Fundação Vanzolini

Dominando Big Data com o uso de Plataformas Gratuitas

Aula 2

LexisNexis® RISK SOLUTIONS

RELX Group

PRO

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

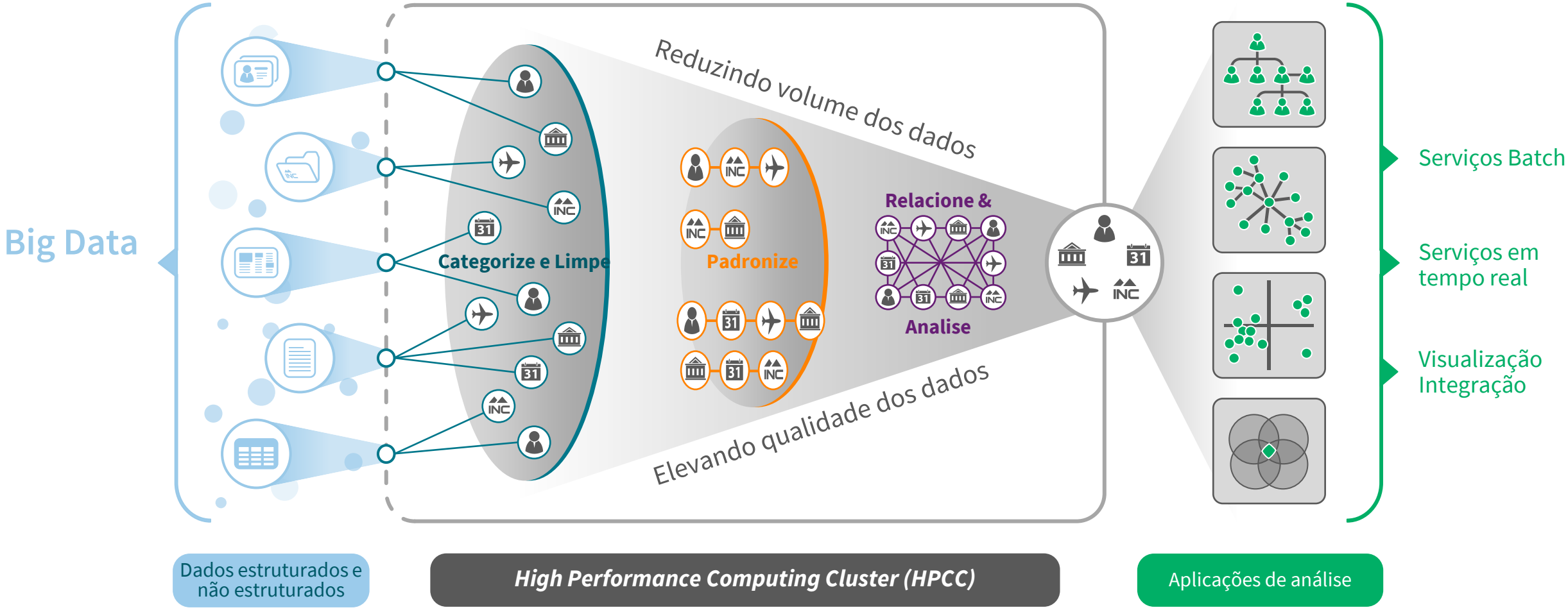
USP UNIVERSIDADE DE SÃO PAULO

Agenda da aula 2

- ✓Revisão
- ✓Extração de dados
- ✓Exercício prático e Desafio

Extração de dados

Extract, Transform, Load



