



Fundação Vanzolini

# Dominando Big Data com o uso de Plataformas Gratuitas (nível intermediário)

Aula 1

# Agenda da aula 1

- ✓ Apresentação do curso
- ✓ Revisão de conceitos: HPCC Systems e ECL
- ✓ Configuração do ambiente

# Treinamento em ECL/HPCC: [learn.lexisnexus.com/hpcc](http://learn.lexisnexus.com/hpcc)

- Introdução ao ECL (parte 1)
  - Conceitos e consultas
- Introdução ao ECL (parte 2)
  - ETL com ECL
- ECL Avançado (parte 1)
  - Dados relacionais
- ECL Avançado (parte 2)
  - Superarquivos, XML/JSON e PLN
- ECL Aplicado
  - Geração e automação de código ECL
- ROXIE ECL (parte 1)
  - Índices e consultas
- ROXIE ECL (parte 2)
  - Otimização de consultas
- Machine Learning com HPCC Systems
  - Tutoriais para uso de plugins
- Administração de Sistemas
  - Conceitos e operação básica
- HPCC para gestores
  - Visão geral e aplicações da plataforma

# Suporte e operação

- Horários das aulas: 19:00hs às 22:00hs
- Dias das aulas: 18, 20, 22, 25, 27 e 29 de julho
- Certificado USP e badges HPCC Systems
- Computador pessoal (ECL IDE v8.2.18/VSCode/GitPod)
- Cluster: <http://52.167.137.162:8010/>
- Repositório: <https://github.com/HWatanuki/TrainingDominandoBigDataInter>
- Moodle: <https://ead.vanzolini.org.br/>

# Coordenação

- Prof. Hugo Watanuki ([hwatanuki@usp.br](mailto:hwatanuki@usp.br))
  - Doutor em engenharia de produção POLI-USP
  - Engenheiro de suporte técnico na LexisNexis Risk Solutions
- Prof. Renato Moraes ([remo@usp.br](mailto:remo@usp.br))
  - Doutor em administração pela FEA-USP
  - Professor do depto. de engenharia de produção POLI-USP

