Acyclic schemata of databases

S. V. Zykin
OB IM SB RAS

Database relations

List of groups

Group's number	Group's code	Speciality's number
1	M-210	1
2	M-220	2

Students

Student's number	Group's number	Student's name
1	1	Ivanov I.I.
1	2	Sidorov S.S.
2	1	Petrov P.P.
2	2	Kovalev K.K.

Subjects

Subject's number	Name of subject
1	Mathematics
2	Physics

Marks

Student's number	Group's number	Subject's number	Mark
1	1	1	5
1	2	1	4
1	2	2	3
1	2	3	5

Specialities

Speciality's number	Speciality's name
1	Programming
2	Pedagogy

Educational plan

Subject's number	Speciality's number	Quantity of hours
1	1	40
1	2	30
2	1	30
2	2	40

New tuple: t=(M-210, Ivanov I.I., Physics, 4)

Acyclic databases

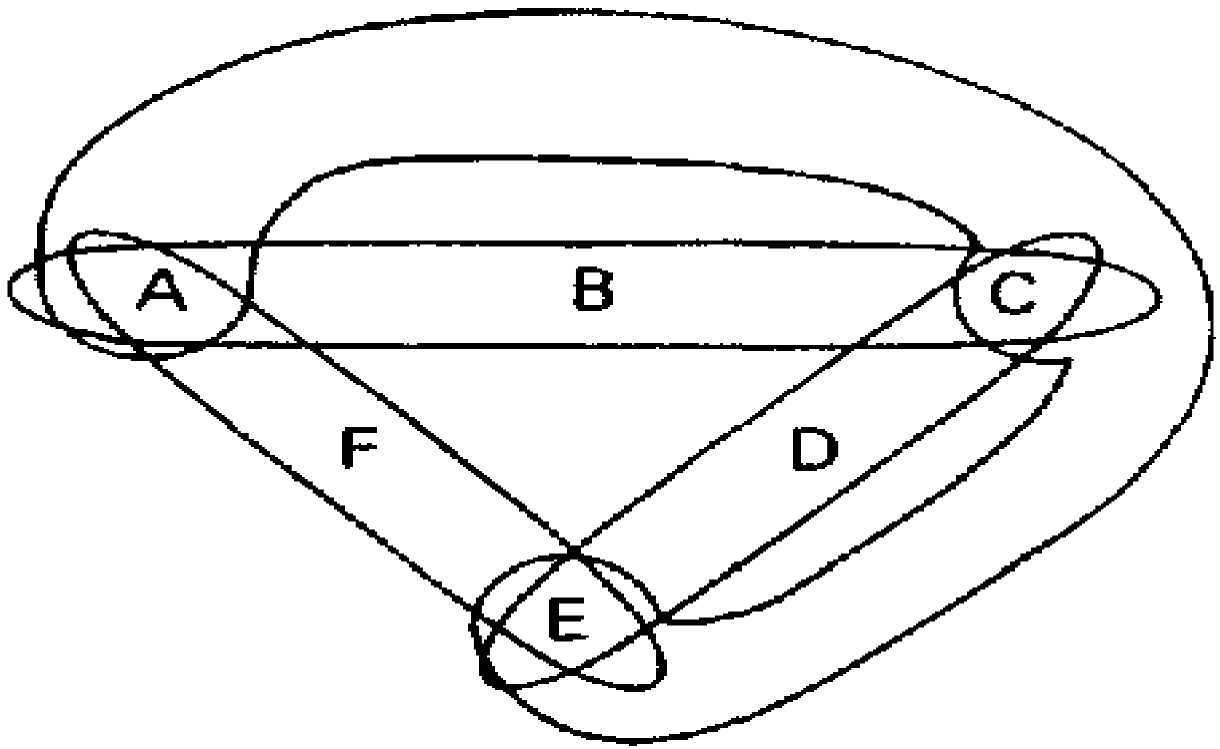
Condition 1**.** The relations set is acyclic, if acyclic hypergraph associated with it.

