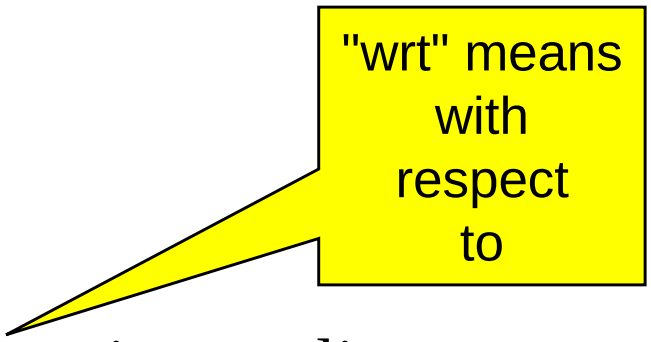
09 – B
ETL Issues and Techniques

Semantic Impedance (SI)
Semantic Impedance Mismatch (SIM)
Integration Techniques
Domain Studies & Cross Footing

ETL Issues and Techniques

- Extraction Transformation and Loading (ETL)
  - useful for DW, DM (including ODM), and / or ODS
  - not a "silver-bullet" (not a cure to all our problems)
  - requires experts in
    - the problem domain
    - the underlying technology
    - the data models (source and destination)
  - can be very useful especially wrt improving quality
    - impacts OLAP (Reports and Queries)
    - impacts **MINING**
    - impacts "single version of the truth"
    - etc.



ETL Issues and Techniques

- our focus here is Transformation and Cleansing
  - Extracting and Loading are important but relatively simple (at least by comparison)
- Transformation and Cleansing attempt to
  - integrate the data (part of the definition of a DWE!)
  - identify semantic impedance
  - address semantic impedance issues
  - identify / quantify data quality
  - address data quality issues

Semantic Impedance and **SIM**

- what does Impedance mean?
  - impedance (an informal definition from Physics / EE):
    - impedance is the measure of opposition to a sinusoidal alternating current
- definition of Semantic Impedance (**SI**) :
  - the measure of opposition to exchanging data (both meaning and content) between two systems
  - when this opposition is "large", we say there is Semantic Impedance Mismatch (**SIM**)
    - i.e., **SIM** occurs when one system has either an inadequate or an excessive ability to accommodate the input from another

Semantic Impedance and **SIM**

- **SI** (and possibly **SIM**) exists across different
  - data modeling technologies / languages
  - data model types
  - data models
- **SI** (and possibly **SIM**) can also be caused by many other things
  - including Non-Technical factors
- i.e., no matter what we do, we cannot eliminate **SI** completely…

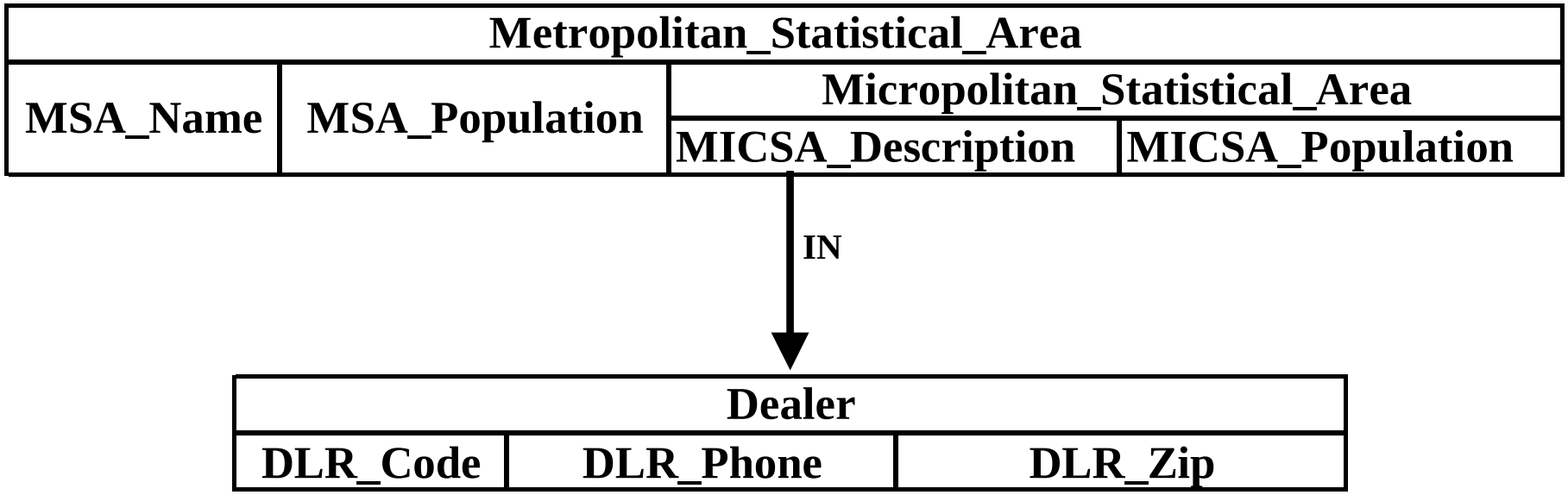
Semantic Impedance and **SIM**

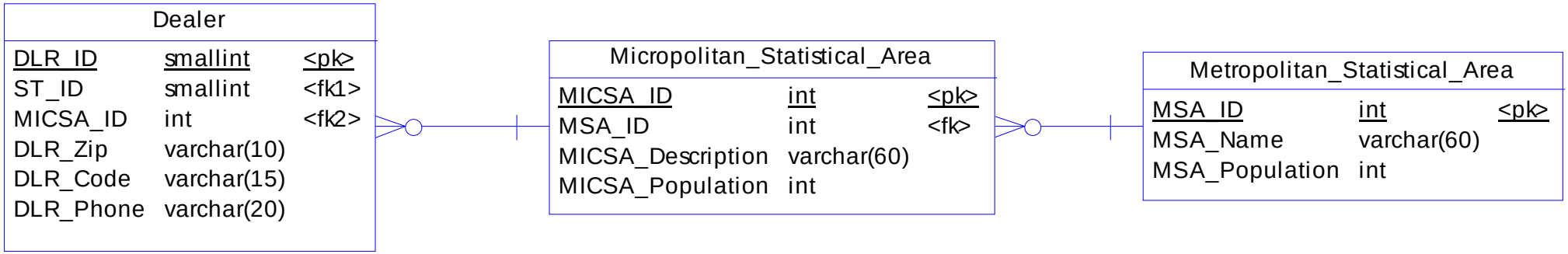
- discussion: **SI / SIM** as a result of
  the data modeling technologies / languages
- begin **NON-Testable** material…
  - examples of data modeling technologies / languages:
    - network data model
    - hierarchical data model
    - relational data model
    - object-oriented data model
    - multidimensional data model

SIM Network Data Model Example

- the network data model uses:
  - record types, data items, and set types
  - repeating, group, & repeating group data items
  - owners and members
  - multimember sets (multi-type relationships)
  - linking / dummy members
  - no M-M relationships (must use 2 sets and 1 record)
  - ordered relationships
  - virtual fields
  - pointers!!!

SIM Network Data Model Example





CAVEAT: Not Accurate Or Complete!

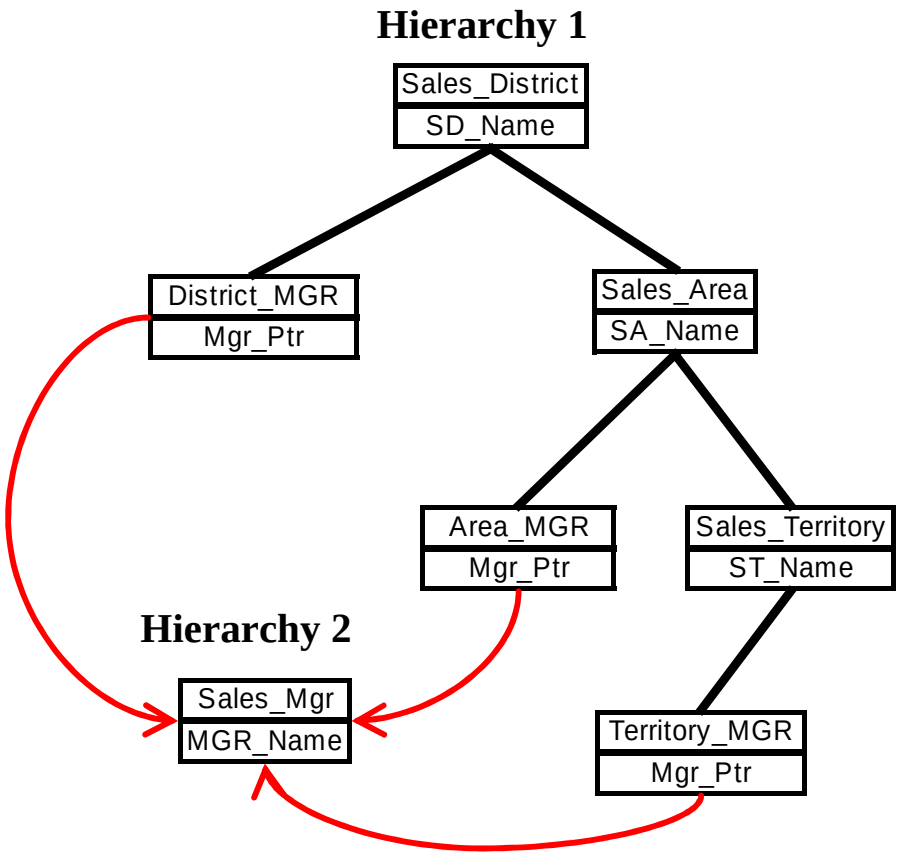
- SCHEMA NAME IS SALES_ORG
- RECORD NAME IS DEALER
  DUPLICATES ARE NOT ALLOWED FOR DLR_CODE
  DLR_ZIP TYPE IS CHARACTER 10
  DLR_CODE TYPE IS CHARACTER 15
  DLR_PHONE TYPE IS CHARACTER 20

