

Two new operations over extended index matrices and their applications in Big Data

Krassimir Atanassov^{*†}, Veselina Bureva[†] *Dept. of Bioinformatics and Mathematical Modelling Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences 105 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria [†]Intelligent Systems Laboratory "Prof. Asen Zlatarov" University Burgas–8010, Bulgaria

E-mail: krat@bas.bg, vbureva@btu.bg

Abstract-The Index Matrices (IMs) are extensions of the matrices of algebra. Over the IMs different operations, relations and operators are defined. In the present paper, two new operations over IMs are defined and some of their properties are studied. An application of these operations for describing of Big Data procedure is discussed. Hortonworks Data Platform (HDP) is used to provide capabilities for data warehouse processing in the Big Data environment. Apache Hive is selected for data warehouse construction and querying. Firstly, data warehouse for product sells is implemented. The tables for employees, customers, products and product sells are created. Thereafter the new index matrix operations for difference between two IMs are executed for the first time in the Big Data environment . SQL queries are written to demonstrate the operations. The new index matrix operations are executed using SQL JOIN notation and logical operator NOT EXISTS.

Index Terms-Big Data, Extended index matrix, Operation.

I. INTRODUCTION

THE CONCEPT of Index Matrix (IM) was introduced in 1984 and in more details - in 1987 [2], but the full description of the research over them was published in [3] exactly 30 years later.

Different extensions and modifications of the concept of an IM are described in [3]. One of them is an Extended IM (EIM), introduced firstly in [4]. They include as partial cases standard IM with elements of real or complex numbers, the (0, 1)-IM with elements from set $\{0, 1\}$, the logical IM with elements variables, propositions or predicates, the intuitionistic fuzzy IMs. The elements of the EIM can be each objects, in this number - whole IMs.

Different relations, operations and operators are defined over IMs and more general - over EIM. Only part of them have analogues in the theory of the standard matrices (see, e.g., [6], [7]).

Here, two new operations over EIMs are defined and some of their properties are studied. It will be obvious that these new operations can be transfer over each one of the partial cases of the EIM.

Firstly, we give the definition of an EIM.

Let \mathcal{I} be a fixed set of indices,

$$\mathcal{I}^n = \{ \langle i_1, i_2, \dots, i_n \rangle | (\forall j : 1 \le j \le n) (i_j \in \mathcal{I}) \}$$

and

$$\mathcal{I}^* = \bigcup_{1 \le n \le \infty} \mathcal{I}^n.$$

Let \mathcal{X} be a fixed set of some objects. In the particular cases, they can be either real numbers, or only the numbers 0 or 1, or logical variables, propositions or predicates, etc.

Let operations $\circ, * : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$ be fixed and let $(\mathcal{X}, \circ, e_{\circ})$ and $(\mathcal{X}, *, e_{*})$ be groups with unit elements e_{\circ} and e_{*} , respectively. For example, when operation " \circ " is "+" or "-", e_{\circ} will be 0, while when it is " \times " or ":" – it will be 1. In some cases, it is suitable to define the unit element by \perp and it to be an empty object.

An EIM with index sets K and $L(K, L \subset \mathcal{I}^*)$ and elements from set \mathcal{X} is called the object (see, [4], [3]):

$$[K, L, \{a_{k_i, l_j}\}] \equiv \frac{l_1 \dots l_j \dots l_n}{k_1 | a_{k_1, l_1} \dots a_{k_1, l_j} \dots a_{k_1, l_n}}$$

$$\vdots | \vdots \dots \vdots \dots \vdots \\k_i | a_{k_i, l_1} \dots a_{k_i, l_j} \dots a_{k_i, l_n},$$

$$\vdots | \vdots \dots \vdots \dots \vdots \\k_m | a_{k_m, l_1} \dots a_{k_m, l_j} \dots a_{k_m, l_n},$$

where $K = \{k_1, k_2, ..., k_m\}$, $L = \{l_1, l_2, ..., l_n\}$, for $1 \le i \le m$, and $1 \le j \le n : a_{k_i, l_j} \in \mathcal{X}$.

