Process Oriented Data Science

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Campus d'Excel·lència Internacional

Josep Carmona
Computer Science Department

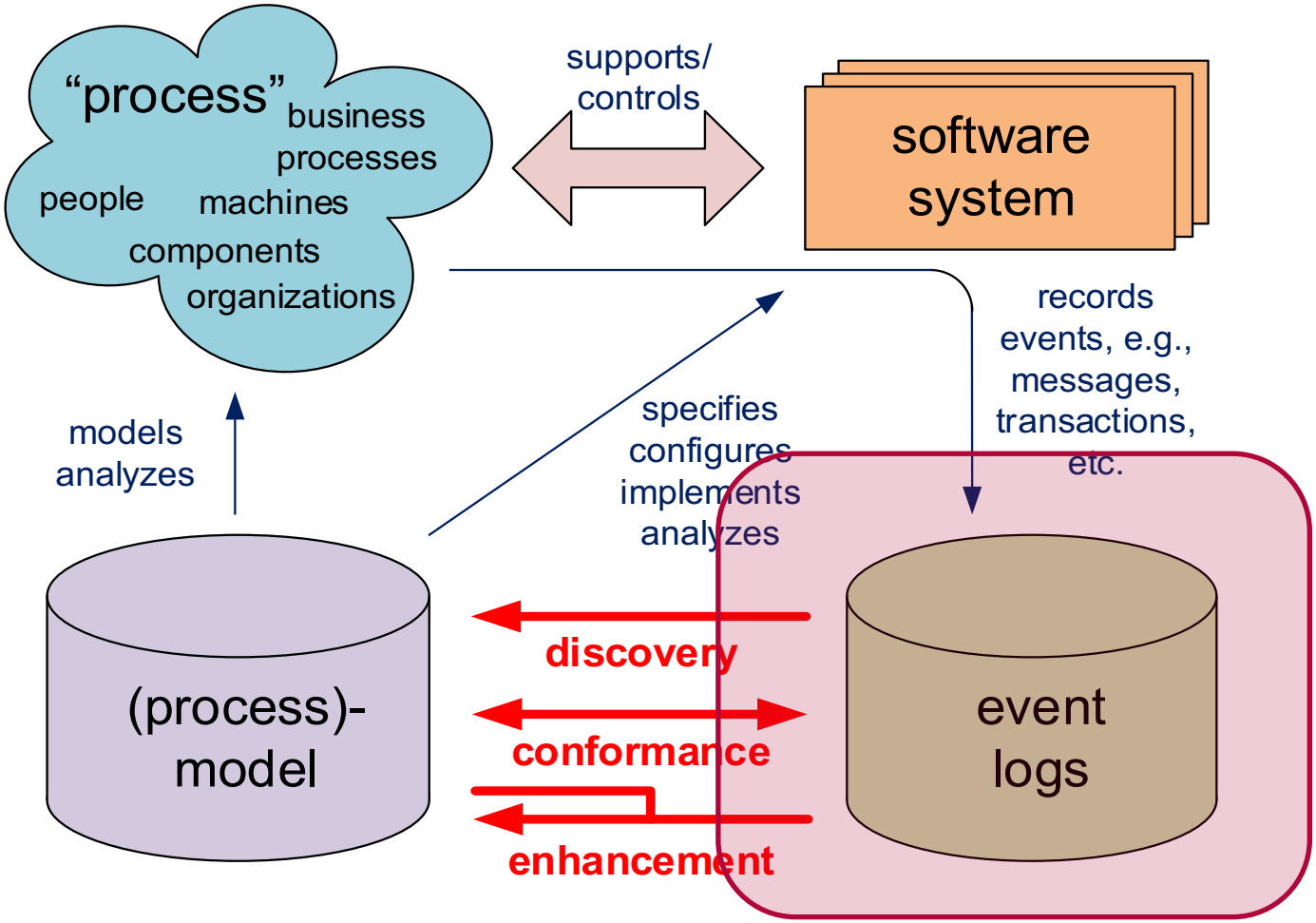
Outline

- **M1: Process Mining Overview, Positioning & Preliminaries (Event data & Process Models)**
- M2: Process Discovery
- M3: Conformance Checking
- M4: Process Enhancement

Disclaimer

- Most of the material of this course is taken from my colleagues:
  - RWTH Aachen (Prof. Wil van der Aalst)
  - **Humboldt University zu Berlin (Prof. Matthias Weidlich)**
  - Technishe Universiteit Eindhoven (Prof. Boudewijn van Dongen)
  - University of Tartu (Prof. Marlon Dumas)
  - University of Melbourne (Prof. Marcello La Rosa)
  - Technical University of Denmark (Prof. Andrea Burattin)
- Hence, this material is only provided for your learning, please do not share nor publish

The Context


"process"
business
processes
people
machines
components
organizations
supports/
controls
software
system
records
events, e.g.,
messages,
transactions,
etc.
models
analyzes
specifies
configures
implements
analyzes
(process)-
model
discovery
conformance
enhancement
event
logs

Logs as Information Source

- Logs contain information to answer questions
  - When have process instances been executed?
  - How many instances have been executed?
  - Have there been recurring patterns in the executions of activities?
  - Is it possible to construct process models based on the log data?
  - Which sequences of activities have been executed very frequently?
  - Does a process model contain execution sequences that have never been executed?
- Logs are the basis for evidence-based answers to these questions
  - Not biased by human perception of how a process is conducted
  - Not biased by fragmentation of process knowledge
  - Yet, assuming high data quality

Log Entries

Example log entries

- *Check of invoice with number 4567 finished on 12.11.2010 at 9:19:57*
- *Function StoreCustomerData("Müller", c1987, "Bad Bentheim") executed on 12.11.2010 at 9:22:24*
- *Invoice sent for invoice number 4567 finished on 12.11.2010 at 9:23:18*
- *Inserted data (c1987, PromoMailing) into customer status database on 12.11.2010 at 9:24:10*
- *Function StoreCustomerData("Miller", c1988, "Osnabrück") executed on 12.11.2010 at 9:26:08*
- *Check of invoice with number 4568 finished on 12.11.2010 at 9:26:38*

