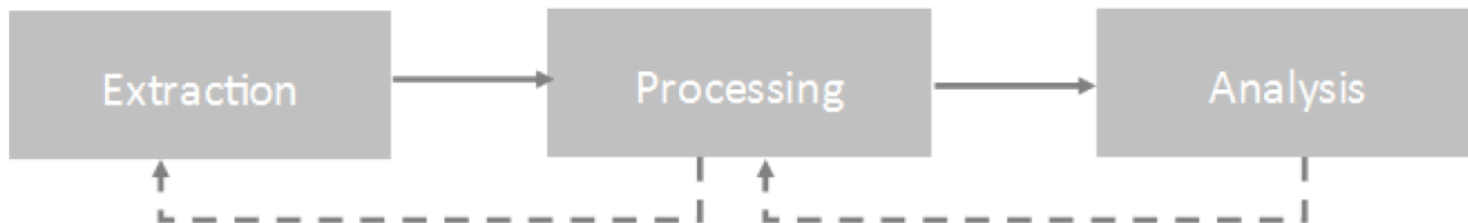


Case Study: Olist.com Event Log Creation

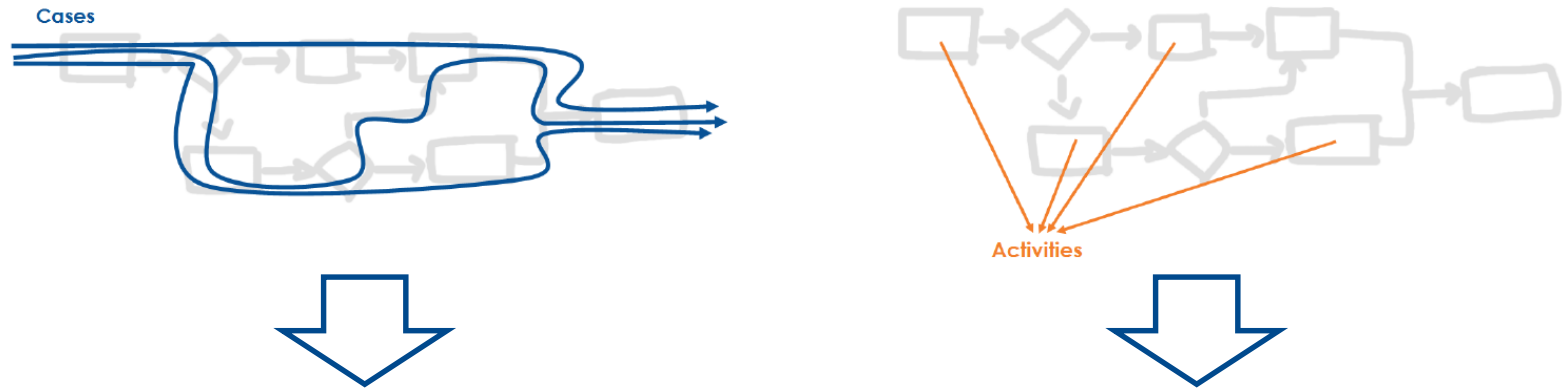
Rene Laub, Bernd Skiera
Goethe-University
rlaub@wiwi.uni-frankfurt.de

Process Analysis Workflow



1. **Extraction:** transform raw data into event data
2. **Processing:** enrich and filter event data
3. **Analysis:** gain useful insights in the process

Event Log Structure



order number	activity	timestamp	user	product	quantity
9901	register order	22-1-2014@09.15	Sara Jones	iPhone5S	1
9902	register order	22-1-2014@09.18	Sara Jones	iPhone5S	2
9903	register order	22-1-2014@09.27	Sara Jones	iPhone4S	1
9901	check stock	22-1-2014@09.49	Pete Scott	iPhone5S	1
9901	ship order	22-1-2014@10.11	Sue Fox	iPhone5S	1
9903	check stock	22-1-2014@10.34	Pete Scott	iPhone4S	1
9901	handle payment	22-1-2014@10.41	Carol Hope	iPhone5S	1
9902	check stock	22-1-2014@10.57	Pete Scott	iPhone5S	2
9902	cancel order	22-1-2014@11.08	Carol Hope	iPhone5S	2
...