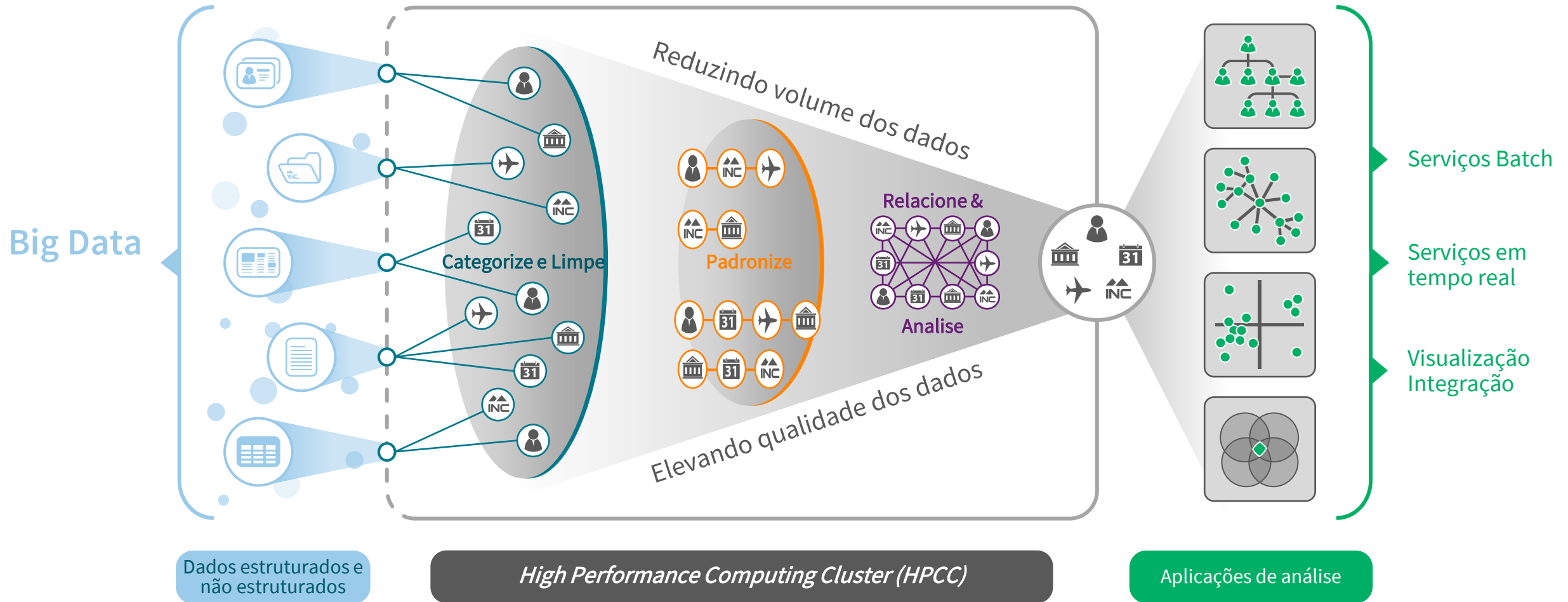


# Introdução

- ✓ Nome
- ✓ Área de atuação
- ✓ Uso e interesse em Big Data

# Revisão de conceitos: HPCC Systems e ECL

# Extract, Transform, Load





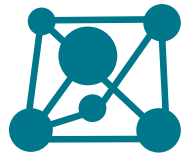
# “Stack” tecnológico



## Ferramentas de consulta e visualização

Entrega online de consultas em *Big Data*

---



## Bibliotecas de *Machine Learning*

Supervisionado, não-supervisionado, aprendizagem profunda

---



