Chapter 4

ROT: one fact table per process

Drilling across: steps and approaches

ROT: One fact table per process

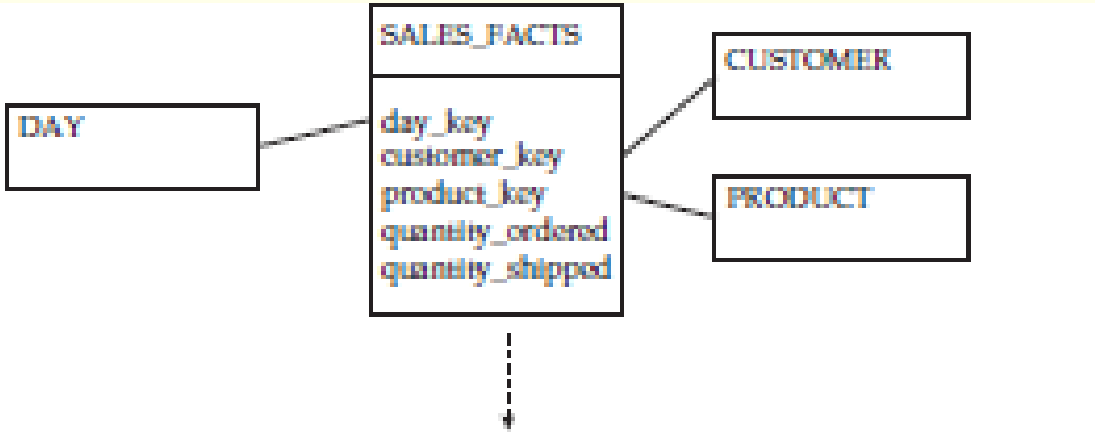
For a given pair of facts, ask:

1. Do these facts occur simultaneously?
2. Are these facts available at the same grain?

If the answer to either of the above is no,
the facts represent different processes.

Subsequently it is possible to combine the individual fact tables in some way to compare the processes, but if done at the outset problems will arise when trying to analyze the processes separately.

Problems arising if ROT not followed



SALES_FACTS

day_key	customer_key	product_key	quantity_ordered	quantity_shipped
123	777	111	100	0
123	777	222	200	0
123	777	333	50	0
456	777	111	0	100
456	777	222	0	75
789	777	222	0	125

These zeros will cause trouble

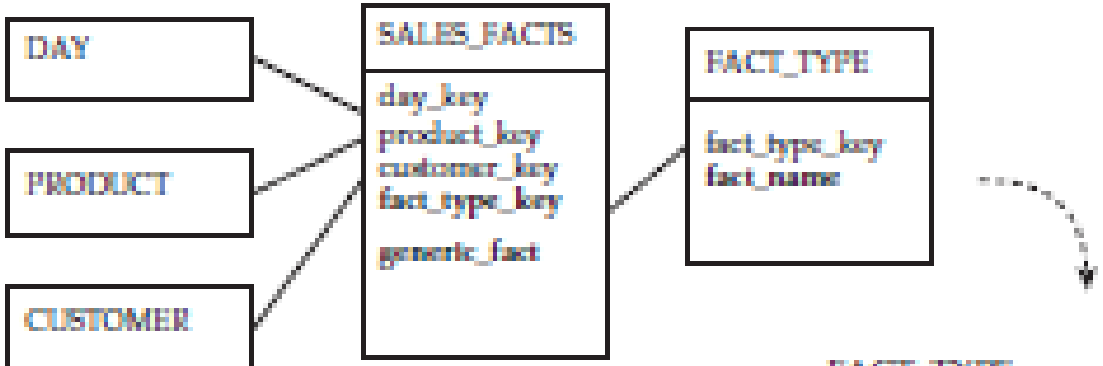
Figure 4-1 Facts with different timing in a single table

Quantity ordered and quantity shipped are available at the same grain, but they do not necessarily appear simultaneously.

Reporting on quantity shipped or ordered for products is slightly complicated - easy to get strange/extraneous results – see figure 4-2.