Definition. The hypergraph of a relations set $R_1,\ldots,R_n$ has as its set of nodes N those attributes that appear in one or more of the R_i's, and as its set of edges $E=\{R_1,\ldots,R_n\}$.

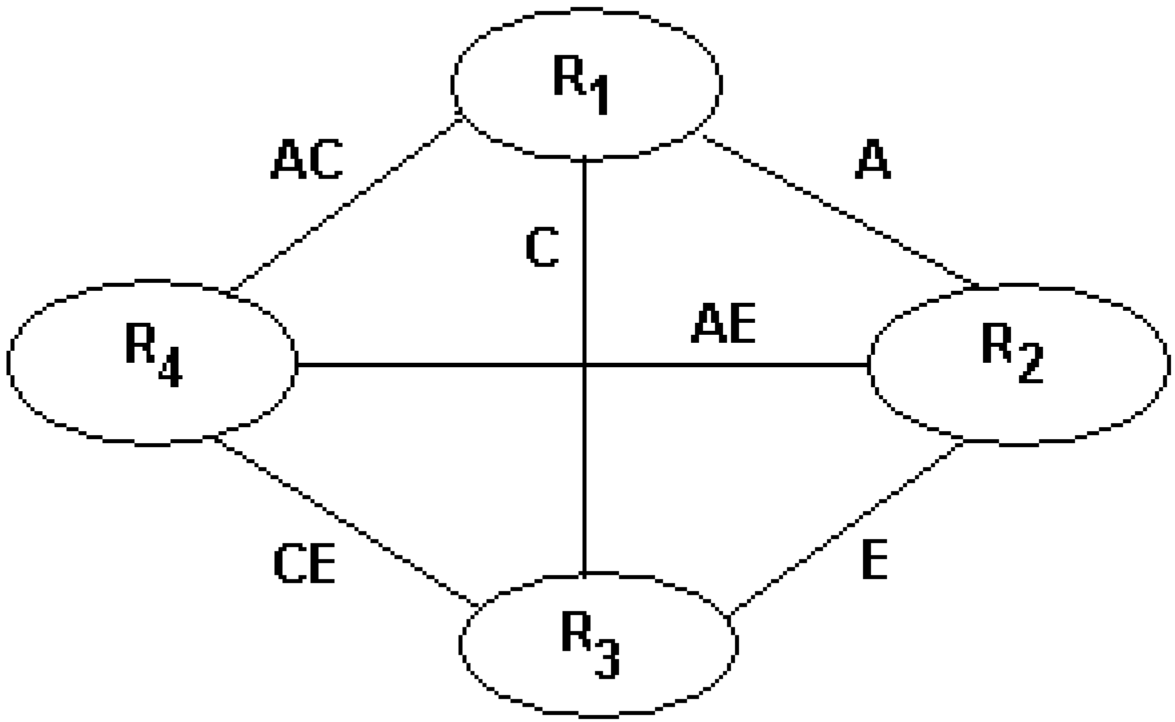
Definition. Let (N,E) be a hypergraph. Its reduction (N,E') is obtained by removing from E each edge that is a proper subset of another edge.

Hypergraph example

$R_1(A,B,C)$, $R_2(A,F,E)$,
$R_3(C,D,E)$, $R_4(A,C,E)$



Intersections graph



Condition 2. The relations set is acyclic, if it has a join tree in intersections graph.

Inclusion dependences

Definition. Let $R_i[A_1, \ldots, A_m]$ and $R_j[B_1, \ldots, B_p]$ – schemes of relations (not necessarily various), $V \subseteq \{A_1, \ldots, A_m\}$ and $W \subseteq \{B_1, \ldots, B_p\}$, $|V|=|W|$, than object $R_i[V] \subseteq R_j[W]$ called as inclusion dependence, where $|V|$ – capacity of set V.

$V=W$ – typed inclusion dependence,

$R_i[V]$ – projection of the relation R_i on attributes V.

