ISTM 6212 - Week 9
Hierarchies, Snapshots

Daniel Chudnov, dchud@gwu.edu

2016-11-08

Agenda

- Schedule check
- Quick review
- Handling Hierarchies
- Snapshots
- Accumulating Snapshots
- Project 02 work session

Schedule check

Quick review

Facts and dimensions

- **Facts** are instances of business processes worthy of measurement
- **Dimensions** are the contexts in which those processes occurred and through which their measurement may be framed

Facts are sparse; dimensions wide

- Facts represent individual events; no records for "all possible events", only what actually happened
- Dimensions represent possible contexts; records for many possible combinations of filter/aggregation attributions

Consistency in dimensions

- Same structure: same attributes, names, types
- Same content: same values, casing, abbreviations
- Queries can account for differences, but early planning and proper ETL can make drilling across easier
- Attributes must match even if tables don't

Affinity in dimensions

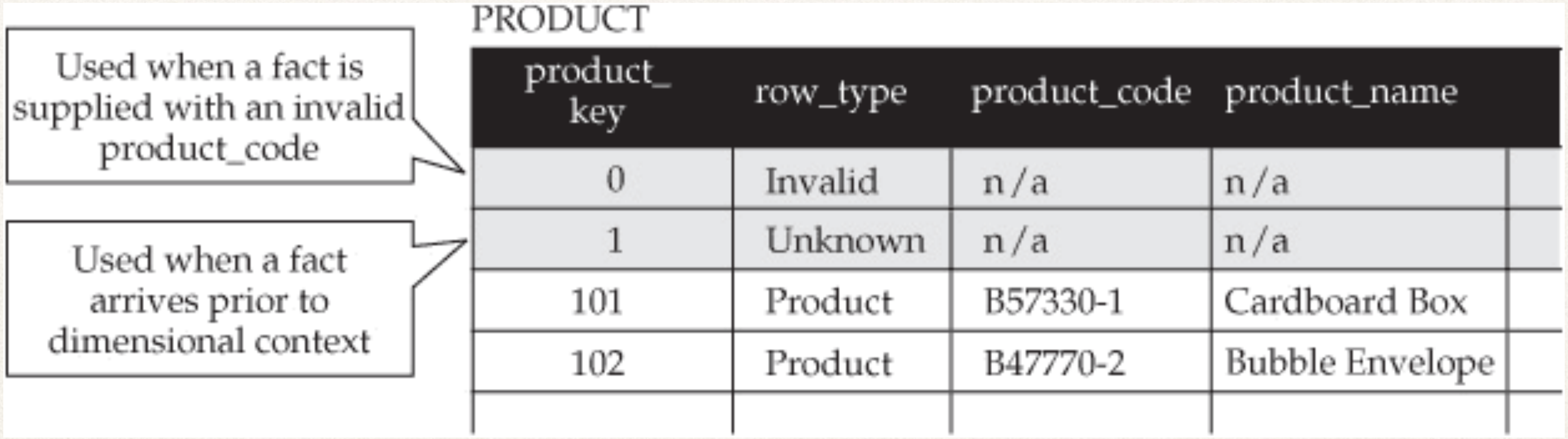
- salesperson + territory vs. salesperson + customer
- product + brand + model
- team + player
- is affinity natural, or process/event-based?
- is affinity within one context or several?

Addressing large dimensions

- is it really more than one dimension? split them up.
- extract subtype specifics to new dimensions (e.g. related product types)
- mini-dimensions: split out possibilities

Avoid NULL in dimensions

- Special-case roles for missing or bad data



PRODUCT

product_ key	row_type	product_code	product_name
0	Invalid	n/a	n/a
1	Unknown	n/a	n/a
101	Product	B57330-1	Cardboard Box
102	Product	B47770-2	Bubble Envelope

Used when a fact is supplied with an invalid product_code

Used when a fact arrives prior to dimensional context

Fig. 6-11

Behavioral dimensions

- Read this closely!
- Use facts to create new dimensions:
  - categorize customers by sales level / frequency
  - categorize products by popularity / seasonality
- A prelude to **feature engineering** in data mining

Handling Hierarchies

Hierarchies appear everywhere

- Many dimensions have inherent hierarchy
- Hierarchies play out in scoping context, summarization
- More than one hierarchy may be equally valid

