CI2355 – Almacenes de datos y OLAP

UNIVERSIDAD DE
COSTA RICA

Múltiples estrellas

- Procesos de negocios y tablas de hechos

Múltiples estrellas

- Procesos de negocios y tablas de hechos
  - Cada **proceso** debería tener su propia tabla de hechos

Múltiples estrellas

- Procesos de negocios y tablas de hechos
  - Cada **proceso** debería tener su propia tabla de hechos
- ¿Qué es un proceso?

Múltiples estrellas

- Procesos de negocios y tablas de hechos
  - Cada **proceso** debería tener su propia tabla de hechos
- ¿Qué es un proceso?
- Los modelos de procesos se utilizan para describir la actividad del negocio

Múltiples estrellas

- Procesos de negocios y tablas de hechos
  - Cada **proceso** debería tener su propia tabla de hechos
- ¿Qué es un proceso?
- Los modelos de procesos se utilizan para describir la actividad del negocio
- ¿Cómo separar mediciones en múltiples tablas de hechos?

Prueba simple

Dado un par de hechos, haga las **dos** siguientes **preguntas**:

Prueba simple

Dado un par de hechos, haga las dos siguientes preguntas:

i. **¿Ocurren estos hechos simultáneamente?**

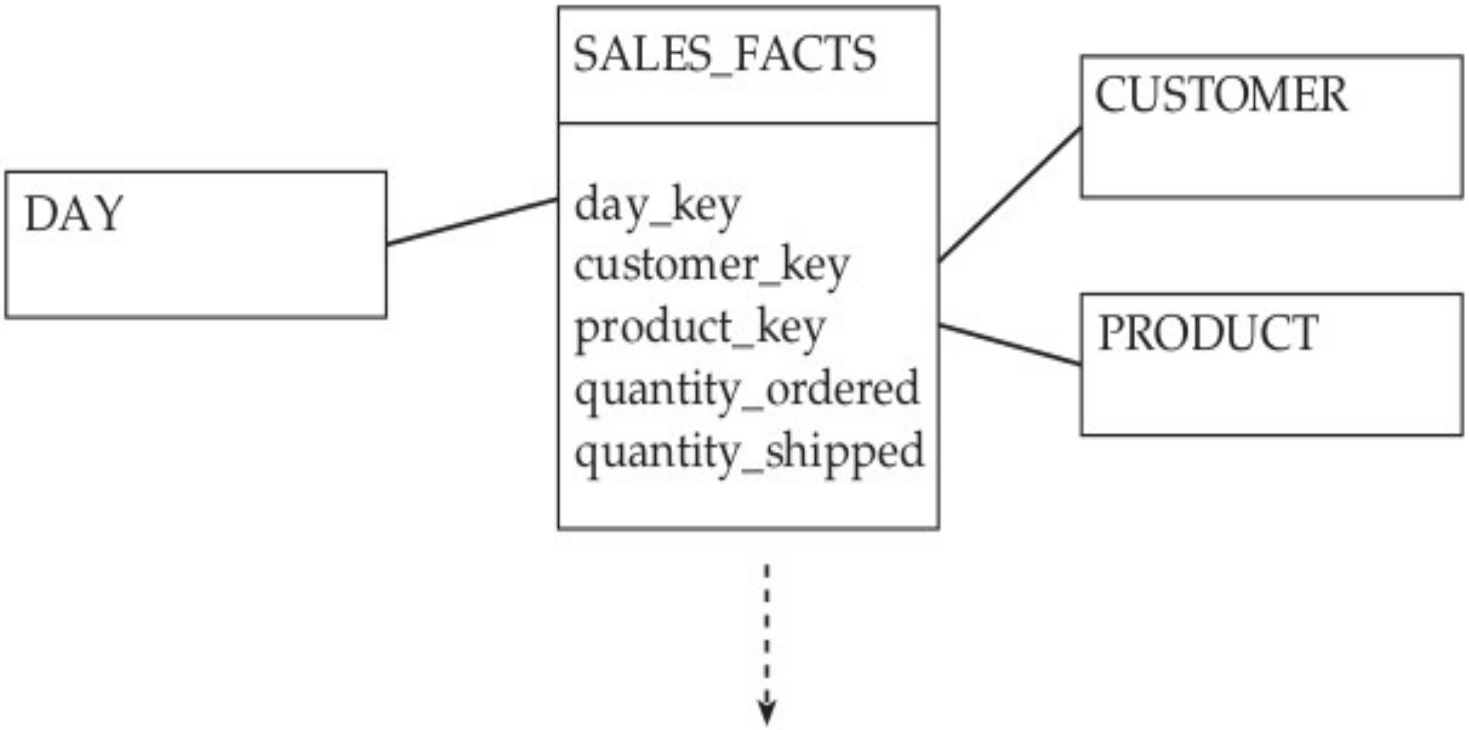
Prueba simple

Dado un par de hechos, haga las dos siguientes preguntas:

i. **¿Ocurren estos hechos simultáneamente?**

ii. **¿Están disponibles estos hechos con un mismo nivel de detalle (granularidad)?**

