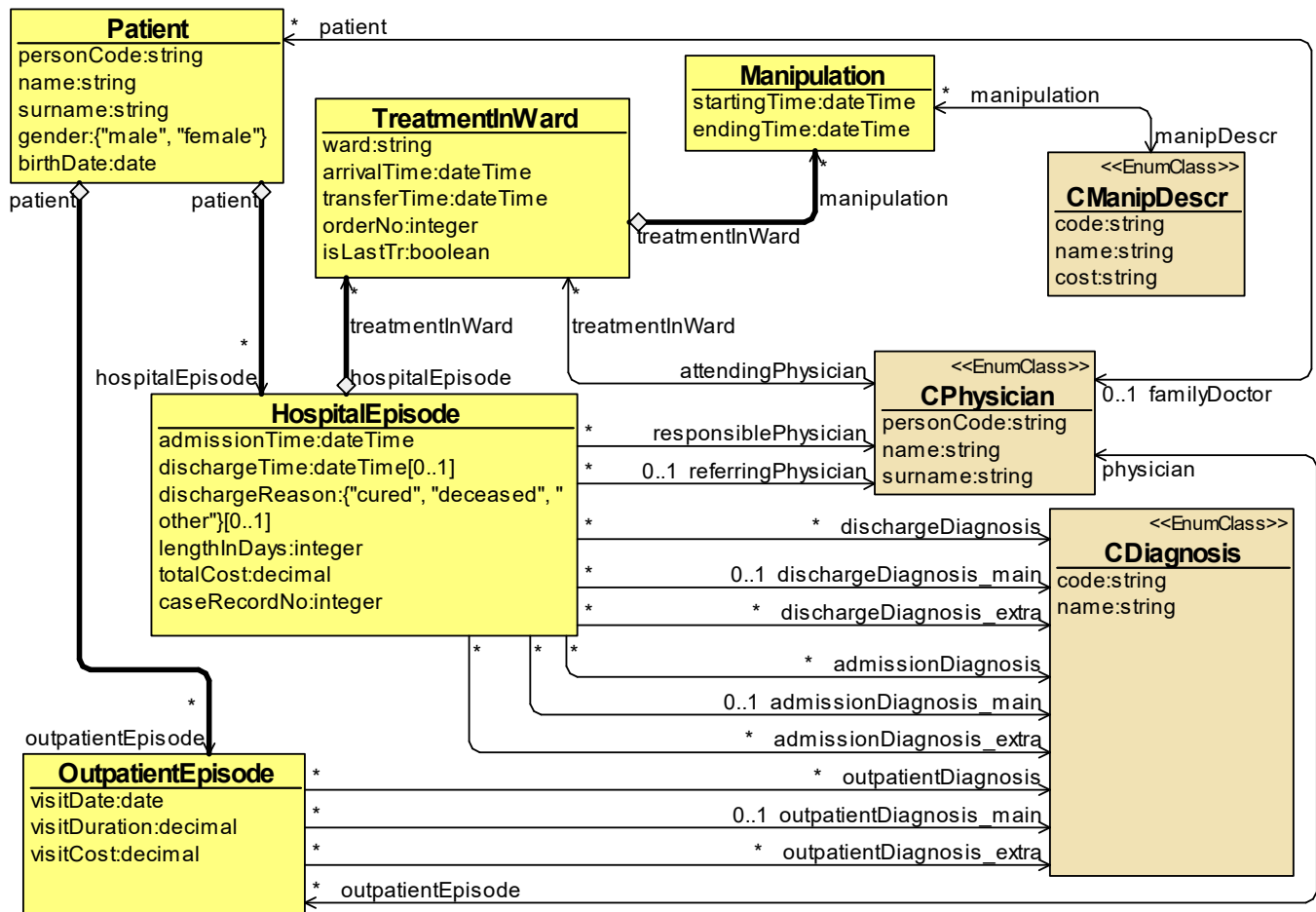


# Visual Query Examples

## Mini-Hospital data schema

The visual query environment needs a data schema defining the available class and property vocabulary, as well as applicability of properties to classes and cardinalities.

The following visualization shows a mini-hospital data schema used in further examples.



The center of the data schema is the *Patient* class; each patient can have a number of hospital episodes (*HospitalEpisode*), and a number of outpatient episodes (*OutpatientEpisode*). There can be a number of treatments in wards (*TreatmentInWard*) for each hospital episode, numbered starting by 1 in the *orderNo* attribute. The attribute *isLastTr* (values "true" or "false") marks each treatment in ward as the last one or non-last one for its episode. Further on, every treatment in ward can have a number of associated manipulations (*Manipulation*).

There are cardinality marks (e.g. [0..1] or \*) at some attributes or links. The attributes or links not having any cardinality mark are assumed to have cardinality 1 that means existence of exactly one required value corresponding to the attribute or the link.