## Ferramentas para manipulação de dados

Perfilamento, limpeza, consolidação de dados

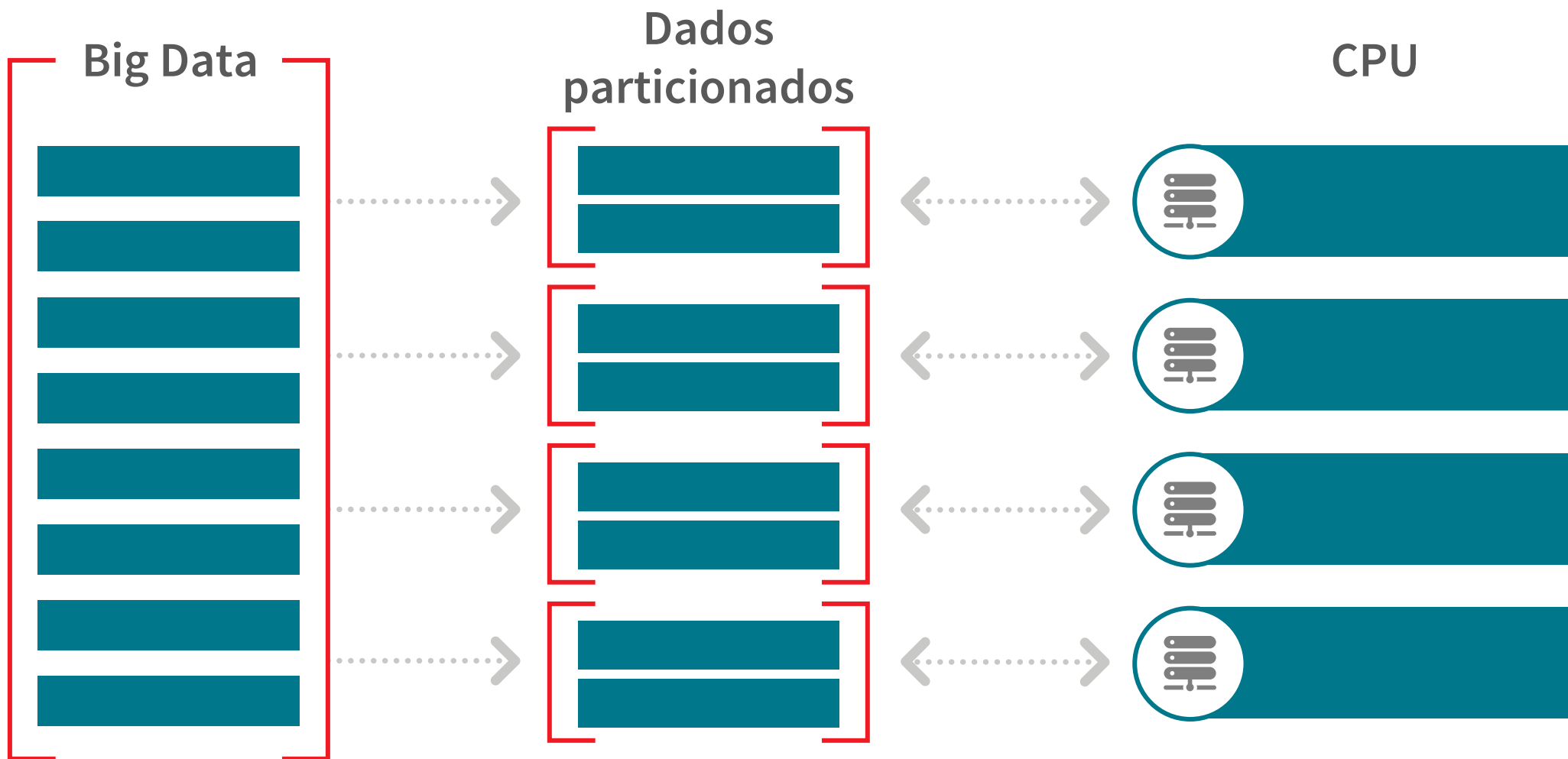
---



## ETL

Extração, transformação e carregamento de dados

# ETL: Supercomputação



**ECL Watch** [User: Wuid, User (ecl:\*, file:\*, dfu:\*, query:\*)...] LOGGED IN AS: [User]

Logical Files | Landing Zones | Workunits | XRef

Logical Files | dcamper::stock\_data::enhanced\_data

Summary | Contents | **Data Patterns** | ECL | DEF | XML | Superfiles | File Parts | Queries | Graphs | Workunit | History