Hechos en diferentes tiempos



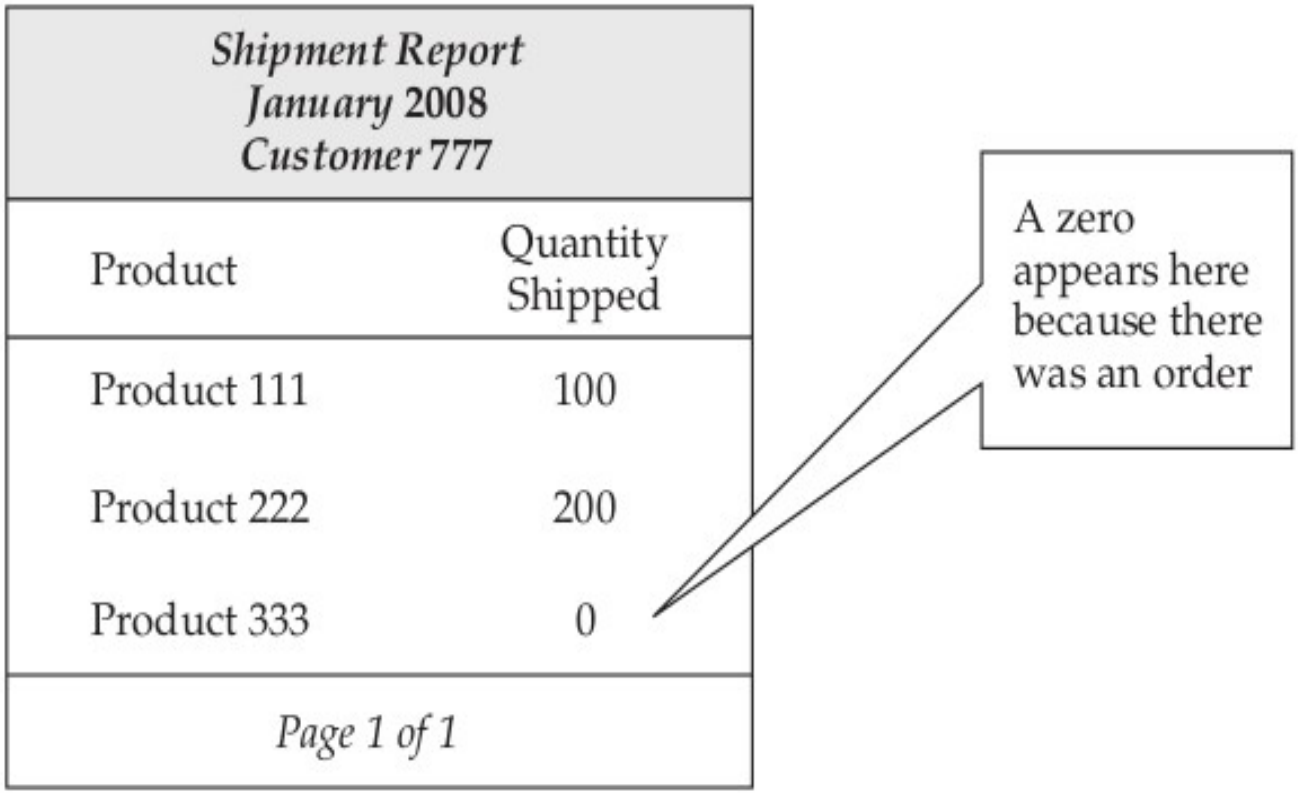
Hechos en diferentes tiempos

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

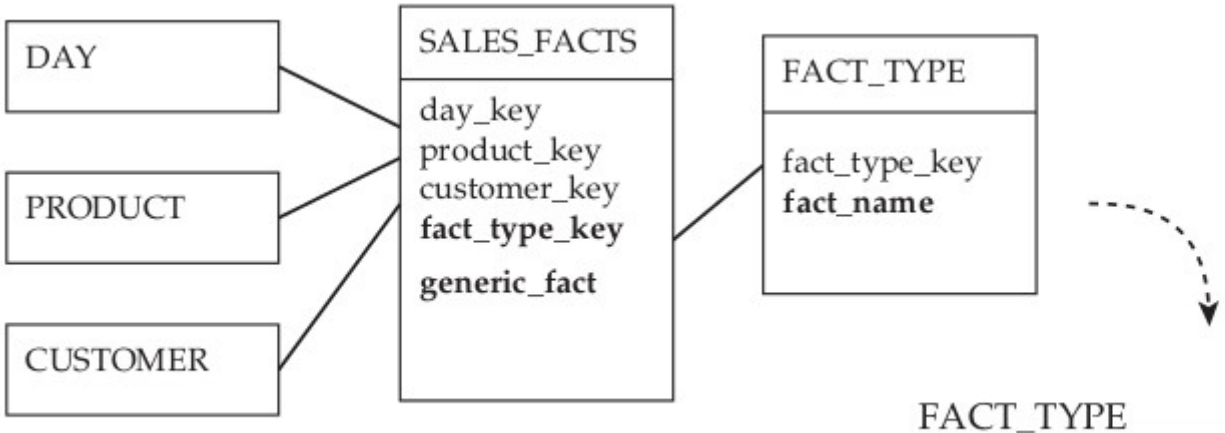
Ejemplo de reporte



¿Solución?

```
select product_key,
sum(quantity_shipped)
from
sales_facts
group by product_key
having sum(quantity_shipped) > 0
```

¿Solución alternativa?



