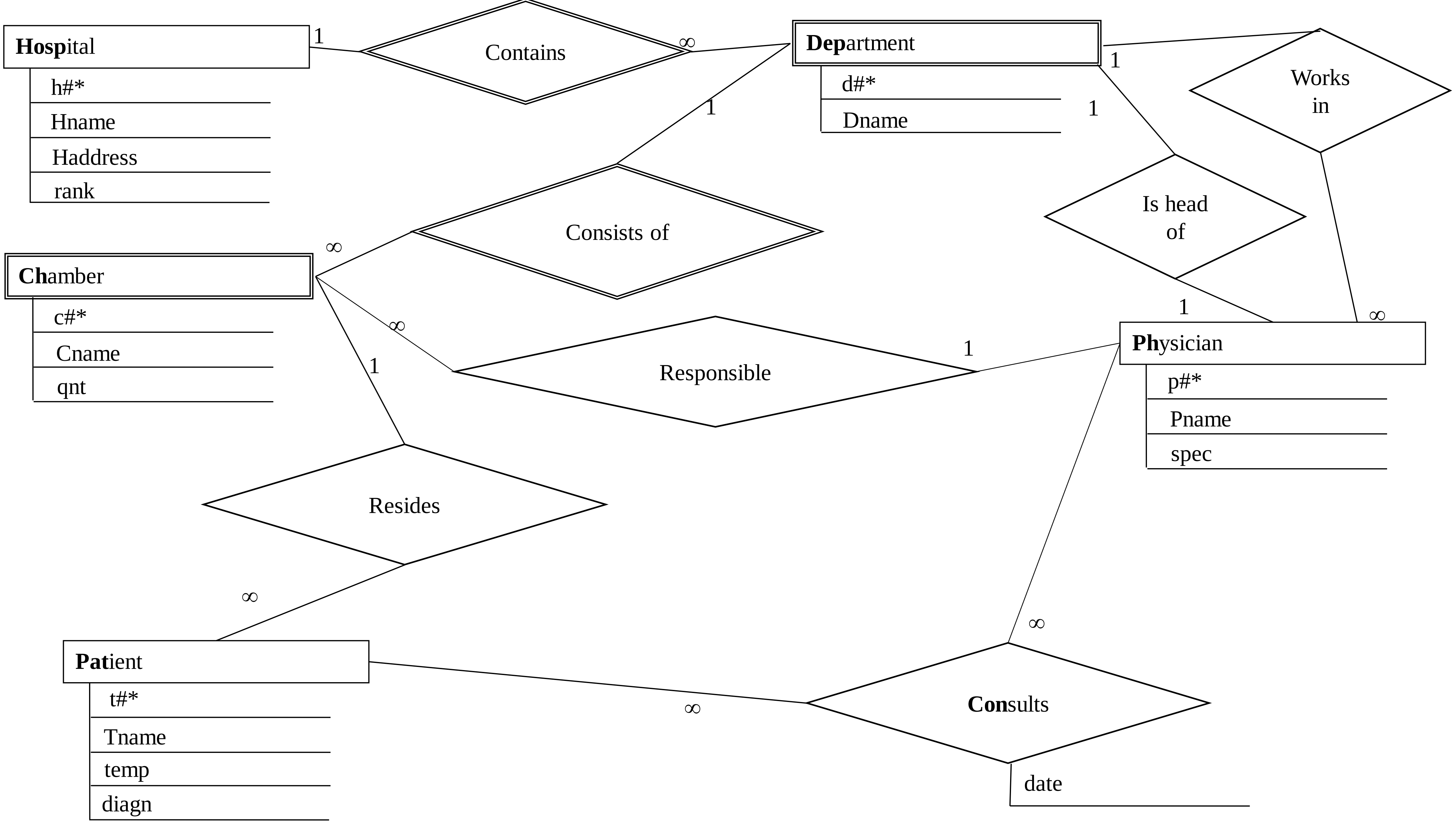
Hybrid technologies for Databases

Anatoly V. Anisimov, Igor O. Zavadskiy, Petro P. Kuliabko


Hospital
h#*
Hname
Haddress
rank
Contains
Department
d#*
Dname
Works
in
Is head
of
Consists of
Chamber
c#*
Cname
qnt
Responsible
Physician
p#*
Pname
spec
Resides
Patient
t#*
Tname
temp
diagn
Consults
date
1
∞