case id

activity name

timestamp

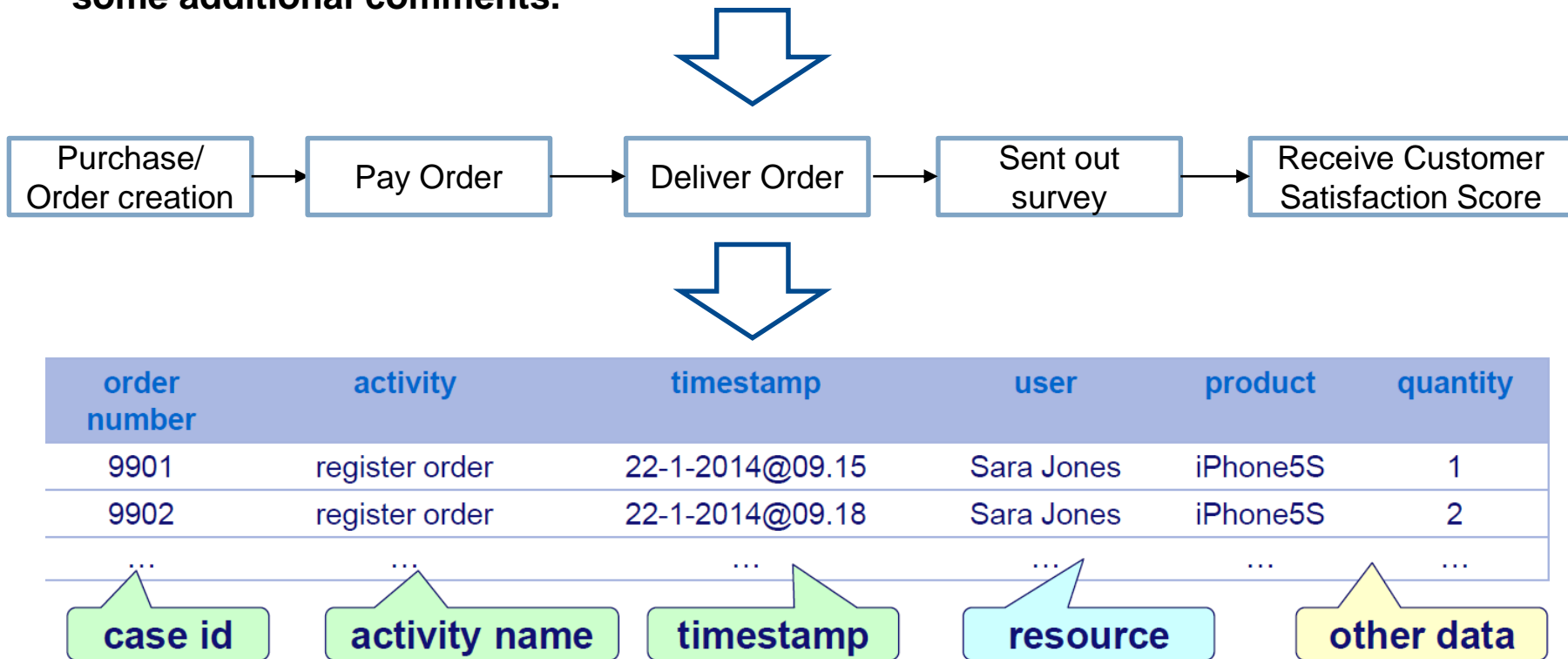
resource

other data

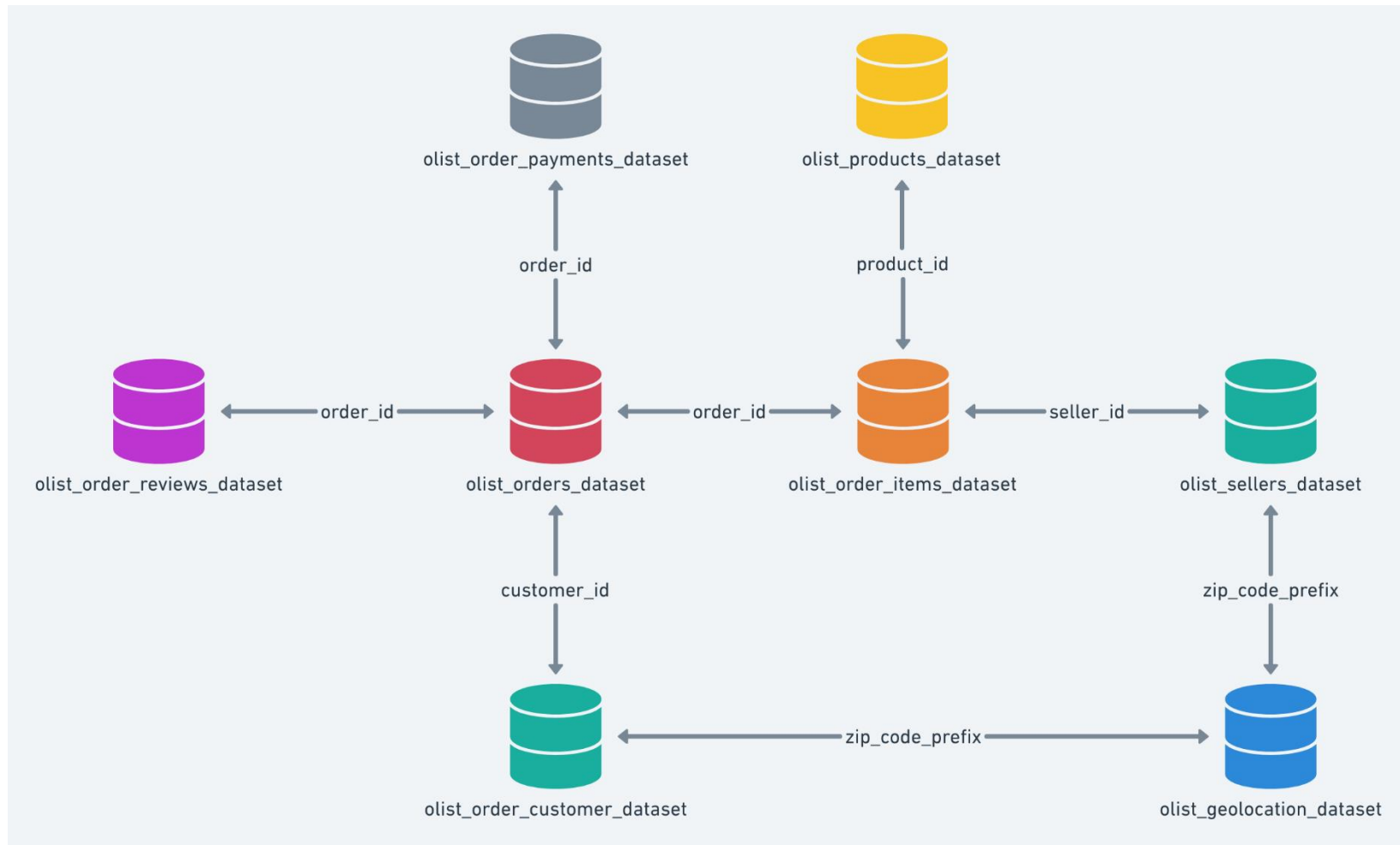
Source: Van Der Aalst, W. (2011). Process mining: discovery, conformance and enhancement of business processes (Vol. 2). Heidelberg: Springer.

