

```
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.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

/**
 * A delegate for BELONGS-table.
 *
 * @author Lauri Satokangas, lauri.n.satokangas@student.jyu.fi
 * @author Tapio Keränen, t.tapio.keranen@student.jyu.fi
 */
public class BelongsQueryDelegate extends PaattiQueryDelegate {
    private static final long serialVersionUID = 1L;

    private Object groupId, userId;

    /**
     * BelongsQueryDelegate constructor.
     *
     * @param groupId The ID of the group.
     * @param userId The ID of the user.
     */
    public BelongsQueryDelegate(Object groupId, Object userId) {
        super(PaattiTableNames.BELONGS, PaattiColumnNames.BELONGS_editTime,
            PaattiColumnNames.getColumnNamesInArray(PaattiTableNames.BELONGS));
    }

    this.groupId = groupId;
    this.userId = userId;
}

/**
 * {@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();
    StringBuilder querySb = new StringBuilder();

    querySb.append("SELECT * FROM ").append(PaattiTableNames.BELONGS);
    querySb.append(" WHERE ").append(PaattiColumnNames.BELONGS_USERGROUPID).append(" = ").append(groupId).append(
        " AND ");
    querySb.append(PaattiColumnNames.BELONGS_USERID).append(" = ").append(userId);

    queryHelper.setQueryString(querySb.toString());
    return queryHelper;
}

/**
 * {@inheritDoc}
 */
@Override
public boolean removeRow(Connection conn, RowItem row) throws SQLException {
    StringBuilder removeQuerySb = new StringBuilder();

    removeQuerySb.append("UPDATE ").append(getTableName()).append(" SET ") .append(getTableName() + " _rowstatus").append(
        " = ?");
    removeQuerySb.append(" WHERE ").append(PaattiColumnNames.BELONGS_USERGROUPID).append(" = ").append(groupId);
    removeQuerySb.append(" AND ").append(PaattiColumnNames.BELONGS_USERID).append(" = ").append(userId);

    PreparedStatement removeStatement = conn.prepareStatement(removeQuerySb.toString());
}
```

```
removeStatement.setString(1, ROWSTATUS_DELETED);  
int rowsChanged = removeStatement.executeUpdate();  
removeStatement.close();  
return rowsChanged == 1;  
}  
}
```