A fix is to adjust the SQL Where clause

Problems arising if ROT not followed



FACT_TYPE

fact_type_key	fact_name
1	Quantity Ordered
2	Quantity Shipped

query

results

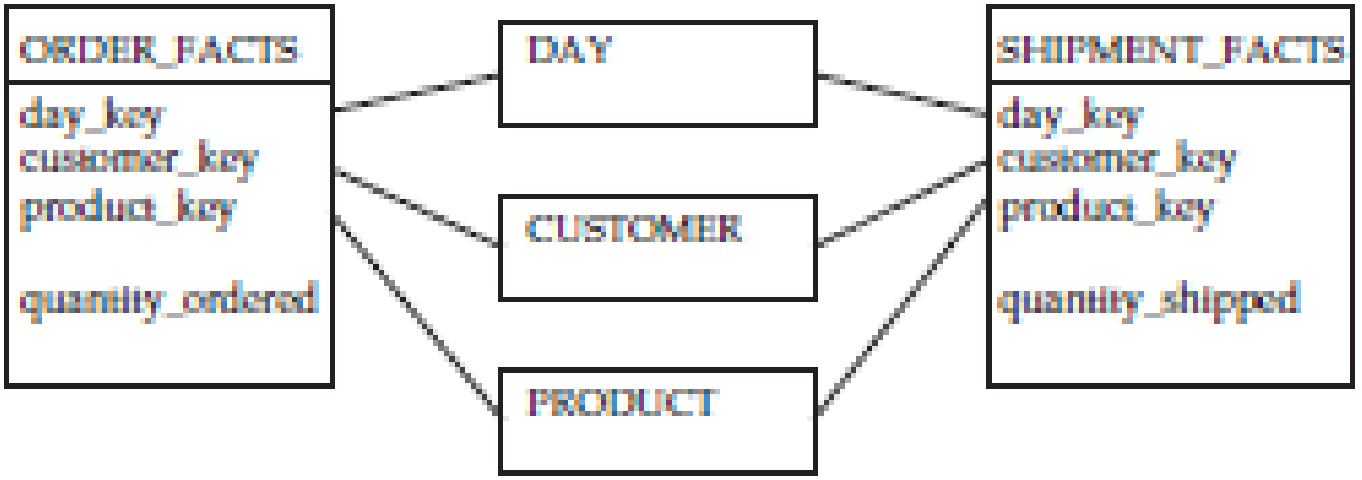
PRODUCT	FACT_NAME	SUM (GENERIC_FACT)
Product 111	Quantity Shipped	100
Product 222	Quantity Shipped	200

Figure 4-3 Modeling a generic fact complicates cross-process analysis

Another fix is to adjust the schema to include a FactType dimension

This is undesirable as well – SQL needs to accommodate the FactType table (easy), but complications arise later when analyzing compare the processes.

Preferred Schema



ORDER_FACTS

day_key	customer_key	product_key	quantity_ordered
123	777	111	100
123	777	222	200
123	777	333	50

SHIPMENT_FACTS

day_key	customer_key	product_key	quantity_shipped
456	777	111	100
456	777	222	75
789	777	222	125

Figure 4-4 Separating the two processes into separate fact tables with shared dimensions