Olist Order Process

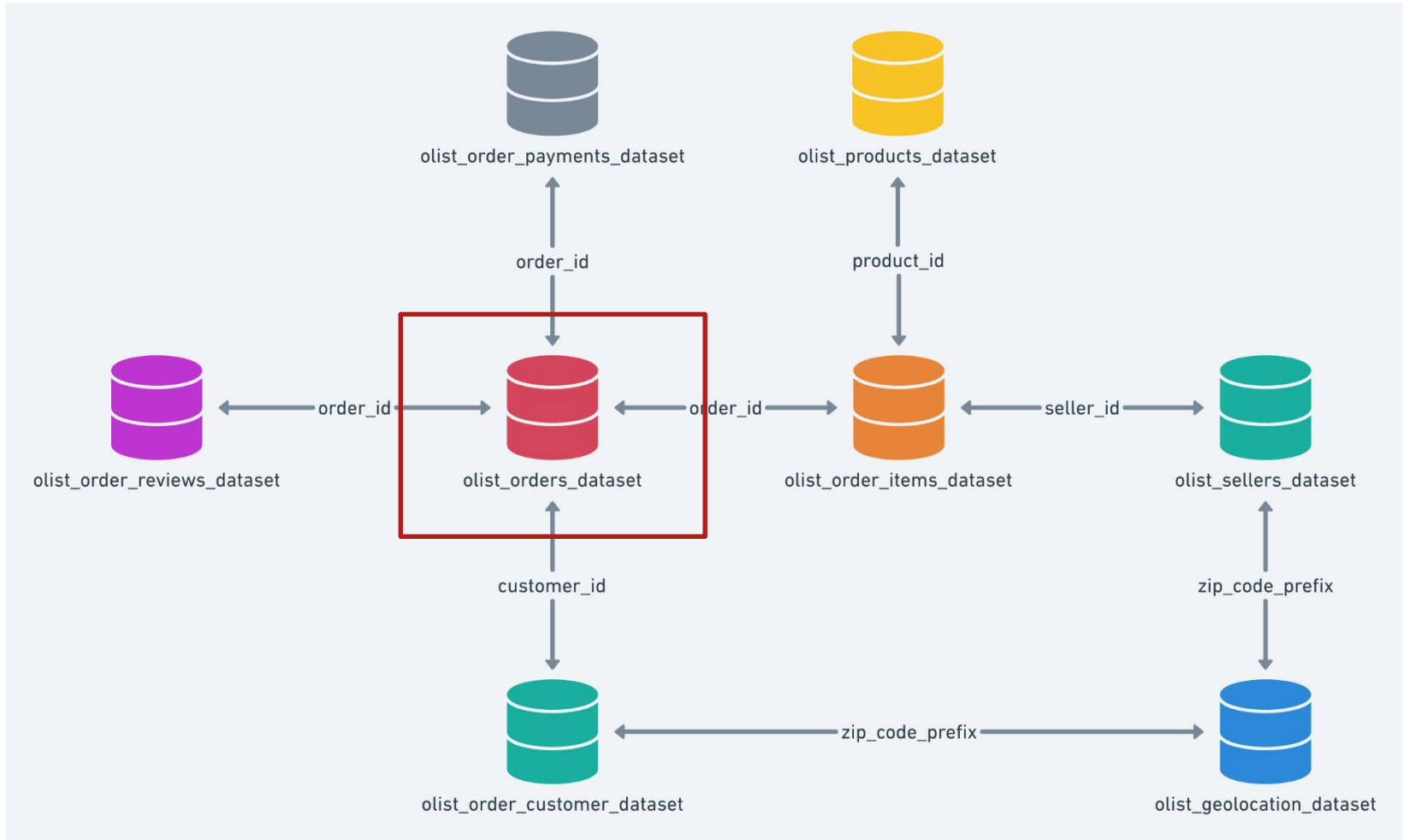
“After a customer orders a product from Olist (on behalf of a small retailer) on a marketplace, the small retailer gets notified from Olist to fulfill that order. After the customer received the product, or the estimated delivery date was due, Olist invites the customer via email to participate in a satisfaction survey, where the customer should provide a rating to express their customer satisfaction and leave some additional comments. “



Olist Data: Schematic Diagram



Olist Data: Orders Dataset



First Glance at Orders Dataset

```
> colnames(orders_data)
```

```
[1] "order_id" "customer_id" "order_status" "order_purchase_timestamp" "order_approved_at"
[6] "order_delivered_carrier_date" "order_delivered_customer_date" "order_estimated_delivery_date"
```

```
> head(orders_data, 20)
```

```
# A tibble: 20 x 8
```