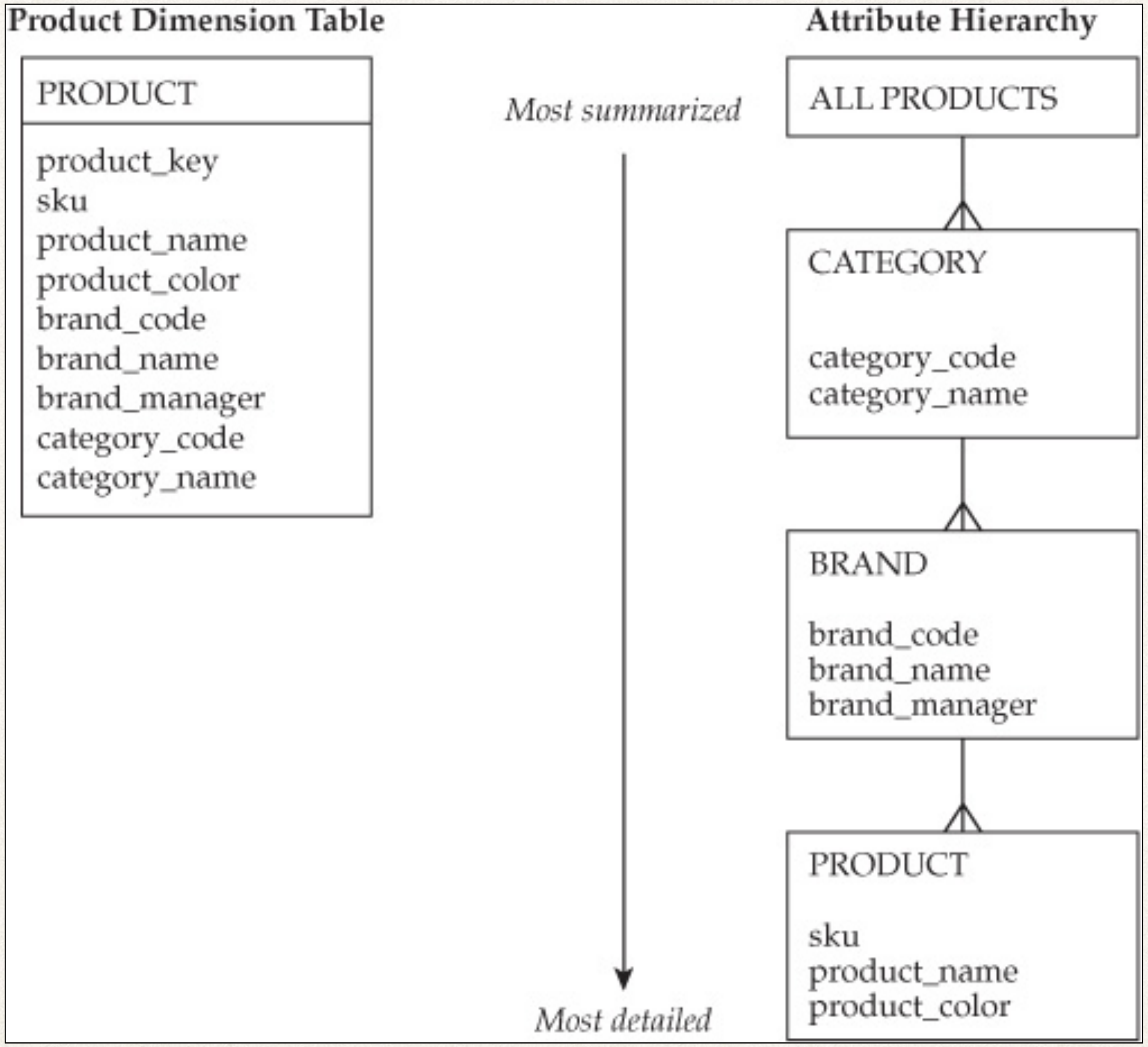


Fig. 7-2

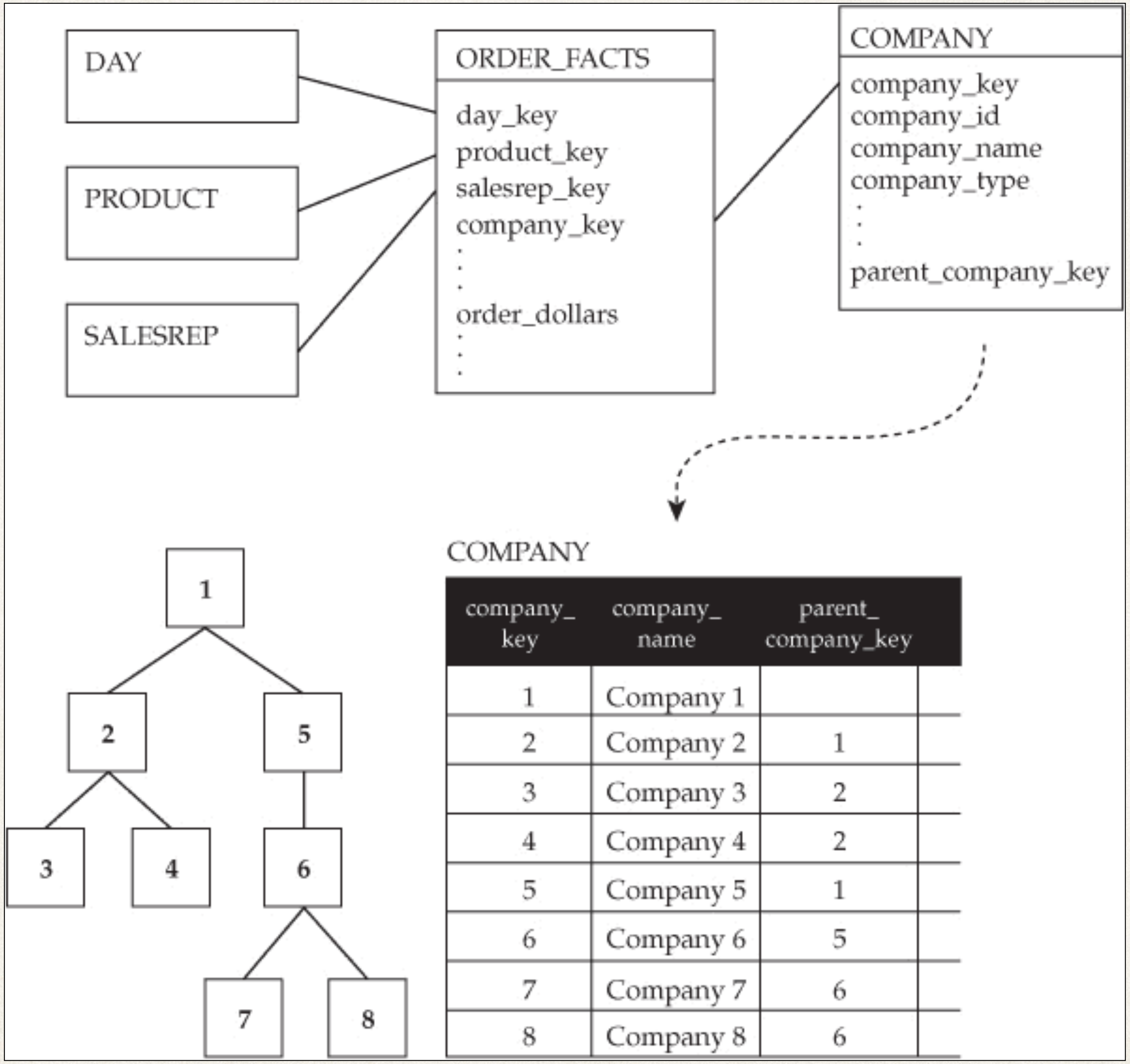
All Products (1) → Categories (25) → Brands (650) → Products (8000)

Instance (recursive) hierarchies

- Organizational structures
  - Departments / sections
  - Owners / subsidiaries
- Employee relationships (e.g. supervisor)
- Product components (parts / assemblies)

Instance hierarchy handling

- Recursive queries
- Flattened dimensions
- Bridge tables



DAY

PRODUCT

SALESREP

ORDER_FACTS

day_key
product_key
salesrep_key
company_key
.
.
order_dollars
.
.