Description of the data domain according to the relational approach

Hospital(h#, Hname, Hadress, rank); **Dep**artment(d#, h#, Dname, Dmng);

Chamber(c#, d#, h#, Cname, qnt); **Ph**ysician(p#, d#, h#, Pname, spec, age);

Patient(t#, c#, d#, h#, Tname, diagn, temp); **Con**sulting(r#, p#, t#, data);

Record Name is Hosp {
H# CHAR(5),
HNAME CHAR(25),
Hadress CHAR(30),
rank CHAR(7) };
	Record Name is Dep {
	D# CHAR(5),
	DNAME CHAR(25),
	Dmng CHAR(30)
	};

Record Name is Ch{
C# CHAR(5),
CNAMECHAR(25),
Qnt integer };
	Record Name is Ph {
	P# CHAR(5),
	PNAME CHAR(25),
	spec CHAR(20),
	age integer
	};

Record Name is Pat{ T# CHAR(5), TNAME CHAR(25), diag CHAR(40), temp float };	Record Name is Con { R# CHAR(5), Data DATE};

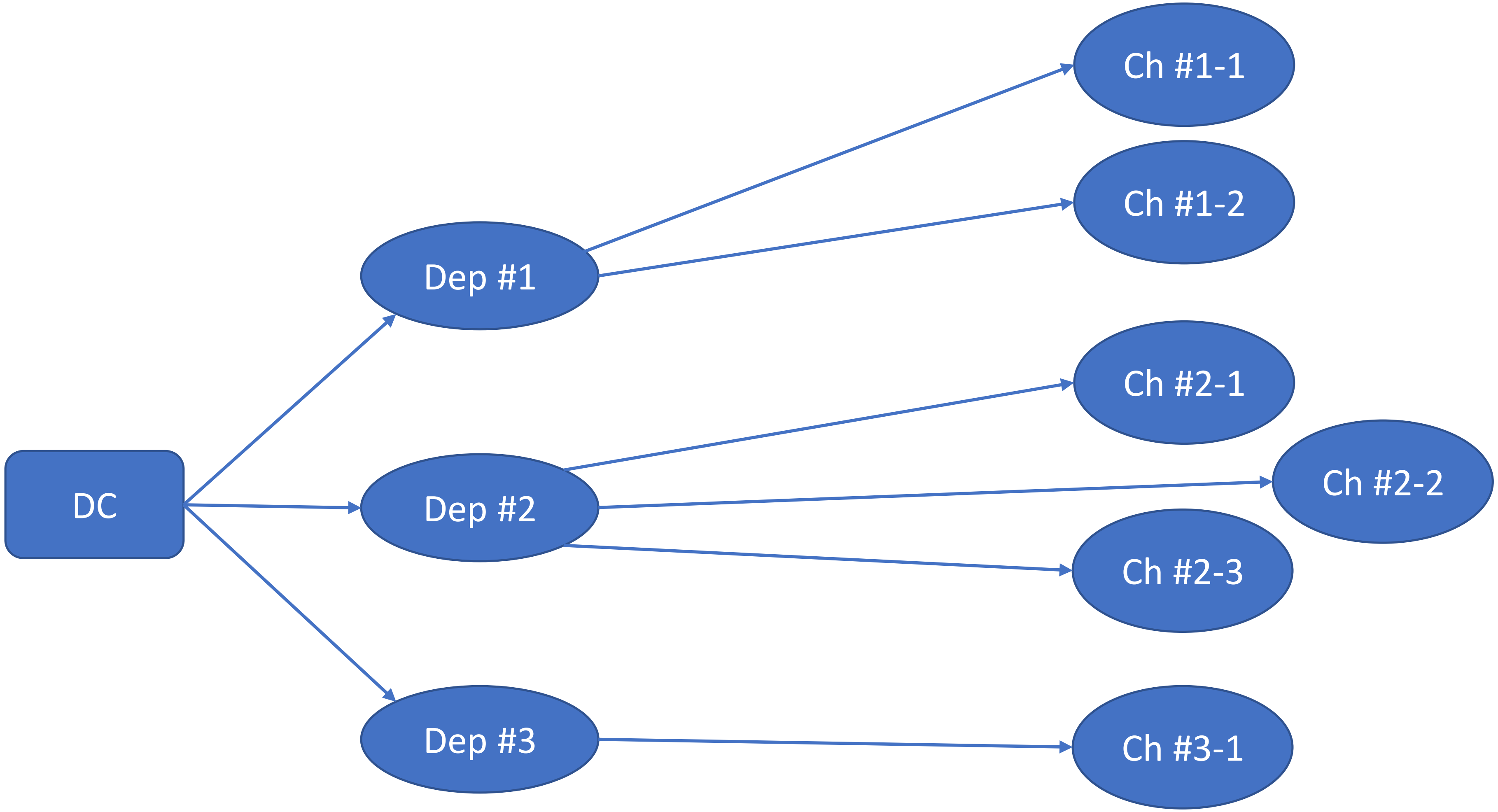
Set HD Owner is Hosp Member is Dep;	Set DC Owner is Dep Member is Ch;	Set CP Owner is Ch Member is Pat;
Set DP Owner is Dep Member is Ph;	Set PhCon Owner is Ph Member is Con;	Set PC Owner is Ph Member is Ch;
Set PCon Owner is Pat Member is Con;		