Fundação Vanzolini

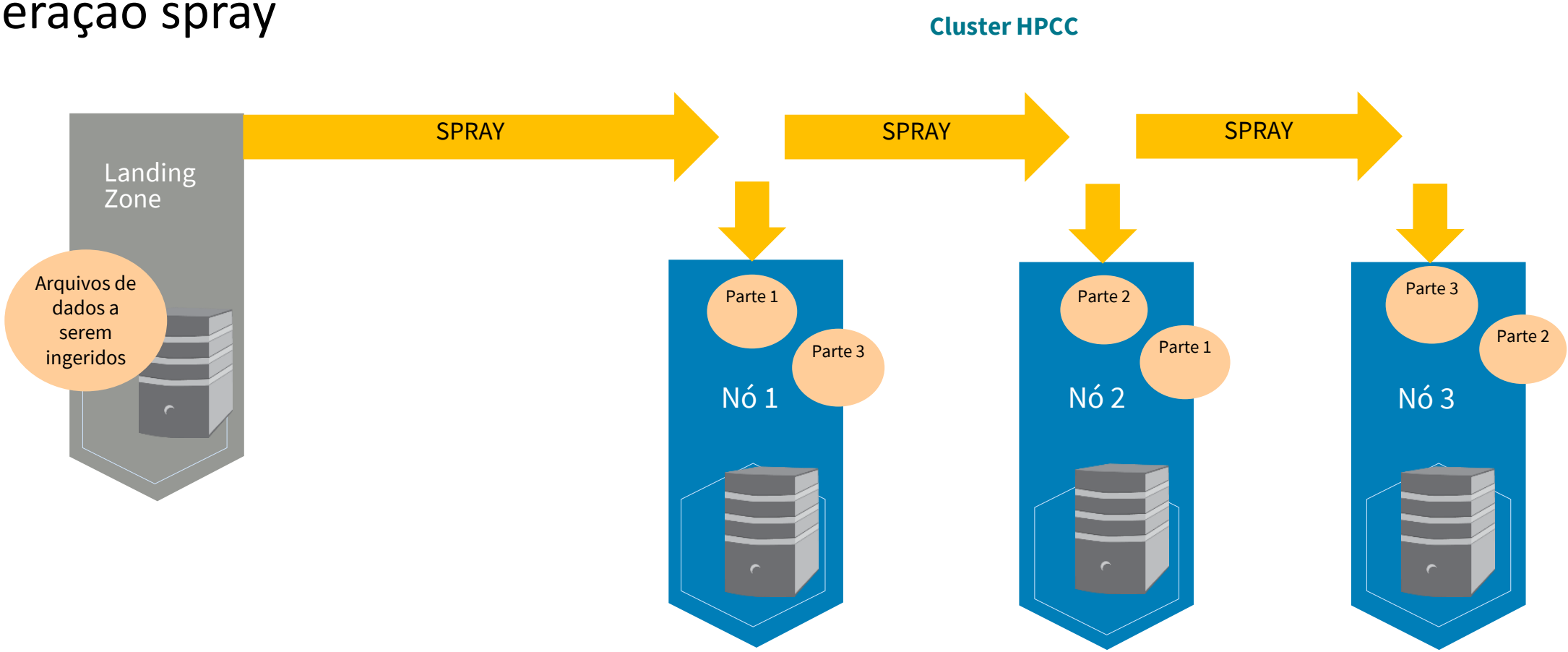
Definição de Extração:

Importação e limpeza de dados brutos provenientes de diferentes fontes

- Importação dos dados brutos
- Definição da estrutura de dados
- Análise do perfil dos dados

Importação de dados brutos

- A operação spray



As partes do arquivo são referenciadas em ECL como um único arquivo lógico...