**Report** | Raw Data | Workunit

Refresh | Analyze | Target: thor | Optimize | Delete

Field	Symbol	Min Length	Avg Length	Max Length	Quartile	Cardinality	Popular Patterns
<b># symbol</b>	string16	6	9	16	N/A	N/A	AAAA:AAA (47%) AAAAA:AAAA (42%) AAAAA:AAAAA (4%) AAAA:AA (4%) AAAA:AAAA (2%) AAAAA:AAA (1%) Other (B) (~0%)
<b># trade_year</b>	unsigned2	Mean 2009.3794 Std. Deviation 4.9092	Quartiles 2002 (Min), 2005 (25%), 2009 (50%), 2014 (75%), 2018 (Max)		Cardinality 2002 (8%), 2003 (7%), 2004 (7%), 2005 (6%), 2006 (6%), 2007 (6%), Other (11) (~60%)	Popular Patterns 9999 (100%)	
<b># trade_month</b>	unsigned1	Mean 6.4693 Std. Deviation 3.423	Quartiles 1 (Min), 4 (25%), 6 (50%), 9 (75%), 12 (Max)		Cardinality 3 (9%), 8 (9%), 10 (9%), 5 (9%), 6 (9%), 1 (8%), Other (6) (~47%)	Popular Patterns 9 (76%), 99 (24%)	
<b># trade_day</b>	unsigned1	Mean 15.7231 Std. Deviation 8.7585	Quartiles 1 (Min), 2 (25%), 3 (50%), 4 (75%), 5 (Max)		Cardinality 9 (3%), 11 (3%), 12 (3%), 13 (3%), 14 (3%), 15 (3%), 16 (3%), 17 (3%), 18 (3%), 19 (3%), 20 (3%), 21 (3%), 22 (3%), 23 (3%), 24 (3%), 25 (3%), 26 (3%), 27 (3%), 28 (3%), 29 (3%), 30 (3%), 31 (3%), 32 (3%), 33 (3%), 34 (3%), 35 (3%), 36 (3%), 37 (3%), 38 (3%), 39 (3%), 40 (3%), 41 (3%), 42 (3%), 43 (3%), 44 (3%), 45 (3%), 46 (3%), 47 (3%), 48 (3%), 49 (3%), 50 (3%), 51 (3%), 52 (3%), 53 (3%), 54 (3%), 55 (3%), 56 (3%), 57 (3%), 58 (3%), 59 (3%), 60 (3%), 61 (3%), 62 (3%), 63 (3%), 64 (3%), 65 (3%), 66 (3%), 67 (3%), 68 (3%), 69 (3%), 70 (3%), 71 (3%), 72 (3%), 73 (3%), 74 (3%), 75 (3%), 76 (3%), 77 (3%), 78 (3%), 79 (3%), 80 (3%), 81 (3%), 82 (3%), 83 (3%), 84 (3%), 85 (3%), 86 (3%), 87 (3%), 88 (3%), 89 (3%), 90 (3%), 91 (3%), 92 (3%), 93 (3%), 94 (3%), 95 (3%), 96 (3%), 97 (3%), 98 (3%), 99 (3%), 100 (3%), 101 (3%), 102 (3%), 103 (3%), 104 (3%), 105 (3%), 106 (3%), 107 (3%), 108 (3%), 109 (3%), 110 (3%), 111 (3%), 112 (3%), 113 (3%), 114 (3%), 115 (3%), 116 (3%), 117 (3%), 118 (3%), 119 (3%), 120 (3%), 121 (3%), 122 (3%), 123 (3%), 124 (3%), 125 (3%), 126 (3%), 127 (3%), 128 (3%), 129 (3%), 130 (3%), 131 (3%), 132 (3%), 133 (3%), 134 (3%), 135 (3%), 136 (3%), 137 (3%), 138 (3%), 139 (3%), 140 (3%), 141 (3%), 142 (3%), 143 (3%), 144 (3%), 145 (3%), 146 (3%), 147 (3%), 148 (3%), 149 (3%), 150 (3%), 151 (3%), 152 (3%), 153 (3%), 154 (3%), 155 (3%), 156 (3%), 157 (3%), 158 (3%), 159 (3%), 160 (3%), 161 (3%), 162 (3%), 163 (3%), 164 (3%), 165 (3%), 166 (3%), 167 (3%), 168 (3%), 169 (3%), 170 (3%), 171 (3%), 172 (3%), 173 (3%), 174 (3%), 175 (3%), 176 (3%), 177 (3%), 178 (3%), 179 (3%), 180 (3%), 181 (3%), 182 (3%), 183 (3%), 184 (3%), 185 (3%), 186 (3%), 187 (3%), 188 (3%), 189 (3%), 