package com.cgi.cacsg.batch.domain.dao;

/\*\*\*

\* Generated DAO Code \*

\*/

import java.util.List;

import com.cgi.cacsg.batch.domain.dto.ArAdjustment;

public interface ArAdjustmentDao {

public static final String TABLE\_NAME = "AR\_ADJUSTMENT";

/\*\*

\* Insert a record in the table.

\*

\* @param ArAdjustment

\* @return The number of records inserted.

\*/

public int insert(ArAdjustment arAdjustment);

/\*\*

\* Update a record in the table.

\*

\* @param ArAdjustment

\* @return The number of records updated.

\*/

public int updateByKey(ArAdjustment arAdjustment);

/\*\*

\* Delete a record in the table.

\*

\* @param ArAdjustment

\* @return The number of records deleted.

\*/

public int deleteByKey(ArAdjustment arAdjustment);

/\*\*

\* Select a record from the database.

\*

\* @param adjustmentId

\* @return The database record. Null if the record does not exist.

\*/

public ArAdjustment selectByKey(String adjustmentId);

/\*\*

\* Insert a batch of records. </br>

\* </br>

\* WARNING: This method will return -2 in the array elements for success, NOT the number

\* of rows inserted. The Oracle documentation reads as follows: </br>

\* </br>

\* For a prepared statement batch, it is not possible to know the number of rows affected in the database by each individual statement in the batch. Therefore, all array elements have a value of -2. According to the JDBC 2.0 specification, a value of -2 indicates that the operation was successful but the number of rows affected is unknown.

\*

\* @param ArAdjustmentList

\* @return Array of integers for the number of rows inserted.

\*/

public int[] insertBatch(List<ArAdjustment> arAdjustmentList);

/\*\*

\* Update a batch of records. </br>

\* </br>

\* WARNING: This method will return -2 in the array elements for success, NOT the number

\* of rows updated. The Oracle documentation reads as follows: </br>

\* </br>

\* For a prepared statement batch, it is not possible to know the number of rows affected in the database by each individual statement in the batch. Therefore, all array elements have a value of -2. According to the JDBC 2.0 specification, a value of -2 indicates that the operation was successful but the number of rows affected is unknown.

\*

\* @param ArAdjustmentList

\* @return Array of integers for the number of rows updated.

\*/

public int[] updateBatchByKey(List<ArAdjustment> arAdjustmentList);

/\*\*

\* Delete a batch of records. </br>

\* </br>

\* WARNING: This method will return -2 in the array elements for success, NOT the number

\* of rows deleted. The Oracle documentation reads as follows: </br>

\* </br>

\* For a prepared statement batch, it is not possible to know the number of rows affected in the database by each individual statement in the batch. Therefore, all array elements have a value of -2. According to the JDBC 2.0 specification, a value of -2 indicates that the operation was successful but the number of rows affected is unknown.

\*

\* @param ArAdjustmentList

\* @return Array of integers for the number of rows deleted.

\*/

public int[] deleteBatchByKey(List<ArAdjustment> arAdjustmentList);

/\*\*

\* Select all records from the table.

\*

\* @return List of all records in the table. An empty list if there are no records in the table.

\*/

public List<ArAdjustment> selectAll();

/\*\*

\* Select all records from the table. Order by the specified columns.

\*

\* @param orderBy Array of column names to order results.

\* @return List of all records in the table. An empty list if there are no records in the table.

\*/

public List<ArAdjustment> selectAllOrderBy(String[] orderBy);

/\*\*

\* Select records from the table. In the SQL, the where clause will use AND.

\*

\* @param columns Array of column names.

\* @param values Array of values. The number of values must equal the number of column names.

\* @return A list of records matching the selection criteria. An empty list if there are no records matching the selection criteria.

\*/

public List<ArAdjustment> selectAnd(String[] columns, Object[] values);

/\*\*

\* Select records from the table. In the SQL, the where clause will use AND. Order by the specified columns.

\*

\* @param columns Array of column names.

\* @param values Array of values. The number of values must equal the number of column names.

\* @param orderBy Array of column names to order results.

\* @return A list of records matching the selection criteria. An empty list if there are no records matching the selection criteria.

\*/

public List<ArAdjustment> selectAndOrderBy(String[] columns, Object[] values, String[] orderBy);

/\*\*

\* Select records from the table.

\*

\* @param column The column name from the table.

\* @param value The value for the column.

\* @return A list of records matching the selection criteria. An empty list if there are no records matching the selection criteria.

\*/

public List<ArAdjustment> selectByColumn(String column, Object value);

/\*\*

\* Select records from the table. Order by the specified columns.

\*

\* @param column The column name from the table.

\* @param value The value for the column.

\* @param orderBy Array of column names to order results.

\* @return A list of records matching the selection criteria. An empty list if there are no records matching the selection criteria.

\*/

public List<ArAdjustment> selectByColumnOrderBy(String column, Object value, String[] orderBy);

/\*\*

\* Select records from the table. In the SQL, the where clause will use OR.

\*

\* @param columns Array of column names.

\* @param values Array of values. The number of values must equal the number of column names.

\* @return A list of records matching the selection criteria. An empty list if there are no records matching the selection criteria.

\*/

public List<ArAdjustment> selectOr(String[] columns, Object[] values);

/\*\*

\* Select records from the table. In the SQL, the where clause will use OR. Order by the specified columns.

\*

\* @param columns Array of column names.

\* @param values Array of values. The number of values must equal the number of column names.

\* @param orderBy Array of column names to order results.

\* @return A list of records matching the selection criteria. An empty list if there are no records matching the selection criteria.

\*/

public List<ArAdjustment> selectOrOrderBy(String[] columns, Object[] values, String[] orderBy);

/\*\*

\* Delete all records from the table.

\*

\* @return The number of records deleted.

\*/

public int deleteAll();

/\*\*

\* Delete records from the table. In the SQL, the where clause will use AND.

\*

\* @param columns Array of column names.

\* @param values Array of values. The number of values must equal the number of column names.

\* @return The number of records deleted.

\*/

public int deleteAnd(String[] columns, Object[] values);

/\*\*

\* Delete records from the table.

\*

\* @param column The column name from the table.

\* @param value The value for the column.

\* @return The number of records deleted.

\*/

public int deleteByColumn(String column, Object value);

/\*\*

\* Delete records from the table. In the SQL, the where clause will use OR.

\*

\* @param columns Array of column names.

\* @param values Array of values. The number of values must equal the number of column names.

\* @return The number of records deleted.

\*/

public int deleteOr(String[] columns, Object[] values);

/\*\*

\* Count all records in the table.

\*

\* @return The number of records in the table.

\*/

public int countAll();

/\*\*

\* Count records in the table. In the SQL, the where clause will use AND.

\*

\* @param columns Array of column names.

\* @param values Array of values. The number of values must equal the number of column names.

\* @return The number of records matching the selection criteria.

\*/

public int countAnd(String[] columns, Object[] values);

/\*\*

\* Count records in the table. In the SQL, the where clause will use OR.

\*

\* @param columns Array of column names.

\* @param values Array of values. The number of values must equal the number of column names.

\* @return The number of records matching the selection criteria.

\*/

public int countOr(String[] columns, Object[] values);

//CUSTOM CODE: START

// BEGIN CACSG00010018

public Long getArAdjustmentIdMaxVal();

// END CACSG00010018

//CUSTOM CODE: END

}