

ASSESSMENT COVER SHEET

Student ID number 27148572	Unit Name and Code: Distributed and big data processing FIT5148 Campus: Caulfield Assignment Title: Practical Assignment 1 Name of Lecturer: Dr Pari Delir Haghghi Name of Tutor: Yong Kong Tutorial Day and Time: Wednesday (12am to 2 pm) Phone Number: 0416378674 Email Address: Jspar18@student.monash.edu Has any part of this assignment been previously submitted as part of another unit/course? <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Given Name Jayesh	Due Date: 08/04/2016	Date Submitted: 08/04/2016	All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor. Extension granted until (date) _____ Signature of lecturer/tutor _____ Please note that it is your responsibility to retain copies of your assessments.
Family name Parab	<p><i>Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations</i></p> <p>Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).</p> <p>Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.</p> <p>Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.</p> <p>Student Statement:</p> <ul style="list-style-type: none"> • I have read the university's Student Academic Integrity Policy and Procedures. • I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes • have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied. • No part of this assignment has been previously submitted as part of another unit/course. • I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and: <ul style="list-style-type: none"> i. provide to another member of faculty and any external marker; and/or ii. submit it to a text matching software; and/or iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking. • I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment. <p>SignatureJayesh Parab..... <u>Date.....08/04/2016.....</u> * delete (iii) if not applicable</p>		

The information on this form is collected for the primary purpose of assessing your assignment and ensuring the academic integrity requirements of the University are met. Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au

Contents

1.	Main Page	1
2.	Task 1 -- Database.....	2
1.	Create Tables:	2
1.	Code for creation of Conference table in Database FIT5148A:	2
2.	Code for creation of Track table in Database FIT5148B:	2
3.	Code for creation of PCMember table in Database FIT5148B.....	2
4.	Code for creation of Paper table in Database FIT5148B:.....	3
5.	Code for creation of Review table in Database FIT5148B:	3
6.	Code for creation of Author table in Database FIT5148B:.....	3
7.	Code for creation of BestPaper table in Database FIT5148B:.....	4
8.	Code for creation of Submission table in Database FIT5148B:.....	4
2.	Entity Relationship Diagram:	4
3.	Create Sequences:	5
9.	Code for sequence of Conference table in Database FIT5148A:	5
10.	Code for sequence of Track table in Database FIT5148B:	5
11.	Code for sequence of PCMember table in Database FIT5148B:.....	5
12.	Code for sequence of Paper table in Database FIT5148B:.....	6
13.	Code for sequence of Review table in Database FIT5148B:	6
14.	Code for sequence of Author table in Database FIT5148B:.....	6
15.	Code for sequence of BestPaper table in Database FIT5148B:.....	6
16.	Code for sequence of Submission table in Database FIT5148B.....	6
4.	Create indexes:	6
5.	Addition of meaningful data:	6
17.	Code for Inserting data in Conference table in Database FIT5148A:.....	6
18.	Code for Inserting data in Track table in Database FIT5148B.....	7
19.	Code for Inserting data in PCMember table in Database FIT5148B	7
20.	Code for Inserting data in Paper table in Database FIT5148B	8
21.	Code for Inserting data in Review table in Database FIT5148B	8
22.	Code for Inserting data in Author table in Database FIT5148B	9
23.	Code for Inserting data in BestPaper table in Database FIT5148B	9
24.	Code for Inserting data in Submission table in Database FIT5148B	9

3.	Task 2 -- Stored Procedure.....	9
6.	Code for updatevenue:.....	9
4.	Task 3 – Triggers.....	12
7.	Code for Conference_Insert_Trigger:	12
8.	Code for Conference_Update_Trigger:.....	15
9.	Code for Conference_Delete_Trigger:	17
5.	Task 4 – Fragmentation.....	19
10.	Vertical fragmentation.....	19
11.	Horizontal Fragmentation.....	22
6.	Task 5 – User Interface.....	24
12.	Login Page	24
25.	Code for checking login credentials:.....	24
26.	Code for Cancel Button:.....	25
13.	Main Page:	25
14.	Conference Frame:.....	28
27.	Code to view the content of Conference table:.....	29
28.	Code to reset the user interface value to null:	30
29.	Code to go back on the Main Page:	30
30.	Code to insert row in Conference table:.....	31
31.	Code to update row in Conference table:.....	33
32.	Code to delete row in Conference table:.....	35
33.	Code to select a row in Conference table by clicking on it:	37
34.	Code for selecting rows by Conference Name:.....	38
15.	Track Frame:	39
35.	Code to view the content of Track table:.....	40
36.	Code to reset the user interface value to null:	42
37.	Code to insert row in Track table:.....	42
38.	Code to update row in Track table:	45
39.	Code to delete row in Track table:.....	48
40.	Code to select a row in Track table by clicking on it:	51
41.	Code for selecting rows by Conference Name:.....	53
16.	Author Frame:	55
42.	Code to view the content of Author table:	55

43.	Code to reset the user interface value to null:	56
44.	Code to insert row in Author table:	56
45.	Code to update row in Author table:.....	59
46.	Code to delete row in Author table:	61
47.	Code to select a row in Author table by clicking on it:	63
48.	Code for selecting rows by Author Country:.....	64
17.	PCMMember Frame	66
49.	Code to view the content of PCMMember table:.....	66
50.	Code to reset the user interface value to null:	67
51.	Code to insert row in PCMMember table:	68
52.	Code to update row in PCMMember table:	70
53.	Code to delete row in PCMMember table:	72
54.	Code to select a row in PCMMember table by clicking on it:	75
55.	Code for selecting rows by PCMMember Affiliation:.....	76
18.	Review Frame.....	77
56.	Code to view the content of Review table:.....	78
57.	Code to reset the user interface value to null:	79
58.	Code to insert row in Review table:.....	80
59.	Code to update row in Review table:.....	82
60.	Code to delete row in Review table:.....	84
61.	Code for selecting rows by Review Recommendations:.....	87
19.	Submission Frame:.....	88
62.	89
63.	Code to view the content of Submission table:.....	89
64.	Code to reset the user interface value to null:	91
65.	Code to insert row in Submission table:	91
66.	Code to update row in Submission table:	94
67.	Code to delete row in Submission table:	97
68.	Code to select a row in Submission table by clicking on it:	100
69.	Code for selecting rows by Conference ID:.....	102
20.	Paper Frame:.....	103
70.	Code to view the content of Paper table:.....	104
71.	Code to reset the user interface value to null:	106

72.	Code to insert row in Paper table:	106
73.	Code to update row in Paper table:.....	108
74.	Code to delete row in Paper table:.....	111
75.	Code to select a row in Paper table by clicking on it:	114
76.	Code for selecting row by Conference ID:	116
21.	BestPaper Frame:.....	118
77.	Code to view the content of BestPaper table:.....	118
78.	Code to reset the user interface value to null:	120
79.	Code to insert row in BestPaper table:	120
80.	Code to update row in BestPaper table:.....	123
81.	Code to delete row in BestPaper table:.....	126
82.	Code to select a row in BestPaper table by clicking on it:	129
83.	Code for selecting rows by Conference ID:.....	131
7.	Task 6 – Web Based GUI Client	133
22.	Index.jsp:.....	133
84.	Code for Index.jsp:.....	133
23.	View_Option.jsp:.....	134
85.	Code for View_Option.jsp:.....	134
24.	Insert_Option.jsp:	136
86.	Code for Insert_Option.jsp:	136
25.	Update_Option.jsp:.....	138
87.	Code for Update_Option.jsp:	138
26.	Delete_Option.jsp:	140
88.	Code for Delete_Option.jsp:	140
27.	Insert.jsp:	141
89.	Code for the page:	141
28.	Update.jsp:.....	144
90.	Code for the page:	144
29.	Delete.jsp:	146
91.	Code for the page:	146

1. Main Page

- Task 1 of creating database: (Completed)
 - In this task I have created a database consisting of 8 tables. Tables are as follow Conference, Track, PCMember, Author, Review, Submission, Paper and Best Paper. Conference table is in database FIT5148A while other tables are in FIT5148B.
 - Created an ER Diagram , sequences, indexes and added meaningful data.
- Task 2 Oracle Stored Procedure: (Completed)
 - Completed the stored procedure with screen shots to update the venue attribute in the Conference table.
- Task 3 Triggers: (Completed)
 - Insert trigger: This trigger is stored in Database FIT5148B. It checks whether a corresponding Conf_id is present in Conference table before inserting it in Track table. If the Conf_Id which is not present in Conference table is been tried to enter in Track table this trigger will generate an error.
 - Update Trigger: This trigger is stored in Database FIT5148B. It checks whether a corresponding Conf_id is present in Conference table before updating it in Track table. If the Conf_Id which is not present in Conference table is been tried to update in Track table this trigger will generate an error.
 - Delete Trigger: This trigger is stored in Database FIT5148B. Before deleting rows in Conference table it will delete all the corresponding rows with same Conf_ID in Track table.
- Task 4 Fragmentation: (Completed)
 - Performed Vertical Fragmentation on Conference table. Provided four queries, Usage Matrix, Access Frequency Matrix. I have choose Conference table as many search fields in database FIT5148B use Conf_id to search.
 - Performed Horizontal Fragmentation on Best Paper Table. Provided list of predicates and minterm predicates.
- Task 5 User Interface: (Completed)
 - In User Interface created 10 frames Login frame (which checks credential), MainWindow (from where you can select any table) and remaining 8 frames for 8 tables.
 - In every table frame you can view table content, insert new row, update existing row, delete existing row, reset fields, go back to MainWindow and search table by the given clause.
 - In every table frame once you click on any row all the values will populate in text boxes and combo boxes.
 - In Paper frame I have made Title unique and for track combo box only papers in that track are visible.
 - In Review frame I have added a constraint that a paper can be reviewed only thrice.
 - I have some text field that will accept only numbers and give beep sound when other things are inserted.
- Task 5 Web : Web based Client: (Completed)
 - Index Page: Main Page which has all the operations that can be performed on Author table.
 - Delete_Option, Insert_Option, Update_Option and View_Option JSP pages contains structure of websites.s

- Insert, update and delete JSP pages contain SQL codes

2. Task 1 -- Database

1. Create Tables:

1. Code for creation of Conference table in Database FIT5148A:

```
Create table Conference(
    Conf_ID number(6,0) not null,
    Conf_name varchar(50),
    Conf_year number(4,0),
    Conf_start_date date,
    Conf_end_date date,
    Conf_country varchar(50),
    Conf_city varchar(50),
    Conf_venue varchar(50),
    Contact_email varchar(50),
    constraint conference_pk Primary key(Conf_ID));
```

2. Code for creation of Track table in Database FIT5148B:

```
Create table Track(
    Track_ID number(6,0) not null,
    Track_name varchar(50),
    Track_description varchar(50),
    Conf_ID number(6,0),
    Constraint Track_PK Primary key(Track_ID));
```

3. Code for creation of PCMember table in Database FIT5148B

```
Create table PCMember(
    Member_ID number(6,0) not null,
    First_name varchar(50),
    Surname varchar(50),
    Member_title varchar(50),
    Member_position varchar(50),
    Member_affiliation varchar(50),
    Member_email varchar(50),
    Track_ID number(6,0),
    Constraint Check_Member_Title check (Member_title in ('DR','Prof','None')),
    Constraint Check_Member_position check (Member_position in
    ('Lecturer','Researcher','Student')),
    Constraint PCMember_PK Primary Key(Member_ID),
```

```
Constraint PCMember_Track_FK Foreign Key (Track_ID) REFERENCES
Track(Track_ID)
on delete cascade);
```

4. Code for creation of Paper table in Database FIT5148B:

```
Create table Paper(
Title_ID number(6,0) not null,
Title varchar(50),
Abstract varchar(50),
Paper_type varchar(50),
Submission_date date,
Track_ID number(6,0),
Constraint Paper_PK Primary Key(Title_ID),
Constraint Check_Paper_Type check (Paper_type in ('Full
Paper','Research_in_progress','Poster')),
Constraint Paper_Track_FK Foreign Key (Track_ID) REFERENCES
ConferenceTrack(Track_ID)
on delete cascade);
```

5. Code for creation of Review table in Database FIT5148B:

```
Create table Review(
Review_ID number(6,0) not null,
Due_Date date,
Reviewed_date date,
Recommendation varchar(50),
Paper_comment varchar(50),
Title_ID number(6,0),
Member_ID number(6,0),
Constraint Review_PK Primary Key(Review_ID),
Constraint Check_Recommendation check (Recommendation in
('Accept','Weakly Accept','Borderline','Weakly Reject','Reject')),
Constraint Review_Paper_FK Foreign Key (Title_ID) REFERENCES
Paper(Title_ID) on delete cascade,
Constraint Review_Member_FK Foreign Key (Member_ID) REFERENCES
PCMember(Member_ID) on delete cascade);
```

6. Code for creation of Author table in Database FIT5148B:

```
Create table Author(
Author_ID number(6,0) not null,
Author_name varchar(50),
```

```
Author_Surname varchar(50),  
Author_Affiliation varchar(50),  
Author_Country varchar(50),  
Author_email varchar(50),  
Author_number number(10,0),  
Constraint Author_PK Primary Key(Author_ID));
```

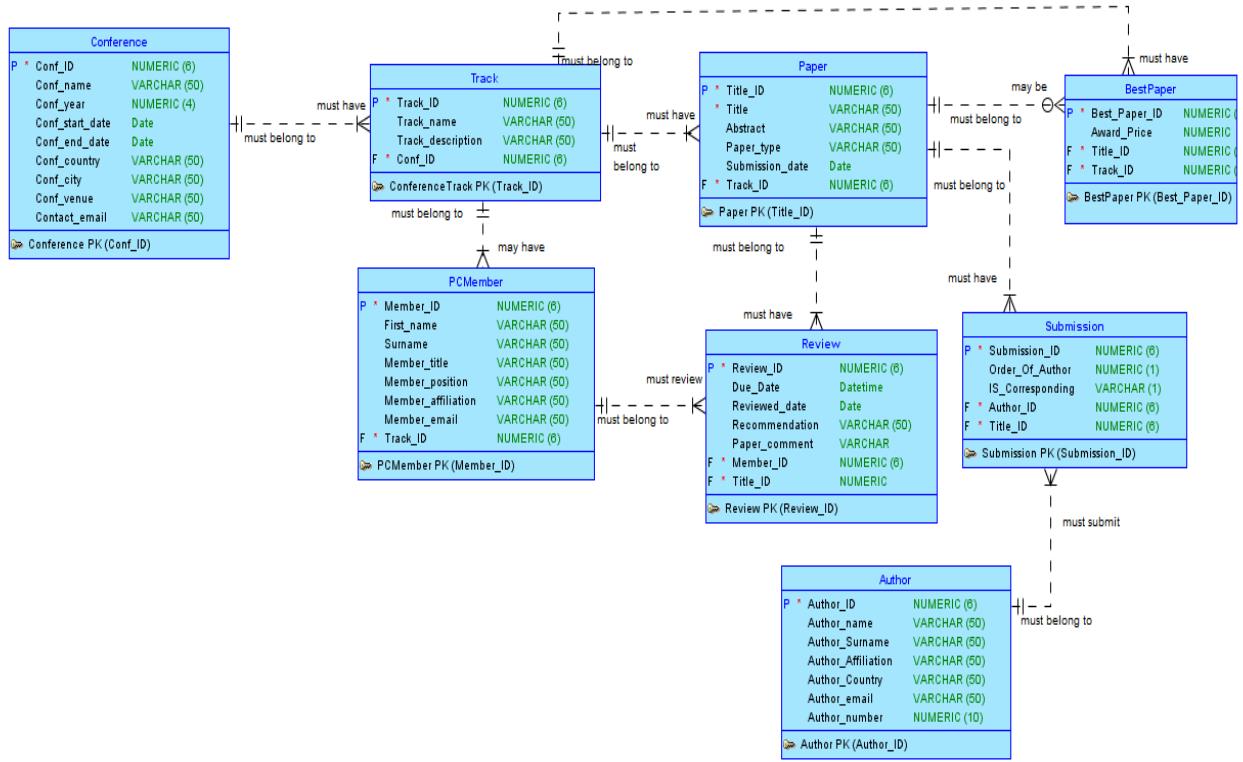
7. Code for creation of BestPaper table in Database FIT5148B:

```
Create table BestPaper(  
Best_Paper_ID number(6,0) not null,  
Title_ID number(6,0),  
Award_Price number(8,0),  
Track_ID number(6,0),  
Constraint BestPaper_PK Primary Key(Best_Paper_ID),  
Constraint BestPaper_Paper_FK Foreign Key (Title_ID) REFERENCES  
Paper(Title_ID) on delete cascade,  
Constraint BestPaper_Track_FK Foreign Key (Track_ID) REFERENCES  
ConferenceTrack(Track_ID)on delete cascade);
```

8. Code for creation of Submission table in Database FIT5148B:

```
Create table Submission(  
Submission_ID number(6,0),  
Title_ID number(6,0),  
Author_ID number(6,0),  
Order_Of_Author number(1,0),  
IS_Corresponding varchar(1),  
Constraint Check_Corresponding check (IS_Corresponding in ('Y','N')),  
Constraint Submission_PK Primary Key(Submission_ID),  
Constraint Submission_Paper_FK Foreign Key (Title_ID) REFERENCES  
Paper(Title_ID) on delete cascade,  
Constraint Submission_Author_FK Foreign Key (Author_ID) REFERENCES  
Author(Author_ID)on delete cascade);
```

2. Entity Relationship Diagram:



3. Create Sequences:

9. Code for sequence of Conference table in Database FIT5148A:

```
CREATE SEQUENCE Conference_seq
START WITH 1
INCREMENT BY 1;
```

10. Code for sequence of Track table in Database FIT5148B:

```
CREATE SEQUENCE track_seq
START WITH 1
INCREMENT BY 1;
```

11. Code for sequence of PCMember table in Database FIT5148B:

```
CREATE SEQUENCE Member_seq
START WITH 1
INCREMENT BY 1;
```

12. Code for sequence of Paper table in Database FIT5148B:

```
CREATE SEQUENCE Paper_seq  
START WITH 1  
INCREMENT BY 1;
```

13. Code for sequence of Review table in Database FIT5148B:

```
CREATE SEQUENCE Review_seq  
START WITH 1  
INCREMENT BY 1;
```

14. Code for sequence of Author table in Database FIT5148B:

```
CREATE SEQUENCE Author_seq  
START WITH 1  
INCREMENT BY 1;
```

15. Code for sequence of BestPaper table in Database FIT5148B:

```
CREATE SEQUENCE Best_Paper_seq  
START WITH 1  
INCREMENT BY 1;
```

16. Code for sequence of Submission table in Database FIT5148B

```
CREATE SEQUENCE Submission_seq  
START WITH 1  
INCREMENT BY 1;
```

4. Create indexes:

- Code for index:

```
Create index track_index  
on ConferenceTrack(Track_name);
```

5. Addition of meaningful data:

17. Code for Inserting data in Conference table in Database FIT5148A:

```
Insert into Conference values(Conference_seq.nextval,'Melbourne Conference',2008,
```

```
to_date('23-sep-2008','DD-MM-YYYY'), to_date('24-sep-2008','DD-MM-YYYY'),'Australia','Melbourne','Federation Square','Melconf@gmail.com');
```

```
Insert into Conference values(Conference_seq.nextval,'Sydney Conference',2009,to_date('25-jun-2008','DD-MM-YYYY'), to_date('26-june-2008','DD-MM-YYYY'),'Australia','Sydney','Morton Street','Sydconf@gmail.com');
```

```
Insert into Conference values(Conference_seq.nextval,'Rome Conference',2009,to_date('13-july-2008','DD-MM-YYYY'), to_date('14-july-2008','DD-MM-YYYY'),'Italy','Rome','City Hall','Romconf@gmail.com');
```

```
Insert into Conference values(Conference_seq.nextval,'Delhi Conference',2012,to_date('03-jan-2012','DD-MM-YYYY'), to_date('04-jan-2012','DD-MM-YYYY'),'India','Delhi','Agra Hall','Delconf@gmail.com');
```

```
Insert into Conference values(Conference_seq.nextval,'New York Conference',2015,to_date('03-july-2015','DD-MM-YYYY'), to_date('04-july-2015','DD-MM-YYYY'),'USA','New York','Wall Street','Nyconf@gmail.com');
```

18. Code for Inserting data in Track table in Database FIT5148B

```
Insert into Track values(track_seq.nextval,'Engineering','Describing various Engineering',1);
```

```
Insert into Track values(track_seq.nextval,'Finance','Optimizing Finances',2);
```

```
Insert into Track values(track_seq.nextval,'Marketing','Marketing Techniques',3);
```

```
Insert into Track values(track_seq.nextval,'Sales','Sales Techniques',4);
```

```
Insert into Track values(track_seq.nextval,'Operation','Operation Techniques',5);
```

19. Code for Inserting data in PCMember table in Database FIT5148B

```
Insert into PCMember  
values(Member_seq.nextval,'Harry','Potter','DR','Lecturer','Mumbai  
University','Potter@gmail.com',1);
```

```
Insert into PCMember  
values(Member_seq.nextval,'Ron','Wisley','Prof','Lecturer','Bangalore  
University','Wisley@gmail.com',2);
```

Insert into PCMember

```
values(Member_seq.nextval,'Hermiane','Granger','DR','Researcher','Chennai  
University','Granger@gmail.com',3);
```

Insert into PCMember

```
values(Member_seq.nextval,'Albus','Dumbledore','None','Student','Kolkatta  
University','Dumbledore@gmail.com',4);
```

Insert into PCMember

```
values(Member_seq.nextval,'Severus','Snape','Prof','Researcher','Delhi  
University','Snape@gmail.com',5);
```

20. Code for Inserting data in Paper table in Database FIT5148B

```
Insert into Paper values(Paper_seq.nextval,'Electronics Engineering','Types of  
Motors','Full Paper',to_date('13-sep-2008','DD-MM-YYYY'),1);
```

```
Insert into Paper values(Paper_seq.nextval,'Finance Analysis','Strategies in  
Finances','Poster',to_date('1-June-2008','DD-MM-YYYY'),2);
```

```
Insert into Paper values(Paper_seq.nextval,'Marketing Analysis','Strategies in  
Marketing','Research_in_progress',to_date('08-July-2008','DD-MM-YYYY'),3);
```

```
Insert into Paper values(Paper_seq.nextval,'Sales Analysis','Strategies in Sales','Full  
Paper',to_date('1-Jan-2012','DD-MM-YYYY'),4);
```

```
Insert into Paper values(Paper_seq.nextval,'Operation Analysis','Strategies in  
Operations','Poster',to_date('06-July-2015','DD-MM-YYYY'),5);
```

21. Code for Inserting data in Review table in Database FIT5148B

```
Insert into Review values(Review_seq.nextval,to_date('15-sep-2008','DD-MM-  
YYYY'),to_date('19-sep-2008','DD-MM-YYYY'),'Accept','Good',1,1);
```

```
Insert into Review values(Review_seq.nextval,to_date('3-June-2008','DD-MM-  
YYYY'),to_date('11-June-2008','DD-MM-YYYY'),'Weakly Accept','Ok',2,2);
```

```
Insert into Review values(Review_seq.nextval,to_date('18-July-2008','DD-MM-  
YYYY'),to_date('20-July-2008','DD-MM-YYYY'),'Borderline','Needs Correction',3,3);
```

```
Insert into Review values(Review_seq.nextval,to_date('11-Jan-2012','DD-MM-  
YYYY'),to_date('21-Jan-2012','DD-MM-YYYY'),'Weakly Reject','Needs Rework',4,4);
```

```
Insert into Review values(Review_seq.nextval,to_date('26-July-2015','DD-MM-YYYY'),to_date('29-July-2015','DD-MM-YYYY'),'Reject','Bad',5,5);
```

22. Code for Inserting data in Author table in Database FIT5148B

```
Insert into Author values(Author_seq.nextval,'James','Bond','Harvard University','England','Bond@gmail.com',045869873);
```

```
Insert into Author values(Author_seq.nextval,'Molly','Bliss','Dallas University','USA','Bliss@gmail.com',045845878);
```

```
Insert into Author values(Author_seq.nextval,'Steve','Austin','Texas University','Russia','Austin@gmail.com',045833373);
```

```
Insert into Author values(Author_seq.nextval,'Kurt','Angle','Berlin University','Germany','Angle@gmail.com',045869845);
```

```
Insert into Author values(Author_seq.nextval,'Jorja','Iswaku','Venice University','Italy','Iswaku@gmail.com',045869555);
```

23. Code for Inserting data in BestPaper table in Database FIT5148B

```
Insert into BestPaper values(Best_Paper_seq.nextval,1,1000,1);
Insert into BestPaper values(Best_Paper_seq.nextval,2,2000,2);
Insert into BestPaper values(Best_Paper_seq.nextval,3,3000,3);
Insert into BestPaper values(Best_Paper_seq.nextval,4,4000,4);
Insert into BestPaper values(Best_Paper_seq.nextval,5,5000,5);
```

24. Code for Inserting data in Submission table in Database FIT5148B

```
Insert into Submission values(Submission_seq.nextval,1,1,1,'Y');
Insert into Submission values(Submission_seq.nextval,2,2,2,'N');
Insert into Submission values(Submission_seq.nextval,3,3,1,'Y');
Insert into Submission values(Submission_seq.nextval,4,4,3,'N');
Insert into Submission values(Submission_seq.nextval,5,5,2,'N');
```

3. Task 2 -- Stored Procedure

6. Code for updatevenue:

```

delimiter//  

create or replace procedure updatevenue(  

    new_venue IN conference.conf_venue%type,  

    new_name IN conference.conf_name%type)  

IS  

begin  

    update conference set conf_venue=new_venue where conf_name=new_name;  

    commit;  

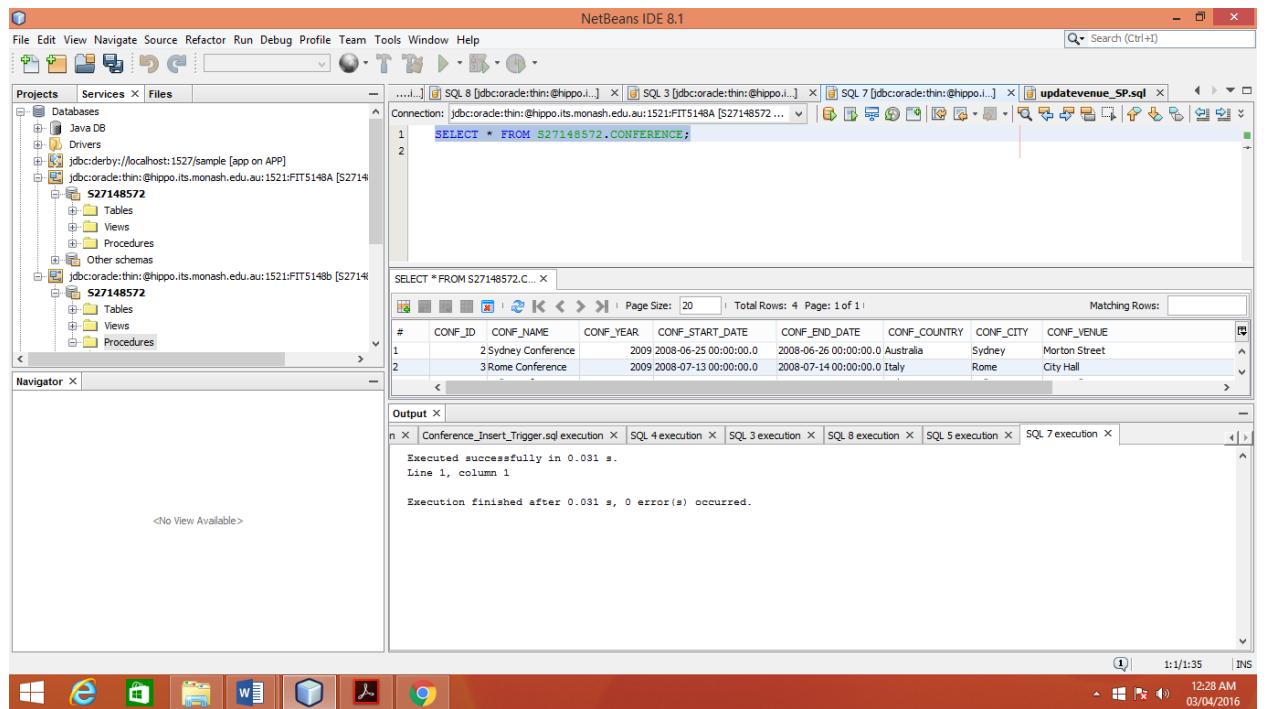
end;/  

delimiter;

```

Data in Conference table before execution of stored procedure:

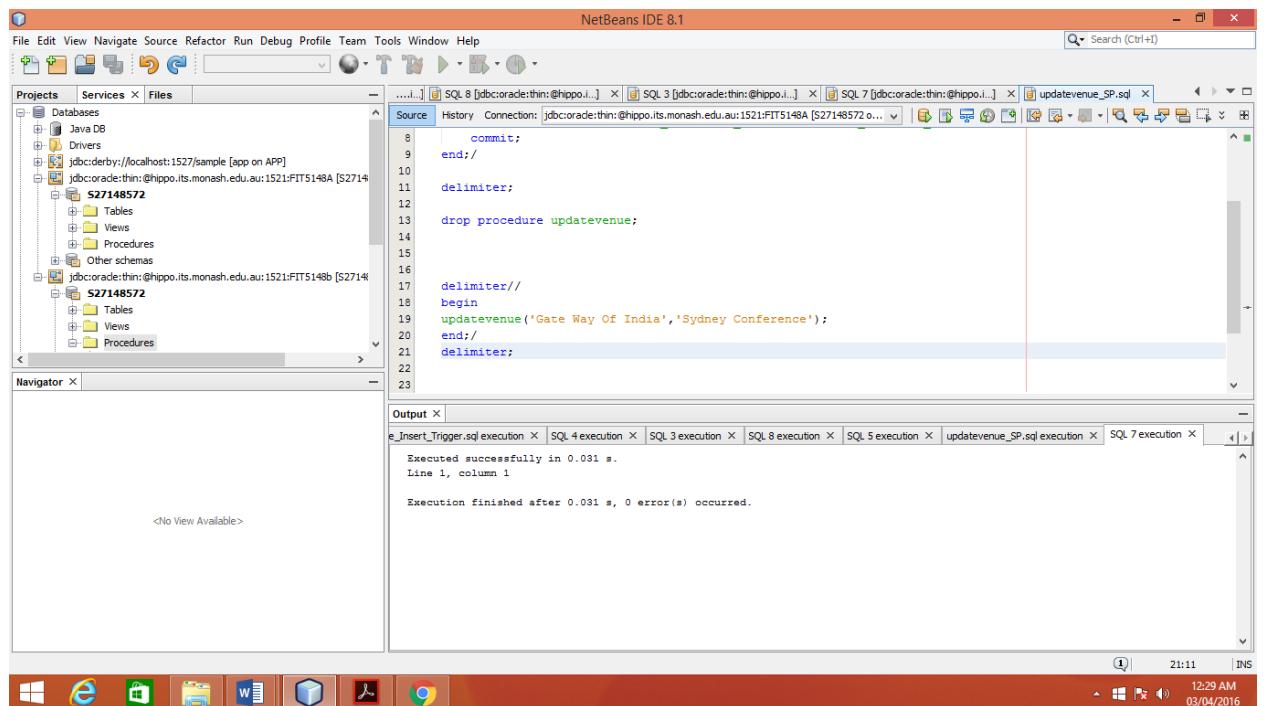


The screenshot shows the NetBeans IDE 8.1 interface. The Database browser on the left displays a tree structure of databases, tables, and procedures under the connection 'S27148572'. The SQL editor on the right contains the query 'SELECT * FROM S27148572.CONFERENCE;'. The results pane shows the following data:

#	CONF_ID	CONF_NAME	CONF_YEAR	CONF_START_DATE	CONF_END_DATE	CONF_COUNTRY	CONF_CITY	CONF_VENUE
1	2	Sydney Conference	2009	2009-06-25 00:00:00.0	2009-06-26 00:00:00.0	Australia	Sydney	Morton Street
2	3	Rome Conference	2009	2009-07-13 00:00:00.0	2009-07-14 00:00:00.0	Italy	Rome	City Hall

The output pane at the bottom indicates the execution was successful.

Sql command executed in order to test Stored Procedure:



The screenshot shows the NetBeans IDE 8.1 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains various icons for file operations. The left sidebar has sections for Projects, Services, and Files. Under Projects, there are Databases, Java DB, Drivers, and several connection nodes like jdbcderby://localhost:1527/sample [app on APP] and jdbc:orade:thin:@hippo.its.monash.edu.au:1521:FIT5148A [S27148572]. The main workspace shows a code editor with the following SQL script:

```
8   commit;
9 end;/
10
11 delimiter;
12
13 drop procedure updatevenue;
14
15
16 delimiter//;
17 begin
18 updatevenue('Gate Way Of India','Sydney Conference');
19 end;/
20
21 delimiter;
```

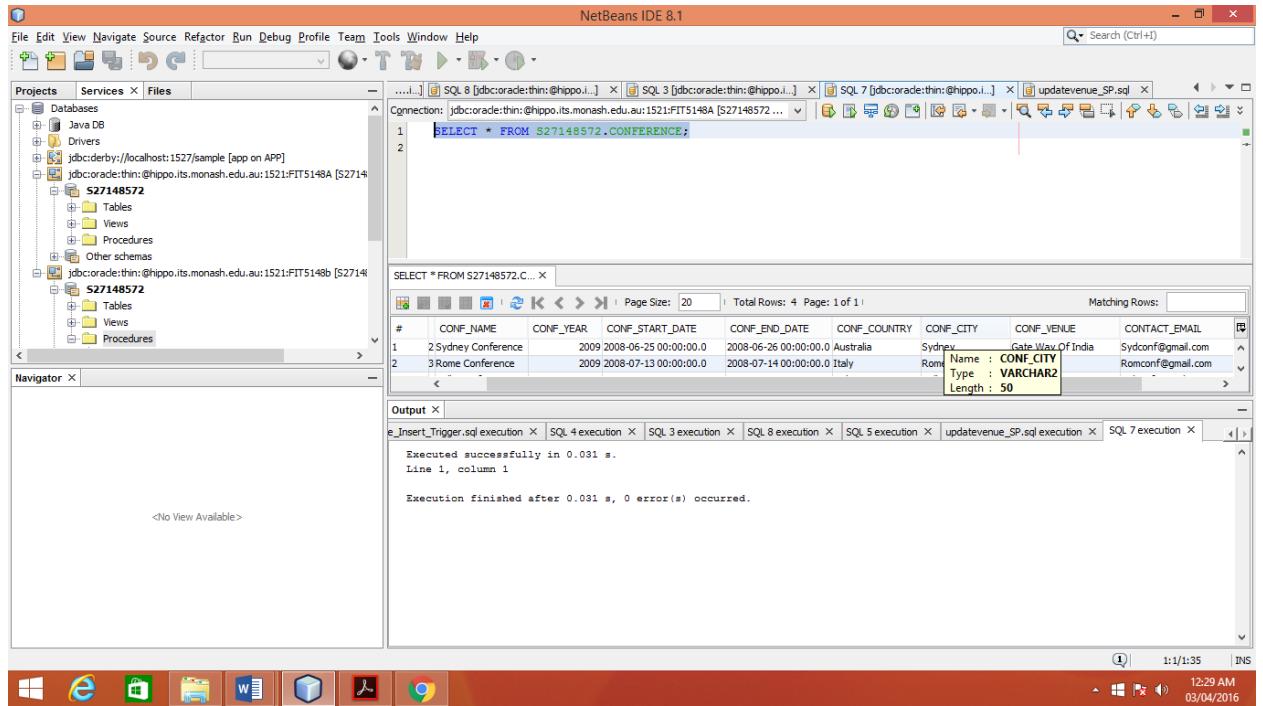
The Output panel below the code editor displays the results of the execution:

```
Executed successfully in 0.031 s.
Line 1, column 1

Execution finished after 0.031 s, 0 error(s) occurred.
```

The system tray at the bottom right shows the date and time as 03/04/2016, 12:29 AM.

