ECL\_Cheat\_Sheet.pdf Friday, May 15, 2020

# **ECL Cheat Sheet**

A simple introduction to ECL — so you can master it with ease. https://github.com/hpcc-systems/HPCC-ECL-Training/blob/master/CheatSheet/ECL\_Cheat\_Sheet.pdf



#### Dataset

A representation of data on disk or created in memory. Most ECL functions return a DATASET.

INPUT			
pickup_dt	Fare		
2019-01-01 01:08:56	25.10		

40.15

#### Summarize

Provides a large set of functions to summarize values in a dataset. Can be used in functions with GROUP and TABLE to create Pivots.

OUTPUT		
typ	val	
sum	65.25	
ave	32,63	
min	25.1	
max	40.15	
count	2	

#### Group

2019-01-01 02:10:22

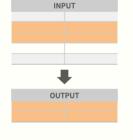
Easily work with cross tab functionality by using GROUP and TABLE functions.

```
Layout := RECORD
                                                                                                      INPUT
    STRING10 pickup_date;
    DECIMAL8_2 fare;
                                                                            pickup_date
                                                                                                      fare
                                                                                                                      distance
    DECIMAL8_2 distance;
                                                                            2019-01-01
                                                                                                      25.10
                                                                            2019-01-01
                                                                                                      40.15
ds := DATASET([{'2019-01-01', 25.10, 5},
                  {'2019-01-01', 40.15, 8}, {'2019-01-02', 30.10, 6},
                                                                            2019-01-02
                                                                                                      30.10
                                                                                                      25.15
                                                                            2019-01-02
                 {'2019-01-02', 25.15, 4}], Layout);
crossTabLayout := RECORD
   ds.pickup_date;
   avgFare := AVE(GROUP, ds.fare);
                                                                                            OUTPUT
   totalFare := SUM(GROUP, ds.fare);
varianceFare := VARIANCE(GROUP, ds.fare);
coVarianceFareDist := COVARIANCE(GROUP,
                                                           pickup_date avgfare totalfare
                                                           2019-01-01 32.625 62.25
                                                                                                       11.28
                           ds.fare, ds.distance);
                                                           2019-01-02 27.625 55.25
   correlateFareDist := CORRELATION(GROUP
                            ds.fare, ds.distance);
crossTabDs := TABLE(ds, crossTabLayout, pickup_date);
OUTPUT(crossTabDs);
```

#### **Observe Subset**

Select a subset of rows in a dataset for observation.

```
Layout := RECORD
    STRING10 pickup_date;
    DECIMAL8_2 fare;
    DECIMAL8_2 distance;
ds := DATASET([{'2019-01-01', 25.10, 5},
               {'2019-01-01', 40.15, 8},
                {'2019-01-02', 30.10, 6},
               {'2019-01-02', 25.15, 4}], Layout);
//Filter records by fields
filterDs := ds(pickup_date='2019-01-01');
//Remove duplicate records
dedupDs := DEDUP(SORT(ds, pickup_date),
pickup date):
choosenDs := CHOOSEN(ds, 2);//Return top 2 records
//Return top N records after sorting
topDs := TOPN(ds, 2, pickup_date);
//Return sample part of set
sampleDs := SAMPLE(ds, 2, 1);//return every 2nd
//Return sample set of records
enthDs := ENTH(ds, 1, 2, 1);//1 out of every 2
OUTPUT(filterDs):
OUTPUT(dedupDs);
OUTPUT(topDs);
OUTPUT(sampleDs);
OUTPUT(enthDs);
```



## **Shape with Project**

Used to transform datasets with the same number of records but transformed columns.