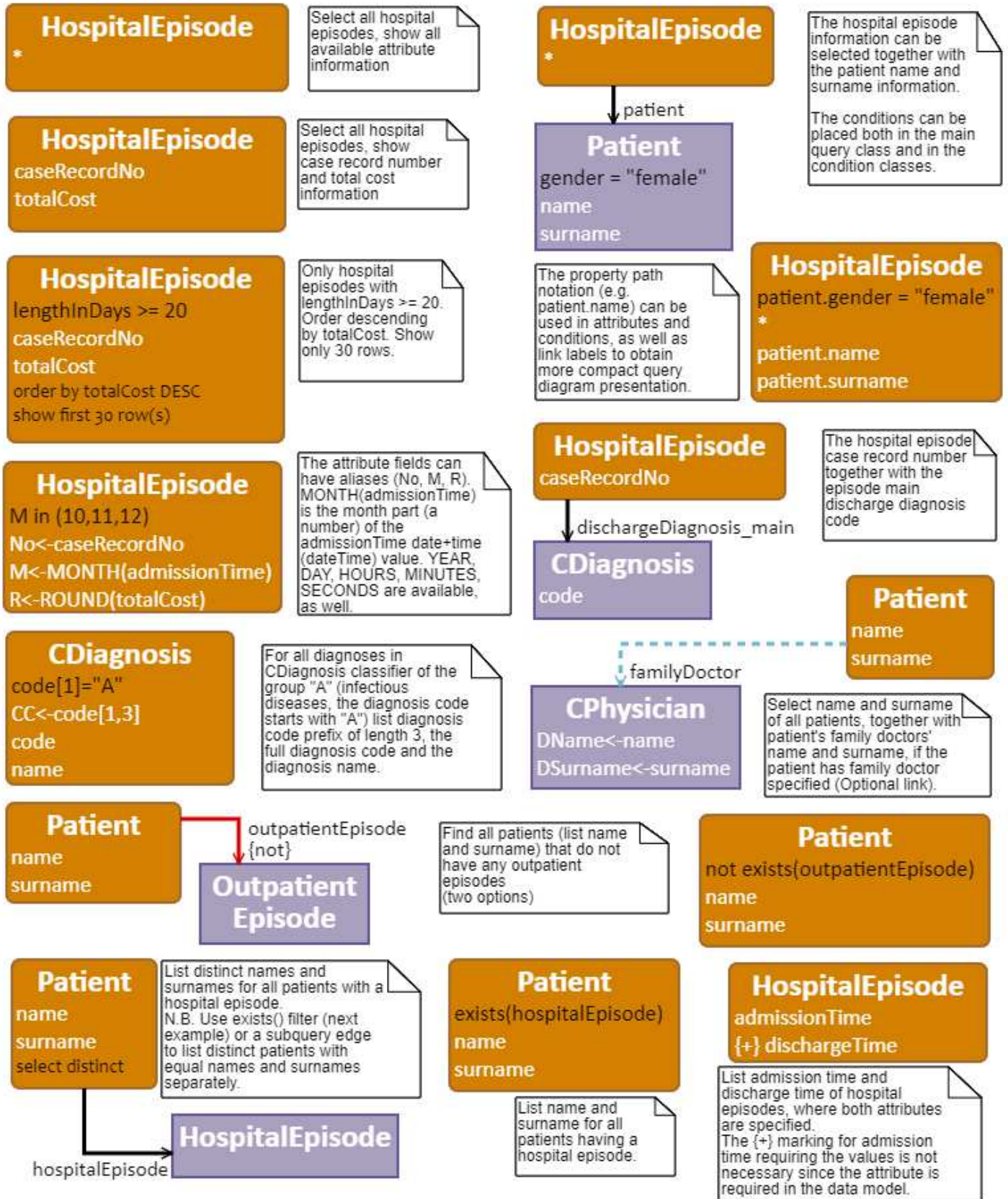
The classes *CManipDescr*, *CPhysician* and *CDiagnosis* contain classifier values of manipulation descriptions, physicians and diagnoses respectively. There are links to the classifier classes from the main data classes. In the query building process these links can be traversed both in the direction from the data class to the classifier class, as well as in the reverse direction.

For every hospital episode there can be several admission and several discharge diagnoses specified; one of these diagnoses is marked as the main admission or discharge diagnosis respectively (accessible from the episode by link *admissionDiagnosis\_main* or *dischargeDiagnosis\_main*), all episode diagnoses that are not main admission or discharge diagnoses are extra diagnoses (accessible from the episode by the link *admissionDiagnosis\_extra* or *dischargeDiagnosis\_extra*). The links *admissionDiagnosis* and *dischargeDiagnosis* can be used to reach all admission or discharge diagnoses (including the main ones and the extra ones) for an episode, respectively).

Similarly, for an outpatient episode the *outpatientDiagnosis* link lists all diagnoses, *outpatientDiagnosis\_main* lists the main diagnosis and *outpatientDiagnosis\_extra* lists the extra diagnoses.