190 (3%), 191 (3%), 192 (3%), 193 (3%), 194 (3%), 195 (3%), 196 (3%), 197 (3%), 198 (3%), 199 (3%), 200 (3%), 201 (3%), 202 (3%), 203 (3%), 204 (3%), 205 (3%), 206 (3%), 207 (3%), 208 (3%), 209 (3%), 210 (3%), 211 (3%), 212 (3%), 213 (3%), 214 (3%), 215 (3%), 216 (3%), 217 (3%), 218 (3%), 219 (3%), 220 (3%), 221 (3%), 222 (3%), 223 (3%), 224 (3%), 225 (3%), 226 (3%), 227 (3%), 228 (3%), 229 (3%), 230 (3%), 231 (3%), 232 (3%), 233 (3%), 234 (3%), 235 (3%), 236 (3%), 237 (3%), 238 (3%), 239 (3%), 240 (3%), 241 (3%), 242 (3%), 243 (3%), 244 (3%), 245 (3%), 246 (3%), 247 (3%), 248 (3%), 249 (3%), 250 (3%), 251 (3%), 252 (3%), 253 (3%), 254 (3%), 255 (3%), 256 (3%), 257 (3%), 258 (3%), 259 (3%), 260 (3%), 261 (3%), 262 (3%), 263 (3%), 264 (3%), 265 (3%), 266 (3%), 267 (3%), 268 (3%), 269 (3%), 270 (3%), 271 (3%), 272 (3%), 273 (3%), 274 (3%), 275 (3%), 276 (3%), 277 (3%), 278 (3%), 279 (3%), 280 (3%), 281 (3%), 282 (3%), 283 (3%), 284 (3%), 285 (3%), 286 (3%), 287 (3%), 288 (3%), 289 (3%), 290 (3%), 291 (3%), 292 (3%), 293 (3%), 294 (3%), 295 (3%), 296 (3%), 297 (3%), 298 (3%), 299 (3%), 300 (3%), 301 (3%), 302 (3%), 303 (3%), 304 (3%), 305 (3%), 306 (3%), 307 (3%), 308 (3%), 309 (3%), 310 (3%), 311 (3%), 312 (3%), 313 (3%), 314 (3%), 315 (3%), 316 (3%), 317 (3%), 318 (3%), 319 (3%), 320 (3%), 321 (3%), 322 (3%), 323 (3%), 324 (3%), 325 (3%), 326 (3%), 327 (3%), 328 (3%), 329 (3%), 330 (3%), 331 (3%), 332 (3%), 333 (3%), 334 (3%), 335 (3%), 336 (3%), 337 (3%), 338 (3%), 339 (3%), 340 (3%), 341 (3%), 342 (3%), 343 (3%), 344 (3%), 345 (3%), 346 (3%), 347 (3%), 348 (3%), 349 (3%), 350 (3%), 351 (3%), 352 (3%), 353 (3%), 354 (3%), 355 (3%), 356 (3%), 357 (3%), 358 (3%), 359 (3%), 360 (3%), 361 (3%), 362 (3%), 363 (3%), 364 (3%), 365 (3%), 366 (3%), 367 (3%), 368 (3%), 369 (3%), 370 (3%), 371 (3%), 372 (3%), 373 (3%), 374 (3%), 375 (3%), 376 (3%), 377 (3%), 378 (3%), 379 (3%), 380 (3%), 381 (3%), 382 (3%), 383 (3%), 384 (3%), 385 (3%), 386 (3%), 387 (3%), 388 (3%), 389 (3%), 390 (3%), 391 (3%), 392 (3%), 393 (3%), 394 (3%), 395 (3%), 396 (3%), 397 (3%), 398 (3%), 399 (3%), 400 (3%), 401 (3%), 402 (3%), 403 (3%), 404 (3%), 405 (3%), 406 (3%), 407 (3%), 408 (3%), 409 (3%), 410 (3%), 411 (3%), 412 (3%), 413 (3%), 414 (3%), 415 (3%), 416 (3%), 417 (3%), 418 (3%), 419 (3%), 420 (3%), 421 (3%), 422 (3%), 423 (3%), 424 (3%), 425 (3%), 426 (3%), 427 (3%), 428 (3%), 429 (3%), 430 (3%), 431 (3%), 432 (3%), 433 (3%), 434 (3%), 435 (3%), 436 (3%), 437 (3%), 438 (3%), 439 (3%), 440 (3%), 441 (3%), 442 (3%), 443 (3%), 444 (3%), 445 (3%), 446 (3%), 447 (3%), 448 (3%), 449 (3%), 450 (3%), 451 (3%), 452 (3%), 453 (3%), 454 (3%), 45		