COMPANY

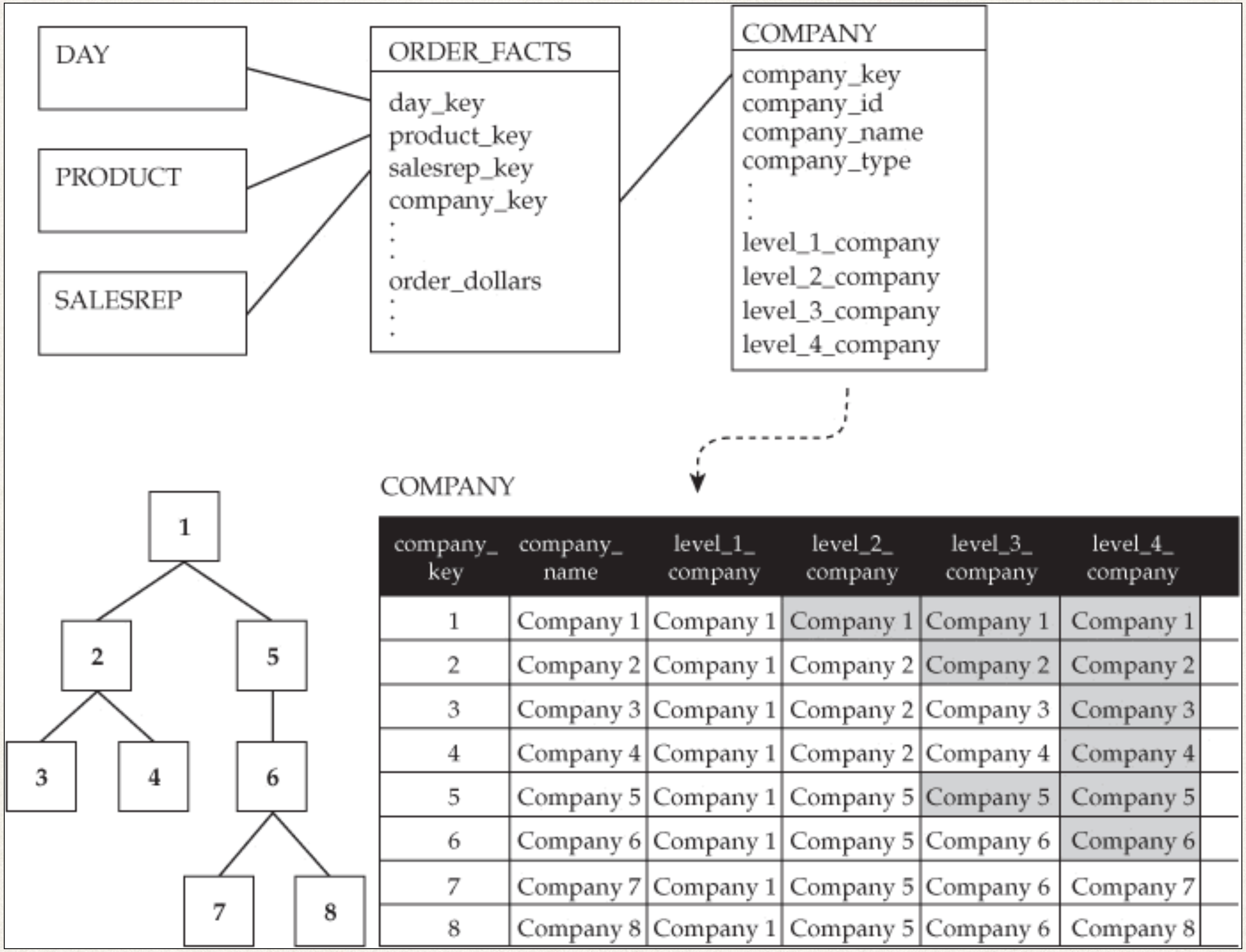
company_key
company_id
company_name
company_type
.
.
parent_company_key

COMPANY

company_ key	company_ name	parent_ company_key
1	Company 1	
2	Company 2	1
3	Company 3	2
4	Company 4	2
5	Company 5	1
6	Company 6	5
7	Company 7	6
8	Company 8	6

Fig. 10-3

Recursive company dimension structure

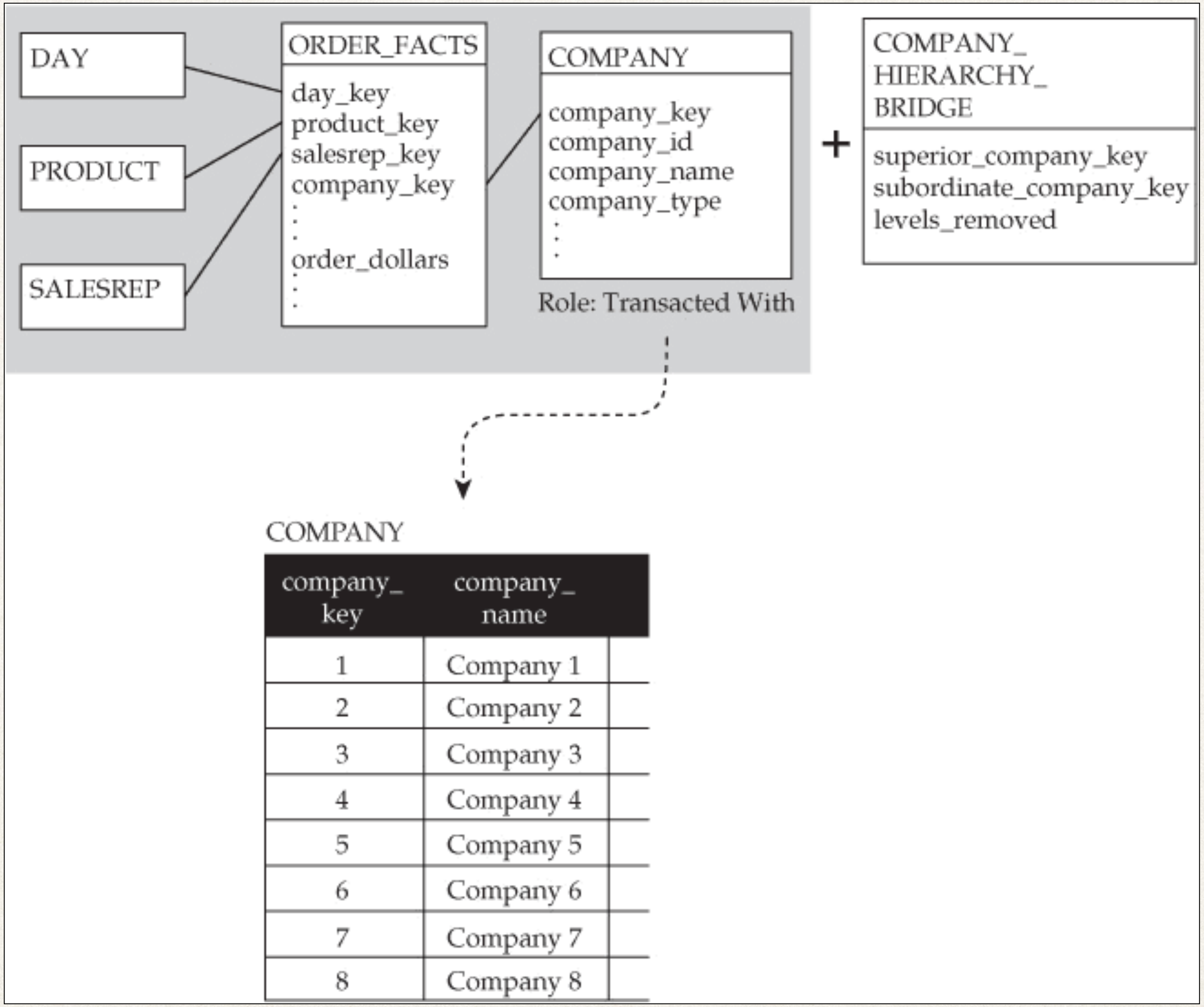


COMPANY

company_ key	company_ name	level_1_ company	level_2_ company	level_3_ company	level_4_ company
1	Company 1	Company 1	Company 1	Company 1	Company 1
2	Company 2	Company 1	Company 2	Company 2	Company 2
3	Company 3	Company 1	Company 2	Company 3	Company 3
4	Company 4	Company 1	Company 2	Company 4	Company 4
5	Company 5	Company 1	Company 5	Company 5	Company 5
6	Company 6	Company 1	Company 5	Company 6	Company 6
7	Company 7	Company 1	Company 5	Company 6	Company 7
8	Company 8	Company 1	Company 5	Company 6	Company 8

