

SHIP: A little logistics language

By Alex Kamil (ak2834@columbia.edu)

SHIP is inspired by UPS (United Parcel Service) - it is a simple logistics language that allows extracting a subset of data from a dataset, packaging it into manageable containers and sending to a target destination. It also allows receiving packages and extracting their content. The language supports some rudimentary synchronization primitives such as waiting for receiver's signature on delivery, and notifying the sender on delivery status. The delivery and data manipulation tasks can be combined into a pipeline.

Some code examples:

Begin Task1:

```
    MyBasket: Select * From Fruits where name= Jincheng and location =  
Jiangxi;
```

```
    MyPackage: Send MyBasket to Helsinki by Sea;  
Go;
```

Begin Task2 like Task1:

```
    MyBasket: Select * from Task1:MyBasket;  
    MyPackage: Send to Lisbon by Air; Get Signature; Notify ALL on Success;  
Go;
```

Begin Task3 like Task1:

```
    Wait for Task2: Notify Sender on Timeout;  
    Send to New York;  
Go;
```

The language supports a subset of SQL for data filtering and extraction, similar to Apache Hive.

```
SELECT [ALL | DISTINCT]select_expr, select_expr, ...  
FROM table_reference  
[WHERE condition]  
[GROUP BY col]  
  | [SORT BY col_list]  
]  
[LIMIT number]
```

- SELECT statement can be part of a union query or a subquery of another query.
- table_reference indicates the input to the query. It can be a regular table, a join construct or a subquery.
- The ALL and DISTINCT options specify whether duplicate rows should be returned
- Sample queries:

```
ship> SELECT * FROM t1
```

```
ship> SELECT * FROM sales WHERE amount > 10 AND region = "US"
```