Data set as Member

Set HDC Owner is Hosp Member is DC;	Set HDP Owner is Hosp Member is DP;	Set CPCon Owner is Ch Member is PCon;
Set HCP Owner is Hosp Member is CP through DC;	Set CPCon Owner is Ch Member is PCon;	Set PhCP Owner is Ph Member is CP;

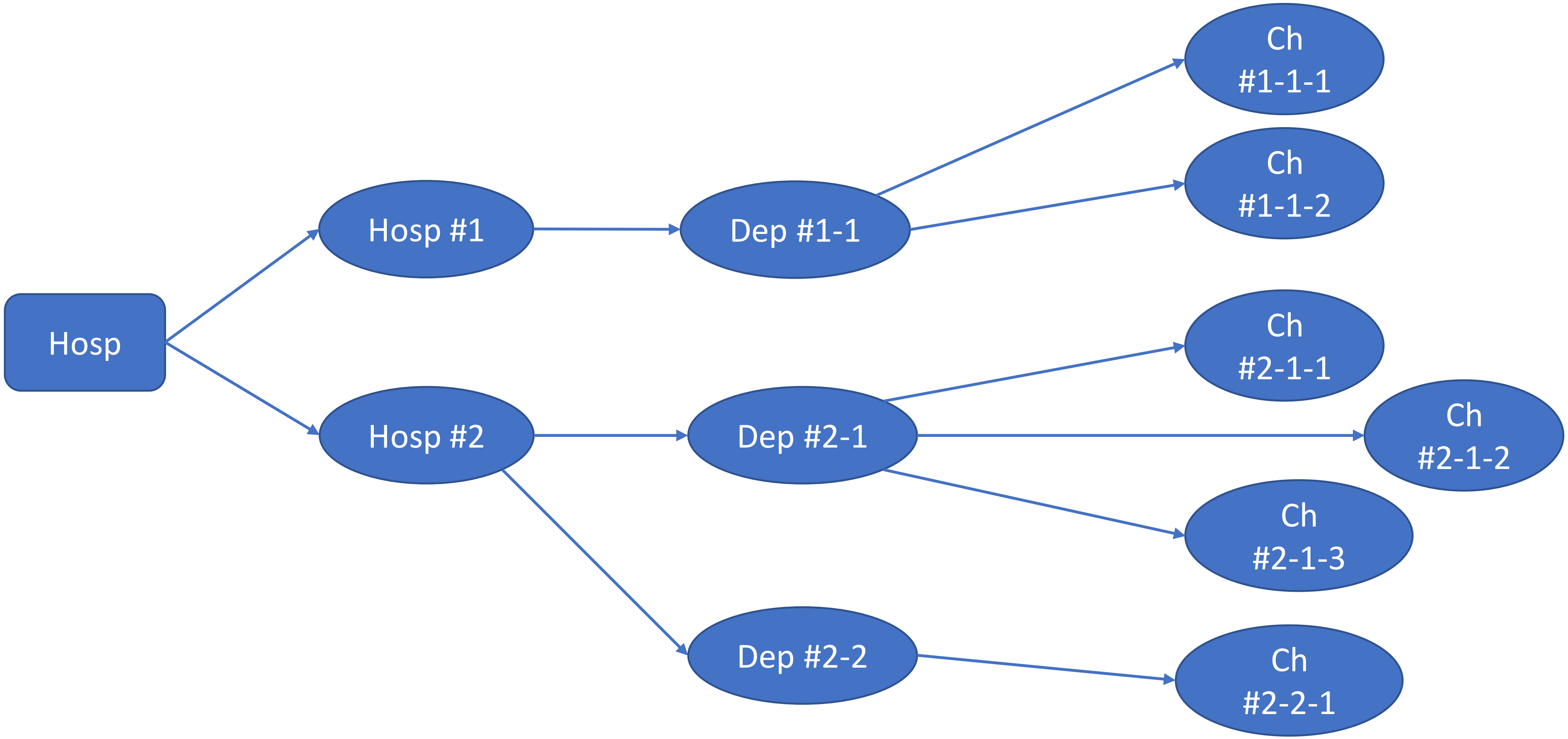
DataSet example

Set DC
Owner is Dep
Member is Chamber;



DataSet as a member of another DataSet

Set HDC
Owner is Hosp
Member is DC;



The operations of the supplement algebra

Operations UNION(U), INTERSECT(∩), DIFFERENCE(\), TIMES(×) and PROJECT

UNION(DS1, DS2) -> DS3;

INTERSECT(DS1, DS2) -> DS3;

DIFFERENCE(DS1, DS2) -> DS3;

TIMES(DS1, DS2) -> DS3; DS3. OWNER = DS1. OWNER$\otimes$DS2. OWNER;

PROJECT (DS1, [Olist], [Mlist]) -> DS2;

Operations of filtration

BFilter(DS1, *condition*) -> DS2;

ExistsFilter(DS1,DS2,*condition*)-> DS3;

AllFilter(DS1,DS2, *condition*)-> DS3;

SetFilter (DS1, DS2, AOlist, BOlist Θ COlist) -> DS3;
ΘAOlist and BOlist are attribute lists from DS1, and AOlist is attribute list from DS1.OWNER and BOlist is attribute list from DS1.Member;
COlist - list of attributes from DS2. Member;
Θ is a binary predicate of set comparison: ⊆, ⊂, ⊇, ⊃, =, ≠.

The order of actions is such:
a) a set of instances of DS1.OWNER is grouped by AOlist;
b) DS2.Member is projected by C;
c) from the groups $\{R1_{AOlist}\}$ are selected those instances for which the condition BOlist Θ COlist is fulfilled
d) into the resulting DS3 passes only those instances of DS1 that meet the specified condition.

Some other operations

JOIN(DS1, DS2, *condition*) -> DS3; D (DS1. OWNER) = D (DS3. OWNER) & D (DS2.Member) = D (DS3. Member) & D (DS1.Member) = D (DS2.OWNER)

JOIN*(DS1, DS2, *condition*) -> DS3 ; Such type of operation will be appropriate when there is another DS on the hierarchical path between DS1 and DS2;

its OWNER will be the same type as DS1.Member, and its Member will be the same type as DS2.OWNER, then we will get DS3, such as D(DS1.OWNER) = D(DS3.OWNER), and D(DS2.Member) = D(DS3.Member).

