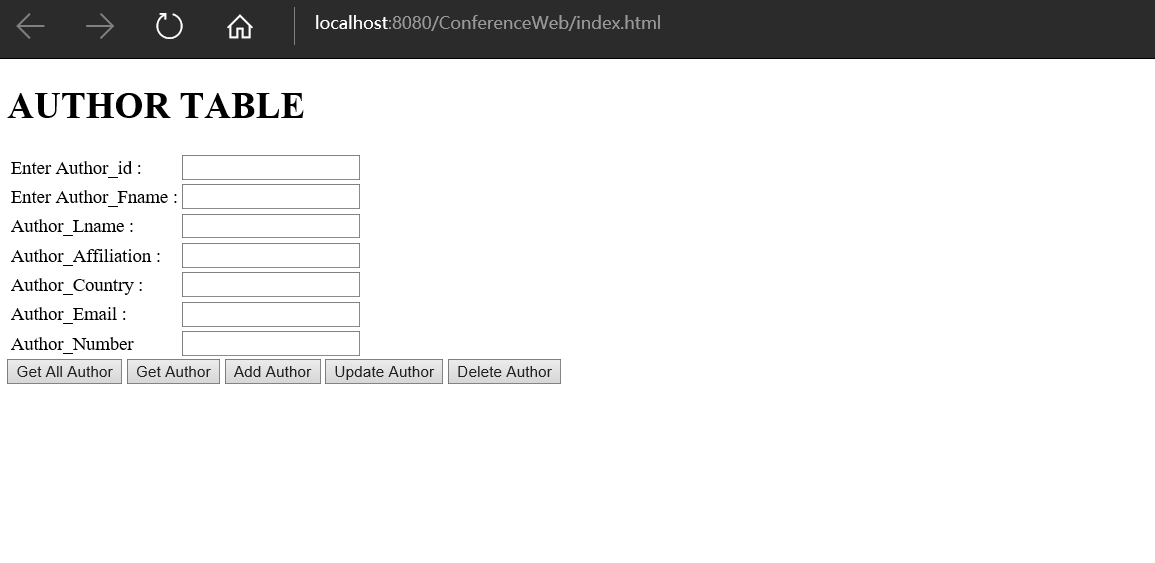
output += "</table>";

document.getElementById("output").innerHTML = output;

}

}

**Index web page**



<!DOCTYPE html>

<html>

<head>

<title>Doctor Service</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src="js/ConferenceService.js"></script>

</head>

<body>

<h1> AUTHOR TABLE</h1>

<table>

<tr>

<td>Enter Author\_id : </td>

<td><input type="text" id="authroId" size="20"/></td>

</tr>

<tr>

<td>Enter Author\_Fname : </td>

<td><input type="text" id="authorFname" size="20"/></td>

</tr>

<tr>

<td>Author\_Lname : </td>

<td> <input type="text" id="authorLname" size="20"/> </td>

</tr>

<tr>

<td>Author\_Affiliation : </td>

<td> <input type="text" id="authorAffiliation" size="20"/></td>

</tr>

<tr>

<td>Author\_Country : </td>

<td> <input type="text" id="authorCountry" size="20"/></td>

</tr>

<tr>

<td>Author\_Email : </td>

<td> <input type="text" id="authorEmail" size="20"/></td>

</tr>

<tr>

<td>Author\_Number </td>

<td> <input type="text" id="authorNumber" size="20"/></td>

</tr>

</table>

<input type="button" value="Get All Author" onclick="getAllAuthor()"/>

<input type="button" value="Get Author" onclick="getAuthor()"/>

<input type="button" value="Add Author" onclick="addAuthor()"/>

<input type="button" value="Update Author" onclick="updateAuthor()"/>

<input type="button" value="Delete Author" onclick="deleteAuthor()"/>

<p id="output"></p>

</body>

</html>