Definição da estrutura de dados

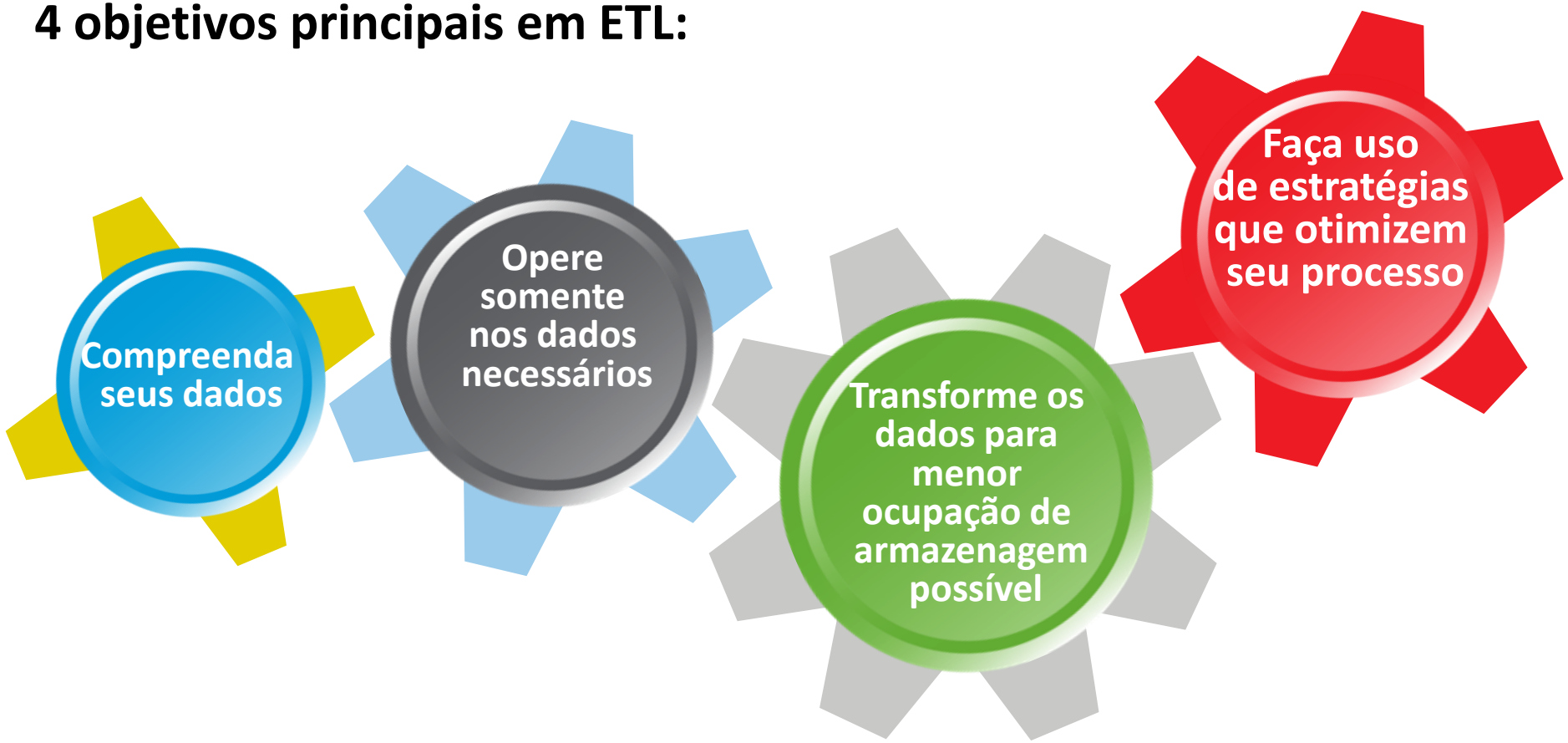
```
EXPORT File_Persons := MODULE
  EXPORT Layout := RECORD
    UNSIGNED8 ID;
    STRING15  FirstName;
    STRING25  LastName;
    STRING15  MiddleName;
    STRING2   NameSuffix;
    STRING8   FileDate;
    UNSIGNED2 BureauCode;
    STRING1   MaritalStatus;
    STRING1   Gender;
    UNSIGNED1 DependentCount;
    STRING8   BirthDate;
    STRING42  StreetAddress;
    STRING20  City;
    STRING2   State;
    STRING5   ZipCode;
    END;
  EXPORT File := DATASET('~CLASS::hmw::Intro::Persons', Layout,FLAT);
END;
```

