



Fundação Vanzolini

Dominando Big Data com o uso de Plataformas Gratuitas

Aula 5

Agenda da aula 5

- ✓ Resolução Chicago Crimes
- ✓ Transformação de dados
 - ✓ Ordenação (e remoção de duplicidade) de registros
 - ✓ Normalização de registros
- ✓ Exercício prático e Desafio

Desafio: Chicago Crimes

Desafio Chicago Crimes:

- Gere um dataset padronizado
 - Adicione um campo de identificador de registro
 - Padronize os campos de hora e data

##	row_id	day	time	id	case_number	block	iucr	primary_type	description
1	1	20190610	235500	11718445	JC301146	022XX S SAWYER AVE	0312	ROBBERY	ARMED:KNIFE/CUTTING INSTRUMENT
2	2	20190610	235500	11718423	JC301185	003XX N PINE AVE	0890	THEFT	FROM BUILDING
3	3	20190610	235500	11718364	JC301127	033XX S MICHIGAN AVE	2093	NARCOTICS	FOUND SUSPECT NARCOTICS
4	4	20190610	235000	11718476	JC301140	057XX S ABERDEEN ST	0497	BATTERY	AGGRAVATED DOMESTIC BATTERY: OTHER DANG WEAPC
5	5	20190610	234700	11718619	JC301294	050XX W DIVISION ST	031A	ROBBERY	ARMED: HANDGUN
6	6	20190610	234500	11718392	JC301160	003XX E 118TH ST	0910	MOTOR VEHICLE THEFT	AUTOMOBILE
7	7	20190610	234000	11718384	JC301137	080XX S LANGLEY AVE	0486	BATTERY	DOMESTIC BATTERY SIMPLE
8	8	20190610	233100	11718398	JC301118	047XX N KEYSTONE AVE	051A	ASSAULT	AGGRAVATED: HANDGUN
9	9	20190610	233100	11718368	JC301109	096XX S MERRION AVE	1310	CRIMINAL DAMAGE	TO PROPERTY
10	10	20190610	232400	11718393	JC301135	105XX S SANGAMON ST	0497	BATTERY	AGGRAVATED DOMESTIC BATTERY: OTHER DANG WEAPC

Solução proposta:

```
IMPORT $,STD;

Crimes:=$.File_crime_optimized;

// New layout for formatted data
new_Layout := RECORD
    UNSIGNED row_id;
    UNSIGNED4 day;
    UNSIGNED4 time;
    Crimes.Layout AND NOT date;
END;

// TRANSFORM structure and associated PROJECT function for cleansing the original dataset
New_Layout Reformatter(Crimes.Layout L, UNSIGNED cnt):=TRANSFORM
    SELF.row_id:=cnt;
    SELF.day:=(UNSIGNED4) STD.Date.FromStringToDate(L.Date[1..10], '%m/%d/%Y');
    SELF.time:=(UNSIGNED4) STD.Date.TimeFromParts
        (IF(L.Date[21..22]='PM',
            IF(L.Date[12..13]='12',12,(UNSIGNED1)L.Date[12..13]+12),
            IF(L.Date[12..13]='12',0,(UNSIGNED1)L.Date[12..13])),
        (UNSIGNED1)L.Date[15..16],(UNSIGNED1)L.Date[18..19]);
    SELF:=L;
END;

EXPORT Formatted_File := PROJECT(Crimes.File,Reformatter(LEFT,COUNTER));

// OUTPUT of the clean dataset
// OUTPUT(Formatted_File,, '~chicago::hmw::out::Formatted_file',overwrite);
```