Event Logs

- Specific model that imposes assumptions
  - (Total) order of events
  - Event can be related to activity and process instance (case)
  - Events in event log relate to instance of a single process
- But
  - Actual logs often contain events of various processes
  - Requires pre-processing
- Format of logs: (timestamp, caseID, activity, additional attributes …)
  - Example
    - *Check of invoice with number 4567 finished on 12.11.2010 at 9:19:57*
    - *Function StoreCustomerData("Müller", c1987, "Bad Bentheim") executed on 12.11.2010 at 9:22:24*
  - *Event log*
    - *(1289553597, 4567, invoice-check),(1289553744, c1987, StoreCustomerData, "Müller", "Bad Bentheim")*

Event Log Abstractions

- Further abstractions depending on use case
  - Abstract from activity names (A's and B's)
  - Abstract from specific timestamp values (preserve only ordering)
  - Abstract from additional event data
- Variations of event logs
  - Not necessarily a 1:1 relation between activities and events
  - Start and end events for execution of activities
- Issues
  - Quality issues in terms of missing events, duplicated events, incorrect ordering (due to race conditions), etc.
  - Implications discussed later

```
case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D
```

Log Assumptions
Reality Check

Relate an event to the execution of an activity?
Relate an event to a process instance?

… Record claim
Check coverage
Request proof of loss
Do field check
Take decision…

… Submit order
Check credit history
Charge credit card
Check availability
Plan shipments…

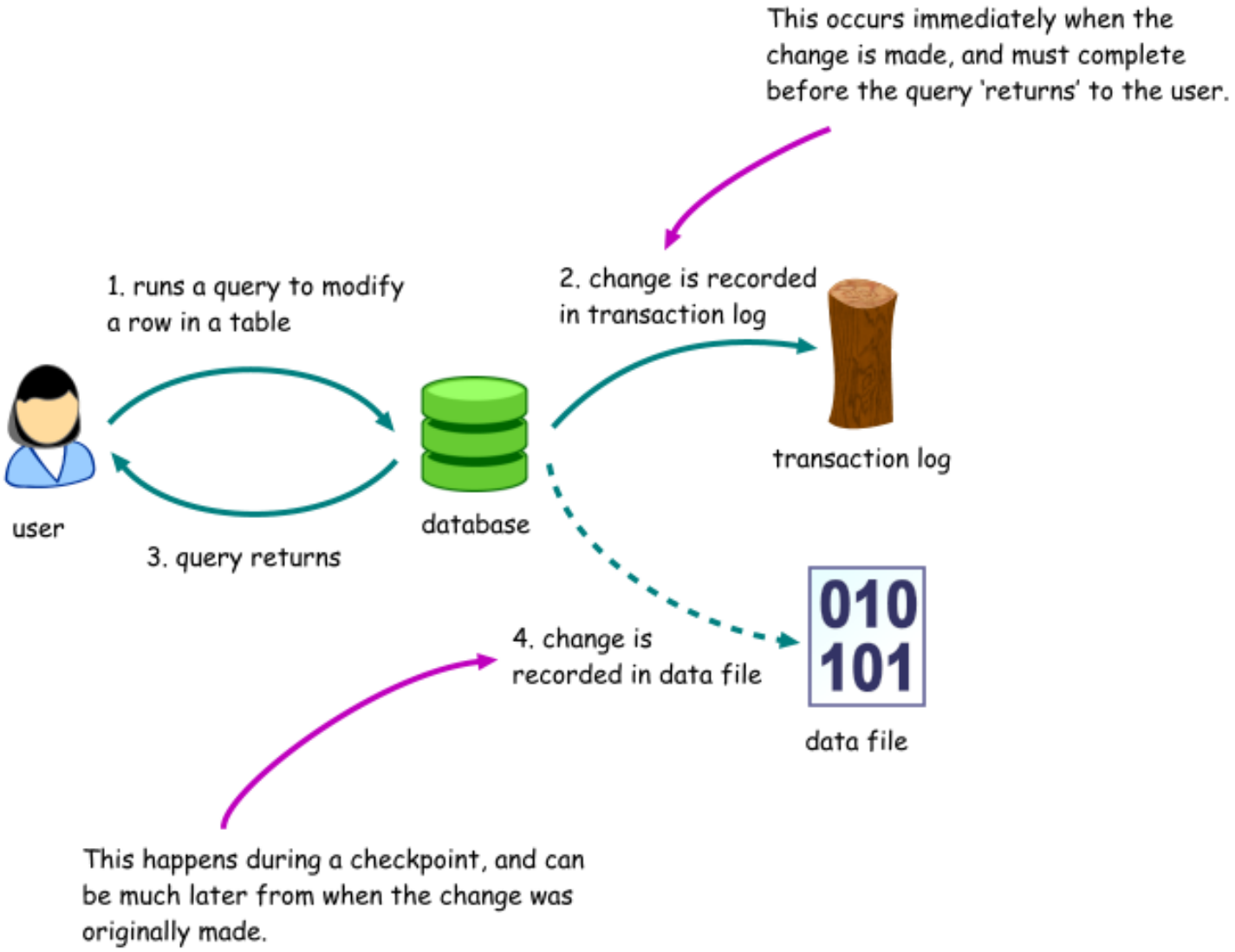
Thinking further about:
Timestamps, snapshots, scoping, granularity

Extraction of Logs From DBs

- Idea: Process-oriented Information Systems store process-related data in ordinary Databases
  - Changes in the data hint at activity execution
  - Logs about data changes enable construction of event logs
- Issue: Relation between processes and data is non-trivial
  - To the rescue: Domain models outlining the entities can serve as a starting point
  - Approach allows for flexibility in the definition of the notion of a process instance

Details:
Eduardo González López de Murillas, Wil M. P. van der Aalst, Hajo A. Reijers: Process Mining on Databases: Unearthing Historical Data from Redo Logs. BPM 2015:367-385

DB Logging


This occurs immediately when the change is made, and must complete before the query 'returns' to the user.
1. runs a query to modify a row in a table
2. change is recorded in transaction log
transaction log
user
database
3. query returns
4. change is recorded in data file
010
101
data file
This happens during a checkpoint, and can be much later from when the change was originally made.

