ANNIS

Zusammenfassung

Inhaltsverzeichnis

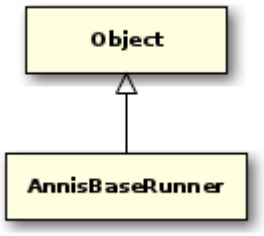
1 Paket annis 1
2 Paket annis.administration 7
3 Paket annis.dao 14
4 Paket annis.ql.parser 23
5 Paket annis.service.internal 32
6 Paket annis.sqlgen 34
7 Paket annis.sqlgen.annopool 68
8 Paket annis.sqlgen.fullfacts 69
9 Paket annis.utils 70
10 Veraltete APIs 72
11 Statistiken 72
Stichwortverzeichnis 75

1. Paket annis

1.1. Klasse AnnisBaseRunner

```
public abstract class AnnisBaseRunner {

  // Public Constructors
  public AnnisBaseRunner();

  // Public Static Methods
  public static String getAnnisHome();
  public static Object getBean(String beanName,
                               boolean logToConsole,
                               String[] contextLocations);
  public static AnnisBaseRunner getInstance(String beanName,
                                            boolean logToConsole,
                                            String[] contextLocations);
  public static AnnisBaseRunner getInstance(String beanName,
                                            String[] contextLocations);

  // Public Methods
  public String getHelloMessage();
  public PrintStream getOut();
  public String getPrompt();
  public void run(String[] args);
  public void setHelloMessage(String helloMessage);
  public void setOut(PrintStream out);
  public void setPrompt(String prompt);
}
```

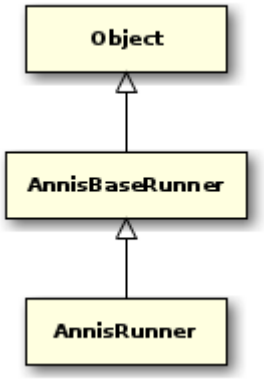
Direkt abgeleitete Klassen: `annis.AnnisRunner`, `annis.administration.AnnisAdminRunner`, `annis.service.internal.AnnisServiceRunner`



1.2. Klasse AnnisRunner

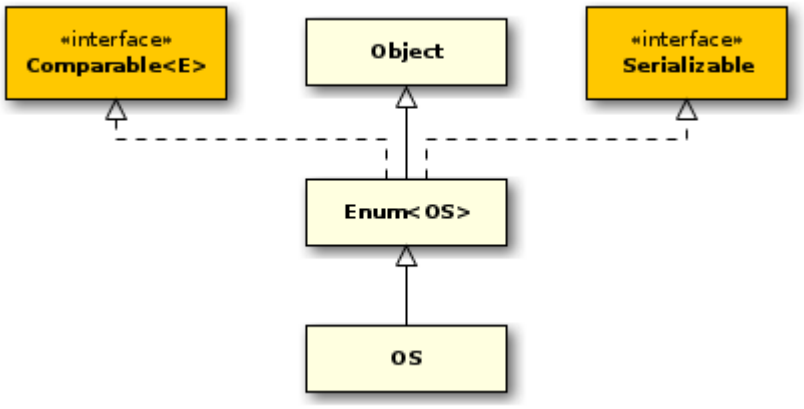
```
public class AnnisRunner extends AnnisBaseRunner {

  // Public Constructors
  public AnnisRunner();

  // Public Static Methods
  public static void main(String[] args);

  // Public Methods
  public String benchmarkOptions(QueryData queryData);
  public void doAnalyze(String functionCall);
  public void doAnnotate(String annisQuery);
  public void doAnnotations(String doListValues);
  public void doBenchmark(String benchmarkCount);
  public void doCorpus(String list);
  public void doCount(String annisQuery);
  public void doDebug(String ignore);
  public void doDoc(String docCall);
  public void doExplain(String functionCall, boolean analyze);
  public void doFind(String annisQuery);
  public void doList(String unused);
  public void doMatrix(String annisQuery);
  public void doMeta(String corpusId);
  public void doParse(String annisQuery);
  public void doPlan(String functionCall);
  public void doQuit(String dummy);
  public void doRecord(String dummy);
  public void doSet(String callToSet);
  public void doShow(String setting);
  public void doSql(String functionCall);
  public void doSqlDoc(String docCall);
  public void doSqlText(String textID);
  public void doText(String textID);
  public AnnisDao getAnnisDao();
  public AnnisParser getAnnisParser();
  public AnnotateSqlGenerator<SaltProject> getAnnotateSqlGenerator();
  public QueryAnalysis getAqlAnalysis();
  public int getContext();
  public List<Long> getCorpusList();
  public SqlGenerator<QueryData, Integer> getCountSqlGenerator();
  public SqlGenerator<QueryData, List<Match>> getFindSqlGenerator();
```

```
  public int getMatchLimit();
  public SqlGenerator<QueryData, List<AnnotatedMatch>> getMatrixSqlGenerator();
  public MetaDataFilter getMetaDataFilter();
  public QueryAnalysis getQueryAnalysis();
  public void setAnnisDao(AnnisDao annisDao);
  public void setAnnisParser(AnnisParser annisParser);
  public void setAnnotateSqlGenerator(AnnotateSqlGenerator<SaltProject> annotateSqlGenerator);
  public void setAqlAnalysis(QueryAnalysis aqlAnalysis);
  public void setContext(int context);
  public void setCorpusList(List<Long> corpusList);
  public void setCountSqlGenerator(SqlGenerator<QueryData,
Integer> countSqlGenerator);
  public void setFindSqlGenerator(SqlGenerator<QueryData,
List<Match>> findSqlGenerator);
  public void setMatchLimit(int matchLimit);
  public void setMatrixSqlGenerator(SqlGenerator<QueryData,
List<AnnotatedMatch>> matrixSqlGenerator);
  public void setMetaDataFilter(MetaDataFilter metaDataFilter);
  public void setQueryAnalysis(QueryAnalysis queryAnalysis);
}
```



1.3. Klasse AnnisRunner.OS

```
public static final class AnnisRunner.OS extends Enum<OS> {

  // Public Static Fields
  public static final OS linux;
  public static final OS other;

  // Public Static Methods
  public static OS valueOf(String name);
  public static OS[] values();
}
```



1.4. Ausnahme AnnisRunnerException

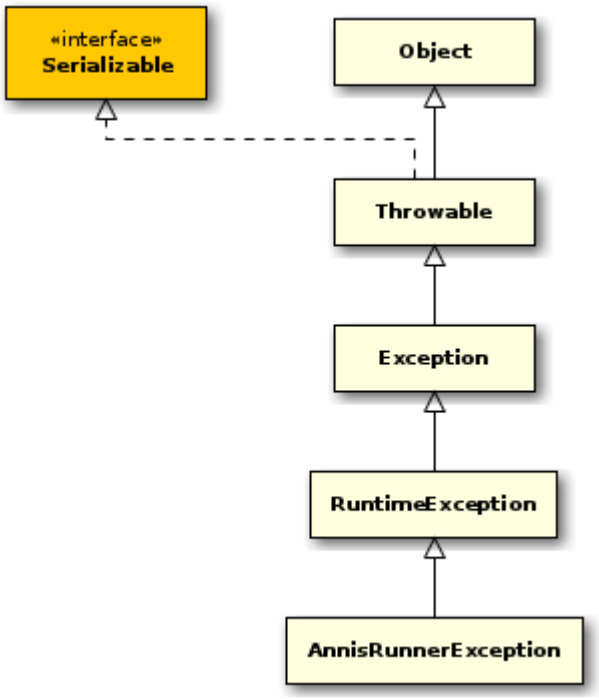
Base class for errors that occur during the execution of an AnnisRunner.

1.4.1. Syntax

```
public class AnnisRunnerException extends RuntimeException {

  // Public Constructors
  public AnnisRunnerException();
  public AnnisRunnerException(String message);
  public AnnisRunnerException(String message, Throwable cause);
  public AnnisRunnerException(Throwable cause);
}
```

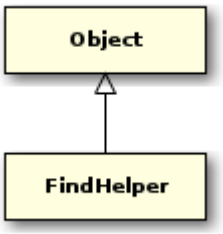
Direkt abgeleitete Klassen: `annis.UsageException`, `annis.administration.DatabaseAccessException`, `annis.administration.FileAccessException`

Autor



1.5. Klasse FindHelper

```
@Deprecated
public class FindHelper {

  // Public Constructors
  public FindHelper();
}
```



1.6. Klasse MemoryUsage

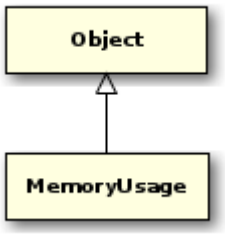
Utility class that prints out memory usage of the Java VM to a logfile.

1.6.1. Syntax

```
public class MemoryUsage {

  // Public Constructors
  public MemoryUsage();

  // Public Static Methods
  public static void logMemoryUsage();
```

```
}
```

Autor



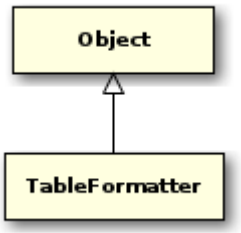
1.6.2. logMemoryUsage()

```
  public static void logMemoryUsage();
```

Print total, free and maximum memory of the VM to a logfile.

1.7. Klasse TableFormatter

```
public class TableFormatter {

  // Public Constructors
  public TableFormatter();

  // Public Methods
  public String formatAsTable(List<?> list, String[] fields);
}
```



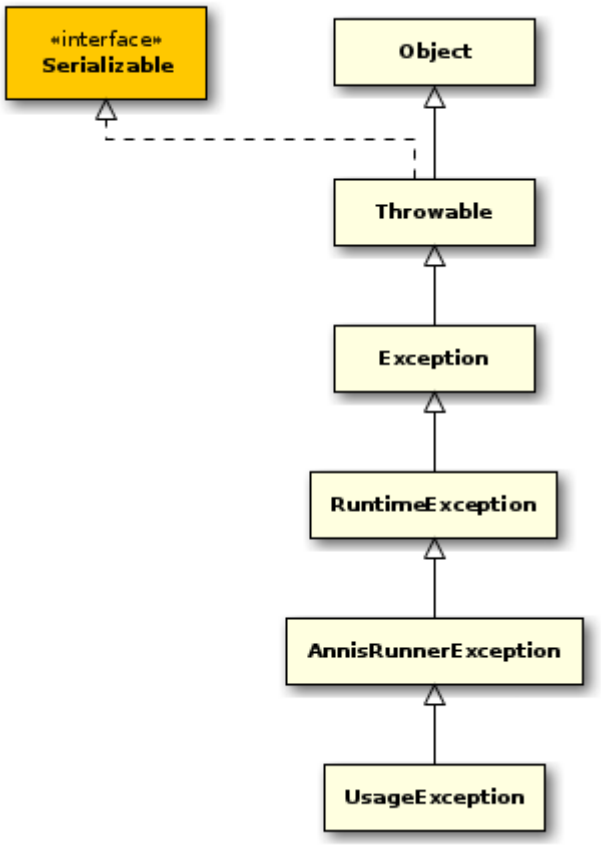
1.8. Ausnahme UsageException

Signifies an error by the user, usually a bad command.

1.8.1. Syntax

```
public class UsageException extends AnnisRunnerException {

  // Public Constructors
  public UsageException();
  public UsageException(String message);
  public UsageException(String message, Throwable cause);
  public UsageException(Throwable cause);
}
```

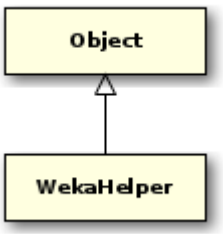
Autor



1.9. Klasse WekaHelper

```
public class WekaHelper {

  // Public Constructors
  public WekaHelper();

  // Public Methods
  public String exportAsArff(List<AnnotatedMatch> annotatedMatches);
}
```

Autor thomas



2. Paket annis.administration

2.1. Schnittstelle AdministrationDao

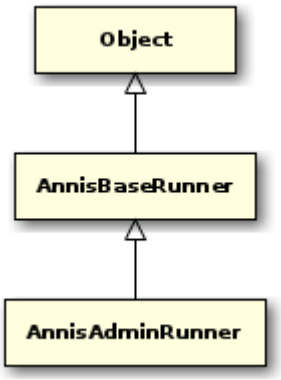
```
public interface AdministrationDao {
```

```
  // Public Methods
  public void createDatabase(String database);
  public void createSchema();
  public void createSchemaIndexes();
  public void createUser(String username, String password);
  public void deleteCorpora(List<Long> ids);
  public void dropDatabase(String database);
  public void dropUser(String username);
  public boolean executeSqlFromScript(String script);
  public boolean executeSqlFromScript(String script, MapSqlParameterSource args);
  public void importCorpus(String path);
  public List<Map<String, Object>> listCorpusStats();
  public List<Map<String, Object>> listTableStats();
  public List<Long> listToplevelCorpora();
  public List<String> listUnusedIndexes();
  public List<String> listUsedIndexes();
  public void populateSchema();
  public void setDataSource(DataSource dataSource);
  public void setupDatabase();
}
```

Autor thomas

2.2. Klasse AnnisAdminRunner

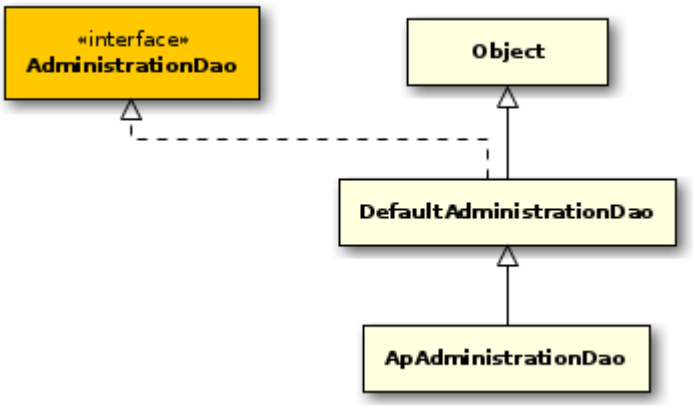
```
public class AnnisAdminRunner extends AnnisBaseRunner {

  // Public Constructors
  public AnnisAdminRunner();

  // Public Static Methods
  public static void main(String[] args);

  // Public Methods
  public CorpusAdministration getCorpusAdministration();
  public void run(String[] args);
  public void setCorpusAdministration(CorpusAdministration administration);
}
```



2.3. Klasse ApAdministrationDao

```
public class ApAdministrationDao extends DefaultAdministrationDao {

  // Public Constructors
  public ApAdministrationDao();

  // Public Methods
  public void populateSchema();
}
```

Autor thomas

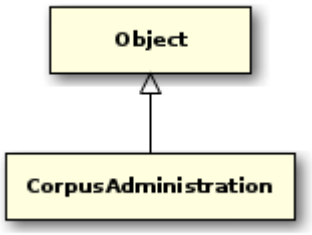


2.4. Klasse CorpusAdministration

```
public class CorpusAdministration {

  // Public Constructors
  public CorpusAdministration();

  // Public Methods
  public void deleteCorpora(List<Long> ids);
  public AdministrationDao getAdministrationDao();
  public void importCorpora(String[] paths);
  public void importCorpora(List<String> paths);
  public void initializeDatabase(String host,
                                 String port,
```

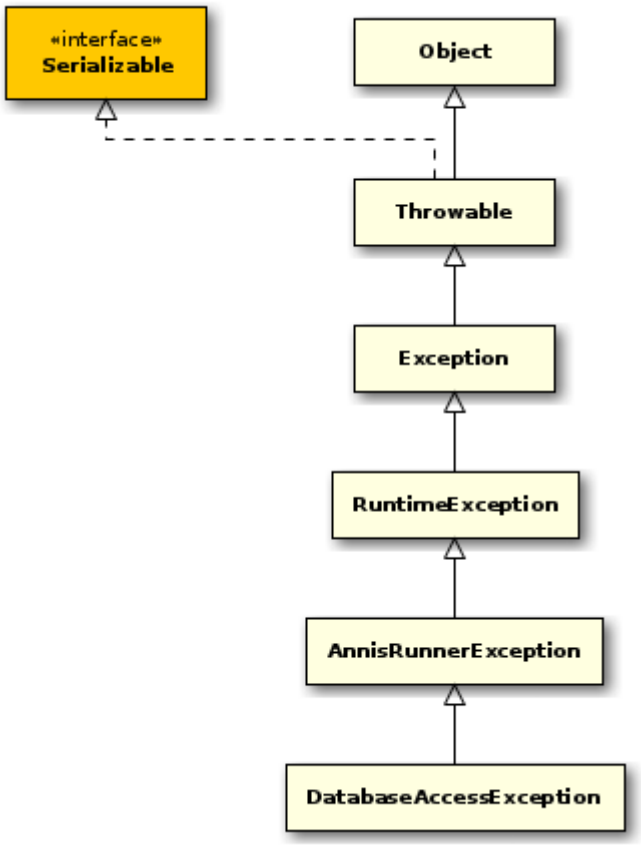
```
                                String database,
                                String user,
                                String password,
                                String defaultDatabase,
                                String superUser,
                                String superPassword);
  public List<Map<String, Object>> listCorpusStats();
  public List<Map<String, Object>> listTableStats();
  public List<String> listUnusedIndexes();
  public List<String> listUsedIndexes();
  public void setAdministrationDao(AdministrationDao administrationDao);
}
```

Autor thomas



2.5. Ausnahme DatabaseAccessException

```
public class DatabaseAccessException extends AnnisRunnerException {

  // Public Constructors
  public DatabaseAccessException();
  public DatabaseAccessException(String message);
  public DatabaseAccessException(String message, Throwable cause);
  public DatabaseAccessException(Throwable cause);
}
```



2.6. Klasse DefaultAdministrationDao

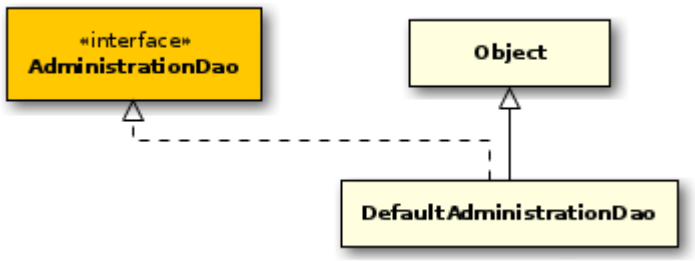
- Transaktionen - Datenbank-Zugriffsrechte für verschiedene Methoden - Reihenfolge der Aufrufe - Skripte in $ANNIS_HOME/scripts

2.6.1. Syntax

```
public class DefaultAdministrationDao implements AdministrationDao {

  // Public Constructors
  public DefaultAdministrationDao();

  // Public Methods
  public void createDatabase(String database);
  public void createSchema();
  public void createSchemaIndexes();
  public void createUser(String username, String password);
  public void deleteCorpora(List<Long> ids);
  public void dropDatabase(String database);
  public void dropUser(String username);
  public boolean executeSqlFromScript(String script);
  public boolean executeSqlFromScript(String script, MapSqlParameterSource args);
  public String getDbLayout();
  public String getExternalFilesPath();
  public NamedParameterJdbcTemplate getJdbcTemplate();
  public String getScriptPath();
```

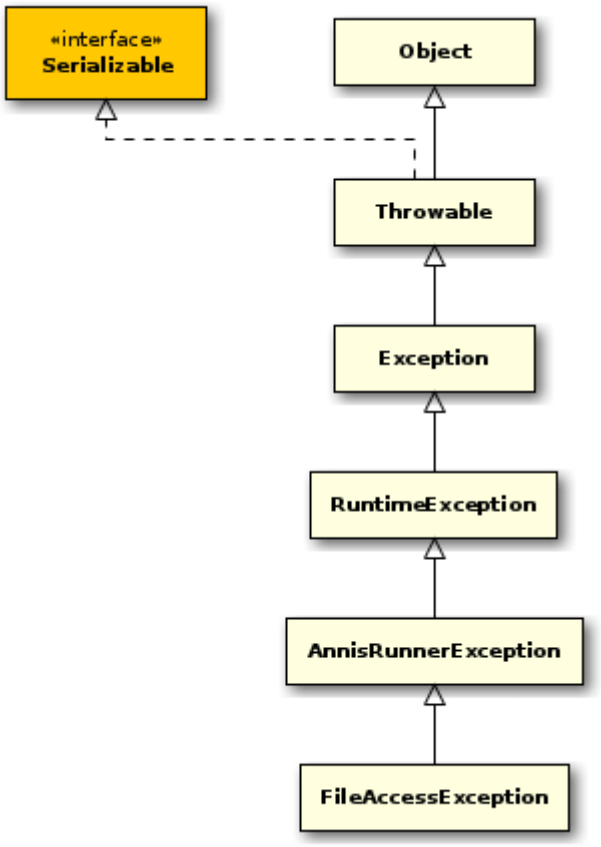
```
  public void importCorpus(String path);
  public boolean isTemporaryStagingArea();
  public List<Map<String, Object>> listCorpusStats();
  public List<String> listIndexDefinitions(boolean used, List<String> tables);
  public List<String> listIndexDefinitions(String[] tables);
  public List<Map<String, Object>> listTableStats();
  public List<Long> listToplevelCorpora();
  public List<String> listUnusedIndexes();
  public List<String> listUsedIndexes();
  public List<String> listUsedIndexes(String[] tables);
  public void populateSchema();
  public boolean resetStatistics();
  public void setDataSource(DataSource dataSource);
  public void setDbLayout(String dbLayout);
  public void setExternalFilesPath(String externalFilesPath);
  public void setScriptPath(String scriptPath);
  public void setTemporaryStagingArea(boolean temporaryStagingArea);
  public void setupDatabase();
}
```