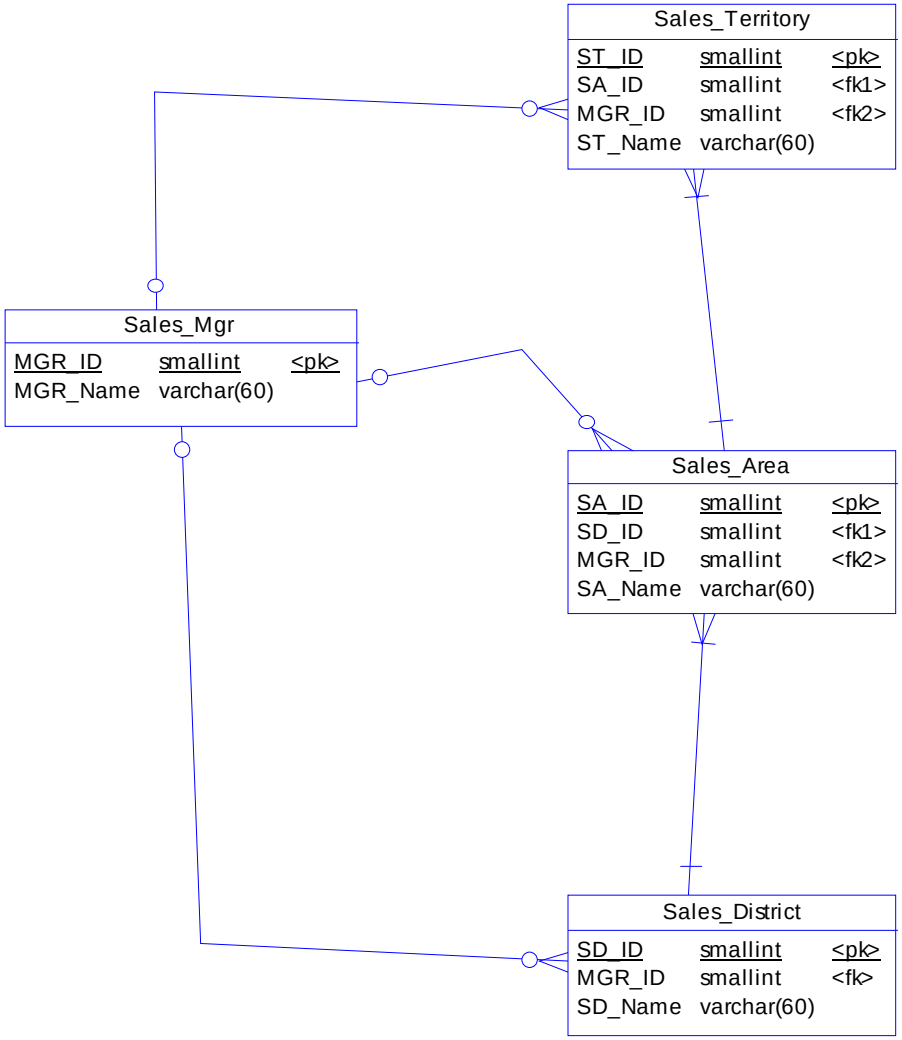
- SET NAME IS IN
  OWNER IS METROPOLITAN_STATISTICAL_AREA
  ORDER IS SORTED BY DEFINED KEYS
  MEMBER IS DEALER
  KEY IS ASCENDING DLR_CODE

SIM Hierarchical Data Model Example

- hierarchical data model uses:
  - record types
  - parent-child-relationship (PCR) types
  - parent-child hierarchy → trees not graphs!
  - ordered siblings
  - virtual pointers (VP)
  - virtual PCR types (uses VP for M-M)

SIM Hierarchical Data Model Example





SI / SIM Issues and Causes

- back to **Testable** topics…consider **SI / SIM** as a result of technical considerations—in general, a result of how things are modeled for different data model types…
  - different languages within CDM / LDM / PDM
    - like the different LDM / PDM examples in previous slides
  - conceptual vs. logical vs. physical constructs
    - e.g. "entity" vs. "relational table" vs. "ORACLE table"
  - application Oriented vs. Subject Oriented models
    - focus on semantics used for design
  - OLTP vs. OLAP models
    - focus on how models are being used

SI / SIM Issues and Causes

- SI / SIM across data models (even of same type)
  - different significance for the same concept
    - consider "manufacturing plant" in our project OLTP models
  - different understandings of the same concept
    - this is why stovepipes are a real, and possible threat…
  - different abilities to capture the concepts
    - data modeling tools
    - data modeling techniques
    - data modeler experience
    - performance considerations
    - evolution of the enterprise
    - available documentation and knowledge

SI / SIM Issues and Causes

- SI / SIM from non-technical considerations
  - corporate mergers and acquisitions
  - changing staff
  - customer relationship management (CRM) initiatives
    - multiple applications with data about same customer
  - migration from legacy systems
    to enterprise resource planning (ERP) systems
  - use of external data sources
    - we had one example of this in our project…
  - etc.

Example **SI / SIM** Considerations

- **NON-Testable** "Relationship aspects" examples
  - optional vs. mandatory
  - maximum and minimum cardinalities
    - 1-1, 1-M, M-1, M-M or more explicit {3,9} to {0,1}
  - inheritance
    - disjoint or overlapping (a.k.a. is it mutually exclusive?)
    - partial or total participation (a.k.a. is it abstract or concrete?)
  - identifying and non-identifying relationships
  - type-less and "N-arry"

Example **SI / SIM** Considerations

- **NON-Testable** "Attribute aspects" examples
  - domains
    - data types (encoding) and formats (pictures)
    - precisions and scales, lengths, defaults, units
  - optional vs. mandatory (a.k.a. is NULLABLE?)
  - scalar (Singular) vs. aggregate (Plural)
  - simple vs. complex
  - constraints and indexes?
    - identifying? unique? any interdependencies?

Example **SI / SIM** Considerations

- **NON-Testable** "Entity aspects" examples
  - what are the identifiers?
  - what are the candidate keys?
  - is it Normalized?
  - what are the functional dependencies?
  - is this part of the problem domain or part of the application's solution?
    - similar to application-oriented versus subject-oriented but not quite the same…

Integration Techniques

- back to **Testable** topics…consider merging Data
  - are the entities semantically compatible?
  - are their keys mutually exclusive?
  - are their attributes mutually exclusive?
  - are the instances mutually exclusive?
    - if not then precedence rules are very important
    - e.g. suppose the same customer is in two systems with a different value for the address, phone, or payment status…
    - is this an omission?
    - is this the result of an update / requested change?
    - how do we integrate?

Integration Techniques

- back to **Testable** topics…consider merging Data
  - let's consider three different scenarios
  - this is NOT AN EXHAUSTIVE SET!
  - should illustrate the basic concepts involved
- in each scenario:
  - there are two (2) source systems (Site-X, Site-Y)
    - OLTP on RDBMS
  - there are two (2) source tables
  - there is one (1) destination system (Site-Z)
    - staging or "DW" on RDBMS
  - there is one (1) destination (DIM) table

Example: Merging Data Scenario #1

- suppose (across the two source tables) we have:
  - key column types
    - semantically unrelated (**not** meaning the same thing)
    - merely a database-specific PK (**not** enterprise-wide identifier)
  - key column values
    - possibly overlapping values, but no overlapping rows
    - i.e., the same value across X and Y for a key column is **NOT** referring to same instance enterprise-wide
  - non-key column types
    - semantically equivalent (meaning **the same** thing)
  - non-key column values
    - no missing values, no conflicting values, no redundant values

Example: Merging Data Scenario #1

Prod

PRD_NO	FLAVOR	_SIZE_	_SKU__
100001	apple	10 oz.	DRK001
100002	cherry	10 oz.	DRK002
100003	grape	20 oz.	DRK003
100004	lime	30 oz.	DRK004
100005	lemon	40 oz.	DRK005

The other columns are SEMANTICALLY EQUIVALENT

*Although the values for the PK might overlap, in this example they have **NO** semantic equivalence IOW, they are disjoint data sets*