	order_id <chr>	customer_id <chr>	order_status <chr>	order_purchase_time~ <dtm>	order_approved_at <dtm>	order_delivered_carri~ <dtm>	order_delivered_custo~ <dtm>	order_estimated_deliv~ <dtm>
1	e481f51cbdc54678~	9ef432eb6251297304~	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-04 19:55:00	2017-10-10 21:25:13	2017-10-18 00:00:00
2	53cdb2fc8bc7dce0~	b0830fb4747a6c6d20~	delivered	2018-07-24 20:41:37	2018-07-26 03:24:27	2018-07-26 14:31:00	2018-08-07 15:27:45	2018-08-13 00:00:00
3	47770eb9100c2d0c~	41ce2a54c0b03bf344~	delivered	2018-08-08 08:38:49	2018-08-08 08:55:23	2018-08-08 13:50:00	2018-08-17 18:06:29	2018-09-04 00:00:00
4	949d5b44dbf5de91~	f88197465ea7920adc~	delivered	2017-11-18 19:28:06	2017-11-18 19:45:59	2017-11-22 13:39:59	2017-12-02 00:28:42	2017-12-15 00:00:00
5	ad21c59c0840e6cb~	8ab97904e6daea8866~	delivered	2018-02-13 21:18:39	2018-02-13 22:20:29	2018-02-14 19:46:34	2018-02-16 18:17:02	2018-02-26 00:00:00
6	a4591c265e18cb1d~	503740e9ca751ccdda~	delivered	2017-07-09 21:57:05	2017-07-09 22:10:13	2017-07-11 14:58:04	2017-07-26 10:57:55	2017-08-01 00:00:00
7	136cce7faa42fdb2~	ed0271e0b7da060a39~	invoiced	2017-04-11 12:22:08	2017-04-13 13:25:17	NA	NA	2017-05-09 00:00:00
8	6514b8ad8028c9f2~	9bdf08b4b3b52b5526~	delivered	2017-05-16 13:10:30	2017-05-16 13:22:11	2017-05-22 10:07:46	2017-05-26 12:55:51	2017-06-07 00:00:00
9	76c6e866289321a7~	f54a9f0e6b351c4314~	delivered	2017-01-23 18:29:09	2017-01-25 02:50:47	2017-01-26 14:16:31	2017-02-02 14:08:10	2017-03-06 00:00:00
10	e69bfb5eb88e0ed6~	31ad1d1b63eb996246~	delivered	2017-07-29 11:55:02	2017-07-29 12:05:32	2017-08-10 19:45:24	2017-08-16 17:14:30	2017-08-23 00:00:00
11	e6ce16cb79ec1d90~	494dded5b201313c64~	delivered	2017-05-16 19:41:10	2017-05-16 19:50:18	2017-05-18 11:40:40	2017-05-29 11:18:31	2017-06-07 00:00:00
12	34513ce0c4fab462~	7711cf624183d843aa~	delivered	2017-07-13 19:58:11	2017-07-13 20:10:08	2017-07-14 18:43:29	2017-07-19 14:04:48	2017-08-08 00:00:00
13	82566a660a982b15~	d3e3b74c766bc6214e~	delivered	2018-06-07 10:06:19	2018-06-09 03:13:12	2018-06-11 13:29:00	2018-06-19 12:05:52	2018-07-18 00:00:00
14	5ff96c15d0b717ac~	19402a48fe860416ad~	delivered	2018-07-25 17:44:10	2018-07-25 17:55:14	2018-07-26 13:16:00	2018-07-30 15:52:25	2018-08-08 00:00:00
15	432aaf21d85167c2~	3df704f53d3f1d4818~	delivered	2018-03-01 14:14:28	2018-03-01 15:10:47	2018-03-02 21:09:20	2018-03-12 23:36:26	2018-03-21 00:00:00
16	dc36b511fcac050~	3b6828a50ffe546942~	delivered	2018-06-07 19:03:12	2018-06-12 23:31:02	2018-06-11 14:54:00	2018-06-21 15:34:32	2018-07-04 00:00:00
17	403b97836b0c04a6~	738b086814c6fcc74b~	delivered	2018-01-02 19:00:43	2018-01-02 19:09:04	2018-01-03 18:19:09	2018-01-20 01:38:59	2018-02-06 00:00:00
18	116f0b09343b4955~	3187789bec99098762~	delivered	2017-12-26 23:41:31	2017-12-26 23:50:22	2017-12-28 18:33:05	2018-01-08 22:36:36	2018-01-29 00:00:00
19	85ce859fd6dc634d~	059f7fc5719c7da6cb~	delivered	2017-11-21 00:03:41	2017-11-21 00:14:22	2017-11-23 21:32:26	2017-11-27 18:28:00	2017-12-11 00:00:00
20	83018ec114eee864~	7f8c8b9c2ae27bf330~	delivered	2017-10-26 15:54:26	2017-10-26 16:08:14	2017-10-26 21:46:53	2017-11-08 22:22:00	2017-11-23 00:00:00

First Glance at Orders Dataset

```
> colnames(orders_data)
```

```
[1] "order_id" "customer_id" "order_status" "order_purchase_timestamp" "order_approved_at"
[6] "order_delivered_carrier_date" "order_delivered_customer_date" "order_estimated_delivery_date"
```

```
> head(orders_data, 20)
```