Direkt abgeleitete Klassen: `annis.administration.ApAdministrationDao`, `annis.administration.SfAdministrationDao`



2.7. Ausnahme FileAccessException

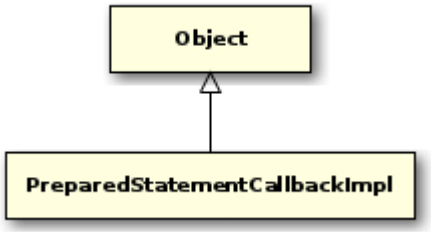
```
public class FileAccessException extends AnnisRunnerException {

  // Public Constructors
  public FileAccessException();
  public FileAccessException(String message);
  public FileAccessException(String message, Throwable cause);
  public FileAccessException(Throwable cause);
}
```



2.8. Klasse PreparedStatementCallbackImpl

```
public class PreparedStatementCallbackImpl {

  // Public Constructors
  public PreparedStatementCallbackImpl(String absolutePath, String corpusRef);

  // Public Methods
  public Boolean doInPreparedStatement(PreparedStatement ps)
    throws SQLException,
           DataAccessException;
}
```

Autor benjamin

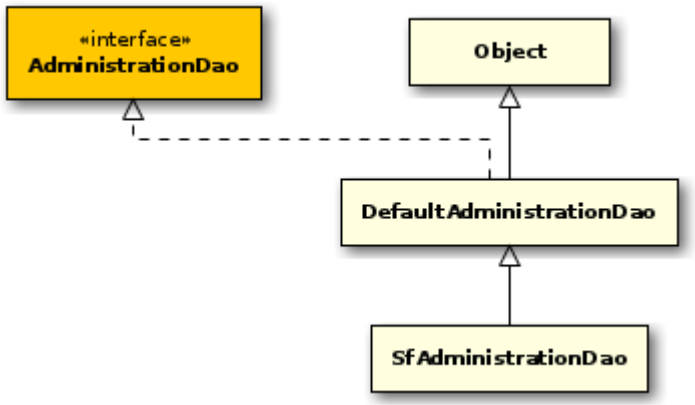


2.9. Klasse SfAdministrationDao

```
public class SfAdministrationDao extends DefaultAdministrationDao {
```

```
  // Public Constructors
  public SfAdministrationDao();

  // Public Methods
  public void populateSchema();
}
```

Autor thomas



3. Paket annis.dao

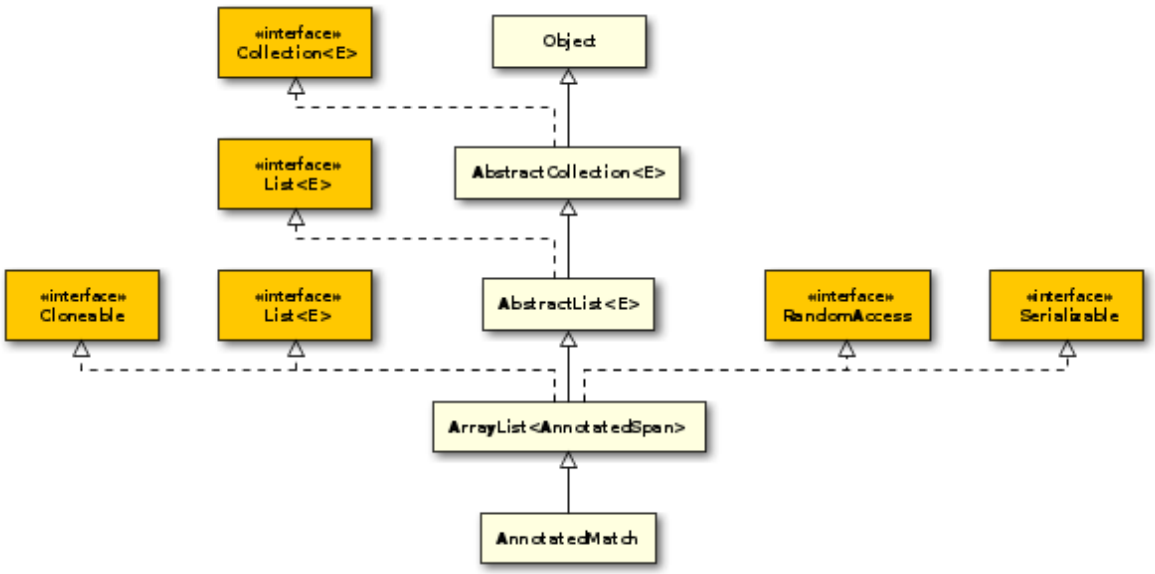
3.1. Schnittstelle AnnisDao

```
public interface AnnisDao {

  // Public Methods
  public SaltProject annotate(QueryData queryData);
  public int count(QueryData queryData);
  public T executeQueryFunction(QueryData queryData,
                                SqlGenerator<QueryData, T> generator);
  public T executeQueryFunction(QueryData queryData,
                                SqlGenerator<QueryData, T> generator,
                                <any> extractor);
  public String explain(SqlGenerator<QueryData, ?> generator,
                        QueryData queryData,
                        boolean analyze);
  public List<Match> find(QueryData queryData);
  public AnnisBinary getBinary(String corpusName, int offset, int length);
  public HashMap<Long, Properties> getCorpusConfiguration();
  public Map<String, String> getCorpusConfiguration(String corpusName);
  public List<ResolverEntry> getResolverEntries(SingleResolverRequest request);
  public int getTimeout();
  public List<AnnisAttribute> listAnnotations(List<Long> corpusList,
                                              boolean listValues,
                                              boolean onlyMostFrequentValues);
  public List<AnnisCorpus> listCorpora();
  public List<Annotation> listCorpusAnnotations(String toplevelCorpusName,
                                                String documentName);
  public List<Annotation> listCorpusAnnotations(long id);
  public List<Long> listCorpusByName(List<String> corpusNames);
```

```
  public List<String> mapCorpusIdsToNames(List<Long> ids);
  public List<AnnotatedMatch> matrix(QueryData queryData);
  public QueryData parseAQL(String aql, List<Long> corpusList);
  public SaltProject retrieveAnnotationGraph(String toplevelCorpusName,
                                             String documentName);
  public SaltProject retrieveAnnotationGraph(long textId);
  public void setCorpusConfiguration(HashMap<Long,
Properties> corpusConfiguration);
  public void setTimeout(int milliseconds);
}
```

3.2. Klasse AnnotatedMatch

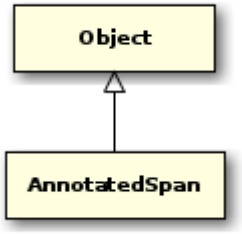
```
public class AnnotatedMatch extends ArrayList<AnnotatedSpan> {

  // Public Constructors
  public AnnotatedMatch();
  public AnnotatedMatch(AnnotatedSpan[] spans);
  public AnnotatedMatch(List<AnnotatedSpan> spans);
}
```



3.3. Klasse AnnotatedSpan

```
public class AnnotatedSpan {

  // Public Constructors
  public AnnotatedSpan(long id,
                       String coveredText,
                       List<Annotation> annotations);
  public AnnotatedSpan(long id,
                       String coveredText,
```

```
                        List<Annotation> annotations,
                        List<Annotation> metadata);

  // Public Methods
  public List<Annotation> getAnnotations();
  public String getCoveredText();
  public long getId();
  public List<Annotation> getMetadata();
  public void setAnnotations(List<Annotation> annotations);
  public void setCoveredText(String coveredText);
  public void setId(long id);
  public void setMetadata(List<Annotation> metadata);
}
```

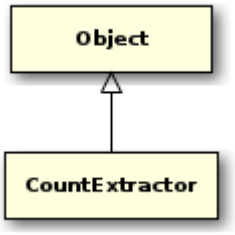


3.4. Schnittstelle CorpusSelectionStrategy

```
public interface CorpusSelectionStrategy {

  // Public Methods
  public void addMetaAnnotations(List<Annotation> annotations);
  public String createViewSql();
  public void registerNodeAdapter(NodeSqlAdapter adapter);
  public boolean usesViews();
  public String viewName(String table);
  public String whereClauseForNode(String docRefColumn);
}
```

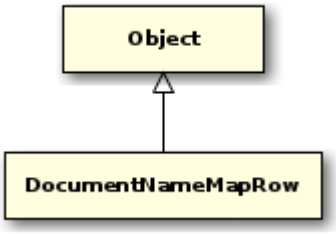
3.5. Klasse CountExtractor

```
public class CountExtractor {

  // Public Constructors
  public CountExtractor();

  // Public Methods
  public String explain(JdbcTemplate jdbcTemplate, boolean analyze);
  public String getMatchedNodesViewName();
  public int queryCount(JdbcTemplate jdbcTemplate);
  public void setMatchedNodesViewName(String matchedNodesViewName);
}
```



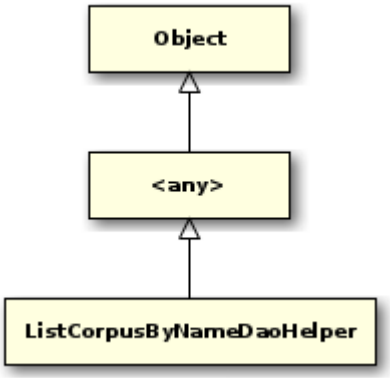
3.6. Klasse DocumentNameMapRow

```
public class DocumentNameMapRow {

  // Public Constructors
  public DocumentNameMapRow();

  // Public Methods
  public String mapRow(ResultSet rs, int rowNum)
    throws SQLException;
}
```



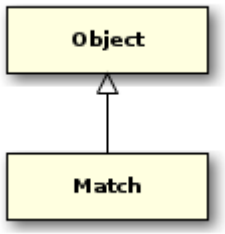
3.7. Klasse ListCorpusByNameDaoHelper

```
public class ListCorpusByNameDaoHelper extends <any> {

  // Public Constructors
  public ListCorpusByNameDaoHelper();

  // Public Methods
  public String createSql(List<String> corpusNames);
}
```



3.8. Klasse Match

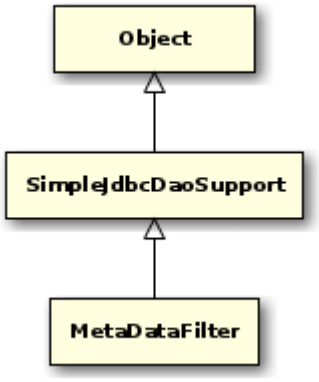
```
public class Match {

  // Public Constructors
  public Match();

  // Public Methods
  public String getSaltId(int i);
  public void setSaltId(String id);
}
```



3.9. Klasse MetaDataFilter

```
public class MetaDataFilter extends SimpleJdbcDaoSupport {

  // Public Constructors
  public MetaDataFilter();

  // Public Methods
  public List<Long> getDocumentsForMetadata(QueryData queryData);
  public SubQueryCorpusSelectionStrategy getSubQueryCorpusSelectionStrategy();
  public void setSubQueryCorpusSelectionStrategy(SubQueryCorpusSelectionStrategy subQueryCorpusSe
}
```

Autor thomas



3.9.1. getDocumentsForMetadata(QueryData)

```
  public List<Long> getDocumentsForMetadata(QueryData queryData);
```

Will query the database which documents are matching according to the given metadata

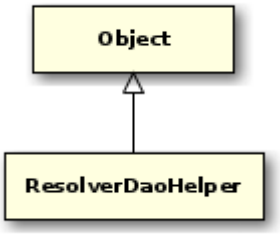
Parameter

queryData	QueryData from which the meta data will be extracted
return	The list of documents matching the meta data or null if no constraints need to be applied (all documents are matching)

3.10. Klasse ResolverDaoHelper

```
public class ResolverDaoHelper {

  // Public Constructors
  public ResolverDaoHelper();

  // Public Methods
  public PreparedStatement createPreparedStatement(Connection cnctn)
    throws SQLException;
  public List<ResolverEntry> extractData(ResultSet rs)
    throws SQLException,
           DataAccessException;
  public void fillPreparedStatement(SingleResolverRequest resolverRequest,
                                    PreparedStatement stmt)
    throws SQLException;
}
```

Autor thomas



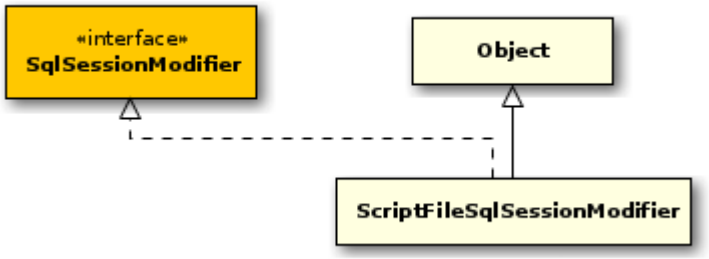
3.11. Schnittstelle ResultSetConverter

```
@Deprecated
public interface ResultSetConverter<T> {

  // Public Methods
  public T convertResultSet(ResultSet resultSet);
}
```

3.12. Klasse ScriptFileSqlSessionModifier

```
public class ScriptFileSqlSessionModifier implements SqlSessionModifier {

  // Public Constructors
  public ScriptFileSqlSessionModifier();

  // Public Methods
```

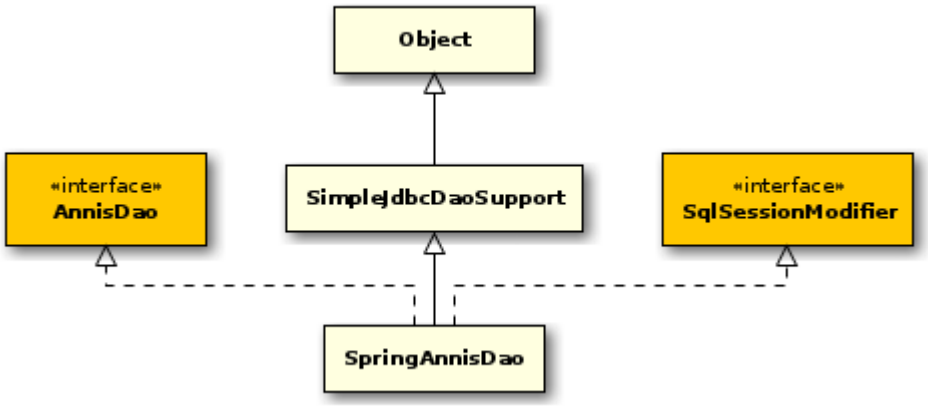
```
  public AdministrationDao getAdministrationDao();
  public String getScriptFile();
  public void modifySqlSession(JdbcTemplate jdbcTemplate, QueryData queryData);
  public void setAdministrationDao(AdministrationDao administrationDao);
  public void setScriptFile(String scriptFile);
}
```



3.13. Klasse SpringAnnisDao

```
public class SpringAnnisDao extends SimpleJdbcDaoSupport
   implements AnnisDao,
              SqlSessionModifier {

  // Public Constructors
  public SpringAnnisDao();

  // Public Methods
  public SaltProject annotate(QueryData queryData);
  public int count(QueryData queryData);
  public T executeQueryFunction(QueryData queryData,
                                SqlGenerator<QueryData, T> generator);
  public T executeQueryFunction(QueryData queryData,
                                SqlGenerator<QueryData, T> generator,
                                <any> extractor);
  public String explain(SqlGenerator<QueryData, ?> generator,
                        QueryData queryData,
                        boolean analyze);
  public List<Match> find(QueryData queryData);
  public AnnotateSqlGenerator<SaltProject> getAnnotateSqlGenerator();
  public AnnisParser getAqlParser();
  public AnnisBinary getBinary(String corpusName, int offset, int length);
  public ByteHelper getByteHelper();
  public HashMap<Long, Properties> getCorpusConfiguration();
  public Map<String, String> getCorpusConfiguration(String corpusName);
  public CountExtractor getCountExtractor();
  public CountSqlGenerator getCountSqlGenerator();
  public FindSqlGenerator getFindSqlGenerator();
  public AnnotateSqlGenerator getGraphExtractor();
  public ListAnnotationsSqlHelper getListAnnotationsSqlHelper();
  public ListCorpusAnnotationsSqlHelper getListCorpusAnnotationsSqlHelper();
  public ListCorpusByNameDaoHelper getListCorpusByNameDaoHelper();
  public ListCorpusSqlHelper getListCorpusSqlHelper();
  public MatrixSqlGenerator getMatrixSqlGenerator();
  public MetaDataFilter getMetaDataFilter();
```

```
  public <any> getPlanRowMapper();
  public QueryAnalysis getQueryAnalysis();
  public List<ResolverEntry> getResolverEntries(SingleResolverRequest request);
  public SaltAnnotateExtractor getSaltAnnotateExtractor();
  public SqlGenerator getSqlGenerator();
  public List<SqlSessionModifier> getSqlSessionModifiers();
  public int getTimeout();
  public void init();
  public List<AnnisAttribute> listAnnotations(List<Long> corpusList,
                                              boolean listValues,
                                              boolean onlyMostFrequentValues);
  public List<AnnisCorpus> listCorpora();
  public List<Annotation> listCorpusAnnotations(String toplevelCorpusName,
                                                String documentName);
  public List<Annotation> listCorpusAnnotations(long corpusId);
  public List<Long> listCorpusByName(List<String> corpusNames);
  public List<String> mapCorpusIdsToNames(List<Long> ids);
  public List<AnnotatedMatch> matrix(QueryData queryData);
  public void modifySqlSession(JdbcTemplate jdbcTemplate, QueryData queryData);
  public QueryData parseAQL(String aql, List<Long> corpusList);
  public SaltProject retrieveAnnotationGraph(String toplevelCorpusName,
                                             String documentName);
  public SaltProject retrieveAnnotationGraph(long textId);
  public void setAnnotateSqlGenerator(AnnotateSqlGenerator<SaltProject> annotateSqlGenerator);
  public void setAqlParser(AnnisParser aqlParser);
  public void setByteHelper(ByteHelper byteHelper);
  public void setCorpusConfiguration(HashMap<Long,
Properties> corpusConfiguration);
  public void setCountExtractor(CountExtractor countExtractor);
  public void setCountSqlGenerator(CountSqlGenerator countSqlGenerator);
  public void setFindSqlGenerator(FindSqlGenerator findSqlGenerator);
  public void setGraphExtractor(AnnotateSqlGenerator graphExtractor);
  public void setListAnnotationsSqlHelper(ListAnnotationsSqlHelper listNodeAnnotationsSqlHelper);
  public void setListCorpusAnnotationsSqlHelper(ListCorpusAnnotationsSqlHelper listCorpusAnnotati
  public void setListCorpusByNameDaoHelper(ListCorpusByNameDaoHelper listCorpusByNameDaoHelper);
  public void setListCorpusSqlHelper(ListCorpusSqlHelper listCorpusHelper);
  public void setMatrixSqlGenerator(MatrixSqlGenerator matrixSqlGenerator);
  public void setMetaDataFilter(MetaDataFilter metaDataFilter);
  public void setPlanRowMapper(<any> planRowMapper);
  public void setQueryAnalysis(QueryAnalysis queryAnalysis);
  public void setSaltAnnotateExtractor(SaltAnnotateExtractor saltAnnotateExtractor);
  public void setSqlGenerator(SqlGenerator sqlGenerator);
  public void setSqlSessionModifiers(List<SqlSessionModifier> sqlSessionModifiers);
  public void setTimeout(int timeout);
}
```

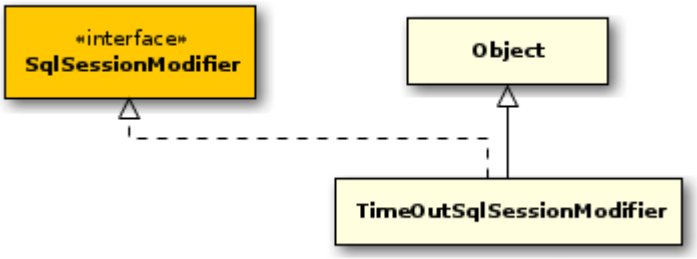


3.14. Schnittstelle SqlSessionModifier

```
public interface SqlSessionModifier {

  // Public Methods
  public void modifySqlSession(JdbcTemplate simpleJdbcTemplate,
                               QueryData queryData);
}
```

3.15. Klasse TimeOutSqlSessionModifier

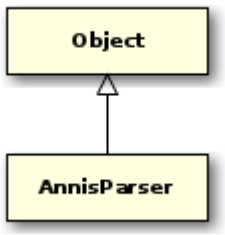
```
public class TimeOutSqlSessionModifier implements SqlSessionModifier {

  // Public Constructors
  public TimeOutSqlSessionModifier();

  // Public Methods
  public int getTimeout();
  public void modifySqlSession(JdbcTemplate jdbcTemplate, QueryData queryData);
  public void setTimeout(int timeout);
}
```



4. Paket annis.ql.parser

4.1. Klasse AnnisParser

```
public class AnnisParser {

  // Public Constructors
  public AnnisParser();

  // Public Static Methods
  public static String dumpTree(Start start);

  // Public Methods
  public String dumpTree(String annisQuery);
  public List<DepthFirstAdapter> getPostProcessors();
  public Start parse(String annisQuery);
  public void setPostProcessors(List<DepthFirstAdapter> postProcessors);
}
```



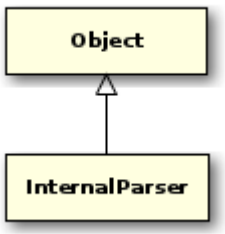
4.1.1. AnnisParser()

```
  public AnnisParser();
```

Creates a parser for AnnisQL statements.