# Bibliotecas de *Machine Learning*



## Não supervisionado

### Clusterização

DBSCAN  
K-Means

### PLN

Text Vectors



## Supervisionado

### Classificação

SVM  
Árvores de decisão  
Regression logística  
Classification Forest

### Regressão

Regressão linear  
GLM  
Regression Forest



## Redes neurais & Deep Learning

Autoencoders

Redes neurais convolucionais

Redes neurais recorrentes

Perceptrons



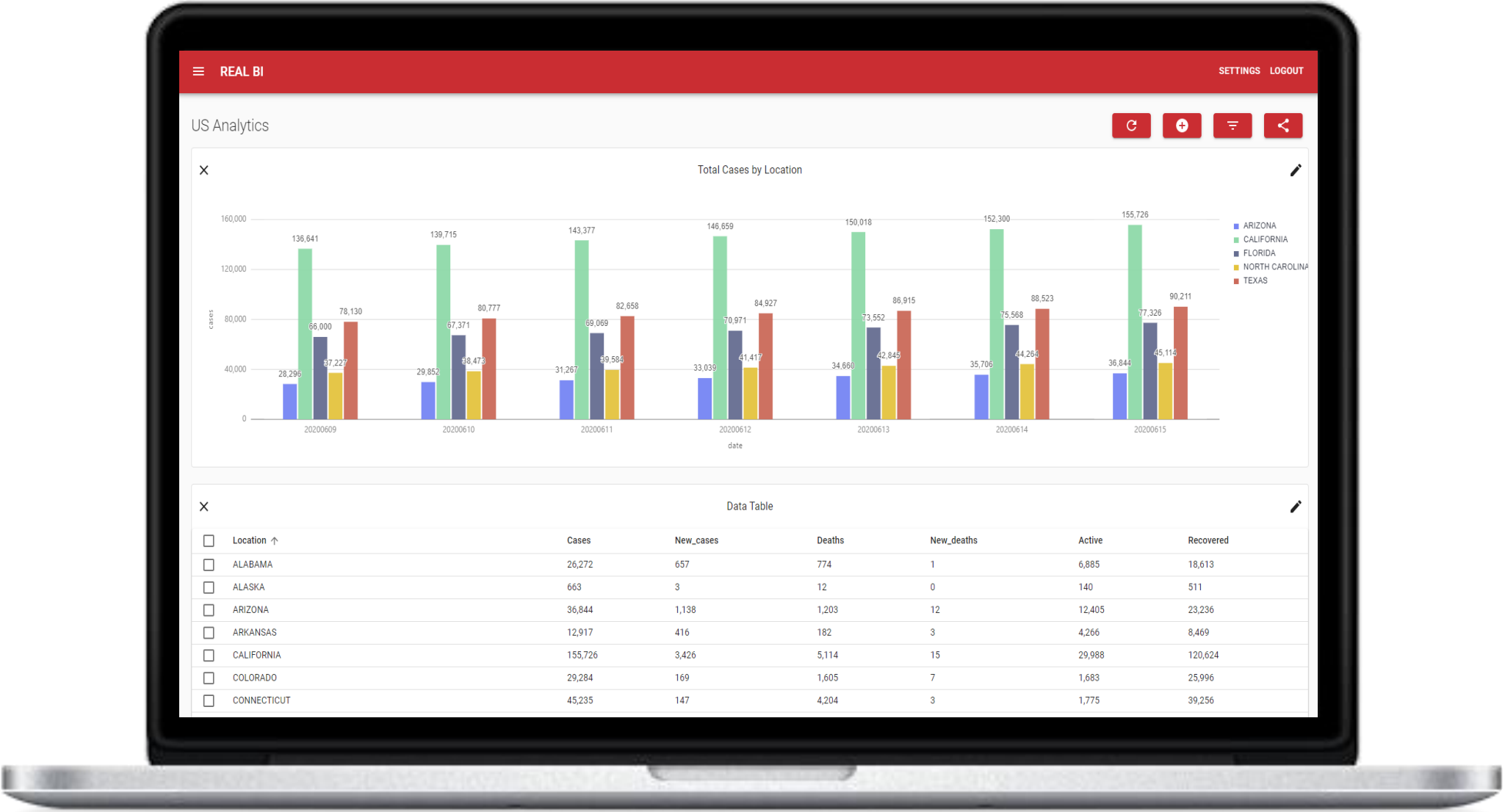
## Métodos ensemble

Random Forest

Gradient Boosted  
Forest

Gradient Boosted  
Trees

# Ferramentas de Consulta e Visualização



# Conceitos básicos de ECL

- Paradigma declarativo (não-procedural)
- Estrutura básica: **Nome := Expressão ;**
- ECL não é sensível a caixa alta/baixa
- Espaço em branco é ignorado
- 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 diretório

# 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

# Tipos de dados primitivos

## BOOLEAN

```
BOOLEAN IsFloridian := 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.birthdate>19600101;
```

## Valor único (*value*)

```
MaleValue := 'M';
```

## Conjunto de valores (*set*)

```
GenderValues := ['M', 'F'];
```

## Conjunto de registros (*recordset*)

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

```
MalePeople := People(Gender = MaleValue);
```

```
FemaleMalePeople := People(Gender IN GenderValues);
```

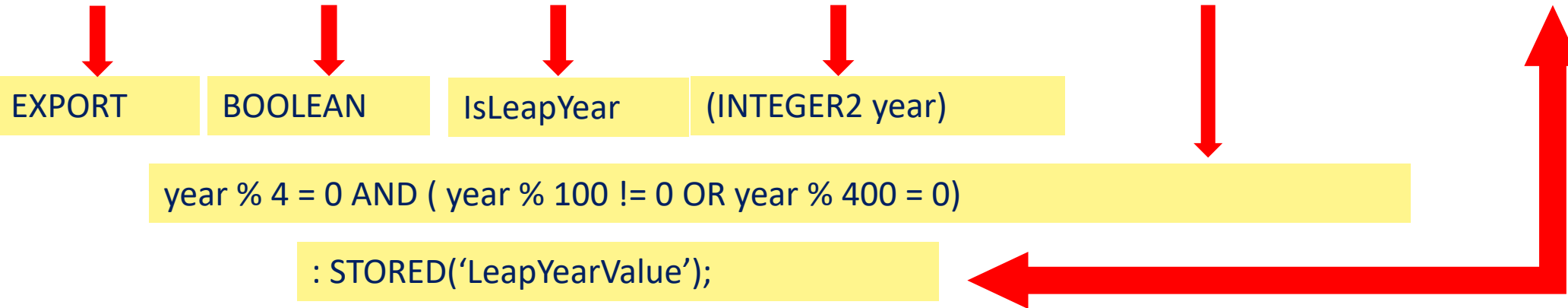
## People

##	firstname	lastname	middlename	namesuffix	filedate	bureaucode	maritalstatus	gender	dependentcount	birthdate	streetaddress
1	Cherianne	Khatchatourian	N		19990922	24		M	0		69 BOULDER RIDGE RD # 25
2	Muyesser	Raplee	X		20001111	353		F	0		55 SWAMP RD
3	Roselin	Viceconte			19990325	344		F	0	19800113	107 HILL TER
4	Inda	Provines			20000909	13		U	0		290 W MOUNT PLEASANT AVE
5	Inderdeep	Laurence	D		20001228	344		M	0		44 PROSPECT PL
6	Chrystine	Mangiapane			19990827	315		F	0	19780306	1806 1ST AVE APT 8F
7	Adelene	Stock	R		20000827	252		M	0		1117 FARM RD
8	Mendy	Rufenblanchette			20000903	24		M	0		3 W 83RD ST APT 4C
9	Lannie	Amerantes	I		20001219	313		U	0		200 W 20TH ST APT 909
10	Tare	Gonyeau	T		19930807	48		F	0	19750801	6 CANDLE CT