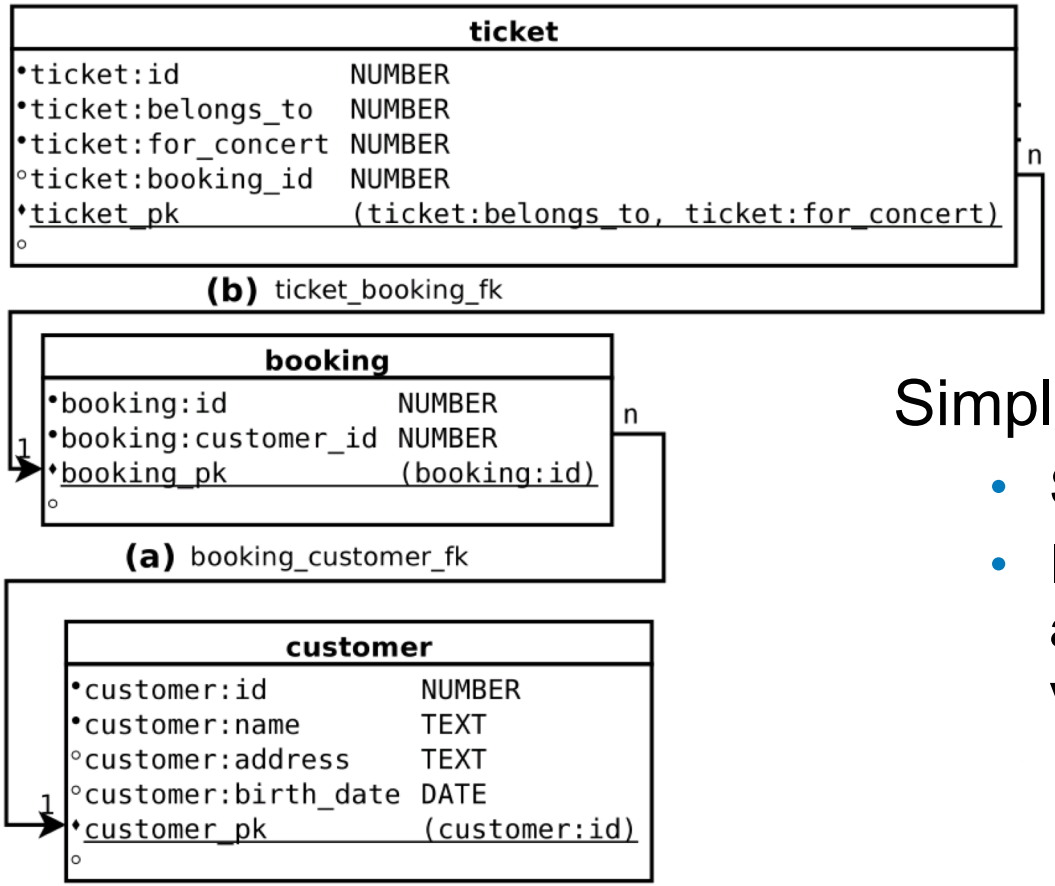
Transaction Log

- Contains information on change operations
- Use to ensure consistent state of DB
  - Operations can be redone and undone
  - Operations are persisted
- Writing to transaction log is faster than writing to actual data file
  - Transaction log is written sequentially
  - Writing to data file: extension of data file, page splits, new allocations, etc.

Extraction Idea

- Event extraction
  - Each record of transaction log is an event
  - Extract attributes affected by change operation
- Exploiting the data model
  - Extract data model from DB
  - Rely on foreign keys in model to identify possible correlations
- Case ID construction
  - Determine the scope to use for correlation

Example
Data Model



Simple scenario:

- Shop sells tickets
- Information between tickets and customers established via bookings

Example Transaction Log

#	Time + Op + Table	Redo	Undo
1	2014-11-27 15:57:08.0 + INSERT + CUSTOMER	`insert into "SAMPLEDB". "CUSTOMER" ("ID", "NAME", "ADDRESS", "BIRTH_DATE") values ('17299', 'Name1', 'Address1', TO_DATE( '01-AUG-06', 'DD-MON-RR'));`	`delete from "SAMPLEDB". "CUSTOMER" where "ID" = '17299' and "NAME" = 'Name1' and "ADDRESS" = 'Address1' and "BIRTH_DATE" = TO_DATE( '01-AUG-06', 'DD-MON-RR') and ROWID = '1';`