4.2. Klasse AnnisParser.InternalParser

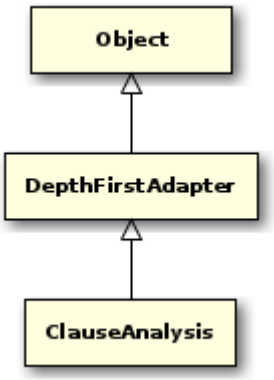
```
public static class AnnisParser.InternalParser {

  // Public Constructors
  public AnnisParser.InternalParser();

  // Public Methods
  public Start parse(String input)
    throws ParserException,
           LexerException,
           IOException;
}
```



4.3. Klasse ClauseAnalysis

```
public class ClauseAnalysis extends DepthFirstAdapter {

  // Public Constructors
  public ClauseAnalysis();
  public ClauseAnalysis(int aliasCount,
                        List<QueryAnnotation> metaAnnotations,
                        Map<String, QueryNode> nodes,
                        int precedenceBound);

  // Public Methods
  public void caseAAndExpr(AAndExpr node);
  public void caseAAnnotationSearchExpr(AAnnotationSearchExpr node);
  public void caseAAnyNodeSearchExpr(AAnyNodeSearchExpr node);
  public void caseAArityLingOp(AArityLingOp node);
  public void caseADirectDominanceSpec(ADirectDominanceSpec node);
  public void caseADirectPointingRelationSpec(ADirectPointingRelationSpec node);
  public void caseADirectPrecedenceSpec(ADirectPrecedenceSpec node);
  public void caseADirectSiblingSpec(ADirectSiblingSpec node);
  public void caseADocumentConstraintExpr(ADocumentConstraintExpr node);
  public void caseAExactOverlapLingOp(AExactOverlapLingOp node);
  public void caseAGroupedExpr(AGroupedExpr node);
  public void caseAIdentityLingOp(AIdentityLingOp node);
  public void caseAImplicitAndExpr(AImplicitAndExpr node);
  public void caseAInclusionLingOp(AInclusionLingOp node);
  public void caseAIndirectDominanceSpec(AIndirectDominanceSpec node);
  public void caseAIndirectPointingRelationSpec(AIndirectPointingRelationSpec node);
  public void caseAIndirectPrecedenceSpec(AIndirectPrecedenceSpec node);
  public void caseAIndirectSiblingSpec(AIndirectSiblingSpec node);
  public void caseALeftAlignLingOp(ALeftAlignLingOp node);
  public void caseALeftOverlapLingOp(ALeftOverlapLingOp node);
  public void caseAMetaConstraintExpr(AMetaConstraintExpr node);
  public void caseAOrExpr(AOrExpr node);
  public void caseAOverlapLingOp(AOverlapLingOp node);
  public void caseARangeDominanceSpec(ARangeDominanceSpec node);
  public void caseARangePointingRelationSpec(ARangePointingRelationSpec node);
  public void caseARangePrecedenceSpec(ARangePrecedenceSpec node);
  public void caseARightAlignLingOp(ARightAlignLingOp node);
  public void caseARightOverlapLingOp(ARightOverlapLingOp node);
  public void caseARootLingOp(ARootLingOp node);
  public void caseASameAnnotationGroupLingOp(ASameAnnotationGroupLingOp node);
  public void caseATextSearchExpr(ATextSearchExpr node);
  public void caseATextSearchNotEqualExpr(ATextSearchNotEqualExpr node);
  public void caseATokenArityLingOp(ATokenArityLingOp node);
  public List<QueryAnnotation> getMetaAnnotations();
  public Collection<QueryNode> getNodes();
  public int getPrecedenceBound();
  public int nodesCount();
  public void setPrecedenceBound(int precedenceBound);
}
```

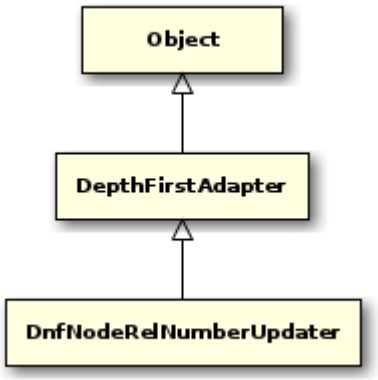
Autor thomas



4.4. Klasse DnfNodeRelNumberUpdater

```
public class DnfNodeRelNumberUpdater extends DepthFirstAdapter {

  // Public Constructors
  public DnfNodeRelNumberUpdater(SearchExpressionCounter origCounter);

  // Public Methods
  public void caseAAndExpr(AAndExpr node);
  public void caseAAnnotationSearchExpr(AAnnotationSearchExpr node);
  public void caseAAnyNodeSearchExpr(AAnyNodeSearchExpr node);
  public void caseALinguisticConstraintExpr(ALinguisticConstraintExpr node);
  public void caseATextSearchExpr(ATextSearchExpr node);
  public void caseATextSearchNotEqualExpr(ATextSearchNotEqualExpr node);
}
```

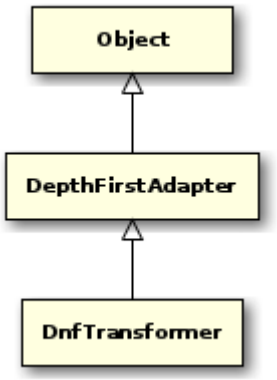
Autor thomas



4.5. Klasse DnfTransformer

```
public class DnfTransformer extends DepthFirstAdapter {

  // Public Constructors
  public DnfTransformer();

  // Public Methods
  public void caseStart(Start node);
```

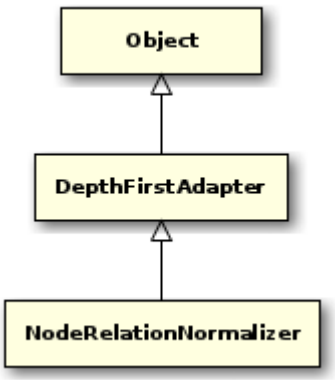
```
  public PExpr clone(PExpr node);
  public List<AAndExpr> distribute(List<AOrExpr> ors, List<AAndExpr> ands);
  public SearchExpressionCounter getCounter();
  public int getPosition(PExpr expr);
  public List<PExpr> listClauses(Start statement)
    throws UnknownExpressionException;
  public PExpr normalize(AAndExpr node);
  public PExpr normalize(AOrExpr node);
  public PExpr normalize(PExpr expr);
  public void setCounter(SearchExpressionCounter counter);
}
```



4.6. Klasse NodeRelationNormalizer

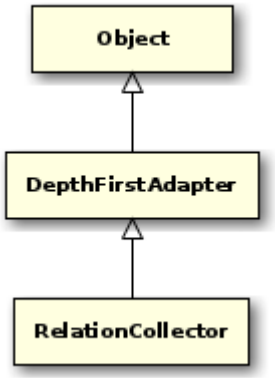
```
public class NodeRelationNormalizer extends DepthFirstAdapter {

  // Public Constructors
  public NodeRelationNormalizer();

  // Public Methods
  public void caseAAndExpr(AAndExpr node);
  public void split(AAndExpr expr,
                    ALinguisticConstraintExpr relation,
                    SearchExpressionCounter counter,
                    boolean replaceLeft);
}
```

Autor thomas



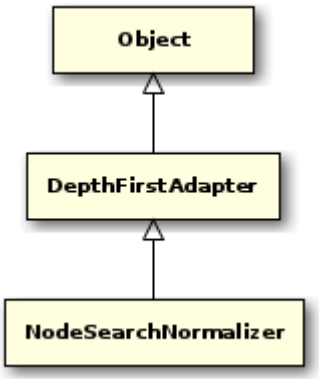
4.7. Klasse NodeRelationNormalizer.RelationCollector

```
public static class NodeRelationNormalizer.RelationCollector extends DepthFirstAd-
apter {

  // Public Constructors
  public NodeRelationNormalizer.RelationCollector();

  // Public Methods
  public void caseALinguisticConstraintExpr(ALinguisticConstraintExpr node);
  public Map<String, List<ALinguisticConstraintExpr>> getIsIn();
}
```



4.8. Klasse NodeSearchNormalizer

```
public class NodeSearchNormalizer extends DepthFirstAdapter {

  // Public Constructors
  public NodeSearchNormalizer();

  // Public Methods
  public void caseAAnnotationSearchExpr(AAnnotationSearchExpr node);
}
```

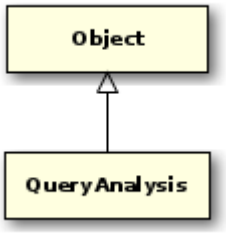


4.9. Klasse QueryAnalysis

```
public class QueryAnalysis {
```

```
  // Public Constructors
  public QueryAnalysis();

  // Public Methods
  public QueryData analyzeQuery(Start statement, List<Long> corpusList);
  public ClauseAnalysis getClauseAnalysis();
  public DnfTransformer getDnfTransformer();
  public NodeRelationNormalizer getNodeRelationNormalizer();
  public boolean isNormalizeNodesInEdgeRelations();
  public void setClauseAnalysis(ClauseAnalysis clauseAnalysis);
  public void setDnfTransformer(DnfTransformer dnfTransformer);
  public void setNodeRelationNormalizer(NodeRelationNormalizer nodeRelationNormalizer);
  public void setNormalizeNodesInEdgeRelations(boolean normalizeNodesInEdgeRelations);
}
```

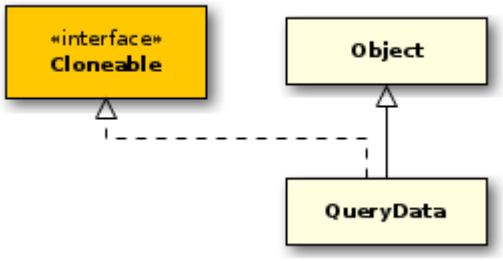
Autor thomas



4.10. Klasse QueryData

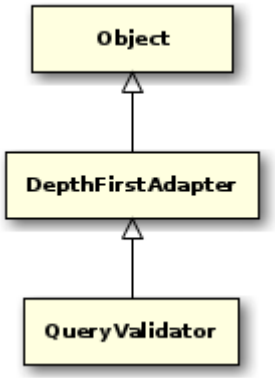
```
public class QueryData implements Cloneable {

  // Public Constructors
  public QueryData();

  // Public Methods
  public boolean addAlternative(List<QueryNode> nodes);
  public boolean addExtension(Object extension);
  public boolean addMetaAnnotations(List<QueryAnnotation> annotations);
  public QueryData clone();
  public List<List<QueryNode>> getAlternatives();
  public HashMap<Long, Properties> getCorpusConfiguration();
  public List<Long> getCorpusList();
  public List<Long> getDocuments();
  public Set<Object> getExtensions();
  public List<T> getExtensions(Class<T> clazz);
  public int getMaxWidth();
  public List<QueryAnnotation> getMetaData();
  public void setAlternatives(List<List<QueryNode>> alternatives);
  public void setCorpusConfiguration(HashMap<Long,
Properties> corpusConfiguration);
  public void setCorpusList(List<Long> corpusList);
  public void setDocuments(List<Long> documents);
  public void setMaxWidth(int maxWidth);
  public void setMetaData(List<QueryAnnotation> metaData);
```

```
  public String toString();
}
```



4.11. Klasse QueryValidator

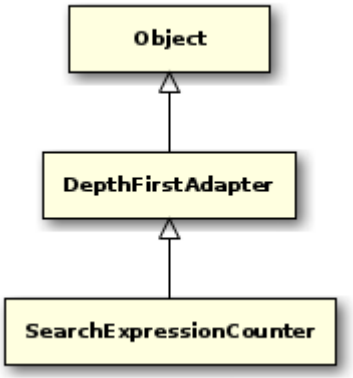
```
public class QueryValidator extends DepthFirstAdapter {

  // Public Constructors
  public QueryValidator();

  // Public Methods
  public void caseAAndExpr(AAndExpr node);
  public void caseALinguisticConstraintExpr(ALinguisticConstraintExpr node);
  public void caseStart(Start node);
}
```



4.12. Klasse SearchExpressionCounter

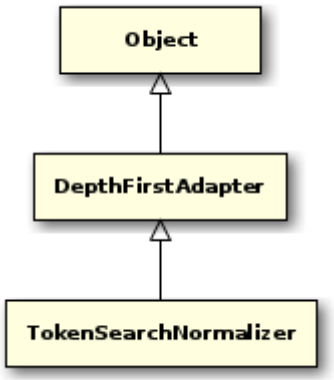
```
public class SearchExpressionCounter extends DepthFirstAdapter {

  // Public Constructors
  public SearchExpressionCounter();

  // Public Methods
  public void caseAAnnotationSearchExpr(AAnnotationSearchExpr node);
  public void caseAAnyNodeSearchExpr(AAnyNodeSearchExpr node);
  public void caseATextSearchExpr(ATextSearchExpr node);
  public void caseATextSearchNotEqualExpr(ATextSearchNotEqualExpr node);
  public int getCount();
  public int getPosition(Node expr);
  public Node getSearchExpression(int i);
```

```
  public void mapSearchExpressionClone(Node clone, Node original);
  public void setSearchPosition(Node node, int pos);
}
```



4.13. Klasse TokenSearchNormalizer

```
public class TokenSearchNormalizer extends DepthFirstAdapter {

  // Public Constructors
  public TokenSearchNormalizer();

  // Public Methods
  public void caseAAnnotationSearchExpr(AAnnotationSearchExpr node);
}
```



4.14. Klasse TreeDumper

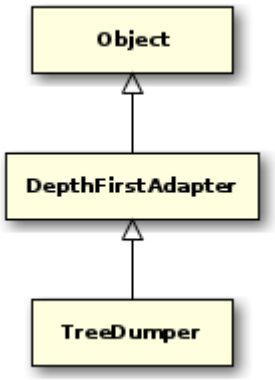
Dump abstract syntax trees on a PrintWriter. Adapted from TreeDumper.java by Nat Pryce: http://nat.truemesh.com/archives/000531.html

4.14.1. Syntax

```
public class TreeDumper extends DepthFirstAdapter {

  // Public Constructors
  public TreeDumper(PrintWriter out);
```

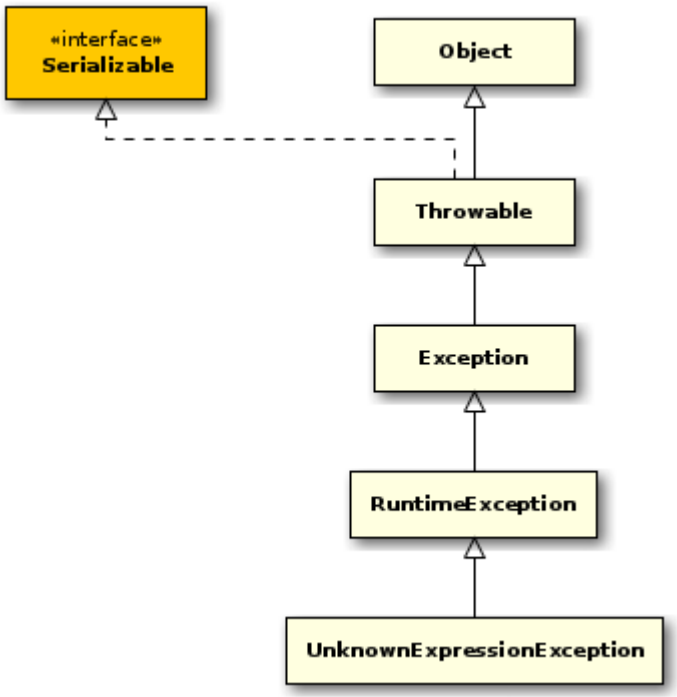
```
  // Public Methods
  public void defaultCase(Node node);
  public void defaultIn(Node node);
  public void defaultOut(Node node);
}
```

Autor



4.15. Ausnahme UnknownExpressionException

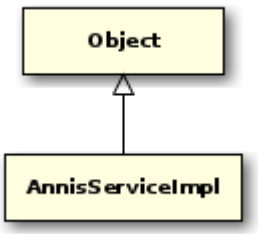
```
public class UnknownExpressionException extends RuntimeException {

  // Public Constructors
  public UnknownExpressionException();
  public UnknownExpressionException(String message);
  public UnknownExpressionException(String message, Throwable cause);
  public UnknownExpressionException(Throwable cause);
}
```



5. Paket annis.service.internal

5.1. Klasse AnnisServiceImpl

```
public class AnnisServiceImpl {

  // Public Constructors
  public AnnisServiceImpl();

  // Public Methods
  public AnnisDao getAnnisDao();
  public AnnisResult getAnnisResult(Long textId)
    throws RemoteException;
  public AnnisBinary getBinary(String corpusName, int offset, int length)
    throws RemoteException;
  public AnnisBinaryMetaData getBinaryMeta(String corpusName);
  public int getMaxContext();
  public List<Annotation> getMetadata(String toplevelCorpusName,
                                      String documentName)
    throws RemoteException;
  public List<Annotation> getMetadata(long corpusId)
    throws RemoteException,
           AnnisServiceException;
  public String getPaula(Long textId)
    throws RemoteException;
  public String getWeka(List<Long> corpusList, String annisQL)
    throws RemoteException,
           AnnisQLSemanticsException,
           AnnisQLSyntaxException,
           AnnisCorpusAccessException;
  public WekaHelper getWekaHelper();
  public boolean isValidQuery(String annisQuery)
    throws RemoteException,
           AnnisQLSemanticsException,
           AnnisQLSyntaxException;
  public void ping()
    throws RemoteException;
  public void sayHello();
  public void setAnnisDao(AnnisDao annisDao);
  public void setMaxContext(int maxContext);
  public void setWekaHelper(WekaHelper wekaHelper);
}
```



5.1.1. sayHello()

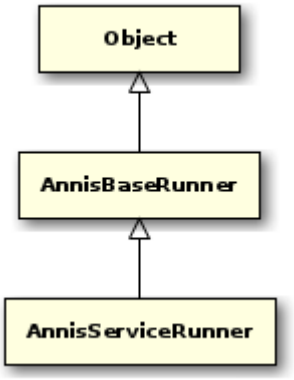
```
  public void sayHello();
```

Log the successful initialization of this bean.

XXX: This should be a private method annotated with `@PostConstruct`, but that doesn't seem to work. As a work-around, the method is called by Spring as an init-method.

5.2. Klasse AnnisServiceRunner

```
public class AnnisServiceRunner extends AnnisBaseRunner {

  // Public Constructors
  public AnnisServiceRunner();

  // Public Static Methods
  public static void main(String[] args)
    throws IOException;
  public static void shutdown();

  // Public Methods
  public void createWebServer();
}
```



5.2.1. shutdown()

```
  public static void shutdown();
```

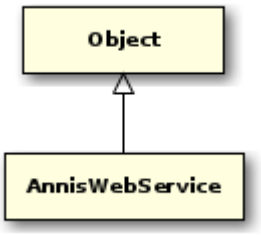
shutdown the AnnisService - ensure that current work load finishes

5.3. Klasse AnnisWebService

```
public class AnnisWebService {

  // Public Constructors
  public AnnisWebService();

  // Public Methods
  public SaltProject annotate(String query,
                              String rawCorpusNames,
                              String offsetRaw,
                              String limitRaw,
                              String leftRaw,
                              String rightRaw,
                              String segmentationLayer)
    throws IOException;
  public List<AnnisAttribute> annotations(String toplevelCorpus,
```

```
                                        String fetchValues,
                                        String onlyMostFrequentValues);
  public List<AnnisCorpus> corpora();
  public CorpusConfig corpusconfig(String toplevelName);
  public Response count(String query, String rawCorpusNames);
  public List<Match> find(String query,
                          String rawCorpusNames,
                          String offsetRaw,
                          String limitRaw)
    throws IOException;
  public AnnisDao getAnnisDao();
  public int getMaxContext();
  public int getPort();
  public WekaHelper getWekaHelper();
  public SaltProject graph(String toplevelCorpusName, String documentName);
  public List<ResolverEntry> resolver(String corpusName,
                                      String namespace,
                                      String type);
  public void sayHello();
  public void setAnnisDao(AnnisDao annisDao);
  public void setMaxContext(int maxContext);
  public void setPort(int port);
  public void setWekaHelper(WekaHelper wekaHelper);
}
```

Autor thomas



5.3.1. sayHello()

```
  public void sayHello();
```

Log the successful initialization of this bean.