FACT_TYPE

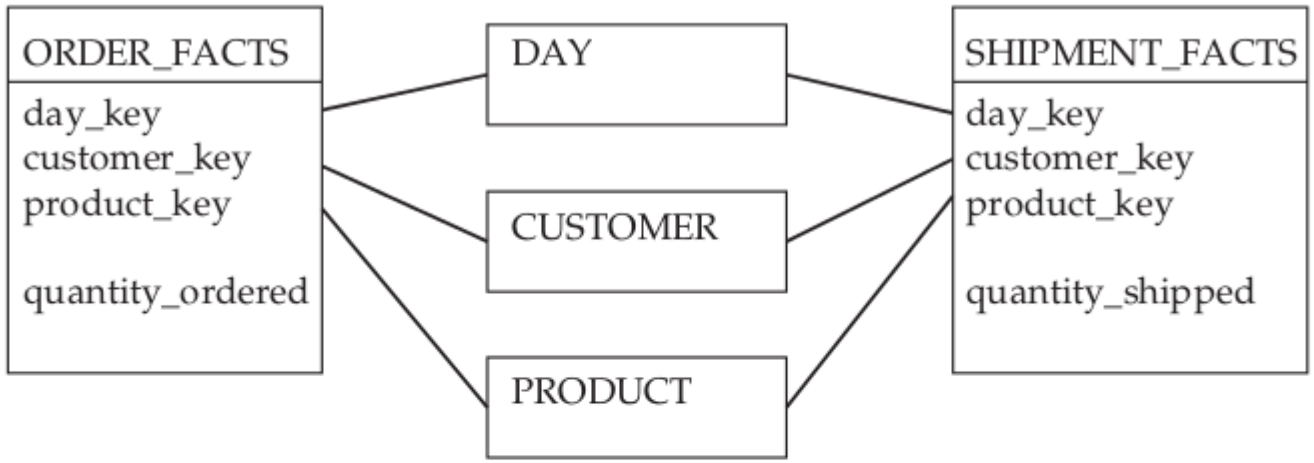
fact_type_key	fact_name
1	Quantity Ordered
2	Quantity Shipped

query

results

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

Separando procesos



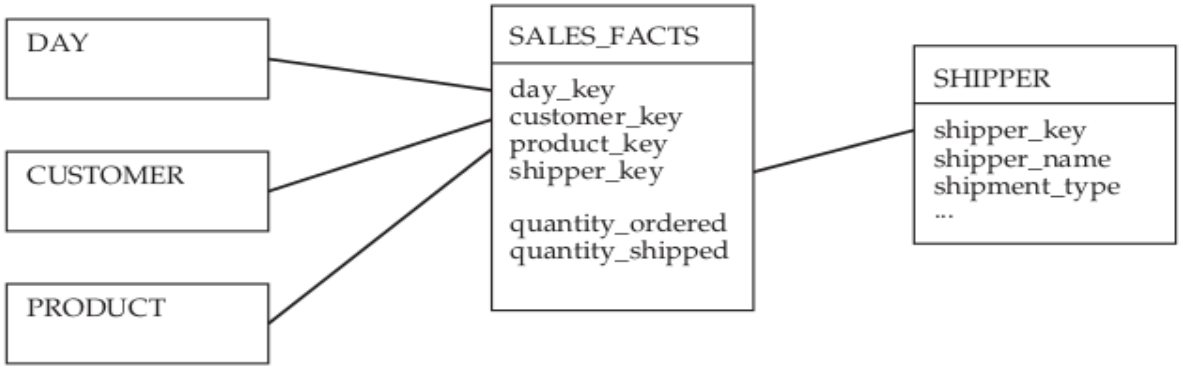
ORDER_FACTS

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

Hechos con diferente grano



SALES_FACTS

day_key	customer_key	product_key	shipper_key	quantity_ordered	quantity_shipped
123	777	111	0	100	0
123	777	222	0	200	0
123	777	333	0	50	0
456	777	111	9999	0	100
456	777	222	9999	0	75
789	777	222	8888	0	120

Same old problem:
These zeros will cause trouble

SHIPPER

shipper_key	shipper_name	shipment_type
0	[Not a shipper]	N/A
777	US Ship Co.	Overnight
8888	US Ship Co.	Standard
9999	Zig Zag	Overnight