##	id	firstname	lastname	middl...	n...	filedate	bureaucode	marit...	gender	dep...	birthdate	streetaddress	city	state	zipcode
1	9108218085885411565	Cherianne	Khatchatourian	N		19990922	24		M	0		69 BOULDER RIDGE RD # 25A	HAWKINS	WI	54530
2	16505326057200398078	Muyesser	Raplee	X		20001111	353		F	0		55 SWAMP RD	DISTRICT HEIGHT	MD	20747
3	2454818069645923666	Roselin	Viceconte			19990325	344		F	0	19800113	107 HILL TER	ENTERPRISE	OR	97828
4	15880908289586509107	Inda	Provines			20000909	13		U	0		290 W MOUNT PLEASANT AVE	LAVACA	AR	72941
5	6512705660523829539	Inderdeep	Laurence	D		20001228	344		M	0		44 PROSPECT PL	GREENSBORO	FL	32330
6	9193989543268753887	Chrystine	Mangiapane			19990827	315		F	0	19780306	1806 1ST AVE APT 8F	ARVADA	CO	80007
7	12286552293562700162	Adelene	Stock	R		20000827	252		M	0		1117 FARM RD	DOVER	DE	19901
8	11459575736386985069	Mendy	Rufenblanchette			20000903	24		M	0		3 W 83RD ST APT 4C	WILLIAMSTON	SC	29697
9	8053906447536575038	Lannie	Amerantes	I		20001219	313		U	0		200 W 20TH ST APT 909	CHARLESTON	WV	25312
10	484768759680234166	Tare	Gonyeau	T		19930807	48		F	0	19750801	6 CANDLE CT	EL PASO	TX	79924
11	16156125023194932930	Finney	Aristilde	P		19900621	344		M	0	19560920	222 1ST AVE APT 2B	MACON	GA	31220
12	13804468446718957143	Oreoluwa	Marthaler			19931006	358		F	0	19731201	176 CLAREMONT GDNS	AUBURN	ME	04210
13	11995825474648190448	Surge	Abbottkrepp	D		20000308	13		F	0		22 LE PARC CT	TWINSBURG	OH	44087
14	15714117310244664573	Dave	Mcjury			20001129	238		U	0		510 COOPER RD # 1	TACOMA	WA	98402
15	12587451362606486546	Ramsay	Ping			20001129	238		M	0		404 AVENUE L	MESQUITE	NV	89024

Visão geral da ECL

Enterprise Control Language (ECL)

4 objetivos principais em ETL:



Definição básica ECL

```
Nome := Expressão ;
```

Conceitos básicos de ECL

- Paradigma declarativo (não-procedural)
- ECL *não é* sensível a caixa alta/baixa
- Espaço em branco é ignorado para melhor leitura
- Comentários em linha (//) e em bloco (/* e */)
- ECL utiliza sintaxe objeto.propriedade

Dataset.Campo // referencia um campo em um dataset

NomedoDiretorio.Definicao // referencia uma definição em outro módulo

Tipos de dados primitivos

BOOLEAN

```
BOOLEAN IsValid    := TRUE;
```

STRING[n]

```
STRING1 Gender := 'M';
```

INTEGER[n], UNSIGNED[n],

```
INTEGER1 ictr := -100;      // -128 to 127
UNSIGNED1 ctr := 0;      // 0 - 255
```

REAL[n], DECIMALn[_y]

```
REAL4 PI := 3.14159;
DECIMAL7_2 Salary := 75000.00;
```

Tipos básicos de definição ECL

Booleana (*boolean*)

```
IsSeniorCitizen := People.Age >= 65;
```

Valor único (*value*)

```
ValueTrue := 1;
```

Conjunto de valores (*set*)

```
SetTrueFalseValues := [0, 1];
```

Conjunto de registros (*recordset*)

```
TopSeniorPeople := People(IsSeniorCitizen);
```

Ações vs. Definições

- ✓ O código ECL é constituído de:
  - ✓ *Definições* estabelecem *o que* as coisas são (arquivos de definição ECL)

```
A := 'People';  // não inicia uma WU
```

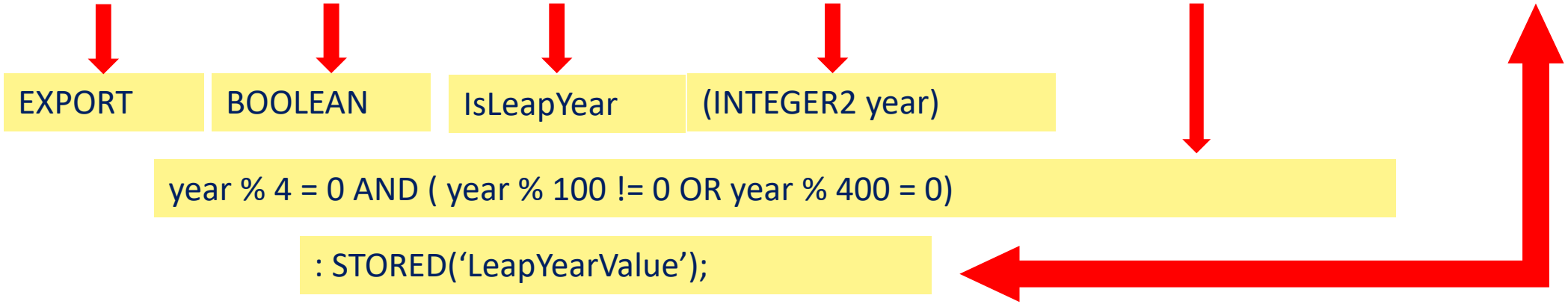
  - ✓ *Ações* resultam em compilação e execução (arquivos BWR)

```
OUTPUT('People');          // inicia uma WU
```