XXX: This should be a private method annotated with `@PostConstruct`, but that doesn't seem to work. As a work-around, the method is called by Spring as an init-method.

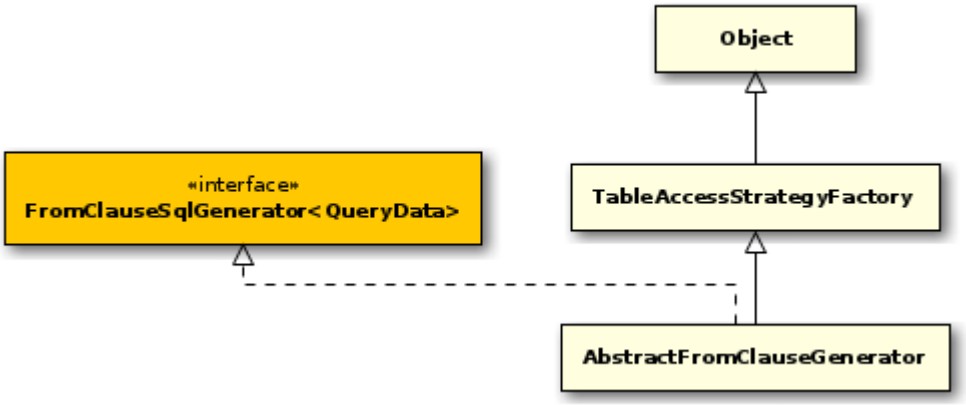
6. Paket annis.sqlgen

6.1. Klasse AbstractFromClauseGenerator

```
public abstract class AbstractFromClauseGenerator extends TableAccessStrategyFacto-
ry
   implements FromClauseSqlGenerator<QueryData> {

  // Public Constructors
  public AbstractFromClauseGenerator();
```

```
}
```

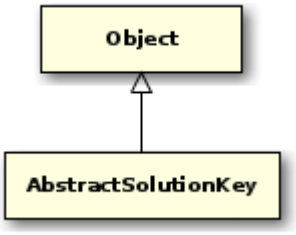
Direkt abgeleitete Klassen: `annis.sqlgen.CorpusPathWhereClauseGenerator`, `annis.sqlgen.TableJoinsInFromClauseSqlGenerator`, `annis.sqlgen.TableJoinsInWhereClauseGenerator`



6.2. Klasse AbstractSolutionKey

```
public class AbstractSolutionKey<BaseType> {

  // Public Constructors
  public AbstractSolutionKey();

  // Public Methods
  public List<String> generateInnerQueryColumns(TableAccessStrategy tableAccessStrategy,
                                                 int index);
  public String getCurrentKeyAsString();
  public String getIdColumnName();
  public int getKeySize();
  public Integer getMatchedNodeIndex(Object id);
  public Object getNodeId(ResultSet resultSet,
                          TableAccessStrategy tableAccessStrategy);
  public boolean isNewKey();
  public void setIdColumnName(String idColumnName);
}
```

Direkt abgeleitete Klassen: `annis.sqlgen.MultipleColumnsSolutionKey`, `annis.sqlgen.PostgreSqlArraySolutionKey`



6.3. Klasse AbstractSqlGenerator

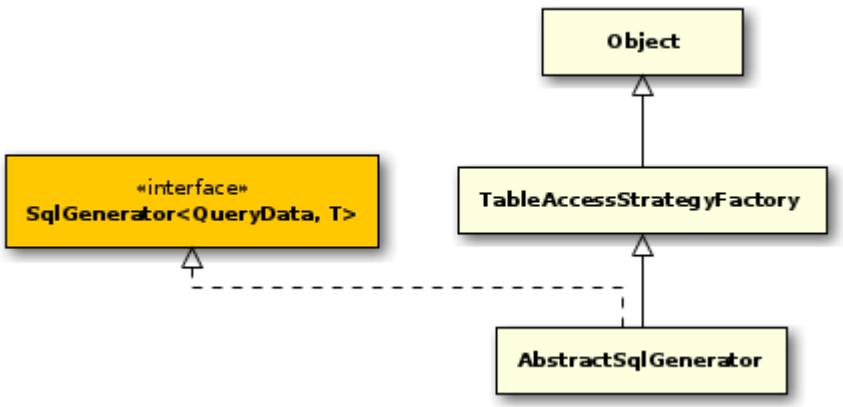
Abstract base class for a complete SQL statement. A SQL statement consists of a mandatory SELECT and FROM clauses and optional WHERE, GROUP BY, ORDER BY and LIMIT/OFFSET clauses. The individual clauses are generated using helper classes which are specified by properties.

6.3.1. Syntax

```
public abstract class AbstractSqlGenerator<T> extends TableAccessStrategyFactory
   implements SqlGenerator<QueryData, T> {

  // Public Static Fields
  public static final String TABSTOP = " ";

  // Public Constructors
  public AbstractSqlGenerator();

  // Public Methods
  public List<FromClauseSqlGenerator<QueryData>> getFromClauseSqlGenerators();
  public GroupByClauseSqlGenerator<QueryData> getGroupByClauseSqlGenerator();
  public LimitOffsetClauseSqlGenerator<QueryData> getLimitOffsetClauseSqlGenerator();
  public OrderByClauseSqlGenerator<QueryData> getOrderByClauseSqlGenerator();
  public SelectClauseSqlGenerator<QueryData> getSelectClauseSqlGenerator();
  public List<WhereClauseSqlGenerator<QueryData>> getWhereClauseSqlGenerators();
  public WithClauseSqlGenerator getWithClauseSqlGenerator();
  public void setFromClauseSqlGenerators(List<FromClauseSqlGenerator<QueryData>> fromClauseSqlGen
  public void setGroupByClauseSqlGenerator(GroupByClauseSqlGenerator<QueryData> groupByClauseSqlG
  public void setLimitOffsetClauseSqlGenerator(LimitOffsetClauseSqlGenerator<QueryData> limitOffs
  public void setOrderByClauseSqlGenerator(OrderByClauseSqlGenerator<QueryData> orderByClauseSqlG
  public void setSelectClauseSqlGenerator(SelectClauseSqlGenerator<QueryData> selectClauseSqlGene
  public void setWhereClauseSqlGenerators(List<WhereClauseSqlGenerator<QueryData>> whereClauseSql
  public void setWithClauseSqlGenerator(WithClauseSqlGenerator withClauseSqlGenerator);
  public String toSql(QueryData queryData);
  public String toSql(QueryData queryData, String indent);
}
```

Direkt abgeleitete Klassen: `annis.sqlgen.AbstractUnionSqlGenerator`, `annis.sqlgen.AnnotateSqlGenerator`, `annis.sqlgen.CountSqlGenerator`, `annis.sqlgen.MatrixSqlGenerator`

Autor



6.4. Klasse AbstractUnionSqlGenerator

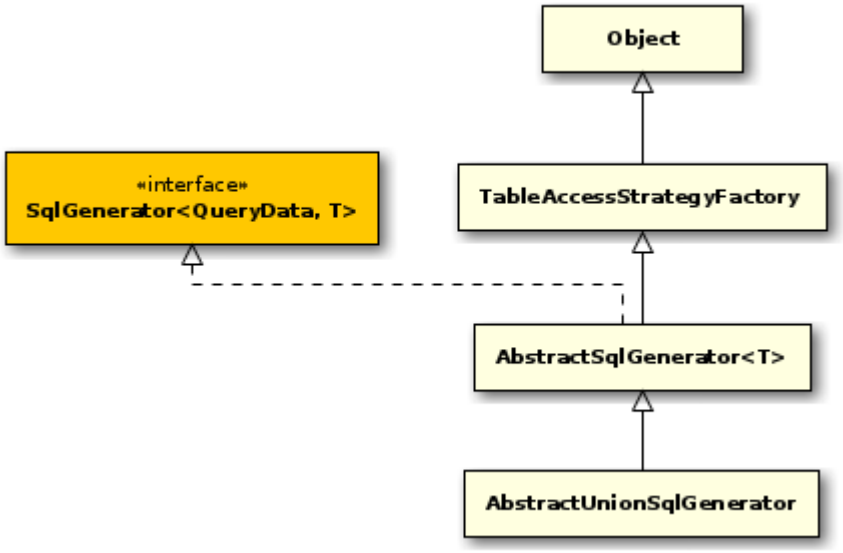
Abstract base class for a SQL statement which coalesces multiple alternatives using UNION. Only the SELECT, FROM, WHERE and GROUP BY clauses are used for each alternative. The ORDER BY and LIMIT/OFFSET clauses are applied to the entire query. It is the responsibility of the calling code to correctly set QueryData.maxWidth and the responsibility of the SelectClauseSqlGenrator to pad the SELECT clause if necessary.

6.4.1. Syntax

```
public abstract class AbstractUnionSqlGenerator<T> extends
AbstractSqlGenerator<T> {

  // Public Constructors
  public AbstractUnionSqlGenerator();

  // Public Methods
  public String toSql(QueryData queryData, String indent);
}
```

Direkt abgeleitete Klassen: `annis.sqlgen.AnnotateInnerQuerySqlGenerator`, `annis.sqlgen.FindSqlGenerator`

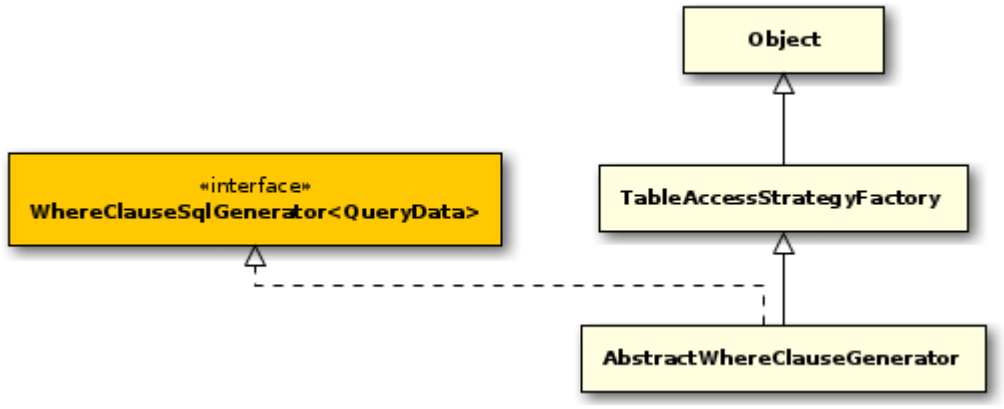
Autor



6.5. Klasse AbstractWhereClauseGenerator

```
public abstract class AbstractWhereClauseGenerator extends TableAccessStrategyFac-
tory
   implements WhereClauseSqlGenerator<QueryData> {

  // Public Constructors
  public AbstractWhereClauseGenerator();

  // Public Methods
  public Set<String> whereConditions(QueryData queryData,
                                     List<QueryNode> alternative,
                                     String indent);
}
```

Direkt abgeleitete Klassen: `annis.sqlgen.DefaultWhereClauseGenerator`



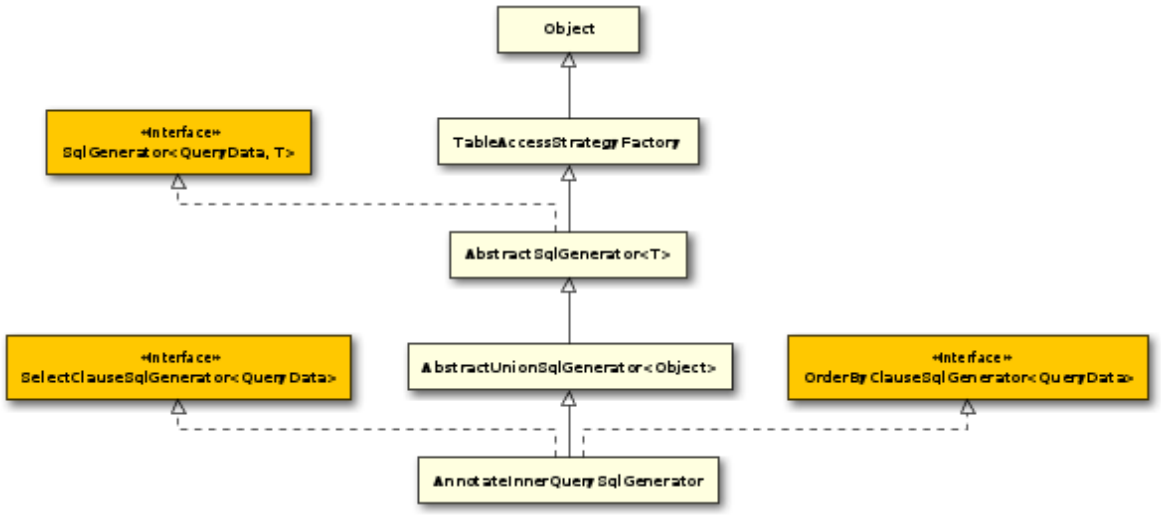
6.6. Schnittstelle AnnotateExtractor

```
public interface AnnotateExtractor<T> {

  // Public Methods
  public void setOuterQueryTableAccessStrategy(TableAccessStrategy outerQueryTableAccessStrategy)
}
```

Autor thomas

6.7. Klasse AnnotateInnerQuerySqlGenerator

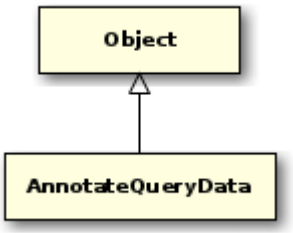
```
public class AnnotateInnerQuerySqlGenerator extends
AbstractUnionSqlGenerator<Object>
   implements SelectClauseSqlGenerator<QueryData>,
              OrderByClauseSqlGenerator<QueryData> {

  // Public Constructors
  public AnnotateInnerQuerySqlGenerator();

  // Public Methods
  public Object extractData(ResultSet rs)
    throws SQLException,
           DataAccessException;
  public SolutionKey<?> getSolutionKey();
  public boolean isSortSolutions();
  public String orderByClause(QueryData queryData,
                              List<QueryNode> alternative,
                              String indent);
  public String selectClause(QueryData queryData,
                             List<QueryNode> alternative,
                             String indent);
  public void setSolutionKey(SolutionKey<?> solutionKey);
  public void setSortSolutions(boolean sortSolutions);
  public String toSql(QueryData queryData, String indent);
}
```



6.8. Klasse AnnotateQueryData

```
public class AnnotateQueryData {

  // Public Constructors
  public AnnotateQueryData(int left, int right);
  public AnnotateQueryData(int left, int right, String segmentationLayer);

  // Public Methods
  public int getLeft();
  public int getRight();
  public String getSegmentationLayer();
  public String toString();
}
```

Autor benjamin



6.9. Klasse AnnotateSqlGenerator

```
public abstract class AnnotateSqlGenerator<T> extends AbstractSqlGenerator<T>
   implements SelectClauseSqlGenerator<QueryData>,
              FromClauseSqlGenerator<QueryData>,
              WhereClauseSqlGenerator<QueryData>,
              OrderByClauseSqlGenerator<QueryData>,
              WithClauseSqlGenerator<QueryData>,
              AnnotateExtractor<T> {

  // Public Constructors
  public AnnotateSqlGenerator();
```

```
  // Public Methods
  public T extractData(ResultSet resultSet)
    throws SQLException,
           DataAccessException;
  public abstract String fromClause(QueryData queryData,
                                    List<QueryNode> alternative,
                                    String indent);
  public CorpusPathExtractor getCorpusPathExtractor();
  public String getDefaultIslandsPolicy();
  public abstract String getDocumentQuery(String toplevelCorpusName,
                                          String documentName);
  public AnnotateInnerQuerySqlGenerator getInnerQuerySqlGenerator();
  public String getMatchedNodesViewName();
  public TableAccessStrategy getOuterQueryTableAccessStrategy();
  public <any> getResultExtractor();
  public TableJoinsInFromClauseSqlGenerator getTableJoinsInFromClauseSqlGenera-
tor();
  @Deprecated
  public abstract String getTextQuery(long textID);
  public boolean isIncludeDocumentNameInAnnotateQuery();
  public boolean isIncludeIsTokenColumn();
  public String orderByClause(QueryData queryData,
                              List<QueryNode> alternative,
                              String indent);
  public T queryAnnotationGraph(JdbcTemplate jdbcTemplate,
                                String toplevelCorpusName,
                                String documentName);
  @Deprecated
  public T queryAnnotationGraph(JdbcTemplate jdbcTemplate,
                                            long textID);
  public abstract String selectClause(QueryData queryData,
                                      List<QueryNode> alternative,
                                      String indent);
  public void setCorpusPathExtractor(CorpusPathExtractor corpusPathExtractor);
  public void setDefaultIslandsPolicy(String defaultIslandsPolicy);
  public void setIncludeDocumentNameInAnnotateQuery(boolean includeDocumentNameInAnnotateQuery);
  public void setIncludeIsTokenColumn(boolean includeIsTokenColumn);
  public void setInnerQuerySqlGenerator(AnnotateInnerQuerySqlGenerator innerQuerySqlGenerator);
  public void setMatchedNodesViewName(String matchedNodesViewName);
  public void setOuterQueryTableAccessStrategy(TableAccessStrategy outerQueryTableAccessStrategy)
  public void setResultExtractor(<any> resultExtractor);
  public void setTableJoinsInFromClauseSqlGenerator(TableJoinsInFromClauseSqlGenerator tableJoins
  public Set<String> whereConditions(QueryData queryData,
                                     List<QueryNode> alternative,
                                     String indent);
  public List<String> withClauses(QueryData queryData,
                                   List<QueryNode> alternative,
                                   String indent);
}
```

Direkt abgeleitete Klassen: `annis.sqlgen.annopool.ApAnnotateSqlGenerator`, `annis.sqlgen.fullfacts.FfAnnotateSqlGenerator`

Autor thomas

6.9.1. queryAnnotationGraph(JdbcTemplate, long)

```
@Deprecated
public T queryAnnotationGraph(JdbcTemplate jdbcTemplate,
                                          long textID);
```

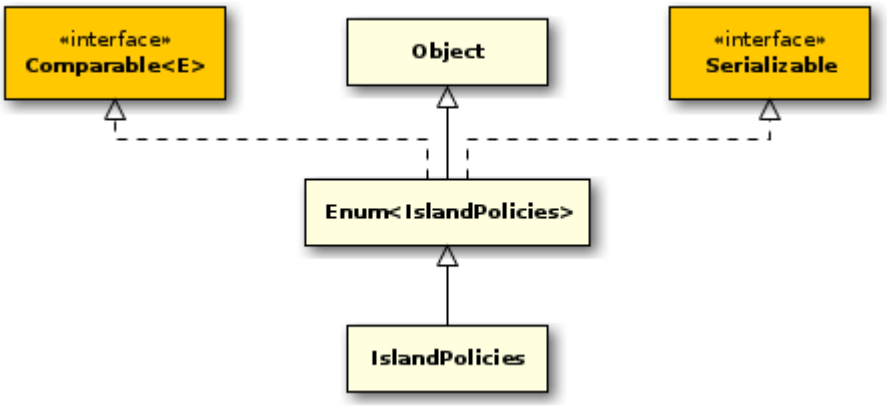
Parameter	
jdbcTemplate	
textID	
return	

Veraltet (Deprecated)

use `annis.sqlgen.AnnotateSqlGenerator` instead

6.10. Klasse AnnotateSqlGenerator.IslandPolicies

```
public static final class AnnotateSqlGenerator.IslandPolicies extends
Enum<IslandPolicies> {

  // Public Static Fields
  public static final IslandPolicies context;
  public static final IslandPolicies none;

  // Public Static Methods
  public static IslandPolicies valueOf(String name);
  public static IslandPolicies[] values();
}
```



6.11. Schnittstelle AnnotationConditionProvider

```
public interface AnnotationConditionProvider {

  // Public Methods
```

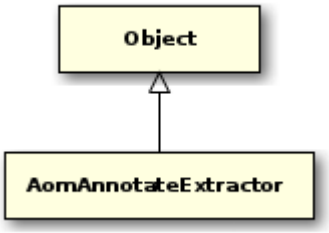
```
  public void addAnnotationConditions(List<String> conditions,
                                      QueryNode node,
                                      int index,
                                      QueryAnnotation annotation,
                                      String table,
                                      QueryData queryData,
                                      TableAccessStrategy tas);
}
```

Autor thomas

6.12. Klasse AomAnnotateExtractor

```
public class AomAnnotateExtractor {

  // Public Constructors
  public AomAnnotateExtractor();

  // Public Methods
  public List<AnnotationGraph> extractData(ResultSet resultSet)
    throws SQLException,
           DataAccessException;
  public TableAccessStrategy getOuterQueryTableAccessStrategy();
  public Annotation mapAnnotation(ResultSet resultSet,
                                  TableAccessStrategy tableAccessStrategy,
                                  String table)
    throws SQLException;
  public Edge mapEdge(ResultSet resultSet,
                      TableAccessStrategy tableAccessStrategy)
    throws SQLException;
  public AnnisNode mapNode(ResultSet resultSet,
                           TableAccessStrategy tableAccessStrategy)
    throws SQLException;
  public void setOuterQueryTableAccessStrategy(TableAccessStrategy outerQueryTableAccessStrategy)
}
```

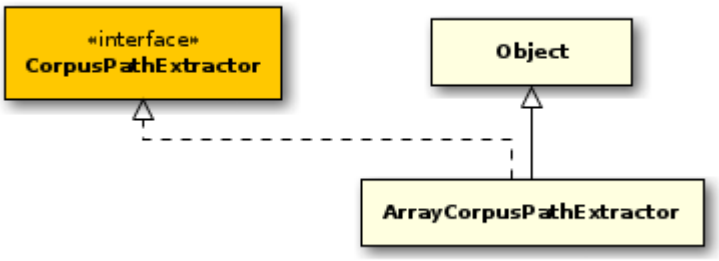
Autor thomas



6.13. Klasse ArrayCorpusPathExtractor

```
public class ArrayCorpusPathExtractor implements CorpusPathExtractor {
```

```
  // Public Constructors
  public ArrayCorpusPathExtractor();

  // Public Methods
  public List<String> extractCorpusPath(ResultSet resultSet, String columnName)
    throws SQLException;
}
```



6.13.1. extractCorpusPath(ResultSet, String)

```
  public List<String> extractCorpusPath(ResultSet resultSet, String columnName)
    throws SQLException;
```

Spezifiziert durch: Methode extractCorpusPath in der Schnittstelle CorpusPathExtractor

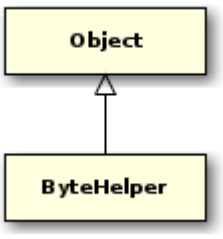
Extracts the path of the document in the corpus tree from a JDBC result set, starting from the root corpus. Note: The path is stored in reverse order in the database. FIXME: Why is the order in the database reversed? Because the ANNOTATE query used to ask for path_name[0] AS document_name? That code appears to be unused.