# 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 disponíveis 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**

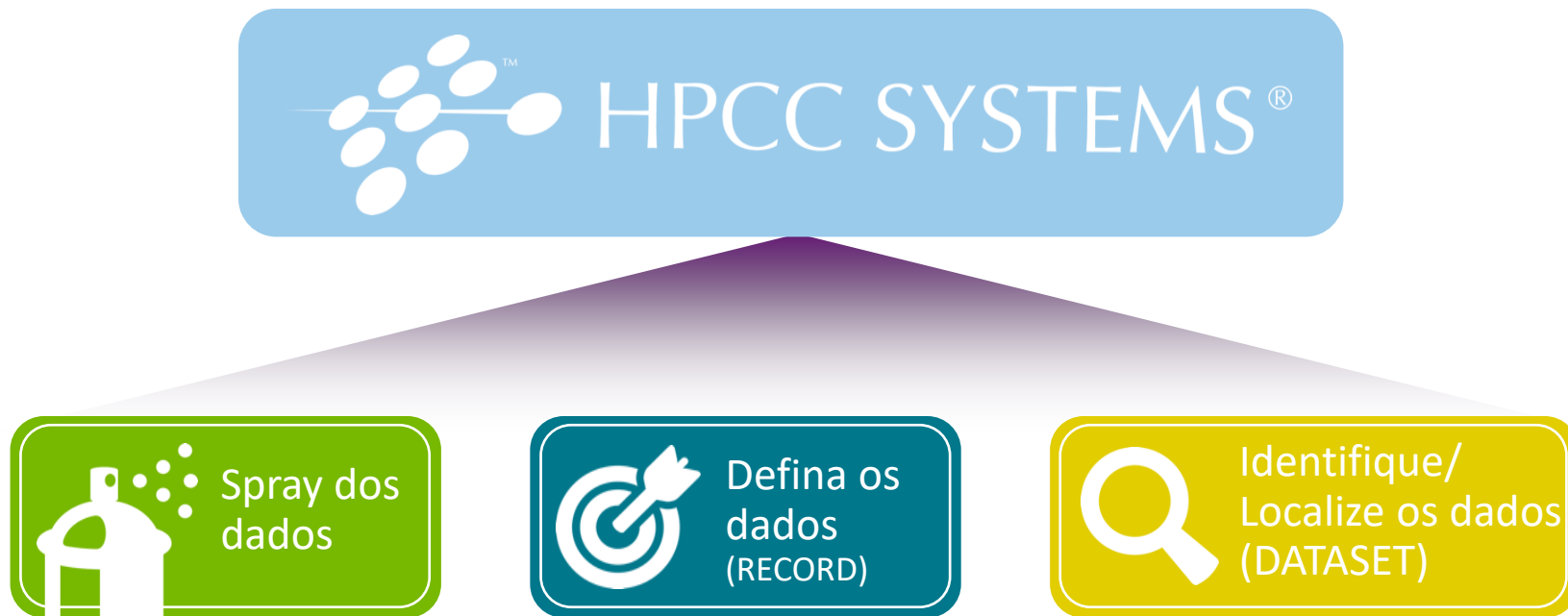
A estrutura **MODULE** permite agrupar e fornecer parâmetros para um conjunto de definições ECL relacionadas.

```
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.

# Começando a trabalhar com dados

Antes de começar a trabalhar com qualquer arquivo de dados na plataforma HPCC Systems, três passos devem ser executados:



# 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 vírgulas. Os campos dentro de cada registro são delimitados por vírgula.