Data in Conference after the execution of above command:



4. Task 3 – Triggers

7. Code for Conference_Insert_Trigger:

```
DELIMITER //
CREATE OR REPLACE TRIGGER S27148572.ReferenceConferenceinsert
BEFORE INSERT ON track
FOR EACH ROW
```

```
DECLARE
conf_count number(5);
BEGIN

IF INSERTING
THEN

SELECT COUNT(*) INTO conf_count FROM Conference@FIT5148A WHERE
CONF_ID = :new.conf_ID;
IF (conf_count <=0)
THEN
raise_application_error
(-20999, 'Invalid Conference');
END IF;
```

END IF;

END; /

DELIMITER ;

Data in track table before execution of command to test insert trigger:

NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Projects Services Files

Connection: jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572 ...]

```
1 SELECT * FROM S27148572.TRACK;
2
```

SELECT * FROM S27148572.T...

#	TRACK_ID	TRACK_NAME	TRACK_DESCRIPTION	CONF_ID
1		1 Engineering	Describing various Engineering	1
2		2 Finance	Optimizing Finances	2
3		3 Marketing	Marketing Techniques	3
4		4 Sales	Sales Techniques	4
5		5 Operation	Operation Techniques	5

Output

WebDemo (run) | Conference_Update_Trigger.sql execution | SQL 1 execution | SQL 3 execution

```
Executed successfully in 0.032 s.
Line 1, column 1

Execution finished after 0.032 s, 0 error(s) occurred.
```

1:1/1:30 10:45 PM 02/04/2016

Query executed to test Insert trigger: (Conf_ID = 25 is not present in Conference table. When a user tries to insert a row in Track table having Conf_ID = 25, an error is generated.):

NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Projects Services Files

Connection: jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572 ...]

```
1 Insert into Track values(track_seq.nextval,'Engineering','Describing various Engineering',25);
```

Output

WebDemo (run) | Conference_Update_Trigger.sql execution | SQL 1 execution

```
Error code 20999, SQL state 72000: ORA-20999: Invalid Conference
ORA-06512: at "S27148572.REFERENCECONFERENCEINSERT", line 12
ORA-04088: error during execution of trigger 'S27148572.REFERENCECONFERENCEINSERT'
Line 1, column 1
```

1:1/1:94 10:45 PM 02/04/2016

8. Code for Conference_Update_Trigger:

```
DELIMITER //
CREATE OR REPLACE TRIGGER S27148572.ReferenceConferenceUpdate
BEFORE UPDATE ON track
FOR EACH ROW

DECLARE
conf_count number(5);
BEGIN

IF UPDATING
THEN
SELECT COUNT(*) INTO conf_count FROM Conference@FIT5148A WHERE CONF_ID =
:new.conf_ID;
IF (conf_count <=0)
THEN
raise_application_error
(-20999, 'Invalid Conference');
END IF;
END IF;

END; /
DELIMITER ;
```

Data in Track table before execution of command:

The screenshot shows the NetBeans IDE 8.1 interface. In the top menu bar, the following items are visible: File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. A search bar is located at the top right. The left sidebar has sections for Projects, Services, Files, Navigator, and Procedures. The Projects section lists several database tables: BESTPAPER, CONFERENCETRACK, DISCOUNTS, Hori_Frag_Conf_S2, PAPER, PCMEMBER, REVIEW, SALES, SALESDETAIL, STORES, SUBMISSION, TEST, TRACK, and VERT_Frag_Conf_S1. The Navigator section shows '<No View Available>'. The main workspace contains a SQL editor tab with the following code:

```
1  SELECT * FROM S27148572.TRACK;
```

Below the code, the results of the query are displayed in a table:

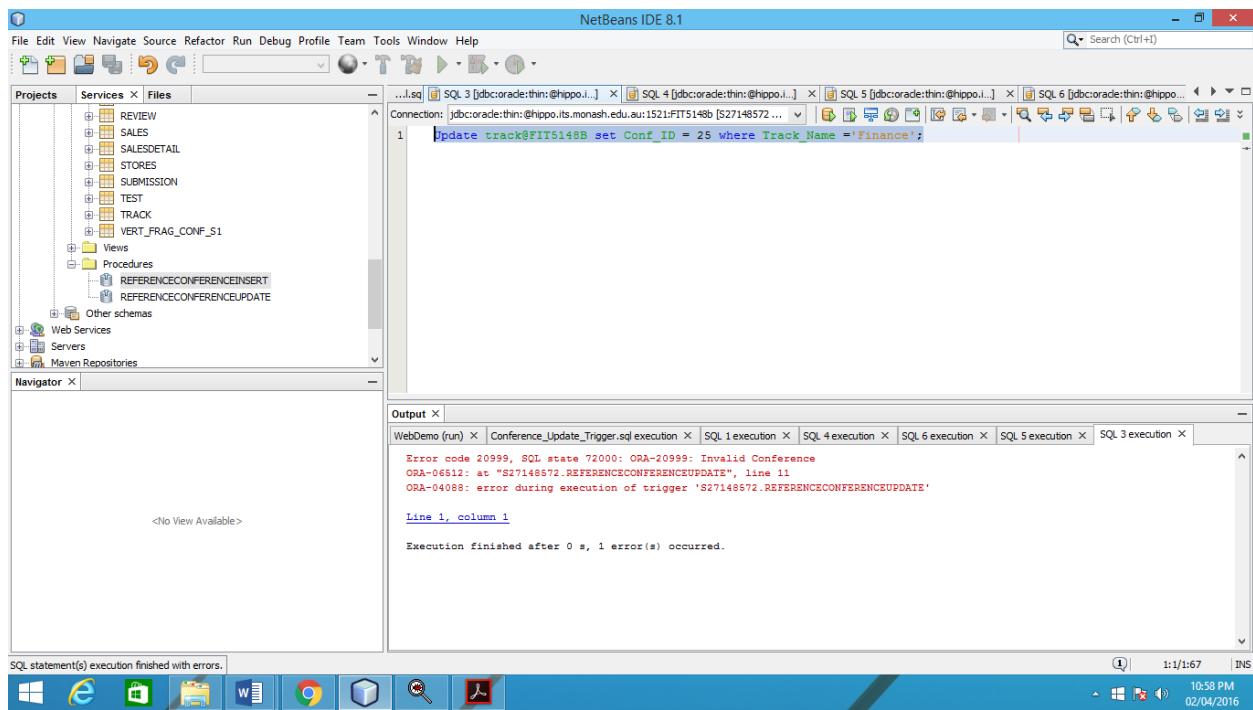
#	TRACK_ID	TRACK_NAME	TRACK_DESCRIPTION	CONF_ID
1		1Engineering	Describing various Engineering	1
2		2Finance	Optimizing Finances	2
3		3Marketing	Marketing Techniques	3
4		4Sales	Sales Techniques	4
5		5Operation	Operation Techniques	5

The bottom pane shows the Output window with the following log:

```
WebDemo (run) | Conference_Update_Trigger.sql execution | SQL 1 execution | SQL 3 execution |
Executed successfully in 0.032 s.
Line 1, column 1
Execution finished after 0.032 s, 0 error(s) occurred.
```

Command executed to test update trigger:

(When a user tries to update Conf_ID to such a value which is not present in Conference table an error is thrown.)



9. Code for Conference_Delete_Trigger:

```

DELIMITER //
CREATE OR REPLACE TRIGGER S27148572.ReferenceConferenceDelete
BEFORE Delete ON Conference
FOR EACH ROW

DECLARE
BEGIN

IF Deleting
THEN

Delete FROM Track@FIT5148B WHERE CONF_ID = :old.conf_ID;
END IF;

END; /
DELIMITER ;

```

Data in Track table before execution of command:

```
NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
...l.sql SQL 3 [jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148A [S27148572 ...
Connection: jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572 ...
1 SELECT * FROM S27148572.TRACK;
2
3 Drop trigger ReferenceConferenceinsert;
4
```

#	TRACK_ID	TRACK_NAME	TRACK_DESCRIPTION	CONF_ID
1		1 Engineering	Describing various Engineering	1
2		2 Finance	Optimizing Finances	2
3		3 Marketing	Marketing Techniques	3
4		4 Sales	Sales Techniques	4
5		5 Operation	Operation Techniques	5

<No View Available>

Output

WebDemo (run) | Conference_Update_Trigger.sql execution | SQL 1 execution | SQL 4 execution | SQL 6 execution | SQL 3 execution | Conference_Insert_Trigger.sql execution | SQL 7 execution | SQL 5 execution

Line 1, column 1

Execution finished after 0.031 s, 0 error(s) occurred.

Output | SQL statement(s) executed successfully.

Windows Taskbar: File Explorer, Internet Explorer, Microsoft Edge, Word, Google Chrome, File Manager, Search, Task View, Taskbar icons, System tray: 11:13 PM 02/04/2016

When a Conference ID is deleted in Conference table all the rows in Track table having same Conf_ID are deleted:

```
NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
...l.sql SQL 5 [jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148A [S27148572 ...
Connection: jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572 ...
1 Delete from Conference where Conf_ID = 1;
2
3
4
```

<No View Available>

Output

Trigger.sql execution | SQL 1 execution | Conference_Insert_Trigger.sql execution | SQL 4 execution | SQL 3 execution | SQL 8 execution | SQL 5 execution

Executed successfully in 0.031 s.

Line 1, column 1

Execution finished after 0.031 s, 0 error(s) occurred.

Windows Taskbar: File Explorer, Internet Explorer, Microsoft Edge, Word, Google Chrome, File Manager, Search, Task View, Taskbar icons, System tray: 11:21 PM 02/04/2016

Data in Track table after execution of above command:

The screenshot shows the NetBeans IDE 8.1 interface with the Database window open. The Projects panel on the left lists various database objects like HORL_FRAG_CONF_S1, PUBLISHERS, ROYSCRED, STUDENT, TITLEAUTHOR, TITLES, and VERT_FRAG_CONF_S2. The Connections tab shows three connections: SQL 5 [dbcorade:thin:@hippo...], SQL 8 [dbcorade:thin:@hippo...], and SQL 3 [dbcorade:thin:@hippo...]. The central workspace contains a SQL editor with the following query:

```
SELECT * FROM S27148572.TRACK;
```

Below the query is a table viewer showing the results of the query:

#	TRACK_ID	TRACK_NAME	TRACK_DESCRIPTION	CONF_ID
1	2	Finance	Optimizing Finances	2
2	3	Marketing	Marketing Techniques	3
3	4	Sales	Sales Techniques	4
4	5	Operation	Operation Techniques	5

The Output panel at the bottom shows the execution results:

```
Trigger.sql execution X SQL 1 execution X Conference_Insert_Trigger.sql execution X SQL 4 execution X SQL 3 execution X SQL 8 execution X SQL 5 execution X
Executed successfully in 0.031 s.
Line 1, column 1

Execution finished after 0.031 s, 0 error(s) occurred.
```

5. Task 4 – Fragmentation

10. Vertical fragmentation

- Q1: select conf_start_date, conf_end_date from conference where conf_id = value;
Q2: select conf_year, conf_name from conference where conf_id = value;
Q3: select conf_country, conf_venue from conference;
Q4: select conf_city, contact_email from conference;

- A1: conf_id
A2: conf_name
A3: conf_year
A4: conf_start_date

A5: conf_end_date
 A6: conf_country
 A7: conf_city
 A8: conf_venue
 A9: contact_email

Usage matrix

	A1	A2	A3	A4	A5	A6	A7	A8	A9
Q1	1	0	0	1	1	0	0	0	0
Q2	1	1	1	0	0	0	0	0	0
Q3	0	0	0	0	0	1	0	1	0
Q4	0	0	0	0	0	0	1	0	1

Access frequency matrix

There are two sites:

S1: FIT5148A

S2: FIT5148B

	S1	S2	
Q1	20	10	30
Q2	30	5	35
Q3	10	0	10
Q4	5	0	5

Attribute affinity matrix

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	65	35	35	30	30	0	0	0	0
A2	35	35	35	0	0	0	0	0	0
A3	35	35	35	0	0	0	0	0	0
A4	0	0	0	30	30	0	0	0	0
A5	0	0	0	30	30	0	0	0	0
A6	0	0	0	0	0	10	0	10	0
A7	0	0	0	0	0	0	5	0	5
A8	0	0	0	0	0	10	0	10	0
A9	0	0	0	0	0	0	5	0	5

Site S2 will use attributes A1(conf_id), A2(conf_name), A3(conf_year), A4(conf_start_date), A5(conf_start_date).

Site S1 will use attributes A1(conf_id), A6(conf_country), A7(conf_city), A8(conf_venue), A9(contact_email).

Create table vert_frag_conf_s1 as select conf_id, conf_name, conf_year, conf_start_date, conf_start_date from S27148572.conference@fit5148a;

The screenshot shows the NetBeans IDE 8.1 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The main workspace has several tabs open: insertauthor.jsp, InsertData.java, error.jsp, SQL 1 [jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572.conference@fit5148a], and SQL 2 [jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572.conference@fit5148a]. The SQL 1 tab contains the following code:

```
1 Create table vert_frag_conf_s1 as select
2   conf_id, conf_name, conf_year, conf_start_date, conf_end_date from S27148572.conference@fit5148a;
3
4 Select * from vert_frag_conf_s1;
```

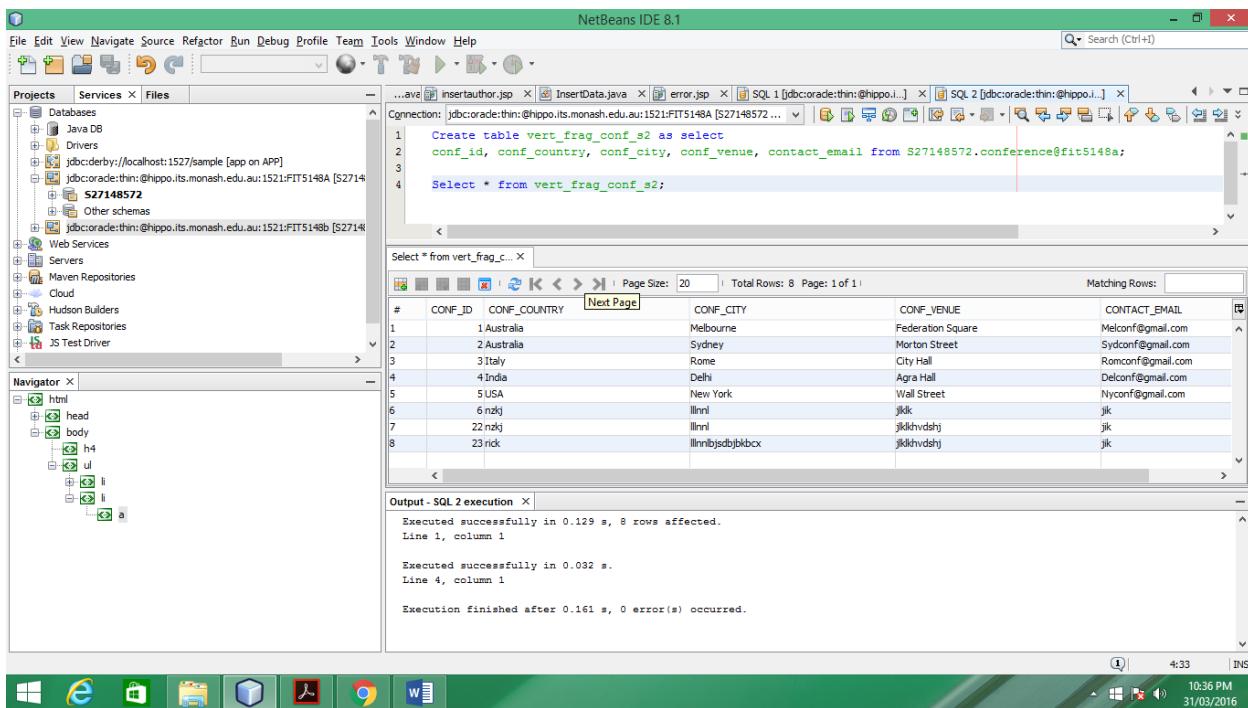
The SQL 2 tab shows the results of the query:

#	CONF_ID	CONF_NAME	CONF_YEAR	CONF_START_DATE	CONF_END_DATE
1		1 Melbourne Conference	2008	2008-09-23 00:00:00.0	2008-09-24 00:00:00.0
2		2 Sydney Conference	2009	2008-06-25 00:00:00.0	2008-06-26 00:00:00.0
3		3 Rome Conference	2009	2008-07-13 00:00:00.0	2008-07-14 00:00:00.0
4		4 Delhi Conference	2012	2012-01-03 00:00:00.0	2012-01-04 00:00:00.0
5		5 New York Conference	2015	2015-07-03 00:00:00.0	2015-07-04 00:00:00.0
6		6 Ild	2016	2016-03-16 00:00:00.0	2016-03-16 00:00:00.0
7		22 Ild	2016	2016-03-16 00:00:00.0	2016-03-16 00:00:00.0
8		23 Ild	2016	2016-03-16 00:00:00.0	2016-03-16 00:00:00.0

The Output - SQL 2 execution panel shows the execution details:

```
Executed successfully in 0.202 s, 8 rows affected.  
Line 1, column 1  
  
Executed successfully in 0.034 s.  
Line 4, column 1  
  
Execution finished after 0.236 s, 0 error(s) occurred.
```

Create table vert_frag_conf_s2 as select conf_id, conf_country, conf_city, conf_venue, contact_email from S27148572.conference@fit5148a;



11. Horizontal Fragmentation

List of Predicates:

- 1) Award Price > 20000 and Award Price <= 20000
- 2) Not(Award Price > 20000) and Award Price <= 20000
- 3) Award Price > 20000 and Not(Award Price <= 20000)
- 4) Not(Award Price > 20000) and Not(Award Price <= 20000)

From the above list we need to derive valid predicates which will be our midterms:

- 1) ~~Award Price > 20000 and Award Price <= 20000~~
- 2) ~~Not(Award Price > 20000) and Award Price <= 20000~~
- 3) ~~Award Price > 20000 and Not(Award Price <= 20000)~~
- 4) ~~Not(Award Price > 20000) and Not(Award Price <= 20000)~~

Therefore the correct midterms predicates are:

Midterm Predicates:

- 1) Award Price > 20000
- 2) Award Price <= 20000

The following queries will help us to generate the above fragments:

- 1) Create table hori_frag_conf_s1 as select * from S27148572.BestPaper@fit5148b where Award_Price > 20000 ;

NetBeans IDE 8.1

```

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Services Files
Projects Databases Java DB Drivers jdbc:derby://localhost:1527/sample [app on APP] jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148A [S27148572]
Schemas Other schemas jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572]
Web Services Servers Maven Repositories Cloud Hudson Builders Task Repositories JS Test Driver
Navigator x
html head body h4 ul li li a
...ave insertauthor.jsp InsertData.java error.jsp SQL 1 [jdbc:oracle:thin:@hippo...] SQL 2 [jdbc:oracle:thin:@hippo...] Connection: jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148A [S27148572...]
1 Create table hori_frag_conf_s1 as
2 select * from S27148572.BestPaper@fit5148b where Award_Price > 20000;
3
4 Select * from hori_frag_conf_s1;

Select * from hori_frag_c... x
Output - SQL 2 execution x
# BEST_PAPER_ID TITLE_ID AWARD_PRICE TRACK_ID
1 6 5 46646 6

```

Output - SQL 2 execution x

```

Executed successfully in 0.366 s, 1 rows affected.
Line 1, column 1

Executed successfully in 0.044 s.
Line 4, column 1

Execution finished after 0.41 s, 0 error(s) occurred.

```

1:1/4:138 | 10:39 PM 31/03/2016

- 2) Create table hori_frag_conf_s2 as select * from S27148572.BestPaper@fit5148b where Award_Price <= 20000;

NetBeans IDE 8.1

```

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
File Services Files
Projects Databases Java DB Drivers jdbc:derby://localhost:1527/sample [app on APP] jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572...]
Schemas Other schemas jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572]
Web Services Servers Maven Repositories Cloud Hudson Builders Task Repositories JS Test Driver
Navigator x
html head body h4 ul li li a
...ave insertauthor.jsp InsertData.java error.jsp SQL 1 [jdbc:oracle:thin:@hippo...] SQL 2 [jdbc:oracle:thin:@hippo...] Connection: jdbc:oracle:thin:@hippo.its.monash.edu.au:1521:FIT5148b [S27148572...]
1 Create table hori_frag_conf_s2 as
2 select * from S27148572.BestPaper@fit5148b where Award_Price <= 20000;
3
4 Select * from hori_frag_conf_s2;

Select * from hori_frag_c... x
Output - SQL 2 execution x
# BEST_PAPER_ID TITLE_ID AWARD_PRICE TRACK_ID
1 2 2 11111 2
2 3 3 3000 3
3 4 4 4000 4
4 5 5 5000 5


```

Output - SQL 2 execution x

```

Executed successfully in 0.085 s, 4 rows affected.
Line 1, column 1

Executed successfully in 0.039 s.
Line 4, column 1

Execution finished after 0.124 s, 0 error(s) occurred.

```

1:1/4:139 | 10:42 PM 31/03/2016

6. Task 5 – User Interface

12. Login Page



- This page is used to check the database credentials. If true it will allow user to go on Main Page else it print an error message.

25. Code for checking login credentials:

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    String User_ID = UserText.getText();  
    String CorrectID ="S27148572";  
    char[] password = PassText.getPassword();  
    char[] correctPass = new char[] {'s', 't', 'u', 'd', 'e', 'n', 't'};  
    if (Arrays.equals(password, correctPass) && User_ID.equals(CorrectID))  
    {  
        dispose();  
        Page02 pg = new Page02();  
        pg.setVisible(true);  
    }  
}
```

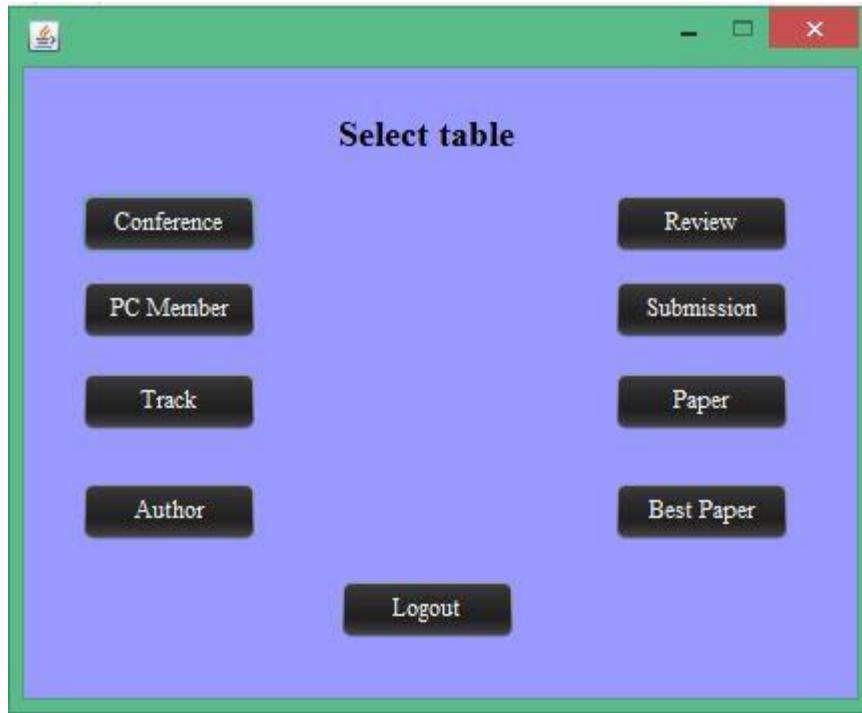
```
else
{
    JOptionPane.showMessageDialog(null,"Invalid Credentials");
}
if (User_ID.equals(CorrectID))
{
    if (Arrays.equals(password, correctPass))
    {

    }
    else
    {
        JOptionPane.showMessageDialog(null,"Invalid Password");
    }
}
else
{
    JOptionPane.showMessageDialog(null,"Invalid User ID");
}
}
```

26. Code for Cancel Button:

```
private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}
```

13. Main Page:



- This page is used to navigate to different frames. It includes 9 buttons.
- Logout button: It is used to close the system. Code for the same:

```
private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    dispose();
    JOptionPane.showMessageDialog(null,"Logout Successful");
}
```

- PCMember button: It is used to navigate to PCMember frame. Code for the same:

```
private void btnPCMemberActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
    PCMember ct = new PCMember();
    ct.setVisible(true);
    Jayesh_Assignment.ConnectDatabase();
    Jayesh_Assignment.ConnectDatabase1();
}
```

- Conference button: It is used to navigate to Conference frame. Code for the same:

```
private void btnConferenceActionPerformed(java.awt.event.ActionEvent evt) {
```

```

        dispose();
        Page03 pg = new Page03();
        pg.setVisible(true);
        Jayesh_Assignment.ConnectDatabase();
        Jayesh_Assignment.ConnectDatabase1();
    }

```

- Track button: It is used to navigate to ConferenceTrack frame. Code for the same:

```

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

    dispose();
    ConferenceTrack ct = new ConferenceTrack();
    ct.setVisible(true);
    Jayesh_Assignment.ConnectDatabase();
    Jayesh_Assignment.ConnectDatabase1();
}

```

- Author button: It is used to navigate to Author frame. Code for the same:

```

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

    dispose();
    Author ct = new Author();
    ct.setVisible(true);
    Jayesh_Assignment.ConnectDatabase();
    Jayesh_Assignment.ConnectDatabase1();
}

```

- Paper button: It is used to navigate to Paper frame. Code for the same:

```

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

    dispose();
    Paper ct = new Paper();
    ct.setVisible(true);
    Jayesh_Assignment.ConnectDatabase();
    Jayesh_Assignment.ConnectDatabase1();
}

```

- Submission button: It is used to navigate to Submission frame. Code for the same:

```

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

    dispose();
    Submission ct = new Submission();
    ct.setVisible(true);
}

```

```
        Jayesh_Assignment.ConnectDatabase();
        Jayesh_Assignment.ConnectDatabase1();
    }
```

- Review button: It is used to navigate to Review frame. Code for the same:

```
private void btnReviewActionPerformed(java.awt.event.ActionEvent evt) {

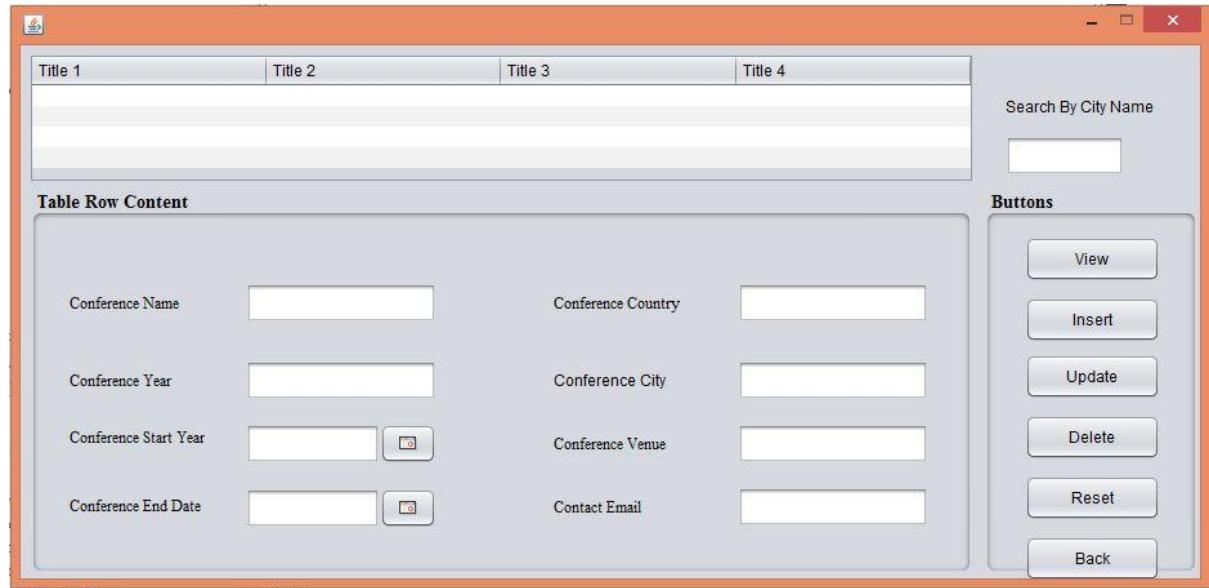
    dispose();
    Review ct = new Review();
    ct.setVisible(true);
    Jayesh_Assignment.ConnectDatabase();
    Jayesh_Assignment.ConnectDatabase1();
}
```

- BestPaper button: It is used to navigate to BestPaper frame. Code for the same:

```
private void btnBestPaperActionPerformed(java.awt.event.ActionEvent evt) {

    dispose();
    Best_Paper ct = new Best_Paper();
    ct.setVisible(true);
    Jayesh_Assignment.ConnectDatabase();
    Jayesh_Assignment.ConnectDatabase1();
}
```

14. Conference Frame:



- This frame is used to perform all SQL operations on Conference table present in database FIT5148A.

27. Code to view the content of Conference table:

```
private void View_Table()
{
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn = abc.getConn();
        String sql ="Select * from Conference@FIT5148A";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
        Conference_table.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
}
```

Clicking on view button you will get content of Conference Table:

The screenshot shows a Java Swing application window titled "Conference Management System". The window has a title bar with standard window controls (minimize, maximize, close). Below the title bar is a toolbar with a search icon and a search input field labeled "Search By City Name".

The main content area is divided into several sections:

- Table Row Content:** A table showing conference data with columns: CONF_ID, CONF_NA..., CONF_YE..., CONF_ST..., CONF_EN..., CONF_CO..., CONF_CITY, CONF_VE..., and CONTACT... . The data includes rows for Sydney Conference (2009), Rome Conference (2009), Delhi Conference (2012), and New York Conference (2015).
- Form Fields:** A form for entering new conference details. It includes fields for Conference Name, Conference Country, Conference Year, Conference City, Conference Start Year, Conference Venue, Conference End Date, Contact Email, and two date pickers.
- Buttons:** A vertical column of buttons on the right side of the form for performing actions: View, Insert, Update, Delete, Reset, and Back.

28. Code to reset the user interface value to null:

```
public void clearAll()
{
    txt_conf_name.setText("");
    txt_conf_year.setText("");
    Conf_Start_Date.setDate(null);
    Conf_End_Date.setDate(null);
    txt_conf_country.setText("");
    txt_conf_city.setText("");
    txt_conf_venue.setText("");
    txt_email.setText("");
}
```

29. Code to go back on the Main Page:

```
private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Jayesh_Assignment abc = new Jayesh_Assignment();
    Connection conn = abc.getConn();
    Connection conn1 = abc.getConn1();
    if(conn != null)
    {
        Jayesh_Assignment.ConnectDatabase();
    }
    else
    {
        dispose();
    }
}
```

```

        }
        if(conn1 != null)
        {
            Jayesh_Assignment.ConnectDatabase1();
        }
        else
        {
            dispose();
        }
        dispose();
        abc.setConn(null);
        abc.setConn1(null);
        Page02 pg;
        pg = new Page02();
        pg.setVisible(true);
    }
}

```

30. Code to insert row in Conference table:

```

private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn = abc.getConn();
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");
        String d = sdf.format(Conf_Start_Date.getDate());
        String d1 = sdf.format(Conf_End_Date.getDate());
        String sql ="Insert into Conference@FIT5148A (Conf_ID,
CONF_NAME, CONF_YEAR, CONF_START_DATE, CONF_END_DATE,
CONF_COUNTRY, CONF_CITY, CONF_VENUE, CONTACT_EMAIL)
values
(Conference_seq.NEXTVAL,"+txt_conf_name.getText()+"','"+txt_conf_year.getText()+"','"+d+"','"+d1+"','"+txt_conf_country.getText()+"','"+txt_conf_city.getText()+"','"+txt_conf_venue.getText()+"','"+txt_email.getText()+"')";
        System.out.println(sql);
        pst = conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null,"Inserted");

    }
    catch (SQLException | HeadlessException e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
}

```