Parameter	
resultSet	The JDBC result set.
columnName	TODO

Beschreibung kopiert aus der Schnittstelle: extractCorpusPath

6.14. Klasse ByteHelper

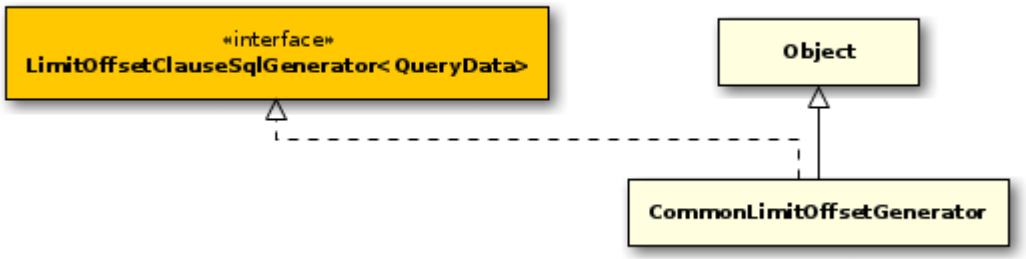
```
public class ByteHelper {

  // Public Constructors
  public ByteHelper();

  // Public Methods
  public AnnisBinary extractData(ResultSet rs)
    throws DataAccessException;
  public String generateSql(String corpusName, int offset, int length);
}
```



6.15. Klasse CommonLimitOffsetGenerator

```
public class CommonLimitOffsetGenerator implements
LimitOffsetClauseSqlGenerator<QueryData> {

  // Public Constructors
  public CommonLimitOffsetGenerator();

  // Public Methods
  public String limitOffsetClause(QueryData queryData,
                                  List<QueryNode> alternative,
                                  String indent);
}
```

Autor benjamin



6.16. Schnittstelle CorpusPathExtractor

```
public interface CorpusPathExtractor {

  // Public Methods
  public List<String> extractCorpusPath(ResultSet resultSet, String columnName)
    throws SQLException;
}
```

6.16.1. extractCorpusPath(ResultSet, String)

```
  public List<String> extractCorpusPath(ResultSet resultSet, String columnName)
    throws SQLException;
```

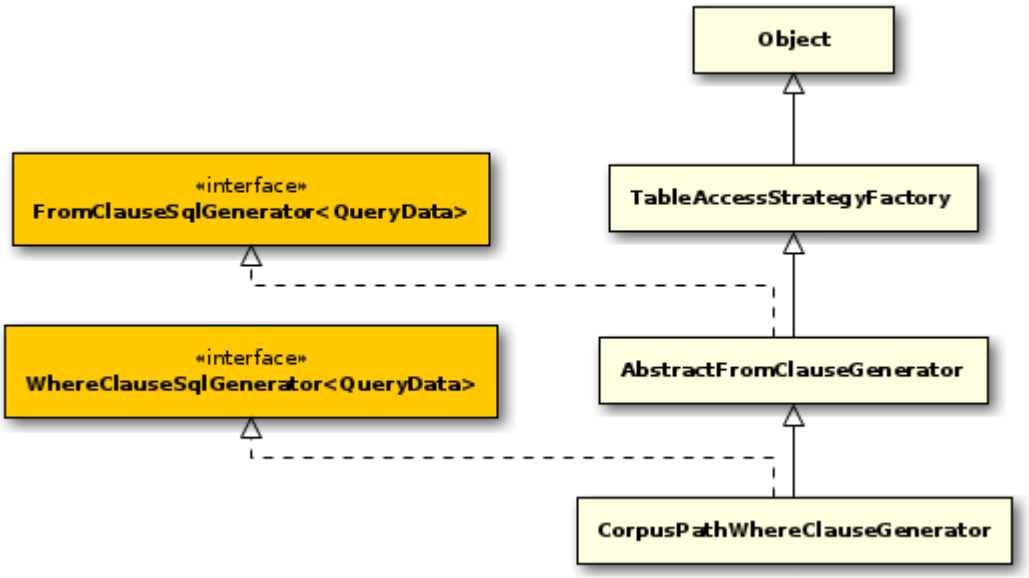
Extracts the path of the document in the corpus tree from a JDBC result set, starting from the root corpus. Note: The path is stored in reverse order in the database. FIXME: Why is the order in the database reversed? Because the ANNOTATE query used to ask for path_name[0] AS document_name? That code appears to be unused.

Parameter	
resultSet	The JDBC result set.
columnName	TODO

6.17. Klasse CorpusPathWhereClauseGenerator

```
public class CorpusPathWhereClauseGenerator extends AbstractFromClauseGenerator
   implements WhereClauseSqlGenerator<QueryData> {

  // Public Constructors
  public CorpusPathWhereClauseGenerator();

  // Public Methods
  public String fromClause(QueryData queryData,
                           List<QueryNode> alternative,
                           String indent);
  public Set<String> whereConditions(QueryData queryData,
                                     List<QueryNode> alternative,
                                     String indent);
}
```

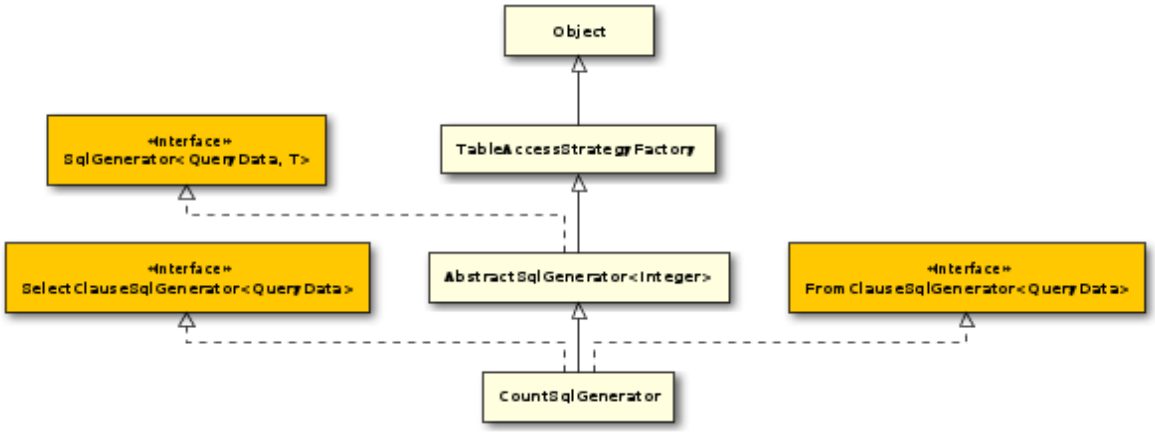
Autor benjamin



6.18. Klasse CountSqlGenerator

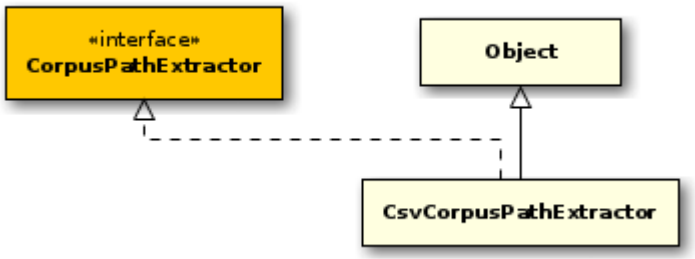
```
public class CountSqlGenerator extends AbstractSqlGenerator<Integer>
   implements SelectClauseSqlGenerator<QueryData>,
              FromClauseSqlGenerator<QueryData> {

  // Public Constructors
  public CountSqlGenerator();

  // Public Methods
  public Integer extractData(ResultSet rs)
    throws SQLException,
           DataAccessException;
  public String fromClause(QueryData queryData,
                           List<QueryNode> alternative,
                           String indent);
  public SqlGenerator getFindSqlGenerator();
  public String selectClause(QueryData queryData,
                             List<QueryNode> alternative,
                             String indent);
```

```
  public void setFindSqlGenerator(SqlGenerator findSqlGenerator);
}
```



6.19. Klasse CsvCorpusPathExtractor

```
public class CsvCorpusPathExtractor implements CorpusPathExtractor {

  // Public Constructors
  public CsvCorpusPathExtractor();

  // Public Methods
  public List<String> extractCorpusPath(ResultSet resultSet, String columnName)
    throws SQLException;
}
```



6.19.1. extractCorpusPath(ResultSet, String)

```
  public List<String> extractCorpusPath(ResultSet resultSet, String columnName)
    throws SQLException;
```

Spezifiziert durch: Methode extractCorpusPath in der Schnittstelle CorpusPathExtractor

Extracts the path of the document in the corpus tree from a JDBC result set, starting from the root corpus. Note: The path is stored in reverse order in the database. FIXME: Why is the order in the database reversed? Because the ANNOTATE query used to ask for path_name[0] AS document_name? That code appears to be unused.

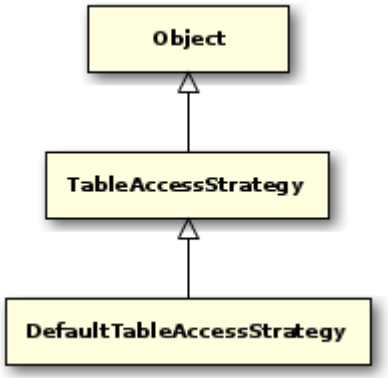
Parameter	
resultSet	The JDBC result set.

columnName	TODO

Beschreibung kopiert aus der Schnittstelle: extractCorpusPath

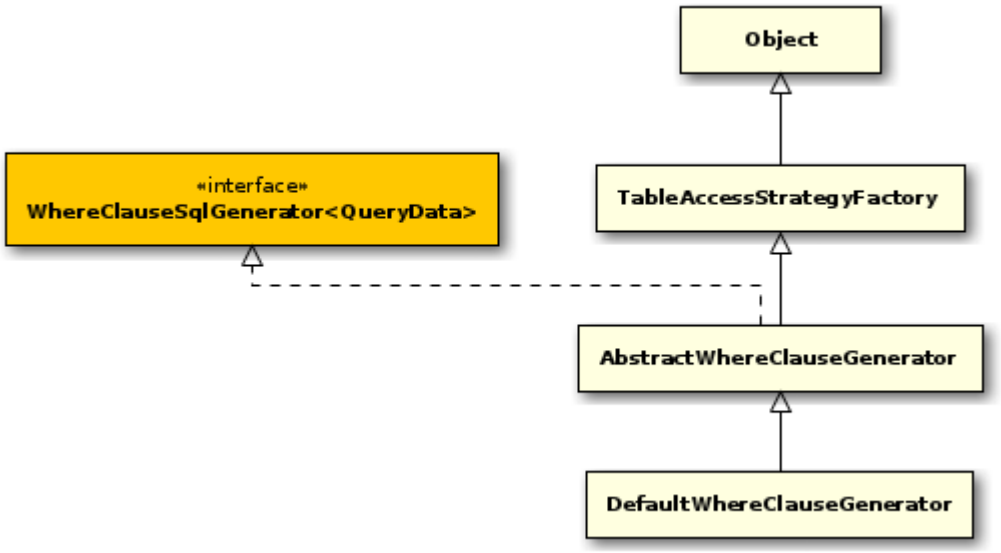
6.20. Klasse DefaultTableAccessStrategy

```
public class DefaultTableAccessStrategy extends TableAccessStrategy {

  // Public Constructors
  public DefaultTableAccessStrategy();
  public DefaultTableAccessStrategy(QueryNode node);
}
```



6.21. Klasse DefaultWhereClauseGenerator

```
public class DefaultWhereClauseGenerator extends AbstractWhereClauseGenerator {

  // Public Constructors
  public DefaultWhereClauseGenerator();

  // Public Methods
  public AnnotationConditionProvider getAnnoCondition();
  public String getComponentPredicates();
  public boolean isAllowIdenticalSibling();
  public boolean isOptimizeInclusion();
  public boolean isOptimizeIndirectPrecedence();
  public boolean isUseComponentRefPredicateInCommonAncestorSubquery();
  public boolean isUseIsTokenColumn();
  public boolean isUseToplevelCorpusPredicateInCommonAncestorSubquery();
  public void setAllowIdenticalSibling(boolean allowIdenticalSibling);
  public void setAnnoCondition(AnnotationConditionProvider annoCondition);
  public void setComponentPredicates(String componentPredicates);
  public void setOptimizeInclusion(boolean optimizeInclusion);
  public void setOptimizeIndirectPrecedence(boolean optimizeIndirectPrecedence);
  public void setUseComponentRefPredicateInCommonAncestorSubquery(boolean useComponentRefPredicat
  public void setUseIsTokenColumn(boolean useIsTokenColumn);
  public void setUseToplevelCorpusPredicateInCommonAncestorSubquery(boolean useToplevelCorpusPred
}
```



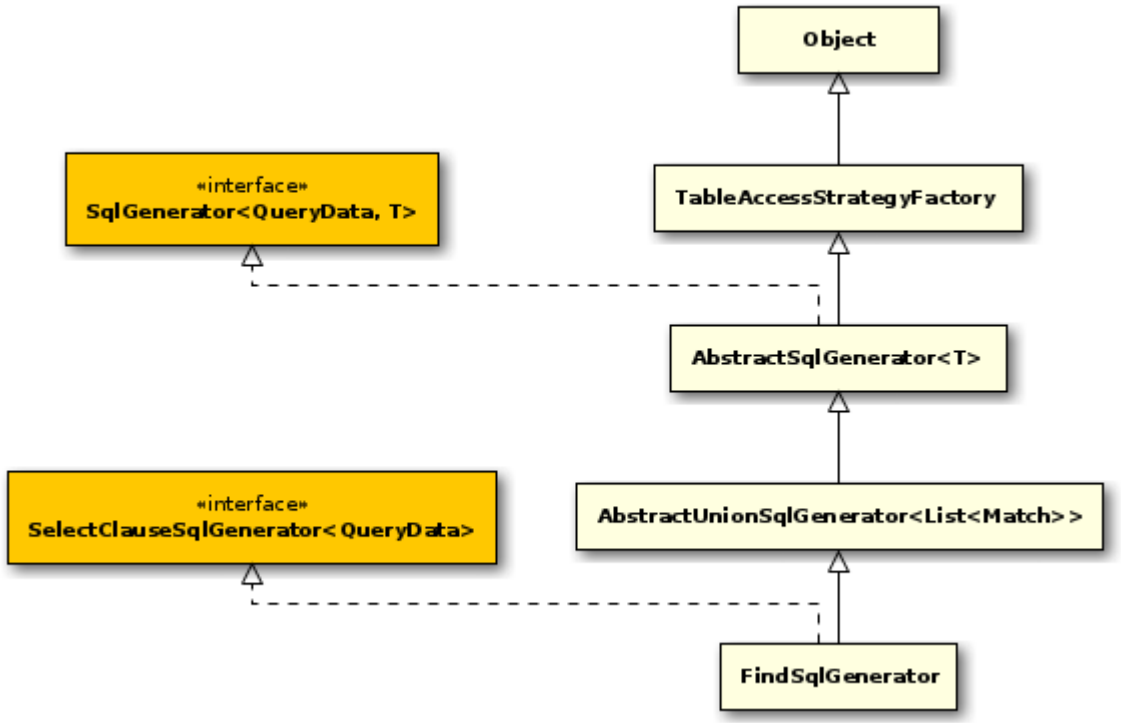
6.22. Klasse FindSqlGenerator

Generates identifers for salt which are needed for the `annis.service.internal.AnnisWebService.find`

6.22.1. Syntax

```
public class FindSqlGenerator extends AbstractUnionSqlGenerator<List<Match>>
   implements SelectClauseSqlGenerator<QueryData> {

  // Public Constructors
  public FindSqlGenerator();

  // Public Methods
  public List<Match> extractData(ResultSet rs)
    throws SQLException,
           DataAccessException;
  public CorpusPathExtractor getCorpusPathExtractor();
  public boolean isOptimizeDistinct();
  public Match mapRow(ResultSet rs, int rowNum)
    throws SQLException;
  public String selectClause(QueryData queryData,
                             List<QueryNode> alternative,
                             String indent);
  public void setCorpusPathExtractor(CorpusPathExtractor corpusPathExtractor);
  public void setOptimizeDistinct(boolean optimizeDistinct);
}
```

Autor Benjamin Weißenfels



6.23. Schnittstelle FromClauseSqlGenerator

```
public interface FromClauseSqlGenerator<T> {

  // Public Methods
  public String fromClause(T queryData,
                           List<QueryNode> alternative,
                           String indent);
}
```

6.24. Schnittstelle GroupByClauseSqlGenerator

```
public interface GroupByClauseSqlGenerator<T> {

  // Public Methods
  public String groupByAttributes(T queryData, List<QueryNode> alternative);
}
```

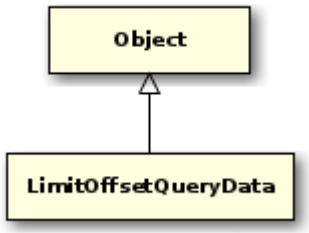
6.25. Schnittstelle LimitOffsetClauseSqlGenerator

```
public interface LimitOffsetClauseSqlGenerator<T> {

  // Public Methods
  public String limitOffsetClause(T queryData,
                                  List<QueryNode> alternative,
                                  String indent);
}
```

6.26. Klasse LimitOffsetQueryData

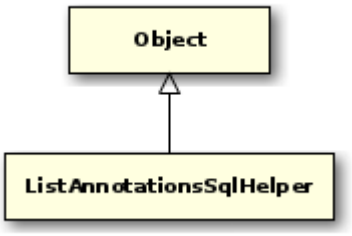
```
public class LimitOffsetQueryData {

  // Public Constructors
  public LimitOffsetQueryData(int offset, int limit);

  // Public Methods
  public int getLimit();
  public int getOffset();
  public boolean isPaged();
}
```

Autor benjamin



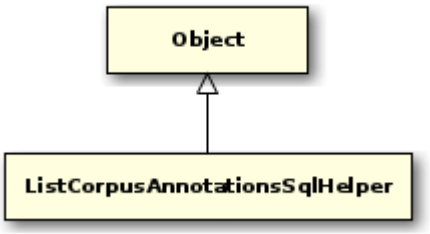
6.27. Klasse ListAnnotationsSqlHelper

```
public class ListAnnotationsSqlHelper {

  // Public Constructors
  public ListAnnotationsSqlHelper();

  // Public Methods
  public String createSqlQuery(List<Long> corpusList,
                               boolean listValues,
                               boolean onlyMostFrequentValue);
  public Object extractData(ResultSet resultSet)
    throws SQLException,
           DataAccessException;
}
```



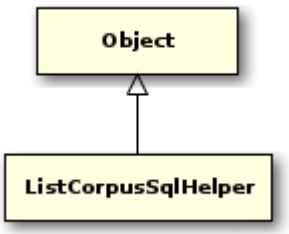
6.28. Klasse ListCorpusAnnotationsSqlHelper

```
public class ListCorpusAnnotationsSqlHelper {

  // Public Constructors
  public ListCorpusAnnotationsSqlHelper();

  // Public Methods
  public String createSqlQuery(String toplevelCorpusName, String corpusName);
  public String createSqlQuery(long corpusId);
  public Annotation mapRow(ResultSet rs, int rowNum)
    throws SQLException;
}
```