```
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 sobrescrever 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);
```

# Filtragem simples de dados

- Uma expressão booleana entre parênteses após um Dataset/Recordset é um **filtro**
- Múltiplos filtros podem ser especificados usando uma vírgula (,) ou usando “AND”

```
ValidNames := People(Lastname >= 'T', Lastname < 'U');
```

```
ValidTrades := Trades(Rate >= 7);
```

```
ValidPeople := People(NOT IsSeniorCitizen AND Lastname < 'U');
```

```
ValidPeople2 := People(state IN ['FL','NY']);
```

# Operadores de comparação

Equivalência	.....	=
Diferente de	.....	<>
Diferente de	.....	!=
Menor que	.....	<
Maior que	.....	>
Menor ou igual que	.....	<=
Maior ou igual que	.....	>=
Comparação de equivalência	.....	<=> retorna -1, 0, or 1

# Operadores aritméticos:

Divisão	.....	/
Divisão Inteira	.....	DIV
Divisão Módulo	.....	%
Multiplicação	.....	*
Adição	.....	+
Subtração	.....	-

**Nota: Qualquer divisão por (0) resulta em zero (0).  
Esse comportamento pode ser alterado especificando-se  
`#OPTION ('divideByZero', 'fail');` //Aborta e reporta erro**

# Funções de agregação

**COUNT**(*recordset*)

**COUNT**(*listavalores*)

**MAX**(*recordset* , *campo* )

**MAX**(*listavalores*)

**MIN**(*recordset* , *campo* )

**MIN**(*listavalores*)

**SUM**(*recordset* , *campo* )

**SUM**(*listavalores*)

**AVE**(*recordset* , *campo* )

**AVE**(*listavalores*)

- *recordset* – O set ou conjunto de registros a serem processados.
- *campo* – O campo ou expressão a partir dos quais o valor deve ser calculado.
- *listavalores* – Uma lista de expressões separadas por vírgula a partir dos quais o valor deve ser calculado. Também pode ser um SET de valores.

```
OldCount:=COUNT(People(IsSeniorCitizen));
```

```
MaxVal := MAX(People, People.age);
```

```
MinVal1 := MIN(People, People.age);
```



# Tarefa: Lending Club

# Lending Club

- LendingClub é uma empresa americana de empréstimos peer-to-peer, com sede em São Francisco, Califórnia.
- A empresa afirma que US \$ 15,98 bilhões em empréstimos foram originados por meio de sua plataforma até 31 de dezembro de 2015.
- Dados: <https://www.kaggle.com/ethon0426/lending-club-20072020q1>

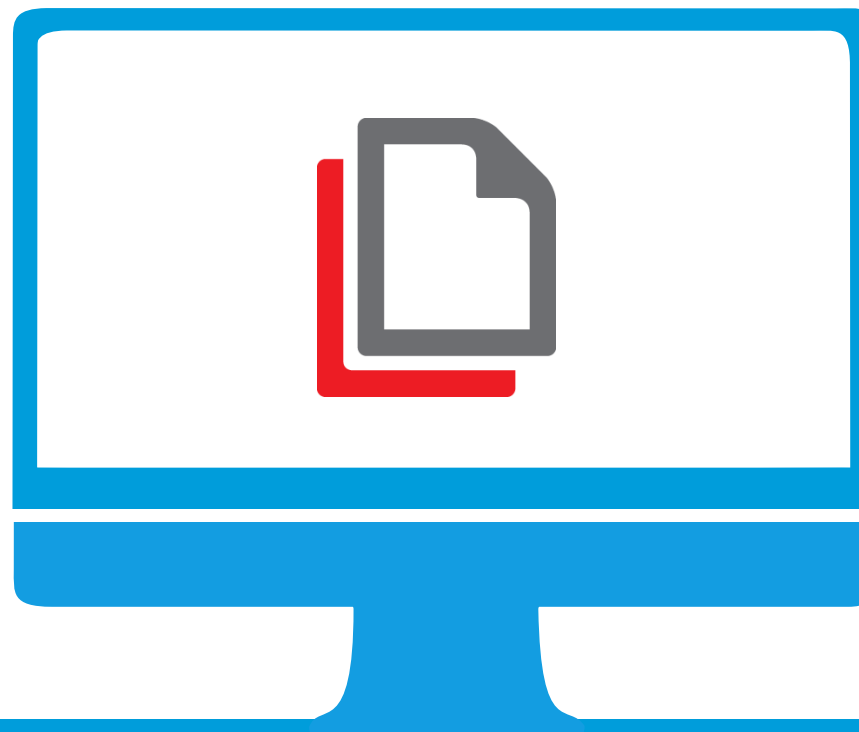
# O que esperamos descobrir

- Identificar um modelo de predição baseado em atributos de operações financeiras que permitam aferir o risco de um pedido de crédito.
- As principais perguntas a serem respondidas são:
  - Quais variáveis podem ser utilizadas para estabelecer com precisão conhecida o risco de crédito?
  - Quais as características do modelo estatístico capaz de prever o risco de crédito?

# Exercício prático:

## Faça a extração do dataset do Lending Club

- Spray
- Estrutura RECORD
- Declaração DATASET



# Até a próxima aula!!!