Relationship on database scheme

$PK(R_i)$ – primary key R_i,

$L_1(i,j)$ – relationship **1:1** from R_i to R_j,

$L_M(i,j)$ – relationship **1:M** from R_i to R_j,

Definition. There is the relationship $L_1(i,j)$ between R_i and R_j, if $PK(R_i)=PK(R_j)$ and $R_j[V]\subseteq R_i[V]$ for any realizations R_i and R_j is true, where $V=[R_i]\cap[R_j]$, $[R_i]$ is the attributes set of the relation R_i.

Definition. There is the relationship $L_M(i,j)$ between relations R_i and R_j, if $PK(R_i)\neq PK(R_j)$ and $PK(R_i)\subseteq[R_j]$.

New definition of acyclic database

Definition. The set of relations B will be named as acyclic, if not exist an ordered the relations subset

$\{R_{m(1)}, R_{m(2)}, \ldots, R_{m(s)}\} \subseteq \mathfrak{R}$ and $L(m(1),m(2))$, $L(m(2),m(3))$,…, $L(m(s-1),m(s))$, $L(m(s),m(1))$, $s > 1$.

Theorem. If the relation set is cyclic, then associated with it hypergraph also will be cyclic. (inverse statement is not correct).

Unstructural definition of superfluous relationship

Definition 4.3. The relationship ***L(i,j)*** is superfluous, if the restrictions, set by this relationship on attributes values, contain in other relationships.

1. On the basis of dependencies (functional, and joins) the set of the relations is formed.
2. By definitions between the relations the relationships are established.
3. The removal of superfluous relationships is carried out.

Algorithm of formation of relationships set

```
L=∅
do i=1 to k
    do j=i+1 to k
        if PK(R_i)=PK(R_j) then
            V= PK(R_i)
            if π_V(R_j)⊆π_V(R_i) then L=L∪L_1(i,j)
            if π_V(R_i)⊆π_V(R_j) then L=L∪L_1(j,i)
        else
            if PK(R_i)⊆R_j then L=L∪L_M(i,j)
            if PK(R_j)⊆R_i then L=L∪L_M(j,i)
        endif
    enddo
enddo
```

- Algorithm has a polynomial difficulty: $O(k^2)$

Removing of superfluous relationships

Theorem. The relationship $\boldsymbol{L(i,j)}$ is superfluous, if there are relationships:

$$\boldsymbol{L(i,m(1))},\ \boldsymbol{L(m(1),m(2))},\ \ldots,\ \boldsymbol{L(m(p-1),m(p))},\ \boldsymbol{L(m(p),j)}$$

and

$$\boldsymbol{PK(i)\subseteq R_{m(s)}},\ \boldsymbol{s=2,3,\ldots,p},$$

where $\boldsymbol{m}$ – array of the relations numbers.