```

        }
        View_Table();
    }
}

```

Add value in user interface and then press insert button and then check the table content. The values will be inserted:

Data in the table before the execution of Query:

The screenshot shows a Windows application window with a title bar and a close button. On the right side, there is a search bar labeled "Search By City Name" and a "Buttons" panel containing six buttons: View, Insert, Update, Delete, Reset, and Back. The main area contains a table with columns: CONF_ID, CONF_NA..., CONF_YE..., CONF_ST..., CONF_EN..., CONF_CO..., CONF_CITY, CONF_VE..., and CONTACT... . Below the table is a "Table Row Content" section with input fields for Conference Name (Banglore Conference), Conference Country (India), Conference Year (2010), Conference City (Banglore), Conference Start Year (2010-04-05), Conference Venue (Star Mall), Conference End Date (2010-04-20), and Contact Email (bang@gmail.com).

CONF_ID	CONF_NA...	CONF_YE...	CONF_ST...	CONF_EN...	CONF_CO...	CONF_CITY	CONF_VE...	CONTACT...
2	Sydney Co...	2009	2008-06-2...	2008-06-2...	Australia	Sydney	Gate Way ...	Sydconf@g...
3	Rome Con...	2009	2008-07-1...	2008-07-1...	Italy	Rome	City Hall	Romconf@...
4	Delhi Conf...	2012	2012-01-0...	2012-01-0...	India	Delhi	Agra Hall	Delconf@g...
5	New York ...	2015	2015-07-0...	2015-07-0...	USA	New York	Wall Street	Nyconf@g...

Data in the table after the execution of Query:

The screenshot shows the same Windows application window after an insertion. The table now includes a new row with CONF_ID 41 and Conference Name "Banglore ...". The rest of the data remains the same. The "Table Row Content" section still displays the same input fields as before.

CONF_ID	CONF_NA...	CONF_YE...	CONF_ST...	CONF_EN...	CONF_CO...	CONF_CITY	CONF_VE...	CONTACT...
41	Banglore ...	2010	2010-04-0...	2010-04-2...	India	Banglore	Star Mall	bang@gm...
2	Sydney Co...	2009	2008-06-2...	2008-06-2...	Australia	Sydney	Gate Way ...	Sydconf@g...
3	Rome Con...	2009	2008-07-1...	2008-07-1...	Italy	Rome	City Hall	Romconf@...
4	Delhi Conf...	2012	2012-01-0...	2012-01-0...	India	Delhi	Agra Hall	Delconf@g...
5	New York ...	2015	2015-07-0...	2015-07-0...	USA	New York	Wall Street	Nyconf@g...

31. Code to update row in Conference table:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if (rowSelected == true)  
    {  
        try  
        {  
            int row = Conference_table.getSelectedRow();  
            String Table_click = (Conference_table.getModel().getValueAt(row,  
0).toString());  
            Jayesh_Assignment abc = new Jayesh_Assignment();  
            Connection conn = abc.getConn();  
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");  
            String d1 = txt_conf_name.getText();  
            String d2 = txt_conf_year.getText();  
            String d3 = sdf.format(Conf_Start_Date.getDate());  
            String d4 = sdf.format(Conf_End_Date.getDate());  
            String d5 = txt_conf_country.getText();  
            String d6 = txt_conf_city.getText();  
            String d7 = txt_conf_venue.getText();  
            String d8 = txt_email.getText();  
            String sql = "Update Conference@FIT5148A set Conf_name ='" + d1 +  
", Conf_year ='" + d2 + "', Conf_start_date ='" + d3 + "','" + "Conf_end_date ='"  
+ d4 + "','" + "Conf_country ='" + d5 + "','" + "Conf_venue ='" + d6 + "','" +  
"Contact_email ='" + d7 + "' where Conf_ID ='" +Table_click+"';  
            System.out.println(sql);  
            pst = conn.prepareStatement(sql);  
            pst.execute();  
            JOptionPane.showMessageDialog(null,"Updated");  
        }  
        catch (SQLException | HeadlessException e)  
        {  
            JOptionPane.showMessageDialog(null,e);  
        }  
        View_Table();  
        rowSelected = false;  
    }  
}
```

```

        }
    else
    {
        JOptionPane.showMessageDialog(null,"Please Select a row");
    }
}

```

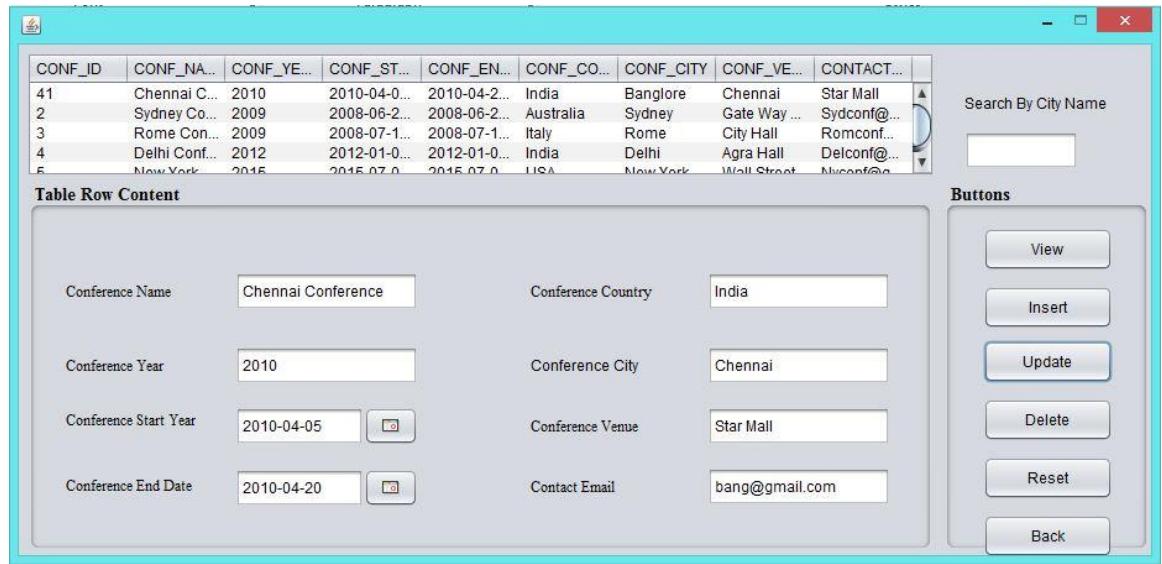
Select the row you want to update and then enter the new values that you want to update and then press update:

Data in the table before the execution of Command:

The screenshot shows a Java Swing application window titled "Conference Management System". On the left, there is a table with columns: CONF_ID, CONF_NAME, CONF_YEAR, CONF_START_DATE, CONF_END_DATE, CONF_COUNTRY, CONF_CITY, CONF_VENUE, and CONTACT_EMAIL. The table contains five rows of data. On the right, there is a "Buttons" panel with buttons for View, Insert, Update, Delete, Reset, and Back. Below the buttons is a "Table Row Content" panel containing input fields for Conference Name, Conference Year, Conference Start Year, Conference End Date, Conference Country, Conference City, Conference Venue, and Contact Email. The "Conference Name" field has "Chennai|Conference" entered into it.

CONF_ID	CONF_NAME	CONF_YEAR	CONF_START_DATE	CONF_END_DATE	CONF_COUNTRY	CONF_CITY	CONF_VENUE	CONTACT_EMAIL
41	Banglore ...	2010	2010-04-0...	2010-04-2...	India	Banglore	Star Mall	bang@gm...
2	Sydney Co...	2009	2008-06-2...	2008-06-2...	Australia	Sydney	Gate Way ...	Sydconf@...
3	Rome Con...	2009	2008-07-1...	2008-07-1...	Italy	Rome	City Hall	Romconf...
4	Delhi Conf...	2012	2012-01-0...	2012-01-0...	India	Delhi	Agra Hall	Delcon@...
5	New York	2015	2015-07-0...	2015-07-0...	USA	New York	Wall Street	Nycconf@...

Data in the table after the execution of Command:



32. Code to delete row in Conference table:

```

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

    // TODO add your handling code here:
    if (rowSelected == true)
    {
        int p = JOptionPane.showConfirmDialog(null, "Do you really want to
delete","Delete",JOptionPane.YES_NO_OPTION );
        if( p == 0)
        {
            try
            {
                Jayesh_Assignment abc = new Jayesh_Assignment();
                Connection conn = abc.getConn();
                int row = Conference_table.getSelectedRow();
                String Table_click = (Conference_table.getModel().getValueAt(row,
0).toString());
                String sql = "Delete from Conference where Conf_id
                =" +Table_click+ "";
                pst = conn.prepareStatement(sql);
                rs = pst.executeQuery();
                JOptionPane.showMessageDialog(null, "Row is deleted");
            }
            catch (SQLException | HeadlessException e)
            {
                JOptionPane.showMessageDialog(null, e);
            }
        }
    }
}

```

```

        }
        View_Table();
        rowSelected = false;
    }
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select a row and then press delete:

Data in the table before the execution of Command:

The screenshot shows a Java Swing application window titled "Conference Management System".

Table Row Content:

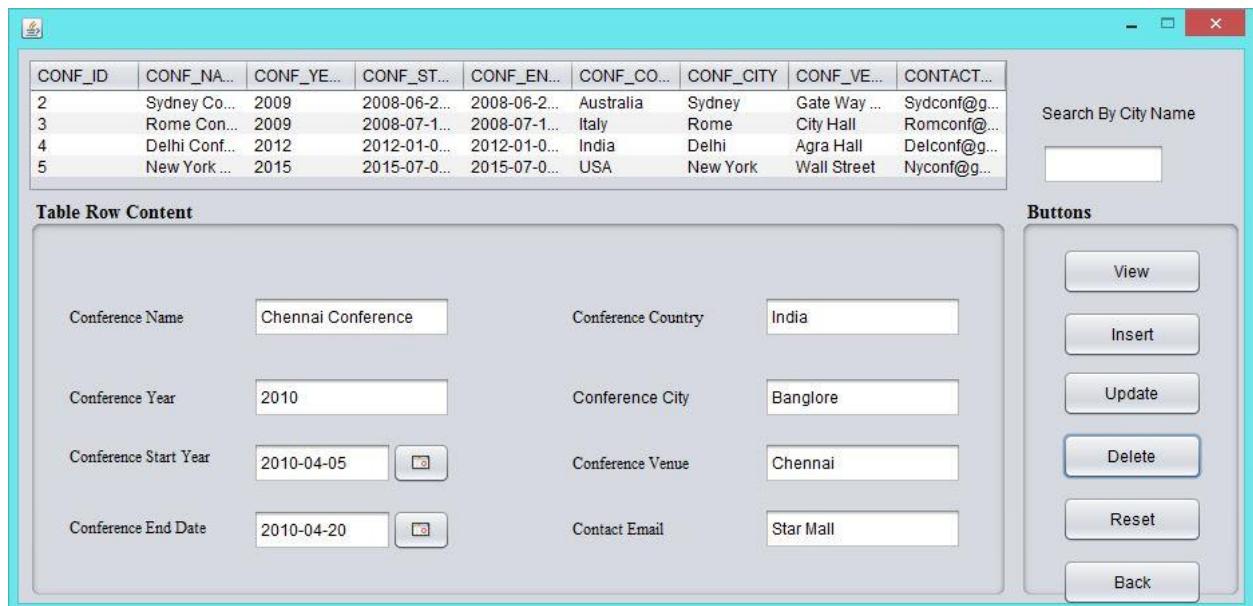
CONF_ID	CONF_NA...	CONF_YE...	CONF_ST...	CONF_EN...	CONF_CO...	CONF_CITY	CONF_VE...	CONTACT...
41	Chennai C...	2010	2010-04-0...	2010-04-2...	India	Banglore	Chennai	Star Mall
2	Sydney Co...	2009	2008-06-2...	2008-06-2...	Australia	Sydney	Gate Way ...	Sydconf@...
3	Rome Con...	2009	2008-07-1...	2008-07-1...	Italy	Rome	City Hall	Romconf...
4	Delhi Conf...	2012	2012-01-0...	2012-01-0...	India	Delhi	Agra Hall	Delconf@...
5	New York	2015	2015-07-0...	2015-07-0...	USA	New York	Wall Street	Nycconf@...

Buttons:

- View
- Insert
- Update
- Delete
- Reset
- Back

Search By City Name:

Data in the table after the execution of Command:



33. Code to select a row in Conference table by clicking on it:

```

private void Conference_tableMouseClicked(java.awt.event.MouseEvent evt) {
    // Click on row of column
    try
    {
        rowSelected = true;
        int row = Conference_table.getSelectedRow();
        String Table_click = (Conference_table.getModel().getValueAt(row,
0).toString());
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn = abc.getConn();
        String sql ="Select * from Conference where Conf_ID =
"+Table_click+"";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
        if(rs.next())
        {
            String add2 = rs.getString("Conf_Name");
            txt_conf_name.setText(add2);
            String add3 = rs.getString("Conf_Year");
            txt_conf_year.setText(add3);
            Date add4 = rs.getDate("Conf_Start_Date");
            Conf_Start_Date.setDate(add4);
            Date add5 = rs.getDate("Conf_End_Date");
            Conf_End_Date.setDate(add5);
            String add6 = rs.getString("Conf_Country");
    
```

```

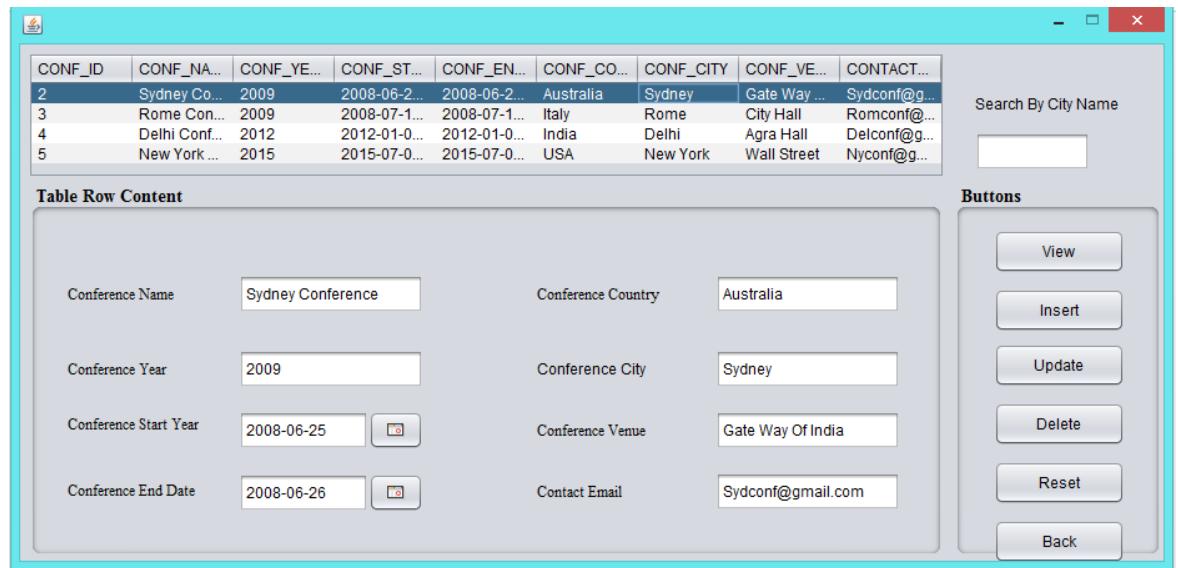
        txt_conf_country.setText(add6);
        String add7 = rs.getString("Conf_City");
        txt_conf_city.setText(add7);
        String add8 = rs.getString("Conf_Venue");
        txt_conf_venue.setText(add8);
        String add9 = rs.getString("Contact_Email");
        txt_email.setText(add9);
    }

}

catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

When you click on a row the values in the row are popped in textboxes and comboboxes:



34. Code for selecting rows by Conference Name:

```

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn = abc.getConn();

        String sql ="Select * from Conference where conf_city = ?";

```

```

pst = conn.prepareStatement(sql);
pst.setString(1,txtSearch.getText());
rs = pst.executeQuery();
Conference_table.setModel(DbUtils.resultSetToTableModel(rs));
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

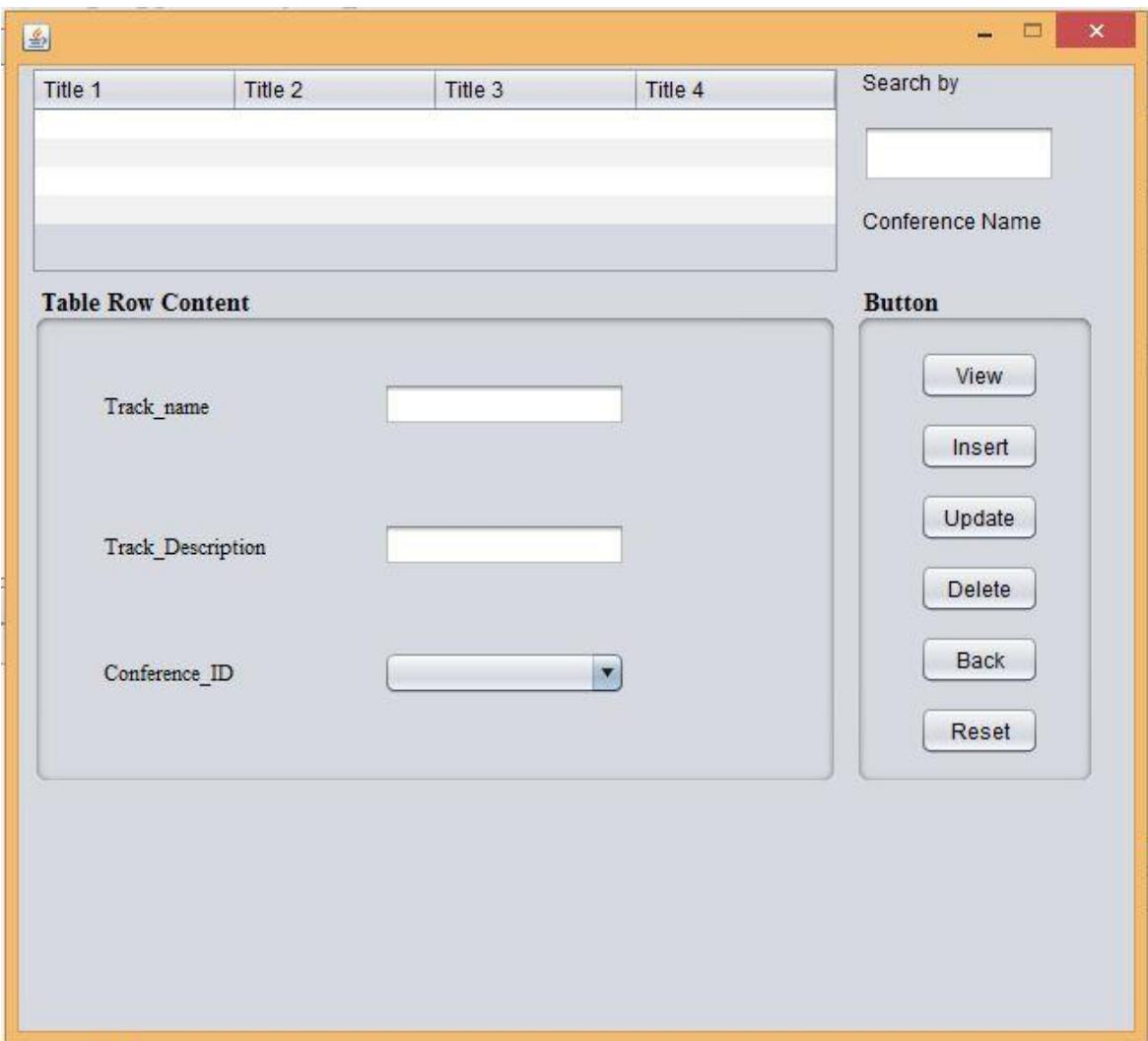
```

When you type conference city name in text search field the rows with the same city name are seen in table:

The screenshot shows a Java Swing application window titled "Conference Management System". The window has a title bar with standard window controls (minimize, maximize, close). Below the title bar is a toolbar with several icons. The main area is divided into sections:

- Table Row Content:** A table showing conference data with columns: CONF_ID, CONF_NA..., CONF_YE..., CONF_ST..., CONF_EN..., CONF_CO..., CONF_CITY, CONF_VE..., and CONTACT... . The data includes rows for Sydney Conference (CONF_ID 2), Rome Conference (CONF_ID 3), Delhi Conference (CONF_ID 4), and New York Conference (CONF_ID 5).
- Search By City Name:** A text input field for searching by city name.
- Buttons:** A vertical stack of buttons labeled View, Insert, Update, Delete, Reset, and Back.
- Form Fields:** A group of input fields for editing conference details:
 - Conference Name: Sydney Conference
 - Conference Country: Australia
 - Conference Year: 2009
 - Conference City: Sydney
 - Conference Start Year: 2008-06-25
 - Conference Venue: Gate Way Of India
 - Conference End Date: 2008-06-26
 - Contact Email: Sydconf@gmail.com

15. Track Frame:



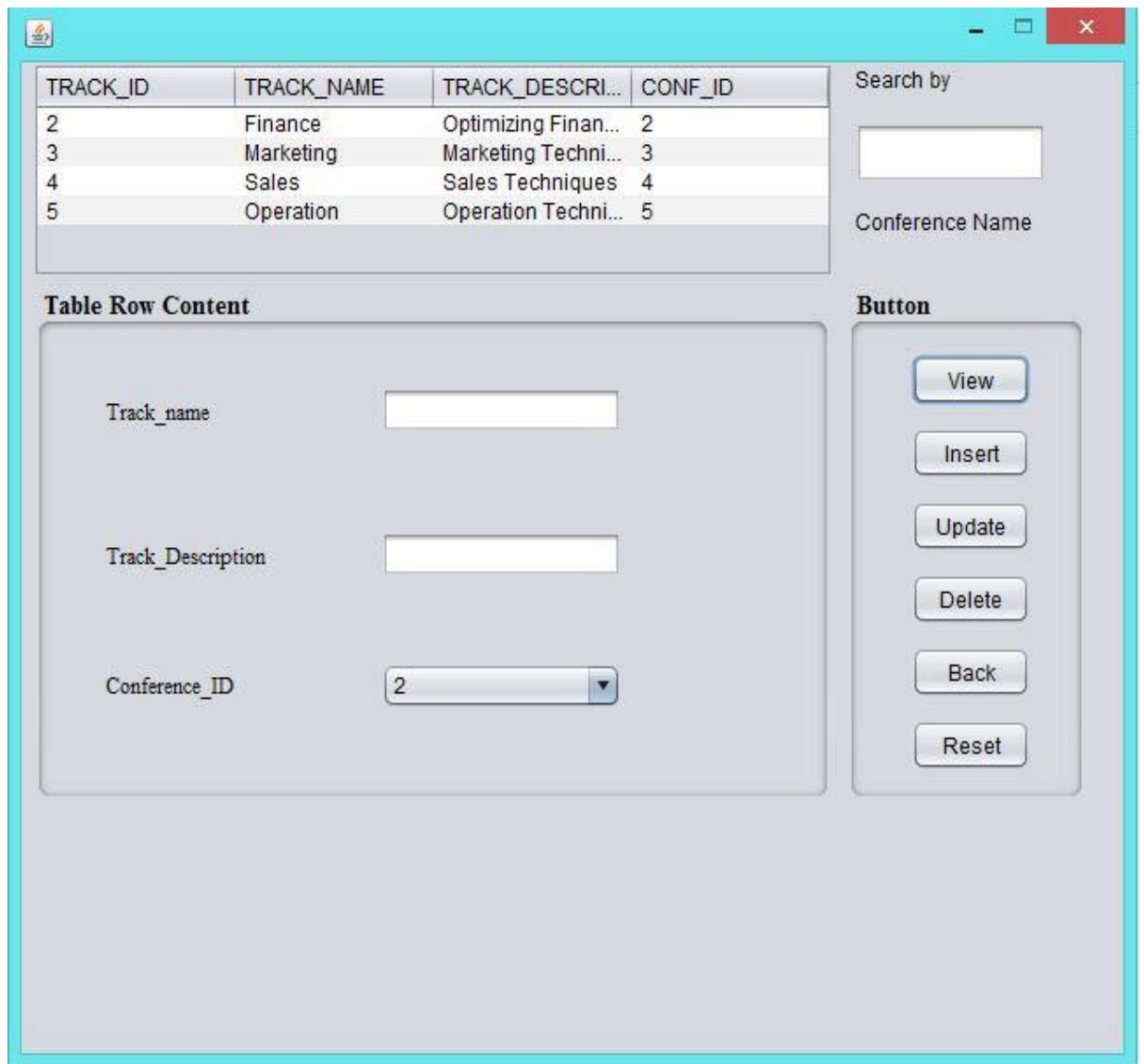
35. Code to view the content of Track table:

```
private void View_Table_Track()
{
    try
    {
        Jayesh_Assignment abc1 = new Jayesh_Assignment();
        Connection conn = abc1.getConn();
        Connection conn1 = abc1.getConn1();
        String sql ="Select * from Track@FIT5148B";
        String sql1 ="Select * from Conference@FIT5148A";
```

```
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery(sql);
        Conference_Track_Table.setModel(DbUtils.resultSetToTableModel(rs));
        pst1 = conn.prepareStatement(sql1);
        rs1 = pst.executeQuery(sql1);
        com_Conf_ID.removeAllItems();
        while (rs1.next())
        {
            String conf_id = rs1.getString("Conf_ID");
            com_Conf_ID.addItem(conf_id);
        }

    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
}
```

Click on View Button to see the table content:



36. Code to reset the user interface value to null:

```
public void clearAll()
{
    txt_track_name.setText("");
    txt_track_description.setText("");
    com_Conf_ID.setSelectedIndex(0);
}
```

37. Code to insert row in Track table:

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
```

```

// TODO add your handling code here:
try
{
    Jayesh_Assignment abc = new Jayesh_Assignment();
    Connection conn1 = abc.getConn1();
    String sql ="Insert into Track@FIT5148B (Track_ID, Track_NAME,
    Track_Description, CONF_ID) values
    (track_seq.nextval"+",""+txt_track_name.getText()+"','"++txt_track_description.getText()+"",
    "+com_Conf_ID.getSelectedItem()+"");
    System.out.println(sql);
    pst = conn1.prepareStatement(sql);
    pst.execute();
    JOptionPane.showMessageDialog(null,"Inserted");

}
catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Track();
}

```

Add value in user interface and then press insert button and then check the table content. The values will be inserted:

Data in the table before the execution of Command:

TRACK_ID	TRACK_NAME	TRACK_DESCRI...	CONF_ID
2	Finance	Optimizing Finan...	2
3	Marketing	Marketing Techni...	3
4	Sales	Sales Techniques	4
5	Operation	Operation Techni...	5

Search by

Conference Name

Table Row Content

Track_name	<input type="text" value="Accounts"/>
Track_Description	<input type="text" value="Accounts Description"/>
Conference_ID	<input type="text" value="3"/>

Button

Data in the table after the execution of Command:

TRACK_ID	TRACK_NAME	TRACK_DESCRI...	CONF_ID
46	Accounts	Accounts Descri...	3
2	Finance	Optimizing Finan...	2
3	Marketing	Marketing Techni...	3
4	Sales	Sales Techniques	4
5	Operation	Operation Techni...	5

Search by

Conference Name

Table Row Content

Track_name	<input type="text" value="Accounts"/>
Track_Description	<input type="text" value="Accounts Description"/>
Conference_ID	<input type="text" value="2"/>

Button

38. Code to update row in Track table:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        try
        {
            int row = Track_Table.getSelectedRow();
```

```

String Table_click = (Track_Table.getModel().getValueAt(row,
0).toString());
    Jayesh_Assignment abc = new Jayesh_Assignment();
    Connection conn1 = abc.getConn1();
    //SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");
    String d1 = txt_track_name.getText();
    String d2 = txt_track_description.getText();
    String d3 = (String) com_Conf_ID.getSelectedItem();
    String sql = "Update Track set Track_name ='" + d1 + "', Track_description
                ='" + d2 + "', Conf_ID ='" + d3 + "' where Track_ID ='" +Table_click+ "'";
    System.out.println(sql);
    pst = conn1.prepareStatement(sql);
    pst.execute();
    JOptionPane.showMessageDialog(null,"Updated");
}
catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Track();
rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select the row you want to update and then enter the new values that you want to update and then press update:

Data in the table before the execution of Command:

TRACK_ID	TRACK_NAME	TRACK_DESCRI...	CONF_ID
46	Accounts	Accounts Descri...	3
2	Finance	Optimizing Finan...	2
3	Marketing	Marketing Techni...	3
4	Sales	Sales Techniques	4
5	Operation	Operation Techni...	5

Search by

Conference Name

Table Row Content

Track_name	<input type="text" value="Financial Management"/>
Track_Description	<input type="text" value="Optimizing Finances"/>
Conference_ID	<input type="text" value="2"/>

Button

Data in the table after the execution of Command:

TRACK_ID	TRACK_NAME	TRACK_DESCRI...	CONF_ID
46	Accounts	Accounts Descri...	3
2	Financial Manag...	Optimizing Finan...	2
3	Marketing	Marketing Techni...	3
4	Sales	Sales Techniques	4
5	Operation	Operation Techni...	5

Search by

Conference Name

Table Row Content

Track_name	Financial Management
Track_Description	Optimizing Finances
Conference_ID	<input type="text" value="2"/>

Button

39. Code to delete row in Track table:

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        int p = JOptionPane.showConfirmDialog(null, "Do you really want to
delete","Delete",JOptionPane.YES_NO_OPTION );
        if( p == 0)
        {
            try
            {
                Jayesh_Assignment abc = new Jayesh_Assignment();

```

```

        Connection conn1 = abc.getConn1();
        int row = Track_Table.getSelectedRow();
        String Table_click = (Track_Table.getModel().getValueAt(row,
0).toString());
        String sql = "Delete from Track@FIT5148B where Track_ID
= '"+Table_click+"'";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        JOptionPane.showMessageDialog(null, "Row is deleted");
    }
    catch (SQLException | HeadlessException e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
    View_Table_Track();
    rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select a row and then press delete:

Data in the table before the execution of Command:

TRACK_ID	TRACK_NAME	TRACK_DESCRI...	CONF_ID
46	Accounts	Accounts Descri...	3
2	Financial Manag...	Optimizing Finan...	2
3	Marketing	Marketing Techni...	3
4	Sales	Sales Techniques	4
5	Operation	Operation Techni...	5

Search by

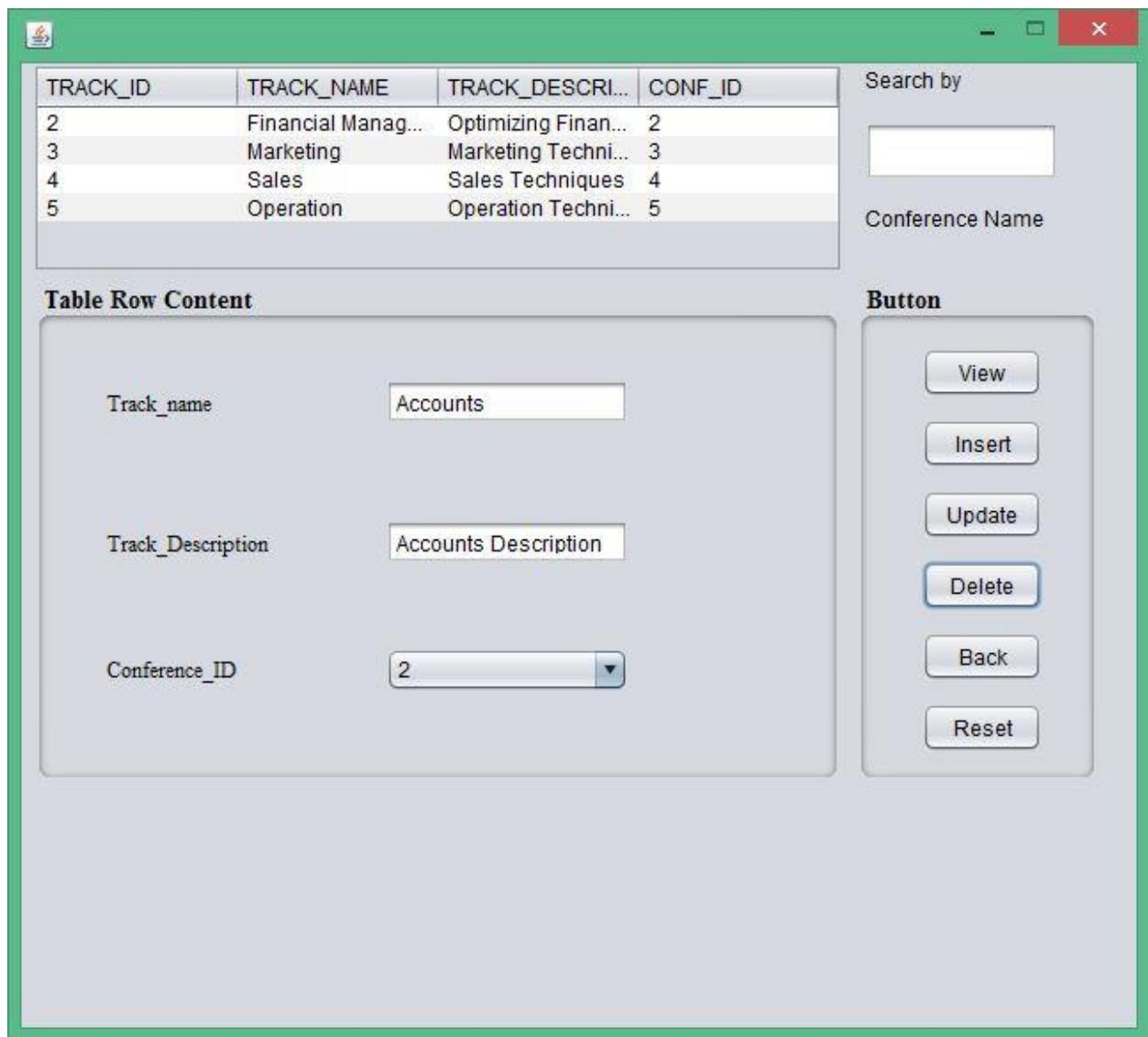
Conference Name

Table Row Content

Track_name	<input type="text" value="Accounts"/>
Track_Description	<input type="text" value="Accounts Description"/>
Conference_ID	<input type="text" value="3"/>

Button

Data in the table after the execution of Command:



40. Code to select a row in Track table by clicking on it:

```
private void Conference_Track_TableMouseClicked(java.awt.event.MouseEvent evt)
{
    // TODO add your handling code here:
    try
    {
        rowSelected = true;
        int row = Conference_Track_Table.getSelectedRow();
        String Table_click = (Conference_Track_Table.getModel().getValueAt(row,
0).toString());
        Jayesh_Assignment abc = new Jayesh_Assignment();
```

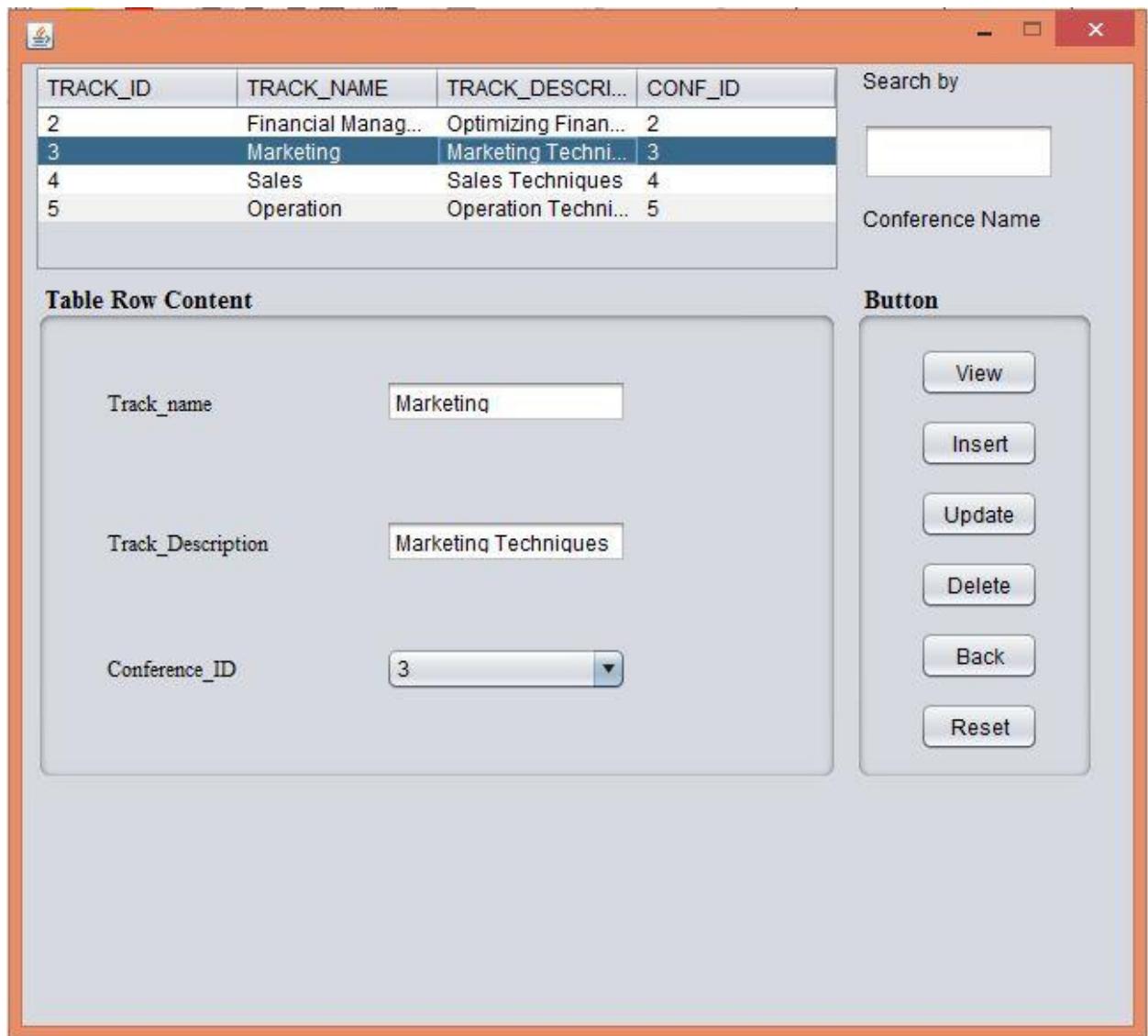
```

Connection conn1 = abc.getConn1();
String sql ="Select * from ConferenceTrack where Track_ID =
"+Table_click+"";
pst = conn1.prepareStatement(sql);
rs = pst.executeQuery();
if(rs.next())
{
    String add2 = rs.getString("Track_Name");
    txt_track_name.setText(add2);
    String add3 = rs.getString("Track_Description");
    txt_track_description.setText(add3);
    String add4 = rs.getString("Conf_ID");
    com_Conf_ID.setSelectedItem(add4);
}

catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

When you click on a row the values in the row are popped in textboxes and combo boxes:



41. Code for selecting rows by Conference Name:

```
private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    try {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn = abc.getConn();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from Track@FIT5148B where Conf_ID = (Select
conf_ID from Conference@FIT5148A where Conf_name = ?)";
        pst1 = conn1.prepareStatement(sql);
```

```

pst1.setString(1,txtSearch.getText());
rs1 = pst1.executeQuery();
Conference_Track_Table.setModel(DbUtils.resultSetToTableModel(rs1));
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

When you type conference name in text search field the rows with the same city name are seen in table:

The screenshot shows a Java Swing application window with an orange border. On the left, there is a table titled "Table Row Content" with columns: TRACK_ID, TRACK_NAME, TRACK_DESCRIP..., and CONF_ID. A single row is visible with values: 2, Financial Manag..., Optimizing Finan..., and 2. To the right of the table is a search bar labeled "Search by" containing the text "/dney Conference". Below the search bar is a label "Conference Name". On the far right, there is a vertical stack of buttons labeled "Button" with options: View, Insert, Update, Delete, Back, and Reset.

TRACK_ID	TRACK_NAME	TRACK_DESCRIP...	CONF_ID
2	Financial Manag...	Optimizing Finan...	2

Table Row Content

Track_name: Marketing

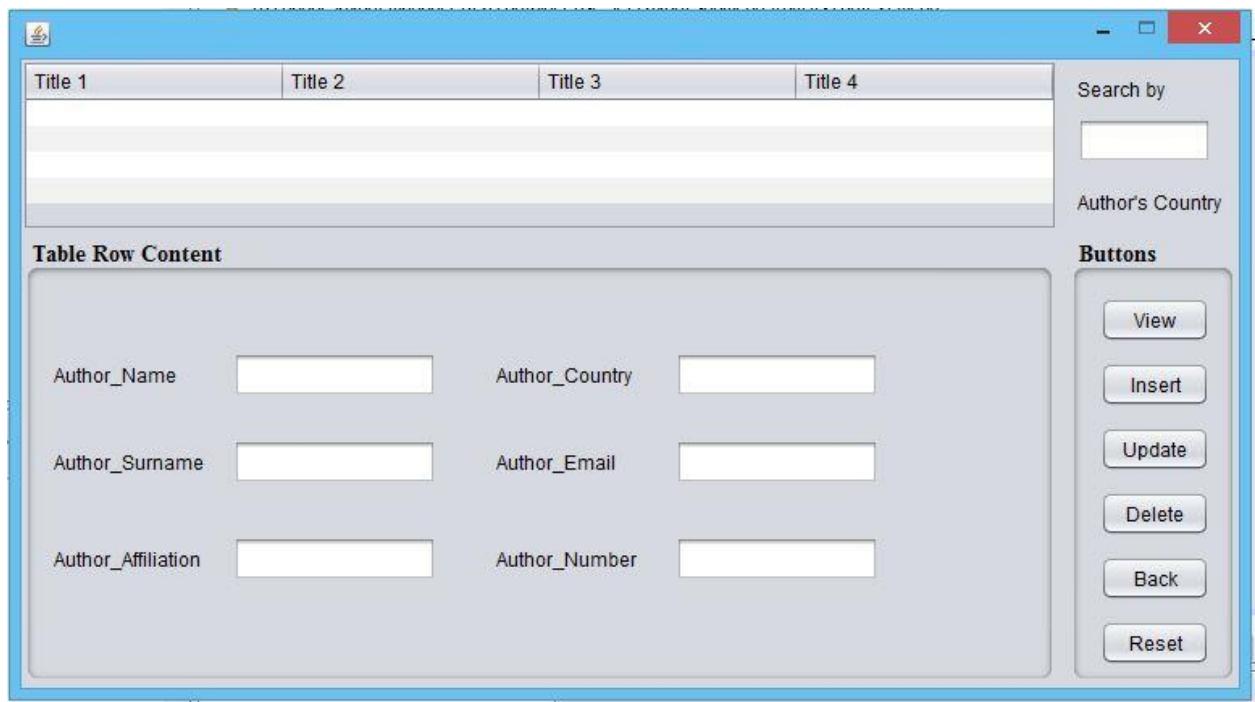
Track_Description: Marketing Techniques

Conference_ID: 3

Button

- View
- Insert
- Update
- Delete
- Back
- Reset

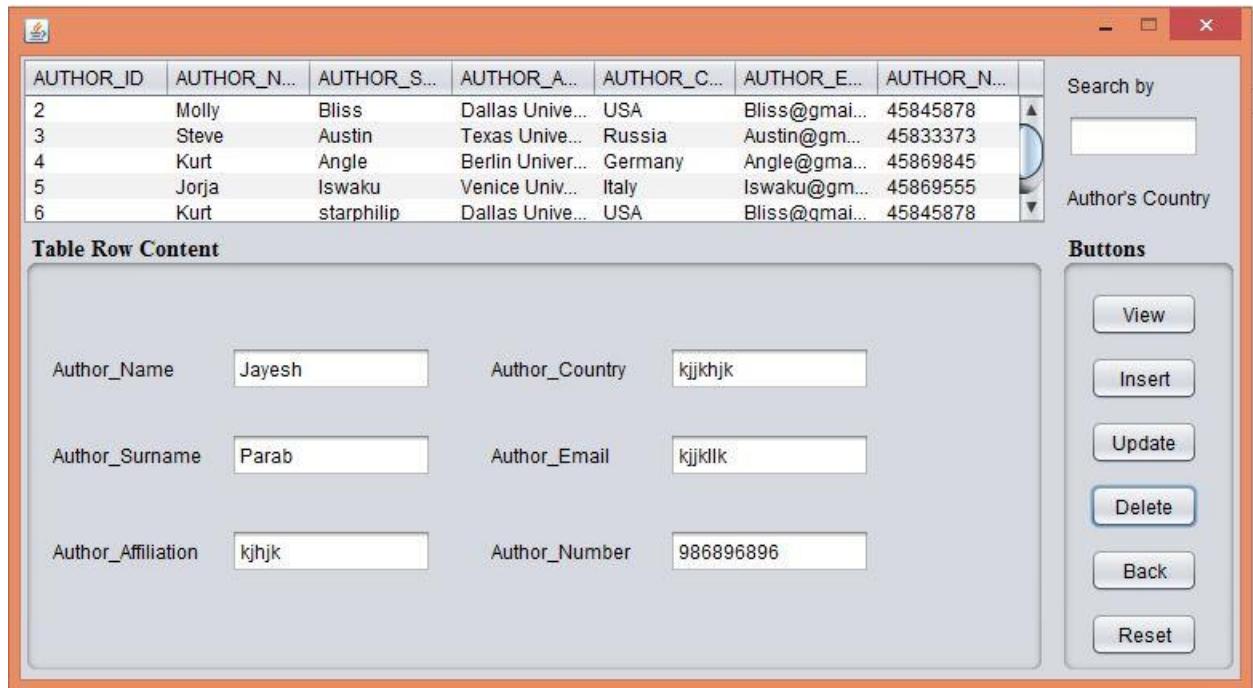
16. Author Frame:



42. Code to view the content of Author table:

```
private void View_Table_Author()
{
    try
    {
        Jayesh_Assignment abc1 = new Jayesh_Assignment();
        Connection conn1 = abc1.getConn1();
        String sql = "Select * from Author@FIT5148B";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        Author_Table.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
}
```

Click on View Button to see the table content:



43. Code to reset the user interface value to null:

```
public void clearAll()
{
    txt_author_name.setText("");
    txt_author_surname.setText("");
    txt_author_affiliation.setText("");
    txt_author_Country.setText("");
    txt_author_email.setText("");
    txt_author_number.setText("");
}
```

44. Code to insert row in Author table:

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
```

```

String sql ="Insert into Author@FIT5148B (Author_ID, Author_name,
Author_Surname, Author_Affiliation, Author_Country, Author_email,
Author_number) values
(Author_seq.nextval"+",""+txt_author_name.getText()+"','"+"+txt_author_surname.getText()+"','"+"+txt_author_affiliation.getText()+"','"+"+txt_author_Country.getText()+"','"+"+txt_author_email.getText()+"','"+"+txt_author_number.getText()+"");
System.out.println(sql);
pst = conn1.prepareStatement(sql);
pst.execute();
JOptionPane.showMessageDialog(null,"Inserted");

}

catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Author();
}

```

Add value in user interface and then press insert button and then check the table content. The values will be inserted:

Data in the table before the execution of Command:

The screenshot shows a Windows application window with a title bar and a main content area. The content area contains a table of author data and a form for adding a new row.

Table Row Content:

AUTHOR_ID	AUTHOR_N...	AUTHOR_S...	AUTHOR_A...	AUTHOR_C...	AUTHOR_E...	AUTHOR_N...
2	Molly	Bliss	Dallas Unive...	USA	Bliss@gmai...	45845878
3	Steve	Austin	Texas Unive...	Russia	Austin@gm...	45833373
4	Kurt	Angle	Berlin Univer...	Germany	Angle@gma...	45869845
5	Jorja	Iswaku	Venice Univ...	Italy	Iswaku@gm...	45869555
6	Kurt	starphilip	Dallas Unive...	USA	Bliss@qmai...	45845878

Buttons:

- View
- Insert
- Update
- Delete
- Back
- Reset

Search by:

Author's Country:

Data in the table after the execution of Command:



45. Code to update row in Author table:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        try
        {
            int row = Author_Table.getSelectedRow();
            String Table_click = (Author_Table.getModel().getValueAt(row,
0).toString());
            Jayesh_Assignment abc = new Jayesh_Assignment();
            Connection conn1 = abc.getConn1();
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");
            String d1 = txt_author_name.getText();
            String d2 = txt_author_surname.getText();
            String d3 = txt_author_affiliation.getText();
            String d4 = txt_author_Country.getText();
            String d5 = txt_author_email.getText();
            String d6 = txt_author_number.getText();
            String sql = "Update Author@FIT5148B set Author_name ='" + d1 + "'",
Author_Surname ='" + d2 + "', Author_Affiliation ='" + d3 + "','" + " Author_Country
```

```

= "" + d4 + "," + "Author_email = " + d5 + "," + "Author_number = " + d6 + " where
Author_ID = "+Table_click+";
System.out.println(sql);
pst = conn1.prepareStatement(sql);
pst.execute();
JOptionPane.showMessageDialog(null,"Updated");
}
catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Author();
rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select the row you want to update and then enter the new values that you want to update and then press update:

Data in the table before the execution of Command:

AUTHOR_ID	AUTHOR_N...	AUTHOR_S...	AUTHOR_A...	AUTHOR_C...	AUTHOR_E...	AUTHOR_N...
21	Rey	Mysterio	usf	Mexico	rey@gmail.c...	9868689
2	Molly	Bliss	Dallas Univ...	USA	Bliss@gmai...	45845878
3	Steve	Austin	Texas Unive...	Russia	Austin@gm...	45833373
4	Kurt	Angle	Berlin Univer...	Germany	Angle@gma...	45869845
5	Jorja	Iswaku	Venice Univ...	Italy	Iswaku@gm...	45869555

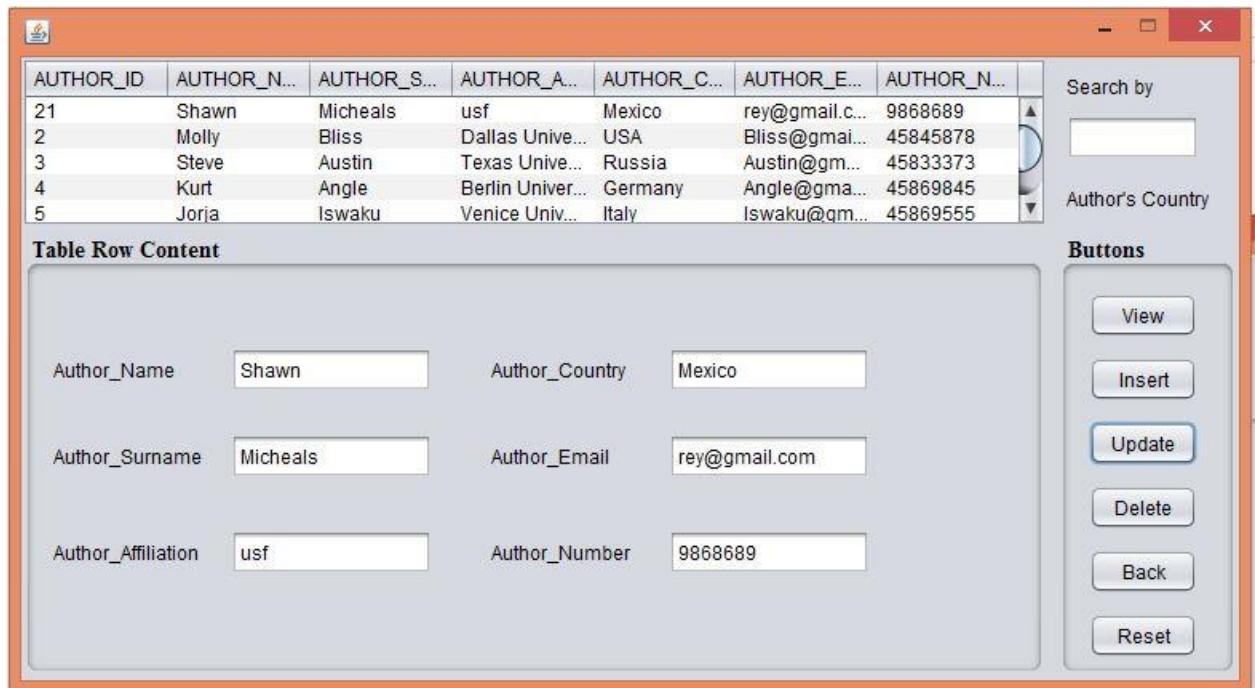
Table Row Content

Author_Name	Shawn	Author_Country	Mexico
Author_Surname	Micheals	Author_Email	rey@gmail.com
Author_Affiliation	usf	Author_Number	9868689

Buttons

-
-
-
-
-
-

Data in the table after the execution of Command:



46. Code to delete row in Author table:

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        int p = JOptionPane.showConfirmDialog(null, "Do you really want to
delete","Delete",JOptionPane.YES_NO_OPTION );
        if( p == 0)
        {
            try
            {
                Jayesh_Assignment abc = new Jayesh_Assignment();
                Connection conn1 = abc.getConn1();
                int row = Author_Table.getSelectedRow();
                String Table_click = (Author_Table.getModel().getValueAt(row,
0).toString());
                String sql = "Delete from Author where Author_ID =" +Table_click+ "";
                pst = conn1.prepareStatement(sql);
                rs = pst.executeQuery();
            }
        }
    }
}
```

```

        JOptionPane.showMessageDialog(null, "Row is deleted");
    }
    catch (SQLException | HeadlessException e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
    View_Table_Author();
    rowSelected = false;
}
}
else
{
    JOptionPane.showMessageDialog(null, "Please Select a row");
}
}

```

Select a row and then press delete:

Data in the table before the execution of Command:

The screenshot shows a Java Swing application window titled 'Table Row Content'. On the left is a JTable displaying data for five authors. On the right is a panel containing a search bar and several buttons.

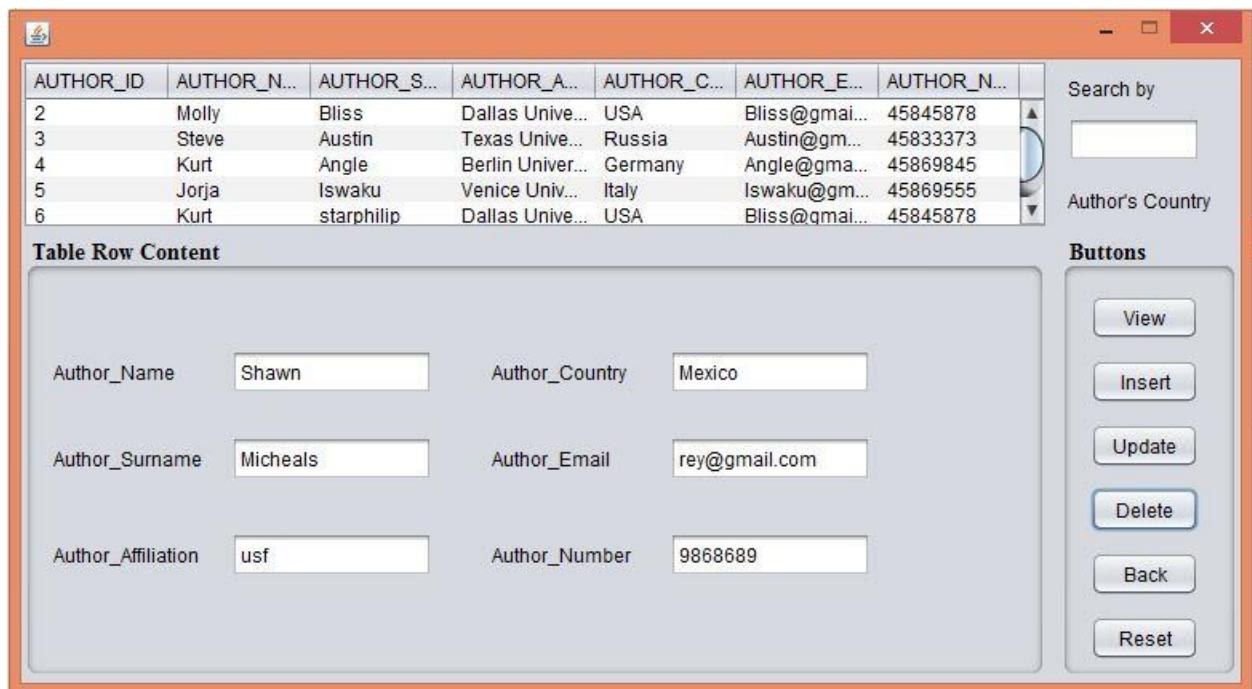
JTable Data:

AUTHOR_ID	AUTHOR_N...	AUTHOR_S...	AUTHOR_A...	AUTHOR_C...	AUTHOR_E...	AUTHOR_N...
21	Shawn	Micheals	usf	Mexico	rey@gmail.c...	9868689
2	Molly	Bliss	Dallas Unive...	USA	Bliss@gmai...	45845878
3	Steve	Austin	Texas Unive...	Russia	Austin@gm...	45833373
4	Kurt	Angle	Berlin Univer...	Germany	Angle@gma...	45869845
5	Joria	Iswaku	Venice Univ...	Italy	Iswaku@gm...	45869555

Buttons Panel:

- Search by: [Search Bar]
- Author's Country: [Spinner]
- Buttons: View, Insert, Update, Delete, Back, Reset

Data in the table after the execution of Command:



47. Code to select a row in Author table by clicking on it:

```
private void Author_TableMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try {
        rowSelected = true;
        int row = Author_Table.getSelectedRow();
        String Table_click = (Author_Table.getModel().getValueAt(row,
0).toString());
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql = "Select * from Author where Author_ID = "+Table_click+"";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        if(rs.next())
        {
            String add2 = rs.getString("Author_Name");
            txt_author_name.setText(add2);
            String add3 = rs.getString("Author_Surname");
            txt_author_surname.setText(add3);
            String add4 = rs.getString("Author_Affiliation");
            txt_author_affiliation.setText(add4);
            String add5 = rs.getString("Author_Country");
            txt_author_Country.setText(add5);
        }
    }
}
```

```

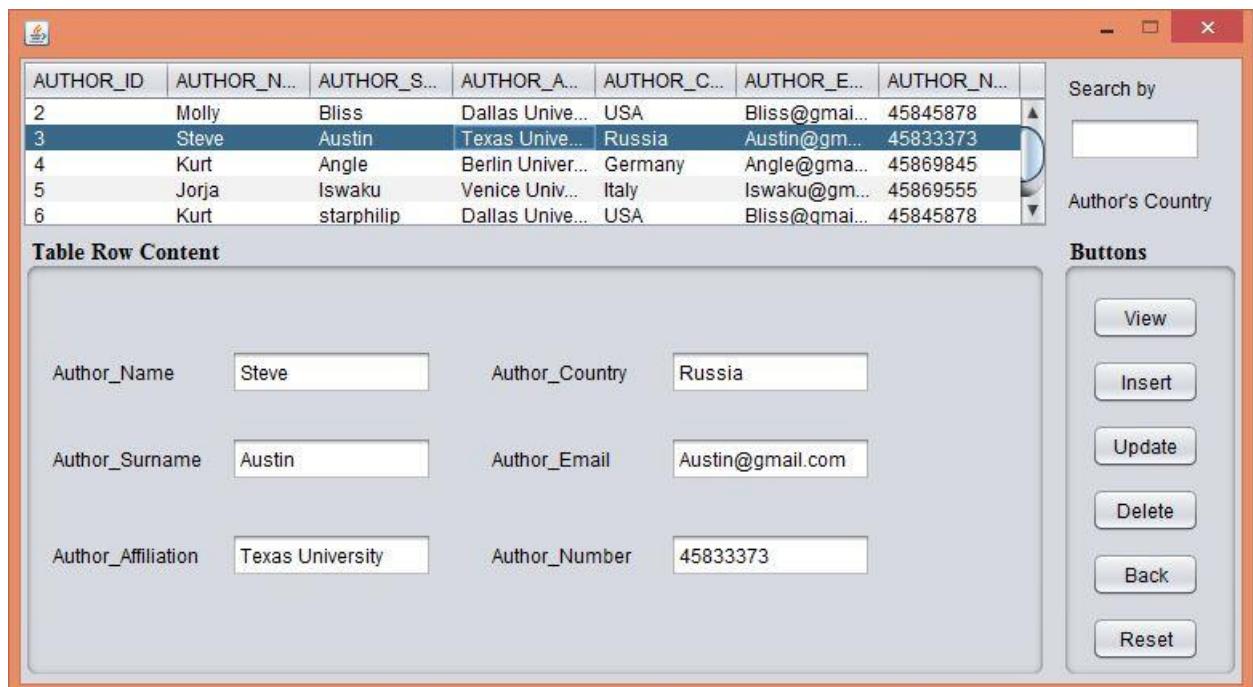
        String add6 = rs.getString("Author_Email");
        txt_author_email.setText(add6);
        String add7 = rs.getString("Author_number");
        txt_author_number.setText(add7);
    }

}

catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

When you click on a row the values in the row are popped in textboxes and combo boxes:



48. Code for selecting rows by Author Country:

```

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

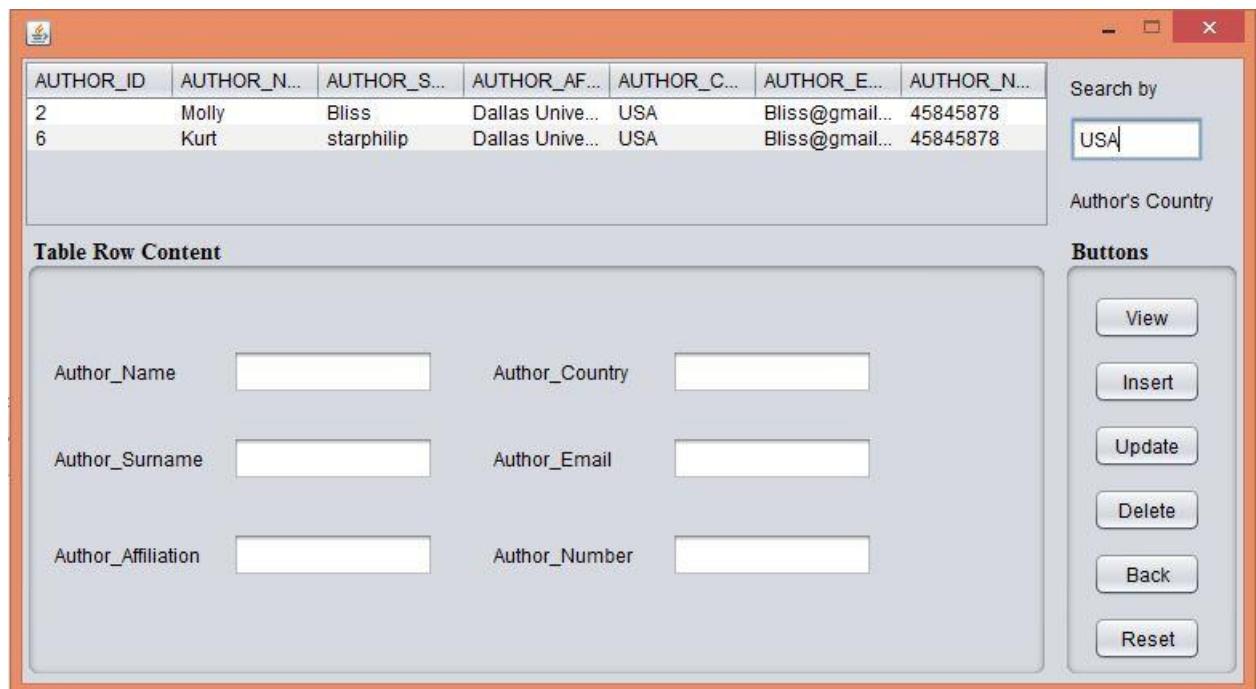
```

```

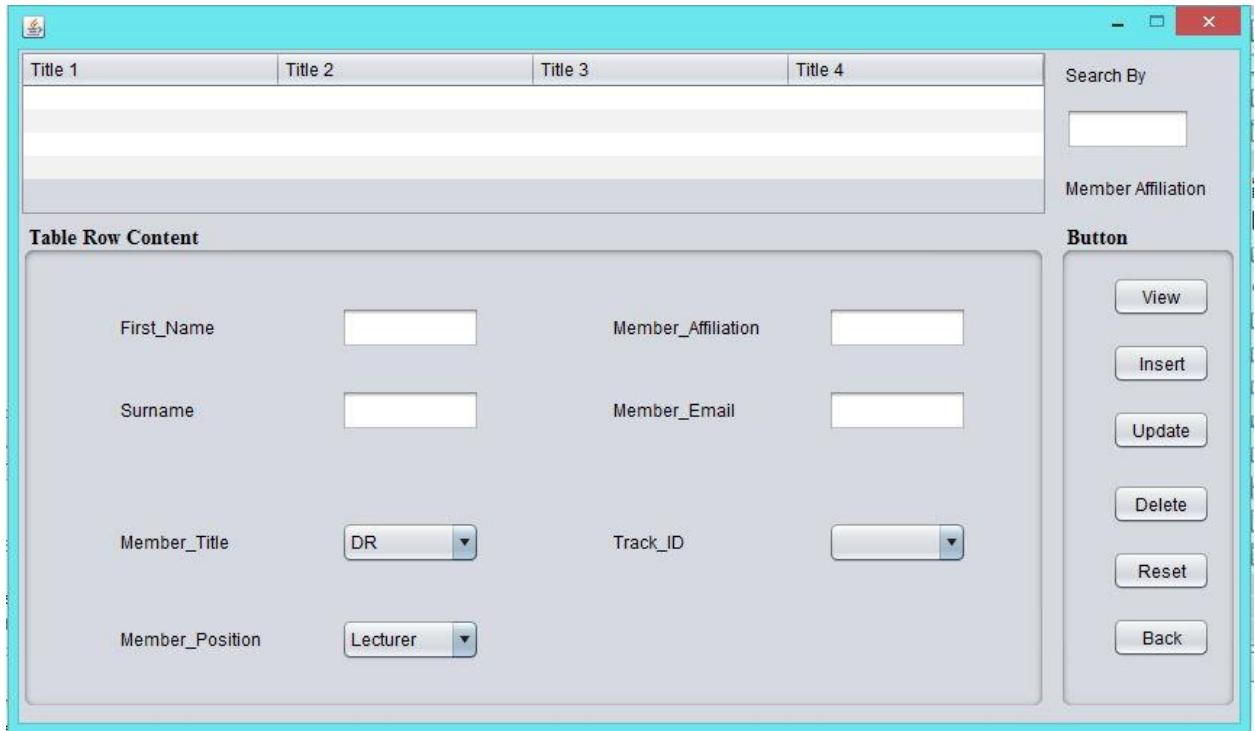
try
{
    Jayesh_Assignment abc = new Jayesh_Assignment();
    Connection conn = abc.getConn();
    String sql ="Select * from Author where Author_Country = ?";
    pst = conn.prepareStatement(sql);
    pst.setString(1,txtSearch.getText());
    rs = pst.executeQuery();
    Author_Table.setModel(DbUtils.resultSetToTableModel(rs));
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

When you type Author's country in text search field the rows with the same country name are seen in table:



17. PCMember Frame



49. Code to view the content of PCMember table:

```
private void View_Table_PCMember()
{
    try
    {
        Jayesh_Assignment abc1 = new Jayesh_Assignment();
        Connection conn1 = abc1.getConn1();
        String sql ="Select * from PCMember@FIT5148B";
        String sql1 ="Select * from Track@FIT5148B";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery(sql);
        PCMember_Table.setModel(DbUtils.resultSetToTableModel(rs));
        pst1 = conn1.prepareStatement(sql1);
        rs1 = pst1.executeQuery(sql1);
        combo_Track_ID.removeAllItems();
        while (rs1.next())
        {
            String track_id = rs1.getString("Track_ID");
            
```

```
    combo_Track_ID.addItem(track_id);
}
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}
```

MEMBER_ID	FIRST_NAME	SURNAME	MEMBER_TI...	MEMBER_P...	MEMBER_A...	MEMBER_E...	TRACK_ID
7	jack	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
2	Ron	Wisley	Prof	Lecturer	Banglore Un...	Wisley@gm...	2
3	Hermiane	Granger	DR	Researcher	Chennai Uni...	Granger@g...	3
4	Albus	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
5	Severus	Snape	Prof	Researcher	Delhi Univer...	Snape@gm...	5

Table Row Content

First_Name

Surname

Member_Title DR

Member_Position Lecturer

Member_Affiliation

Member_Email

Track_ID 2

Search By

Member Affiliation

Button

50. Code to reset the user interface value to null:

```
public void clearAll()
{
    txt_member_first_name.setText("");
    txt_member_surname.setText("");
    txt_member_first_name.setText("");
    combo_Member_Title.setSelectedIndex(0);
    combo_Member_Position.setSelectedIndex(0);
    txt_member_affiliation.setText("");
    txt_member_email.setText("");
    combo_Track_ID.setSelectedIndex(0);
}
```

51. Code to insert row in PCMember table:

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try  
    {  
        Jayesh_Assignment abc = new Jayesh_Assignment();  
        Connection conn1 = abc.getConn1();  
        String sql ="Insert into PCMember@FIT5148B (Member_ID, First_name,  
        Surname, Member_title, Member_position, Member_affiliation,  
        Member_email, Track_ID) values  
        (member_seq.nextval"+",""+txt_member_first_name.getText()+"','"+  
        txt_member_surname.getText()+"','"++combo_Member_Title.getSelectedItem()  
        +"','"++combo_Member_Position.getSelectedItem()+"','"++txt_member_affili  
        ation.getText()+"','"++txt_member_email.getText()+"','"++combo_Track_ID.  
        getSelectedItem()+"");  
        System.out.println(sql);  
        pst = conn1.prepareStatement(sql);  
        pst.execute();  
        JOptionPane.showMessageDialog(null,"Inserted");  
  
    }  
    catch (SQLException | HeadlessException e)  
    {  
        JOptionPane.showMessageDialog(null,e);  
    }  
    View_Table_PCMember();  
}
```

Add value in user interface and then press insert button and then check the table content. The values will be inserted:

Data in the table before the execution of Command:

MEMBER_ID	FIRST_NAME	SURNAME	MEMBER_TI...	MEMBER_P...	MEMBER_A...	MEMBER_E...	TRACK_ID
7	jack	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
2	Ron	Wisley	Prof	Lecturer	Banglore Un...	Wisley@gm...	2
3	Hermiane	Granger	DR	Researcher	Chennai Uni...	Granger@g...	3
4	Albus	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
5	Severus	Snape	Prof	Researcher	Delhi Univer...	Snape@gm...	5

Table Row Content

First_Name	<input type="text" value="Jonty"/>	Member_Affiliation	<input type="text" value="usf"/>
Surname	<input type="text" value="Rhoades"/>	Member_Email	<input type="text" value="jonty@gmail.com"/>
Member_Title	<input type="button" value="Prof"/>	Track_ID	<input type="button" value="2"/>
Member_Position	<input type="button" value="Lecturer"/>		

Button

-
-
-
-
-
-

Data in the table after the execution of Command:

MEMBER_ID	FIRST_NAME	SURNAME	MEMBER_TI...	MEMBER_P...	MEMBER_A...	MEMBER_E...	TRACK_ID
7	jack	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
21	Jonty	Rhoades	Prof	Lecturer	usf	jonty@gmail...	2
2	Ron	Wisley	Prof	Lecturer	Banglore Un...	Wisley@gm...	2
3	Hermiane	Granger	DR	Researcher	Chennai Uni...	Granger@g...	3
4	Albus	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
5	Severus	Snape	Prof	Researcher	Delhi Univer...	Snape@gm...	5

Table Row Content

First_Name	<input type="text" value="Jonty"/>	Member_Affiliation	<input type="text" value="usf"/>
Surname	<input type="text" value="Rhoades"/>	Member_Email	<input type="text" value="jonty@gmail.com"/>
Member_Title	<input type="button" value="Prof"/>	Track_ID	<input type="button" value="2"/>
Member_Position	<input type="button" value="Lecturer"/>		

Button

-
-
-
-
-
-

52. Code to update row in PCMember table:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if (rowSelected == true)  
    {  
        try  
        {  
            int row = PCMember_Table.getSelectedRow();  
            String Table_click = (PCMember_Table.getModel().getValueAt(row,  
0).toString());  
            Jayesh_Assignment abc = new Jayesh_Assignment();  
            Connection conn1 = abc.getConn1();  
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");  
            String d1 = txt_member_first_name.getText();  
            String d2 = txt_member_surname.getText();  
            String d3 = (String) combo_Member_Title.getSelectedItem();  
            String d4 = (String) combo_Member_Position.getSelectedItem();  
            String d5 = txt_member_affiliation.getText();  
            String d6 = txt_member_email.getText();  
            String d7 = (String) combo_Track_ID.getSelectedItem();  
            String sql = "Update PCMember@FIT5148B set First_name ='" + d1 + "',  
Surname ='" + d2 + "', Member_title ='" + d3 + "','" + Member_position ='" + d4 +  
"',' + Member_affiliation ='" + d5 + "','" + Member_email ='" + d6 + "','" +  
"Track_ID ='" + d7 + "' where Member_ID = '" + Table_click + "'";  
            System.out.println(sql);  
            pst = conn1.prepareStatement(sql);  
            pst.execute();  
            JOptionPane.showMessageDialog(null,"Updated");  
        }  
        catch (SQLException | HeadlessException e)  
        {  
            JOptionPane.showMessageDialog(null,e);  
        }  
        View_Table_PCMember();  
        rowSelected = false;  
    }  
    else  
    {  
        JOptionPane.showMessageDialog(null,"Please Select a row");  
    }  
}
```

```

    }
}

```

Select the row you want to update and then enter the new values that you want to update and then press update:

Data in the table before the execution of Command:

The screenshot shows a Windows application window with a title bar and a main content area. The content area contains a table of member data and a form for updating a specific row.

Table Row Content:

Member_ID	First_Name	Surname	Member_Title	Member_Position	Member_Affiliation	Member_Email	Track_ID
7	jack	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
21	Jonty	Rhoades	Prof	Lecturer	usf	jonty@gmail...	2
2	Ron	Wisley	Prof	Lecturer	Banglore Un...	Wisley@gm...	2
3	Hermiane	Granger	DR	Researcher	Chennai Uni...	Granger@g...	3
4	Albus	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
5	Graeme	Smith	Prof	Researcher	Dalhousie Uni...	Graeme...	5

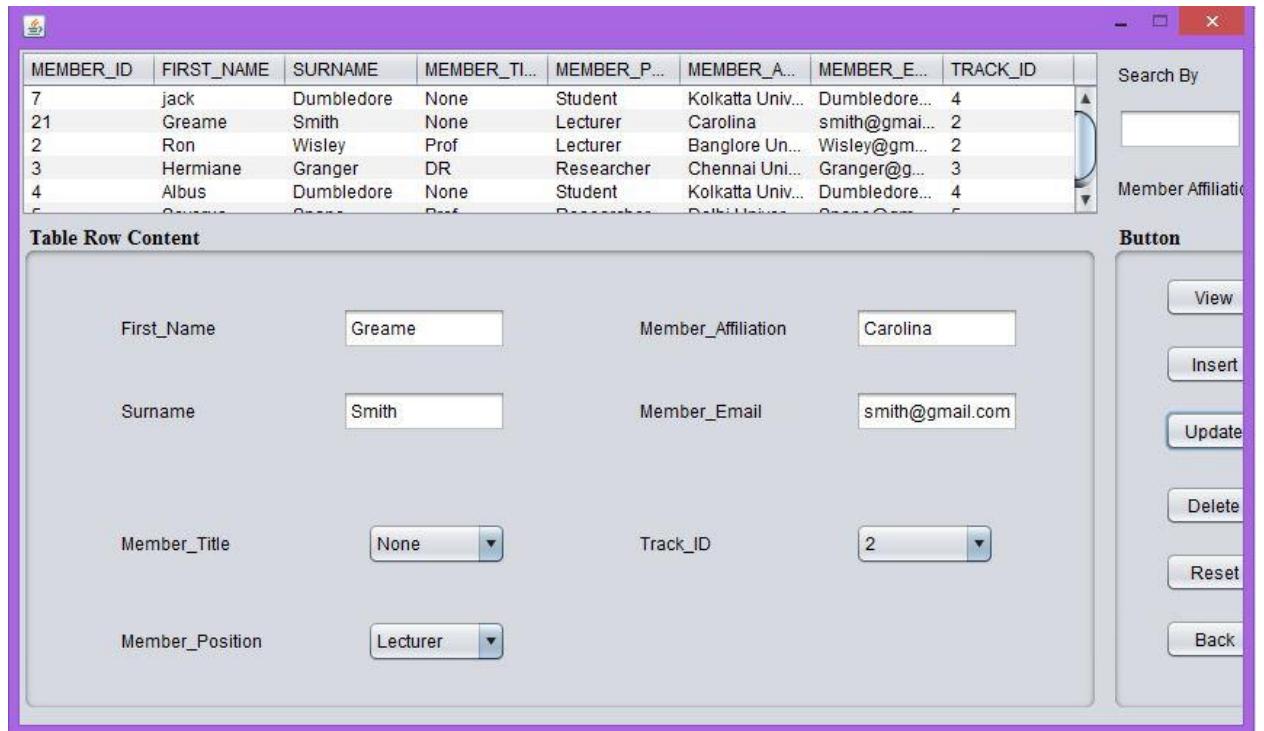
Form Fields (Row 5 Selected):

- First_Name: Graeme
- Member_Affiliation: Carolina
- Surname: Smith
- Member_Email: smith@gmail.com
- Member_Title: None
- Track_ID: 2
- Member_Position: Lecturer

Buttons on the right:

- Search By
- Member Affiliation
- Button
- View
- Insert
- Update
- Delete
- Reset
- Back

Data in the table after the execution of Command:



53. Code to delete row in PCMember table:

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        int p = JOptionPane.showConfirmDialog(null, "Do you really want to
delete","Delete",JOptionPane.YES_NO_OPTION );
        if( p == 0)
        {
            try
            {
                Jayesh_Assignment abc = new Jayesh_Assignment();
                Connection conn1 = abc.getConn1();
                int row = PCMember_Table.getSelectedRow();
                String Table_click = (PCMember_Table.getModel()).getValueAt(row,
0).toString());
            }
        }
    }
}
```

```

        String sql = "Delete from PCMember@FIT5148B where Member_ID
        =" +Table_click+ "";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        JOptionPane.showMessageDialog(null, "Row is deleted");
    }
    catch (SQLException | HeadlessException e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
    View_Table_PCMember();
    rowSelected = false;
}
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select a row and then press delete:

Data in the table before the execution of Command:

MEMBER_ID	FIRST_NAME	SURNAME	MEMBER_TL...	MEMBER_P...	MEMBER_A...	MEMBER_E...	TRACK_ID
7	jack	Dumbledore	None	Student	Kolkatta Univ...	Dumbledore...	4
21	Greame	Smith	None	Lecturer	Carolina	smith@gmai...	2
2	Ron	Wisley	Prof	Lecturer	Banglore Uni...	Wisley@gm...	2
3	Hermiane	Granger	DR	Researcher	Chennai Uni...	Granger@g...	3
4	Albus	Dumbledore	None	Student	Kolkatta Univ...	Dumbledore...	4
5	Severus	Snape	Prof	Researcher	Delhi Univer...	Snape@gma...	5

Table Row Content

First_Name	<input type="text" value="Greame"/>	Member_Affiliation	<input type="text" value="Carolina"/>
Surname	<input type="text" value="Smith"/>	Member_Email	<input type="text" value="smith@gmail.com"/>
Member_Title	<input type="button" value="None"/>	Track_ID	<input type="button" value="2"/>
Member_Position	<input type="button" value="Lecturer"/>		

Button

<input type="button" value="View"/>
<input type="button" value="Insert"/>
<input type="button" value="Update"/>
<input type="button" value="Delete"/>
<input type="button" value="Reset"/>
<input type="button" value="Back"/>

Data in the table after the execution of Command:

MEMBER_ID	FIRST_NAME	SURNAME	MEMBER_TL...	MEMBER_P...	MEMBER_AF...	MEMBER_E...	TRACK_ID
7	jack	Dumbledore	None	Student	Kolkatta Univ...	Dumbledore...	4
2	Ron	Wisley	Prof	Lecturer	Banglore Uni...	Wisley@gm...	2
3	Hermiane	Granger	DR	Researcher	Chennai Uni...	Granger@g...	3
4	Albus	Dumbledore	None	Student	Kolkatta Univ...	Dumbledore...	4
5	Severus	Snape	Prof	Researcher	Delhi Univer...	Snape@gma...	5

Table Row Content

First_Name	<input type="text" value="Greame"/>	Member_Affiliation	<input type="text" value="Carolina"/>
Surname	<input type="text" value="Smith"/>	Member_Email	<input type="text" value="smith@gmail.com"/>
Member_Title	<input type="button" value="None"/>	Track_ID	<input type="button" value="2"/>
Member_Position	<input type="button" value="Lecturer"/>		

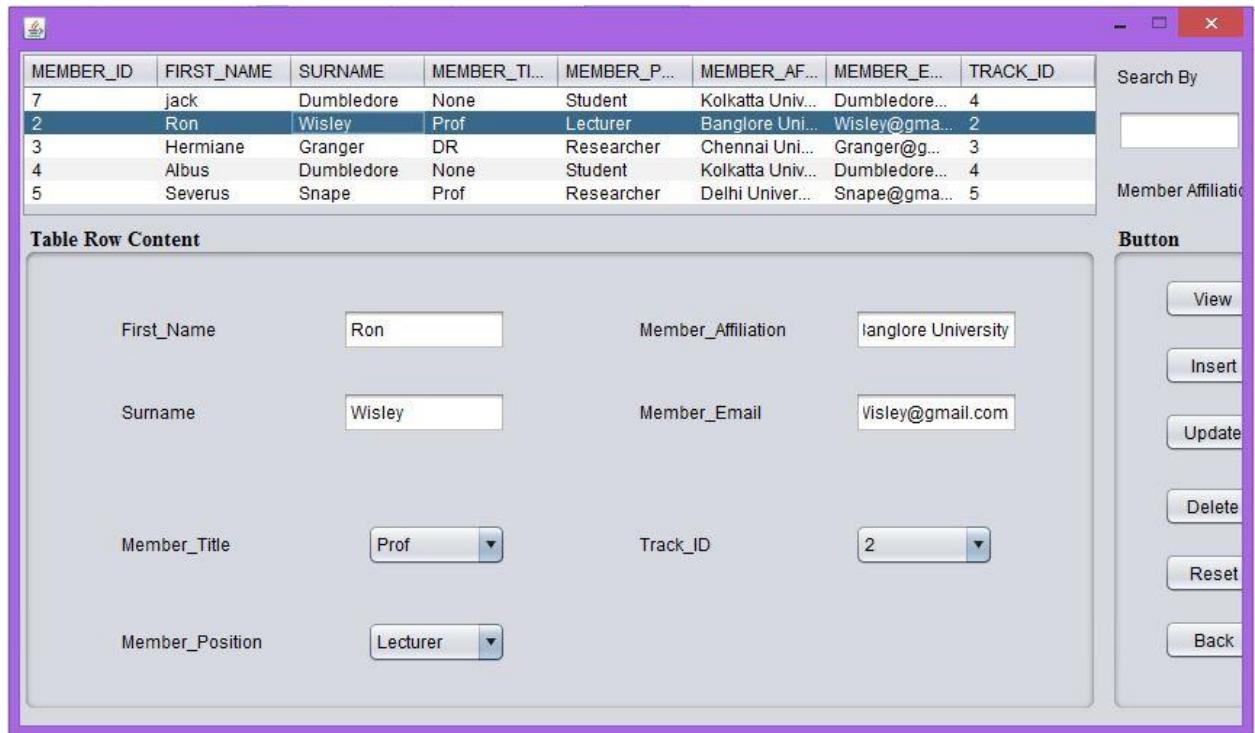
Button

<input type="button" value="View"/>
<input type="button" value="Insert"/>
<input type="button" value="Update"/>
<input type="button" value="Delete"/>
<input type="button" value="Reset"/>
<input type="button" value="Back"/>

54. Code to select a row in PCMember table by clicking on it:

```
private void PCMember_TableMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    try  
    {  
        rowSelected = true;  
        int row = PCMember_Table.getSelectedRow();  
        String Table_click = (PCMember_Table.getModel().getValueAt(row,  
0).toString());  
        Jayesh_Assignment abc = new Jayesh_Assignment();  
        Connection conn1 = abc.getConn1();  
        String sql ="Select * from PCMember@FIT5148B where Member_ID =  
""+Table_click+"";  
        pst = conn1.prepareStatement(sql);  
        rs = pst.executeQuery();  
        if(rs.next())  
        {  
            String add2 = rs.getString("First_Name");  
            txt_member_first_name.setText(add2);  
            String add3 = rs.getString("Surname");  
            txt_member_surname.setText(add3);  
            String add4 = rs.getString("Member_Title");  
            combo_Member_Title.setSelectedItem(add4);  
            String add5 = rs.getString("Member_Position");  
            combo_Member_Position.setSelectedItem(add5);  
            String add6 = rs.getString("Member_Affiliation");  
            txt_member_affiliation.setText(add6);  
            String add7 = rs.getString("Member_Email");  
            txt_member_email.setText(add7);  
            String add8 = rs.getString("Track_ID");  
            combo_Track_ID.setSelectedItem(add8);  
        }  
    }  
    catch(Exception e)  
    {  
        JOptionPane.showMessageDialog(null,e);  
    }  
}
```

When you click on a row the values in the row are popped in textboxes and combo boxes:



55. Code for selecting rows by PCMember Affiliation:

```
private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn = abc.getConn();
        String sql ="Select * from PCMember@FIT5148B where Member_affiliation
= ?";
        pst = conn.prepareStatement(sql);
        pst.setString(1,txtSearch.getText());
        rs = pst.executeQuery();
        PCMember_Table.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
}
```

}

When you type Member Affiliation in text search field the rows with the same member affiliation name are seen in table:

The screenshot shows a Windows application window with a title bar and a menu bar. The main area contains a table with columns: MEMBER_ID, FIRST_NAME, SURNAME, MEMBER_TI..., MEMBER_P..., MEMBER_A..., MEMBER_E..., and TRACK_ID. Two rows are visible: one for 'jack' and one for 'Albus'. To the right of the table is a search bar labeled 'Search By' with the text 'Kolkata University' and a dropdown menu labeled 'Member Affiliation'. Below the table is a section titled 'Table Row Content' containing fields for First_Name, Member_Affiliation, Surname, Member_Email, Member_Title, Track_ID, and Member_Position. On the far right is a vertical panel titled 'Button' with six buttons: View, Insert, Update, Delete, Reset, and Back.

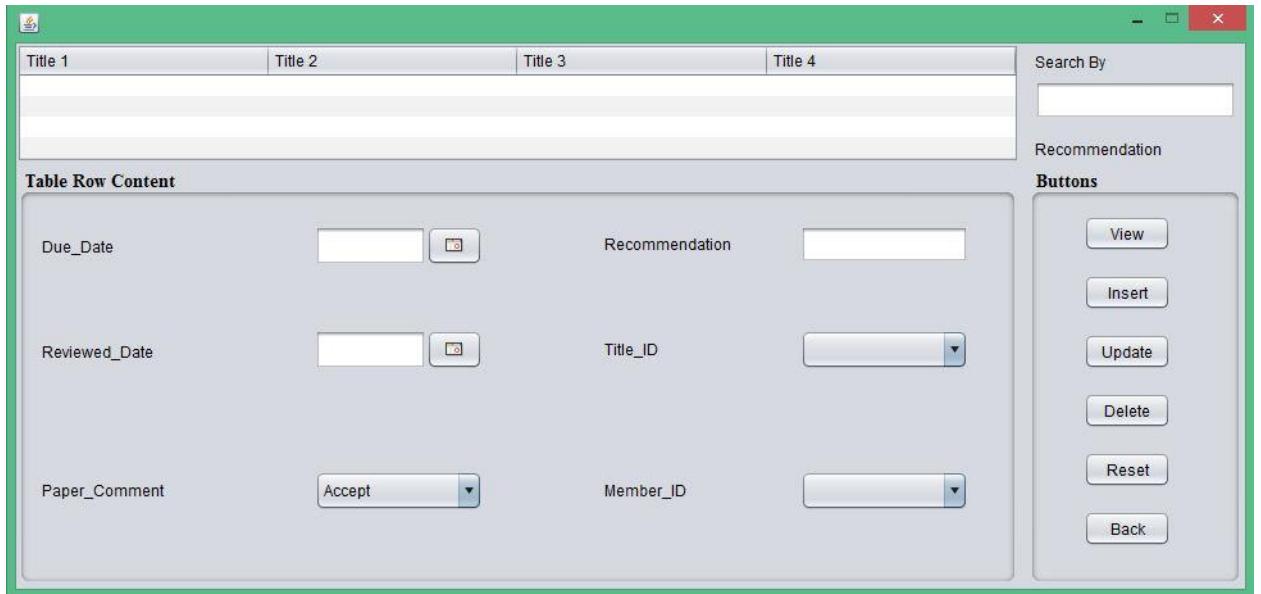
MEMBER_ID	FIRST_NAME	SURNAME	MEMBER_TI...	MEMBER_P...	MEMBER_A...	MEMBER_E...	TRACK_ID
7	jack	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4
4	Albus	Dumbledore	None	Student	Kolkatta Uni...	Dumbledore...	4

Table Row Content

First_Name	<input type="text"/>	Member_Affiliation	<input type="text"/>
Surname	<input type="text"/>	Member_Email	<input type="text"/>
Member_Title	<input type="button" value="DR"/>	Track_ID	<input type="button" value="2"/>
Member_Position	<input type="button" value="Lecturer"/>		

Button

18. Review Frame



56. Code to view the content of Review table:

```

private void View_Table_Review()
{
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from Review@FIT5148B";
        String sql1 ="Select * from PCMember@FIT5148B";
        String sql2 ="Select * from Paper@FIT5148B";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        Review_table.setModel(DbUtils.resultSetToTableModel(rs));
        pst1 = conn1.prepareStatement(sql1);
        rs1 = pst1.executeQuery(sql1);
        combo_Member_ID.removeAllItems();
        while (rs1.next())
        {
            String member_id = rs1.getString("Member_ID");
            combo_Member_ID.addItem(member_id);
        }
        pst2 = conn1.prepareStatement(sql2);
        rs2 = pst2.executeQuery(sql2);
        combo_Title_ID.removeAllItems();
        while (rs2.next())
    }
}

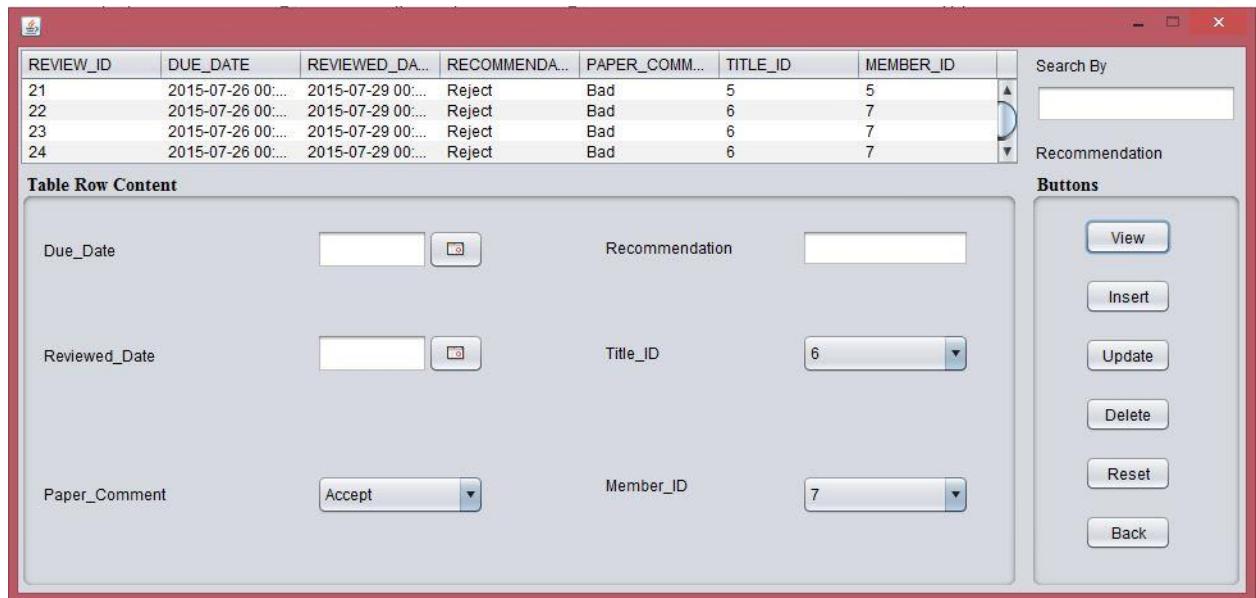
```

```

{
    String title_id = rs2.getString("Title_ID");
    combo_Title_ID.addItem(title_id);
}
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

Click on View Button to see the table content:



57. Code to reset the user interface value to null:

```

public void clearAll()
{
    txt_Due_Date.setDate(null);
    txt_Review_Date.setDate(null);
    combo_Recommendation.setSelectedIndex(0);
    txt_Paper_Comment.setText("");
    combo_Title_ID.setSelectedIndex(0);
    combo_Member_ID.setSelectedIndex(0);
}

```

58. Code to insert row in Review table:

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try  
    {  
        Jayesh_Assignment abc = new Jayesh_Assignment();  
        Connection conn1 = abc.getConn1();  
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");  
        String d = sdf.format(txt_Due_Date.getDate());  
        String d1 = sdf.format(txt_Review_Date.getDate());  
        String sql ="Insert into Review@FIT5148B (Review_ID,Due_Date,  
Reviewed_date, Recommendation, Paper_comment, Title_ID, Member_ID) values  
(Reviw_seq.nextval"+",""+ d +",""+ d1  
+", "+combo_Recommendation.getSelectedItem()+"",""+txt_Paper_Comment.getText()  
()+"",""+combo_Title_ID.getSelectedItem()+"",""+combo_Member_ID.getSelectedItem()  
+");  
        System.out.println(sql);  
        pst = conn1.prepareStatement(sql);  
        pst.execute();  
        JOptionPane.showMessageDialog(null,"Inserted");  
  
    }  
    catch (SQLException | HeadlessException e)  
    {  
        JOptionPane.showMessageDialog(null,e);  
    }  
    View_Table_Review();  
}
```

Add value in user interface and then press insert button and then check the table content. The values will be inserted:

Data in the table before the execution of Command:

REVIEW_ID	DUE_DATE	REVIEWED_DATE	RECOMMENDA...	PAPER_COMM...	TITLE_ID	MEMBER_ID
22	2015-07-26 00:0...	2015-07-29 00:0...	Reject	Bad	6	7
3	2008-07-18 00:0...	2008-07-20 00:0...	Borderline	Needs Correction	3	3
4	2012-01-11 00:0...	2012-01-21 00:0...	Weakly Reject	Needs Rework	4	4
5	2015-07-26 00:0...	2015-07-29 00:0...	Reject	Bad	5	5

Table Row Content

Due_Date	Jul 10, 2015	...	Recommendation	Good Presentation
Reviewed_Date	Jul 24, 2015	...	Title_ID	2
Paper_Comment	Accept	...	Member_ID	2

Buttons

-
-
-
-
-
-

Data in the table after the execution of Command:

REVIEW_ID	DUE_DATE	REVIEWED_DA...	RECOMMENDA...	PAPER_COMMENT	TITLE_ID	MEMBER_ID
22	2015-07-26 00:0...	2015-07-29 00:0...	Reject	Bad	6	7
41	2015-07-10 00:0...	2015-07-24 00:0...	Accept	Good Presentation	2	2
3	2008-07-18 00:0...	2008-07-20 00:0...	Borderline	Needs Correction	3	3
4	2012-01-11 00:0...	2012-01-21 00:0...	Weakly Reject	Needs Rework	4	4

Table Row Content

Due_Date	Jul 10, 2015	...	Recommendation	Good Presentation
Reviewed_Date	Jul 24, 2015	...	Title_ID	6
Paper_Comment	Accept	...	Member_ID	7

Buttons

-
-
-
-
-
-

59. Code to update row in Review table:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if (rowSelected == true)  
    {  
        try  
        {  
            int row = Review_table.getSelectedRow();  
            String Table_click = (Review_table.getModel().getValueAt(row,  
0).toString());  
            Jayesh_Assignment abc = new Jayesh_Assignment();  
            Connection conn1 = abc.getConn1();  
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");  
            Integer n =  
            Integer.parseInt((java.lang.String)combo_Title_ID.getSelectedItem());  
            Integer count = 0;  
            String sql1 = "select count(*) from review where title_id = '" + n + "'";  
            pst1 = conn1.prepareStatement(sql1);  
            rs1 = pst1.executeQuery();  
            if (rs1.next())  
            {  
                count = rs.getInt(1);  
            }  
            if (count<3)  
            {  
                String d1 = sdf.format(txt_Due_Date.getDate());  
                String d2 = sdf.format(txt_Review_Date.getDate());  
                String d3 = (String) combo_Recommendation.getSelectedItem();  
                String d4 = txt_Paper_Comment.getText();  
                String d5 = (String) combo_Title_ID.getSelectedItem();  
                String d6 = (String) combo_Member_ID.getSelectedItem();  
                String sql = "Update Review@FIT5148B set Due_Date ='" + d1 + "',  
Reviewed_date ='" + d2 + "', Recommendation ='" + d3 + "','" + Paper_comment ='"  
+ d4 + "','" + Title_ID ='" + d5 + "','" + Member_ID ='" + d6 + "' where Review_ID =  
'" + Table_click + "'";  
                System.out.println(sql);  
                pst = conn1.prepareStatement(sql);  
                pst.execute();  
                JOptionPane.showMessageDialog(null,"Updated");  
            }  
            else  
            {
```

```

        JOptionPane.showMessageDialog(null,"Already 3 Review. Cannot insert
more than 3 review for a paper");
    }
}
catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Review();
rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

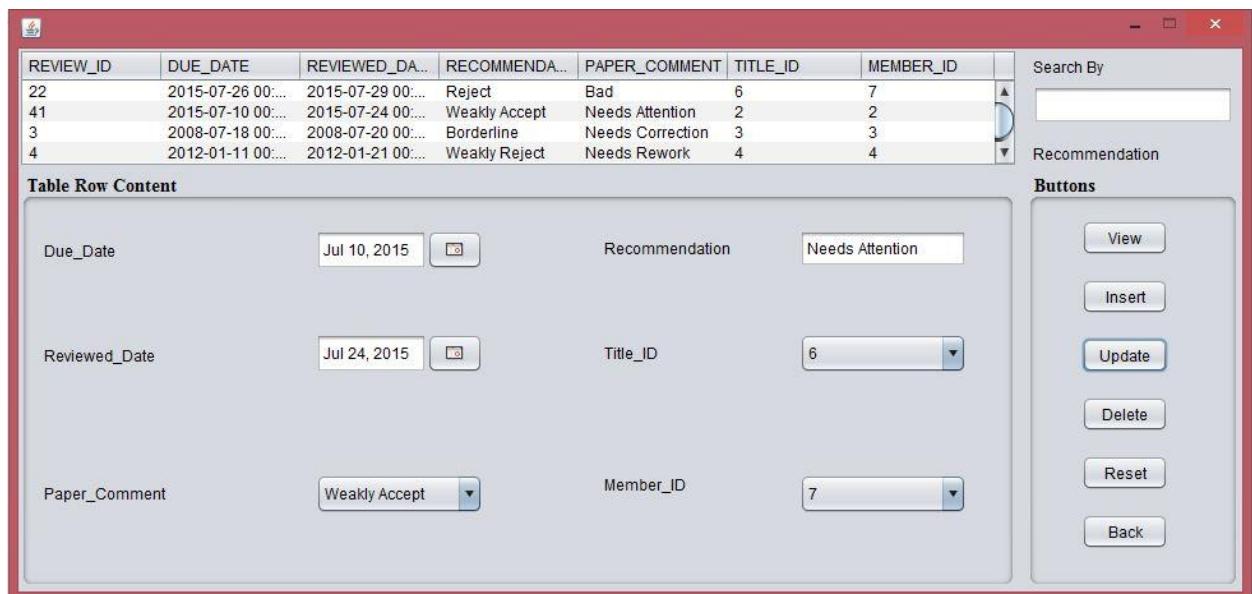
Select the row you want to update and then enter the new values that you want to update and then press update:

Data in the table before the execution of Command:

The screenshot shows a Java Swing application window. On the left is a JTable with columns: REVIEW_ID, DUE_DATE, REVIEWED_DA..., RECOMMENDA..., PAPER_COMMENT, TITLE_ID, and MEMBER_ID. A row with REVIEW_ID 41 is selected. To the right of the table is a panel titled 'Table Row Content' containing four input fields: 'Due_Date' (Jul 10, 2015), 'Reviewed_Date' (Jul 24, 2015), 'Paper_Comment' (Weakly Accept), and 'Recommendation' (Needs Attention). To the right of the content panel is a vertical stack of buttons labeled 'Buttons' with options: View, Insert, Update, Delete, Reset, and Back.

REVIEW_ID	DUE_DATE	REVIEWED_DA...	RECOMMENDA...	PAPER_COMMENT	TITLE_ID	MEMBER_ID
22	2015-07-26 00:...	2015-07-29 00:...	Reject	Bad	6	7
41	2015-07-10 00:...	2015-07-24 00:...	Accept	Good Presentation	2	2
3	2008-07-18 00:...	2008-07-20 00:...	Borderline	Needs Correction	3	3
4	2012-01-11 00:...	2012-01-21 00:...	Weakly Reject	Needs Rework	4	4

Data in the table after the execution of Command:



60. Code to delete row in Review table:

```

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        int p = JOptionPane.showConfirmDialog(null, "Do you really want to
delete","Delete",JOptionPane.YES_NO_OPTION );
        if( p == 0)
        {
            try
            {
                Jayesh_Assignment abc = new Jayesh_Assignment();
                Connection conn1 = abc.getConn1();
                int row = Review_table.getSelectedRow();
                String Table_click = (Review_table.getModel().getValueAt(row,
0).toString());
                String sql = "Delete from Review@FIT5148B where Review_ID
= '"+Table_click+"'";
                pst = conn1.prepareStatement(sql);
                rs = pst.executeQuery();
                JOptionPane.showMessageDialog(null, "Row is deleted");
            }
            catch (SQLException | HeadlessException e)
            {

```

```

        JOptionPane.showMessageDialog(null, e);
    }
    View_Table_Review();
    rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select a row and then press delete:

Data in the table before the execution of Command:

The screenshot shows a Java Swing application window titled "View Table". On the left is a table with columns: REVIEW_ID, DUE_DATE, REVIEWED_DA..., RECOMMENDA..., PAPER_COMMENT, TITLE_ID, and MEMBER_ID. A row with REVIEW_ID 41 is selected. On the right is a panel titled "Buttons" containing buttons for View, Insert, Update, Delete, Reset, and Back. Below the table is a panel titled "Table Row Content" containing fields for Due Date (Jul 10, 2015), Recommendation (Needs Attention), Reviewed Date (Jul 24, 2015), Title ID (2), Paper Comment (Weakly Accept), and Member ID (2).