Special row for use with orders

Ejemplo de reporte

Shipment Report by Shipper
January 2008
Customer 777

Product	Shipper Name	Quantity Shipped
Product 111	[not a shipper]	0
Product 111	Zig-Zag Inc.	100
Product 222	[not a shipper]	0
Product 222	Zig-Zag Inc.	75
Product 222	US Ship Co.	125
Product 333	[not a shipper]	0

Page 1 of 1

Utilizando NULL

SALES_FACTS

day_key	customer_ key	product_ key	shipper_ key	quantity_ ordered	quantity_ shipped
123	777	111	(NULL)	100	0
123	777	222	(NULL)	200	0
123	777	333	(NULL)	50	0
456	777	111	9999	0	100
456	777	222	9999	0	75
789	777	222	8888	0	125

SHIPPER

shipper_ key	shipper_ name	shipment_ type
7777	US Ship Co.	Overnight
8888	US Ship Co.	Standard
9999	Zig Zag	Overnight

❶

Shipment Rerport by Shipper
January 2008
Customer 777

Product	Shipper Name	Quantity Shipped
Product 111	Zig-Zag Inc.	100
Product 222	Zig-Zag Inc.	75
Product 222	US Ship Co.	125

NULL keys help when the Shipper table participates

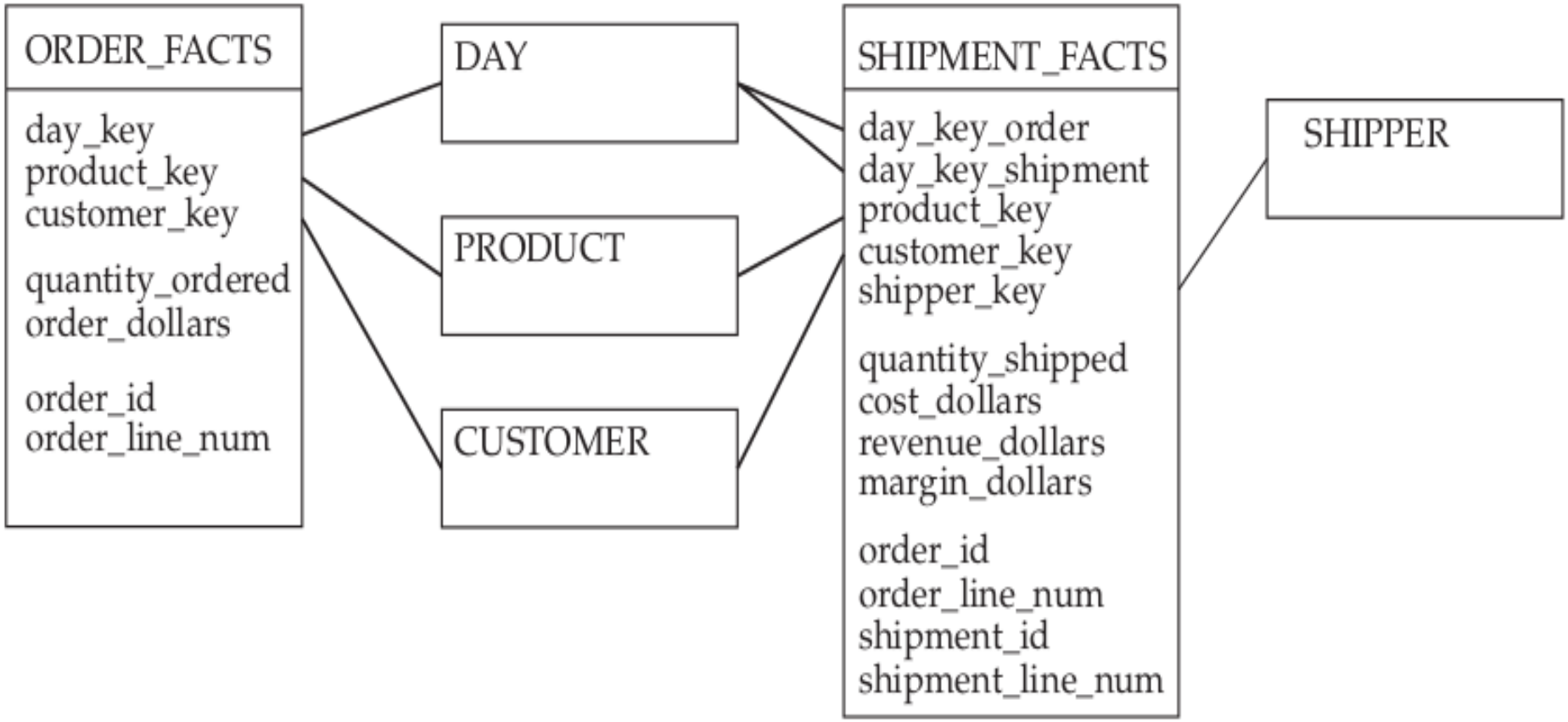
❷

Shipment Report
January 2008
Customer 777

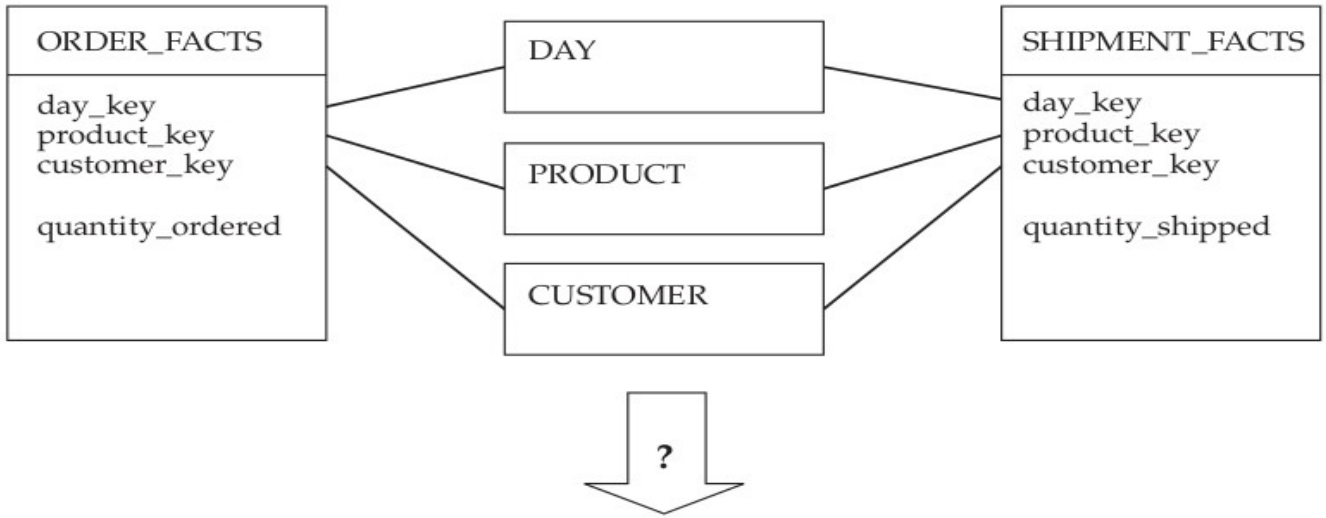
Product	Quantity Shipped
Product 111	100
Product 222	200
Product 333	0

When SHIPPER is not in the query, 0-valued facts return

Modelando con 2 tablas de hechos



Analizando hechos de más de una tabla



Orders vs. Shipments *January 2008*		
Product	*Quantity Ordered*	*Quantity Shipped*
Product 111	100	100
Product 222	200	200
Product 333	50	
Page 1 of 1		

Problemas con los JOINs

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 date...
GROUP BY
   product.product
```

Problemas con los JOINs

