SCRIPT :INSERT VALUES TO SCHEDULAR TABLE

```
const Chance = require('chance');
//const db = require('../mysql-store/mysql-connection');
const mysql = require('mysql2/promise');
const chance = new Chance();


function convertToTimeStamp(isoString) {
  if (isoString) {
    const timeStamp = isoString.split('.')[0];
    return timeStamp.replace('T', ' ');
  }
  return null;
}

function getValues(eventId, batchSize) {
  let schedulerJobInstanceValues = [];
  console.log('bbbbb')
  for (let i = 0; i < batchSize; i++) {
    let clientId = chance.guid();
    // let jobId = 'e550f504-55f6-5da0-8b2d-9e13c3a83ca7';
    let executionTime = convertToTimeStamp(new Date().toISOString());
    let entityId = chance.guid();
    //let eventId = chance.guid();
    let businessEventId = chance.guid();
    //let triggerType = 'automatic';
    let status = null;
    //let execData = `{"status": "Completed", "execRequestId": "$
{execRequestId}"}`;
     let dateCreated = convertToTimeStamp(new Date().toISOString());
     let lastModified = convertToTimeStamp(new Date().toISOString());

    schedulerJobInstanceValues.push([
        clientId, eventId, businessEventId, entityId, executionTime, status,
dateCreated, lastModified
    ]);
  }

  return schedulerJobInstanceValues;
}

async function insertInstance(eventId, recordsToBeInserted, batchSize = 500) {
  //  let connection;
  while (recordsToBeInserted) {
    const query = 'INSERT INTO ScheduledBusinessEvents(clientId, eventId,
businessEventId, entityId, executionTime, status, dateCreated, lastModified)
VALUES ?';
    const values = getValues(eventId, batchSize);

    try {
            const connection = await mysql.createConnection({
              host: '127.0.0.1',
              database: 'business_events',
              user: 'rootroot',
              password: 'rootroot',
              port: '5506',
            });

     const y =  await connection.query(query, [values]);
      console.log('yyyy',y)
    } catch (error) {
      console.log('failed to create new instane', error);
    }
    recordsToBeInserted -= batchSize;
  }
}

let eventId = 'ee70a2493f06481e806748d1c39b70c0';
let noOfRecords = 5000;
let batchSize = 5000;

insertInstance(eventId, noOfRecords, batchSize)
  .then(() => console.log('inserted!!!'));
```