Transformação de dados

124.38 MB

id	firstname	lastname	middlename	namesuffix	filedate	bureaucode	marital	gender	dependentcount	birthdate	streetaddress	city	state	zipcode
91082180...	Cherianne	Khatchatourian	N		19990922	24		M	0		69 BOULDER RIDGE RD # 25A	HAWKINS	WI	54530
16505326...	Muyesser	Raplee	X		20001111	353		F	0		55 SWAMP RD	DISTRICT HEIGHT	MD	20747
24548180...	Roselin	Viceconte			19990325	344		F	0	19800113	107 HILL TER	ENTERPRISE	OR	97828
15880908...	Inda	Provines			20000909	13		U	0		290 W MOUNT PLEASANT AVE	LAVACA	AR	72941
65127056...	Inderdeep	Laurence	D		20001228	344		M	0		44 PROSPECT PL	GREENSBORO	FL	32330
91939895...	Chrystine	Mangiapane			19990827	315		F	0	19780306	1806 1ST AVE APT 8F	ARVADA	CO	80007
12286552...	Adelene	Stock	R		20000827	252		M	0		1117 FARM RD	DOVER	DE	19901
11459575...	Mendy	Rufenblanchette			20000903	24		M	0		3 W 83RD ST APT 4C	WILLIAMSTON	SC	29697
80539064...	Lannie	Amerantes	I		20001219	313		U	0		200 W 20TH ST APT 909	CHARLESTON	WV	25312
48476875...	Tare	Gonyeau	T		19930807	48		F	0	19750801	6 CANDLE CT	EL PASO	TX	79924
16156125...	Finney	Aristilde	P		19900621	344		M	0	19560920	222 1ST AVE APT 2B	MACON	GA	31220
13804468...	Oreoluwa	Marthaler			19931006	358		F	0	19731201	176 CLAREMONT GDNS	AUBURN	ME	04210
11995825...	Surge	Abbottkrepp	D		20000308	13		F	0		22 LE PARC CT	TWINSBURG	OH	44087
15714117...	Dave	Mcjury			20001129	238		U	0		510 COOPER RD # 1	TACOMA	WA	98402

89.87 MB

recid	id	firstname	lastname	middlename	namesuffi	filedate	bureaucode	gender	birthdate	streetaddress	csz_id
1	91082180...	CHERIANNE	KHATCHATOURIAN	N		19990922	24	M	0	69 BOULDER RIDGE RD # 25A	1
2	16505326...	MUYESSER	RAPLEE	X		20001111	353	F	0	55 SWAMP RD	2
3	24548180...	ROSELIN	VICECONTE			19990325	344	F	19800113	107 HILL TER	3
4	15880908...	INDA	PROVINES			20000909	13	U	0	290 W MOUNT PLEASANT AVE	4
5	65127056...	INDERDEEP	LAURENCE	D		20001228	344	M	0	44 PROSPECT PL	5
6	91939895...	CHRYSTINE	MANGIAPANE			19990827	315	F	19780306	1806 1ST AVE APT 8F	6
7	12286552...	ADELENE	STOCK	R		20000827	252	M	0	1117 FARM RD	7
8	11459575...	MENDY	RUFENBLANCHETTE			20000903	24	M	0	3 W 83RD ST APT 4C	8
9	80539064...	LANNIE	AMERANTES	I		20001219	313	U	0	200 W 20TH ST APT 909	9
10	48476875...	TARE	GONYEAU	T		19930807	48	F	19750801	6 CANDLE CT	10
11	16156125...	FINNEY	ARISTILDE	P		19900621	344	M	19560920	222 1ST AVE APT 2B	11
12	13804468...	OREOLUNA	MARTHALER			19931006	358	F	19731201	176 CLAREMONT GDNS	12
13	11995825...	SURGE	ABBOTTKREPP	D		20000308	13	F	0	22 LE PARC CT	13
14	15714117...	DAVE	MCJURY			20001129	238	U	0	510 COOPER RD # 1	14

586.32 KB

csz_id	city	state	zipcode
1	HAWKINS	WI	54530
2	DISTRICT HEIGHT	MD	20747
3	ENTERPRISE	OR	97828
4	LAVACA	AR	72941
5	GREENSBORO	FL	32330
6	ARVADA	CO	80007
7	DOVER	DE	19901
8	WILLIAMSTON	SC	29697
9	CHARLESTON	WV	25312
10	EL PASO	TX	79924
11	MACON	GA	31220
12	AUBURN	ME	4210
13	TWINSBURG	OH	44087
14	TACOMA	WA	98402

Funções de ordenação e remoção de duplicidades

Função de ordenação

`SORT(recordset, campo)`

- *recordset* – O conjunto de registros a ser processado.
- *campo* – Um expressão ou campo no *recordset* a ser ordenado.