```
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 date...
GROUP BY
   product.product
```

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

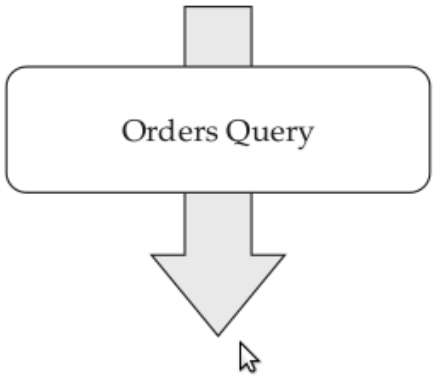
Drilling Across

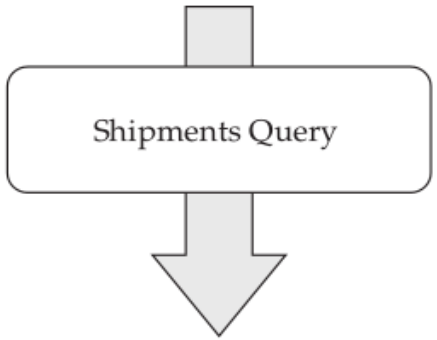
ORDER_FACTS

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





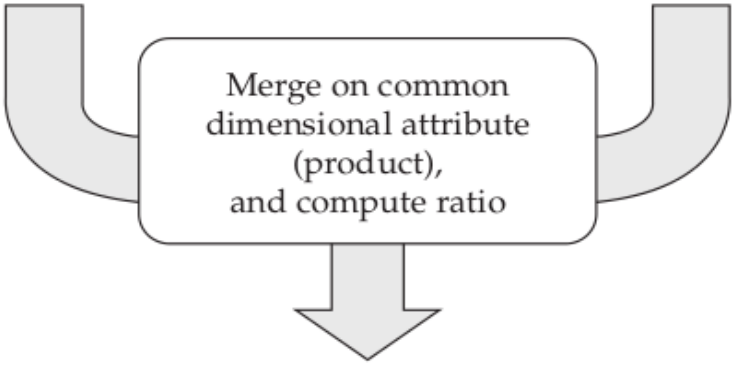
product	quantity ordered
Product 111	100
Product 222	200
Product 333	50

product	quantity shipped
Product 111	100
Product 222	200

Drilling Across

product	quantity ordered
Product 111	100
Product 222	200
Product 333	50

product	quantity shipped
Product 111	100
Product 222	200

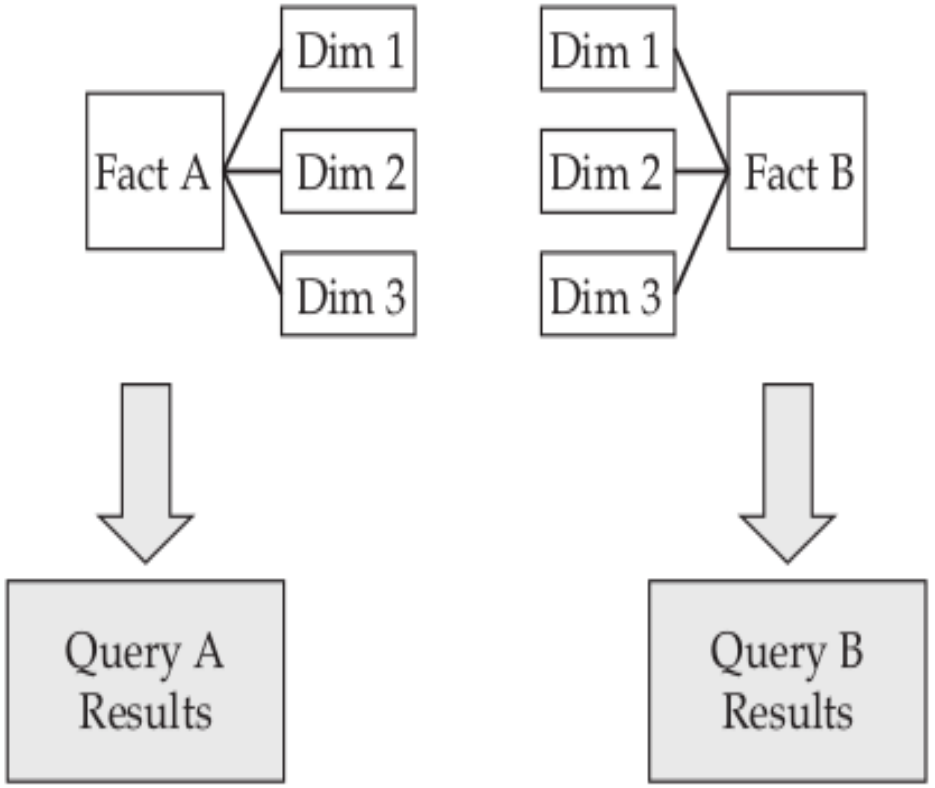


product	quantity ordered	quantity shipped	ratio
Product 111	100	100	100%
Product 222	200	200	100%
Product 333	50		0%

Fase 1

Phase 1: Issue a separate query for each fact table

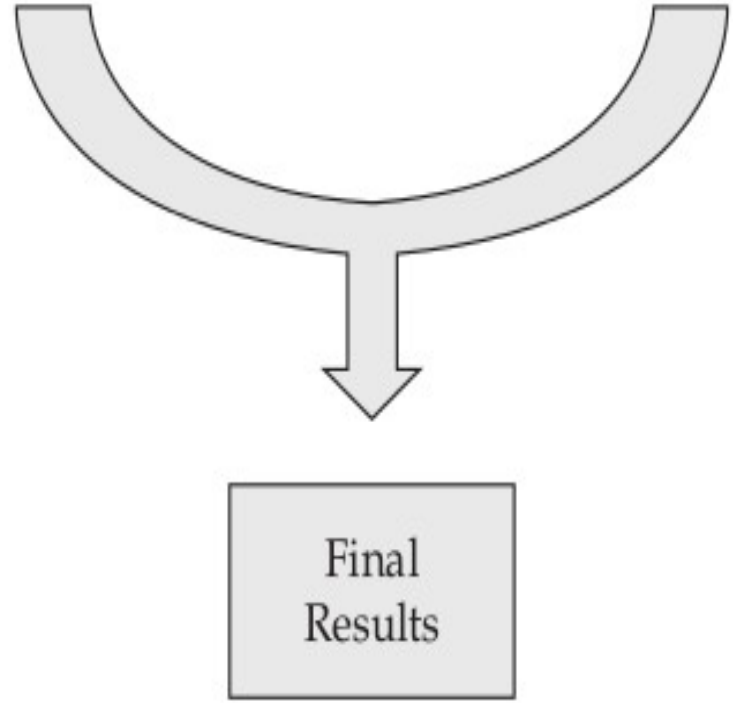