6.29. Klasse ListCorpusSqlHelper

```
public class ListCorpusSqlHelper {

  // Public Constructors
  public ListCorpusSqlHelper();

  // Public Methods
  public String createSqlQuery();
  public AnnisCorpus mapRow(ResultSet rs, int rowNum)
    throws SQLException;
}
```



6.30. Klasse MatrixSqlGenerator

```
public class MatrixSqlGenerator extends AbstractSqlGenerator<List<AnnotatedMatch>>
   implements SelectClauseSqlGenerator<QueryData>,
              FromClauseSqlGenerator<QueryData>,
              WhereClauseSqlGenerator<QueryData>,
              GroupByClauseSqlGenerator<QueryData> {

  // Public Constructors
  public MatrixSqlGenerator();

  // Public Methods
  public List<AnnotatedMatch> extractData(ResultSet resultSet)
    throws SQLException,
           DataAccessException;
  public String fromClause(QueryData queryData,
                           List<QueryNode> alternative,
                           String indent);
  public SqlGenerator<QueryData, ?> getInnerQuerySqlGenerator();
  public String getMatchedNodesViewName();
  @Deprecated
  public String getMatrixQuery(List<Long> corpusList,
                                              int maxWidth);
  public TableJoinsInFromClauseSqlGenerator getTableJoinsInFromClauseGenerator();
  public String groupByAttributes(QueryData queryData,
                                  List<QueryNode> alternative);
  public List<AnnotatedMatch> queryMatrix(JdbcTemplate jdbcTemplate,
                                          List<Long> corpusList,
                                          int maxWidth);
  public String selectClause(QueryData queryData,
                             List<QueryNode> alternative,
                             String indent);
  public void setInnerQuerySqlGenerator(SqlGenerator<QueryData, ?> innerQuerySqlGe-
nerator);
  public void setMatchedNodesViewName(String matchedNodesViewName);
  public void setTableJoinsInFromClauseGenerator(TableJoinsInFromClauseSqlGenerator tableJoinsInF
  public Set<String> whereConditions(QueryData queryData,
                                     List<QueryNode> alternative,
                                     String indent);
}
```

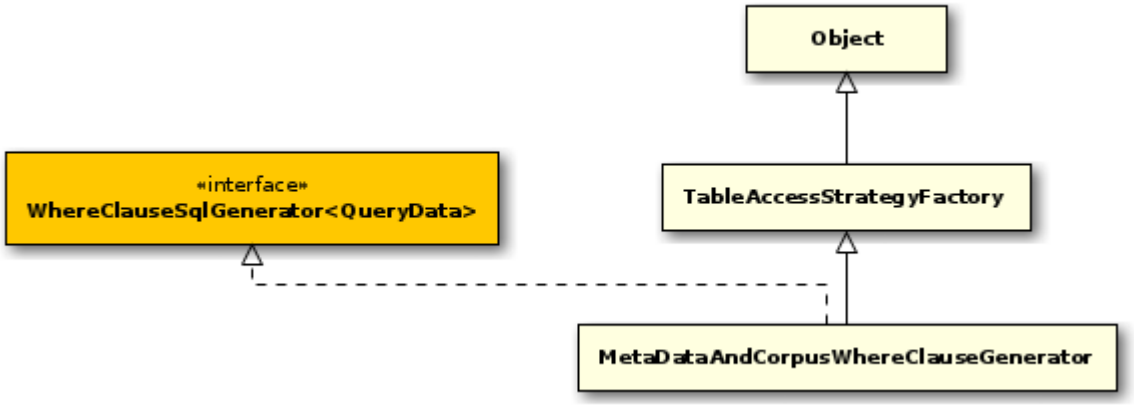
Autor thomas

6.31. Klasse MetaDataAndCorpusWhereClauseGenerator

```
public class MetaDataAndCorpusWhereClauseGenerator extends TableAccessStrategyFac-
tory
```

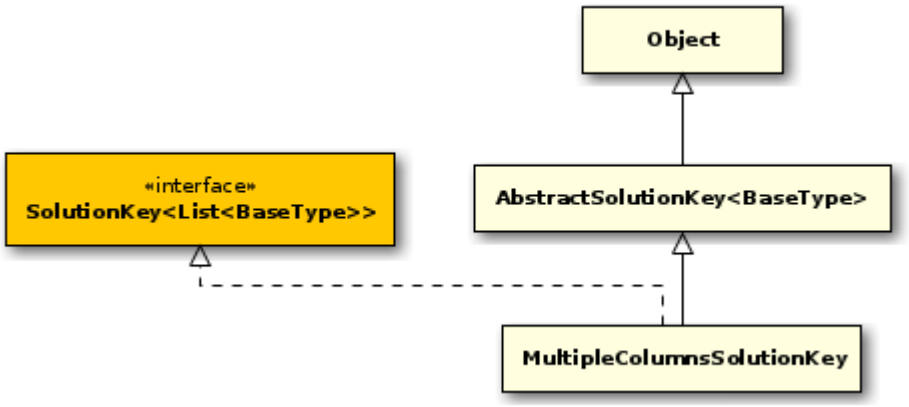
```
   implements WhereClauseSqlGenerator<QueryData> {

  // Public Constructors
  public MetaDataAndCorpusWhereClauseGenerator();

  // Public Methods
  public Set<String> whereConditions(QueryData queryData,
                                     List<QueryNode> alternative,
                                     String indent);
}
```

Autor thomas



6.32. Klasse MultipleColumnsSolutionKey

```
public class MultipleColumnsSolutionKey<BaseType> extends
AbstractSolutionKey<BaseType>
   implements SolutionKey<List<BaseType>> {

  // Public Constructors
  public MultipleColumnsSolutionKey();

  // Public Methods
  public List<String> generateOuterQueryColumns(TableAccessStrategy tableAccessStrategy,
                                                 int size);
  public String getKeyColumnName();
  public List<String> getKeyColumns(int size);
  public List<BaseType> retrieveKey(ResultSet resultSet);
  public void setKeyColumnName(String keyColumnName);
}
```



6.32.1. generateOuterQueryColumns(TableAccessStrategy, int)

```
public List<String> generateOuterQueryColumns(TableAccessStrategy tableAccessStrategy,
                                               int size);
```

Spezifiziert durch: Methode generateOuterQueryColumns in der Schnittstelle SolutionKey

Generate the key(s) for an annotation graph.

Parameter	
tableAccessStrategy	TODO
size	TODO
return	A list of column aliases that are used in the SELECT clause of the outer ANNOTATE query.

Beschreibung kopiert aus der Schnittstelle: generateOuterQueryColumns

6.32.2. getKeyColumns(int)

```
public List<String> getKeyColumns(int size);
```

Spezifiziert durch: Methode getKeyColumns in der Schnittstelle SolutionKey

Returns the name of the key columns.

Parameter	
size	The number of matched nodes in a solution.

Beschreibung kopiert aus der Schnittstelle: getKeyColumns

6.32.3. retrieveKey(ResultSet)

```
public List<BaseType> retrieveKey(ResultSet resultSet);
```

Spezifiziert durch: Methode retrieveKey in der Schnittstelle SolutionKey

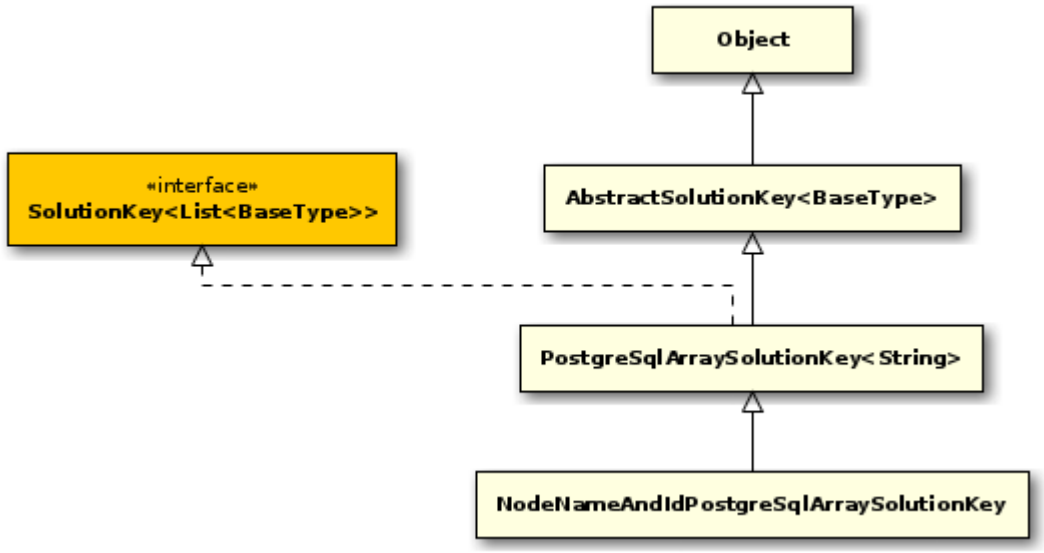
Retrieve (and validate) the annotation graph key from the current row of the JDBC result set.

Parameter	
resultSet	The JDBC result set returned by an ANNOTATE query.

Beschreibung kopiert aus der Schnittstelle: retrieveKey

6.33. Klasse NodeNameAndIdPostgreSqlArraySolutionKey

```
public class NodeNameAndIdPostgreSqlArraySolutionKey extends
PostgreSqlArraySolutionKey<String> {

  // Public Constructors
  public NodeNameAndIdPostgreSqlArraySolutionKey();

  // Public Methods
  public List<String> generateInnerQueryColumns(TableAccessStrategy tableAccessStrategy,
                                                 int index);
  public List<String> generateOuterQueryColumns(TableAccessStrategy tableAccessStrategy,
                                                 int size);
}
```



6.34. Schnittstelle NodeSqlAdapter

```
public interface NodeSqlAdapter {

  // Public Methods
  public String fromClause();
  public String selectClause();
  public String selectClauseNullValues();
  public List<String> whereClause();
}
```

6.35. Schnittstelle OrderByClauseSqlGenerator

```
public interface OrderByClauseSqlGenerator<T> {

  // Public Methods
  public String orderByClause(T queryData,
                              List<QueryNode> alternative,
                              String indent);
}
```

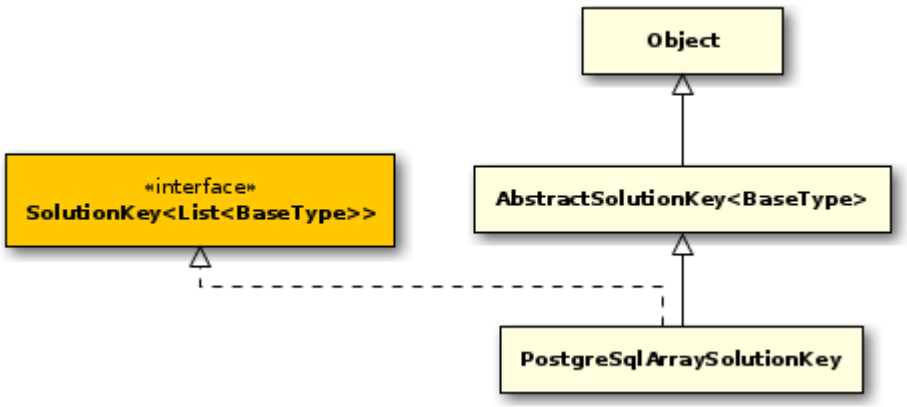
6.36. Klasse PostgreSqlArraySolutionKey

TODO Document semantics of of multiple ID columns TODO Code supporting multiple ID columns complicates things unnecessarily

6.36.1. Syntax

```
public class PostgreSqlArraySolutionKey<BaseType> extends
AbstractSolutionKey<BaseType>
   implements SolutionKey<List<BaseType>> {

  // Public Constructors
  public PostgreSqlArraySolutionKey();

  // Public Methods
  public List<String> generateOuterQueryColumns(TableAccessStrategy tableAccessStrategy,
                                                 int size);
  public String getKeyColumnName();
  public List<String> getKeyColumns(int size);
  public List<BaseType> retrieveKey(ResultSet resultSet);
  public void setKeyColumnName(String keyColumnName);
}
```

Direkt abgeleitete Klassen: `annis.sqlgen.NodeNameAndIdPostgreSqlArraySolutionKey`



6.36.2. generateOuterQueryColumns(TableAccessStrategy, int)

```
  public List<String> generateOuterQueryColumns(TableAccessStrategy tableAccessStrategy,
```

```
                                                    int size);
```

Spezifiziert durch: Methode generateOuterQueryColumns in der Schnittstelle SolutionKey

Generate the key(s) for an annotation graph.

Parameter	
tableAccessStrategy	TODO
size	TODO
return	A list of column aliases that are used in the SELECT clause of the outer ANNOTATE query.

Beschreibung kopiert aus der Schnittstelle: generateOuterQueryColumns

6.36.3. getKeyColumns(int)

```
public List<String> getKeyColumns(int size);
```

Spezifiziert durch: Methode getKeyColumns in der Schnittstelle SolutionKey

Returns the name of the key columns.

Parameter	
size	The number of matched nodes in a solution.

Beschreibung kopiert aus der Schnittstelle: getKeyColumns

6.36.4. retrieveKey(ResultSet)

```
public List<BaseType> retrieveKey(ResultSet resultSet);
```

Spezifiziert durch: Methode retrieveKey in der Schnittstelle SolutionKey

Retrieve (and validate) the annotation graph key from the current row of the JDBC result set.

Parameter	
resultSet	The JDBC result set returned by an ANNOTATE query.

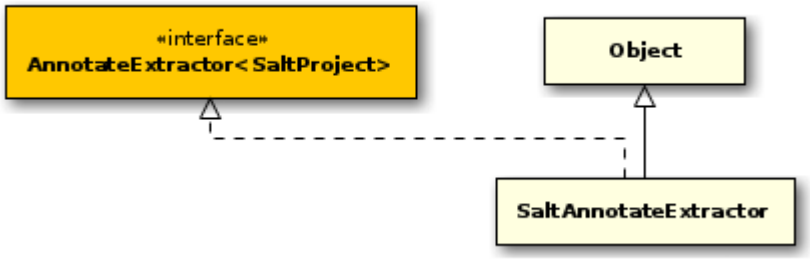
Beschreibung kopiert aus der Schnittstelle: retrieveKey

6.37. Klasse SaltAnnotateExtractor

```
public class SaltAnnotateExtractor implements AnnotateExtractor<SaltProject> {

  // Public Constructors
  public SaltAnnotateExtractor();

  // Public Methods
  public SaltProject extractData(ResultSet resultSet)
    throws SQLException,
           DataAccessException;
  public CorpusPathExtractor getCorpusPathExtractor();
  public TableAccessStrategy getOuterQueryTableAccessStrategy();
  public void setCorpusPathExtractor(CorpusPathExtractor corpusPathExtractor);
  public void setOuterQueryTableAccessStrategy(TableAccessStrategy outerQueryTableAccessStrategy)
```

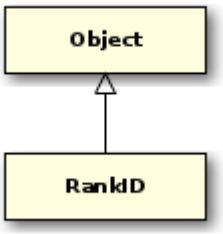
}

Autor



6.38. Klasse SaltAnnotateExtractor.RankID

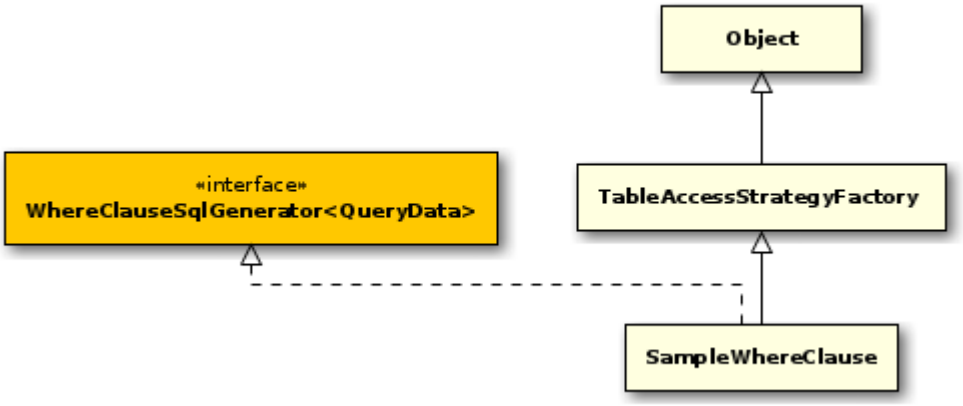
```
public static class SaltAnnotateExtractor.RankID {

  // Public Constructors
  public SaltAnnotateExtractor.RankID(long componentID, long pre);

  // Public Methods
  public boolean equals(Object obj);
  public long getComponentID();
  public long getPre();
  public int hashCode();
}
```



6.39. Klasse SampleWhereClause

```
public class SampleWhereClause extends TableAccessStrategyFactory
   implements WhereClauseSqlGenerator<QueryData> {

  // Public Constructors
  public SampleWhereClause();

  // Public Methods
  @Deprecated
  public List<String> whereConditions(QueryNode node,
                                      List<Long> corpusList,
                                      List<Long> documents);
  public Set<String> whereConditions(QueryData queryData,
                                     List<QueryNode> alternative,
                                     String indent);
}
```

Autor thomas



6.40. Schnittstelle SelectClauseSqlGenerator

```
public interface SelectClauseSqlGenerator<T> {

  // Public Methods
  public String selectClause(T queryData,
                             List<QueryNode> alternative,
                             String indent);
}
```

6.41. Schnittstelle SolutionKey

```
public interface SolutionKey<KeyType> {

  // Public Methods
  public List<String> generateInnerQueryColumns(TableAccessStrategy tableAccessStrategy,
                                                 int index);
  public List<String> generateOuterQueryColumns(TableAccessStrategy tableAccessStrategy,
                                                 int size);
  public String getCurrentKeyAsString();
  public List<String> getKeyColumns(int size);
  public int getKeySize();
  public Integer getMatchedNodeIndex(Object id);
  public Object getNodeId(ResultSet resultSet,
                          TableAccessStrategy tableAccessStrategy);
  public boolean isNewKey();
  public KeyType retrieveKey(ResultSet resultSet);
}
```

6.41.1. generateInnerQueryColumns(TableAccessStrategy, int)

```
public List<String> generateInnerQueryColumns(TableAccessStrategy tableAccessStrategy,
                                               int index);
```

Generate list of column aliases that are used to identify a node in a matching solution in the inner query of an ANNOTATE function query.

Parameter	
tableAccessStrategy	TODO
index	TODO
return	A list of column aliases that are used in the SELECT clause of the inner query.

6.41.2. generateOuterQueryColumns(TableAccessStrategy, int)

```
public List<String> generateOuterQueryColumns(TableAccessStrategy tableAccessStrategy,
                                               int size);
```

Generate the key(s) for an annotation graph.

Parameter	
tableAccessStrategy	TODO
size	TODO
return	A list of column aliases that are used in the SELECT clause of the outer ANNOTATE query.

6.41.3. getCurrentKeyAsString()

```
public String getCurrentKeyAsString();
```

Returns the String representation of the key that can be used in the Salt model for the annis.model.AnnisConstants.FEAT_MACHTEDIDS property.

6.41.4. getKeyColumns(int)

```
public List<String> getKeyColumns(int size);
```

Returns the name of the key columns.

Parameter	
size	The number of matched nodes in a solution.

6.41.5. getMatchedNodeIndex(Object)

```
public Integer getMatchedNodeIndex(Object id);
```

Retrieve the search term index for which a given node is a match. A node is a match for a given search term if its name is part of the current row's key.

Parameter	
name	A node name
return	The index of the search term for which the node is a match (starting with 1) or `null` if the node is not a match.

6.41.6. getNodeId(ResultSet, TableAccessStrategy)

```
public Object getNodeId(ResultSet resultSet,
```

```
                                TableAccessStrategy tableAccessStrategy);
```

Returns the node ID of the current row in a result set.

6.41.7. isNewKey()

```
  public boolean isNewKey();
```

Has the key changed from the last row to this one.

Parameter	
return	True, if the key has changed from the last row to this one.

6.41.8. retrieveKey(ResultSet)

```
  public KeyType retrieveKey(ResultSet resultSet);
```

Retrieve (and validate) the annotation graph key from the current row of the JDBC result set.

Parameter	
resultSet	The JDBC result set returned by an ANNOTATE query.