Fig. 10-4

Flattened corporate hierarchy dimension



COMPANY

company_ key	company_ name
1	Company 1
2	Company 2
3	Company 3
4	Company 4
5	Company 5
6	Company 6
7	Company 7
8	Company 8

Fig. 10-5

Bridge structure for corporate ownership hierarchy

- these structures occur frequently!
- see *Star Schema,* Chapter 10

Key tactics

- Document attribute hierarchies in dimensions
- Document dimension conformance
- Design to meet analysis needs, not all possible configurations

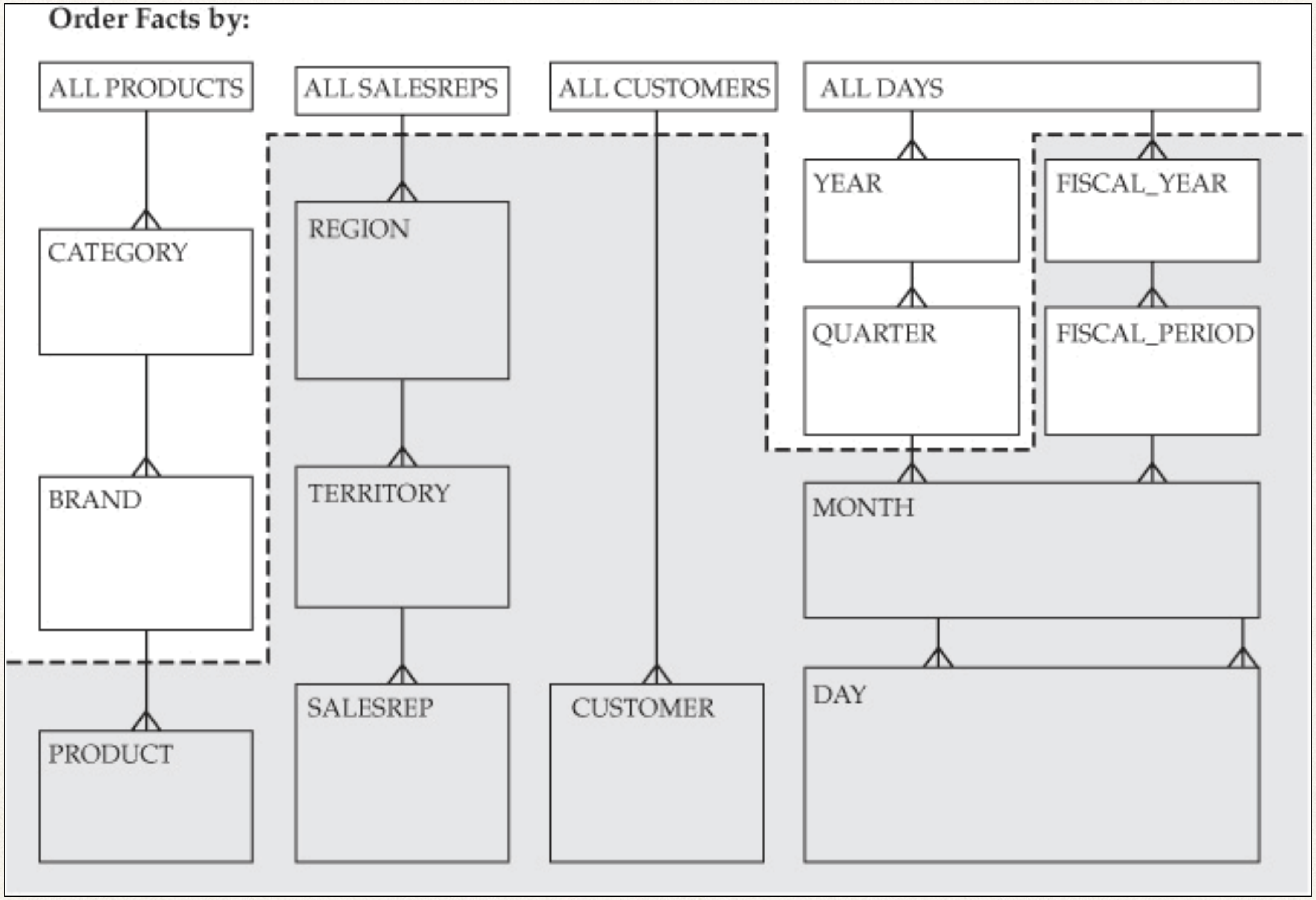


Fig. 7-4

Hierarchy diagram defines aggregate

Quick thoughts on Snowflakes

- Avoid urge to normalize in dimensional models
- Have a good reason:
  - BI products expect or benefit
  - Some database engines optimize for it
  - Multi-value / repeating groups
  - Cleaner ETL / modeling outweighs query / update penalty

Snapshots

So far: Transaction fact tables

- Facts as records of transaction / process / event at a specific granularity
- Facts are additive - sums have useful meaning
- Each record / row represents measurable facts based on what actually occurred
- "Sparse" - records of actualized activity, not theoretical possibilities

Transaction fact tables awkward for:

- Status (inventory levels)
- Sensor readings (changing state)
- Prices or balances (moving averages)

- Not transactions, not additive, not events, not sparse

Bank accounts: txns vs. balances

- Reasonable to study transaction patterns: these fit the transaction fact table model well

- What about demographics or seasonality of balances?