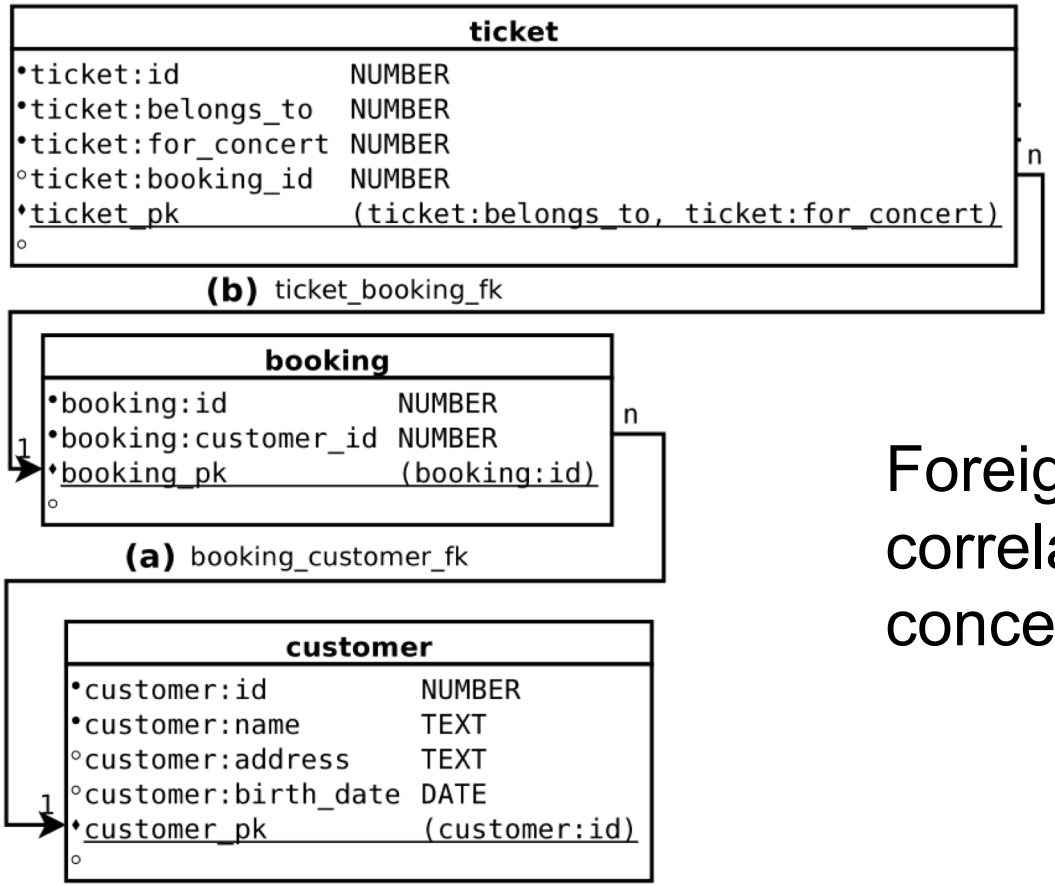
Example
Extracted Events

#	Attribute name	Value after event	Value before event
1	Customer:id	17299	-
	Customer:name	Name1	-
	Customer:address	Address1	-
	Customer:birth_date	01-AUG-06	-
	RowID	=	1
2	Customer:id	=	{17299}
	Customer:name	Name2	Name1
	Customer:address	=	{Address1}
	Customer:birth_date	=	{01-AUG-06}
	RowID	=	1
3	Booking:id	36846	-
	Booking:customer_id	17299	-
	RowID	=	2
4	Ticket:booking_id	36846	NULL
	Ticket:id	=	(317132)
	Ticket:belongs_to	=	(172935)
	Ticket:for_concert	=	(1277)
	RowID	=	3
5	Booking:id	36876	-
	Booking:customer_id	17299	-
	RowID	=	4
6	Ticket:booking_id	36876	NULL
	Ticket:id	=	(317435)
	Ticket:belongs_to	=	(173238)
	Ticket:for_concert	=	(1277)
	RowID	=	5

"=" reads "did not change"

"{}" reads "not part of log, but extracted from DB"

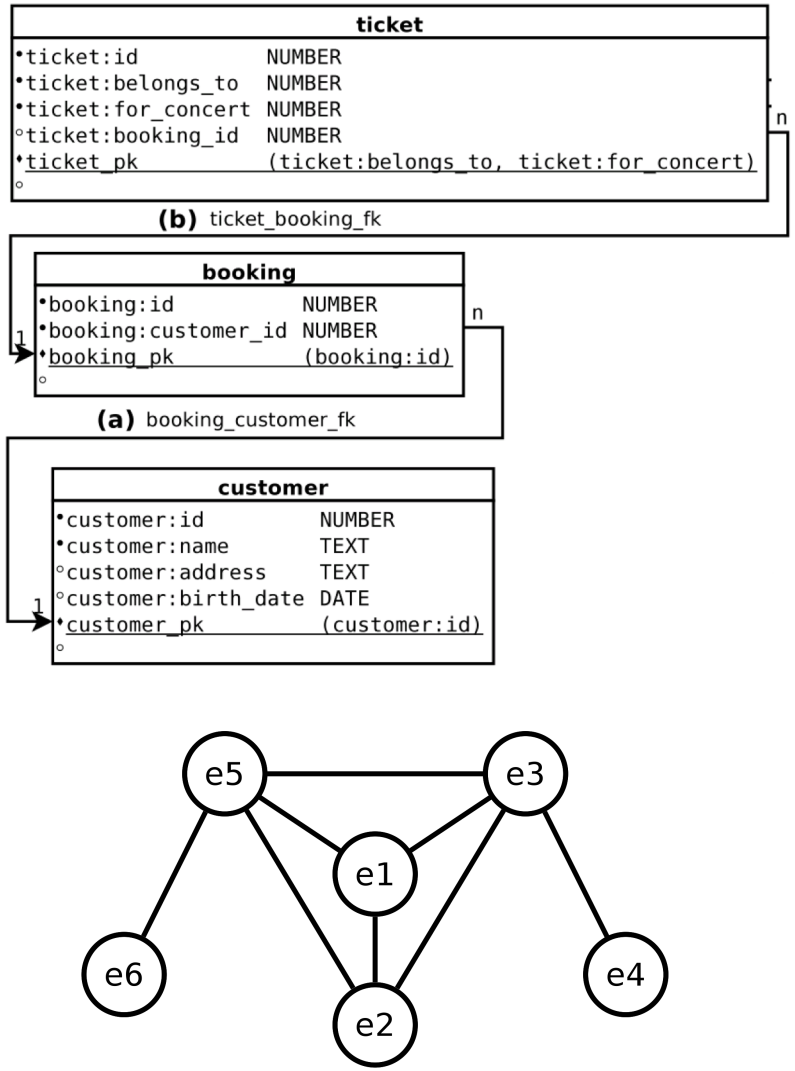
Data Model Revisited



Foreign key relations hint at correlations between concepts!