In [3], for the IMs $A = [K, L, \{a_{k_i, l_j}\}], B = [P, Q, \{b_{p_r, q_s}\}]$, operations that are analogous to the usual matrix operations of addition and multiplication are defined, as well as other, specific ones. Here, we give only three of these operations, that will be used below.

Addition

$$A \oplus_{(\circ)} B = [K \cup P, L \cup Q, \{c_{t_u, v_w}\}],$$

where

$$\begin{array}{l} c_{t_u,v_w} = \\ \left\{ \begin{array}{ll} a_{k_i,l_j}, & \text{if } t_u = k_i \in K \text{ and } v_w = l_j \in L - Q \\ & \text{or } t_u = k_i \in K - P \text{ and } v_w = l_j \in L; \end{array} \right. \\ \left. b_{p_r,q_s}, & \text{if } t_u = p_r \in P \text{ and } v_w = q_s \in Q - L \\ & \text{or } t_u = p_r \in P - K \text{ and } v_w = q_s \in Q; \end{array} \right. \\ \left. a_{k_i,l_j} \circ b_{p_r,q_s}, & \text{if } t_u = k_i = p_r \in K \cap P \\ & \text{and } v_w = l_j = q_s \in L \cap Q \\ 0, & \text{otherwise} \end{array} \right.$$

Structural subtraction

$$A \ominus B = [K - P, L - Q, \{c_{t_u, v_w}\}],$$

where here and below "-" is the set-theoretic difference operation and

 $c_{t_u,v_w} = a_{k_i,l_j}$, for $t_u = k_i \in K - P$ and $v_w = l_j \in L - Q$.

Projection

$$pr_{M,N}A = [M, N, \{b_{k_i, l_j}\}],$$

where $M \subseteq K$, $N \subseteq L$, and for each $k_i \in M$ and each $l_j \in N$, $b_{k_i,l_j} = a_{k_i,l_j}$.

II. DEFINITIONS OF THE TWO NEW OPERATIONS

Let the two EIMs $A = [K, L, \{a_{k_i, l_j}\}]$ and $B = [P, Q, \{b_{p_r, q_s}\}]$ are given.

The first, simpler, operation is defined by

$$A \stackrel{\bullet}{}_{1}B = [K \div P, L \div Q, \{c_{u_v, w_t}\}],$$

where and below for every two arbitrary sets X, Y:

$$X \div Y = (X - Y) \cup (Y - X);$$

 $c_{u_v,w_t} = \begin{cases} a_{k_i,l_j}, & \text{if } u_v = k_i \in K - P \text{ and } w_t = l_j \in L - Q \\ b_{p_r,q_s}, & \text{if } w_t = p_r \in P - K \text{ and } w_t = q_s \in Q - L \\ \bot & \text{otherwise} \end{cases}$

The second operation is defined by

$$A \stackrel{\bullet}{\underset{2}{\bullet}} B = [K \cup P, L \cup Q, \{c_{u_v, w_t}\}],$$

where

c

$$\begin{cases} u_v, w_t - \\ a_{k_i, l_j}, & \text{if } u_v = k_i \in K - P \text{ and } w_t = l_j \in L \\ & \text{or } u_v = k_i \in K \text{ and } w_t = l_j \in L - Q \\ b_{p_r, q_s}, & \text{if } w_t = p_r \in P - K \text{ and } w_t = q_s \in Q \\ & \text{if } w_t = p_r \in P \text{ and } w_t = q_s \in Q - L \\ \perp & \text{otherwise} \end{cases}$$

The geometrical interpretations of both operations are shown on Figures 1 and 2.

From the definitions and geometrical interpretations of both operations we see immediately that the following assertions are valid.