Sintaxe Completa de uma Definição ECL

Nome := Expressão ;

[Escopo] [TipoValor] Nome [(parâmetros)] := Expressão [:ServiçoWorkflow] ;



Escopo da definição (Visibilidade)

Global –

A palavra-chave **EXPORT** torna a definição disponível "globalmente" no repositório

```
EXPORT PeopleCount := COUNT(People);
```

Módulo –

A palavra-chave **SHARED** torna a definição disponível somente no modulo/diretório que a contém

```
SHARED StateCount := 50;
```

Local –

A **ausência dessas palavras-chave** torna a definição disponível somente no arquivo que a contém e até a próxima definição ECL que contenha EXPORT ou SHARED

```
Num5 := 5;
EXPORT NumTotal := Num5 + 10 + StateCount;
```

Escopo da definição (Visibilidade)

IMPORT listadiretorios

- listadiretorios – Uma lista de diretórios separados por vírgula.

A palavra-chave **IMPORT** define uma lista de diretórios cujos arquivos de definições exportados tornam-se diponíveis para uso no código.

```
IMPORT Companies;          // Definições Exportadas em Companies estão disponíveis
FloridaCompanies := Companies.File_Company(state='FL');

IMPORT $;                  // Definições Exportadas no Módulo atual estão disponíveis
FloridaCompanies := $.File_Company(state='FL');
```

Estrutura MODULE

A estrutura **MODULE** permite agrupar e fornecer parâmetros para um conjunto de definições ECL relacionadas.

nome [(*parametros*)] := **MODULE**

definições;

END;

- *Nome* – O nome da definição ECL do modulo.
- *parametros* – Os parâmetros disponíveis para todas as *definições*.
- *definições* – As definições ECL que compõem o módulo.

Retomando a extração...

Exemplo de estrutura de dados

```
EXPORT File_Persons := MODULE
  EXPORT Layout := RECORD
    UNSIGNED8 ID;
    STRING15     FirstName;
    STRING25     LastName;
    STRING15     MiddleName;
    STRING2  NameSuffix;
    STRING8  FileDate;
    UNSIGNED2 BureauCode;
    STRING1  MaritalStatus;
    STRING1  Gender;
    UNSIGNED1 DependentCount;
    STRING8  BirthDate;
    STRING42     StreetAddress;
    STRING20     City;
    STRING2  State;
    STRING5  ZipCode;
    END;
  EXPORT File := DATASET('~CLASS::hmw::Intro::Persons', Layout,FLAT);
END;
```