## Example Queries

### Class-attribute-link-condition queries



Simple and attribute-based statistics

count(.)  
**OutpatientEpisode**

Count all outpatient episodes

count(.)  
**TreatmentInWard**  
ward

For each ward, count the number of treatments in the ward

count(.)  
**OutpatientEpisode**  
visitCost >= 50

patient → **Patient**  
gender = "male"  
Count outpatient episodes, costing at least 50 Euro, of male patients

count(.)  
avg(visitDuration)  
**OutpatientEpisode**  
M<-month(visitDate)  
order by M

For each month count and calculate the average duration of outpatient episodes in the month. Order by the month.

count(.)  
**OutpatientEpisode**  
visitCost >= 50

patient → **Patient**  
gender  
Count outpatient episodes, costing at least 50 Euro, grouped by patient's gender

count(.)  
**Patient**  
exists(hospitalEpisode)  
not exists(outpatientEpisode)

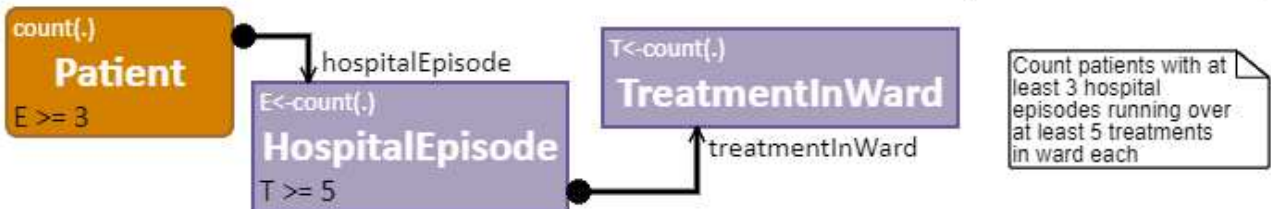
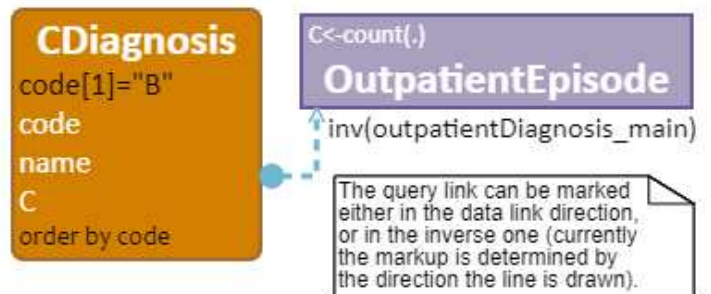
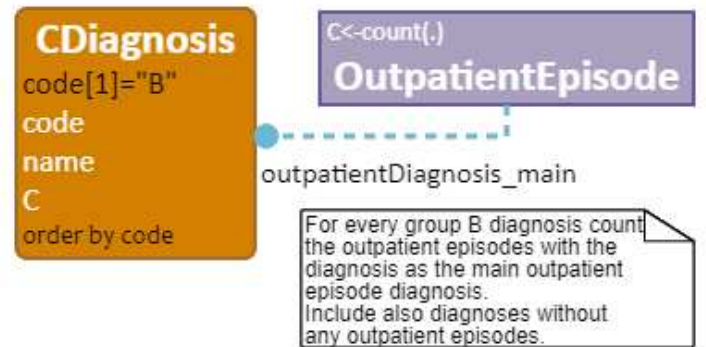
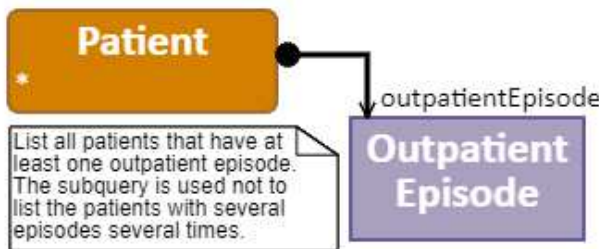
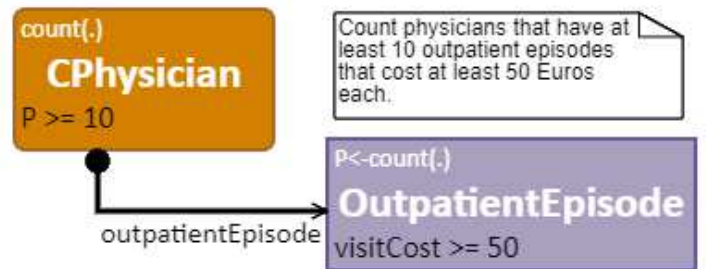
Count (distinct!) patients having a hospital episode and not having an outpatient episode

count\_distinct(.)  
**Patient**

hospitalEpisode → **Hospital Episode**

outpatientEpisode {not} → **Outpatient Episode**

Subqueries: Statistics around data items (class instances)



Advanced examples: expressions, condition links, free links and free nodes