REVIEW_ID	DUE_DATE	REVIEWED_DA...	RECOMMENDA...	PAPER_COMMENT	TITLE_ID	MEMBER_ID
22	2015-07-26 00:00:00	2015-07-29 00:00:00	Reject	Bad	6	7
41	2015-07-10 00:00:00	2015-07-24 00:00:00	Weakly Accept	Needs Attention	2	2
3	2008-07-18 00:00:00	2008-07-20 00:00:00	Borderline	Needs Correction	3	3
4	2012-01-11 00:00:00	2012-01-21 00:00:00	Weakly Reject	Needs Rework	4	4

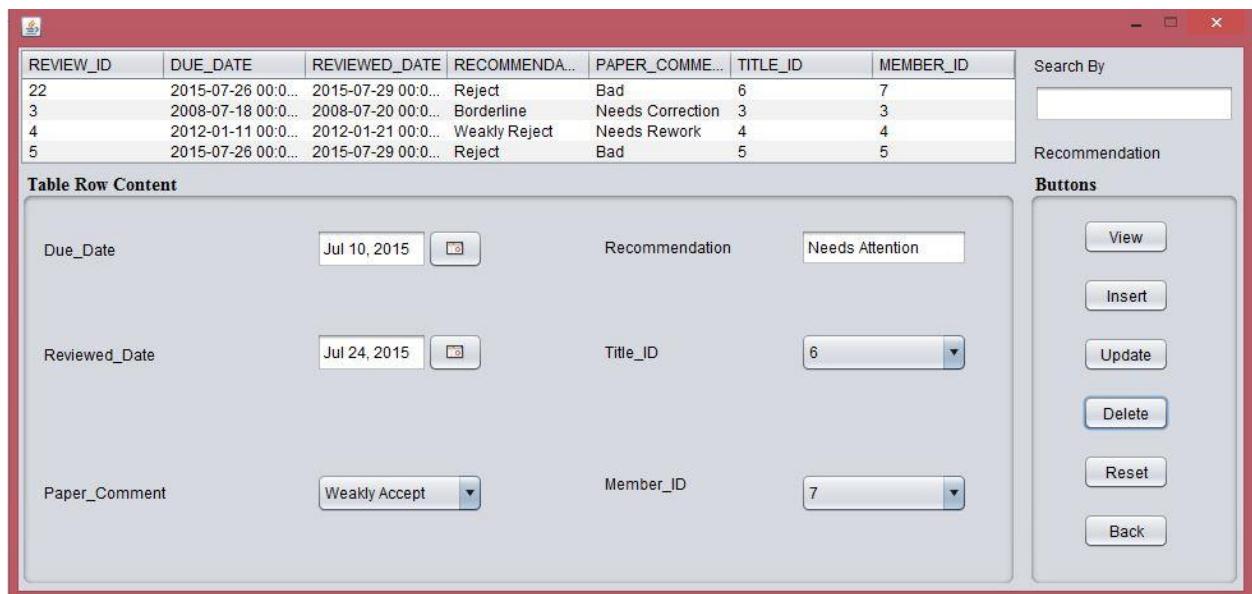
Table Row Content

Due_Date	Jul 10, 2015	...	Recommendation	Needs Attention
Reviewed_Date	Jul 24, 2015	...	Title_ID	2
Paper_Comment	Weakly Accept	...	Member_ID	2

Buttons

View
Insert
Update
Delete
Reset
Back

Data in the table after the execution of Command:



- Code to select a row in Review table by clicking on it:

```
private void Review_tableMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try {
        rowSelected = true;
        int row = Review_table.getSelectedRow();
        String Table_click = (Review_table.getModel().getValueAt(row, 0).toString());
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from Review@FIT5148B where Review_ID =
        "+Table_click+"";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        if(rs.next())
        {
            Date add2 = rs.getDate("Due_Date");
            txt_Due_Date.setDate(add2);
            Date add3 = rs.getDate("Reviewed_Date");
            txt_Review_Date.setDate(add3);
            String add4 = rs.getString("Recommendation");
            combo_Recommendation.setSelectedItem(add4);
        }
    }
}
```

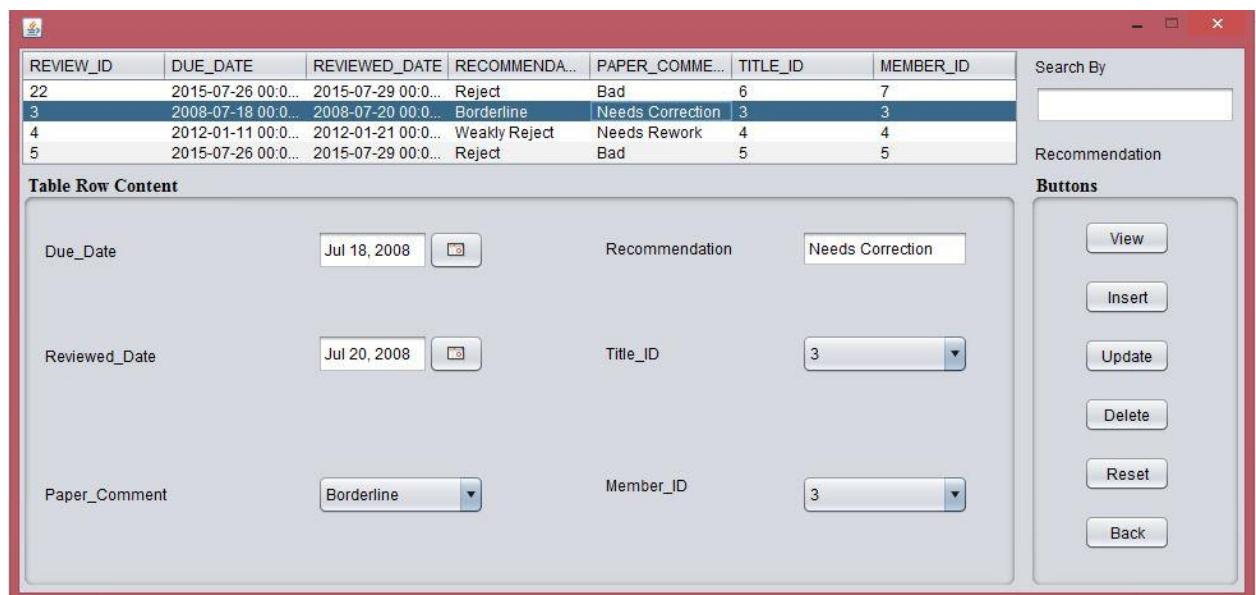
```

        String add5 = rs.getString("Paper_Comment");
        txt_Paper_Comment.setText(add5);
        String add6 = rs.getString("Title_ID");
        combo_Title_ID.setSelectedItem(add6);
        String add7 = rs.getString("Member_ID");
        combo_Member_ID.setSelectedItem(add7);
    }

}

catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```



61. Code for selecting rows by Review Recommendations:

```

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from Review@FIT5148B where Recommendation = ?";
        pst = conn1.prepareStatement(sql);
        pst.setString(1,txtSearch.getText());
    }
}

```

```

        rs = pst.executeQuery();
        Review_table.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
}

```

The screenshot shows a Java Swing application window with a title bar and a main content area. In the top right corner, there is a search bar labeled "Search By" containing the value "Borderline". Below the search bar, there is a section labeled "Recommendation". On the right side of the window, there is a vertical panel titled "Buttons" containing several buttons: "View", "Insert", "Update", "Delete", "Reset", and "Back". The main content area contains a table with columns: REVIEW_ID, DUE_DATE, REVIEWED_DATE, RECOMMENDATION, PAPER_COMMENT, TITLE_ID, and MEMBER_ID. A single row is selected, showing values: 3, 2008-07-18 00:00:00, 2008-07-20 00:00:00, Borderline, Needs Correction, 3, 3. Below the table, there is a section titled "Table Row Content" which displays the row's data in edit mode. It includes fields for Due Date (Jul 18, 2008), Recommendation (Needs Correction), Reviewed Date (Jul 20, 2008), Title ID (3), Paper Comment (Borderline), and Member ID (3).

REVIEW_ID	DUE_DATE	REVIEWED_DATE	RECOMMENDATION	PAPER_COMMENT	TITLE_ID	MEMBER_ID
3	2008-07-18 00:00:00	2008-07-20 00:00:00	Borderline	Needs Correction	3	3

Table Row Content

Due_Date	Jul 18, 2008	...	Recommendation	Needs Correction
Reviewed_Date	Jul 20, 2008	...	Title_ID	3
Paper_Comment	Borderline	...	Member_ID	3

19. Submission Frame:

62.

63. Code to view the content of Submission table:

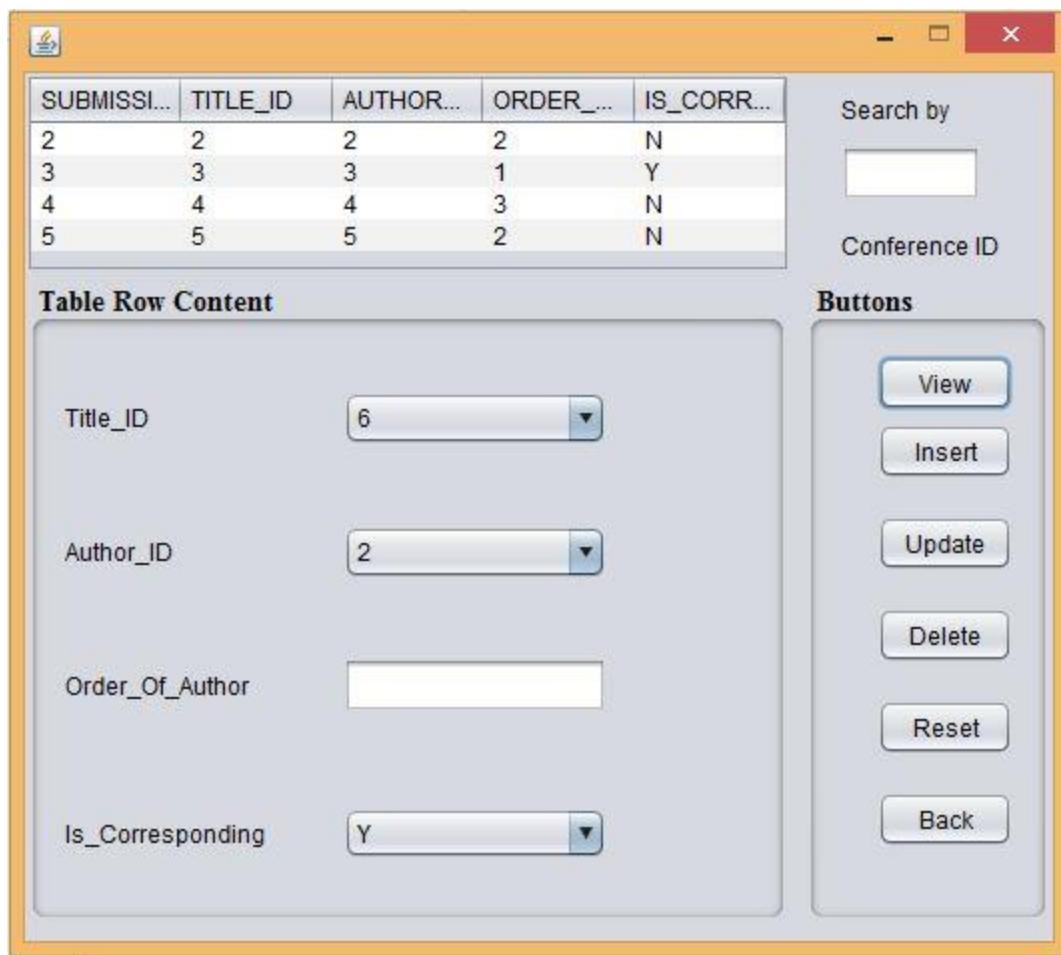
```
private void View_Table_Submission()
{
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from Submission@FIT5148B";
        String sql1 ="Select * from Author@FIT5148B";
        String sql2 ="Select * from Paper@FIT5148B";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        Submission_table.setModel(DbUtils.resultSetToTableModel(rs));
        pst1 = conn1.prepareStatement(sql1);
        rs1 = pst1.executeQuery(sql1);
        combo_Author_ID.removeAllItems();
```

```

while (rs1.next())
{
    String author_id = rs1.getString("Author_ID");
    combo_Author_ID.addItem(author_id);
}
pst2 = conn1.prepareStatement(sql2);
rs2 = pst2.executeQuery(sql2);
combo_Title_ID.removeAllItems();
while (rs2.next())
{
    String title_id = rs2.getString("Title_ID");
    combo_Title_ID.addItem(title_id);
}
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

Click on View Button to see the table content:



64. Code to reset the user interface value to null:

```
public void clearAll()
{
    combo_Title_ID.setSelectedIndex(0);
    combo_Author_ID.setSelectedIndex(0);
    txt_Order_Of_Author.setText("");
    combo_Is_Corresponding.setSelectedIndex(0);
}
```

65. Code to insert row in Submission table:

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
```

```

        String sql ="Insert into Submission@FIT5148B (Submission_ID,Title_ID,
Author_ID, Order_Of_Author, IS_Corresponding) values
(Submission_seq.nextval"+","+combo_Title_ID.getSelectedItem()+"."+combo_Auth
or_ID.getSelectedItem()+"."+txt_Order_Of_Author.getText()+"."+combo_Is_Corre
sponding.getSelectedItem()+"')";

        System.out.println(sql);
        pst = conn1.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null,"Inserted");

    }

    catch (SQLException | HeadlessException e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
    View_Table_Submission();
}

```

Add value in user interface and then press insert button and then check the table content. The values will be inserted:

Data in the table before the execution of Command:

SUBMISSI...	TITLE_ID	AUTHOR...	ORDER_...	IS_CORR...
2	2	2	2	N
3	3	3	1	Y
4	4	4	3	N
5	5	5	2	N

Table Row Content

Title_ID	<input type="text" value="3"/>
Author_ID	<input type="text" value="4"/>
Order_Of_Author	<input type="text" value="2"/>
Is_Corresponding	<input type="text" value="Y"/>

Buttons

<input type="button" value="View"/>
<input type="button" value="Insert"/>
<input type="button" value="Update"/>
<input type="button" value="Delete"/>
<input type="button" value="Reset"/>
<input type="button" value="Back"/>

Data in the table after the execution of Command:

SUBMISS...	TITLE_ID	AUTHOR...	ORDER_...	IS_COR...
21	3	4	2	Y
2	2	2	2	N
3	3	3	1	Y
4	4	4	3	N
5	5	5	2	N

Search by Conference ID

Table Row Content

Title_ID	<input type="text" value="6"/>
Author_ID	<input type="text" value="2"/>
Order_Of_Author	<input type="text" value="2"/>
Is_Corresponding	<input type="text" value="Y"/>

Buttons

66. Code to update row in Submission table:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        try
        {
            int row = Submission_table.getSelectedRow();
            String Table_click = (Submission_table.getModel().getValueAt(row,
0).toString());
            Jayesh_Assignment abc = new Jayesh_Assignment();
            Connection conn1 = abc.getConn1();
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");
            String d1 = (String) combo_Title_ID.getSelectedItem();
            String d2 = (String) combo_Author_ID.getSelectedItem();
        }
    }
}
```

```

String d3 = txt_Order_Of_Author.getText();
String d4 = (String) combo_Is_Corresponding.getSelectedItem();
String sql = "Update Submission@FIT5148B set Title_ID ='" + d1 + "' ,
Author_ID ='" + d2 + "', Order_Of_Author ='" + d3 + "','" + " IS_Corresponding ='" +
d4 + "' where Submission_ID ='" +Table_click+"'";
System.out.println(sql);
pst = conn1.prepareStatement(sql);
pst.execute();
JOptionPane.showMessageDialog(null,"Updated");
}
catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Submission();
rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select the row you want to update and then enter the new values that you want to update and then press update:

Data in the table before the execution of Command:

SUBMISS...	TITLE_ID	AUTHOR...	ORDER_...	IS_COR...
21	3	4	2	Y
2	2	2	2	N
3	3	3	1	Y
4	4	4	3	N
5	5	5	?	N

Table Row Content

Title_ID	22
Author_ID	14
Order_Of_Author	2
Is_Corresponding	Y

Search by

Conference ID

Buttons

Data in the table after the execution of Command:

SUBMISS...	TITLE_ID	AUTHOR...	ORDER_...	IS_COR...
21	22	14	2	Y
2	2	2	2	N
3	3	3	1	Y
4	4	4	3	N
5	5	5	2	N

Search by

Conference ID

Table Row Content

Title_ID	<input type="text" value="6"/>
Author_ID	<input type="text" value="2"/>
Order_Of_Author	<input type="text" value="2"/>
Is_Corresponding	<input type="text" value="Y"/>

Buttons

67. Code to delete row in Submission table:

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        int p = JOptionPane.showConfirmDialog(null, "Do you really want to
delete","Delete",JOptionPane.YES_NO_OPTION );
        if( p == 0)
        {
            try
            {
                Jayesh_Assignment abc = new Jayesh_Assignment();

```

```

        Connection conn1 = abc.getConn1();
        int row = Submission_table.getSelectedRow();
        String Table_click = (Submission_table.getModel().getValueAt(row,
0).toString());
        String sql = "Delete from Submission@FIT5148B where Submission_ID
= '"+Table_click+"'";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        JOptionPane.showMessageDialog(null, "Row is deleted");
    }
    catch (SQLException | HeadlessException e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
    View_Table_Submission();
    rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select a row and then press delete:

Data in the table before the execution of Command:

Table Row Content

SUBMISS...	TITLE_ID	AUTHOR...	ORDER_...	IS_COR...
21	22	14	2	Y
2	2	2	2	N
3	3	3	1	Y
4	4	4	3	N
5	5	5	?	N

Search by

Conference ID

Buttons

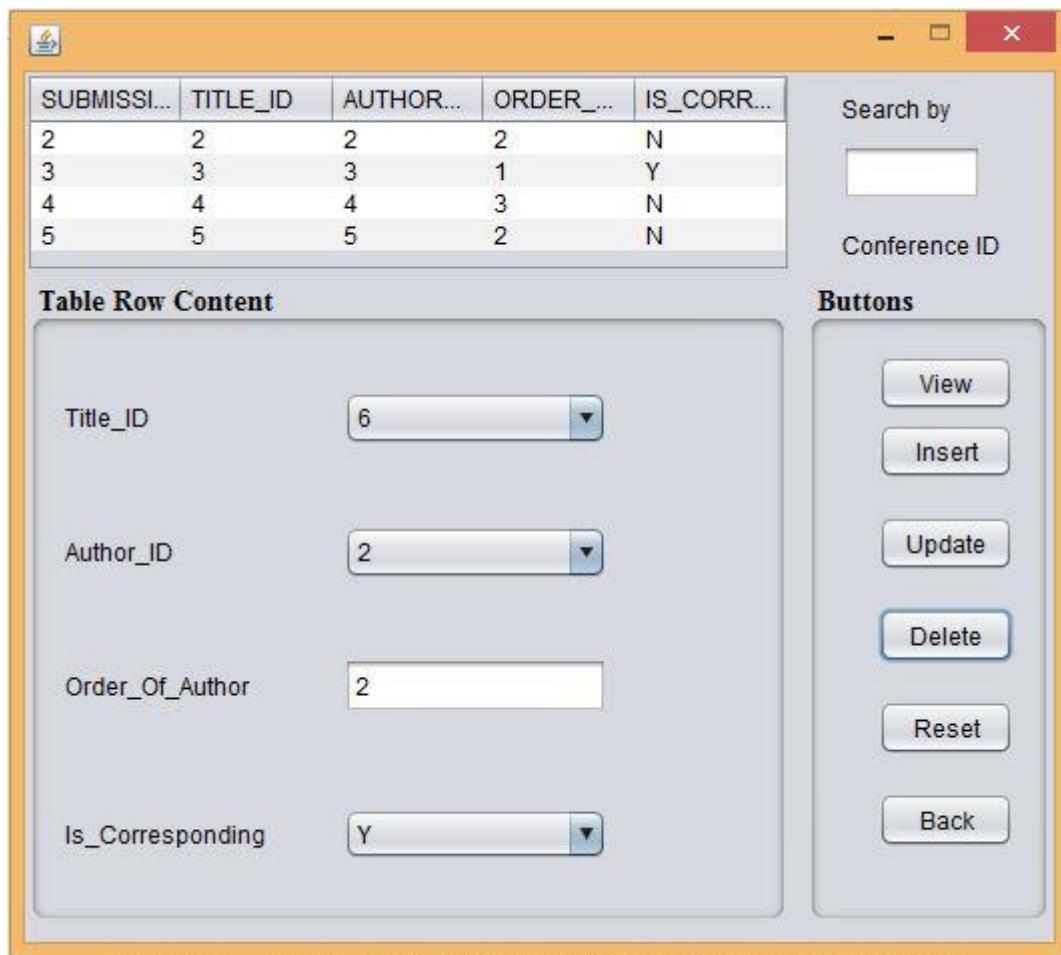
Title_ID

Author_ID

Order_Of_Author

Is_Corresponding

Data in the table after the execution of Command:



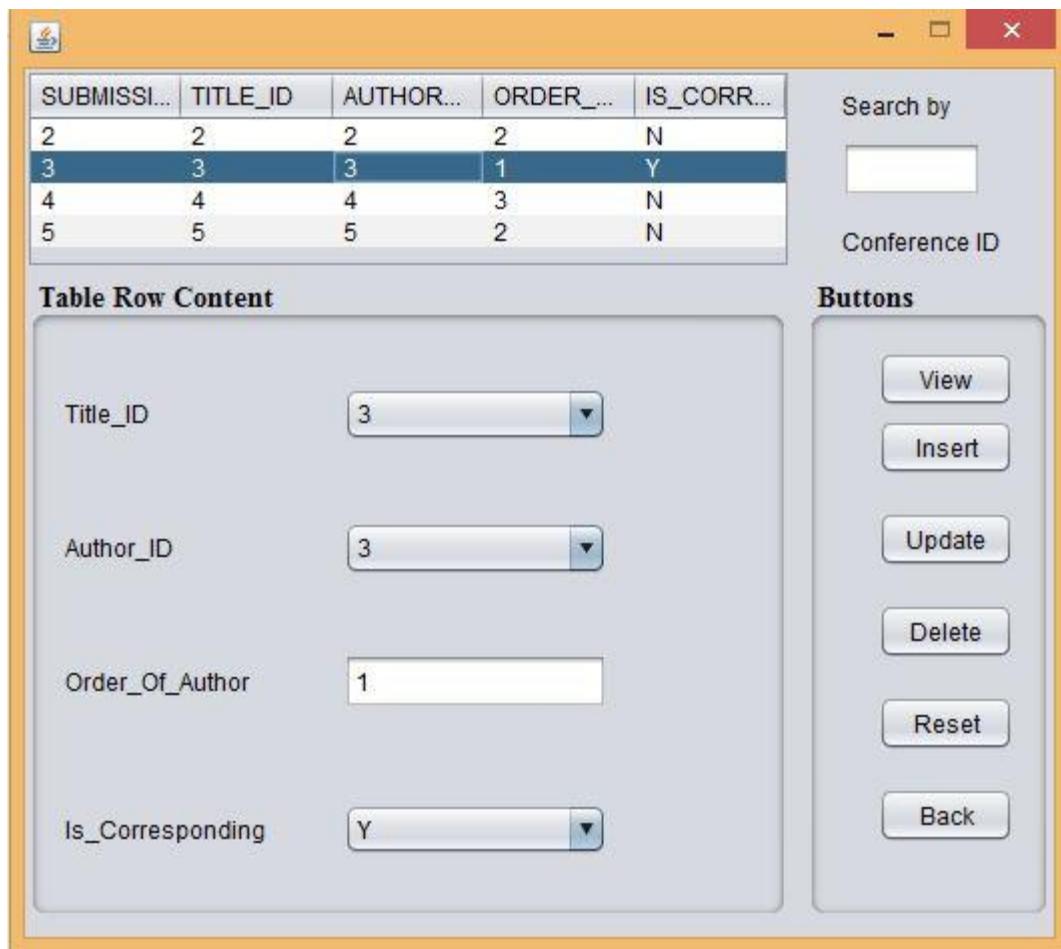
68. Code to select a row in Submission table by clicking on it:

```
private void Submission_tableMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try {
        rowSelected = true;
        int row = Submission_table.getSelectedRow();
        String Table_click = (Submission_table.getModel().getValueAt(row,
0).toString());
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from Submission@FIT5148B where Submission_ID =
"+Table_click+"";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        if(rs.next())
    {
```

```
String add1 = rs.getString("Title_ID");
combo_Title_ID.setSelectedItem(add1);
String add2 = rs.getString("Author_ID");
combo_Author_ID.setSelectedItem(add2);
String add3 = rs.getString("Order_of_Author");
txt_Order_Of_Author.setText(add3);
String add4 = rs.getString("Is_Corresponding");
combo_Is_Corresponding.setSelectedItem(add4);
}

}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}
```

When you click on a row the values in the row are popped in textboxes and combo boxes:



69. Code for selecting rows by Conference ID:

```

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    try {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn = abc.getConn();
        Connection conn1 = abc.getConn1();
        String sql1 = "Select * from Submission@Fit5148b where Title_ID IN (Select
Title_ID from Paper@Fit5148B where Track_id IN (Select Track_id from
Track@fit5148b where Conf_ID= (Select Conf_ID from Conference@Fit5148A
where Conf_ID=?))";
        pst1=conn1.prepareStatement(sql1);
        pst1.setString(1, txtSearch.getText());
        rs1=pst1.executeQuery();
        Submission_table.setModel(DbUtils.resultSetToTableModel(rs1));
    }
}
```

```

}
catch (Exception e7)
{
    JOptionPane.showMessageDialog(null, e7);
}
}
}

```

When you type conference ID in text search field the rows with the same conference ID are seen in table:

SUBMISSION_ID	TITLE_ID	AUTHOR_ID	ORDER_OF_AUTHOR	IS_CORRESPONDING
2	2	2	2	N

Table Row Content

Search by
2
Conference ID

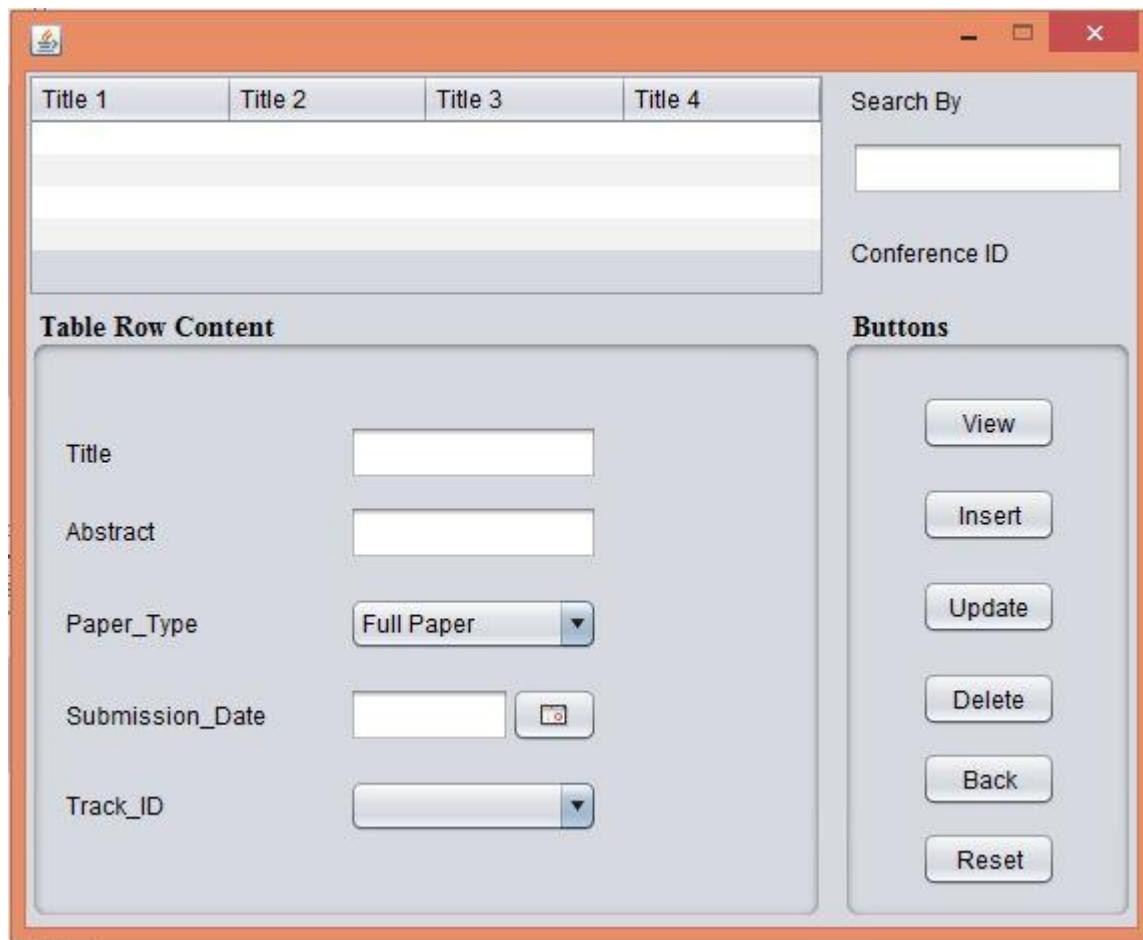
Buttons

Title_ID: 3
Author_ID: 3
Order_Of_Author: 1
Is_Corresponding: Y

Buttons

- View
- Insert
- Update
- Delete
- Reset
- Back

20. Paper Frame:



70. Code to view the content of Paper table:

```
private void View_Table_Paper()
{
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from Paper@FIT5148B";
        String sql1 ="Select * from Track@FIT5148B";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery(sql);
        Paper_table.setModel(DbUtils.resultSetToTableModel(rs));
        pst1 = conn1.prepareStatement(sql1);
        rs1 = pst.executeQuery(sql1);
```

```

combo_Track_ID.removeAllItems();
while (rs1.next())
{
    String track_id = rs1.getString("Track_ID");
    combo_Track_ID.addItem(track_id);
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

Click on View Button to see the table content:

The screenshot shows a Java Swing application window titled "Conference Management System".