Theorem 1. For every two EIMs A and B, for each operation \circ , and for operation * defined for every two $x, y \in \mathcal{X}$ by $x * y = e_*$ there are follows:

$$A \stackrel{\bullet}{\underset{}{}_{1}} B = (A \ominus B) \oplus_{(\circ)} (B \ominus A),$$
$$A \stackrel{\bullet}{\underset{}{}_{2}} B = A \oplus_{(*)} B.$$

Proof. For the definitions of the operations \ominus and \oplus_\circ we obtain

$$(A \ominus B) \oplus_{(\circ)} (B \ominus A) = [K - P, L - Q, \{c_{t_u, v_w}\}] \oplus_{(\circ)} [P - K, Q - L, \{d_{e_f, g_h}\}]$$
$$= [(K - P) \cup (P - K), (L - Q) \cup (Q - L), \{c'_{t'_{u'}, v'_{w'}}\}]$$
$$= [K \div P, L \div Q, \{c'_{t'_{u'}, v'_{u'}}\}] = A \div B,$$

because

$$c'_{t'_{u'},v'_{w'}} = \begin{cases} c_{t_u,v_w} = a_{k_i,l_j}, \text{ for } t'_{u'} = t_u = k_i \in K - P \text{ and } v'_{w'} = v_w \\ d_{e_f,g_h} = b_{p_r,q_s}, \text{ for } t'_{u'} = e_f = p_r \in P - K \text{ and } v'_{w'} = g_h \end{cases}$$

Obviously, operation \circ cannot be applied over the elements of both EIMs because there are not at least two elements from the both EIMs that have equal indices.

The second equality is proved by a similar way. \Box **Theorem 2.** For every two EIMs *A* and *B*, for each operation \circ :

$$A \stackrel{\bullet}{}_{_{1}} B = pr_{K-P,L-Q} A \oplus_{(\circ)} pr_{P-K,Q-L} B,$$

$$A \stackrel{-}{_{2}} B = pr_{K-P,L} A \oplus_{(\circ)} pr_{K\cap P,L} \div Q A \oplus_{(\circ)} pr_{P-K,Q} A$$

$$= pr_{K,L-Q}A \oplus_{(\circ)} pr_{K \div P,L\cap Q}A \oplus_{(\circ)} pr_{P,Q-L}A.$$

The proof is similar to the above one.

Let \mathcal{M} be the set of all EIMs with elements from \mathcal{X}^1 . Let

$$I_{\emptyset} = [\emptyset, \emptyset, \{\bot\}].$$

Theorem 3. $(\mathcal{M}, \underline{\bullet}, I_{\emptyset})$ is a commutative group.

Proof. From the definition of operation " $\stackrel{\bullet}{_}$ " it follows directly that for each two EIMs $A, B \in \mathcal{M}, A \stackrel{\bullet}{_} B \in \mathcal{M}$.

Using the well-known equality for every three sets X, Y and Z:

$$(X \div Y) \div Z = X \div (Y \div Z),$$

for every three EIMs $A, B, C \in \mathcal{M}$, where

$$C = [D, E, \{c_{d_f, e_g}\}]$$

we obtain

$$(A \stackrel{\bullet}{_{-}} B) \stackrel{\bullet}{_{-}} C = ([K, L, \{a_{k_i, l_j}\}] \stackrel{\bullet}{_{-}} [P, Q, \{b_{p_r, q_s}\}]) \stackrel{\bullet}{_{-}} [D, E, \{c_{d_f, e_g}\}]$$
$$= [K \div P, L \div Q, \{x_{u_v, w_t}\}] \stackrel{\bullet}{_{-}} [D, E, \{c_{d_f, e_g}\}]$$

 ${}^{l}When \mathcal{X}$ is a set (class) in the sense of NBG-set theory of all predicates, then $\mathcal M$ will be a set (class).