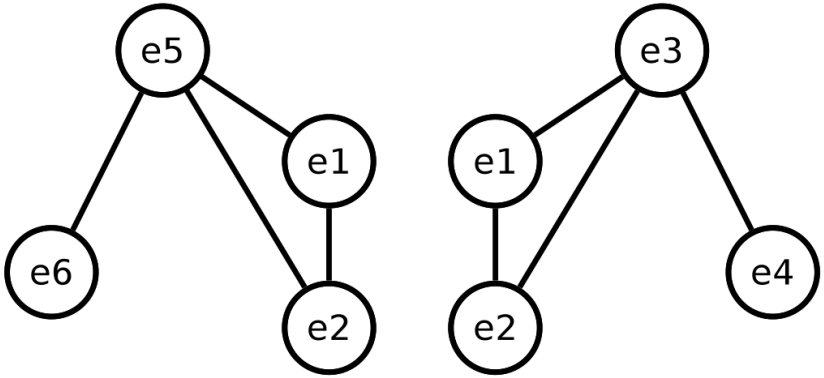
Possible Correlations

#	Attribute name	Value after event	Value before event
1	Customer:id	17299	-
	Customer:name	Name1	-
	Customer:address	Address1	-
	Customer:birth_date	01-AUG-06	-
	RowID	=	1
2	Customer:id	=	{17299}
	Customer:name	Name2	Name1
	Customer:address	=	{Address1}
	Customer:birth_date	=	{01-AUG-06}
	RowID	=	1
3	Booking:id	36846	-
	Booking:customer_id	17299	-
	RowID	=	2
4	Ticket:booking_id	36846	NULL
	Ticket:id	=	(317132)
	Ticket:belongs_to	=	(172935)
	Ticket:for_concert	=	(1277)
	RowID	=	3
5	Booking:id	36876	-
	Booking:customer_id	17299	-
	RowID	=	4
6	Ticket:booking_id	36876	NULL
	Ticket:id	=	(317435)
	Ticket:belongs_to	=	(173238)
	Ticket:for_concert	=	(1277)
	RowID	=	5



Notions of a Case

- Relations between foreign keys define "space" for case extraction
  - All possible correlations
  - Exact scoping depends on context
- Here: case may be given by booking
  - Split up dependency graph of events
  - Extraction of two traces



Limitations

- Strong link between CRUD operations and activities is assumed
  - Record claim, Check coverage
  - Submit order, Check credit history
- Handling of operations related to multiple cases
- Linking a case with primary keys in relations may be non-trivial

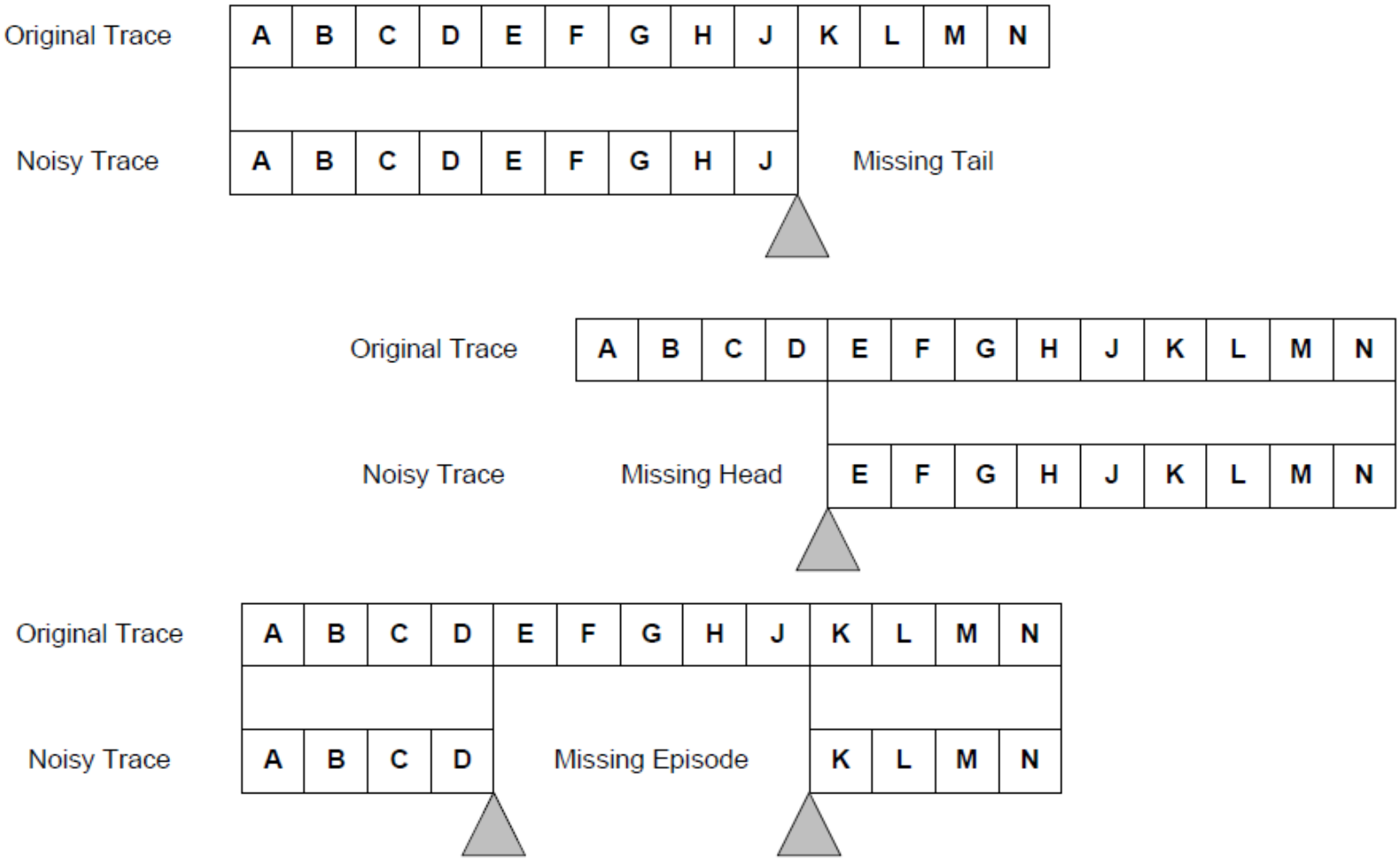
Quality of Event Logs

- "Clean event logs"
  - Event relates to one activity and one process instance
  - All traces are valid execution sequences of the process
- Not realistic in practice, there is "noise"
  - Erroneous logging mechanisms
  - Not all log entries are written, some are lost or inserted in a wrong order