A função **SORT** ordena o *recordset* de acordo com o campo especificado.

```
OldPeople := SORT(Persons, birthdate);  
YoungPeople := SORT(Persons, -birthdate);
```

##	firstname	lastname	birthdate
1	Jinkon	Rushford	20091231
2	Demetri	Awan	20091231
3	Stellene	Gavrich	20091230
4	Fungjen	Mcquaide	20091228
5	Tatil	Tsenter	20091226
6	Tupong	Misko	20091226
7	Motek	Cashel	20091226
8	El	Panasci	20091225
9	Thomila	Beverly	20091225
10	Laurine	Foad	20091223

Exemplo de SORT

- Training_Examples.SORT_Example

Remoção de duplicidades

DEDUP(*recset* [,*condição*] [,**ALL**])

- *recset* – O conjunto de registros a ser processado.
- *condição* – A expressão que define registros “duplicados”.
- **ALL** – Compara todos os registros entre si usando a *condição*, não apenas os registros adjacentes.

A função **DEDUP** remove os registros duplicados do *recordset*.

SortRecs := **SORT**(Persons, firstname, lastname);

DeDupRecs := **DEDUP**(SortRecs, firstname, lastname); //compara apenas registros adjacentes

##	recid	firstname	lastname
1	1	Alysson	Oliveira
2	2	Artur	Baruchi
3	3	Baruchi	Watanuki
4	4	Hugo	Watanuki
5	5	Hugo	Watanuki



##	recid	firstname	lastname
1	1	Alysson	Oliveira
2	2	Artur	Baruchi
3	3	Baruchi	Watanuki
4	4	Hugo	Watanuki

Palavras-chave LEFT e RIGHT

LEFT.campo / RIGHT.campo

As palavras-chave **LEFT** e **RIGHT** qualificam o registro de origem de cada *campo* nas operações que processam um par de registros.

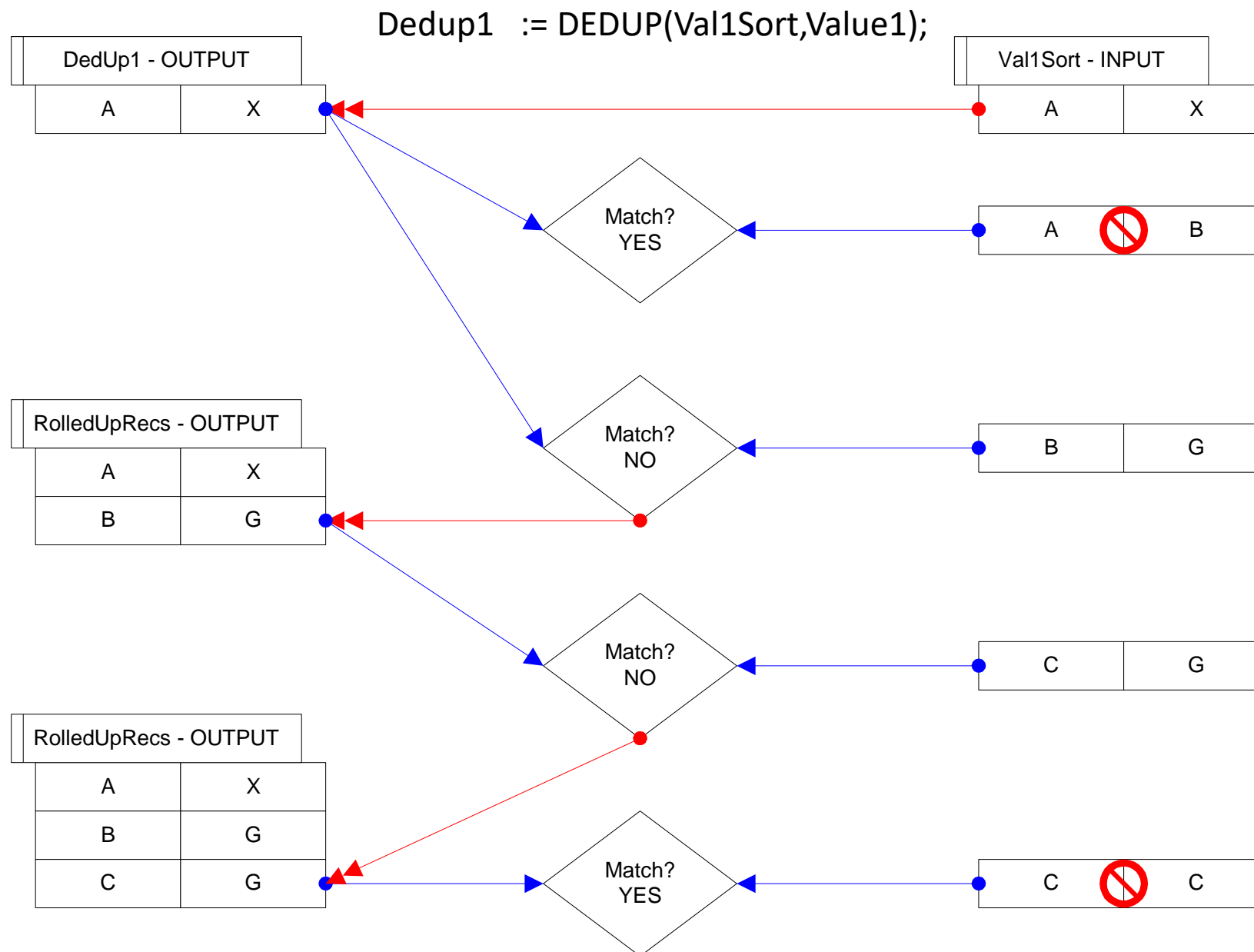
Typo_Dedup := **DEDUP**(DeDupRecs, **LEFT**.lastname = **RIGHT**.firstname);