Table Row Content:

TITLE_ID	TITLE	ABSTR...	PAPER...	SUBMIS...	TRACK...
6	SHOT	Strategi...	Poster	2008-0...	2
2	Finance...	Strategi...	Poster	2008-0...	2
3	Marketi...	Strategi...	Resear...	2008-0...	3
4	Sales	Strategi...	Full Pa...	2012-0...	4
5	Operati...	Strategi...	Poster	2015-0...	5

Buttons:

- View
- Insert
- Update
- Delete
- Back
- Reset

71. Code to reset the user interface value to null:

```
public void clearAll()
{
    txt_title.setText("");
    txt_abstract.setText("");
    combo_Paper_Type.setSelectedIndex(0);
    txt_Submission_date.setDate(null);
    combo_Track_ID.setSelectedIndex(0);
}
```

72. Code to insert row in Paper table:

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");
        String d = sdf.format(txt_Submission_date.getDate());
        String sql = "Insert into Paper@FIT5148B (Title_ID, Title, Abstract,
Paper_type, Submission_date, Track_ID) values
(paper_seq.nextval"+",""+txt_title.getText()+"','"+"+txt_abstract.getText()+"','"+"+combo_
Paper_Type.getSelectedItem()+"','"+" +d+"','"+"+combo_Track_ID.getSelectedItem()+"')";
        System.out.println(sql);
        pst = conn1.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null,"Inserted");

    }
    catch (SQLException | HeadlessException e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
    View_Table_Paper();
}
```

Add value in user interface and then press insert button and then check the table content. The values will be inserted:

Data in the table before the execution of Command:

The screenshot shows a Windows application window with a title bar and a main content area. In the top-left corner of the content area, there is a small icon. To its right is a search bar labeled "Search By" with a text input field. Below the search bar is another label "Conference ID".

The main content area contains a table with the following data:

TITLE_ID	TITLE	ABSTRA...	PAPER_...	SUBMIS...	TRACK_...
2	Finance ...	Strategie...	Poster	2008-06...	2
3	Marketin...	Strategie...	Researc...	2008-07...	3
4	Sales	Strategies	Full Paper	2012-01...	4
5	Operation	Strategie...	Poster	2015-07...	5
22	Operatio...	Strategie...	Poster	2015-07...	2

Below the table, there is a section titled "Table Row Content" which displays the details of the selected row (Title_ID 2). The fields are:

- Title: Mainframes
- Abstract: Coding
- Paper_Type: Poster
- Submission_Date: Jun 1, 2008
- Track_ID: 2

To the right of the content area is a vertical column of buttons labeled "Buttons" with the following options:

- View
- Insert
- Update
- Delete
- Back
- Reset

Data in the table after the execution of Command:

TITLE_ID	TITLE	ABSTR...	PAPER...	SUBMIS...	TRACK...
41	Mainfra...	Coding	Poster	2008-0...	2
2	Finance...	Strategi...	Poster	2008-0...	2
3	Marketi...	Strategi...	Resear...	2008-0...	3
4	Sales	Strategi...	Full Pa...	2012-0...	4
5	Operati...	Strategi...	Poster	2015-0...	5

Table Row Content

Title	Mainframes
Abstract	Coding
Paper_Type	Poster
Submission_Date	Jun 1, 2008
Track_ID	2

Buttons

-
-
-
-
-
-

73. Code to update row in Paper table:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        try
        {
            int row = Paper_table.getSelectedRow();
            String Table_click = (Paper_table.getModel().getValueAt(row, 0).toString());
            Jayesh_Assignment abc = new Jayesh_Assignment();
            Connection conn1 = abc.getConn1();
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");
            String n = txt_title.getText();
            Integer count = 0;
            String sql1 = "select count(*) from paper where title = '" + n + "'";
            pst1 = conn1.prepareStatement(sql1);
```

```

rs1 = pst1.executeQuery();
while(rs1.next())
{
    count = rs1.getInt(1);
}
if (count== 0)
{
    String d1 = txt_title.getText();
    String d2 = txt_abstract.getText();
    String d3 = (String) combo_Paper_Type.getSelectedItem();
    String d4 = sdf.format(txt_Submission_date.getDate());
    String d5 = (String) combo_Track_ID.getSelectedItem();
    String sql = "Update Paper@FIT5148B set Title ='" + d1 + "', Abstract ="
+ d2 + "', Paper_type ='" + d3 + "','" + " Submission_date ='" + d4 + "','" + "Track_ID"
= "'" + d5 + "' where Title_ID ='" + Table_click + "'";
    System.out.println(sql);
    pst = conn1.prepareStatement(sql);
    pst.execute();
    JOptionPane.showMessageDialog(null,"Updated");
}
else
{
    JOptionPane.showMessageDialog(null,"Title already there. Please change
the name of paper.");
}
}
catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Paper();
rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select the row you want to update and then enter the new values that you want to update and then press update:

Data in the table before the execution of Command:

The screenshot shows a Windows application window with a title bar and a main content area. In the top-left corner, there is a small icon. On the right side, there are standard window controls (minimize, maximize, close) and a search bar labeled "Search By" with a dropdown menu and a text input field. Below the search bar, there is a label "Conference ID".

The main content area contains a table with the following columns: TITLE_ID, TITLE, ABSTR..., PAPER..., SUBMIS..., and TRACK... . The table has five rows of data:

TITLE_ID	TITLE	ABSTR...	PAPER...	SUBMIS...	TRACK...
41	Mainfra...	Coding	Poster	2008-0...	2
2	Finance...	Strategi...	Poster	2008-0...	2
3	Marketi...	Strategi...	Resear...	2008-0...	3
4	Sales	Strategi...	Full Pa...	2012-0...	4
5	Operati...	Strategi...	Poster	2015-0...	5

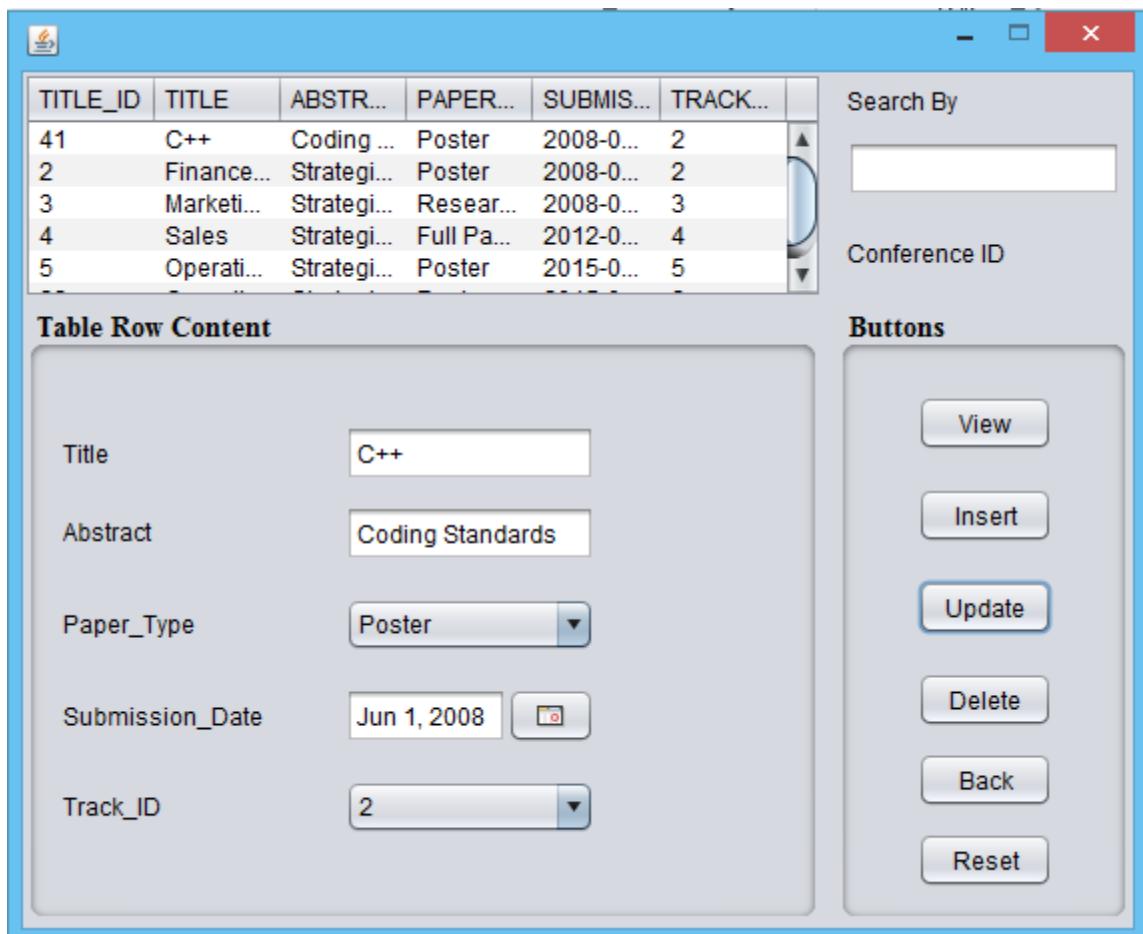
Below the table, there is a section titled "Table Row Content" containing five input fields corresponding to the table columns:

- Title: C++
- Abstract: Coding Standards
- Paper_Type: Poster
- Submission_Date: Jun 1, 2008 (with a calendar icon)
- Track_ID: 2

To the right of the input fields is a vertical column of buttons labeled "Buttons":

- View
- Insert
- Update
- Delete
- Back
- Reset

Data in the table after the execution of Command:



74. Code to delete row in Paper table:

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        int p = JOptionPane.showConfirmDialog(null, "Do you really want to
delete","Delete",JOptionPane.YES_NO_OPTION );
        if( p == 0)
        {
            try
            {
                Jayesh_Assignment abc = new Jayesh_Assignment();
                Connection conn1 = abc.getConn1();
                int row = Paper_table.getSelectedRow();
```

```

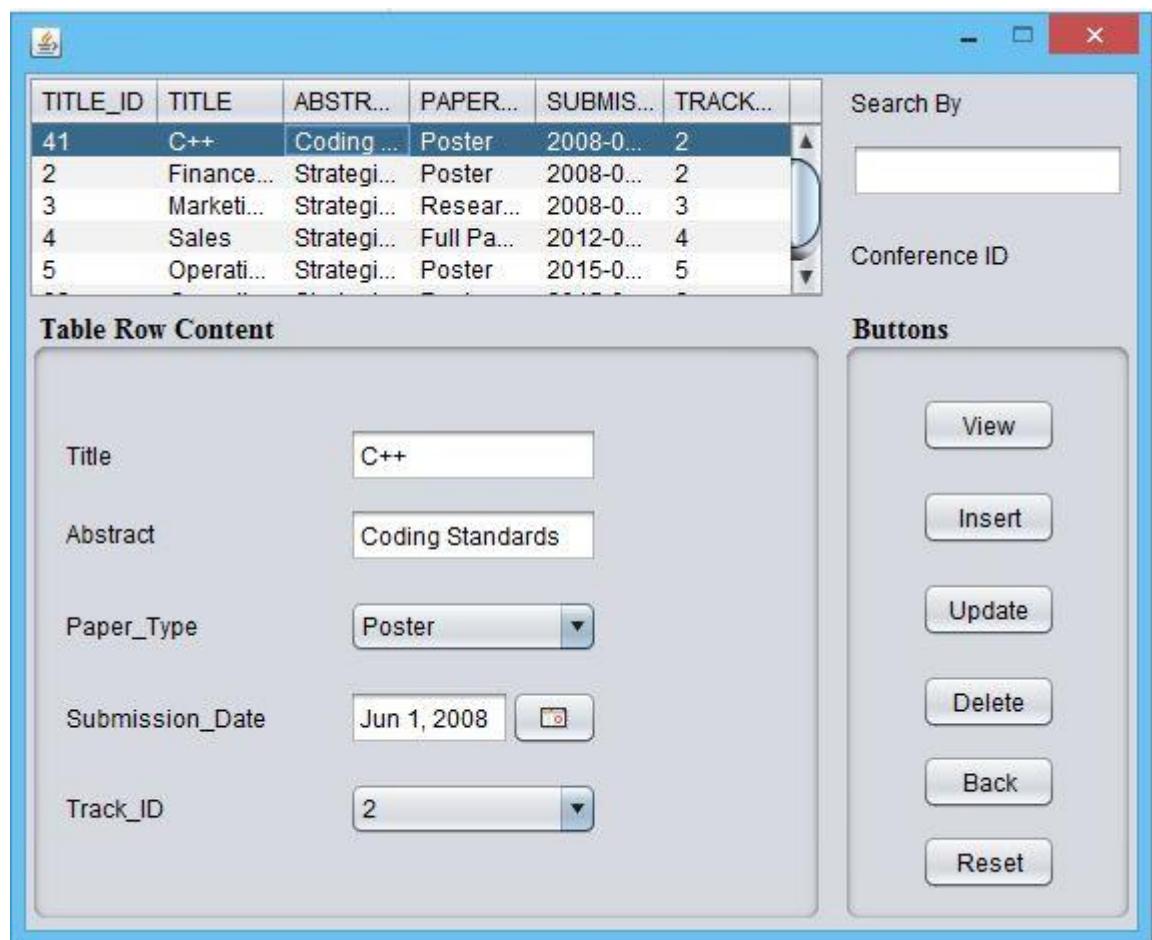
        String Table_click = (Paper_table.getModel().getValueAt(row,
0).toString());
        String sql = "Delete from Paper@FIT5148B where Title_ID
= '"+Table_click+"'";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        JOptionPane.showMessageDialog(null, "Row is deleted");
    }
    catch (SQLException | HeadlessException e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
    View_Table_Paper();
    rowSelected = false;
}

}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select a row and then press delete:

Data in the table before the execution of Command:

A screenshot of a Windows application window titled "Conference Management System". The window contains a table of conference titles, a search bar, and a form for editing a selected row.

The table has columns: TITLE_ID, TITLE, ABSTRACT, PAPER_TYPE, SUBMISSION_DATE, and TRACK_ID. The data is as follows:

TITLE_ID	TITLE	ABSTRACT	PAPER_TYPE	SUBMISSION_DATE	TRACK_ID
41	C++	Coding Standards	Poster	2008-06-01	2
2	Finance	Strategic Planning	Poster	2008-06-01	2
3	Marketing	Strategic Marketing	Research Paper	2008-06-01	3
4	Sales	Strategic Sales	Full Paper	2012-06-01	4
5	Operations	Strategic Operations	Poster	2015-06-01	5

Search By:

Conference ID:

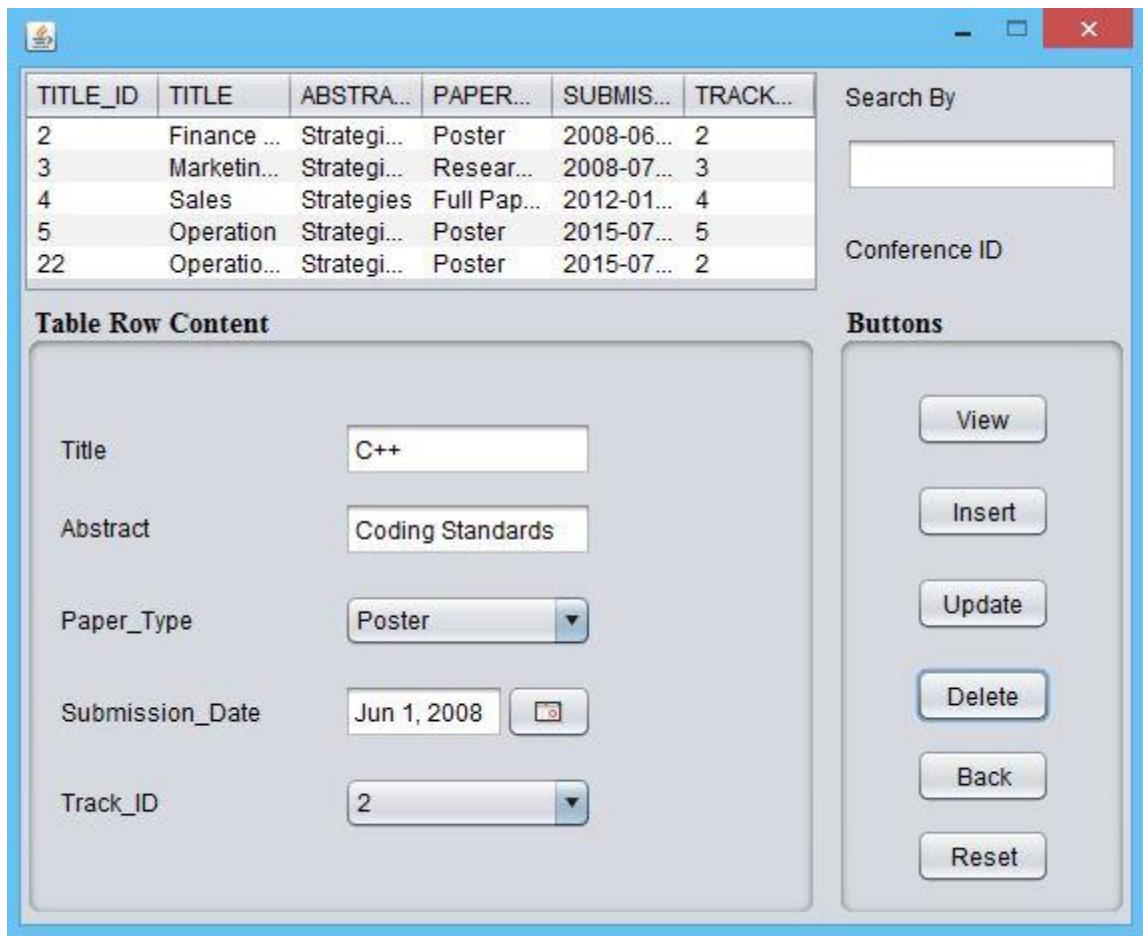
Table Row Content

Title	<input type="text" value="C++"/>
Abstract	<input type="text" value="Coding Standards"/>
Paper_Type	<input style="width: 100px; height: 25px; border: none; background-color: #f0f0f0; padding: 2px 5px; font-size: 10px; font-weight: bold; border-radius: 5px; margin-bottom: 5px;" type="button" value="Poster"/>
Submission_Date	<input type="text" value="Jun 1, 2008"/> <input type="button" value=""/>
Track_ID	<input style="width: 100px; height: 25px; border: none; background-color: #f0f0f0; padding: 2px 5px; font-size: 10px; font-weight: bold; border-radius: 5px; margin-bottom: 5px;" type="button" value="2"/>

Buttons

<input type="button" value="View"/>
<input type="button" value="Insert"/>
<input type="button" value="Update"/>
<input type="button" value="Delete"/>
<input type="button" value="Back"/>
<input type="button" value="Reset"/>

Data in the table after the execution of Command:



75. Code to select a row in Paper table by clicking on it:

```
private void Paper_tableMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try {
        rowSelected = true;
        int row = Paper_table.getSelectedRow();
        String Table_click = (Paper_table.getModel().getValueAt(row, 0).toString());
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from Paper@FIT5148B where Title_ID =
        "+Table_click+"";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        if(rs.next())
        {
            String add1 = rs.getString("Title");
```

```

txt_title.setText(add1);
String add2 = rs.getString("Abstract");
txt_abstract.setText(add2);
String add3 = rs.getString("Paper_Type");
combo_Paper_Type.setSelectedItem(add3);
Date add4 = rs.getDate("Submission_Date");
txt_Submission_date.setDate(add4);
String add5 = rs.getString("Track_ID");
combo_Track_ID.setSelectedItem(add5);
}

}

catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

When you click on a row the values in the row are popped in textboxes and combo boxes:

TITLE_ID	TITLE	ABSTRA...	PAPER...	SUBMIS...	TRACK...
2	Finance ...	Strategi...	Poster	2008-06...	2
3	Marketin...	Strategi...	Resear...	2008-07...	3
4	Sales	Strategies	Full Pap...	2012-01...	4
5	Operation	Strategi...	Poster	2015-07...	5
22	Operatio...	Strategi...	Poster	2015-07...	2

Table Row Content

Title	Sales
Abstract	Strategies
Paper_Type	Full Paper
Submission_Date	Jan 5, 2012
Track_ID	4

Buttons

-
-
-
-
-
-

76. Code for selecting row by Conference ID:

```

private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn = abc.getConn();
        Connection conn1 = abc.getConn1();
        String sql1 = "Select * from Paper@Fit5148B where Track_id IN (Select
Track_id from Track@fit5148b where Conf_ID= (Select Conf_ID from
Conference@Fit5148A where Conf_ID=?))";
        pst1=conn1.prepareStatement(sql1);
        pst1.setString(1, txtSearch.getText());
        rs1=pst1.executeQuery();
        Paper_table.setModel(DbUtils.resultSetToTableModel(rs1));
    }
}

```

```

        }
        catch (Exception e7)
        {
            JOptionPane.showMessageDialog(null, e7);
        }
    }
}

```

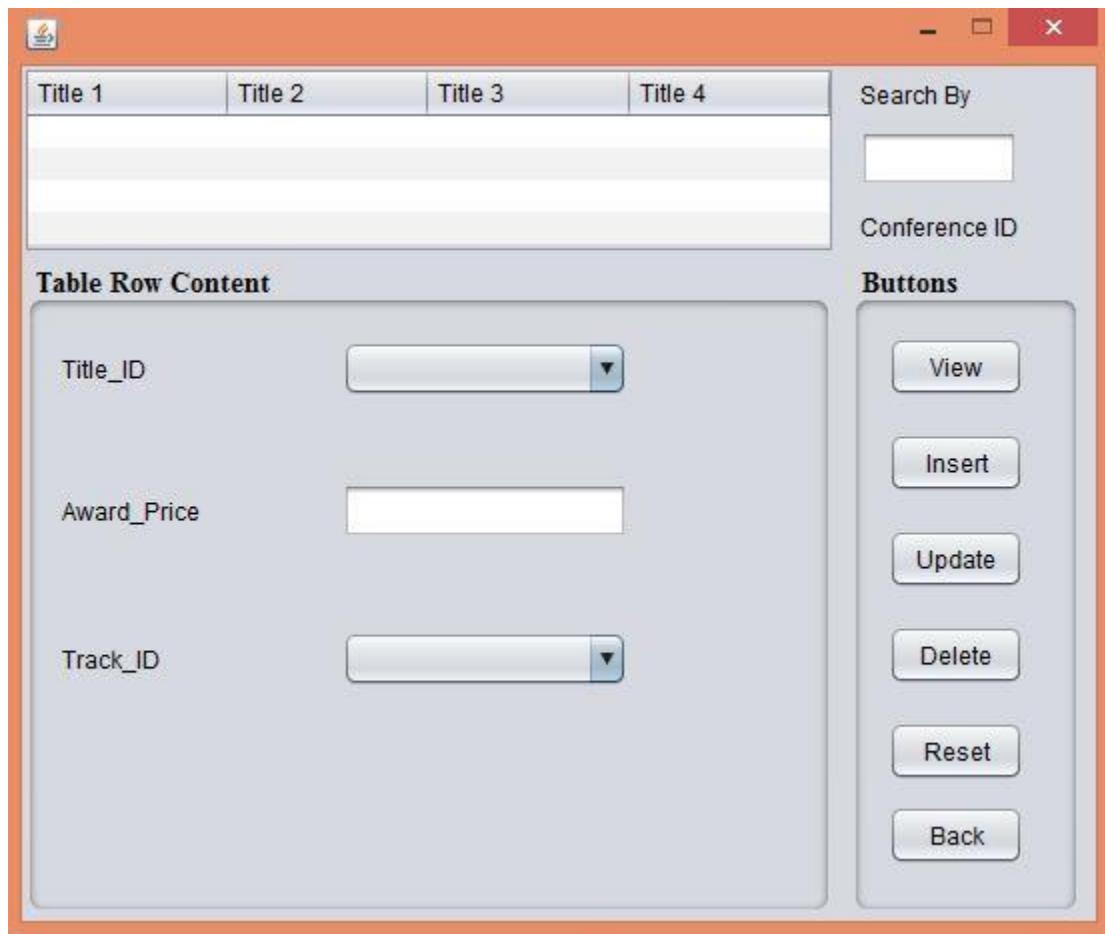
When you type conference city name in text search field the rows with the same city name are seen in table:

The screenshot shows a Java Swing application window with the following components:

- Table:** A JTable displaying conference data with columns: TITLE_ID, TITLE, ABSTRA..., PAPER..., SUBMIS..., and TRACK... . The visible data is:

TITLE_ID	TITLE	ABSTRA...	PAPER...	SUBMIS...	TRACK...
2	Finance ...	Strategi...	Poster	2008-06...	2
22	Operatio...	Strategi...	Poster	2015-07...	2
- Search By:** A text input field containing the value "2".
- Conference ID:** A label next to the search field.
- Table Row Content:** A panel containing five input fields corresponding to the table columns:
 - Title: Sales
 - Abstract: Strategies
 - Paper_Type: Full Paper
 - Submission_Date: Jan 5, 2012 (with a calendar icon)
 - Track_ID: 2
- Buttons:** A vertical stack of buttons:
 - View
 - Insert
 - Update
 - Delete
 - Back
 - Reset

21. BestPaper Frame:



77. Code to view the content of BestPaper table:

```
private void View_Table_Best_Paper()
{
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from BestPaper@FIT5148B";
        String sql1 ="Select * from Track@FIT5148B";
        String sql2 ="Select * from Paper@FIT5148B";
        pst = conn1.prepareStatement(sql);
```

```

rs = pst.executeQuery();
Best_Paper_table.setModel(DbUtils.resultSetToTableModel(rs));
pst1 = conn1.prepareStatement(sql1);
rs1 = pst.executeQuery(sql1);
combo_Track_ID.removeAllItems();
while (rs1.next())
{
    String track_id = rs1.getString("Track_ID");
    combo_Track_ID.addItem(track_id);
}
pst2 = conn1.prepareStatement(sql2);
rs2 = pst.executeQuery(sql2);
combo_Title_ID.removeAllItems();
while (rs2.next())
{
    String title_id = rs2.getString("Title_ID");
    combo_Title_ID.addItem(title_id);
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

```

Click on View Button to see the table content:

BEST_PAPER...	TITLE_ID	AWARD_PRICE	TRACK_ID
6	5	46646	6
2	2	11111	2
3	3	3000	3
5	5	5000	5

Table Row Content

Title_ID: 2

Award_Price:

Track_ID: 2

Buttons

View

Insert

Update

Delete

Reset

Back

78. Code to reset the user interface value to null:

```
public void clearAll()
{
    combo_Title_ID.setSelectedIndex(0);
    txt_Award_Price.setText("");
    combo_Track_ID.setSelectedIndex(0);
}
```

79. Code to insert row in BestPaper table:

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try
    {
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
```

```

String sql ="Insert into BestPaper@FIT5148B (Best_Paper_ID,Title_ID,
Award_Price, Track_ID) values
(Best_Paper_seq.nextval"+","+combo_Title_ID.getSelectedItem()+"','"+txt_Award_P
rice.getText()"+","+combo_Track_ID.getSelectedItem()+"");
System.out.println(sql);
pst = conn1.prepareStatement(sql);
pst.execute();
JOptionPane.showMessageDialog(null,"Inserted");

}

catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Best_Paper();
}

```

Add value in user interface and then press insert button and then check the table content. The values will be inserted:

Data in the table before the execution of Command:

Table Row Content

BEST_PAPER...	TITLE_ID	AWARD_PRICE	TRACK_ID
6	5	46646	6
2	2	11111	2
3	3	3000	3
5	5	5000	5

Search By

Conference ID

Buttons

Title_ID ▼

Award_Price

Track_ID ▼

View

Insert

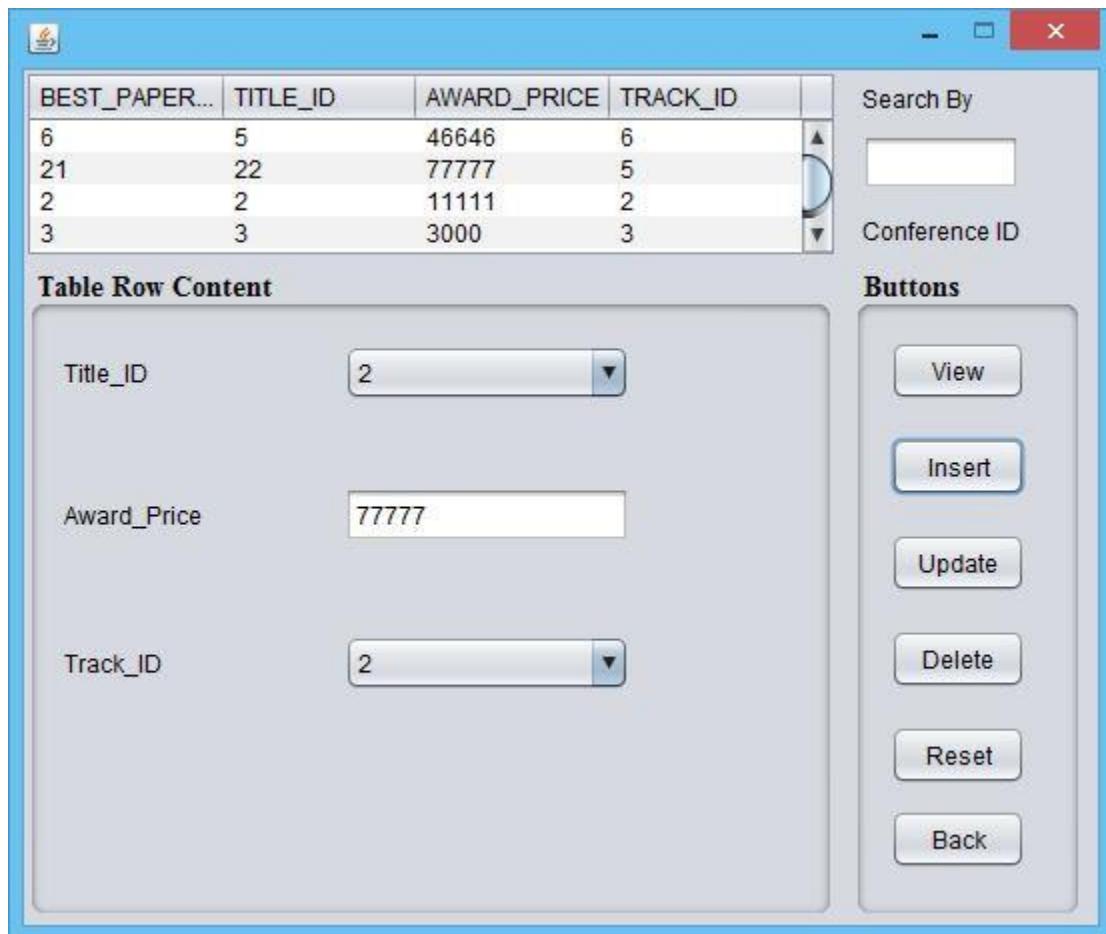
Update

Delete

Reset

Back

Data in the table after the execution of Command:



80. Code to update row in BestPaper table:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        try
        {
            int row = Best_Paper_table.getSelectedRow();
            String Table_click = (Best_Paper_table.getModel().getValueAt(row,
0).toString());
            Jayesh_Assignment abc = new Jayesh_Assignment();
            Connection conn1 = abc.getConn1();
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/YYYY");
            String d1 = (String) combo_Title_ID.getSelectedItem();
        }
    }
}
```

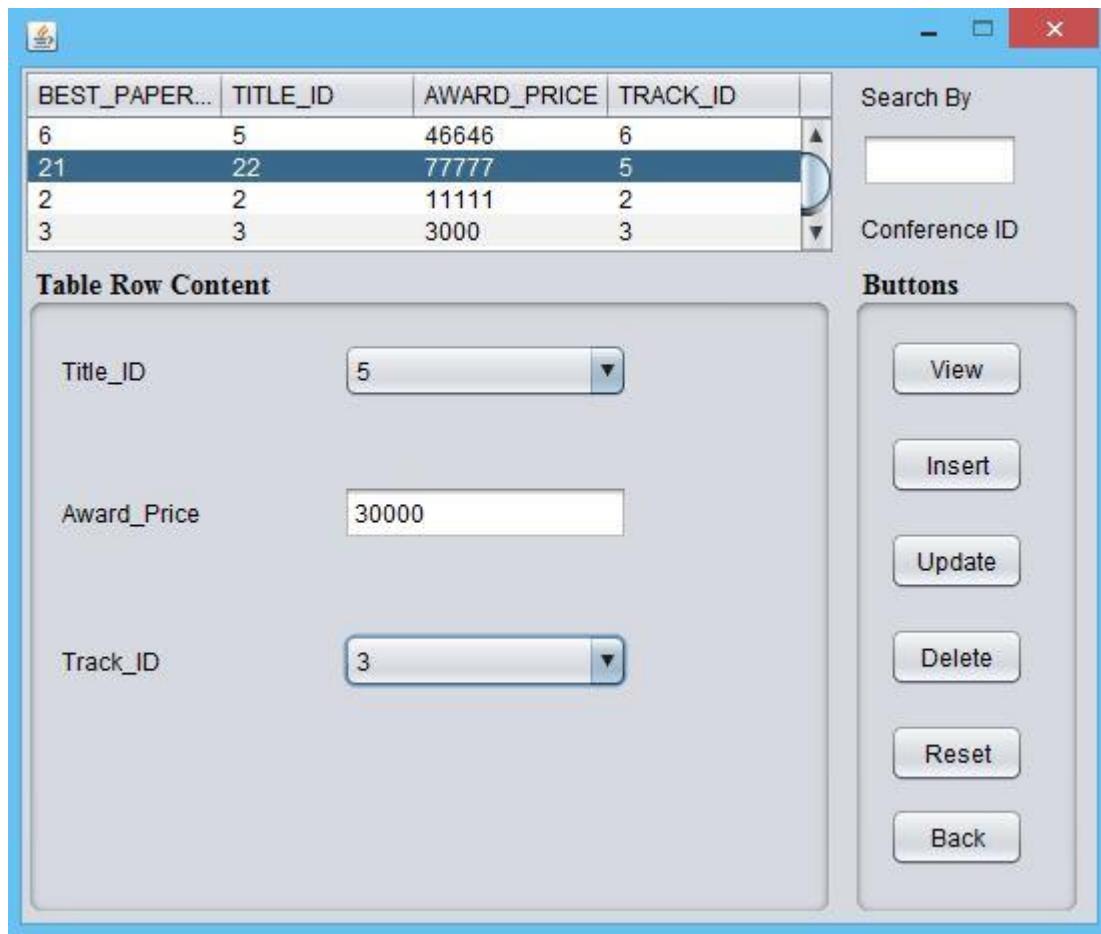
```

String d2 = txt_Award_Price.getText();
String d3 = (String) combo_Track_ID.getSelectedItem();
String sql = "Update BestPaper@FIT5148B set Title_ID ='" + d1 + "'",
Award_Price ='" + d2 + "'", Track_ID ='" + d3 + "'where Best_Paper_ID =
"+Table_click+""";
System.out.println(sql);
pst = conn1.prepareStatement(sql);
pst.execute();
JOptionPane.showMessageDialog(null,"Updated");
}
catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null,e);
}
View_Table_Best_Paper();
rowSelected = false;
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select the row you want to update and then enter the new values that you want to update and then press update:

Data in the table before the execution of Command:



BEST_PAPER...	TITLE_ID	AWARD_PRICE	TRACK_ID
6	5	46646	6
21	22	77777	5
2	2	11111	2
3	3	3000	3

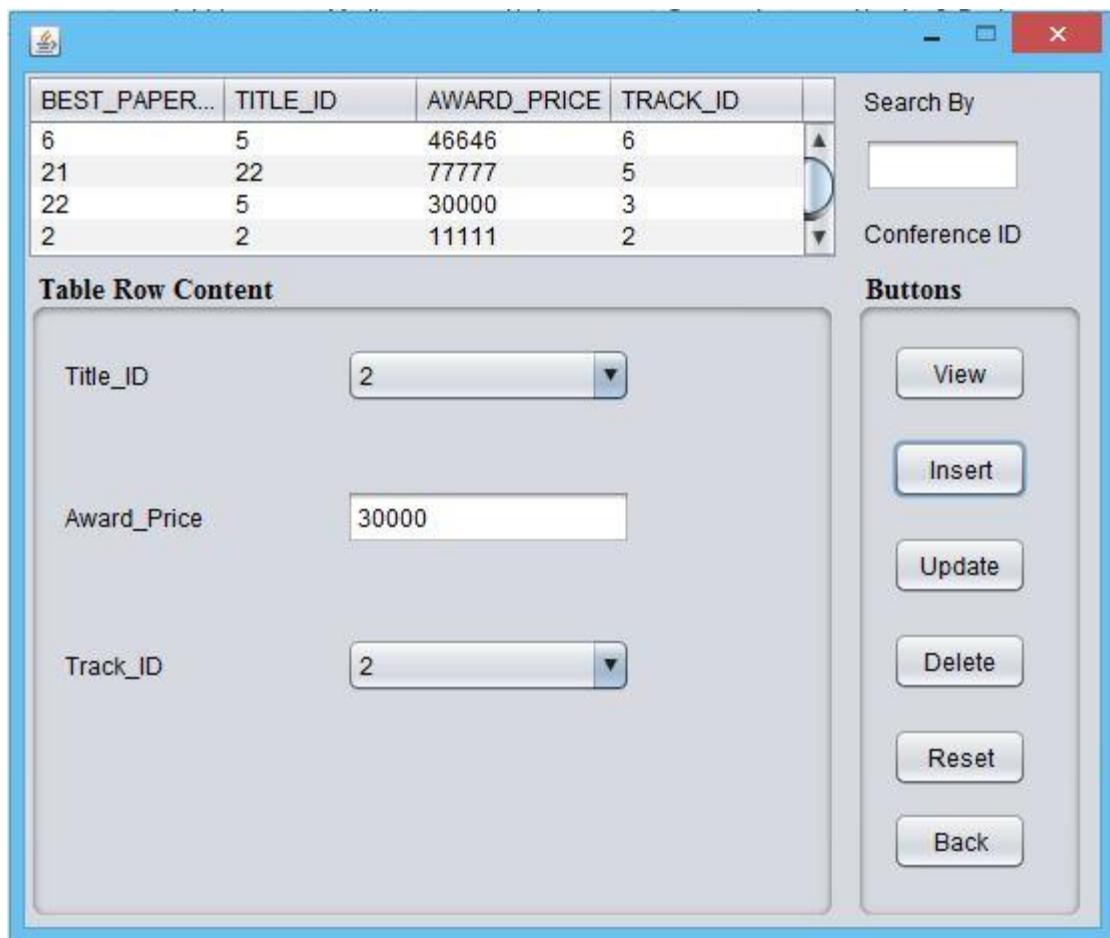
Table Row Content

Search By

Conference ID

Buttons

Data in the table after the execution of Command:



81. Code to delete row in BestPaper table:

```

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rowSelected == true)
    {
        int p = JOptionPane.showConfirmDialog(null, "Do you really want to
delete","Delete",JOptionPane.YES_NO_OPTION );
        if( p == 0)
        {
            try
            {
                Jayesh_Assignment abc = new Jayesh_Assignment();
                Connection conn1 = abc.getConn1();
                int row = Best_Paper_table.getSelectedRow();

```

```

String Table_click = (Best_Paper_table.getModel().getValueAt(row,
0).toString());
String sql = "Delete from BestPaper@FIT5148B where Best_Paper_ID
= '"+Table_click+"'";
pst = conn1.prepareStatement(sql);
rs = pst.executeQuery();
JOptionPane.showMessageDialog(null, "Row is deleted");
}
catch (SQLException | HeadlessException e)
{
    JOptionPane.showMessageDialog(null, e);
}
View_Table_Best_Paper();
rowSelected = false;
}
}
else
{
    JOptionPane.showMessageDialog(null,"Please Select a row");
}
}

```

Select a row and then press delete:

Data in the table before the execution of Command:

Table Row Content

BEST_PAPER...	TITLE_ID	AWARD_PRICE	TRACK_ID
6	5	46646	6
21	22	77777	5
22	5	30000	3
2	2	11111	2

Buttons

Search By

Conference ID

View

Insert

Update

Delete

Reset

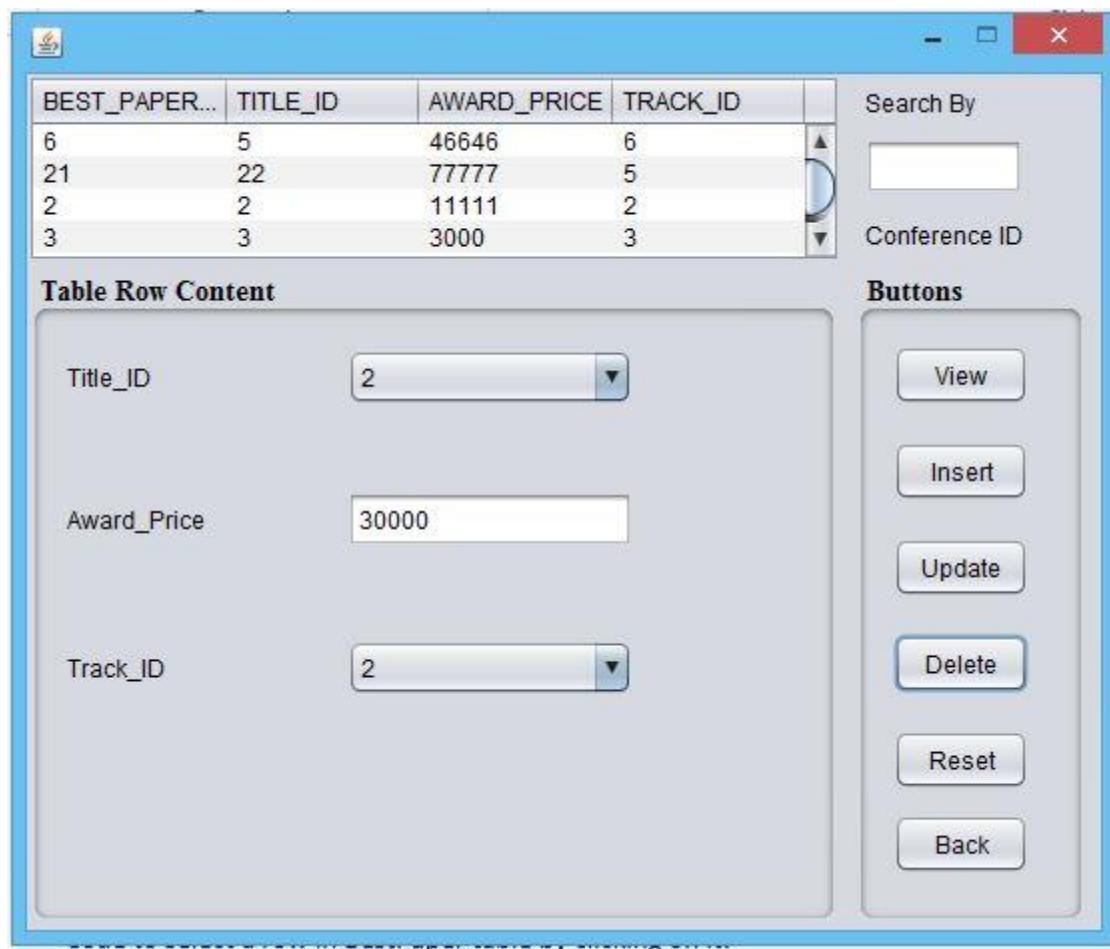
Back

Title_ID: 5

Award_Price: 30000

Track_ID: 5

Data in the table after the execution of Command:



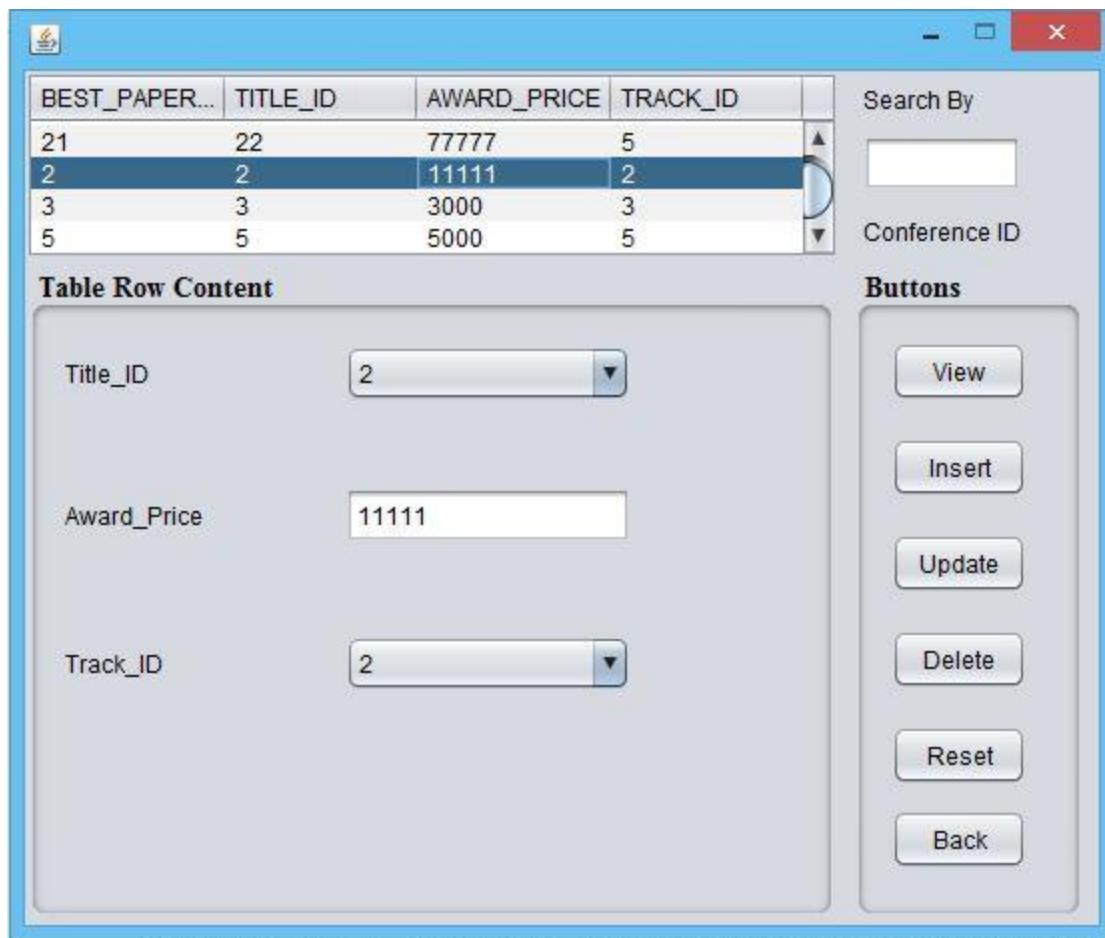
82. Code to select a row in BestPaper table by clicking on it:

```
private void Best_Paper_tableMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try {
        rowSelected = true;
        int row = Best_Paper_table.getSelectedRow();
        String Table_click = (Best_Paper_table.getModel().getValueAt(row,
0).toString());
        Jayesh_Assignment abc = new Jayesh_Assignment();
        Connection conn1 = abc.getConn1();
        String sql ="Select * from BestPaper@FIT5148B where Best_Paper_ID =
"+Table_click+"'";
        pst = conn1.prepareStatement(sql);
        rs = pst.executeQuery();
        if(rs.next())
        {
```

```
        String add1 = rs.getString("Title_ID");
        combo_Title_ID.setSelectedItem(add1);
        String add2 = rs.getString("Award_Price");
        txt_Award_Price.setText(add2);
        String add3 = rs.getString("Track_ID");
        combo_Track_ID.setSelectedItem(add1);
    }

}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}
```

When you click on a row the values in the row are popped in textboxes and combo boxes:



83. Code for selecting rows by Conference ID:

```
private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    try  
    {  
        Jayesh_Assignment abc = new Jayesh_Assignment();  
        Connection conn = abc.getConn();  
        Connection conn1 = abc.getConn1();  
        String sql1 = "Select * from BestPaper@Fit5148B where Title_id IN (Select  
        Title_id from Paper@fit5148b where Track_ID = (Select Track_ID from  
        Track@Fit5148B where Conf_ID = (Select Conf_ID from Conference@Fit5148A  
        where Conf_ID = ?)))";  
        pst1=conn1.prepareStatement(sql1);  
        pst1.setString(1, txtSearch.getText());
```

```

        rs1=pst1.executeQuery();
        Best_Paper_table.setModel(DbUtils.resultSetToTableModel(rs1));

    }
    catch (Exception e7)
    {
        JOptionPane.showMessageDialog(null, e7);
    }
}

```

When you type conference ID in text search field the rows with the same Conference ID are seen in table:

The screenshot shows a Java Swing application window titled "BEST_PAPER...". On the left, there is a table with four columns: "BEST_PAPER...", "TITLE_ID", "AWARD_PRICE", and "TRACK_ID". A single row is visible with values 3, 3, 3000, and 3 respectively. To the right of the table is a search panel labeled "Search By" with a text input field containing the value "3". Below the table is a section titled "Table Row Content" containing three form fields: "Title_ID" with value "2", "Award_Price" with value "11111", and "Track_ID" with value "2". To the right of the form is a vertical stack of buttons labeled "Buttons" with options: View, Insert, Update, Delete, Reset, and Back.

BEST_PAPER...	TITLE_ID	AWARD_PRICE	TRACK_ID
3	3	3000	3

Table Row Content

Title_ID: 2

Award_Price: 11111

Track_ID: 2

Buttons

View

Insert

Update

Delete

Reset

Back

7. Task 6 – Web Based GUI Client

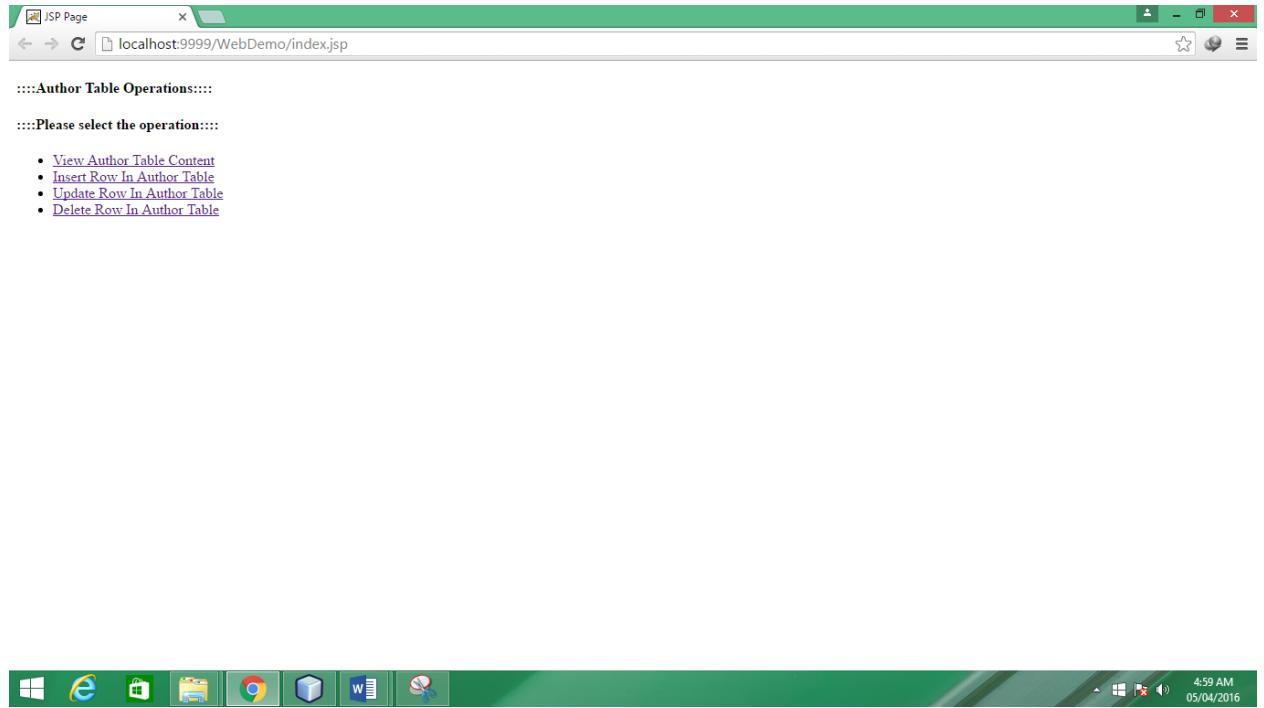
22. Index.jsp:

This page is the main front page. It has various options that can be performed on Author table. Users can select any option.

84. Code for Index.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h4>:::Author Table Operations:::</h4>
    <h4>:::Please select the operation:::</h4>
    <ul>
      <li><a href ="View_Option.jsp">View Author Table Content</a></li>
      <li><a href ="Insert_Option.jsp">Insert Row In Author Table</a></li>
      <li><a href ="Update_Option.jsp">Update Row In Author Table</a></li>
      <li><a href ="Delete_Option.jsp">Delete Row In Author Table</a></li>
    </ul>
  </body>
</html>
```

Screenshot of Web Page:



23. View_Option.jsp:

On this Web page there is a table in which you can see all the values that are already present in the Author Table.

85. Code for View_Option.jsp:

```
<%@page import="java.sql.SQLException"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.beans.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>

<%@ page language="java" import="java.sql.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>View Option</title>
    </head>
    <body>
        <%
```

```

ResultSet rs =null;

try
{
    Connection conn = null;
    Statement st = null;
    PreparedStatement pst=null;

    try
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        conn =
        DriverManager.getConnection("jdbc:oracle:thin:@hippo.its.monash.edu:1521:FIT5148B", "S27148572", "student");
        String sql ="Select * from Author@FIT5148B";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
    }
    catch (SQLException e)
    {
        System.out.println("SQL error " + e);
    }
}
catch(Exception ex)
{
}
}

%>
<TABLE BORDER="1">
<TR>
<TH>Author_ID</TH>
<TH>Name</TH>
<TH>Surname</TH>
<TH>affiliation</TH>
<TH>country</TH>
<TH>email</TH>
<TH>contact</TH>
</TR>
<% while(rs.next()){ %>
<TR>
<TD><%= rs.getString(1) %></TD>
<TD><%= rs.getString(2) %></TD>
<TD><%= rs.getString(3) %></TD>

```

```

<TD><%= rs.getString(4) %></TD>
<TD><%= rs.getString(5) %></TD>
<TD><%= rs.getString(6) %></TD>
<TD><%= rs.getString(7) %></TD>

</TR>
<% } %>
</TABLE>
</body>
</html>

```

Screenshot:

Author_ID	Name	Surname	affiliation	country	email	contact
2	Molly	Bliss	Dallas University	USA	Bliss@gmail.com	45845878
3	Steve	Austin	Texas University	Russia	Austin@gmail.com	45833373
4	Kurt	Angle	Berlin University	Germany	Angle@gmail.com	45869845
5	Jorja	Iswaku	Venice University	Italy	Iswaku@gmail.com	45869555
6	Kurt	starphilip	Dallas University	USA	Bliss@gmail.com	45845878
14	jonny	kld	jnkj	klnkjkl	nknkh	76567

24. Insert_Option.jsp:

On this Web page there is a table in which you can insert all the values that you want to add to the Author Table.

86. Code for Insert_Option.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Insert Option</title>
  </head>
  <body>
    <form action="insert.jsp" method="POST">
      <table border="1">

        <tbody>
          <tr>
            <td>Author Id :</td>
            <td><input type="text" name="ID" value="" size="50" /></td>
          </tr>
          <tr>
            <td>Author Name :</td>
            <td><input type="text" name="first" value="" size="50" /></td>
          </tr>
          <tr>
            <td>Author Surname :</td>
            <td><input type="text" name="last" value="" size="50" /></td>
          </tr>
          <tr>
            <td>Affiliation :</td>
            <td><input type="text" name="affiliation" value="" size="50" /></td>
          </tr>
          <tr>
            <td>Country :</td>
            <td><input type="text" name="country" value="" size="50" /></td>
          </tr>
          <tr>
            <td>Email :</td>
            <td><input type="text" name="email" value="" size="50" /></td>
          </tr>
          <tr>
            <td>Contact number :</td>
            <td><input type="text" name="number" value="" size="50" /></td>
          </tr>
        </tbody>
      </table>
    </form>
  </body>
</html>

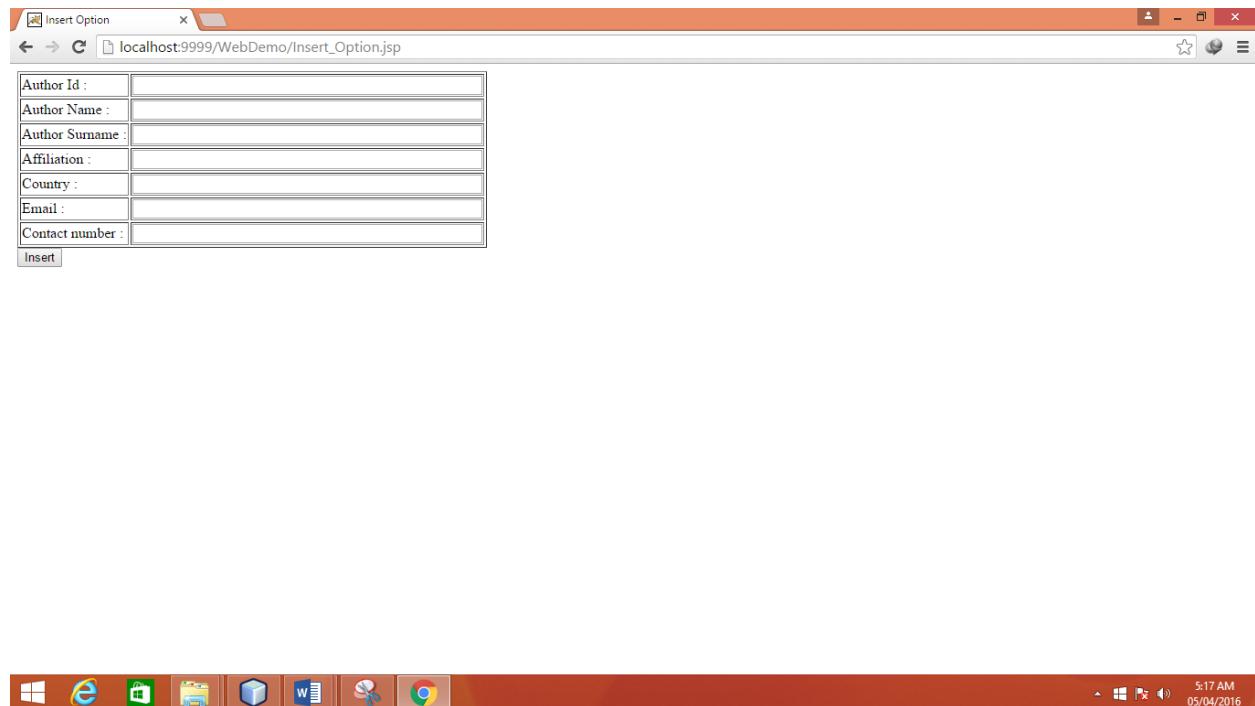
```

```

        </tbody>
    </table>
    <input type="submit" value="Insert" name="Insert" />
</form>
</body>
</html>

```

Screenshot:



25. Update_Option.jsp:

On this Web page there is a table in which you can insert all the values that you want to update to the Author Table.

87. Code for Update_Option.jsp:

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Update Option</title>

```

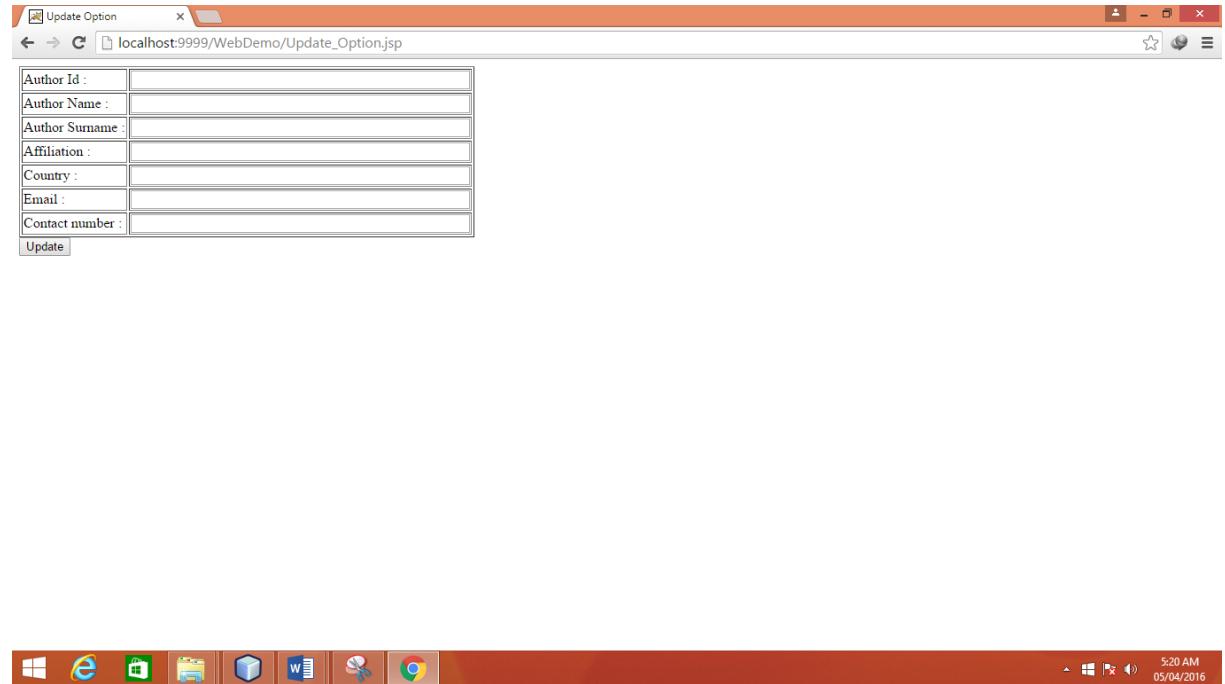
```

</head>
<body>
<form action="update.jsp" method="POST">
<table border="1">

<tbody>
<tr>
<td>Author Id :</td>
<td><input type="text" name="ID" value="" size="50" /></td>
</tr>
<tr>
<td>Author Name :</td>
<td><input type="text" name="first" value="" size="50" /></td>
</tr>
<tr>
<td>Author Surname :</td>
<td><input type="text" name="last" value="" size="50" /></td>
</tr>
<tr>
<td>Affiliation :</td>
<td><input type="text" name="affiliation" value="" size="50" /></td>
</tr>
<tr>
<td>Country :</td>
<td><input type="text" name="country" value="" size="50" /></td>
</tr>
<tr>
<td>Email :</td>
<td><input type="text" name="email" value="" size="50" /></td>
</tr>
<tr>
<td>Contact number :</td>
<td><input type="text" name="number" value="" size="50" /></td>
</tr>
</tbody>
</table>
<input type="submit" value="Update" name="Update" />
</form>
</body>

```

```
</html>
```



26. Delete_Option.jsp:

On this Web page there is a table in which you can insert the Author_ID that you want to delete.

88. Code for Delete_Option.jsp:

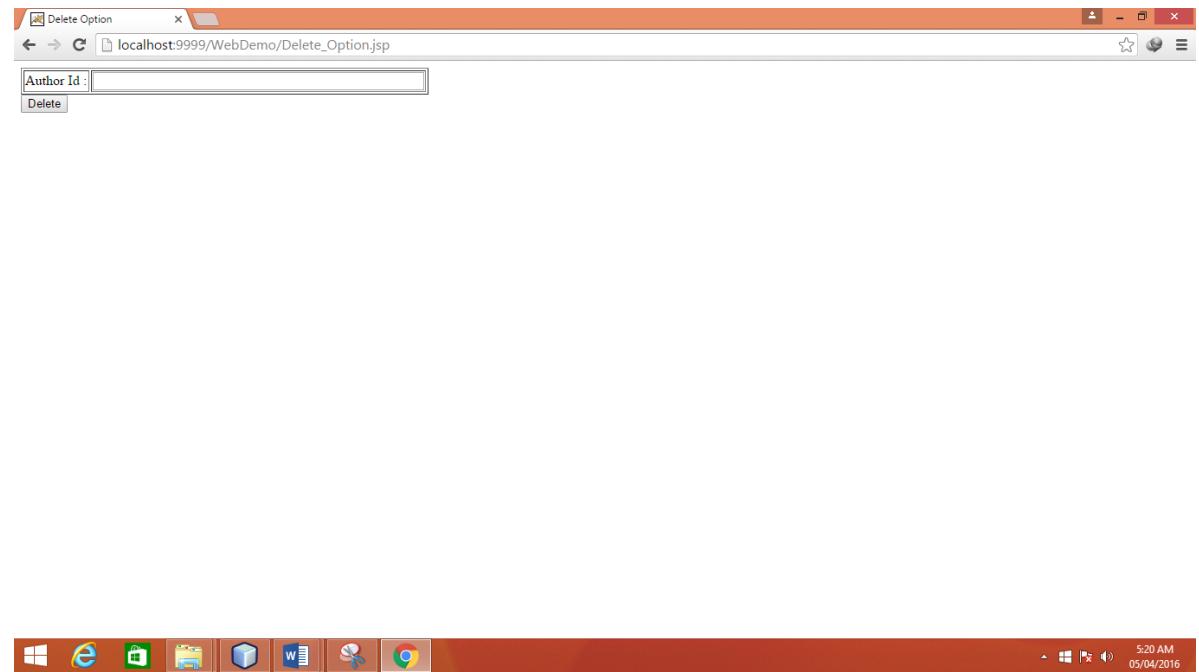
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Delete Option</title>
    </head>
    <body>
        <form action="delete.jsp" method="POST">
            <table border="1">

                <tbody>
                    <tr>
```

```

<td>Author Id :</td>
<td><input type="text" name="ID" value="" size="50" /></td>
</tr>
</tbody>
</table>
<input type="submit" value="Delete" name="Delete" />
</form>
</body>
</html>

```



27. Insert.jsp:

The Sql Code for the Insert operation to take place is in this file.

89. Code for the page:

```

<%@page import="java.sql.SQLException"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.beans.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>

```

```

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Insert</title>
</head>
<body>
<%
try
{
    ResultSet rs =null;
    Connection con = null;
    Statement st = null;
    PreparedStatement ps=null;

    try
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con =
DriverManager.getConnection("jdbc:oracle:thin:@hippo.its.monash.edu:1521:FIT5148B
", "S27148572", "student");
        System.out.println("connection established");
    }
    catch (SQLException e)
    {
        System.out.println("SQL error " + e);
    }

String id = new String();
String fname = new String();
String lname = new String();
String affi = new String();
String country = new String();
String email = new String();
String num = new String();

if(request.getParameter("ID")!=null)
{
    id=request.getParameter("ID");
}
if(request.getParameter("first")!=null)
{
    fname=request.getParameter("first");
}

```

```

        }
        if(request.getParameter("last")!=null)
        {
            lname=request.getParameter("last");
        }
        if(request.getParameter("affiliation")!=null)
        {
            affi=request.getParameter("affiliation");
        }
        if(request.getParameter("country")!=null)
        {
            country=request.getParameter("country");
        }
        if(request.getParameter("email")!=null)
        {
            email=request.getParameter("email");
        }
        if(request.getParameter("number")!=null)
        {
            num=request.getParameter("number");
        }
    
```

```

String sql=("Insert into Author@FIT5148B (Author_ID, Author_name,
Author_Surname, Author_Affiliation, Author_country, Author_email, Author_number)"
+ "values ("+id+"','"+fname+"", "+lname+", "+affi+", "+country+", "+email+",
"+num+ ")");
int r=con.createStatement().executeUpdate(sql);
con.setAutoCommit(true);
if(r>=1)
{
    out.println("saved");
}
else
{
    out.println("cannot save");
}
catch(Exception e)
{
    e.printStackTrace();
}
    
```

%>

```
</body>
</html>
```

28. Update.jsp:

The Sql Code for the Update operation to take place is in this file.

90. Code for the page:

```
<%@page import="java.sql.SQLException"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.beans.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Update</title>
    </head>
    <body>
        <%
        try
        {
            ResultSet rs =null;
            Connection con = null;
            Statement st = null;
            PreparedStatement ps=null;

            try
            {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                con =
                    DriverManager.getConnection("jdbc:oracle:thin:@hippo.its.monash.edu:1521:FIT5148B",
                    "S27148572", "student");
                System.out.println("connection established");
            }
            catch (SQLException e)
            {

```

```

        System.out.println("SQL error " + e);
    }

String id = new String();
String fname = new String();
String lname = new String();
String affi = new String();
String country = new String();
String email = new String();
String num = new String();

if(request.getParameter("ID")!=null)
{
    id=request.getParameter("ID");
}
if(request.getParameter("first")!=null)
{
    fname=request.getParameter("first");
}
if(request.getParameter("last")!=null)
{
    lname=request.getParameter("last");
}
if(request.getParameter("affiliation")!=null)
{
    affi=request.getParameter("affiliation");
}
if(request.getParameter("country")!=null)
{
    country=request.getParameter("country");
}
if(request.getParameter("email")!=null)
{
    email=request.getParameter("email");
}
if(request.getParameter("number")!=null)
{
    num=request.getParameter("number");
}

String sql = "Update Author@FIT5148B set Author_name ='" + fname + "',  

Author_Surname ='" + lname + "', Author_Affiliation ='" + affi + "','" + "

```

```

Author_Country ="" + country + ",'" + "Author_email ="" + email + ",'" +
"Author_number ="" + num + "" where Author_ID = ""+id+""";
    int r=con.createStatement().executeUpdate(sql);
    con.setAutoCommit(true);
    if(r>=1)
    {
        out.println("saved");
    }
    else
    {
        out.println("cannot save");
    }
}
catch(Exception e)
{
    e.printStackTrace();
}

%>

</body>
</html>

```

29. Delete.jsp:

The Sql Code for the Delete operation to take place is in this file.

91. Code for the page:

```

<% @page import="java.sql.SQLException"%>
<% @page import="java.sql.DriverManager"%>
<% @page import="java.sql.PreparedStatement"%>
<% @page import="java.beans.Statement"%>
<% @page import="java.sql.Connection"%>
<% @page import="java.sql.ResultSet"%>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Delete</title>
    </head>
    <body>

```

```

<%
try
{
    ResultSet rs=null;
    Connection con = null;
    Statement st = null;
    PreparedStatement ps=null;

    try
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con =
DriverManager.getConnection("jdbc:oracle:thin:@hippo.its.monash.edu:1521:FIT5148B
", "S27148572", "student");
        System.out.println("connection established");
    }
    catch (SQLException e)
    {
        System.out.println("SQL error " + e);
    }
}

```

```
String id = new String();
```

```

if(request.getParameter("ID")!=null)
{
    id=request.getParameter("ID");
}

```

```

String sql = "Delete from Author@FIT5148B where Author_ID ='"+id+"'";
int r = con.createStatement().executeUpdate(sql);
con.setAutoCommit(true);
if(r>=1)
{
    out.println("Deleted");
}
else
{
    out.println("cannot delete");
}

```

```
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }

%>

</body>
</html>
```