- Qualify each query as needed
- Get same dimensions in each query
- Summarize facts by chosen dimensions



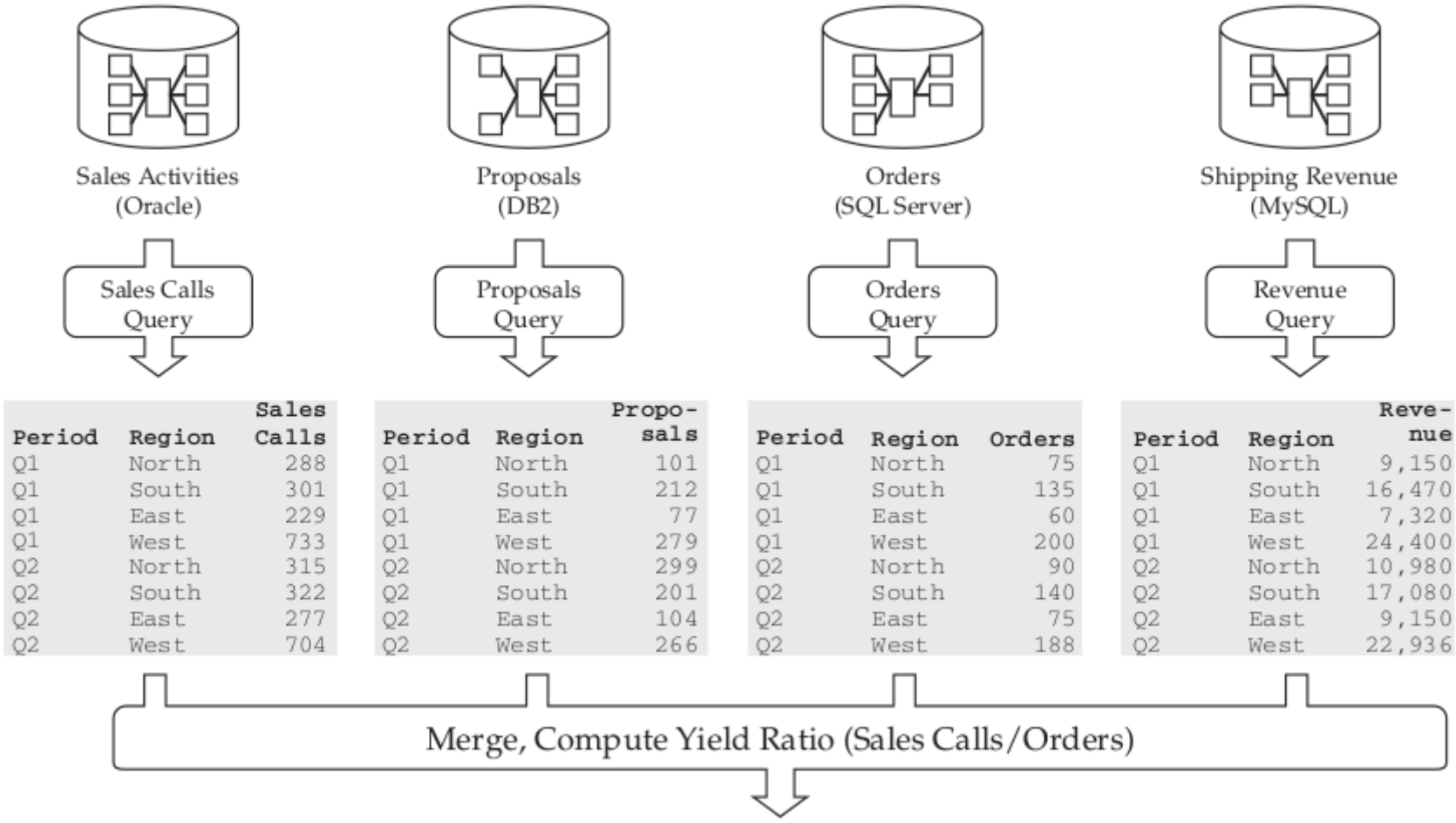
Fase 2

Phase 2: Combine the result sets

- Perform a full outer join based on common dimensions
- Compute comparisons or ratios of facts if desired



Ejemplo con 4 tablas



Period	Region	Sales Calls
Q1	North	288
Q1	South	301
Q1	East	229
Q1	West	733
Q2	North	315
Q2	South	322
Q2	East	277
Q2	West	704

Period	Region	Propo-sals
Q1	North	101
Q1	South	212
Q1	East	77
Q1	West	279
Q2	North	299
Q2	South	201
Q2	East	104
Q2	West	266

Period	Region	Orders
Q1	North	75
Q1	South	135
Q1	East	60
Q1	West	200
Q2	North	90
Q2	South	140
Q2	East	75
Q2	West	188

Period	Region	Reve-nue
Q1	North	9,150
Q1	South	16,470
Q1	East	7,320
Q1	West	24,400
Q2	North	10,980
Q2	South	17,080
Q2	East	9,150
Q2	West	22,936

Ejemplo con 4 tablas

Period	Region	Sales Calls
Q1	North	288
Q1	South	301