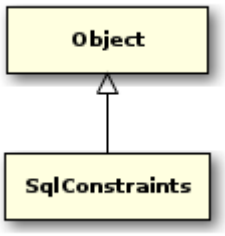
6.42. Klasse SqlConstraints

TODO: write documentation for BaseSqlClauseGenerator, fix name

6.42.1. Syntax

```
public class SqlConstraints {

  // Public Constructors
  public SqlConstraints();

  // Public Static Methods
  public static String between(String lhs, int min, int max);
  public static String between(String lhs, String rhs, int min, int max);
  public static String bitSelect(String column, boolean[] bits);
  public static String in(String lhs, String rhs);
  public static String in(String lhs, Collection<?> values);
  public static boolean isDisableBetweenPredicate();
  public static boolean isDisableBetweenSymmetricPredicate();
  public static String isFalse(String lhs);
  public static String isNotNull(String lhs);
  public static String isNull(String lhs);
  public static String isTrue(String lhs);
  public static String join(String op, String lhs, String rhs);
  public static String numberJoin(String op,
                                  String lhs,
                                  String rhs,
                                  int offset);
  public static void setDisableBetweenPredicate(boolean disableBetweenPredicate);
  public static void setDisableBetweenSymmetricPredicate(boolean disableBetweenPredicate);
  public static String sqlString(String string);
  public static String sqlString(String string, TextMatching textMatching);
}
```

Autor



6.42.2. isFalse(String)

```
public static String isFalse(String lhs);
```

Test a column for true.

Parameter	
lhs	The column which should be checked for false.

6.42.3. isNotNull(String)

```
public static String isNotNull(String lhs);
```

Generate IS NOT NULL predicate on column.

Parameter	
lhs	The column which should be checked for values that are not NULL.

6.42.4. isNull(String)

```
public static String isNull(String lhs);
```

Generate IS NULL predicate on column.

Parameter	
lhs	The column which should be checked for NULL values.

6.42.5. isTrue(String)

```
public static String isTrue(String lhs);
```

Test a column for true.

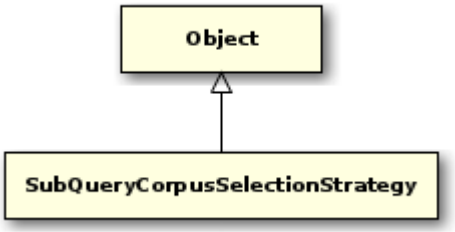
Parameter	
lhs	The column which should be checked for true.

6.43. Schnittstelle SqlGenerator

```
public interface SqlGenerator<QueryType,ResultType> {

  // Public Methods
  public String toSql(QueryType queryData);
  public String toSql(QueryType queryData, String indent);
}
```

6.44. Klasse SubQueryCorpusSelectionStrategy

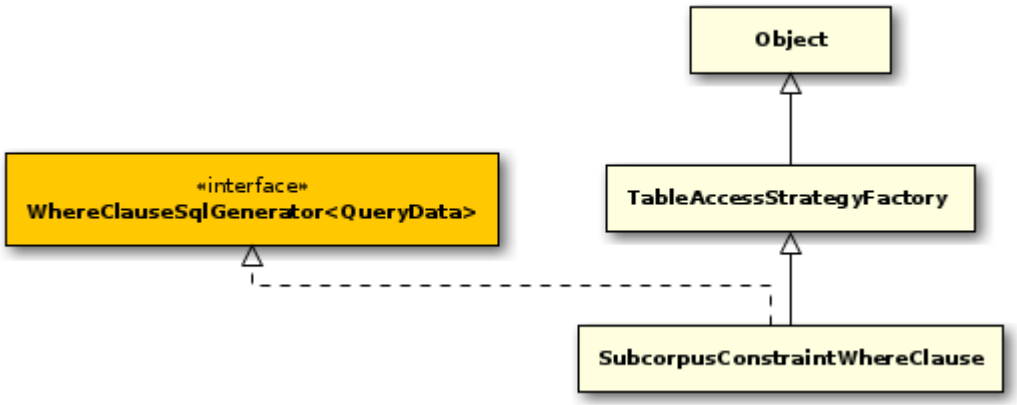
```
public class SubQueryCorpusSelectionStrategy {

  // Public Constructors
  public SubQueryCorpusSelectionStrategy();

  // Public Methods
  public String buildSubQuery(List<Long> corpusList,
                              List<QueryAnnotation> metaData);
  public boolean hasCorpusSelection(List<Long> corpusList,
                                    List<QueryAnnotation> metaData);
}
```



6.45. Klasse SubcorpusConstraintWhereClause

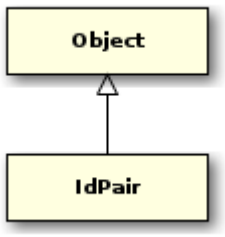
```
public class SubcorpusConstraintWhereClause extends TableAccessStrategyFactory
   implements WhereClauseSqlGenerator<QueryData> {

  // Public Constructors
  public SubcorpusConstraintWhereClause();

  // Public Methods
  @Deprecated
  public List<String> commonWhereConditions(List<QueryNode> nodes,
                                            List<Long> corpusList,
                                            List<Long> documents);
  public Set<String> whereConditions(QueryData queryData,
                                     List<QueryNode> alternative,
                                     String indent);
}
```

Autor thomas



6.46. Klasse SubcorpusConstraintWhereClause.IdPair

```
public static class SubcorpusConstraintWhereClause.IdPair {

  // Public Fields
  public long id1;
  public long id2;

  // Public Constructors
  public SubcorpusConstraintWhereClause.IdPair(long id1, long id2);

  // Public Methods
  public boolean equals(Object obj);
  public int hashCode();
}
```

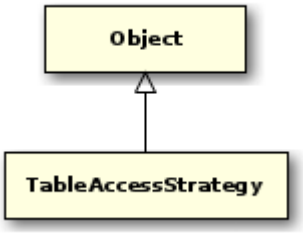


6.47. Klasse TableAccessStrategy

```
public class TableAccessStrategy {

  // Public Static Fields
  public static final String ANNOTATION_POOL_TABLE = "annotation_pool";
  public static final String COMPONENT_TABLE = "component";
  public static final String CORPUS_ANNOTATION_TABLE = "corpus_annotation";
  public static final String CORPUS_TABLE = "corpus";
  public static final String EDGE_ANNOTATION_TABLE = "edge_annotation";
  public static final String FACTS_TABLE = "facts";
  public static final String NODE_ANNOTATION_TABLE = "node_annotation";
  public static final String NODE_TABLE = "node";
  public static final String RANK_TABLE = "rank";
  public static final String TEXT_TABLE = "text";

  // Public Constructors
  public TableAccessStrategy();
  public TableAccessStrategy(QueryNode node);

  // Public Methods
  public void addColumnAlias(String table, String column, String alias);
  public void addTableAlias(String table, String alias);
  public String aliasedColumn(String table, String column);
  public String aliasedColumn(String table, String column, int count);
  public String aliasedTable(String table, int count);
  public String columnName(String table, String column);
  public Map<String, Map<String, String>> getColumnAliases();
```

```
  public QueryNode getNode();
  public Map<String, String> getTableAliases();
  public boolean isMaterialized(String table, String otherTable);
  public void setColumnAliases(Map<String, Map<String, String>> columnAliases);
  public void setNode(QueryNode node);
  public void setTableAliases(Map<String, String> tableAliases);
  public String tableName(String table);
  public boolean usesComponentTable();
  public boolean usesEdgeAnnotationTable();
  public boolean usesNodeAnnotationTable();
  public boolean usesRankTable();
}
```

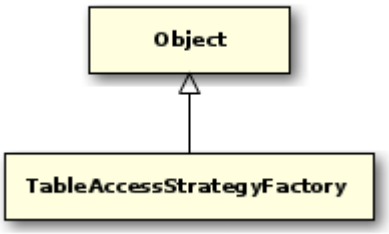
Direkt abgeleitete Klassen: `annis.sqlgen.DefaultTableAccessStrategy`



6.48. Klasse TableAccessStrategyFactory

```
public class TableAccessStrategyFactory {

  // Public Constructors
  public TableAccessStrategyFactory();

  // Public Methods
  public TableAccessStrategy tables(QueryNode node);
}
```

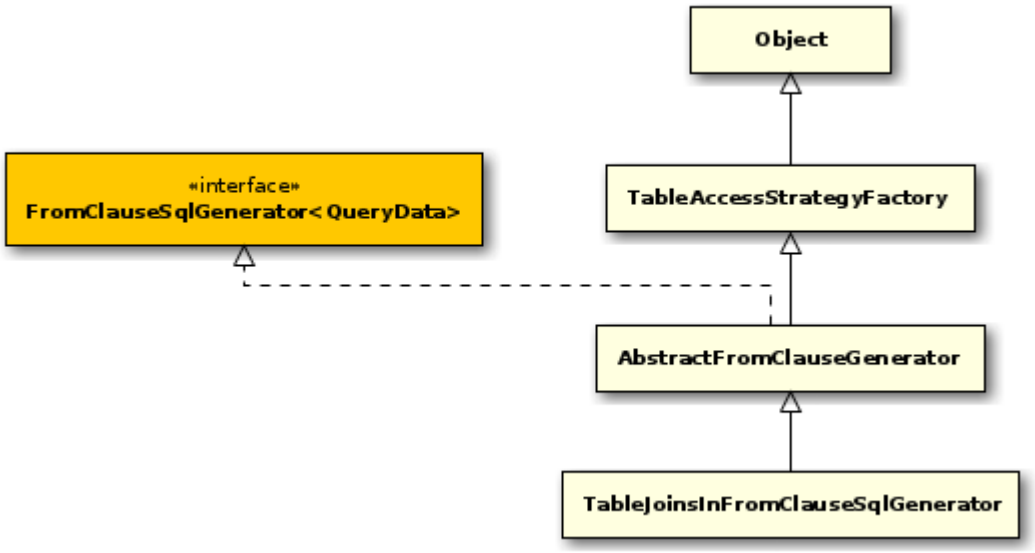
Direkt abgeleitete Klassen: `annis.sqlgen.AbstractFromClauseGenerator`, `annis.sqlgen.AbstractSqlGenerator`, `annis.sqlgen.AbstractWhereClauseGenerator`, `annis.sqlgen.MetaDataAndCorpusWhereClauseGenerator`, `annis.sqlgen.SampleWhereClause`, `annis.sqlgen.SubcorpusConstraintWhereClause`



6.49. Klasse TableJoinsInFromClauseSqlGenerator

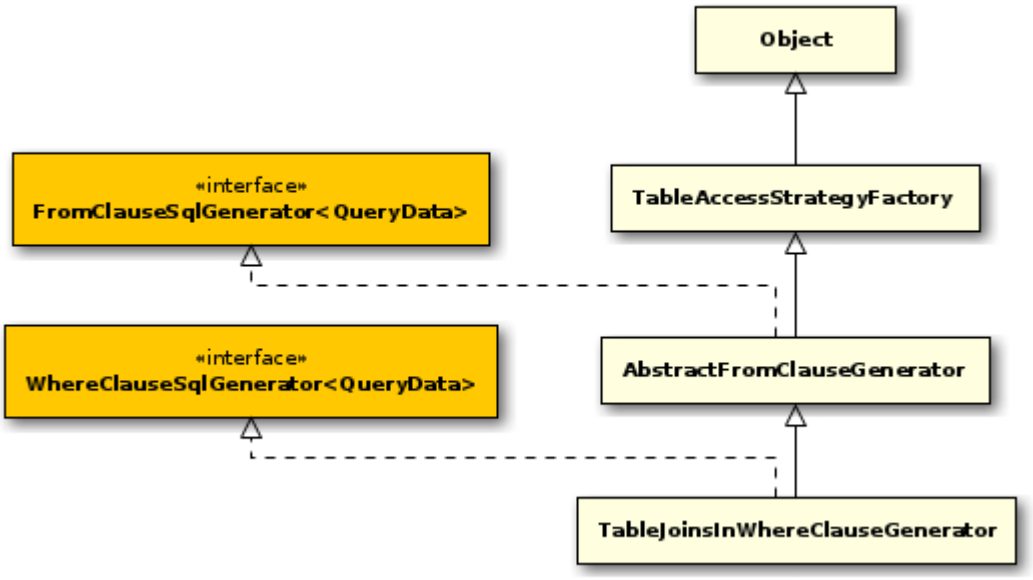
```
public class TableJoinsInFromClauseSqlGenerator extends AbstractFromClauseGenera-
tor {
```

```
  // Public Constructors
  public TableJoinsInFromClauseSqlGenerator();

  // Public Methods
  public String fromClause(QueryData queryData,
                           List<QueryNode> alternative,
                           String indent);
  public String fromClauseForNode(QueryNode node, boolean leftJoin);
}
```



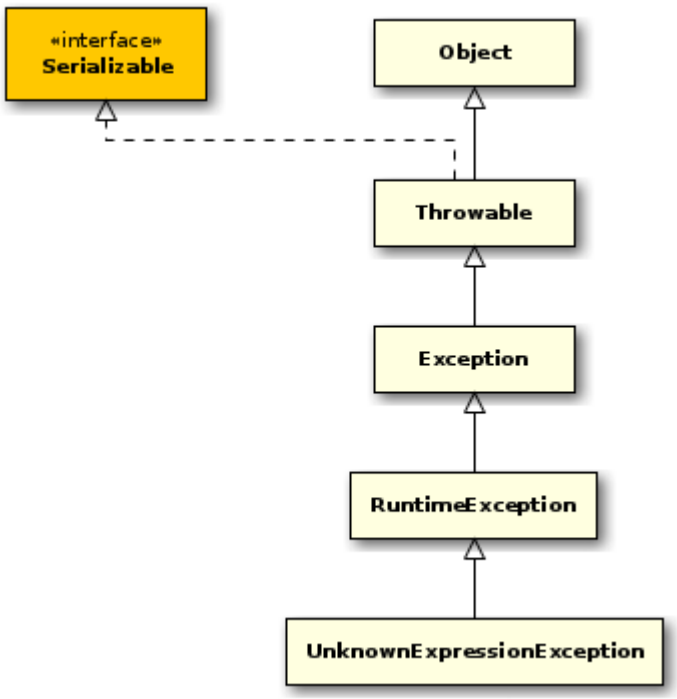
6.50. Klasse TableJoinsInWhereClauseGenerator

```
public class TableJoinsInWhereClauseGenerator extends AbstractFromClauseGenerator
   implements WhereClauseSqlGenerator<QueryData> {

  // Public Constructors
  public TableJoinsInWhereClauseGenerator();

  // Public Methods
  public String fromClause(QueryData queryData,
                           List<QueryNode> alternative,
                           String indent);
  public String fromClauseForNode(QueryNode node);
  public String fromClauseForNode(QueryNode node, boolean leftJoin);
  public Set<String> whereConditions(QueryData queryData,
                                     List<QueryNode> alternative,
                                     String indent);
  public Set<String> whereConditionsForNode(QueryNode node);
}
```



6.51. Ausnahme UnknownExpressionException

```
public class UnknownExpressionException extends RuntimeException {

  // Public Constructors
  public UnknownExpressionException();
  public UnknownExpressionException(String message);
  public UnknownExpressionException(String message, Throwable cause);
  public UnknownExpressionException(Throwable cause);
}
```



6.52. Schnittstelle WhereClauseSqlGenerator

```
public interface WhereClauseSqlGenerator<T> {

  // Public Methods
  public Set<String> whereConditions(T queryData,
                                     List<QueryNode> alternative,
                                     String indent);
}
```

6.53. Schnittstelle WithClauseSqlGenerator

```
public interface WithClauseSqlGenerator<T> {

  // Public Methods
  public List<String> withClauses(T queryData,
                                   List<QueryNode> alternative,
                                   String indent);
}
```

7. Paket annis.sqlgen.annopool

7.1. Klasse ApAnnotateSqlGenerator

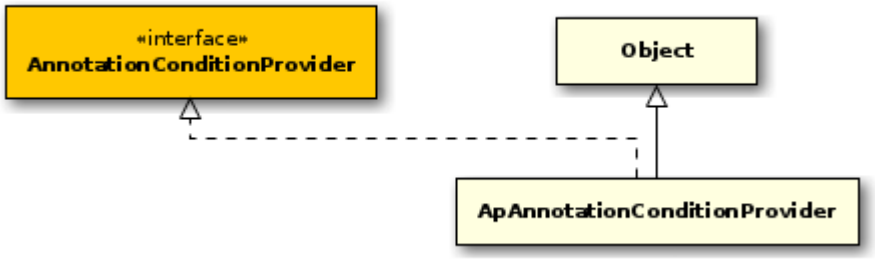
```
public class ApAnnotateSqlGenerator<T> extends AnnotateSqlGenerator<T> {

  // Public Constructors
  public ApAnnotateSqlGenerator();

  // Public Methods
  public String fromClause(QueryData queryData,
                           List<QueryNode> alternative,
                           String indent);
  public String getDocumentQuery(String toplevelCorpusName, String documentName);
  public String getTextQuery(long textID);
  public String selectClause(QueryData queryData,
                             List<QueryNode> alternative,
                             String indent);
}
```

Autor thomas

7.2. Klasse ApAnnotationConditionProvider

```
public class ApAnnotationConditionProvider implements AnnotationConditionProvider {

  // Public Constructors
  public ApAnnotationConditionProvider();

  // Public Methods
  public void addAnnotationConditions(List<String> conditions,
                                      QueryNode node,
                                      int index,
                                      QueryAnnotation annotation,
                                      String table,
                                      QueryData queryData,
                                      TableAccessStrategy tas);
}
```

Autor thomas

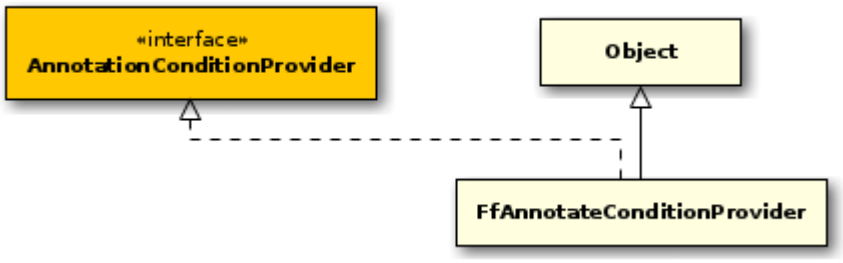


8. Paket annis.sqlgen.fullfacts

8.1. Klasse FfAnnotateConditionProvider

```
public class FfAnnotateConditionProvider implements AnnotationConditionProvider {

  // Public Constructors
  public FfAnnotateConditionProvider();

  // Public Methods
  public void addAnnotationConditions(List<String> conditions,
                                      QueryNode node,
                                      int index,
                                      QueryAnnotation annotation,
                                      String table,
                                      QueryData queryData,
                                      TableAccessStrategy tas);
}
```

Autor thomas



8.2. Klasse FfAnnotateSqlGenerator

```
public class FfAnnotateSqlGenerator<T> extends AnnotateSqlGenerator<T> {

  // Public Constructors
  public FfAnnotateSqlGenerator();

  // Public Methods
  public String fromClause(QueryData queryData,
                           List<QueryNode> alternative,
                           String indent);
  public String getDocumentQuery(String toplevelCorpusName, String documentName);
  public String getTextQuery(long textID);
  public String selectClause(QueryData queryData,
                             List<QueryNode> alternative,
                             String indent);
}
```

Autor thomas

9. Paket annis.utils

9.1. Klasse LegacyGraphConverter

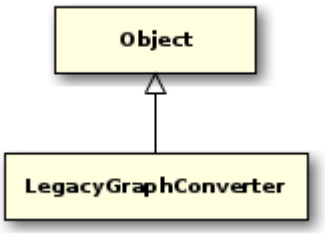
This class can convert the current Salt graph model into the legacy model AOM (Annis Object Model) and "PaulaInline"

9.1.1. Syntax

```
public class LegacyGraphConverter {

  // Public Constructors
  public LegacyGraphConverter();

  // Public Static Methods
  public static List<AnnotationGraph> convertToAOM(SaltProject p);
```

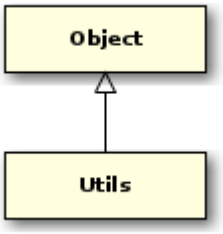
```
  public static AnnotationGraph convertToAnnotationGraph(SDocument document);
  public static AnnotationGraph convertToAnnotationGraph(SDocumentGraph docGraph,
                                                         List<String> matchedIDs);
  public static AnnisResultSet convertToResultSet(SaltProject p);
}
```

Autor



9.2. Klasse Utils

```
public class Utils {

  // Public Constructors
  public Utils();

  // Public Static Methods
  public static String avg(List<Long> runtimeList);
  public static String calculateSHAHash(String s)
    throws NoSuchAlgorithmException,
           UnsupportedEncodingException;
  public static File getAnnisFile(String subpath);
  public static File getAnnisHomeLocation();
  public static String max(List<Long> runtimeList);
  public static String min(List<Long> runtimeList);
  public static Long[] split2Long(String text, char seperator);
}
```



9.2.1. calculateSHAHash(String)

```
  public static String calculateSHAHash(String s)
    throws NoSuchAlgorithmException,
           UnsupportedEncodingException;
```

Hashes a string using SHA-256.