Fig. 1. Geometrical interpretation of operation $\frac{\bullet}{1}$



Fig. 2. Geometrical interpretation of operation $\frac{\bullet}{2}$

(where each element x_{u_v,w_t} is some element a_{k_i,l_j} or some element b_{p_r,q_s})

$$= [(K \div P) \div D, (L \div Q) \div E, \{y_{\alpha_{\beta}, \gamma_{\delta}}\}]$$

(where each element $y_{\alpha_{\beta},\gamma_{\delta}}$ is some element a_{k_i,l_j} or some element b_{p_r,q_s}), or some element c_{d_f,e_g})

$$= [K \div (P \div D), L \div (Q \div E), \{y_{\alpha_{\beta}, \gamma_{\delta}}\}]$$
$$= ([K, L, \{a_{k_{i}, l_{j}}\}] \stackrel{\bullet}{=} ([P \div D, Q \div E, \{z_{\varepsilon_{\zeta}, \eta_{\theta}}\}])$$

(where each element $z_{\varepsilon_{\zeta},\eta_{\theta}}$ is some element b_{p_r,q_s}), or some element c_{d_f,e_g})

$$= [K, L, \{a_{k_i, l_j}\}] \stackrel{\bullet}{\underset{\scriptstyle \neg}{=}} ([P, Q, \{b_{p_r, q_s}\}]) \stackrel{\bullet}{\underset{\scriptstyle \neg}{=}} [D, E, \{c_{d_f, e_g}\}]).$$

Now, for EIM I_{\emptyset} we obtain

$$A \stackrel{\bullet}{\underset{1}{\bullet}} I_{\emptyset} = [K \div \emptyset, L \div \emptyset, \{c_{u_v, w_t}\}]$$

(where each element c_{u_v,w_t} coincides with the element a_{k_i,l_j} from A for $u_v = k_i, w_t = l_j$)

$$= [K, L, \{a_{k_i, l_j}\}] = A.$$

Analogously is checked that

$$I_{\emptyset} \stackrel{\bullet}{\ }_{1}A = A$$

From the well-known equality $X \div Y = Y \div X$ it follows that

$$A \stackrel{\bullet}{\underset{}{}} B = [K \div P, L \div Q, \{c_{u_v, w_t}\}]$$
$$m = [P \div K, Q \div L, \{c_{u_v, w_t}\}] = B \stackrel{\bullet}{\underset{}{}} A,$$

i.e., the operation $\stackrel{\bullet}{\neg}_1$ is commutative.

Finally,

$$A \stackrel{\bullet}{\underset{\scriptstyle 1}{\bullet}} A = [K \div K, L \div L, \{x_{u_v, w_t}\}] = [\emptyset, \emptyset, \{x_{u_v, w_t}\}] = I_{\emptyset}, [X_{u_v, w_t}] = I_{\emptyset}, [X_{u_v, w_t}$$

because of lack of indices, element x_{u_v,w_t} must be \perp . \Box

Theorem 4. $(\mathcal{M}, \frac{\bullet}{2}, I)$ is a commutative monoid.

The proof is similar to this of Theorem 3, without the last part, i.e., as above, we can check that $(\mathcal{M}, \frac{\bullet}{2}, I)$ is a commutative monoid, but it is not a group because the fact that there is not a set Y that for some non-empty set X to be valid $X \cup Y = \emptyset$.

III. AN EXAMPLE OF DIFFERENCE OPERATIONSIN THE BIG DATA ENVIRONMENT

Hortonworks Data Platform (HDP) is a an open source framework for distributed storage and processing of huge datasets retrieving from different sources. HDP is used to discover insights from structured and unstructured data in the cloud or on-premises. It includes Big Data tools as Hadoop,

🥵 Ambari - Sandbox - Google Chrome				
localhost:8080/#/main/view/HIVE/au	to_hive20_instance			
🚕 Ambari Sandbox 💶 ops	0 alerts	Dashboard	Services Hosts	Alerts Admin
HIVE A QUERY A JOBS III TABLES	% SAVED QUERIES . ♣ UDFs	SETTINGS		+ N
DATABASE Select or search database/schema	db			Browse
TABLES 4 2 +	TABLE > PRODUCTSALES			
Search D		ORAGE INFORMATION	DETAILED INFORMATIO	
tustomers	MUTHORIZATION			
employees	COLUMN NAME	COLUMN TYPE	COMMENT	
	sn	int		
I products	employeen	int		
	customern	int		
	productn	int		
	quantity	int		

