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

HPCC SYSTEMS®

Dataset

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

```
Layout := RECORD
    STRING pickup_dt;
    DECIMAL8_2 fare;
END;

//Reading embedded data
memDs := DATASET([{'2019-01-01 01:08:56', 25.10},
                  {'2019-01-01 02:10:22', 40.15}],
Layout);

OUTPUT(memDs);

//Reading CSV file data
fileDs := DATASET(

'~tutorials::cheatsheet::in::sample_trip_1.csv',
          Layout, CSV);

OUTPUT(fileDs);
```

INPUT

pickup_dt	Fare
2019-01-01 01:08:56	25.10
2019-01-01 02:10:22	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.

```
Layout := RECORD
    STRING pickup_dt;
    DECIMAL8_2 fare;
END;

ds := DATASET([{'2019-01-01 01:08:56', 25.10},
               {'2019-01-01 02:10:22', 40.15}],
Layout);

sumVal := SUM(ds, ds.fare);
avgVal := AVE(ds, ds.fare);
minVal := MIN(ds, ds.fare);
maxVal := MAX(ds, ds.fare);
countVal := COUNT(ds);

OUTPUT(DATASET([{'sum',sumVal},
                {'avg',avgVal},
                {'min', minVal},
                {'max', maxVal},
                {'count', countVal}],
                {String typ, DECIMAL8_2 val}));
```

OUTPUT

typ	val
sum	65.25
ave	32.63
min	25.1
max	40.15
count	2

Group

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

```
Layout := RECORD
    STRING10 pickup_date;
    DECIMAL8_2 fare;
    DECIMAL8_2 distance;
END;

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);

crossTabLayout := RECORD
   ds.pickup_date;
   avgFare := AVE(GROUP, ds.fare);
   totalFare := SUM(GROUP, ds.fare);
   varianceFare := VARIANCE(GROUP, ds.fare);
   coVarianceFareDist := COVARIANCE(GROUP,
                            ds.fare, ds.distance);
   correlateFareDist := CORRELATION(GROUP,
                            ds.fare, ds.distance);
END;

crossTabDs := TABLE(ds, crossTabLayout, pickup_date);
OUTPUT(crossTabDs);
```

INPUT

pickup_date	fare	distance
2019-01-01	25.10	5
2019-01-01	40.15	8
2019-01-02	30.10	6
2019-01-02	25.15	4

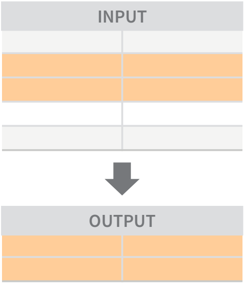
OUTPUT

pickup_date	avgfare	totalfare	variance fare	covariancefaredist	correlatefaredist
2019-01-01	32.625	62.25	56.62	11.28	1
2019-01-02	27.625	55.25	6.125	2.47	1

Observe Subset

Select a subset of rows in a dataset for observation.

```
Layout := RECORD
    STRING10 pickup_date;
    DECIMAL8_2 fare;
    DECIMAL8_2 distance;
END;

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);

//Returns top N records
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
record

//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.

```
IMPORT Std;

InputLayout := RECORD
    STRING pickup_datetime;
    DECIMAL8_2 fare;
    DECIMAL8_2 distance;
END;

OutputLayout := RECORD
    Std.Date.Date_t pickup_date;
    Std.Date.Time_t pickup_time;
    DECIMAL8_2 fare;
    DECIMAL8_2 distance;
END;

inputDs := DATASET([{'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 11:00:00', 25.15, 4}],
                       InputLayout);

outputDs := PROJECT(inputDs, TRANSFORM(OutputLayout,
   SELF.pickup_date :=
Std.Date.FromStringToDate(LEFT.pickup_datetime[..10],
'%Y-%m-%d'),
   SELF.pickup_time :=
Std.Date.FromStringToTime(LEFT.pickup_datetime[12..],
'%H:%M:%S'),
   SELF.fare := LEFT.fare,
   SELF.distance := LEFT.distance));
OUTPUT(outputDs);
```

INPUT

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

OUTPUT

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