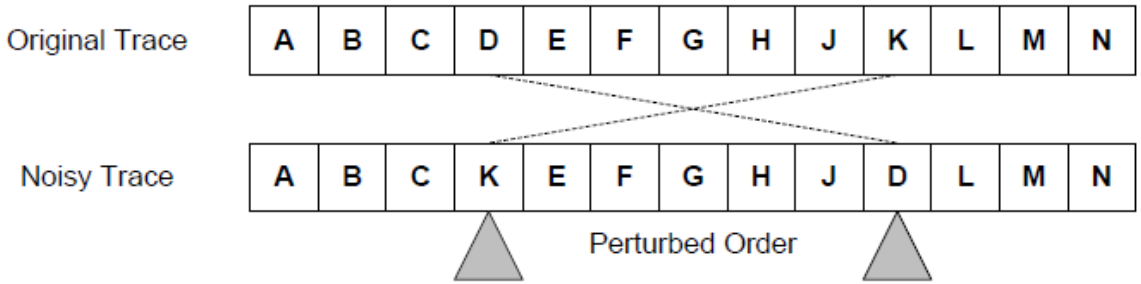
- Think about the example scenarios again….

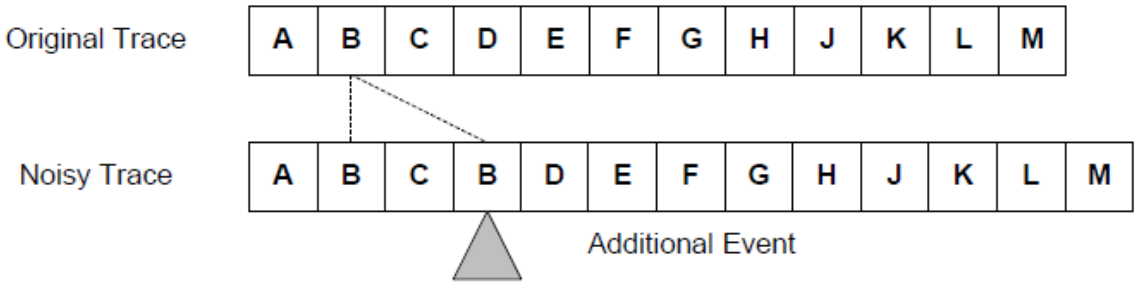
Types of Noise

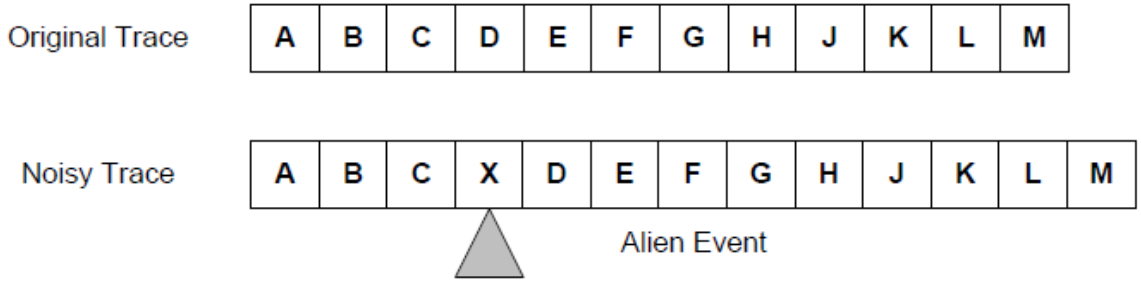
Logging was temporarily not available



Types of Noise (cont'd)