Considering fact tables of different grain

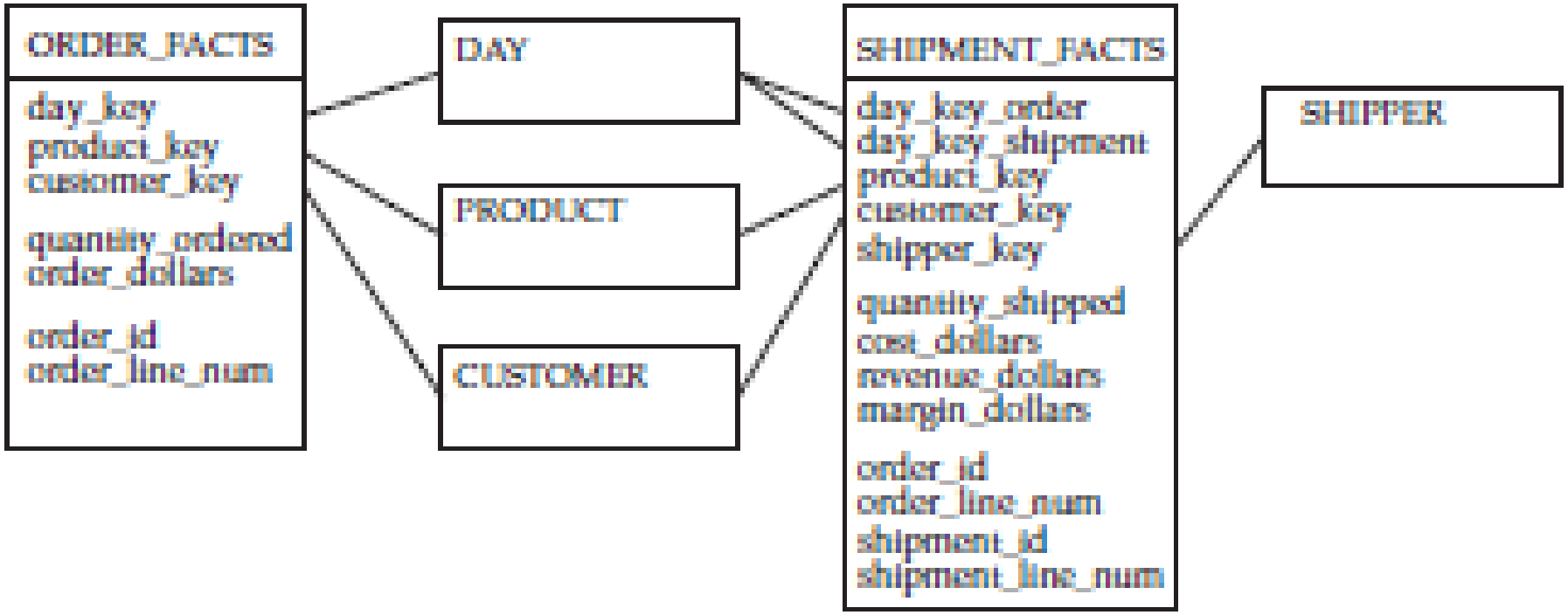


Figure 4-8 Separating the two processes into separate fact tables

Shipment facts are extended to have a Shipper

Each process can be analyzed separately. They may share 3 dimensions – any number of fact tables could reference a dimension.

Drilling Across

Recommended technique when you need to access information from more than one fact table.

Involves 2 steps

1. Issue separate queries for each fact table
2. Combine the results using a full outer join on the common dimensions

Figure 4-10 indicates joining fact tables can be a problem

Drilling Across

ORDER_FACTS

day_key	customer_key	product_key	quantity_key
123	777	111	100
123	777	222	200
123	777	333	50

SHIPMENT_FACTS

day_key	customer_key	product_key	quantity_shipped
456	777	111	100
456	777	222	75
789	777	222	125

```
SELECT
   product.product,
   sum( order_facts.quantity_ordered ),
   sum( shipment_facts.quantity_shipped )
FROM
   product,
   day,
   order_facts,
   shipment_facts
WHERE
   order_facts.product_key = product.product_key AND
   order_facts.day_key = day.day_key AND
   shipment_facts.product_key = product.product_key AND
   shipment_facts.day_key = day.day_key AND
   ...additional qualifications on data...
GROUP BY
   product.product
```

FROM with both fact tables

The order for product 222 is double counted

product	sum(quantity_ordered)	sum(quantity_shipped)
product 111	100	100
product 222	400	200

