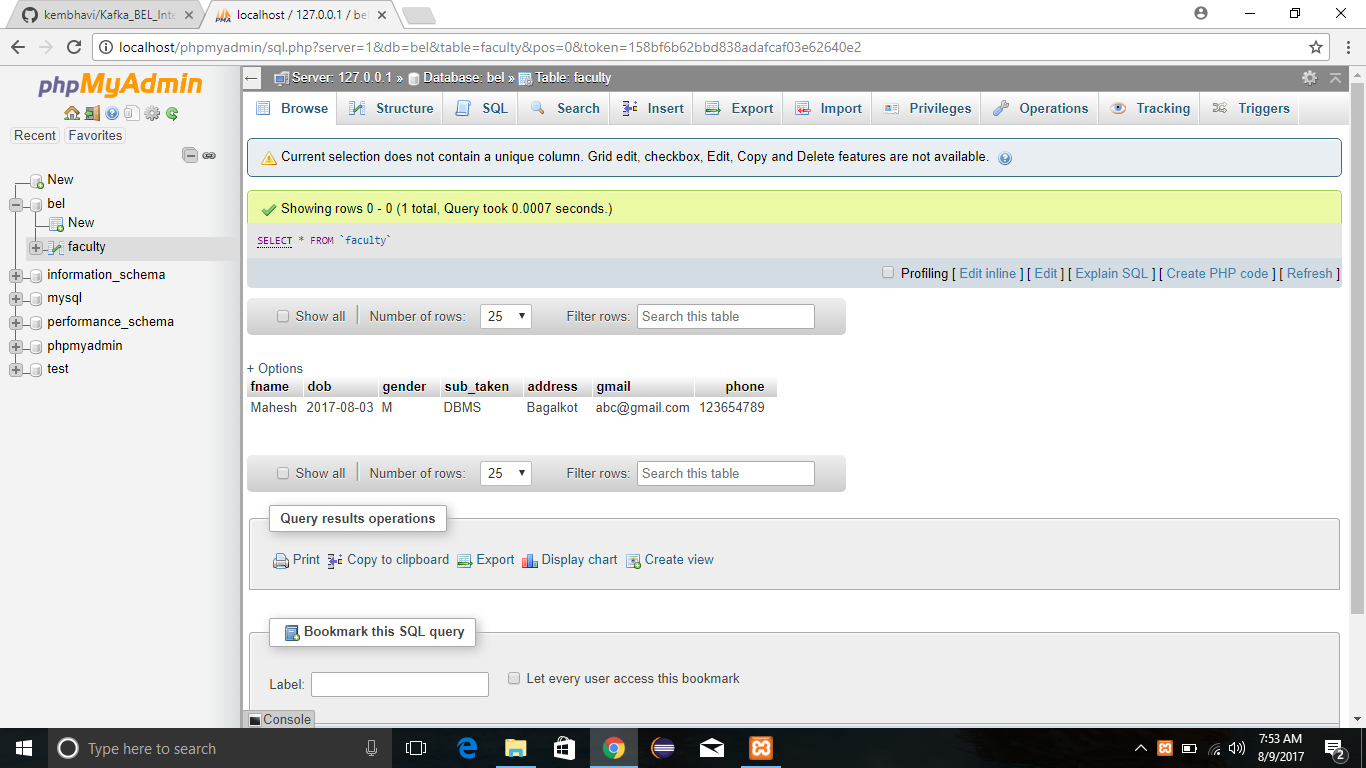
**JDBC Coonection**

1. **Download Java MySQL Connector *‘mysql-connector-java-5.1.34-bin.jar’* and keep it on the desktop**
2. **Create a Java project in Eclipse IDE**

* Open Eclipse IDE. Create a new Java Project and name it as “mydbproj”.

1. **Configure JDBC driver in Eclipse IDE**

* You need to add the downloaded Java MySQL Connector JAR in client **project’s classpath**. To do this,**right click on your Java Project (mydbproj) -> Properties -> Buildpath -> Libraries -> Add External JAR and select “mysql-connector-java-5.1.34-bin.jar” JAR file**.



1. **Set up a simple Consumer program**

**import** java.util.Arrays;

**import** java.util.Properties;

**import** org.apache.kafka.clients.consumer.ConsumerRecord;

**import** org.apache.kafka.clients.consumer.ConsumerRecords;

**import** org.apache.kafka.clients.consumer.KafkaConsumer;

**public** **class** Consumer {

**public** **static** **void** main(String[] args) {

Properties props = **new** Properties();

props.put("bootstrap.servers", "localhost:9092");

props.put("group.id", "group-1");

props.put("enable.auto.commit", "true");

props.put("auto.commit.interval.ms", "100");

props.put("auto.offset.reset", "earliest");

props.put("auto.commit.interval.ms", "1000");

props.put("session.timeout.ms", "30000");

props.put("key.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");

props.put("value.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");

KafkaConsumer<String, String> kafkaConsumer = **new** KafkaConsumer<>(props);

kafkaConsumer.subscribe(Arrays.*asList*("faculty"));

**while** (**true**) {

ConsumerRecords<String, String> records = kafkaConsumer.poll(10);

**for** (ConsumerRecord<String, String> record : records) {

System.***out***.println("Partition: " + record.partition() + " Offset: " + record.offset()