PRD

PRDNUM	PRDTYP	PRDOZS	PRDSKU
100003	mixed	10 oz.	DRK029
100004	banana	20 oz.	DRK030
100005	kiwi	10 oz.	DRK031
100006	orange	16 oz.	DRK032

Case 1- Source

Example: Merging Data Scenario #1

The tables were completely compatible and the row instances were non-overlapping (mutually exclusive)

Product

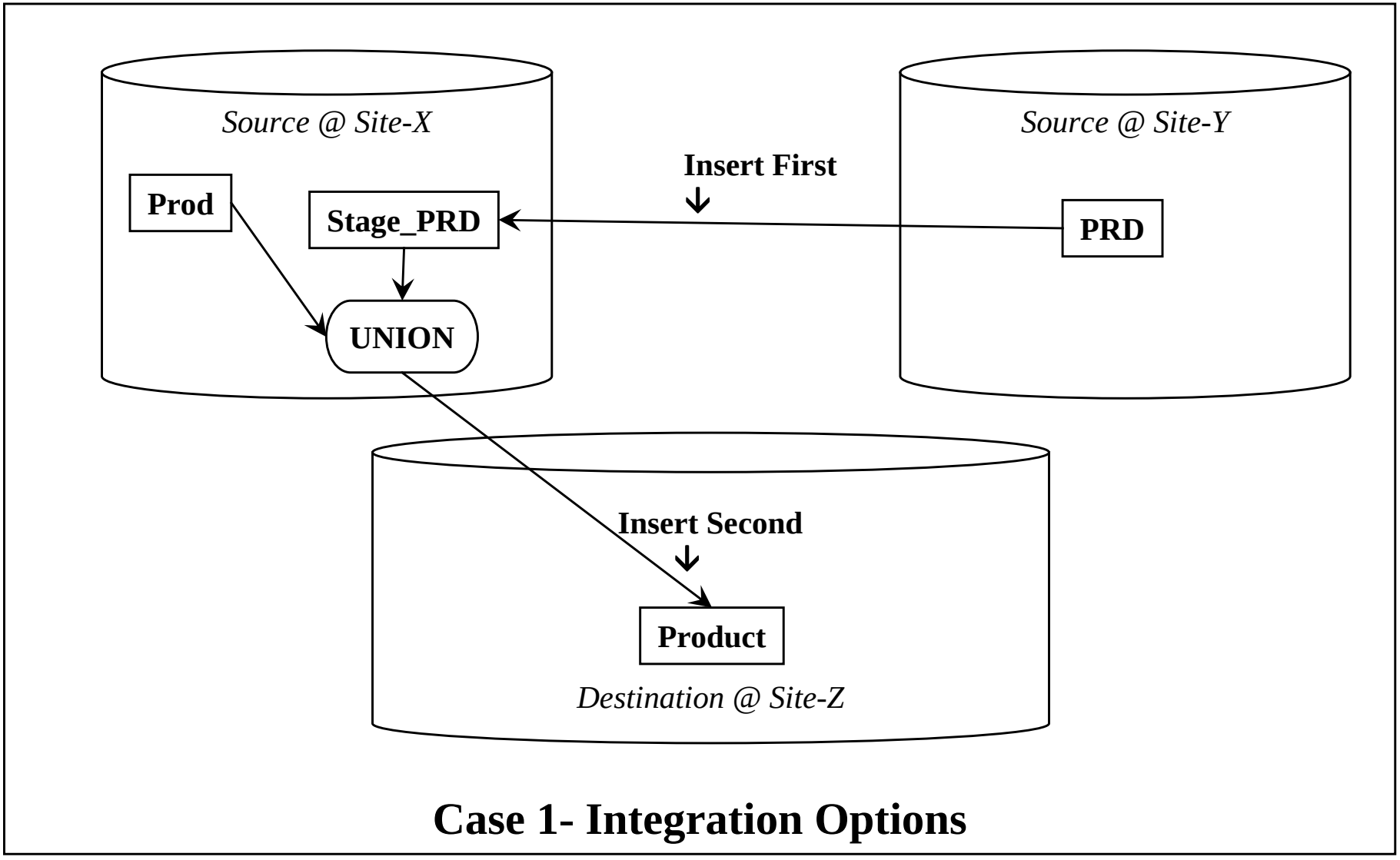
Product_Key	Product_Flavor	Product_Size	Product_SKU
SURK-000001	apple	10 oz.	DRK001
SURK-000002	cherry	10 oz.	DRK002
SURK-000003	grape	20 oz.	DRK003
SURK-000004	lime	30 oz.	DRK004
SURK-000005	lemon	40 oz.	DRK005
SURK-000006	mixed	10 oz.	DRK029
SURK-000007	banana	20 oz.	DRK030
SURK-000008	kiwi	10 oz.	DRK031
SURK-000009	orange	16 oz.	DRK032

Case 1- (Integrated) Destination

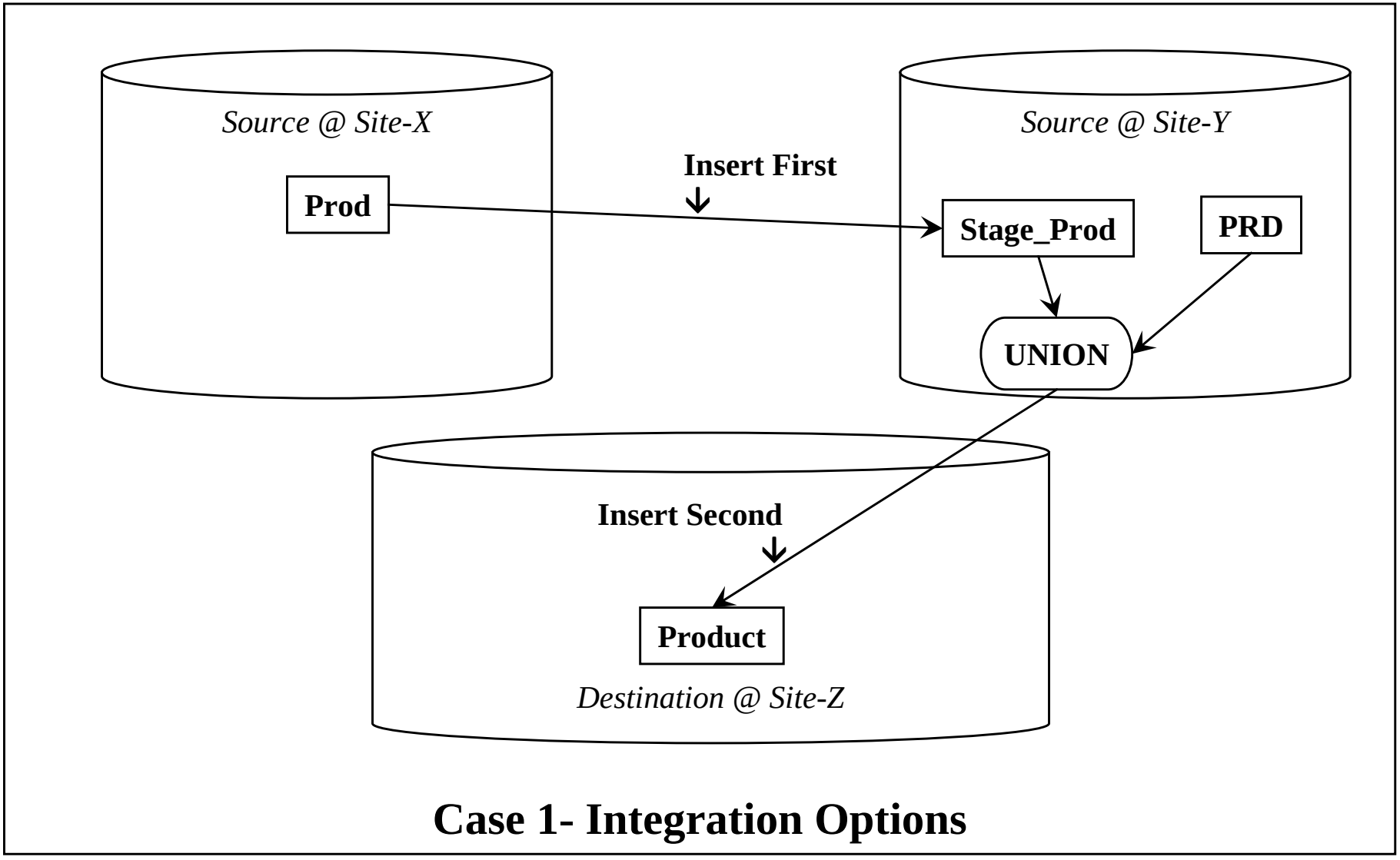
Example: Merging Data Scenario #1

- in this scenario, we have very low semantic impedance (**SI**)—no **SIM**!
- how do we integrate Scenario #1?
  - what is the main SQL operation?
  - where do we perform this operation?
  - what are the tradeoffs?