##	id	firstname	lastname	middl...	n...	filedate	bureaucode	marit...	gender	dep...	birthdate	streetaddress	city	state	zipcode
1	9108218085885411565	Cherianne	Khatchatourian	N		19990922	24		M	0		69 BOULDER RIDGE RD # 25A	HAWKINS	WI	54530
2	16505326057200398078	Muyesser	Raplee	X		20001111	353		F	0		55 SWAMP RD	DISTRICT HEIGHT	MD	20747
3	2454818069645923666	Roselin	Viceconte			19990325	344		F	0	19800113	107 HILL TER	ENTERPRISE	OR	97828
4	15880908289586509107	Inda	Provines			20000909	13		U	0		290 W MOUNT PLEASANT AVE	LAVACA	AR	72941
5	6512705660523829539	Inderdeep	Laurence	D		20001228	344		M	0		44 PROSPECT PL	GREENSBORO	FL	32330
6	9193989543268753887	Chrystine	Mangiapane			19990827	315		F	0	19780306	1806 1ST AVE APT 8F	ARVADA	CO	80007
7	12286552293562700162	Adelene	Stock	R		20000827	252		M	0		1117 FARM RD	DOVER	DE	19901
8	11459575736386985069	Mendy	Rufenblanchette			20000903	24		M	0		3 W 83RD ST APT 4C	WILLIAMSTON	SC	29697
9	8053906447536575038	Lannie	Amerantes	I		20001219	313		U	0		200 W 20TH ST APT 909	CHARLESTON	WV	25312
10	484768759680234166	Tare	Gonyeau	T		19930807	48		F	0	19750801	6 CANDLE CT	EL PASO	TX	79924
11	16156125023194932930	Finney	Aristilde	P		19900621	344		M	0	19560920	222 1ST AVE APT 2B	MACON	GA	31220
12	13804468446718957143	Oreoluwa	Marthaler			19931006	358		F	0	19731201	176 CLAREMONT GDNS	AUBURN	ME	04210
13	11995825474648190448	Surge	Abbottkrepp	D		20000308	13		F	0		22 LE PARC CT	TWINSBURG	OH	44087
14	15714117310244664573	Dave	Mcjury			20001129	238		U	0		510 COOPER RD # 1	TACOMA	WA	98402
15	12587451362606486546	Ramsay	Ping			20001129	238		M	0		404 AVENUE L	MESQUITE	NV	89024

Estrutura RECORD

Uma estrutura **RECORD** define o layout de campos do DATASET.

Nome **:= RECORD**
    campos;
END;

- *Nome* – O nome da estrutura RECORD.
- *campos* – O tipo e o nome de cada campo.

Nota: As palavras-chave RECORD e END podem ser substituídas com chaves ({}) e os delimitadores de campos (;) podem ser substituídos por vírgulas (,).

Declaração DATASET

DATASET introduz um novo arquivo de dados no sistema com o layout *record* especificado.

nome **:= DATASET(** *arquivo, record,* **FLAT[THOR] [***opções***]);**
nome **:= DATASET(***arquivo, record,* **CSV [(** *opções* **)]);**
nome **:= DATASET(***arquivo, record,* **XML(** *caminho,***[***opções***]));**
nome **:= DATASET(***arquivo, record,* **JSON(** *caminho,***[***opções***]));**

- ✓ *nome* – O nome da definição pelo qual o arquivo passará a ser referenciado.
- ✓ *arquivo* – Uma constante string contendo o nome do arquivo lógico.
- ✓ *record* – A estrutura RECORD do dataset.

Nota: Um conjunto de registros pode ser definido inline entre colchetes (indicando uma definição set). Dentro dos colchetes, cada registro é delimitado por chaves ({}) e separado por virgulas. Os campos dentro de cada registo são delimitados por virgula.

```
Names := DATASET([{'John','Jones'}, {'Jane','Smith'}], {STRING first_name, STRING last_name});
```

Atenção! Escopo e Nomes de arquivos lógicos

- Nomes de arquivos sempre começam com um escopo (estrutura de diretórios) e terminando com o nome do arquivo.

- O HPCC busca por **arquivos cujos nomes começam com um escopo padrão (THOR)**:

```
'DIR1::DIR2::NomeArquivo'          //dado isso, HPCC procura por:
'THOR::DIR1::DIR2::NomeArquivo'    //esse arquivo
```

- O sinal de "til" (~) indica a **supressão do escopo padrão**:

```
'~DIR1::DIR2::NomeArquivo'  //dado isso, HPCC procura por:
'DIR1::DIR2:: NomeArquivo'  //esse arquivo
```

Já posso ver os meus dados?

A ação **OUTPUT** grava o *recordset* em um arquivo e formatos especificados.

OUTPUT(*recordset* [,*formato*] [,*arquivo* [,OVERWRITE]]**)**

- *recordset* – O conjunto de registros a processar.
- *formato* – O formato de saída dos registros: uma estrutura RECORD previamente definida, ou um layout de registros "on-the-fly" entre chaves ({ }).
- *arquivo* – Nome opcional do arquivo onde os registros serão gravados. Caso seja omitido, os dados formatados são mostrados na linha de comando ou no ECL IDE.
- OVERWRITE – Permite sobreescrever o arquivo, caso ele já exista.