##	recid	firstname	lastname
1	1	Alysson	Oliveira
2	2	Artur	Baruchi
3	3	Baruchi	Watanuki
4	4	Hugo	Watanuki



##	recid	firstname	lastname
1	1	Alysson	Oliveira
2	2	Artur	Baruchi
3	4	Hugo	Watanuki

Simple DEDUP Functional Example Diagram



Exemplo de DEDUP

- Training_Examples.DEDUP_Example

Exemplo de ROLLUP:

```
myrec Mytransf(myrec Le, myrec Ri) := TRANSFORM  
  SELF.account := Le.account + ',' + Ri.account;  
  SELF.balance := Le.balance + Ri.balance;  
  SELF := Le;  
END;
```

```
rolledrecs := ROLLUP(myds, LEFT.owner=RIGHT.owner, Mytransf(LEFT, RIGHT));
```

##	account	owner	balance
1	501	AlyssonO	1000
2	502	ArturB	500
3	503	ArturB	1500
4	504	HugoW	200
5	505	HugoW	500



##	account	owner	balance
1	501	AlyssonO	1000
2	502,503	ArturB	2000
3	504,505	HugoW	700

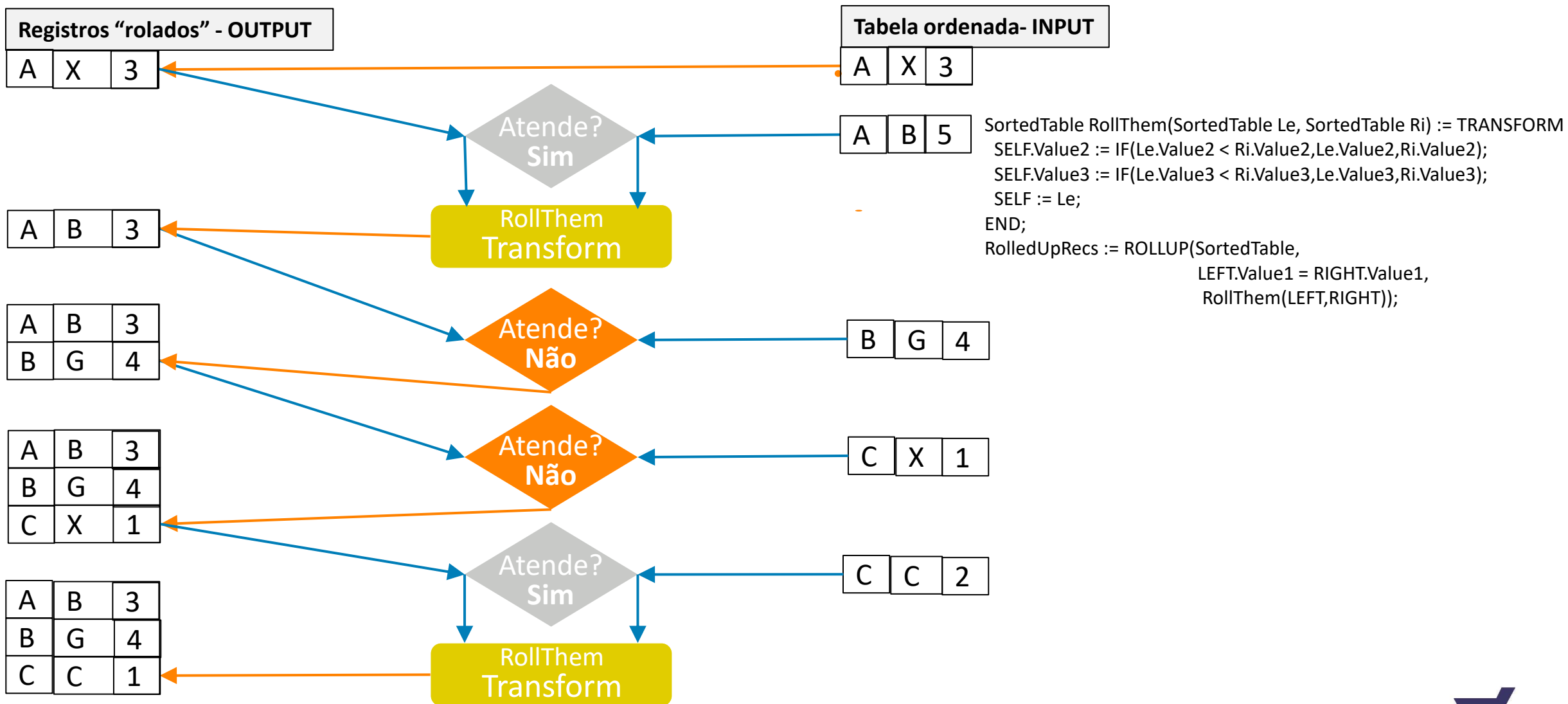
Função ROLLUP

A função **ROLLUP** é similar ao DEDUP com a adição de uma chamada ao *transform*.

ROLLUP(*recordset, condição, transform*)

- *recordset* – Conjunto de registros a serem processados.