Fig. 3. Data warehouse bigdatadb view in Apache Hive environment

Yarn, MapReduce, Hbase, Hive, Flume, Kafka, Druid [5]. The example of the difference operations is performed in the environment of Apache Hive. Apache Hive is a data warehouse used for reading, writing, and managing large amounts of data stored in distributed storage in SQL. Apache Hive flexibility can be extended using the used-defined functions (UDF) [1]. In the current investigation a data warehouse for product sells is implemented. The tables are uploaded using previously prepared csv files. Apache Hive supports only the sets operations Union All and Union [Distinct]. The Intersect and Except (Minus) operations are not included. In the current investigation for the first time the new index matrix operations performing operation difference will be applied by the analogy of the SQL JOIN clause and the logical operator NOT EXISTS in the Big Data environment. The tables Customers and Employees from bigdatadb data warehouse (Fig. 3). The authors will not compare the indexed field Number - the comparison will be performed using the columns for the first name and the last name of the tables. There are from the same data type. The tables Employees (Fig. 4) and Customers (Fig. 5) have the following from:

An example of the standard JOIN operation has the following form:

SELECT FName, Lname, CFname, CLName FROM Employees JOIN ProductSales ON Employees.Number=ProductSales.EmployeeN JOIN Customers ON Customers.CNumber=ProductSales.CustomerN

employees.number	employees.fname	employees.Iname	employees.town
1	Ivan	Dimitrov	Burgas
2	George	Radev	Burgas
3	Simona	Ivanova	Sofia
4	Radi	Hristov	Burgas
5	Vasilena	Moneva	Sofia
6	Hristo	Rachev	Burgas
7	Qni	Simov	Burgas
8	Radina	Tomova	Sofia
9	Stanislav	Stoyanov	Sofia
10	Radostina	Dimitrova	Burgas
11	Dimo	Dimov	Burgas
12	Maria	Yaneva	Burgas
13	Dimitar	Stoilov	Sofia
14	Svetla	Iliyanova	Burgas
15	Margarita	Simova	Burgas
16	Radomira	Kirova	Sofia
17	Radovesta	Todorova	Sofia
18	Valeria	Taneva	Sofia
19	Silviya	Stancheva	Burgas
20	Stilyan	Stoilov	Burgas

Fig. 4. Table of Employees

customers.cnumber	customers.cfname	customers.clname	customers.ctown
1	Ivan	Dimitrov	Burgas
2	George	Radev	Burgas
3	Siyana	Tumova	Sofia
4	Radina	Mihaylova	Sofia
5	Stancho	Dimitrov	Burgas
6	Hari	Dimov	Burgas
7	Qni	Simov	Burgas
8	Radina	Tomova	Sofia
9	Mihail	Radev	Burgas
10	Hrisi	Staneva	Sofia
11	Dimo	Dimov	Burgas
12	Maria	Yaneva	Burgas
13	Monika	Ganeva	Burgas
14	Stanimit	Ganeva	Sofia
15	Hristo	ivanov	Sofia
16	Iliyan	Fotev	Sofia
17	Hristiana	Racheva	Burgas
18	Valeria	Taneva	Sofia
19	Silviya	Stancheva	Burgas
20	Stilyan	Stoilov	Burgas

