Transactional dataset

Transaction ≡ set of items (plus other info)

E.g. set of products bought by a customer in a visit to an (online) store

Market Basket Analysis: concerns the analysis of transactional data, typically from retail world

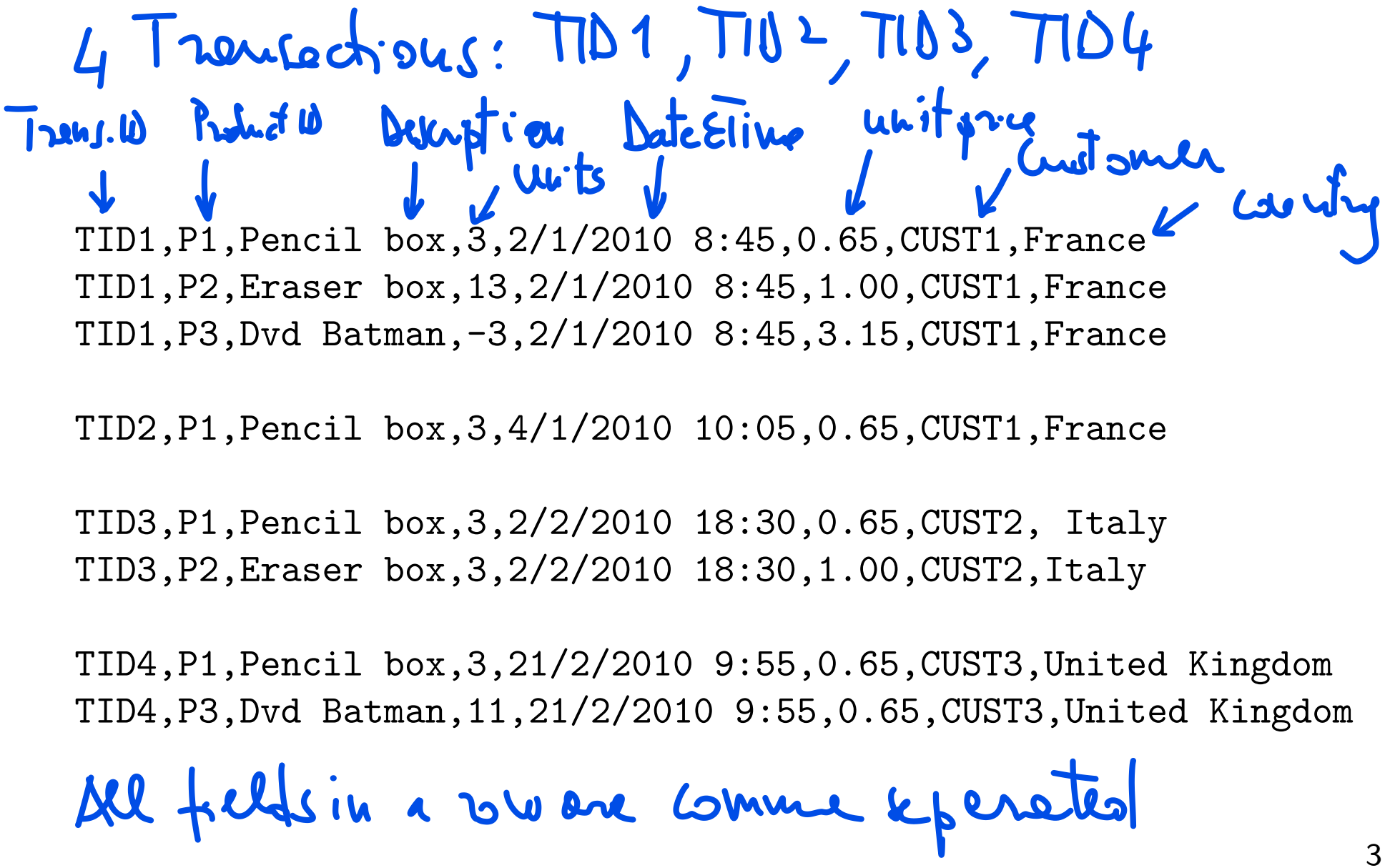
Transactional dataset

In the homework, a transaction consists of

* Transaction ID
* Set of products. For each product:
  - Product ID
  - 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
Trans.ID
Product ID
Description
Units
DateTime
unitprice
Customer
Country
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
All fields in a row are comma separated

Homework 1: task

* Command line arguments:
  - K ≡ # partitions
  - H ≡ goal is to identify top-H products by popularity
  - S ≡ country (S = "all" ⇒ all countries)
  - file-path
* Read file into an RDD of strings (1 string = 1 row)

Homework 1: task

* Compute set of (Product, Customer) pairs that satisfy:
  - Customer from country S and must have bought at least once the product with positive quantity
  - no duplicates (don't use spark method distinct())

* Compute (Product, Popularity) pairs
  → # customers from previous step

Homework 1: task

Repeat the computation 2 times (=> 2 RDDs)
- using mapPartitionsToPair / mapPartitions
- using reduceByKey

* Extract Top-H products based on popularity (if H=0 must print all (products, popularity) pairs from the 2 RDD computed before)

IMPORTANT: cannot gather together all rows relative to the same product (too many), but you can assume few rows for (product, customer)