- *condição* – Especifica a condição de duplicidade dos registros.
- *transform* – Nome do TRANSFORM a ser chamado para cada par de registros considerados duplicados.

Diagrama funcional do ROLLUP:



Normalização de registros

Normalização de registros

- Função ROLLUP
- Função JOIN

Exemplo de JOIN:

```
outrec := RECORD
  myrec;
  string email;
END;
```

```
outrec Mytransf2(myrec Le, myrec2 Ri) := TRANSFORM
  SELF := Le;
  SELF := Ri;
END;
```

```
joinedrecs := JOIN(rolledrecs, myds2,
  LEFT.owner=RIGHT.owner,
  Mytransf2(LEFT, RIGHT));
```

##	account	owner	balance
1	501	AlyssonO	1000
2	502,503	ArturB	2000
3	504,505	HugoW	700

##	owner	email
1	AlyssonO	alysson.o@ecl.com
2	ArturB	artur.b@ecl.com
3	HugoW	hugo.w@ecl.com



##	account	owner	balance	email
1	501	AlyssonO	1000	alysson.o@ecl.com
2	502,503	ArturB	2000	artur.b@ecl.com
3	504,505	HugoW	700	hugo.w@ecl.com

Função JOIN

A função JOIN processa dois conjuntos de registros avaliando uma *condição*. Um *transform* pode ser executado para cada par de registros que atendam à *condição*.