Fig. 5. Table of Customers

IIVE												NEW JOB + NEW TABLE
4 QUER	4.0	8 8	TABLES	% SAVED QUERIES	A UDFs	O SETTINGS						& NOTIFICATIONS
Worksheet	•											
DATABAS Select or pr	E Inter databased	ichema	< 🖬 bigdi	rtadb							Browse	•
1 SELECT	Filane, Lnae sployees 201	ie, Cfinae 35 Produc	ne, CLName ttSales						1	bigdatad	•	Tables(4)
3 ON Emp. 4 200N C	loyees.Numbe ustomers	r=Produc	tSales.Empl	oyeeN						Search Tab	les	Q
5 ON Case	tomers. Chund	er+Produ	ctSeles.Cus	toeerh						III custo	mers	
										III empli	iyees	
										III produ	ets	
										I produ	ctuales	
✓ Execute	S Save A	a inse	1105- 1	visual Explain								
P RESUL	15 = 0		S VISUAL EX	PLAIN 🐗 TEZ U								
Filter col							-	+ +	2			
fname	iname	chame	ciname									
Ivan	Dimitrov	liyan	Foter									
Oeorge	Radev	Dimo	Dimov									
Simona	Nanova	Qni	Simov									
Radi	Histor	Hari	Dimov									
H1650	Bachev	Valeria	Taneva									

Fig. 6. Execution of standard JOIN operation

I SELECT * PROM custome HHERE NOT EXI:	rs STS (SELECT * FROM emplo WHERE (emp	yees loyees.FName = cus	tomers.Cfname AND employees.LName = customers.CLname)
Execute	e As Insert UDF +	€ Visual Explain	=
RESULTS	LOG % VISU	AL EXPLAIN	TEZ UI
Filter columns	LOG % VISU	AL EXPLAIN 🦼	TEZ UI
Filter columns	LOG % VISU x customers.cfname Siyana	AL EXPLAIN	customers.ctown Sofia
Filter columns customers.cnumber	LOG % VISU x customers.cfname Siyana Radina	customers.ciname Tumova Mihaylova	oustomers.clown Sofia Sofia
Filter columns customers.cnumber 3 4 5	LLOG % VISU x customers.cfname Siyana Radina Stancho	customers.clname Tumova Mihayfova Dimitrov	Customers.ctown Sofia Bogas
Riter columns customers.cnumber 3 4 5 6	LOG % VISU x customers.cfname Silyana Radina Stancho Hari	Customers.clname Tumova Mihaylova Dimitrov Dimov	Customers.ctown Sofia Sofia Burges Burges
Filter columns customers.cnumber 3 4 5 6 9	LOG % VISU customers.cfname Siyana Radina Stancho Hari Mihail	AL EXPLAIN 4 customers.chame Tumova Mihaytova Dimitrov Dimov Radev	customers.ctown Sofia Sofia Burgas Burgas
Filter columns Filter columns Customers.cnumber 3 4 5 6 9 10	k LOG % VISU customers.cfname Styana Stancho Hari Mihail Hrisi	AL EXPLAIN A customers.cliname Tumova Mihaylova Dimitrov Dimov Radev Staneva	Customers.clown Sofia Sofia Burgas Burgas Sofia
Filter columns Filter columns customers.cnumber	k LOG % VISU customers.cfname Siyana Stancho Hari Hrisi Monika	ALLEXPLAIN A customers.clname Tumova Mihayiova Dimitrov Dimov Dimov Staneva Ganeva	Durgas Defa

Fig. 7. SQL query for difference operation - case 1

1 SELECT * 2 FROM employee 3 where NOT EXI 4 5 6 7 8	BS ISTS (SELECT ≭ FROM cust WHERE (em	omers ployees.FName = «	customers.CFname	MO employees.(Hame = customers.(Liname));
Execute Sav RESULTS Filter columns	Insert UDF	 Visual Explanation Visual Explanation 	nin 🖈 TEZ UI	
employees.number	employees.fname	employees.Iname	employees.town	
3	Simona	Ivanova	Sofia	
4	Radi	Hristov	Burgas	
5	Vasilena	Moneva	Sofia	
6	Hristo	Rachev	Burgas	
9	Stanislav	Stoyanov	Sofia	
10	Radostina	Dimitrova	Burgas	
13	Dimitar	Stoilov	Sofia	
14	Svetla	Iliyanova	Burgas	