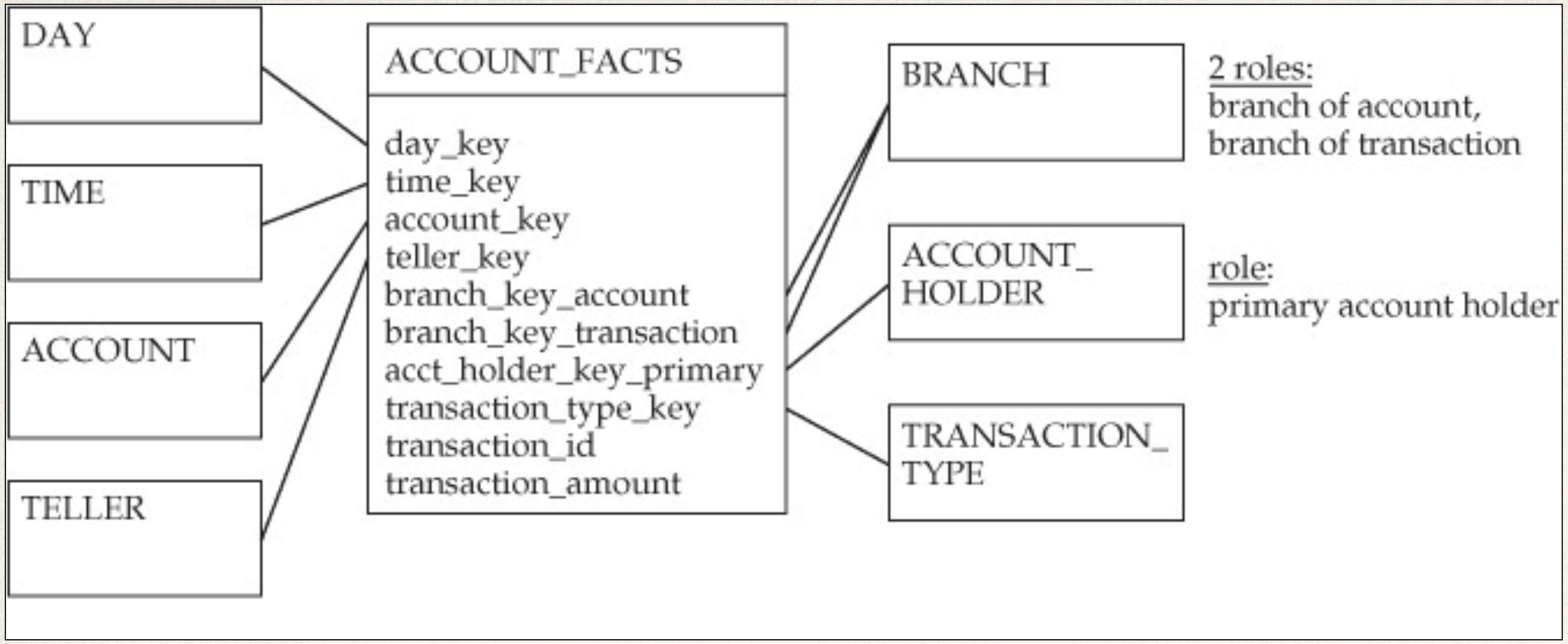
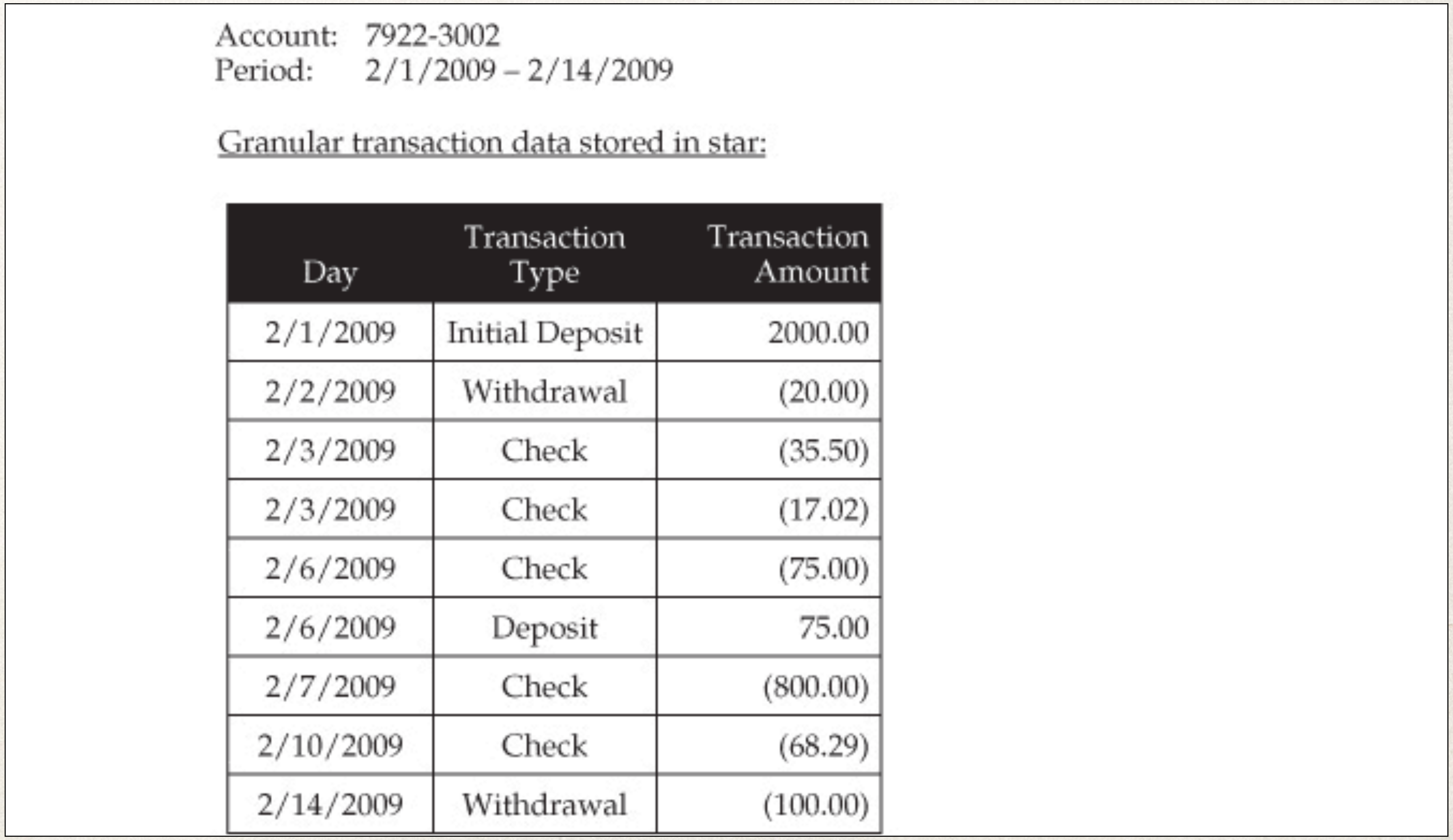


Fig. 11-1

Account transaction fact table design & instances



Account: 7922-3002
Period: 2/1/2009 – 2/14/2009

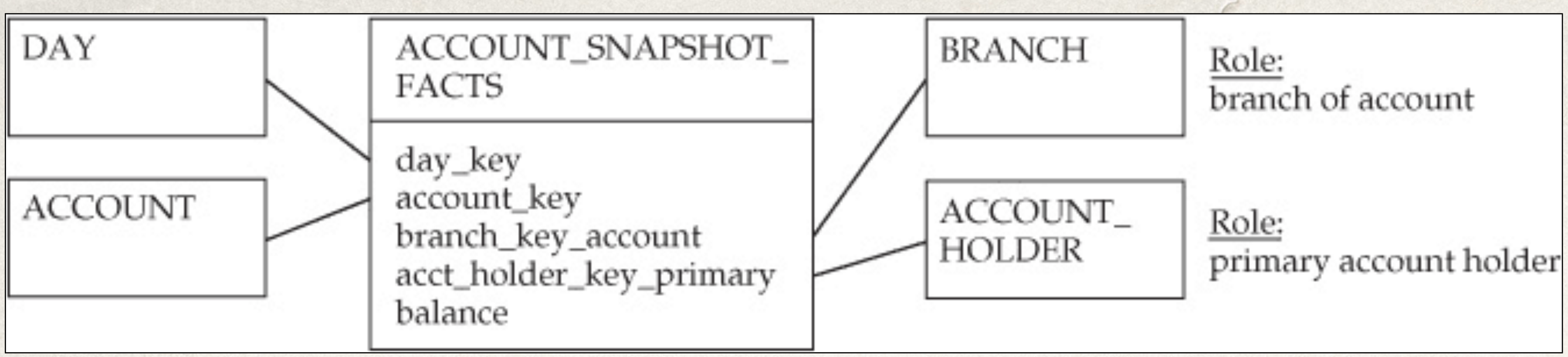
Granular transaction data stored in star:

Day	Transaction Type	Transaction Amount
2/1/2009	Initial Deposit	2000.00
2/2/2009	Withdrawal	(20.00)
2/3/2009	Check	(35.50)
2/3/2009	Check	(17.02)
2/6/2009	Check	(75.00)
2/6/2009	Deposit	75.00
2/7/2009	Check	(800.00)
2/10/2009	Check	(68.29)
2/14/2009	Withdrawal	(100.00)

Issues with transactions

- Tells us nothing about account balances
- Balances are not additive
- Many days without transactions
- Many days with transactions

Account snapshot fact table design & instances


DAY

ACCOUNT

ACCOUNT_SNAPSHOT_FACTS

day_key
account_key
branch_key_account
acct_holder_key_primary
balance

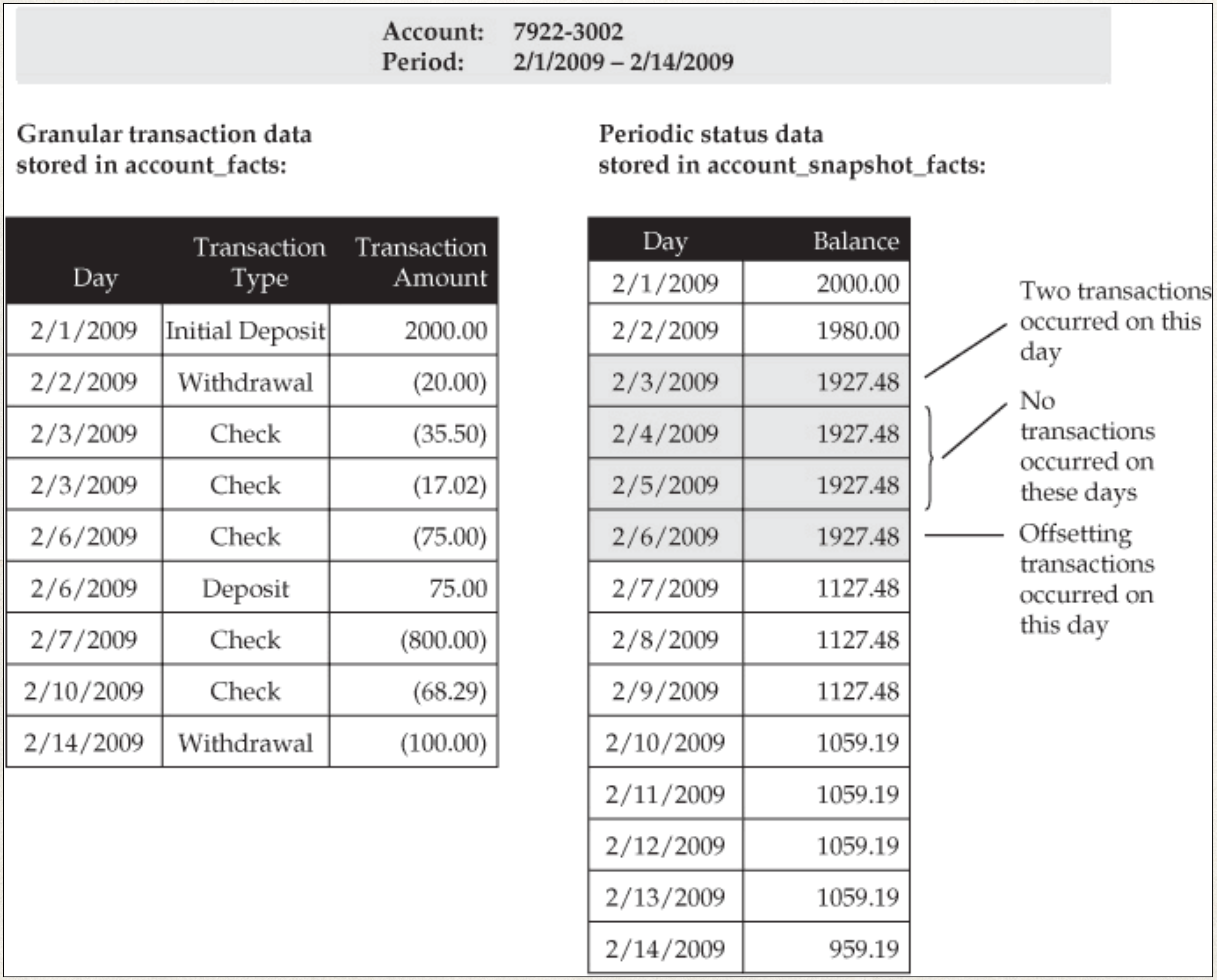
BRANCH

Role:
branch of account

ACCOUNT_HOLDER

Role:
primary account holder

Fig. 11-2


Account: **7922-3002**
Period: **2/1/2009 – 2/14/2009**

Granular transaction data stored in account_facts:

Day	Transaction Type	Transaction Amount
2/1/2009	Initial Deposit	2000.00
2/2/2009	Withdrawal	(20.00)
2/3/2009	Check	(35.50)
2/3/2009	Check	(17.02)
2/6/2009	Check	(75.00)
2/6/2009	Deposit	75.00
2/7/2009	Check	(800.00)
2/10/2009	Check	(68.29)
2/14/2009	Withdrawal	(100.00)

Periodic status data stored in account_snapshot_facts:

Day	Balance
2/1/2009	2000.00
2/2/2009	1980.00
2/3/2009	1927.48
2/4/2009	1927.48
2/5/2009	1927.48
2/6/2009	1927.48
2/7/2009	1127.48
2/8/2009	1127.48
2/9/2009	1127.48
2/10/2009	1059.19
2/11/2009	1059.19
2/12/2009	1059.19
2/13/2009	1059.19
2/14/2009	959.19