JOIN(*leftset*, *rightset*, *condição*[, *transform*] [,*tipo*] [,*flag*])

- *leftset* – O conjunto de registros LEFT a ser processado. Deve ser o maior conjunto de registros.
- *rightset* – O conjunto de registros RIGHT a ser processado.
- *condição* – Condição de ligação entre os registros LEFT e RIGHT.
- *transform* – TRANSFORM a ser chamado.
- *tipo* – Tipo de associação (padrão é “inner join”).
- *flag* – Opções para especificar como o JOIN deve operar.

Tipos de JOIN

INNER – Todos os registros que atendam à condição de ligação.

LEFT OUTER – Todos os registros que atendam à condição de ligação + registros do *leftset* que não a atendam.

RIGHT OUTER – Todos os registros que atendam à condição de ligação + registros do *rightset* que não a atendam.

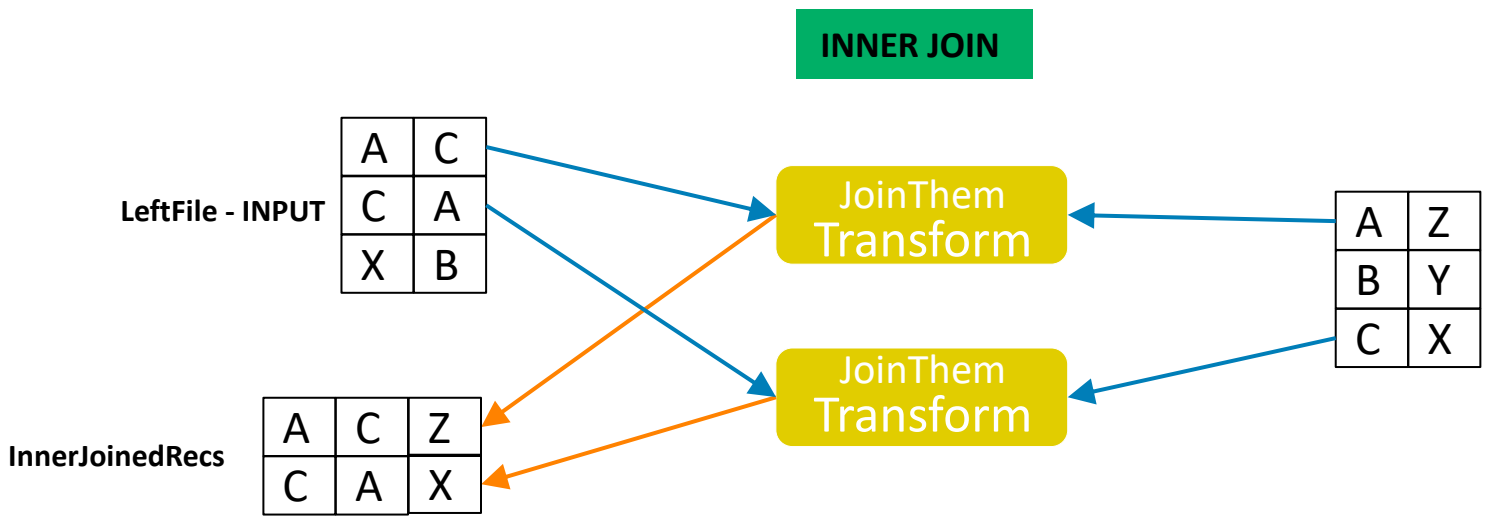
FULL OUTER – Todos os registros que atendam à condição de ligação + registros do *leftset* e *rightset* que não a atendam (JOIN de inclusão).

LEFT ONLY – Apenas os registros do *leftset* que não atendam à condição de ligação.

RIGHT ONLY – Apenas os registros do *rightset* que não atendam à condição de ligação.