	- Sandbox - Google Chrome
) localh	nost:8080/#/main/view/HIVE/auto_hive20_instance
	🙏 Ambari Sandbox 🕫 upa 🛛 akata
IVE	
🔺 QU	IERY 刘 JOBS 🎟 TABLES 🚿 SAVED QUERIES 🏟 UDFs 🔷 SETTINGS
01*	+
DATA	BASE
Select	or search database/schema
1 SEL	ECT FName, LName
2 FR	OM employees
3 WH	ERE NOT EXISTS (SELECT CFName, CLName
4	FROM customers
5	IHERE (employees.FName=customers.CFName AND employees.LName=customers.CLName))
6 unio	on
/ SELL	ELI CFNano, CLNano
O PRO	OF CONCERNS
2 MI	ENCE HOT EALSTS (SEECET Phane, Chane
11	WHERE (employees, FName=customers, CFName AND employees, LName=customers, CLName));
	······································

Fig. 9. SQL query for difference operation - case 3

The result presents the names of the customers and the employees. The SQL query in the Apache Hive environment is presented on the Fig. 6.

The simulation of difference operation is performed using the *NOT EXIST* logical operator in the *SELECT* statement and comparison of the desired columns from the tables in the sub-query (Fig. 7).

```
SELECT * FROM customers
WHERE NOT EXISTS (SELECT *
FROM employees
WHERE (employees.FName = customers.CFname
AND employees.LName = customers.CLname)
);
```

The result of the query presents customers not included in the list of the employees. The columns for first names and last names are compared.

The same query with replaced tables in the query and sub-query can be executed to receive the employees not included in the table of the customers (Fig. 8).

The result of the query is presented on the Fig. 10. It combines the employees that are not customers and the customers that are not employees.

The combination of the previous queries using the UNION operator is executed (Fig. 9). RESULTS I LOG % VISUAL EXPLAIN Filter columns × _u2.fname _u2.lname Stoilov Dimitar Hari Dimov Hrisi Staneva Hristiana Racheva Hristo Rachev Hristo ivanov Iliyan Fotev Margarita Simova Mihail Radev Monika Ganeva Radi Hristov Radina Mihaylova Radomira Kirova Radostina Dimitrova Radovesta Todorova Simona lvanova Siyana Tumova Stancho Dimitrov Stanimit Ganeva Stanislav Stoyanov Svetla Iliyanova Vasilena Moneva

Fig. 10. Result of SQL query for difference operation - case 3

SELECTF Name, LName FROM employees WHERE NOT EXISTS (SELECT CFName, CLName FROM customers WHERE (employees.FName=customers.CFName employees.LName=customers.CLName) AND) UNION SELECT CFName, CLName FROM customers WHERE NOT EXISTS (SELECT FName, LName FROM employees WHERE (employees.FName=customers.CFName AND employees.LName=customers.CLName));

IV. CONCLUSION

Two new operations are introduced in the paper. In future, other properties of them will be studied and some other applications will be discussed. A part of them will be related to Big Data and Data Mining as continuation of the discussed in the present paper.

ACKNOWLEDGMENT

The authors acknowledge the support from the project UNITe BG05M2OP001-1.001-0004 /28. 02.2018 (2018-2023).

REFERENCES

- [1] Apache Hive, https://hive.apache.org/
- [2] Atanassov K. Generalized index matrices, Comptes rendus de l'Academie Bulgare des Sciences, Vol. 40, 1987, No. 11, 15–18.
- [3] Atanassov, K., Index Matrices: Towards an Augmented Matrix Calculus, Springer, Cham, 2014.
- [4] Atanassov, K., Extended index matrices, Proc. of the 7-th IEEE Conference "Intelligent Systems", Warsaw, 24-26 Sept. 2014 (P. Angelov, K. Atanassov, L. Doukovska, M. Hadjiski, V. Jotsov, J. Kacprzyk, N. Kasabov, S. Sotirov, E. Szmidt, S. Zadrozny, Eds.), Springer, Cham, 2015, 305-317.
- [5] Hortonworks Data platform, https://www.cloudera.com/products/hdp.html
- [6] Lankaster, P. Theory of Matrices, Academic Press, New York, 1969.
- [7] Zhang, F. Matrix Theory. Springer, New York, 2011.