Exemplos de OUTPUT:

```
OUTPUT(File_Accounts.File);

OUTPUT(Persons,{FirstName, LastName}, NAMED('Names_Only'));

OUTPUT(MyRecordset,,'~CLASS::BMF::NewData', OVERWRITE);

//THOR é o formato padrão, mas também é possível gerar saída como:
OUTPUT(MyRecordset,,'~CLASS::BMF::NewData', CSV);
OUTPUT(MyRecordset,,'~CLASS::BMF::NewData', XML);
OUTPUT(MyRecordset,,'~CLASS::BMF::NewData', JSON);
```

Exercício prático:

Exercícios 4 e 5 – Defina seus dados (Persons e Accounts)

- Crie um MODULE
- Use RECORD e DATASET, ambos EXPORTados
- Valide a sintaxe
- Gere uma saída do dataset
- Analise os dados brutos

Desafio: Chicago Crimes

Desafio:

- Definir uma estrutura de dados para o dataset de crimes da polícia de Chicago (data.cityofchicago.org)

##	id	case_number	date	block	iucr	primary_type	description
1	11718445	JC301146	06/10/2019 11:55:00 PM	022XX S SAWYER AVE	0312	ROBBERY	ARMED:KNIFE/CUTTING I
2	11718423	JC301185	06/10/2019 11:55:00 PM	003XX N PINE AVE	0890	THEFT	FROM BUILDING
3	11718364	JC301127	06/10/2019 11:55:00 PM	033XX S MICHIGAN AVE	2093	NARCOTICS	FOUND SUSPECT NARCOTI
4	11718476	JC301140	06/10/2019 11:50:00 PM	057XX S ABERDEEN ST	0497	BATTERY	AGGRAVATED DOMESTIC B
5	11718619	JC301294	06/10/2019 11:47:00 PM	050XX W DIVISION ST	031A	ROBBERY	ARMED: HANDGUN
6	11718392	JC301160	06/10/2019 11:45:00 PM	003XX E 118TH ST	0910	MOTOR VEHICLE THEFT	AUTOMOBILE
7	11718384	JC301137	06/10/2019 11:40:00 PM	080XX S LANGLEY AVE	0486	BATTERY	DOMESTIC BATTERY SIMP
8	11718398	JC301118	06/10/2019 11:31:00 PM	047XX N KEYSTONE AVE	051A	ASSAULT	AGGRAVATED: HANDGUN
9	11718368	JC301109	06/10/2019 11:31:00 PM	096XX S MERRION AVE	1310	CRIMINAL DAMAGE	TO PROPERTY
10	11718393	JC301135	06/10/2019 11:24:00 PM	105XX S SANGAMON ST	0497	BATTERY	AGGRAVATED DOMESTIC B
11	11718444	JC301119	06/10/2019 11:21:00 PM	013XX N PAULINA ST	0860	THEFT	RETAIL THEFT
12	11718411	JC301116	06/10/2019 11:15:00 PM	065XX N FAIRFIELD AVE	0486	BATTERY	DOMESTIC BATTERY SIMP
13	11718351	JC301114	06/10/2019 11:15:00 PM	057XX N WINTHROP AVE	0820	THEFT	$500 AND UNDER
14	11722011	JC301099	06/10/2019 11:15:00 PM	015XX N WASHTENAW AVE	051A	ASSAULT	AGGRAVATED: HANDGUN
15	11718459	JC301152	06/10/2019 11:15:00 PM	054XX W WALTON ST	3710	INTERFERENCE WITH PU...	RESIST/OBSTRUCT/DISAR

Links úteis

- Site principal: **hpccsystems.com**
- Primeiros passos: **hpccsystems.com/Why-HPCC-Systems**
- Treinamento Online: **learn.lexisnexis.com/hpcc**
- Download: **hpccsystems.com/download**
- Fórum da Comunidade: **hpccsystems.com/forums**

Faça parte da Comunidade

Registre-se em hpccsystems.com

Até a próxima aula!!!