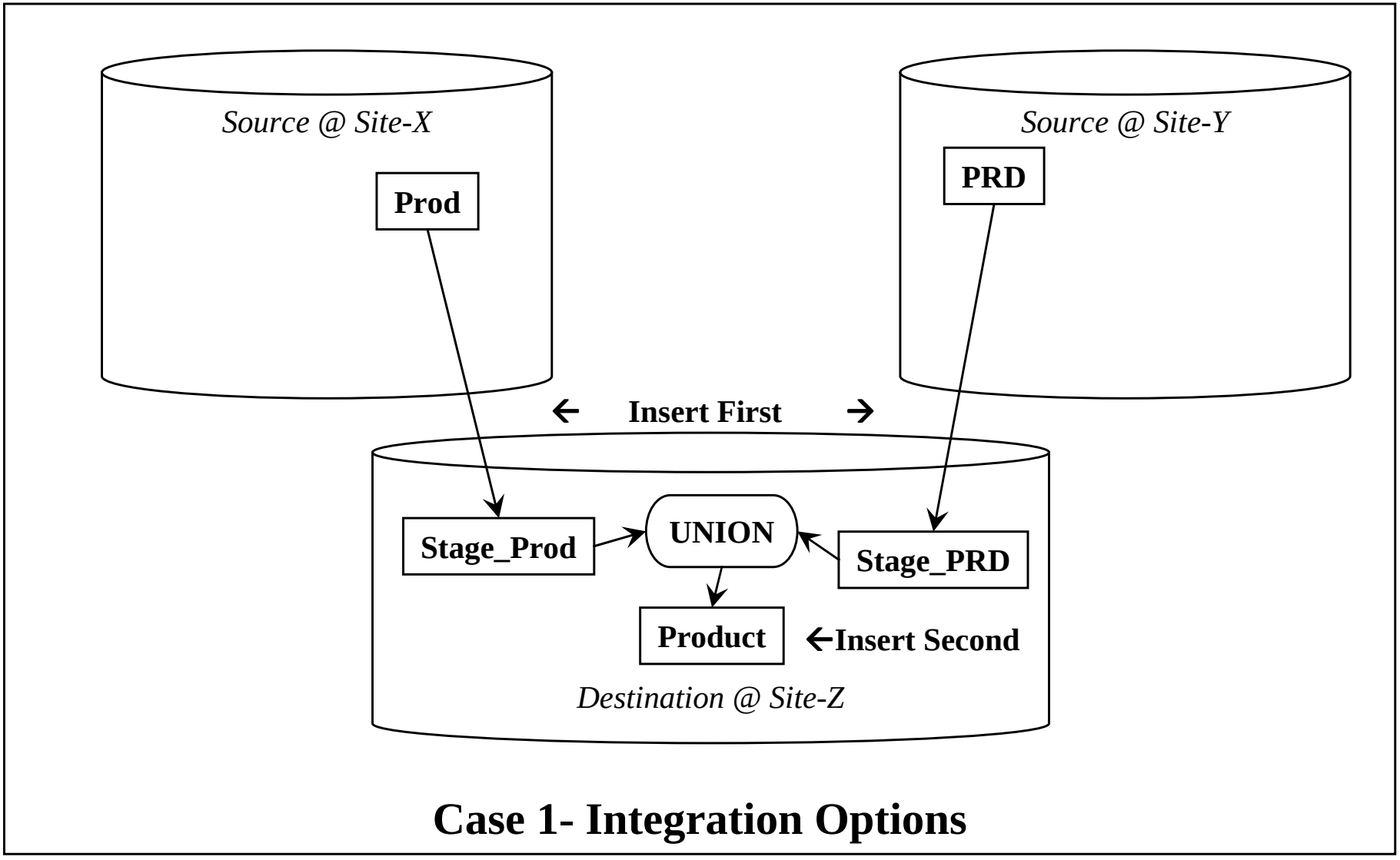
Example: Merging Data Scenario #1-A



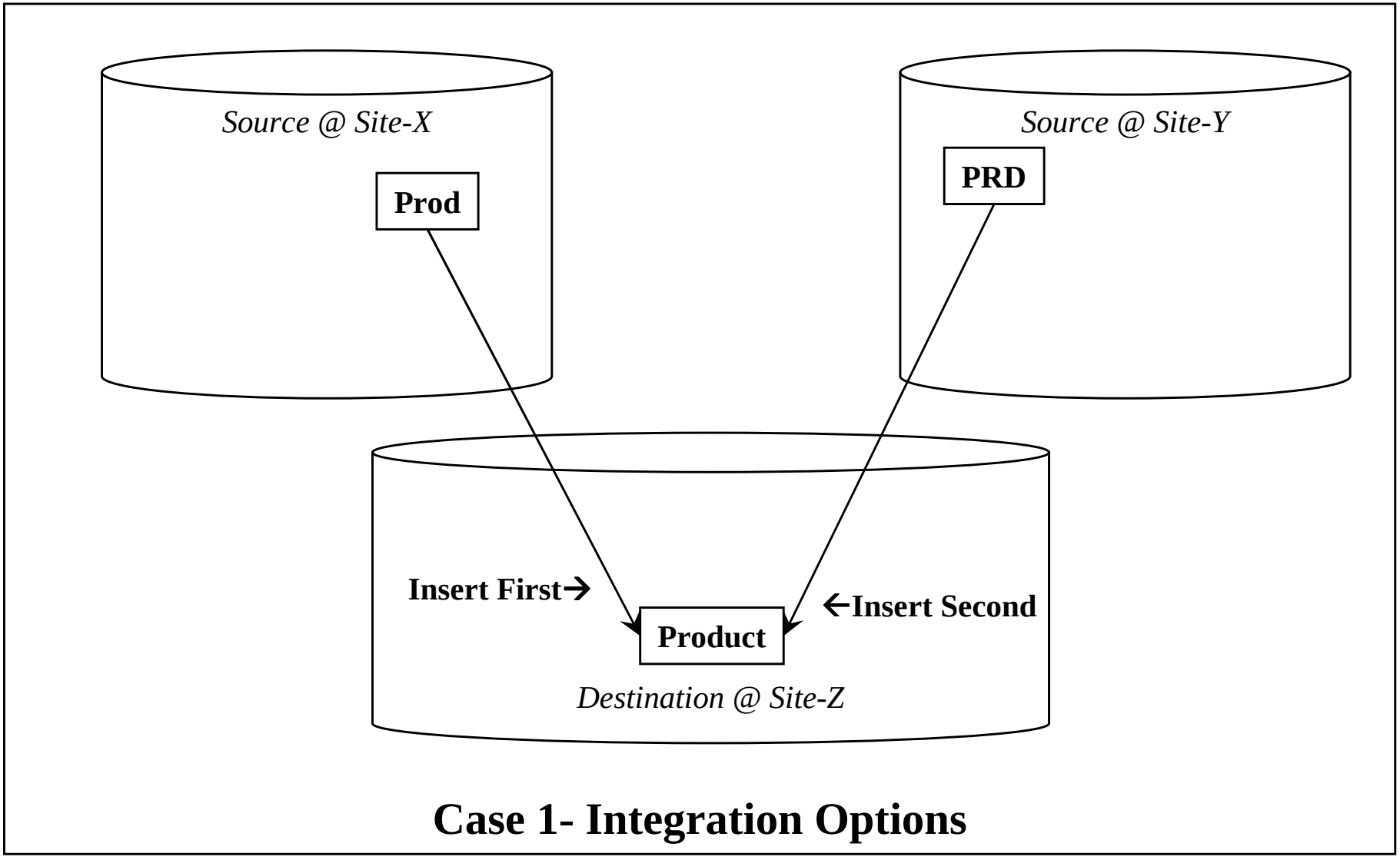
Example: Merging Data Scenario #1-B



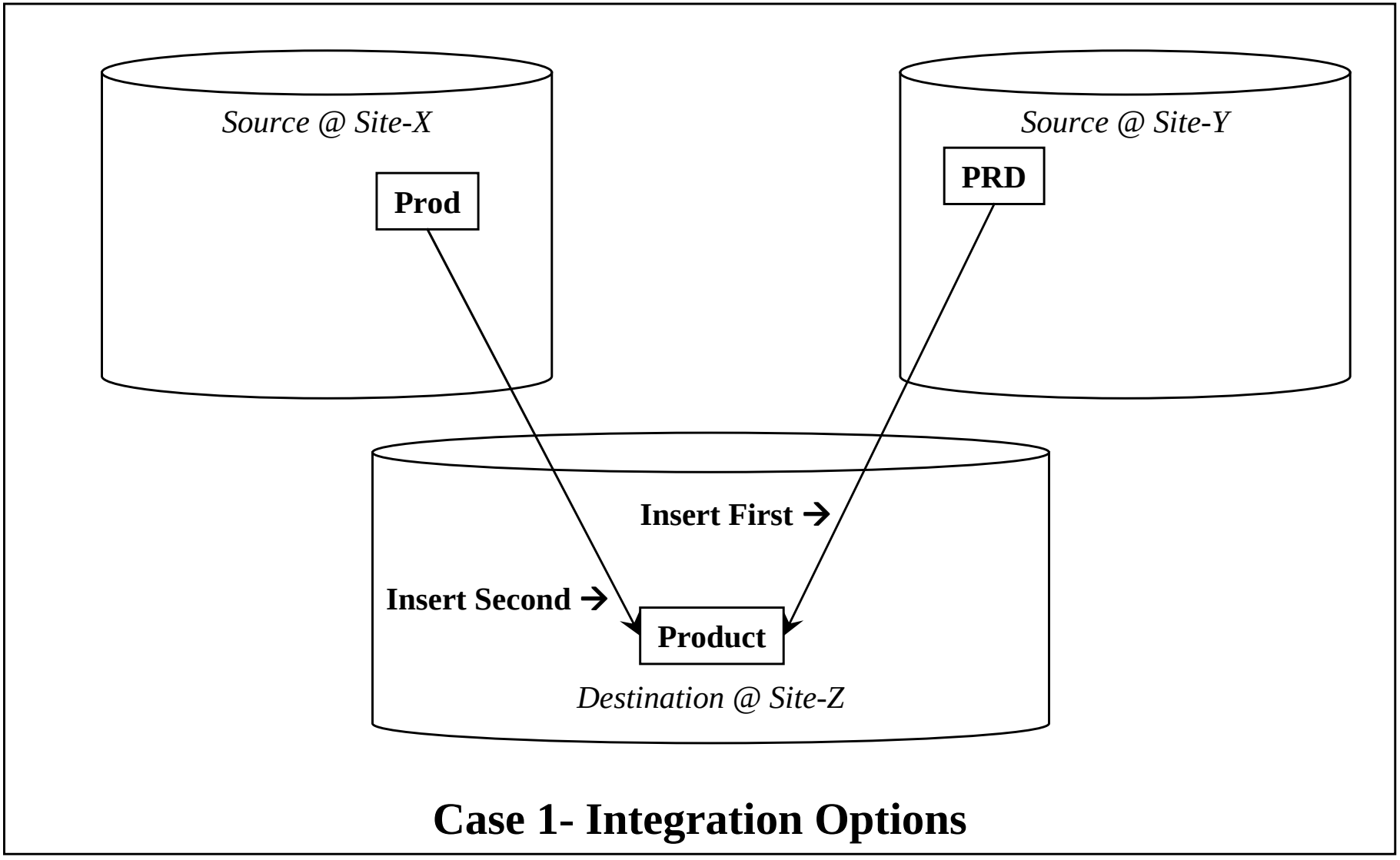
Example: Merging Data Scenario #1-C



Example: Merging Data Scenario #1-D



Example: Merging Data Scenario #1-E



Example: Merging Data Scenario #2

- suppose (across the two source tables) we have:
  - key column types
    - semantically equivalent (meaning **the same** thing)
    - IOW, this is an enterprise-wide identifier!
  - key column values
    - totally overlapping values, totally overlapping rows
    - same value for key refers to same instance
    - no missing instances, no extra instances
  - non-key column types
    - semantically unrelated (**not** meaning the same thing)
  - non-key column values
    - no missing values, no conflicting values, no redundant values