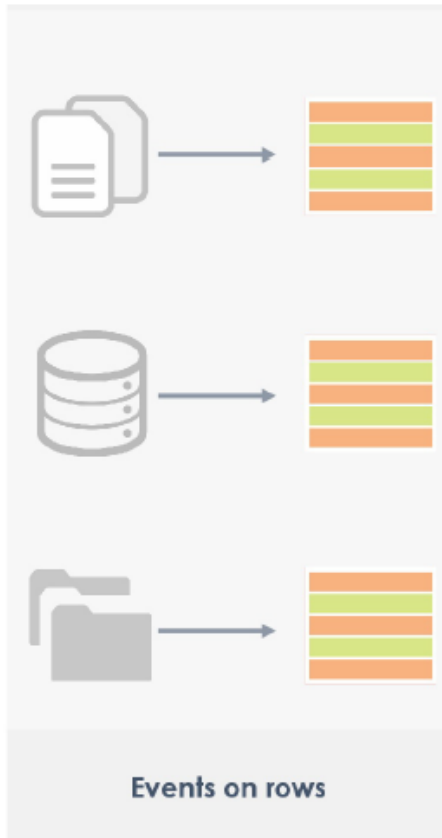
```
# A tibble: 20 x 8
  order_id customer_id order_status order_purchase_time~ order_approved_at order_delivered_carri~ order_delivered_custo~ order_estimated_deliv~
  <chr> <chr> <chr> <dtm> <dtm> <dtm> <dtm> <dtm>
1 e481f51cbdc54678~ 9ef432eb6251297304~ delivered 2017-10-02 10:56:33 2017-10-02 11:07:15 2017-10-04 19:55:00 2017-10-10 21:25:13 2017-10-18 00:00:00
2 53cdb2fc8bc7dce0~ b0830fb4747a6c6d20~ delivered 2018-07-24 20:41:37 2018-07-26 03:24:27 2018-07-26 14:31:00 2018-08-07 15:27:45 2018-08-13 00:00:00
3 47770eb9100c2d0c~ 41ce2a54c0b03bf344~ delivered 2018-08-08 08:38:49 2018-08-08 08:55:23 2018-08-08 13:50:00 2018-08-17 18:06:29 2018-09-04 00:00:00
4 949d5b44dbf5de91~ f88197465ea7920adc~ delivered 2017-11-18 19:28:06 2017-11-18 19:45:59 2017-11-22 13:39:59 2017-12-02 00:28:42 2017-12-15 00:00:00
5 ad21c59c0840e6cb~ 8ab97904e6daea8866~ delivered 2018-02-13 21:18:39 2018-02-13 22:20:29 2018-02-14 19:46:34 2018-02-16 18:17:02 2018-02-26 00:00:00
6 a4591c265e18cb1d~ 503740e9ca751ccdda~ delivered 2017-07-09 21:57:05 2017-07-09 22:10:13 2017-07-11 14:58:04 2017-07-26 10:57:55 2017-08-01 00:00:00
7 136cce7faa42fdb2~ ed0271e0b7da060a39~ invoiced 2017-04-11 12:22:08 2017-04-13 13:25:17 NA 2017-05-09 00:00:00
8 6514b8ad8028c9f2~ 9bdf08b4b3b52b5526~ delivered 2017-05-16 13:10:30 2017-05-16 13:22:11 2017-05-22 10:07:46 2017-05-26 12:55:51 2017-06-07 00:00:00
9 76c6e866289321a7~ f54a9f0e6b351c4314~ delivered 2017-01-23 18:29:09 2017-01-25 02:50:47 2017-01-26 14:16:31 2017-02-02 14:08:10 2017-03-06 00:00:00
10 e69bfb5eb88e0ed6~ 31ad1d1b63eb996246~ delivered 2017-07-29 11:55:02 2017-07-29 12:05:32 2017-08-10 19:45:24 2017-08-16 17:14:30 2017-08-23 00:00:00
11 e6ce16cb79ec1d90~ 494dded5b201313c64~ delivered 2017-05-16 19:41:10 2017-05-16 19:50:18 2017-05-18 11:40:40 2017-05-29 11:18:31 2017-06-07 00:00:00
12 34513ce0c4fab462~ 7711cf624183d843aa~ delivered 2017-07-13 19:58:11 2017-07-13 20:10:08 2017-07-14 18:43:29 2017-07-19 14:04:48 2017-08-08 00:00:00
13 82566a660a982b15~ d3e3b74c766bc6214e~ delivered 2018-06-07 10:06:19 2018-06-09 03:13:12 2018-06-11 13:29:00 2018-06-19 12:05:52 2018-07-18 00:00:00
14 5ff96c15d0b717ac~ 19402a48fe860416ad~ delivered 2018-07-25 17:44:10 2018-07-25 17:55:14 2018-07-26 13:16:00 2018-07-30 15:52:25 2018-08-08 00:00:00
15 432aaf21d85167c2~ 3df704f53d3f1d4818~ delivered 2018-03-01 14:14:28 2018-03-01 15:10:47 2018-03-02 21:09:20 2018-03-12 23:36:26 2018-03-21 00:00:00
16 dcb36b511fcac050~ 3b6828a50ffe546942~ delivered 2018-06-07 19:03:12 2018-06-12 23:31:02 2018-06-11 14:54:00 2018-06-21 15:34:32 2018-07-04 00:00:00
17 403b97836b0c04a6~ 738b086814c6fcc74b~ delivered 2018-01-02 19:00:43 2018-01-02 19:09:04 2018-01-03 18:19:09 2018-01-20 01:38:59 2018-02-06 00:00:00
18 116f0b09343b4955~ 3187789bec99098762~ delivered 2017-12-26 23:41:31 2017-12-26 23:50:22 2017-12-28 18:33:05 2018-01-08 22:36:36 2018-01-29 00:00:00
19 85ce859fd6dc634d~ 059f7fc5719c7da6cb~ delivered 2017-11-21 00:03:41 2017-11-21 00:14:22 2017-11-23 21:32:26 2017-11-27 18:28:00 2017-12-11 00:00:00
20 83018ec114eee864~ 7f8c8b9c2ae27bf330~ delivered 2017-10-26 15:54:26 2017-10-26 16:08:14 2017-10-26 21:46:53 2017-11-08 22:22:00 2017-11-23 00:00:00
```

