| | |
|---|---|
| | initWithEntity:qualifierFormat: |
| Combining qualifiers | conjoinWithQualifier:<br>disjoinWithQualifier:<br>negate |
| Getting the entity | entity |
| Checking the definition | isEmpty |
| Setting distinct selection | setUsesDistinct:<br>usesDistinct |

destinationEntity (EORelationship)

destinationEntity (EORelationship)

(void)conjoinWithQualifier:(EOQualifier *)aQualifier

Replaces the contents of the receiver with the conjunction (logical AND) of the existing contents a
example, this code excerpt creates a qualifier that selects employees whose salaries are between 40

example, this code excerpt creates a qualifier that selects all employees in both the Personnel and
departments:

(EOEntity *)entity

Returns the entity that defines the context for attributes and relationships. Any attribute named in t
belong to this entity, and data paths must begin with a relationship defined in this entity.

initWithEntity:(EOEntity *)anEntity qualifierFormat:(NSString *)format, ...

Initializes a newly allocated EOQualifier to be rooted in anEntity and to build itself from a format
printf()-style format string as presented in the class description. This is the designated initializer fo
Returns self if format is successfully parsed, nil otherwise.

(BOOL)isEmpty

Returns YES if the receiver selects all rows for its entity (that is, if it has no qualifying informatio
Returns NO otherwise.

(void)negate

Negates the receiver's qualifying information.

(void)setUsesDistinct:(BOOL)flag

Sets according to flag whether the receiver eliminates duplicates from the results of a select operat
operation would yield the same row many times, a qualifier that uses distinct results eliminates all
one is retrieved.