Example: Merging Data Scenario #2

P_info

_SKU__	_clr__	_size_	_msrp_
200001	brown	medium	$52.00
200002	red	small	$61.99
200003	black	large	$84.99
200004	yellow	medium	$62.99
200005	green	small	$47.99

In this example suppose the PKs are SEMANTICALLY EQUIVALENT

Pro

_SKU__	_dept_	_status_	_cost_
200001	kids	on-order	$33.99
200002	kids	in-stock	$43.99
200003	kids	in-stock	$50.99
200004	mens	on-order	$35.99
200005	womens	in-stock	$42.99

Case 2- Source

Product

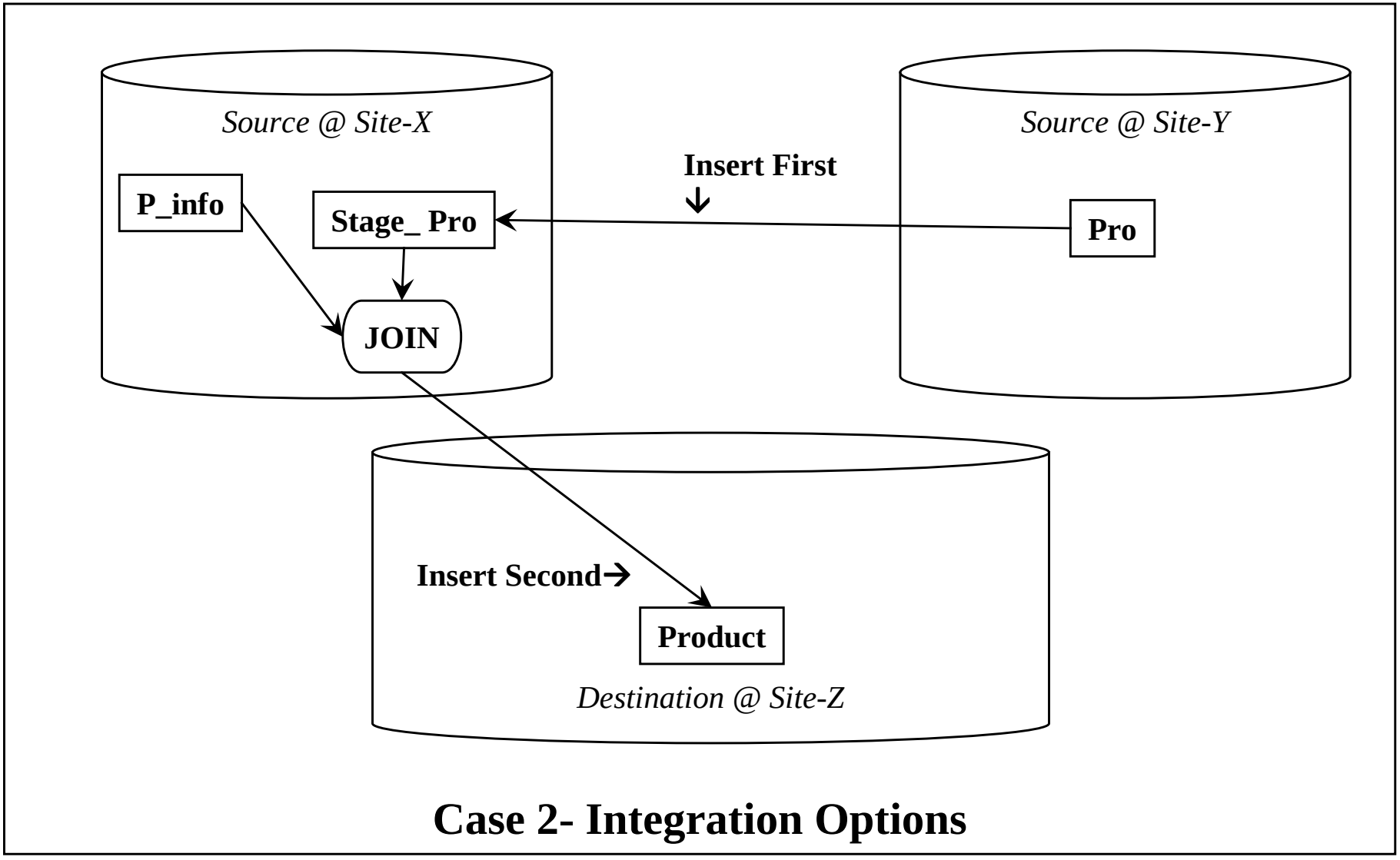
__Surk_	_SKU__	_clr__	_size_	_msrp_	_dept_	_status_	_cost_
0000001	200001	brown	medium	$52.00	kids	on-order	$33.99
0000002	200002	red	small	$61.99	kids	in-stock	$43.99
0000003	200003	black	large	$84.99	kids	in-stock	$50.99
0000004	200004	yellow	medium	$62.99	mens	on-order	$35.99
0000005	200005	green	small	$47.99	womens	in-stock	$42.99