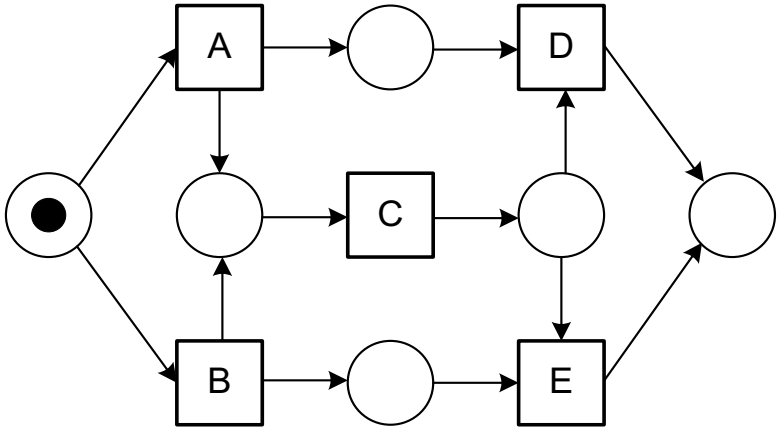


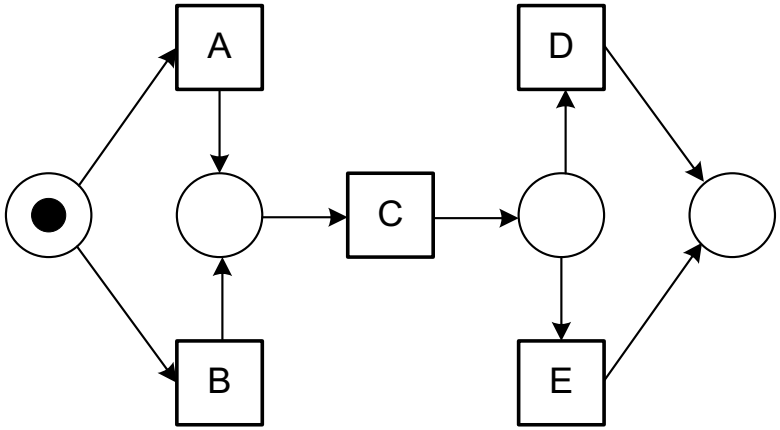
Consequences of Noise

- Massive impact on discovery, conformance, and enhancement techniques – we will get back to this
- Already an issue in the construction of event logs
- Major issue: *what* is noise is close to impossible to characterise without domain knowledge

Noise Example

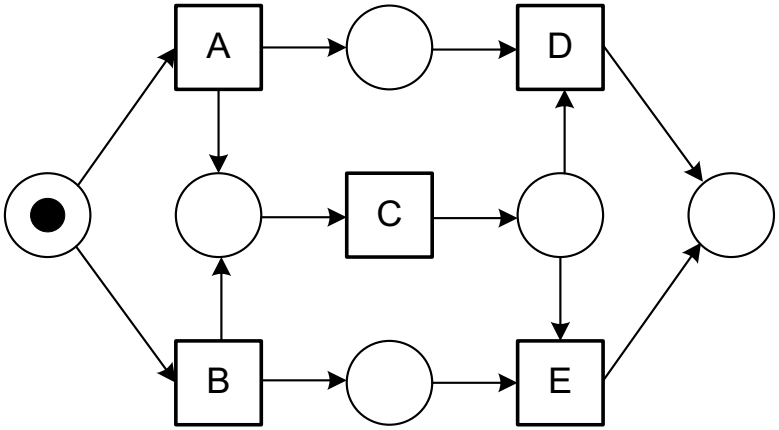
ACD	99
ACE	0
BCE	85
BCD	0

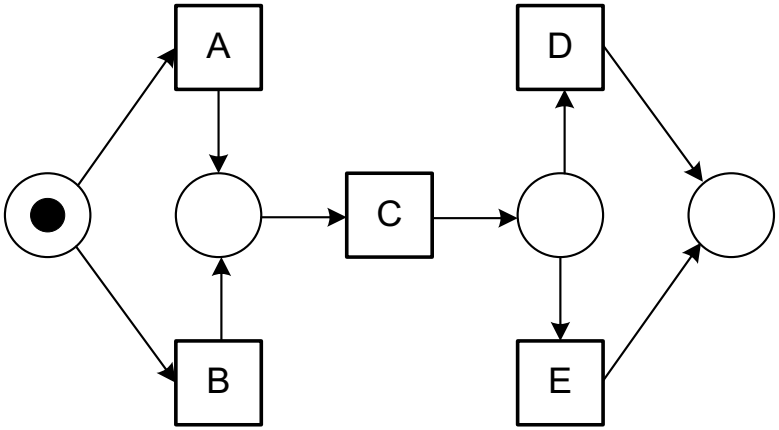




Noise Example (cont'd)

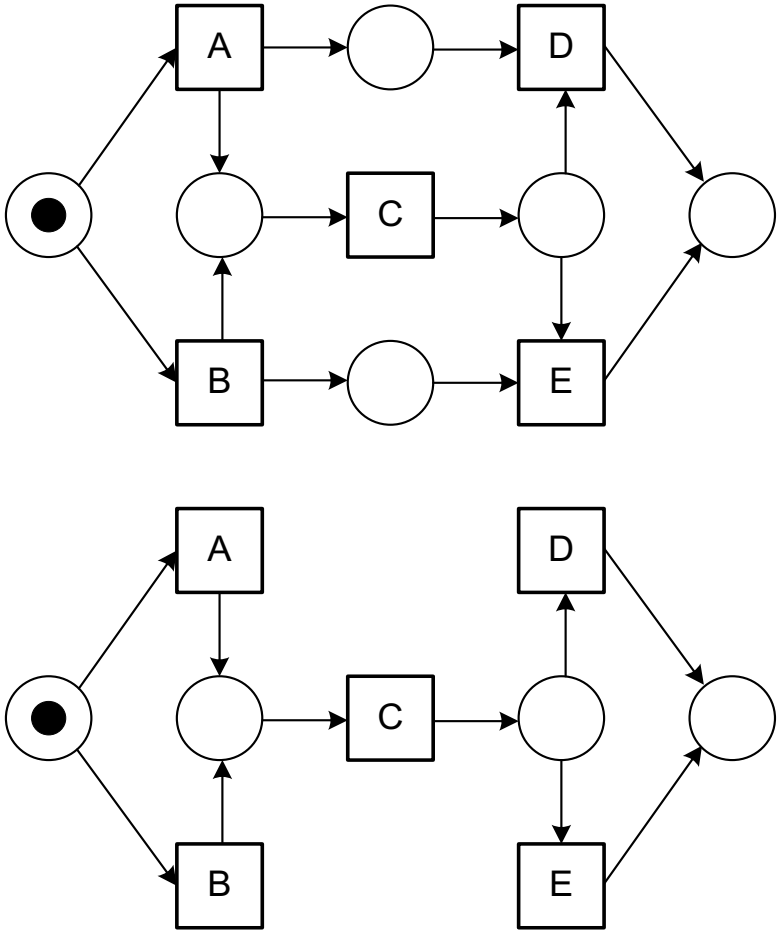
ACD 99
ACE 88
BCE 85
BCD 78





Noise Example (cont'd)

ACD	99
ACE	2
BCE	85
BCD	3



Log-based Noise Handling

- Rely on frequency analysis to identify noise in event log
- Assumption: noise is rare
  - Very infrequent traces can be seen as noise
  - Traces that contain very infrequent transitions can be seen as noise
  - Operationalization based on standard data mining techniques – association rules mining
- Again, this assumption may be wrong! Advanced techniques:
  - Mine association rules and filter out events not covered by supported rules
  - Same as before but for traces, not events.