Two transactions occurred on this day

No transactions occurred on these days

Offsetting transactions occurred on this day

Fig. 11-3

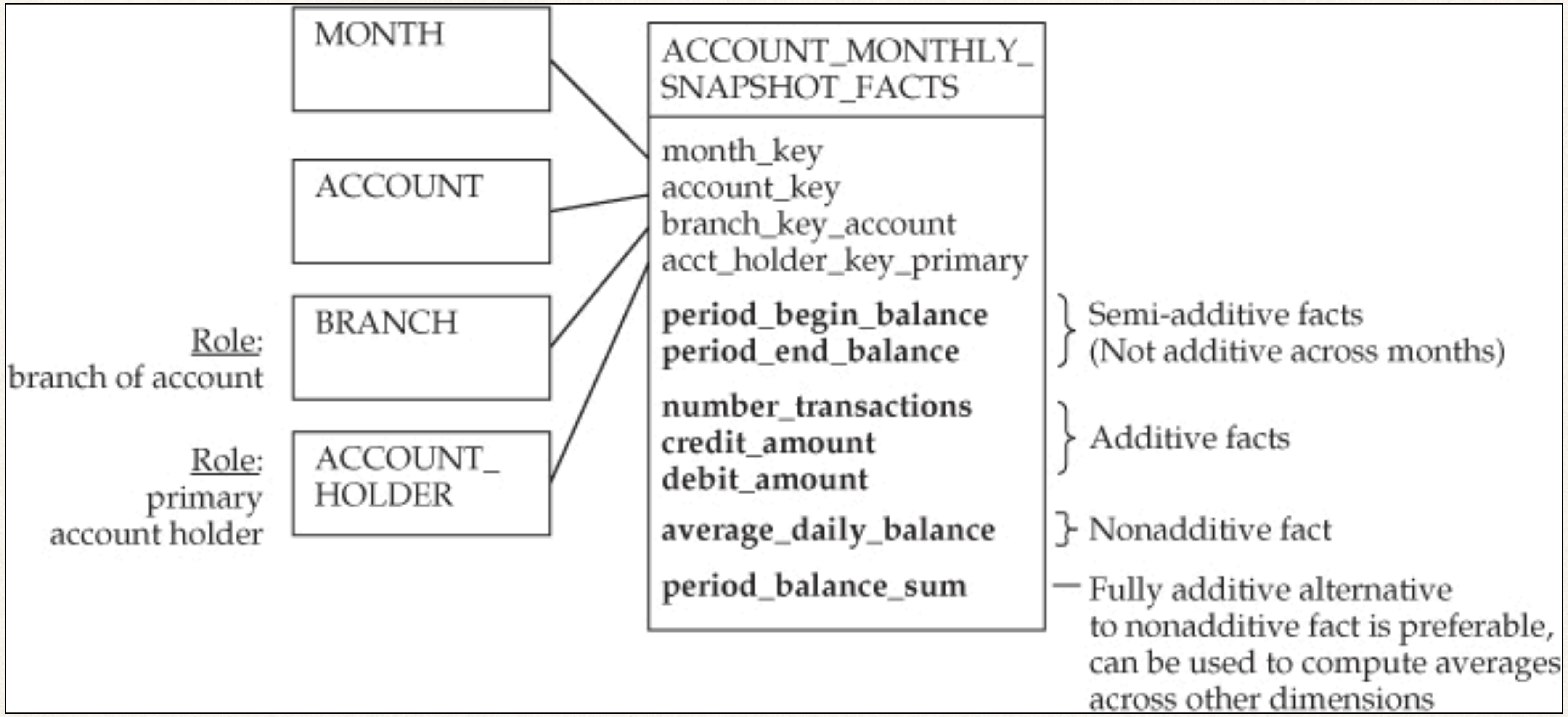


Fig. 11-4

Additional snapshot facts fit well together - natural affinity

Snapshot fact tables - differences

- **Dense**, not sparse
- **Periodic**, not event-based
- **Semi-additive** (e.g. account balances, temperatures)
- Granularity aligns with dimensions, not events

Transactions and Snapshots

- These two designs **complement** each other
- Each may be viable for different kinds of analysis
- To simplify ETL, transaction table should be source for snapshot tables
- In this way, snapshot tables **derive** from snapshots

Accumulating Snapshots

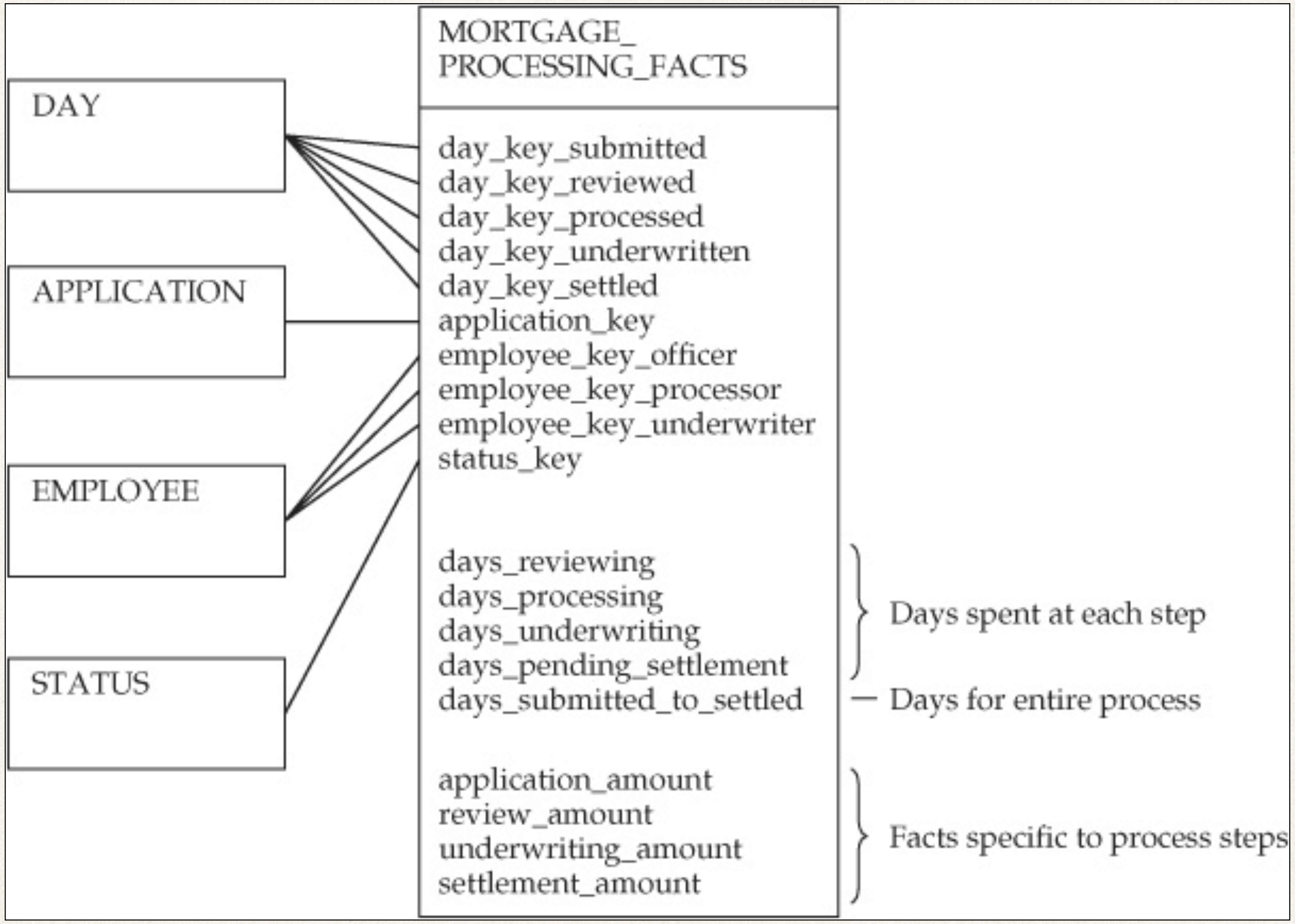
- Transaction fact tables support measuring simple processes (based on single events)

- Snapshot fact tables support measuring patterns in state over time (based on periodic observations)

- How do we measure complex processes?

