

1. Create a JDBC Connectivity and perform CRUD Operation for the following Schema. COURSE ID NAME INSTRUCTOR DURATION FEES\

Insert

```
1 package com.Java.Connection;
2
3 import java.sql.*;
4
5 public class Insert {
6     public static void main(String[] args) {
7         //load the driver
8         try {
9             Class.forName("com.mysql.jdbc.Driver");
10        } catch (ClassNotFoundException e) {
11
12            // TODO: handle exception
13            e.printStackTrace();
14        }
15        try(Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/idefc","root","Amity1999@1");
16            Statement st = con.createStatement();) {
17
18            String query = "INSERT INTO course_id(NAME,INSTRUCTOR,DURATION,FEES) VALUES('A','A','18:24:23',20000)";
19
20            int count = st.executeUpdate(query);
21            if (count==0) {
22                System.out.println("Record Not inserted");
23            }else {
24                System.out.println("Record inserted");
25            }
26        }catch (SQLException se) {
27            System.out.println(se.getMessage());
28            se.printStackTrace();
29        }catch (Exception e) {
30            System.out.println(e.getMessage());
31            e.printStackTrace();
32        }
33    }
34 }
35
```

Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and loaded on demand. Record inserted

SCHEMAS

Filter objects

- idfc
 - Tables
 - course_id
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions
 - student
 - sys

1 • `SELECT * FROM idfc.course_id;`

Limit to 1000 rows

Result Grid

	NAME	INSTRUCTOR	DURATION	FEES
▶	A	A	18:24:23	20000
	A	A	18:24:23	20000
	B	A	18:24:13	30000

Delete

```

1 package com.Java.Connection;
2 import java.sql.*;
3
4 public class Delete {
5     public static void main(String[] args) {
6         //load the driver
7         try {
8             Class.forName("com.mysql.jdbc.Driver");
9         } catch (ClassNotFoundException e) {
10
11             // TODO: handle exception
12             e.printStackTrace();
13         }
14         try(Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/dfc","root","Amity1999@1");
15             Statement st = con.createStatement()); {
16
17             String query = "DELETE FROM course_id WHERE NAME = 'A'";
18
19             int count = st.executeUpdate(query);
20             if (count==0) {
21                 System.out.println("Record Not Deleted");
22             } else {
23                 System.out.println("Record Deleted");
24             }
25         } catch (SQLException se) {
26             System.out.println(se.getMessage());
27             se.printStackTrace();
28         } catch (Exception e) {
29             System.out.println(e.getMessage());
30             e.printStackTrace();
31         }
32     }
33 }
34

```

Loading class "com.mysql.jdbc.Driver". This is deprecated. The new driver class is "com.mysql.cj.jdbc.Driver". The driver is automatically registered via the SP
Record Deleted

SCHEMAS

Filter objects

- idfc
 - Tables
 - course_id
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions
 - student
 - sys

1 • SELECT * FROM idfc.course_id;

Limit to 1000 rows

Result Grid

	NAME	INSTRUCTOR	DURATION	FEES
▶	B	A	18:24:13	30000

UPDATE
(Before Update database is)

SCHEMAS

Filter objects

- idfc
 - Tables
 - course_id
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions
 - student
 - sys

1 • `SELECT * FROM idfc.course_id;`

Limit to 1000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [↗](#)

	NAME	INSTRUCTOR	DURATION	FEES
▶	B	A	18:24:13	30000
	B	A	18:14:23	32000
	C	A	18:14:23	32300
	D	A	18:14:23	32300

```

1 package com.Java.Connection;
2 import java.sql.*;
3 public class Update {
4     public static void main(String[] args) {
5         //load the driver
6         try {
7             Class.forName("com.mysql.jdbc.Driver");
8         } catch (ClassNotFoundException e) {
9
10            // TODO: handle exception
11            e.printStackTrace();
12        }
13        try(Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/idfc","root","Amity1999@1");
14            Statement st = con.createStatement();) {
15
16            String query = "UPDATE course_id SET FEES = 45000 WHERE NAME = 'D'";
17
18            int count = st.executeUpdate(query);
19            if (count==0) {
20                System.out.println("Record Not Updated");
21            }else {
22                System.out.println("Record Updated");
23            }
24        } catch (SQLException se) {
25            System.out.println(se.getMessage());
26            se.printStackTrace();
27        } catch (Exception e) {
28            System.out.println(e.getMessage());
29            e.printStackTrace();
30        }
31    }
32 }
33 }

```

<terminated> Update [Java Application] C:\Users\Amit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_18.0.2.v20220903-1139\jre
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. T
Record Updated

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' panel displays a tree view of the database structure. The 'idfc' database is expanded, showing a table named 'course_id'. The table structure is as follows:

NAME	INSTRUCTOR	DURATION	FEES
B	A	18:24:13	30000
B	A	18:14:23	32000
C	A	18:14:23	32300
D	A	18:14:23	45000

On the right, the SQL editor shows the query: `SELECT * FROM idfc.course_id;`. Below the editor, the 'Result Grid' displays the query results in a table format, matching the data shown in the schema panel.

2. Describe the COURSE DETAIL Stable.

- Student table has to be created based on the above schema (i)
Name should not be null

SCHEMAS

Filter objects

- idfc
 - Tables
 - course
 - course_detail
 - course_id
 - Views
 - Stored Procedures
 - Functions
- student
 - Tables
 - personal
 - product
 - Views
 - Stored Procedures
 - Functions
- sys

1 • `SELECT * FROM idfc.course_detail;`

Limit to 1000 rows

Result Grid

	STUDENT_ID	NAME	PLACE	CGPA	CID
1	1	a	muMBAU	10	1
2	2	B	mUMBAI	9	2

b. Describe the student table

SCHEMAS

Filter objects

- idfc
 - Tables
 - course
 - course_detail
 - course_id
 - Views
 - Stored Procedures
 - Functions
- student
 - Tables
 - personal
 - product
 - Views
 - Stored Procedures
 - Functions
- sys

1 • `DESCRIBE COURSE_DETAIL`

Limit to 1000 rows

Result Grid

	Field	Type	Null	Key	Default	Extra
1	STUDENT_ID	int	YES		NULL	
2	NAME	varchar(5)	NO		NULL	
3	PLACE	varchar(10)	YES		NULL	
4	CGPA	int	YES		NULL	
5	CID	int	YES		NULL	

C. Add an extra column credit in the course table (Note: a single digit number will be stored).

f. Rename the COURSE_DETAILS as COURSE.

Navigator

SCHEMAS

Filter objects

idfc

Tables

course

course_detail

course_id

Views

Stored Procedures

student

Tables

personal

product

Views

Stored Procedures

Functions

sys

1*

course_id

course_id

student - Schema

personal

course_id

course_detail

course_detail

Limit to 1000 rows

1 • ALTER TABLE course_detail RENAME TO COURSES;

1 • SELECT * FROM idfc.courses;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	STUDENT_ID	NAME	PLACE	CGPA	CID
▶	1	a	muMBAU	10	1
	2	B	mUMBAI	9	2