In practice ...

- Event logs take various different forms and instantiations
- Differences in semantics, e.g., related to
  - Timestamps
  - Total vs. Partial order
- Difference in quality, e.g., related to
  - Completeness
  - Noise-level
  - Data richness
- Technical alignment by means of standards

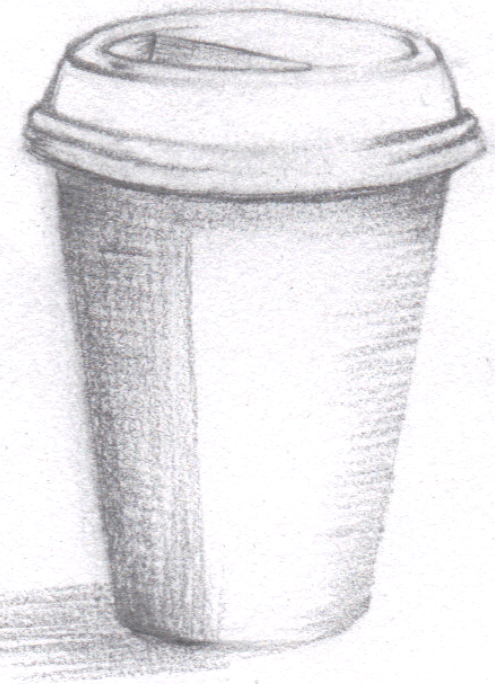
Take Away

Event logs as the basis of process mining techniques

Essential assumptions on event ordering and identifiers for activities and cases

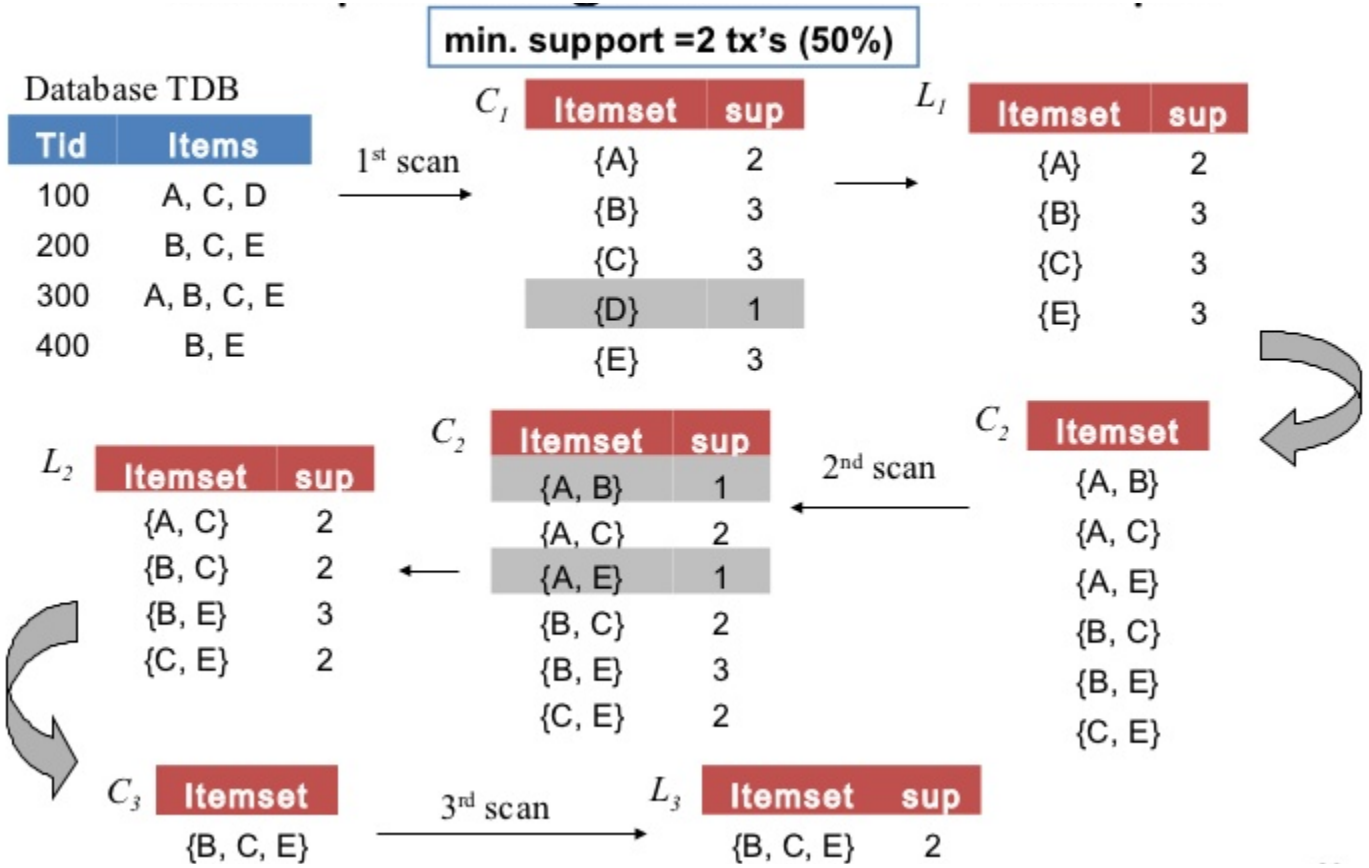
Other scenarios (e.g., unlabeled sequences) are possible, but left out of this course

Noise has an important role, cannot be disregarded



Association Rules (Events)

Apriori algorithm for frequent itemset mining



Subsequent construction of association rules by investigating support of subsets