JOINMember (DS1, DS2, *condition*) -> R; Another one specific type of connection, oriented on the situation when DS1 and DS2 have a common Member, which implements a connection between two objects of the type "many-to-many" (ER-model). The result will be a relationship that is DS1.OWNER and satisfies the *condition*.

COMPOSE(REC1, REC2, *condition*) -> DS1; REC1 and REC2 are the records from which DS1 is formed.

ADDMember(DS, REC, *condition*)

Query examples

Find hospital names and address for the hospitals with rank = “xxx”.

F_β(Hosp, Rank = “xxx”) -> R1; R1[Hname, Hadress] -> res;	BFilter(HD, Rank = “xxx”) -> DS1; PROJECT-OWNER(DS1,[Hname, Hadress]) -> res;

For such simple query proposed algebra has no benefits in comparison with traditional relation algebra.

Find department names and hospital number in the hospitals with rank = “xxx”.

F_β(Hosp, Rank = “xxx”) -> R1; F∃ (Dep, R1,Dep.h# = R1.h#) -> R2; R2[Dep.h#, Dname] ->res;	BFilter(HD, Rank = “xxx”& NotEmptyMember) -> DS1; PROJECT (DS1, [h#], [Dname]) ->res

Supplementary algebra uses one operation less than relational algebra.

Find the patient names with a temperature greater than 37 from hospital № 1, department №3, chamber №6.

F_β(Dep, h# = 1) -> R1; F∃ (Ch, R1, Ch.c# = 6 & Ch.d# = R1.d#) ->R2; F∃ (Pat, R2, Pat.temp >37 & Pat.c# = R2.c#) -> R3; R3[Tname] -> res;	JOIN (HD, DC, HD.H# = 1 & C# = 6) -> DS1; JOIN (DS1, CP, temp >37) ->DS2; PROJECT (DS2, [], [Tname]) -> res; /* or the same with DS HCP */ BFilter(HCP, temp>37 & h#=1 & d#=3 & c#=6) ->DS2;

Relational algebra uses 4 operations, while proposed algebra uses 3 operations, even 2 if it will work through data set HCP.

Find the physician names who consult at least all patients in chamber №6 of hospital №1 and department “XXX”.

(F_β(Dep, h# =1 & Dname = «XXX») -> R1; F∃ (Ch, R1, c# =6 & Ch.d# = R1.d#) -> R2; F∃ (Pat, R2, Pat.c#=R2.c#)-> R3; F_S(Consulting, R3, p#, Consulting.t# Ê R3.t#) -> R4; F∃ (Ph, R4, Ph.p# = R4.p#)-> R5; R5[Pname] -> res;	BFilter(HD, h# =1 & Dname = XXX») -> DS1; JOIN(DS1, DC, c# = 6) -> DS2; JOIN(DS2, CP, TRUE) -> DS3; JOIN(DS3, PCon, TRUE) -> DS4; /* JOIN*(DS2,PCon,TRUE) -> DS4; Join over the hierarchical path */ SetFilter(PhCon, DS4, Ph.p#, PhCon.Con.r# Ê DS4.Con.t#) ->DS5; PROJECT-OWNER(DS5, [PNAME]) ->res;

This query is classified as complex because it needs to perform the set comparison.

Query examples

Find the patient names consulted by Dr. Sidorchuk.	
F_{β}(Ph, Pname = “ Sidorchuk ”) -> R1; F∃ (Con,R1,Con.p#=R1.p#)->R2; F∃ (Pat,R2, Pat.t#=R2.t#) ->R3; R3[Tname] -> res;	JOINMember(PCon,PhCon, PCon.Con.r# = PhCon.Con.r# & PhCon.Ph.Pname = “ Sidorchuk ”) -> R1; R1[Tname] -> res;
This query is performed on the base of connection of “many-to-many” type.	

From the given examples it is seen that notations of some queries in terms of the proposed algebra look more compact.