01			
pickup_datetime	fare	dist	
2019-01-01 10:00:00	25.10	5	
2019-01-01 11:00:00	40.15	8	
2019-01-02 10:00:00	30.10	6	
2019-01-02 10:00:00	25.15	4	
2019-01-02 10:00:00	25.15	4	

INPUT



OUTPUT				
	20190101	100000	25.10	5
	20190101	110000	40.15	8
	20190102	100000	30.10	6
	20190102	110000	25.15	4

ECL\_Cheat\_Sheet.pdf Friday, May 15, 2020

### **Shape with Rollup** In one way, ROLLUP is used combine related records into a single aggregate record, like an aggregating SQL self join. Layout := RECORD STRING10 pickup\_date; DECIMAL8 2 fare; DECIMAL8\_2 mileageDeduction := 0; {'2019-01-02', 25.15, 4}], Layout); outputDs := ROLLUP(SORT(inputDs, pickup\_date), LEFT.pickup\_date=RIGHT.pickup\_date, M(Layout, SELF.pickup date := LEFT.pickup\_date, SELF.fare := LEFT.fare + T.fare. SELF.distance := LEFT.distance RIGHT.distance. SELF.mileageDeduction := self.distance \* 0.545)); OUTPUT(outputDs);

# **Shape Parent Child Rollup**

Rollup records into a parent child layout.

INPUT			
pickup_datetime	fare	distance	
2019-01-01 10:00:00	25.10	5	
2019-01-01 11:00:00	40.15	8	
2019-01-02 10:00:00	30.10	6	
2019-01-02 10:00:00	25.15	4	

# **Shape with Normalize**

Break contents of record into normal form.

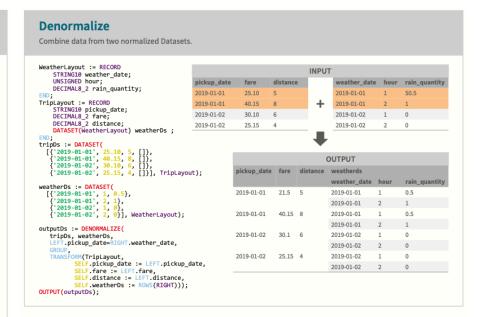
```
IMPORT Std:
InputLayout := RECORD
   UNSIGNED ride id;
   STRING passenger_state;
{4, 'drunk smell'}], InputLayout);
OutputLayout := RECORD
    UNSIGNED ride id:
    STRING100 word;
wordDs := NORMALIZE(inputDs.
STD.Str.WordCount(LEFT.passenger_state),
                    M(OutputLayout,
                     SELF.ride_id :=
                     LEFT.ride id.
                     SELF.word :=
STD.Str.ToUpperCase(
STD.Str.GetNthWord(LEFT.passenger_state,
COUNTER))));
```

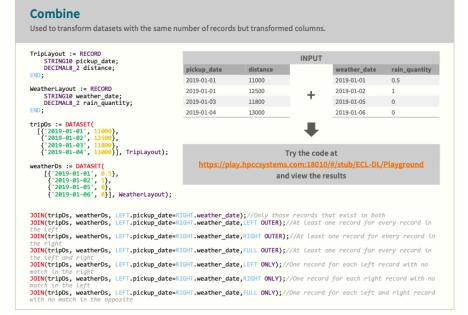
	INPUT		OU	TPUT
ride_id	passenger_state	,	ride_id	word
1	group cool talkative		1	GROUP
2	calm quiet		1	COOL
3	temper nasty		1	TALKATIVE
4	drunk smell		2	CALM
			2	QUIET
			3	TEMPER
			3	NASTY
			4	DRUNK
			4	SMELL

2019-01-01	65.25 13	7.09			
2019-01-02	55.25 20	5.45			
OUTPUT: S	HAPING WITH PA	RENT CH	ILD ROLLUP		
pickup_date	trips	trips			
	pickup_date	fare	distance		
2019-01-01	2019-01-01	25.1	5		
	2019-01-01	40.15	8		
2019-01-02	2019-01-02	30.1	6		
	2019-01-02	25.15	4		

**OUTPUT: SHAPING WITH ROLLUP** 

pickup date fare distance mileagededuction





Try these examples at https://play.hpccsystems.com:18010/#/stub/ECL-DL/Playground