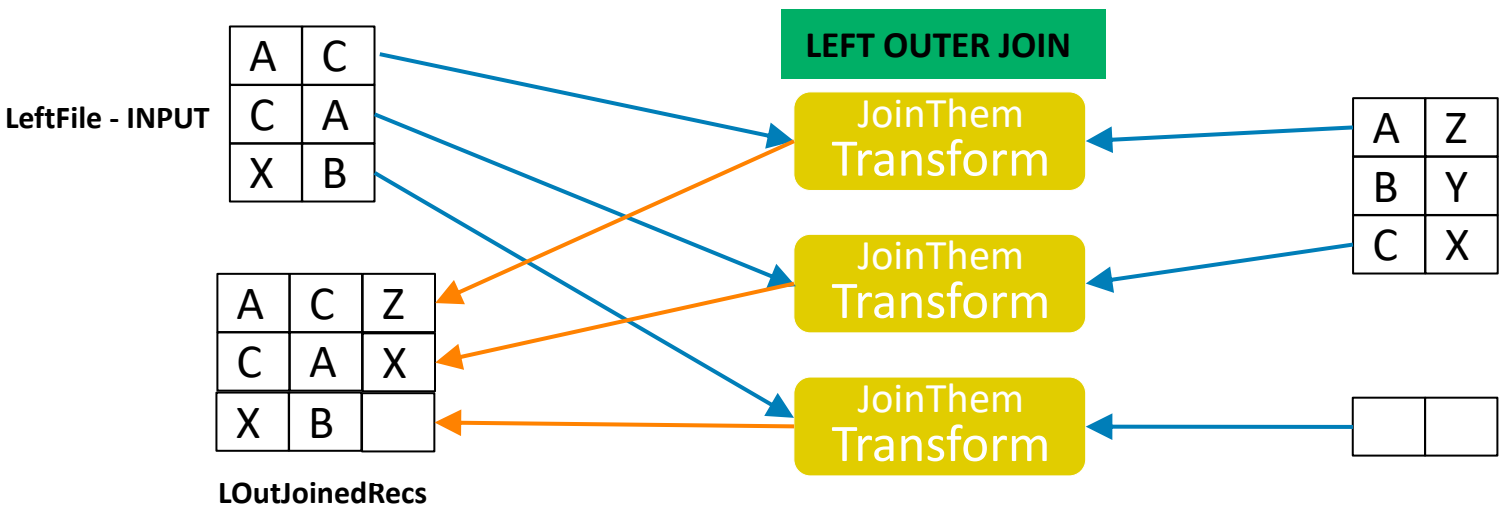
FULL ONLY – Apenas os registros do *leftset* e *rightset* que não atendam à condição de ligação (JOIN de exclusão).

Diagrama funcional do JOIN



```
MyOutRec JoinThem(MyRec L, MyRec R) := TRANSFORM
    SELF.Value1 := IF(L.Value1<>"", L.Value1, R.Value1);
    SELF.LeftValue2 := L.Value2;
    SELF.RightValue2 := R.Value2;
END;
```

```
InnerJoinedRecs := JOIN(LeftFile,RightFile,
    LEFT.Value1 = RIGHT.Value1,
    JoinThem(LEFT,RIGHT),
    LOOKUP);
```

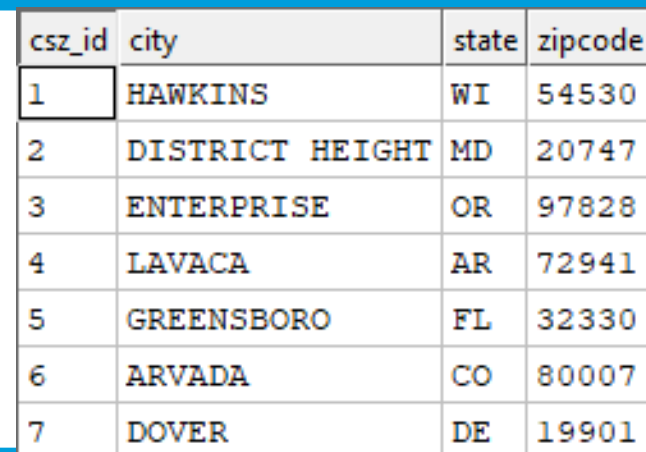


```
LOutJoinedRecs := JOIN(LeftFile,RightFile,
    LEFT.Value1 = RIGHT.Value1,
    JoinThem(LEFT,RIGHT),
    LEFT OUTER);
```