Case 2- (Integrated) Destination

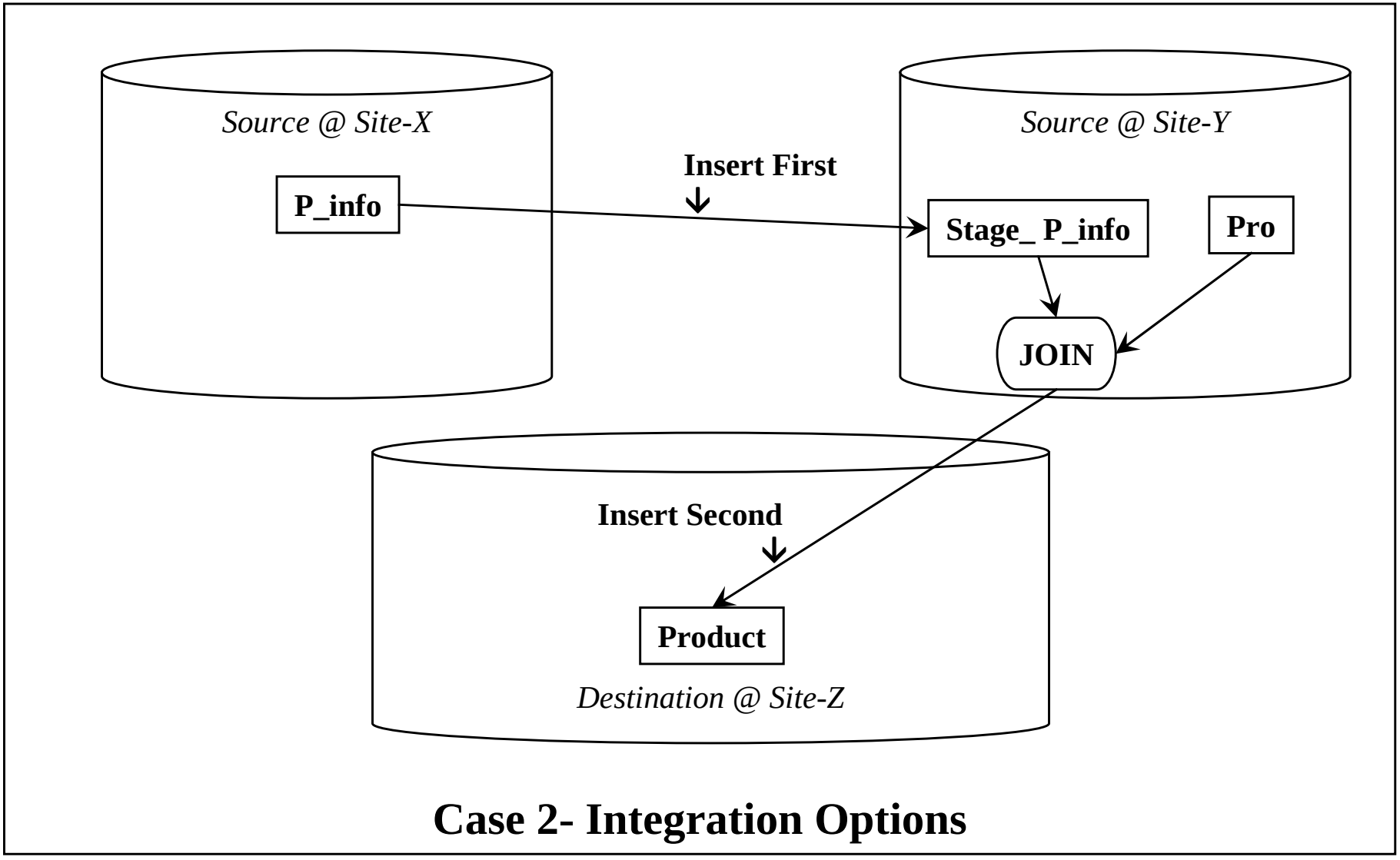
Example: Merging Data Scenario #2

- in this scenario, we again have very low semantic impedance (**SI**)—no **SIM**!
- how do we integrate Scenario #2?
  - what is the main SQL operation?
  - where do we perform this operation?
  - what are the tradeoffs?