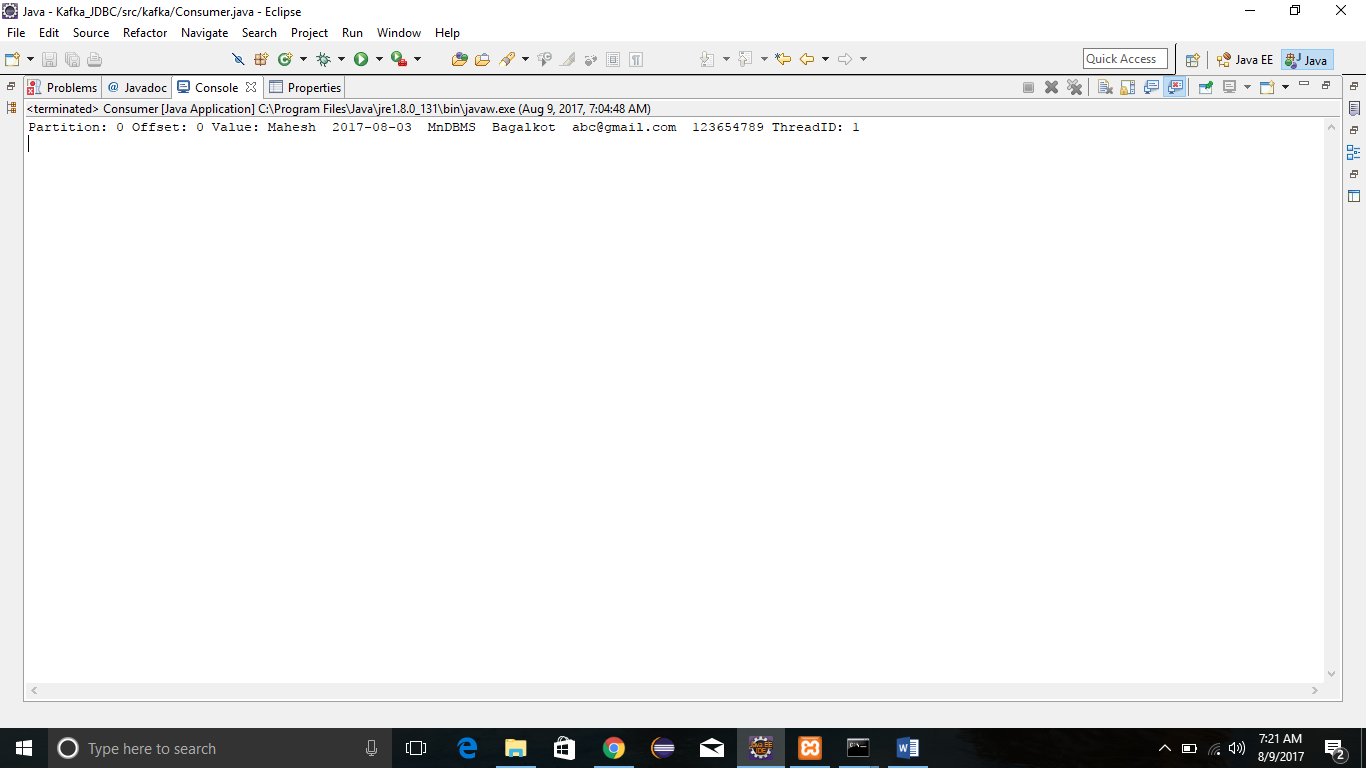
+ " Value: " + record.value() + " ThreadID: " + Thread.*currentThread*().getId());

}

}

}

}

1. **Run the program ->click on Java file -> RunAs-> Java Application. You will get the following output.** 

**Producer\_program**

package kafka;

import java.io.File;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.util.Properties;

import javax.swing.text.Document;

import org.apache.kafka.clients.producer.KafkaProducer;

import org.apache.kafka.clients.producer.Producer;

import org.apache.kafka.clients.producer.ProducerRecord;

import com.mysql.jdbc.Statement;

public class ProducerTest {

public static void main(String[] args) {

Properties props = new Properties();

props.put("bootstrap.servers", "localhost:9092");

props.put("acks", "all");

props.put("retries", 0);

props.put("batch.size", 16384);

props.put("linger.ms", 1);

props.put("max.block.ms", 1000);

props.put("buffer.memory", 33554432);

props.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");

props.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");

Producer<String, String> producer = null;

try {

producer = new KafkaProducer<String, String>(props);

String msg=null;

Connection con=null;

Class.forName("com.mysql.jdbc.Driver").newInstance();

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bel", "root", "");

if (!con.isClosed())

System.out.println("Successfully connected to MySQL server...");

String query="select \* from faculty";

Statement stmt = (Statement) con.createStatement();

ResultSet rs = stmt.executeQuery(query);

String f1,f2,f3,f4,f5,f6,f7;

while (rs.next())

{

f1 = rs.getString(1);

f2 = rs.getString(2);

f3 = rs.getString(3);

f4 = rs.getString(4);

f5 = rs.getString(5);

f6 = rs.getString(6);

f7 = rs.getString(7);

msg= f1+" "+f2+" "+f3+"n"+f4+" "+f5+" "+f6+" "+f7;

System.out.println(msg);

producer.send(new ProducerRecord<String, String>("faculty", msg));

}

con.close();

} catch (Exception e) {

e.printStackTrace();

} finally {

producer.close();

}

}

}