Case_id

Timestamps

Activities !?

Create Rows of Events



Reshape Data

Reshape Data - change the layout of values in a table

Use **gather()** and **spread()** to reorganize the values of a table into a new layout.

gather()(data, key, value, ..., na.rm = FALSE, convert = FALSE, factor_key = FALSE)

gather() moves column names into a **key** column, gathering the column values into a single **value** column.

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

key value

`gather(table4a, `1999`, `2000`,
key = "year", value = "cases")`

spread()(data, key, value, fill = NA, convert = FALSE, drop = TRUE, sep = NULL)

spread() moves the unique values of a **key** column into the column names, spreading the values of a **value** column across the new columns.

table2

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

key value

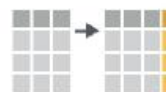
`spread(table2, type, count)`

Further Data Processing Commands


MAKE NEW VARIABLES

These apply **vectorized functions** to columns. Vectorized funs take vectors as input and return vectors of the same length as output (see back).

vectorized function


 **mutate(.data, ...)**
Compute new column(s).
mutate(mtcars, gpm = 1/mpg)


COMBINE CASES


 **bind_rows(..., .id = NULL)**
Returns tables one on top of the other as a single table. Set .id to a column name to add a column of the original table names (as pictured)


COMBINE VARIABLES

Use a "**Mutating Join**" to join one table to columns from another, matching values with the rows that they correspond to. Each join retains a different combination of values from the tables.

 **left_join(x, y, by = NULL, copy=FALSE, suffix=c(".x", ".y"),...)**
Join matching values from y to x.

 **right_join(x, y, by = NULL, copy = FALSE, suffix=c(".x", ".y"),...)**
Join matching values from x to y.

 **inner_join(x, y, by = NULL, copy = FALSE, suffix=c(".x", ".y"),...)**
Join data. Retain only rows with matches.

 **full_join(x, y, by = NULL, copy=FALSE, suffix=c(".x", ".y"),...)**
Join data. Retain all values, all rows.

Reviews dataset

```
-- Variable type:character -----
      variable missing complete      n min max empty n_unique
order_id          0   100000 100000   32 32     0    99441
review_comment_message 58255   41745 100000    1 208     0    36506
review_comment_title   88287   11713 100000    1  26     0     4248
review_id           0   100000 100000   32 32     0    99173

-- Variable type:numeric -----
      variable missing complete      n mean  sd p0 p25 p50 p75 p100 hist
review_score          0   100000 100000 4.07 1.36  1   4   5   5   5
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|


-- Variable type:POSIXct -----
      variable missing complete      n      min      max      median n_unique
review_answer_timestamp  0   100000 100000 2016-10-07 2018-10-29 2018-02-04    99010
review_creation_date     0   100000 100000 2016-10-02 2018-08-31 2018-02-02      637
```