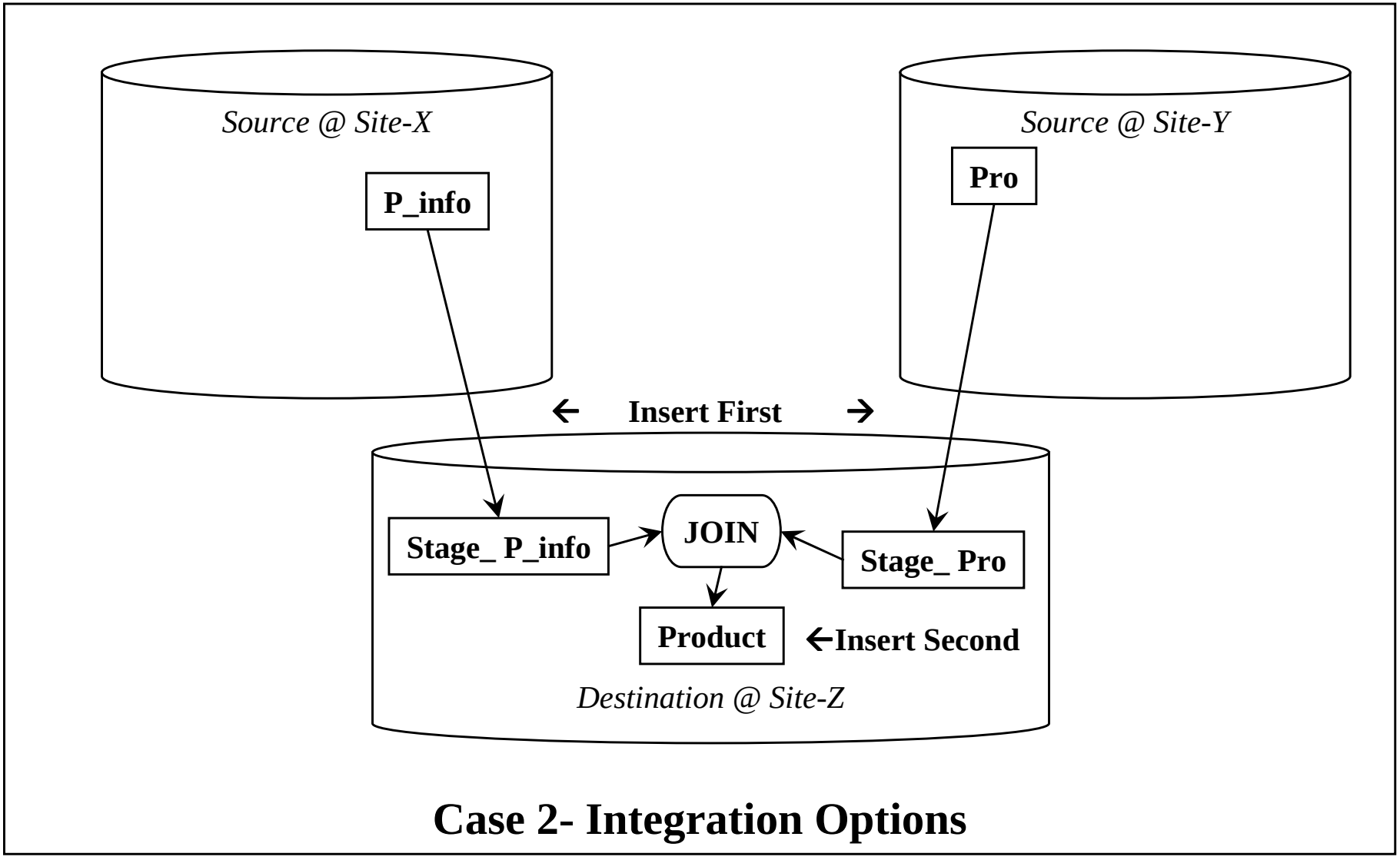
Example: Merging Data Scenario #2-A



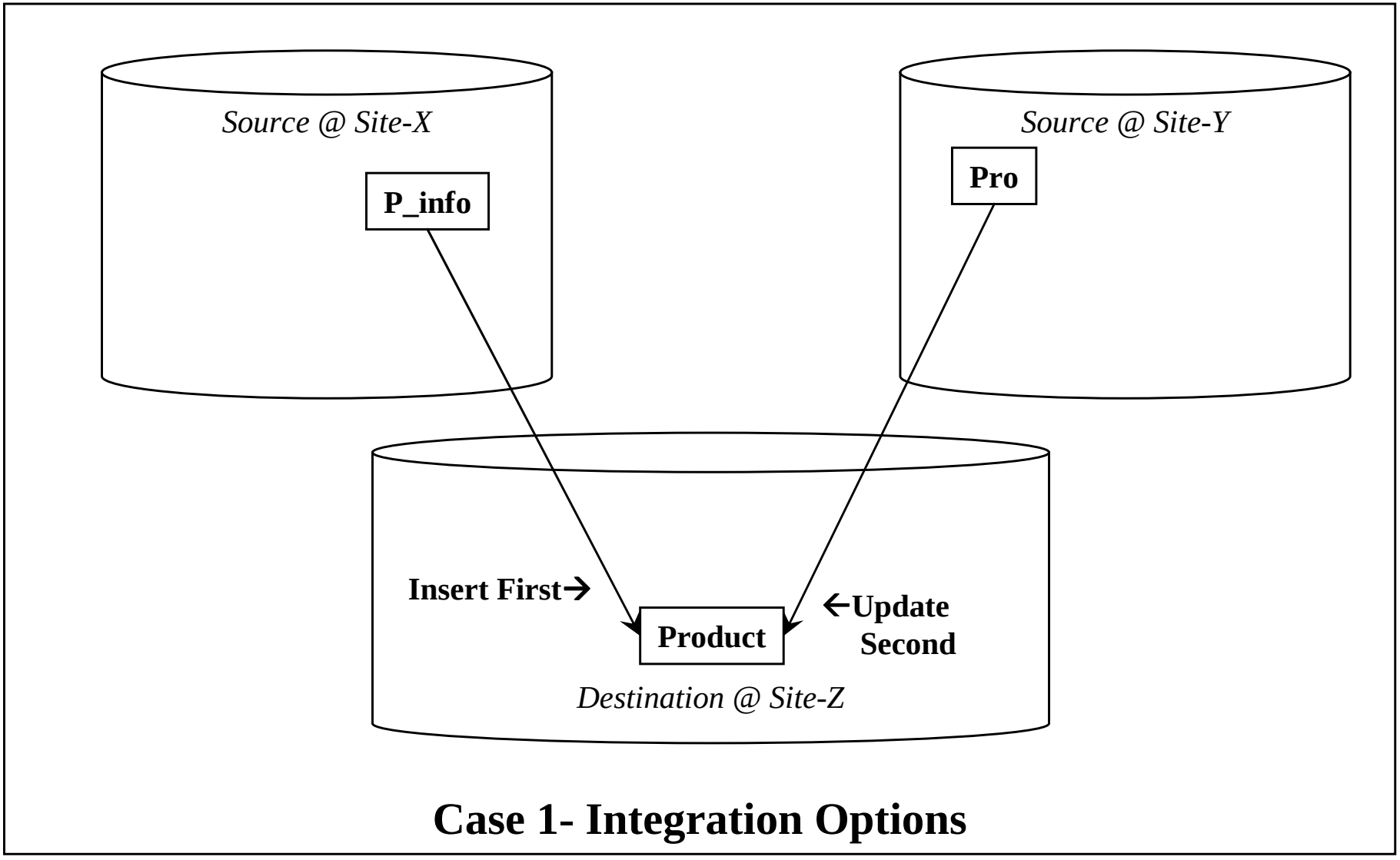
Example: Merging Data Scenario #2-B



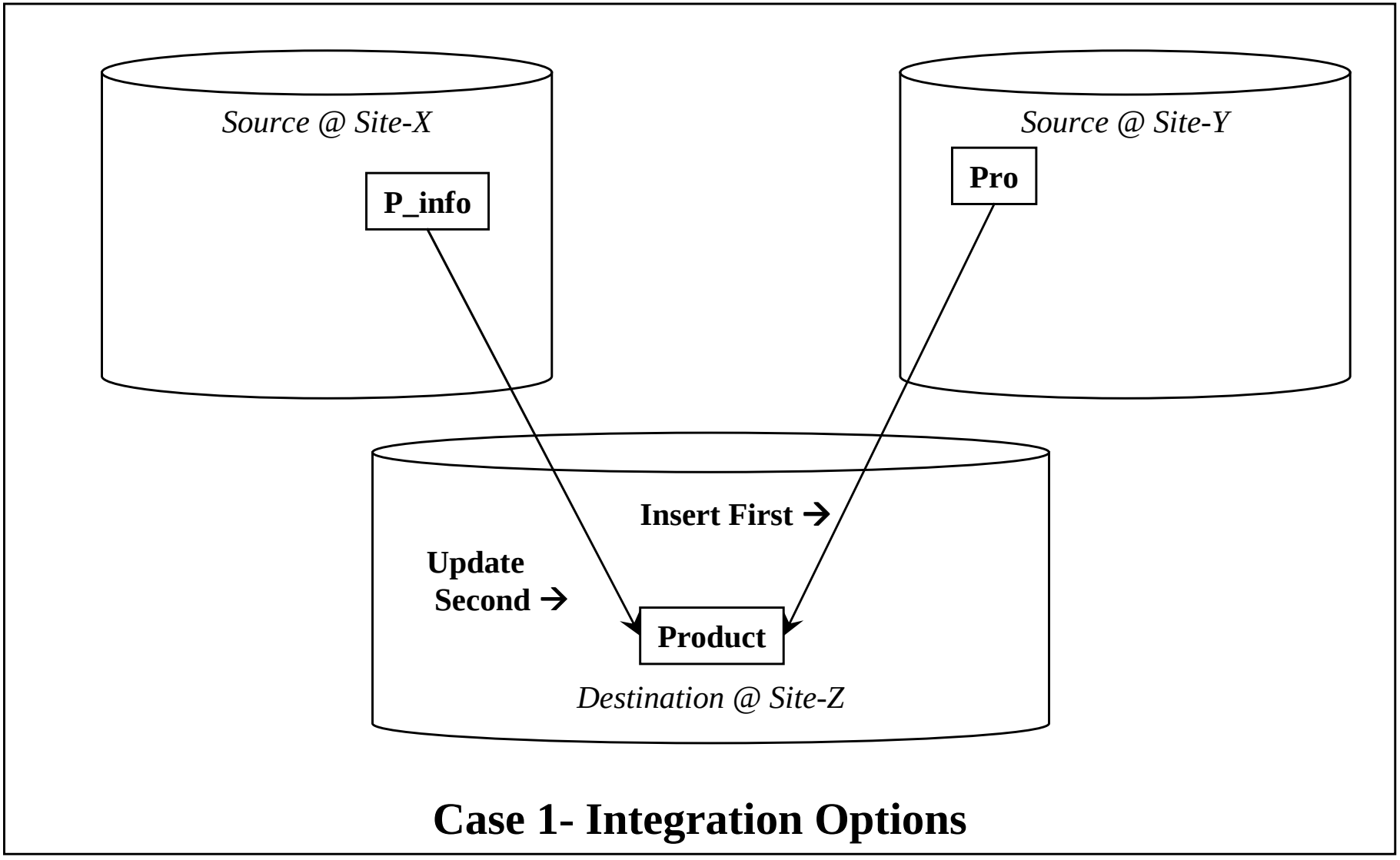
Example: Merging Data Scenario #2-C



Example: Merging Data Scenario #2-D



Example: Merging Data Scenario #2-E



Example: Merging Data Scenario #3

- suppose (across the two source tables) we have:
  - key column types
    - semantically equivalent (meaning **the same** thing)
    - IOW, this is an enterprise-wide identifier!
  - key column values
    - partially overlapping values
    - partially overlapping rows
    - same value for key refers to same instance
    - some missing instances
    - some extra instances

Example: Merging Data Scenario #3

- suppose (across the two source tables) we have:
  - non-key column types
    - some are semantically equivalent (meaning **the same** thing)
    - some are semantically unrelated (not meaning **the same** thing)
  - non-key column values
    - some missing values
    - some extra values
    - some redundant values
    - some conflicting values

Example: Merging Data Scenario #3

In this example suppose the PKs are SEMANTICALLY EQUIVALENT

Retail_Pro

RPID	_clr__	_size_	_price__	_dept_	_cost_
2001	brown	medium	$52.00	kids	$34.99
2002	red	small	$61.99	kids	$43.99
2003	black	large	$84.99	kids	$50.99
2004	yellow	medium	$62.99	mens	$75.99
2005	green	small	$47.99	womens	$42.99

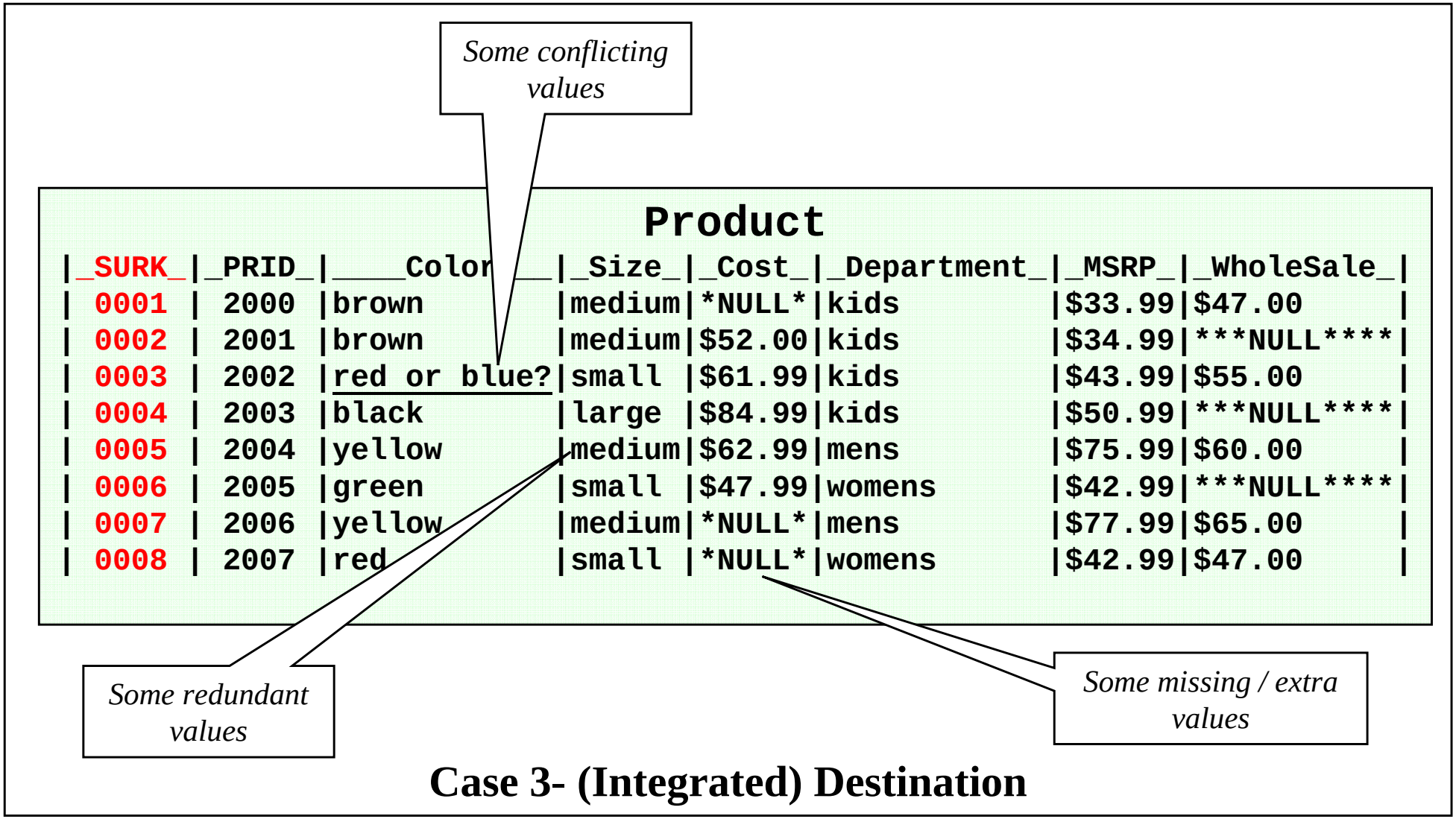
Suppose some columns are SEMANTICALLY EQUIVALENT but some are not