10. Veraltete APIs

10.1. Veraltete Klassen

annis.FindHelper

10.2. Veraltete Schnittstellen

annis.dao.ResultSetConverter

10.3. Veraltete Methoden

queryAnnotationGraph(JdbcTemplate, use `annis.sqlgen.AnnotateSqlGenerator` instead long)

annis.sqlgen.AnnotateSqlGenerator

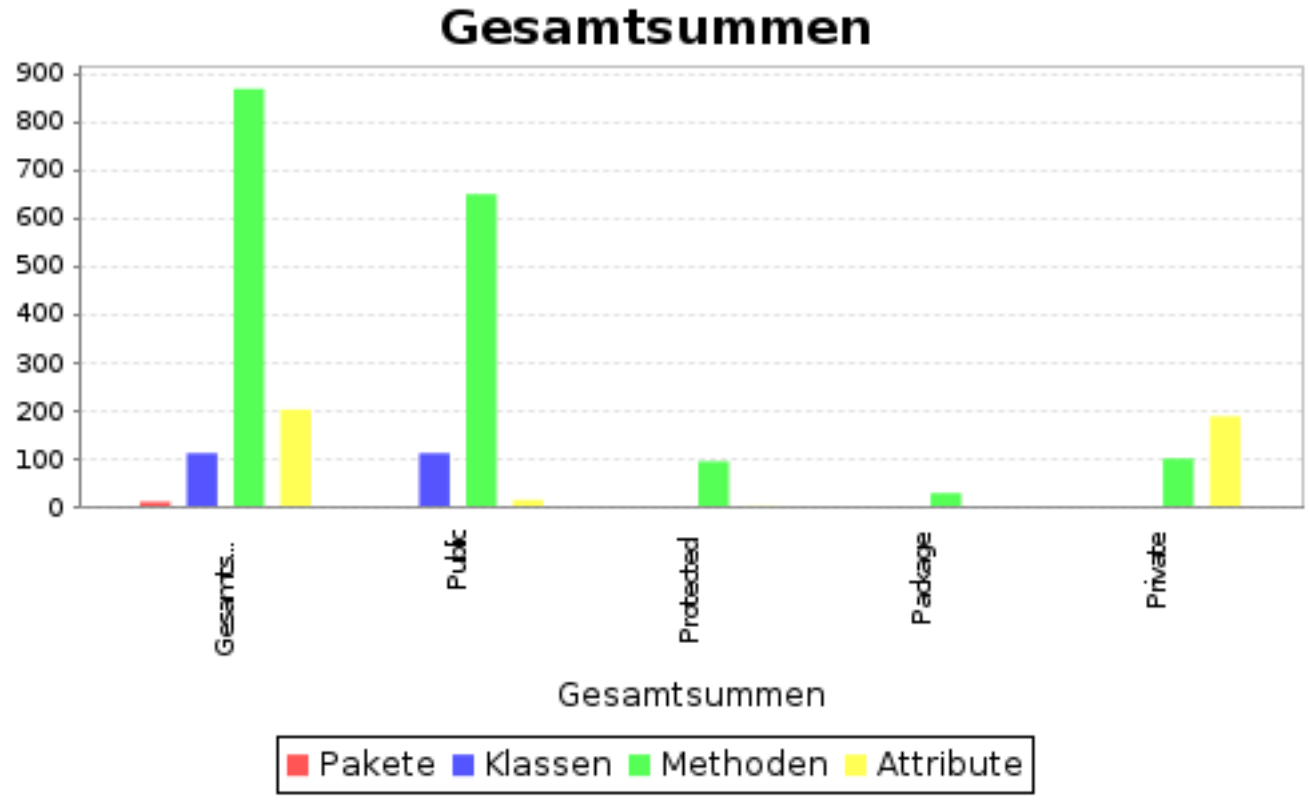
annis.sqlgen.MatrixSqlGenerator

annis.sqlgen.SampleWhereClause

annis.sqlgen.SubcorpusConstraintWhereClause

11. Statistiken

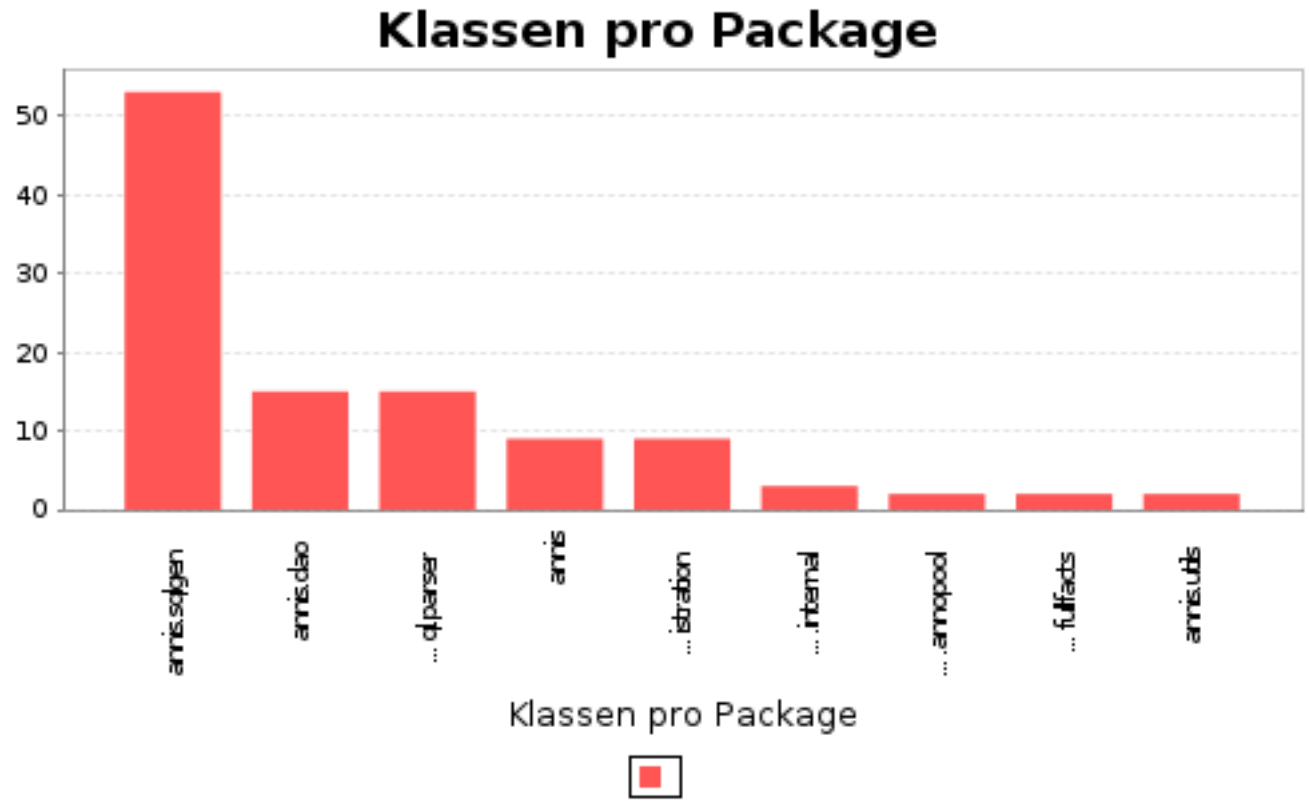
11.1. Gesamtsummen



Gesamtsummen	
Gesamtsummen Pakete	9
Gesamtsummen Klassen	110

Gesamtsummen Attribute	201
Gesamtsummen Methoden	868
Public Klassen	110
Public Attribute	13
Public Methoden	649
Protected Attribute	1
Protected Methoden	93
Package Klassen	0
Package Attribute	0
Package Methoden	27
Private Attribute	187
Private Methoden	99

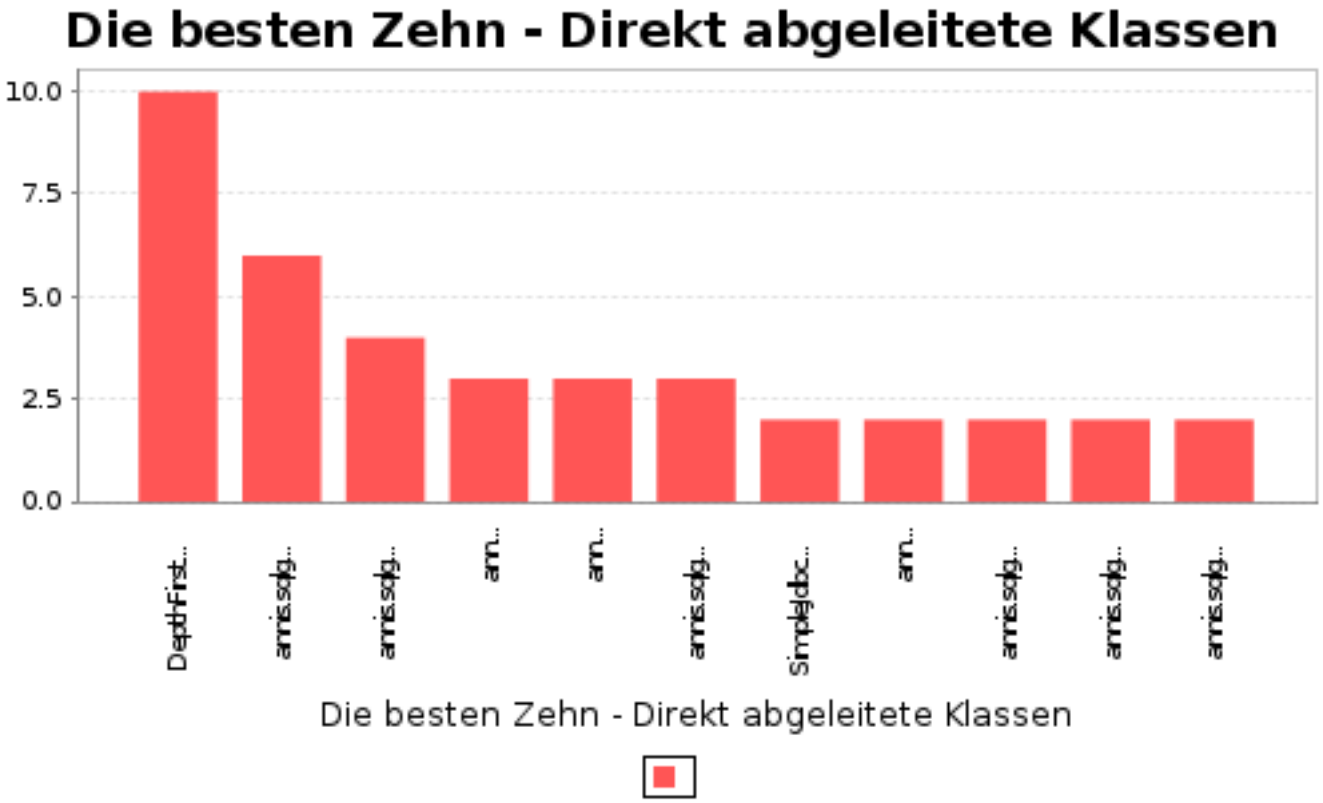
11.2. Klassen pro Package



Klassen pro Package	
annis.sqlgen	53
annis.dao	15
annis.ql.parser	15
annis	9
annis.administration	9
annis.service.internal	3
annis.sqlgen.annopool	2
annis.sqlgen.fullfacts	2

annis.utils	2

11.3. Die besten Zehn - Direkt abgeleitete Klassen



Direkt abgeleitete Klassen	
DepthFirstAdapter	10
annis.sqlgen.TableAccessStrategyFactory	6
annis.sqlgen.AbstractSqlGenerator	4
annis.AnnisBaseRunner	3
annis.AnnisRunnerException	3
annis.sqlgen.AbstractFromClauseGenerator	3
SimpleJdbcDaoSupport	2

annis.administration.DefaultAdministrationDao	2
annis.sqlgen.AbstractSolutionKey	2
annis.sqlgen.AbstractUnionSqlGenerator	2
annis.sqlgen.AnnotateSqlGenerator	2

Stichwortverzeichnis

A

AbstractFromClauseGenerator, 34
AbstractSolutionKey, 35
AbstractSqlGenerator, 35
AbstractUnionSqlGenerator, 36
AbstractWhereClauseGenerator, 37
AdministrationDao, 7
AnnisAdminRunner, 8
AnnisBaseRunner, 1
AnnisDao, 14
AnnisParser, 23, 23
AnnisParser.InternalParser, 23
AnnisRunner, 2
AnnisRunner.OS, 3
AnnisRunnerException, 4
AnnisServiceImpl, 32
AnnisServiceRunner, 33
AnnisWebService, 33
AnnotatedMatch, 15
AnnotatedSpan, 15
AnnotateExtractor, 38
AnnotateInnerQuerySqlGenerator, 38
AnnotateQueryData, 39
AnnotateSqlGenerator, 39
AnnotateSqlGenerator.IslandPolicies, 41
AnnotationConditionProvider, 41
AomAnnotateExtractor, 42
ApAdministrationDao, 9
ApAnnotateSqlGenerator, 68
ApAnnotationConditionProvider, 69
ArrayCorpusPathExtractor, 42

B

ByteHelper, 43

C

calculateSHAHash, 71
Classes
  AbstractFromClauseGenerator, 34
  AbstractSolutionKey, 35
  AbstractSqlGenerator, 35
  AbstractUnionSqlGenerator, 36
  AbstractWhereClauseGenerator, 37
  AnnisAdminRunner, 8
  AnnisBaseRunner, 1
  AnnisParser, 23
  AnnisParser.InternalParser, 23
  AnnisRunner, 2

AnnisRunner.OS, 3
AnnisServiceImpl, 32
AnnisServiceRunner, 33
AnnisWebService, 33
AnnotatedMatch, 15
AnnotatedSpan, 15
AnnotateInnerQuerySqlGenerator, 38
AnnotateQueryData, 39
AnnotateSqlGenerator, 39
AnnotateSqlGenerator.IslandPolicies, 41
AomAnnotateExtractor, 42
ApAdministrationDao, 9
ApAnnotateSqlGenerator, 68
ApAnnotationConditionProvider, 69
ArrayCorpusPathExtractor, 42
ByteHelper, 43
ClauseAnalysis, 24
CommonLimitOffsetGenerator, 44
CorpusAdministration, 9
CorpusPathWhereClauseGenerator, 45
CountExtractor, 16
CountSqlGenerator, 45
CsvCorpusPathExtractor, 46
DefaultAdministrationDao, 11
DefaultTableAccessStrategy, 47
DefaultWhereClauseGenerator, 47
DnfNodeRelNumberUpdater, 25
DnfTransformer, 25
DocumentNameMapRow, 17
FfAnnotateConditionProvider, 69
FfAnnotateSqlGenerator, 70
FindHelper, 5
FindSqlGenerator, 48
LegacyGraphConverter, 70
LimitOffsetQueryData, 50
ListAnnotationsSqlHelper, 50
ListCorpusAnnotationsSqlHelper, 51
ListCorpusByNameDaoHelper, 17
ListCorpusSqlHelper, 51
Match, 18
MatrixSqlGenerator, 52
MemoryUsage, 5
MetaDataAndCorpusWhereClauseGenerator, 52
MetaDataFilter, 18
MultipleColumnsSolutionKey, 53
NodeNameAndIdPostgreSqlArraySolutionKey, 55
NodeRelationNormalizer, 26
NodeRelationNormalizer.RelationCollector, 27
NodeSearchNormalizer, 27
PostgreSqlArraySolutionKey, 56
PreparedStatementCallbackImpl, 13
QueryAnalysis, 27
QueryData, 28
QueryValidator, 29
ResolverDaoHelper, 19
SaltAnnotateExtractor, 57
SaltAnnotateExtractor.RankID, 58
SampleWhereClause, 58

ScriptFileSqlSessionModifier, 19
SearchExpressionCounter, 29
SfAdministrationDao, 13
SpringAnnisDao, 20
SqlConstraints, 61
SubcorpusConstraintWhereClause, 63
SubcorpusConstraintWhereClause.IdPair, 64
SubQueryCorpusSelectionStrategy, 63
TableAccessStrategy, 64
TableAccessStrategyFactory, 65
TableFormatter, 6
TableJoinsInFromClauseSqlGenerator, 65
TableJoinsInWhereClauseGenerator, 66
TimeOutSqlSessionModifier, 22
TokenSearchNormalizer, 30
TreeDumper, 30
Utils, 71
WekaHelper, 7
ClauseAnalysis, 24
CommonLimitOffsetGenerator, 44
CorpusAdministration, 9
CorpusPathExtractor, 44
CorpusPathWhereClauseGenerator, 44
CorpusSelectionStrategy, 16
CountExtractor, 16
CountSqlGenerator, 45
CsvCorpusPathExtractor, 46

D

DatabaseAccessException, 10
DefaultAdministrationDao, 11
DefaultTableAccessStrategy, 47
DefaultWhereClauseGenerator, 47
DnfNodeRelNumberUpdater, 25
DnfTransformer, 25
DocumentNameMapRow, 17

E

Exceptions
AnnisRunnerException, 4
DatabaseAccessException, 10
FileAccessException, 12
UnknownExpressionException, 31, 67
UsageException, 6
extractCorpusPath, 43, 44, 46

F

FfAnnotateConditionProvider, 69
FfAnnotateSqlGenerator, 70
FileAccessException, 12
FindHelper, 5
FindSqlGenerator, 48
FromClauseSqlGenerator, 49

G

generateInnerQueryColumns, 60
generateOuterQueryColumns, 54, 56, 60

getCurrentKeyAsString, 60
getDocumentsForMetadata, 18
getKeyColumns, 54, 57, 60
getMatchedNodeIndex, 60
getNodeId, 60
GroupByClauseSqlGenerator, 49

I

Interfaces
  AdministrationDao, 7
  AnnisDao, 14
  AnnotateExtractor, 38
  AnnotationConditionProvider, 41
  CorpusPathExtractor, 44
  CorpusSelectionStrategy, 16
  FromClauseSqlGenerator, 49
  GroupByClauseSqlGenerator, 49
  LimitOffsetClauseSqlGenerator, 50
  NodeSqlAdapter, 55
  OrderByClauseSqlGenerator, 56
  ResultSetConverter, 19
  SelectClauseSqlGenerator, 59
  SolutionKey, 59
  SqlGenerator, 62
  SqlSessionModifier, 22
  WhereClauseSqlGenerator, 68
  WithClauseSqlGenerator, 68
isFalse, 62
isNewKey, 61
isNotNull, 62
isNull, 62
isTrue, 62

L

LegacyGraphConverter, 70
LimitOffsetClauseSqlGenerator, 49
LimitOffsetQueryData, 50
ListAnnotationsSqlHelper, 50
ListCorpusAnnotationsSqlHelper, 51
ListCorpusByNameDaoHelper, 17
ListCorpusSqlHelper, 51
logMemoryUsage, 6

M

Match, 18
MatrixSqlGenerator, 52
MemoryUsage, 5
MetaDataAndCorpusWhereClauseGenerator, 52
MetaDataFilter, 18
Methods
  AnnisParser, 23
  calculateSHAHash, 71
  extractCorpusPath, 43, 44, 46
  generateInnerQueryColumns, 60
  generateOuterQueryColumns, 54, 56, 60
  getCurrentKeyAsString, 60
  getDocumentsForMetadata, 18

getKeyColumns, 54, 57, 60
getMatchedNodeIndex, 60
getNodeId, 60
isFalse, 62
isNewKey, 61
isNotNull, 62
isNull, 62
isTrue, 62
logMemoryUsage, 6
queryAnnotationGraph, 41
retrieveKey, 54, 57, 61
sayHello, 32, 34
shutdown, 33
MultipleColumnsSolutionKey, 53

N

NodeNameAndIdPostgreSqlArraySolutionKey, 55
NodeRelationNormalizer, 26
NodeRelationNormalizer.RelationCollector, 27
NodeSearchNormalizer, 27
NodeSqlAdapter, 55

O

OrderByClauseSqlGenerator, 55

P

PostgreSqlArraySolutionKey, 56
PreparedStatementCallbackImpl, 13

Q

QueryAnalysis, 27
queryAnnotationGraph, 41
QueryData, 28
QueryValidator, 29

R

ResolverDaoHelper, 19
ResultSetConverter, 19
retrieveKey, 54, 57, 61

S

SaltAnnotateExtractor, 57
SaltAnnotateExtractor.RankID, 58
SampleWhereClause, 58
sayHello, 32, 34
ScriptFileSqlSessionModifier, 19
SearchExpressionCounter, 29
SelectClauseSqlGenerator, 59
SfAdministrationDao, 13
shutdown, 33
SolutionKey, 59
SpringAnnisDao, 20
SqlConstraints, 61
SqlGenerator, 62
SqlSessionModifier, 22
SubcorpusConstraintWhereClause, 63
SubcorpusConstraintWhereClause.IdPair, 64

SubQueryCorpusSelectionStrategy, 63

T

TableAccessStrategy, 64
TableAccessStrategyFactory, 65
TableFormatter, 6
TableJoinsInFromClauseSqlGenerator, 65
TableJoinsInWhereClauseGenerator, 66
TimeOutSqlSessionModifier, 22
TokenSearchNormalizer, 30
TreeDumper, 30

U

UnknownExpressionException, 31, 67
UsageException, 6
Utils, 71

W

WekaHelper, 7
WhereClauseSqlGenerator, 68
WithClauseSqlGenerator, 68