Transactional dataset

TRANSACTION ≡ set of items (plus other info)

E.g. Set of products bought by a customer in a visit to a, possibly online, store

MARKET BASKET ANALYSIS: concerns the analysis of transactional data, typically from the retail world

Transactional dataset

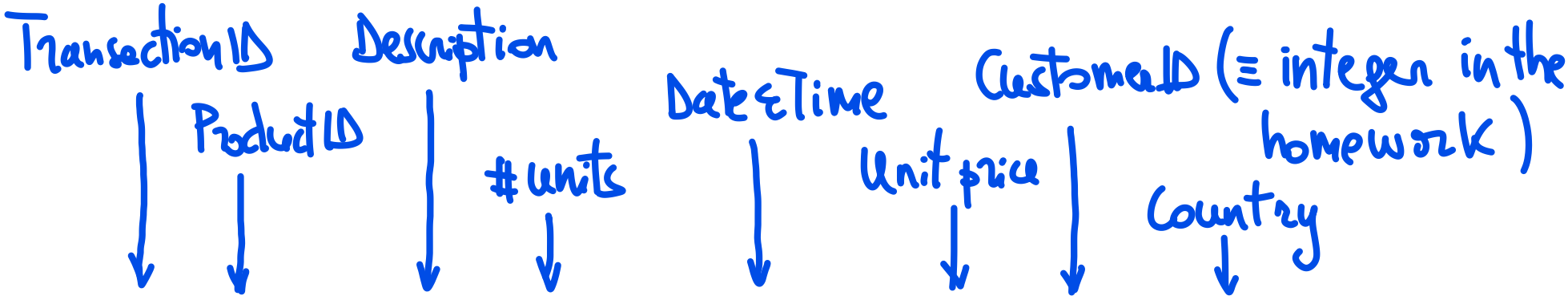
In the homework, a transaction consists of

* TransactionID
* Set of products. For each product
  - ProductID
  - Description
  - # units
  - unit price
* Date and time of the transaction
* Customer ID
* Country where the customer resides

REPRESENTATION: file where each transaction occupies multiple rows, one row per product

Example

4 Transactions: TID1, TID2, TID3, TID4




```
TID1,P1,Pencil box,3,2/1/2010 8:45,0.65,CUST1,France
TID1,P2,Eraser box,13,2/1/2010 8:45,1.00,CUST1,France
TID1,P3,Dvd Batman,-3,2/1/2010 8:45,3.15,CUST1,France

TID2,P1,Pencil box,3,4/1/2010 10:05,0.65,CUST1,France

TID3,P1,Pencil box,3,2/2/2010 18:30,0.65,CUST2, Italy
TID3,P2,Eraser box,3,2/2/2010 18:30,1.00,CUST2,Italy

TID4,P1,Pencil box,3,21/2/2010 9:55,0.65,CUST3,United Kingdom
TID4,P3,Dvd Batman,11,21/2/2010 9:55,0.65,CUST3,United Kingdom
```

Note that all fields are comma separated and, in the homework, CustomerID is an integer

Homework 1: task

* COMMAND LINE ARGUMENTS
  - k ≡ # partitions
  - H ≡ # products with highest popularity to find
  - S ≡ country
  - file_path

* Read file into an RDD of Strings (1 string ≡ 1 row)
* Compute the set of (Product, customer) pairs that satisfy the following conditions

Homework 1: task

- Customer from country S must have bought at least once a positive number of units of the product (i.e., must ignore rows with negative #units)
- no duplicates (do not use Spark method distinct())

* Compute (Product, Popularity) pairs, where Popularity = # customers detected in previous step using 2 approaches (→ 2 RDDs)

Homework 1: task

1) using mapPartitionsToPair / mapPartitions
2) using reduceByKey

* Extract top-H products based on popularity.
(if H=0 print all (Product, Popularity) pairs from the 2 RDDs computed before)

IMPORTANT: Cannot gether together all rows relative to the same product (TOO MANY), but can assume that there are few rows for each (Product, Customer) pair