Prodct

PNum	_depart_	_cost_	_color_	_size_	_wprice_
2000	kids	$33.99	brown	medium	$47.00
2002	kids	$43.99	blue	small	$55.00
2004	mens	$75.99	yellow	medium	$60.00
2006	mens	$77.99	yellow	medium	$65.00
2007	womens	$42.99	red	small	$47.00

Case 3- Source

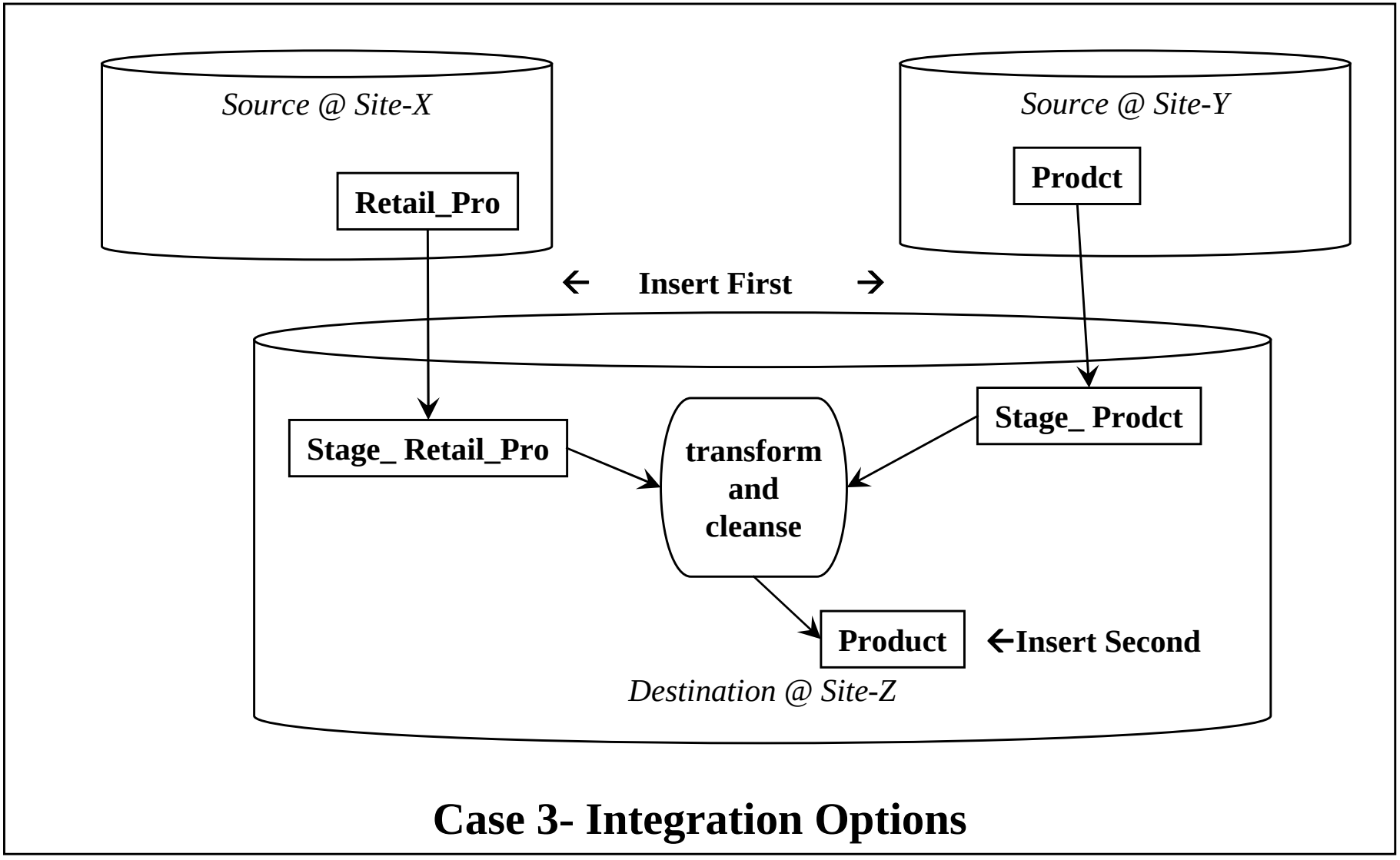
Example: Merging Data Scenario #3



Example: Merging Data Scenario #3

- in this scenario, we have
  high semantic impedance (**SI**)—we have **SIM**!
- how do we integrate Scenario #3?
  - what is the main SQL operation?
  - where do we perform this operation?
  - what are the tradeoffs?

Example: Merging Data Scenario #3



SI / SIM Issues and Causes

- potential factors:
  - lack of data management
  - denial of complexity
  - aging documentation
  - cultural bias and resistance to change
  - expert attrition

Techniques for Addressing **SI / SIM**

- integration approach:
  - be thorough
    - (not just looking at the names and formats)
  - be skeptical
    - don't believe everything you read or everything you are told
  - be systematic & scientific
    - divide and conquer
    - impartial, objective, methodical, precise, and empirical
  - be persistent
    - every data source has a semantic model, sometimes it is difficult to find
    - some (or all) of the model might be implicit

Techniques for Addressing **SI / SIM**

- do look at formats, lengths, and constraints
  - in the data models
  - in the data stores/databases
  - in the applications
- when the semantics are not clear
  - analyze
  - question
  - experiment
  - test and verify
  - document

Techniques for Addressing **SI / SIM**

- create current documentation
  - metadata
  - reverse engineering
  - "translations" and "cookbooks"
  - version control and issue tracking
  - review process
  - communication is important
- use special techniques
  - create mapping definitions
  - create VIEWS!
  - perform cross footing and domain studies

Domain Studies

- simple technique involving analysis of the source columns
- often implemented using simple SQL queries…
  - how many discrete (distinct) values are observed?
  - what are the lowest observed values and their observed frequency?
  - what are the highest observed values and their observed frequency?
  - what are the most commonly observed values and their observed frequency?
  - how many rows have a NULL value observed for a given column?

Domain Studies

- other examples
  - same as previous slide but within a specific combination of values and involving more than one field / column
  - queries that verify referential integrity rules and assumptions
    - min, max cardinalities, etc.
  - queries that perform reasonability tests
    - requires domain knowledge and familiarity with the data
  - frequency distributions and more advanced analysis techniques
    - statistical analysis
    - chi-squared, etc.

Cross Footing

- in the simplest case this is merely counting the number of instances in the source system and in the destination system and verifying that they balance
- if the counts are not identical then you should know (and document) why
- can be much more sophisticated
  - essentially equivalent to performing domain studies on both source and destination systems and comparing the results
  - can also compare to IDD formulas (but not necessarily the actual estimated numbers)
- this can be implemented as part of the ETL or as a separate process

ETL Issues and Techniques Summary

- different data models will model identify and represent concepts differently within the same business area / process
- the enterprise-wide, integrated view of these business areas and processes is (by definition) an attempt to achieve consensus across these different models while retaining as much added value as possible
- for technical and non-technical reasons, there is always some degree of difficulty when attempting to share data among the these OLTP models and between the OLTP and OLAP models

ETL Issues and Techniques Summary

- ETL processes attempt to identify and address semantic impedance and semantic impedance mismatches using various techniques
- there are often several possible techniques to choose from and also several different implementations possible for a given technique
- there is no "magic" solution or "silver bullet" for ETL, but there are several tools and techniques available that can help use plan, design, implement and verify a given ETL process
- two of the most fundamental techniques available are domain studies and cross footing