Exercício prático

Exercícios 6a e 6b – Criação de tabela de referência

- Vertical slice (TABLE)
- SORT
- ROLLUP (IF)
- Estrutura MODULE

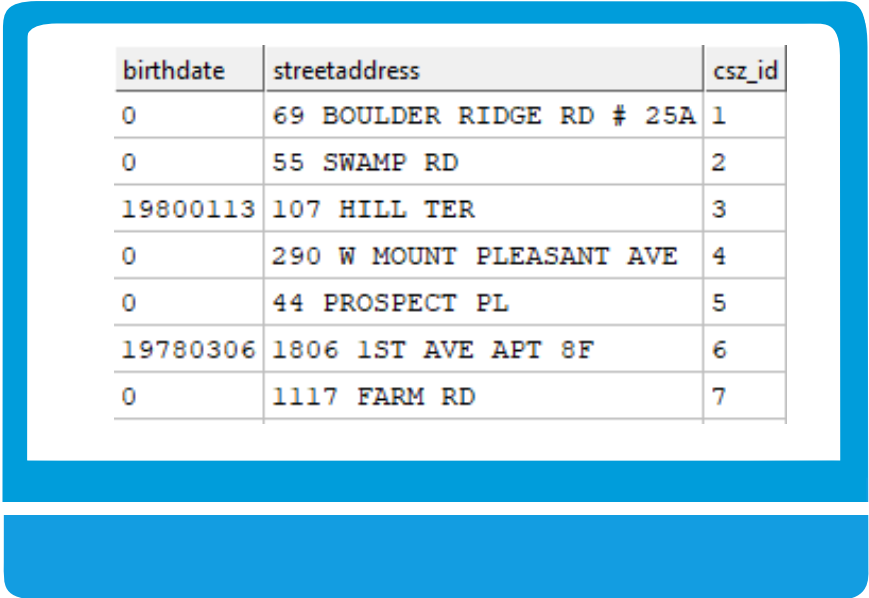


csz_id	city	state	zipcode
1	HAWKINS	WI	54530
2	DISTRICT HEIGHT	MD	20747
3	ENTERPRISE	OR	97828
4	LAVACA	AR	72941
5	GREENSBORO	FL	32330
6	ARVADA	CO	80007
7	DOVER	DE	19901

Exercício prático

Exercício 7a – Finalizando a normalização (Persons)

- Estrutura MODULE
- JOIN



birthdate	streetaddress	csz_id
0	69 BOULDER RIDGE RD # 25A	1
0	55 SWAMP RD	2
19800113	107 HILL TER	3
0	290 W MOUNT PLEASANT AVE	4
0	44 PROSPECT PL	5
19780306	1806 1ST AVE APT 8F	6
0	1117 FARM RD	7

Desafio: Chicago Crimes

Desafio Chicago Crimes:

- Normalize o dataset de crimes em duas tabelas:
 - uma tabela de crimes e;
 - uma tabela de endereços (block, community area, district)
- Mantenha o campo de identificador de registro criado no desafio anterior (row_id) como campo de ligação entre as duas tabelas

##	row_id	day	time	id	case_number	block	iucr	primary_type	description
1	1	20200317	213000	12014684	JD189901	039XX N LECLAIRE AVE	0820	THEFT	\$500 AND UNDER
2	2	20190924	80000	11864018	JC476123	022XX S MICHIGAN AVE	1154	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THE
3	3	20191013	203000	11859805	JC471592	024XX W CHICAGO AVE	0860	THEFT	RETAIL THEFT
4	4	20200318	20300	12012127	JD189186	039XX W JACKSON BLVD	0910	MOTOR VEHICLE THEFT	AUTOMOBILE
5	5	20191005	183000	11863808	JC476236	0000X N LOOMIS ST	0810	THEFT	OVER \$500
6	6	20191013	190000	11859727	JC471542	016XX W ADDISON ST	1320	CRIMINAL DAMAGE	TO VEHICLE
7	7	20191013	141000	11859656	JC471240	051XX N BROADWAY	0560	ASSAULT	SIMPLE
8	8	20191013	195000	11859827	JC471571	011XX W JACKSON BLVD	0860	THEFT	RETAIL THEFT
9	9	20191013	500	11859127	JC470662	064XX S VERNON AVE	0486	BATTERY	DOMESTIC BATTERY SIMPL
10	10	20200318	85000	12012330	JD189367	023XX N KEELER AVE	0560	ASSAULT	SIMPLE

Até a próxima aula!!!