- Case ID: Order Id
- Activities:
 - Sent out survey
 - Receive Feedback

Payment dataset

```
-- Variable type:character -----
  variable missing complete      n min max empty n_unique
  order_id      0   103886 103886   32 32    0   99440
  payment_type   0   103886 103886    6 11    0     5

-- Variable type:numeric -----
  variable missing complete      n  mean    sd p0  p25 p50  p75  p100  hist
  payment_installments 0   103886 103886  2.85  2.69  0  1    1    4    24
  payment_sequential   0   103886 103886  1.09  0.71  1  1    1    1    29
  payment_value        0   103886 103886 154.1 217.49 0 56.79 100 171.84 13664.08
```



```
> head(payments_data)
# A tibble: 6 x 5
  order_id                payment_sequential payment_type payment_installments payment_value
  <chr>                  <dbl> <chr>          <dbl>          <dbl>
1 b81ef226f3fe1789b1e8b2acac839d17          1 credit_card      8           99.3
2 a9810da82917af2d9aefd1278f1dcfa0          1 credit_card      1           24.4
3 25e8ea4e93396b6fa0d3dd708e76c1bd          1 credit_card      1           65.7
4 ba78997921bbcdc1373bb41e913ab953          1 credit_card      8          108.
5 42fdf880ba16b47b59251dd489d4441a          1 credit_card      2          128.
6 298fcdf1f73eb413e4d26d01b25bc1cd          1 credit_card      2           96.1
```

Problem:

- No payment status
- No timestamp

-> No payment activity can be derived, BUT maybe there are interesting attributes to the orders in the data set?

Other data sets

- Use other Olist Data sets to create new attributes to the orders
- No timestamps included in the other data sets

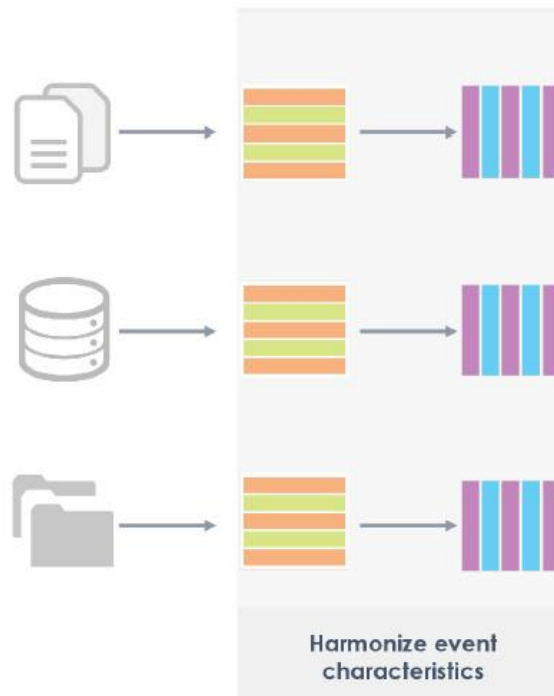
Mutate new variables

Make New Variables



Harmonize Case Attributes

- When row-binding different data sets, missing values can occur in some of the attributes
- Harmonize case attributes over all events related to the case



BupaR Package

- Integrated suite of **R**-packages for the handling and analysis of business process data
- <https://www.bupar.net/>
- **Useful Links:**
 - <https://www.datacamp.com/courses/business-process-analytics-in-r>
 - <https://www.r-bloggers.com/process-mining-part-1-3-introduction-to-bupar-package/>
- Event Log version of common dplyr functions

Source: Janssenswillen, G., Depaire, B., Swennen, M., Jans, M., & Vanhoof, K. (2019). bupaR: Enabling reproducible business process analysis. Knowledge-Based Systems, 163, 927-930.

Event Log Mapping

- Bupar Event Log Structure**

- a timestamp
- a case identifier
- an activity label
- a activity instance identifier
- a transactional life cycle stage
- a resource identifier

- Example**

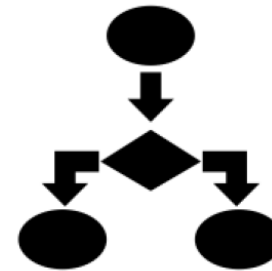
patient	activity	timestamp	status	activity_instance	resource
John Doe	check-in	2017-05-10 08:33:26	complete	1	Samantha
John Doe	surgery	2017-05-10 08:38:21	schedule	2	Danny
John Doe	surgery	2017-05-10 08:53:16	start	2	Richard
John Doe	surgery	2017-05-10 09:25:19	complete	2	Richard
John Doe	treatment	2017-05-10 10:01:25	start	3	Danny
John Doe	treatment	2017-05-10 10:35:18	complete	3	Danny
John Doe	surgery	2017-05-10 10:41:35	start	4	William
John Doe	surgery	2017-05-10 11:05:56	complete	4	William
John Doe	check-out	2017-05-11 14:52:36	complete	5	Samantha

Event Data Analysis

Organizational



Control-flow



Performance



And also

- Multivariate analysis
- Include additional data attributes