The order for product 333 does not appear

Figure 4-10 Joining two fact tables leads to trouble

Drill Across: a 2-step technique

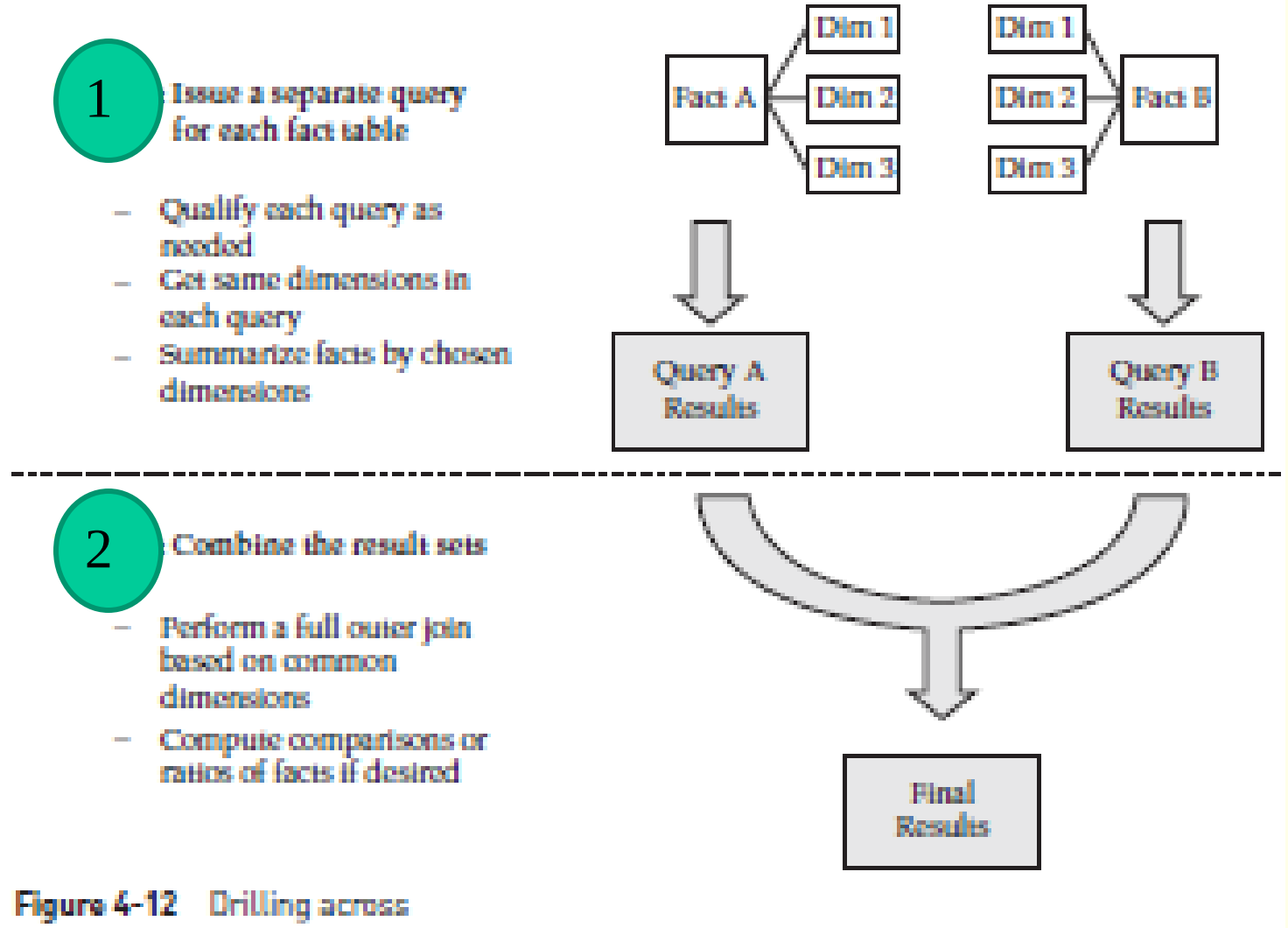


Figure 4-12 Drilling across

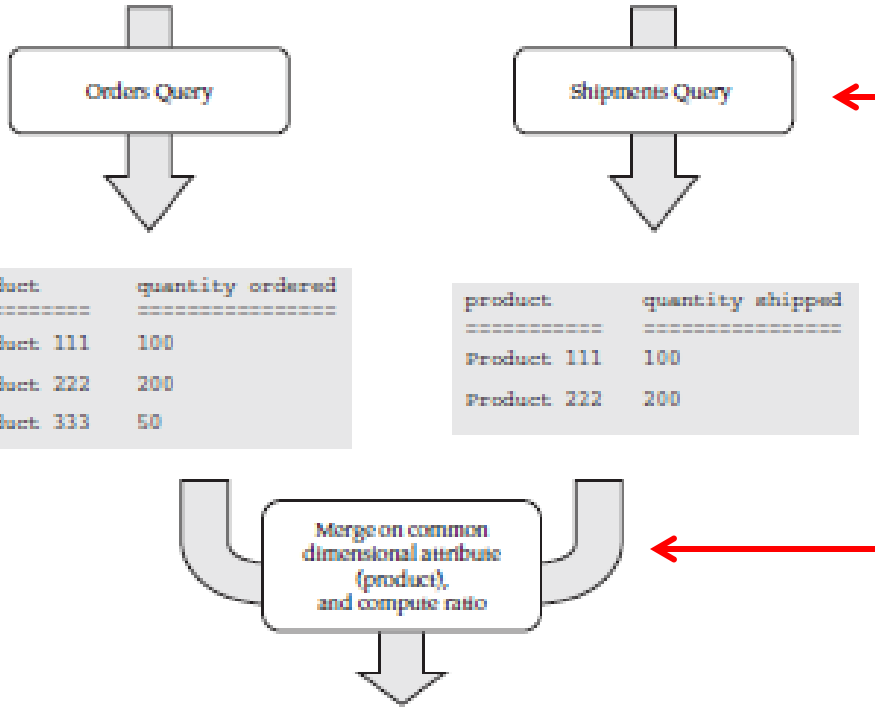
Drill Across: a 2-step technique

ORDER_FACTS

day_key	customer_key	product_key	quantity_ordered
123	777	111	100
123	777	222	200
123	777	333	50