What's a complex process?

- think "workflow" - more than one service station, more than one role type, and transitions between each
- measure time at each step in the process - **intervals**, not events
- use to find **bottlenecks** to feed into process modeling and simulation, among others



DAY

APPLICATION

EMPLOYEE

STATUS

MORTGAGE_
PROCESSING_FACTS

day_key_submitted
day_key_reviewed
day_key_processed
day_key_underwritten
day_key_settled
application_key
employee_key_officer
employee_key_processor
employee_key_underwriter
status_key

days_reviewing
days_processing
days_underwriting
days_pending_settlement
days_submitted_to_settled

application_amount
review_amount
underwriting_amount
settlement_amount

Days spent at each step

Days for entire process

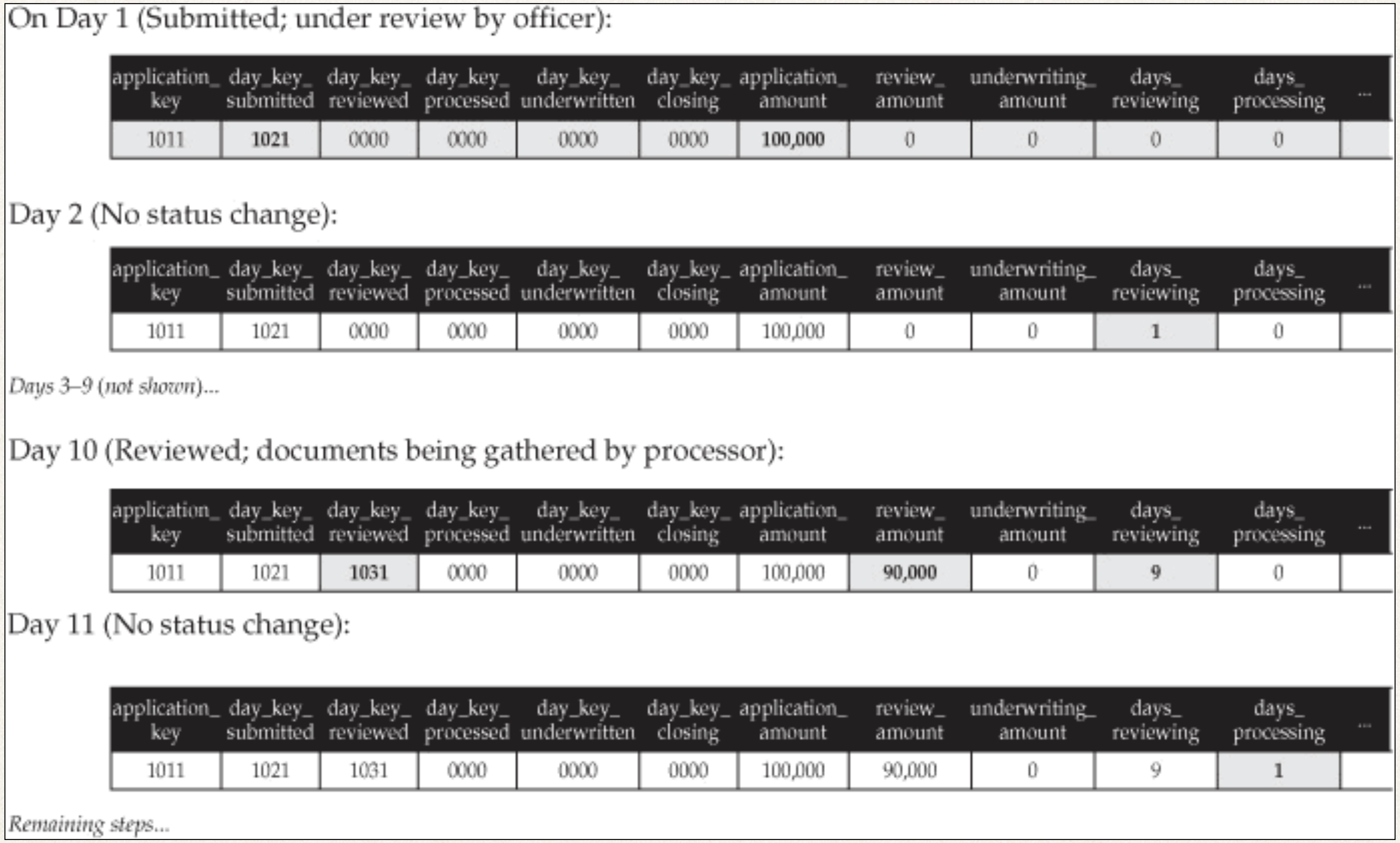
Facts specific to process steps

Fig. 11-6

Accumulating snapshot fact table design

Values accumulate over time

- Each row represents one process - multiple steps
- Row is updated as time passes and processing continues at one stage or another
- Time may pass with little or no progress; time elapsed increases at one stage or another



On Day 1 (Submitted; under review by officer):

application_key	day_key_submitted	day_key_reviewed	day_key_processed	day_key_underwritten	day_key_closing	application_amount	review_amount	underwriting_amount	days_reviewing	days_processing	...
1011	**1021**	0000	0000	0000	0000	**100,000**	0	0	0	0	

Day 2 (No status change):

application_key	day_key_submitted	day_key_reviewed	day_key_processed	day_key_underwritten	day_key_closing	application_amount	review_amount	underwriting_amount	days_reviewing	days_processing	...
1011	1021	0000	0000	0000	0000	100,000	0	0	**1**	0	

Days 3–9 (not shown)...

Day 10 (Reviewed; documents being gathered by processor):

application_key	day_key_submitted	day_key_reviewed	day_key_processed	day_key_underwritten	day_key_closing	application_amount	review_amount	underwriting_amount	days_reviewing	days_processing	...
1011	1021	**1031**	0000	0000	0000	100,000	**90,000**	0	**9**	0	

Day 11 (No status change):

application_key	day_key_submitted	day_key_reviewed	day_key_processed	day_key_underwritten	day_key_closing	application_amount	review_amount	underwriting_amount	days_reviewing	days_processing	...
1011	1021	1031	0000	0000	0000	100,000	90,000	0	9	**1**	

Remaining steps...

Fig. 11-7

Evolution of one accumulating snapshot record

Accumulating vs. transaction facts

- Accumulating snapshots excel at capturing / updating intervals
- Ideal for studying performance for each stage
- Complements transaction fact tables where volume, workloads, etc. may be more easily queried
- Like snapshots, transaction fact table should be source to simplify ETL

Project 02 work session