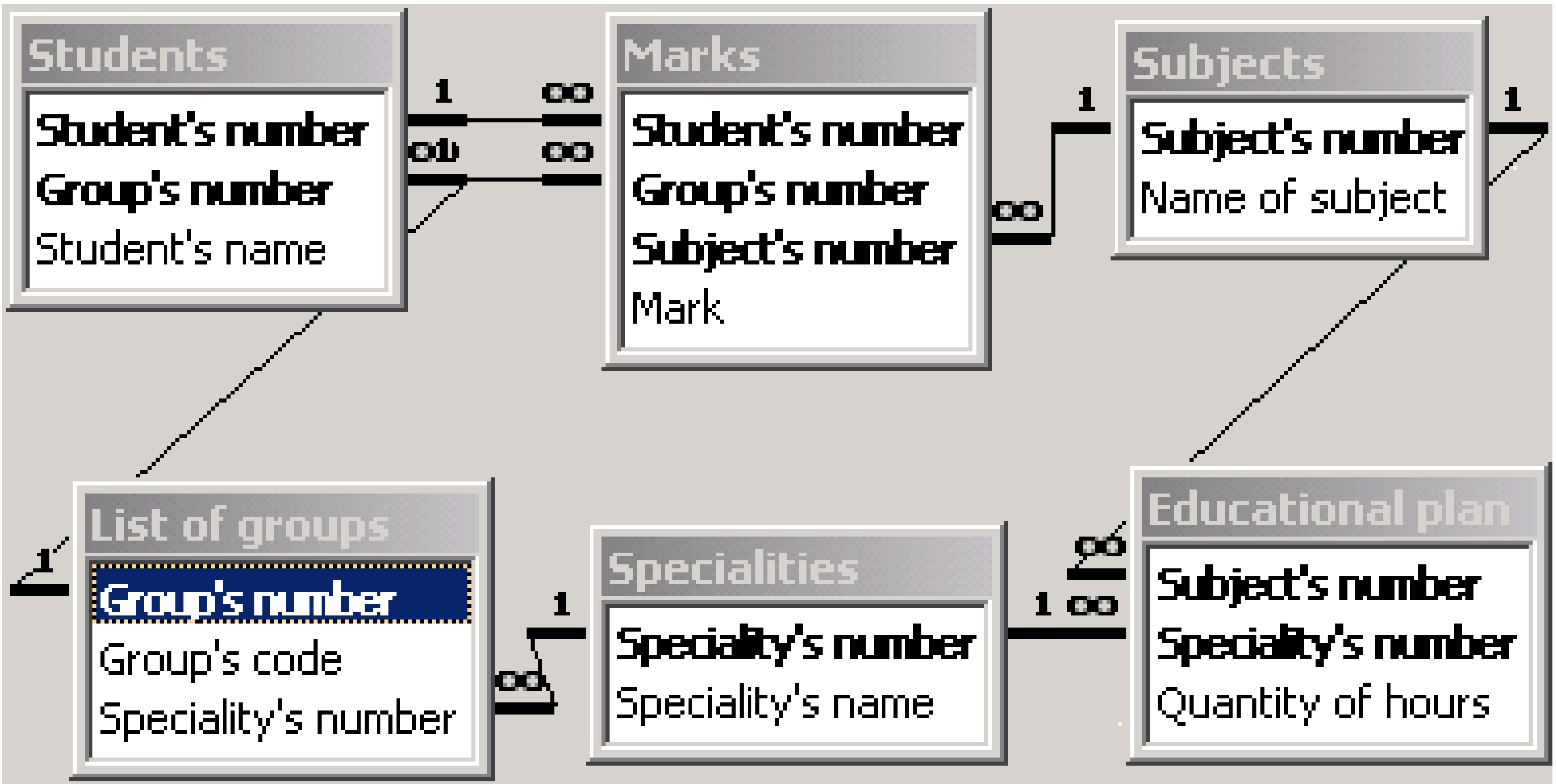
$\boldsymbol{L(i,v)}$, $\boldsymbol{L(v,j)}$, $\boldsymbol{v\neq m(s)}$, $\boldsymbol{s=1,2,\ldots,p}$.

Algorithm

```
for each L(i,j) in L; l=1; m(l)=i; iterations=true
  do while iterations
    for each L(v,w) in L where L(i,j)≠L(v,w); substitution=false
      if v∈m[1,…,l] then
        if w=j then; L=L–L(i,j); exit do;  else
          if w∉m[1,…,l] then; l=l+1; m(l)=w; substitution=true
          endif
        endif
      endif
    endfor
    if not substitution then iterations=false
  enddo
endfor
```

- Algorithm has a polynomial difficulty: $O(|L|^2k^2)$

Database scheme



Thank you for attention

Table of Join

$R_1 =$

A	B	C
a_1	b_1	c_1
a_2	b_2	c_1

$R_2 =$

B	D	E
b_1	d_1	e_1
b_2	d_2	e_2

$R_3 =$

A	D	F
a_1	d_1	f_1
a_2	d_2	f_2
a_3	d_1	f_2

$S =$

A	B	C	D	E	F	l
a_1	b_1	c_1	d_1	e_1	f_1	111
a_2	b_2	c_1	d_2	e_2	f_2	111
a_3	b_1	$*$	d_1	e_1	f_2	011

где $*$ - значение emp.

Semantic transformation

$R =$ A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_2	c_1	d_2
a_3	b_1	c_2	d_1

$ST =$ A	B	c_1	c_2
a_1	b_1	d_1	-
a_2	b_2	d_2	-
a_3	b_1	-	d_1