SHIPMENT_FACTS

day_key	customer_key	product_key	quantity_shipped
456	777	111	100
456	777	222	75
789	777	222	125



Query each fact table separately

Combine the results

Figure 4-11 Drilling across orders and shipments

Drilling across – 3 approaches

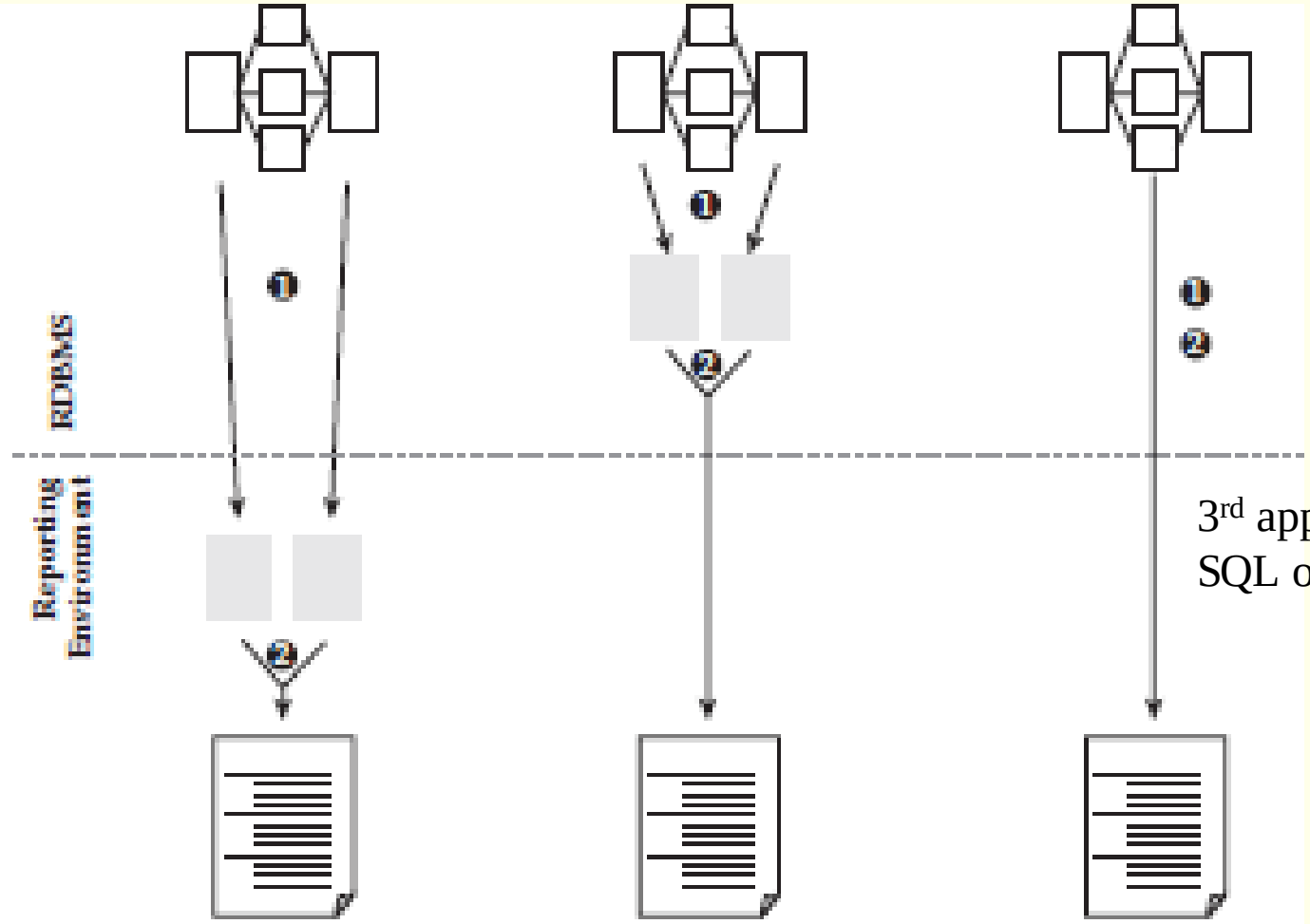


Figure 4-14 Three ways to drill across

These differ with regards to where the selects and merging takes place – in database, in reporting environment

3rd approach: See SQL on pages 80-81