```java
package fi.paatti.paattidatabaseutils.dbservice.querydelegates;


import com.vaadin.data.util.sqlcontainer.RowItem;
import com.vaadin.data.util.sqlcontainer.query.generator.StatementHelper;
import fi.paatti.paattidatabaseutils.names.PaattiColumnNames;
import fi.paatti.paattidatabaseutils.names.PaattiTableNames;
import java.io.Serializable;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;


/**
 * A delegate for combining the EVENTS and SCHEDULE table related queries.
 *
 * @author Lauri Satokangas, lauri.n.satokangas@student.jyu.fi
 */
public class EventsInScheduleDelegate extends PaattiQueryDelegate implements Serializable {

  private static final long serialVersionUID = 1L;
  private final Object scheduleID;

  /**
   * Constructor.
   *
   * @param scheduleID The ID of the schedule.
   */
  public EventsInScheduleDelegate(Object scheduleID) {
    super(PaattiTableNames.EVENT, PaattiColumnNames.EVENT_eventID,
        PaattiColumnNames.getColumnNamesInArray(PaattiTableNames.EVENT));
    this.scheduleID = scheduleID;

  }

  /**
   * {@inheritDoc }
   * <p/>
   * Not in use.
   */
  @Override
  public int storeRow(Connection conn, RowItem row) throws SQLException {
    throw new UnsupportedOperationException("Not supported yet.");
  }

  /**
```

```
   * {@inheritDoc }
   * <p/>
   * Not in use.
   */
  @Override
  protected void setRowValues(PreparedStatement statement, RowItem row) throws SQLException {
    throw new UnsupportedOperationException("Not supported yet.");
  }

  /**
   * Returns a new StatementHelper containing the SELECT query statement.
   *
   * @param offset the starting row.
   * @param limit the number of rows to get.
   * @return The StatementHelper containing the SELECT query.
   * @throws UnsupportedOperationException
   */
  public StatementHelper getQueryStatement(int offset, int limit) throws UnsupportedOperationException {
    StatementHelper queryHelper = new StatementHelper();

    queryHelper.setQueryString(createQueryString(queryHelper, offset, limit));
    return queryHelper;
  }

  /**
   * Creates the SELECT query statement.
   *
   * @param queryHelper
   * @param offset the starting row.
   * @param limit the number of rows to get.
   * @return The SELECT query string.
   */
  private String createQueryString(StatementHelper queryHelper, int offset, int limit) {
    StringBuilder querySb = new StringBuilder();

    querySb.append(" SELECT * FROM ").append(PaattiTableNames.EVENT);
    querySb.append(" LEFT OUTER JOIN ").append(PaattiTableNames.EVENTTIME).append(" ON ");
    querySb.append(PaattiColumnNames.EVENT_EVENTTIME_eventTimeID).append(" = ").append(
        PaattiColumnNames.EVENTTIME_eventTimeID);
    querySb.append(" WHERE ").append(PaattiColumnNames.EVENT_SCHEDULE_scheduleID).append(" = ").append(scheduleID);
    querySb.append(" AND ").append(PaattiColumnNames.EVENT_rowStatus).append(" = '").append(ROWSTATUS_ACTIVE).append("'");

    if (getFilters() != null && !getFilters().isEmpty()) {
      querySb.append(getFilterStringWithoutWhere(queryHelper, true));
    }
```

```
    return querySb.toString();
  }

  /**
   * {@inheritDoc }
   */
  @Override
  public StatementHelper getCountStatement() throws UnsupportedOperationException {
    StatementHelper countHelper = new StatementHelper();
    StringBuilder querySb = new StringBuilder("SELECT COUNT(*) FROM (");

    querySb.append(createQueryString(countHelper, -1, -1)).append(") as COUNT");
    System.out.println(querySb.toString());
    countHelper.setQueryString(querySb.toString());
    return countHelper;
  }
}
```