Q1	East	229
Q1	West	733
Q2	North	315
Q2	South	322
Q2	East	277
Q2	West	704

Period	Region	Propo-sals
Q1	North	101
Q1	South	212
Q1	East	77
Q1	West	279
Q2	North	299
Q2	South	201
Q2	East	104
Q2	West	266

Period	Region	Orders
Q1	North	75
Q1	South	135
Q1	East	60
Q1	West	200
Q2	North	90
Q2	South	140
Q2	East	75
Q2	West	188

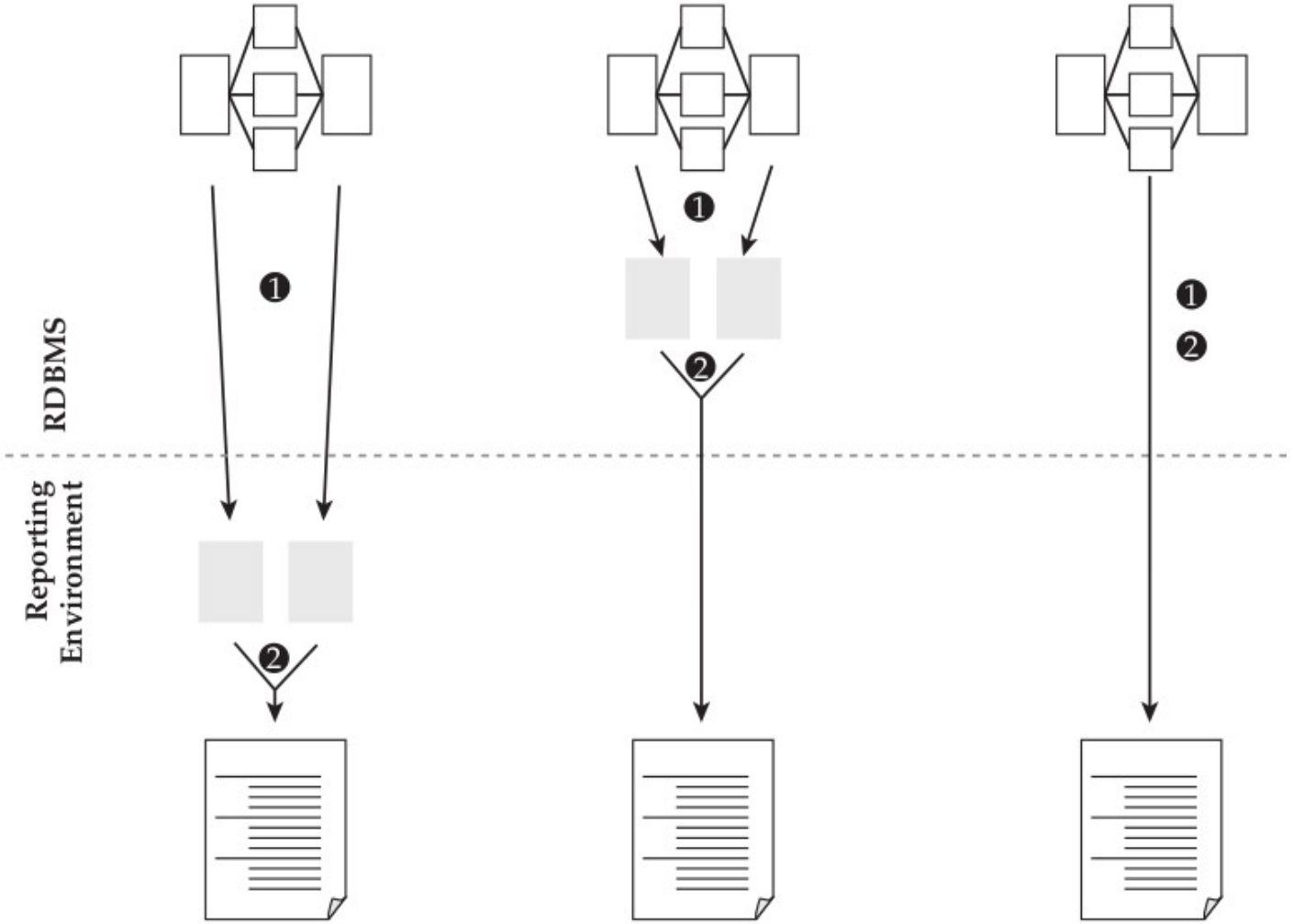
Period	Region	Reve-nue
Q1	North	9,150
Q1	South	16,470
Q1	East	7,320
Q1	West	24,400
Q2	North	10,980
Q2	South	17,080
Q2	East	9,150
Q2	West	22,936

Merge, Compute Yield Ratio (Sales Calls/Orders)

Sales Report
2008

Period	*Region*	*Sales Calls*	*Proposals*	*Orders*	*Revenue*	*Yield*
Q1	North	288	101	75	9,150	26%
	South	301	212	135	16,470	45%
	East	229	77	60	7,320	26%
	West	733	279	200	24,400	27%
Q2	North	315	299	90	10,980	29%
	South	322	201	140	17,080	43%
	East	277	104	75	9,150	27%
	West	704	266	188	22,936	27%

Implementaciones



Sólo con SQL

```
SELECT
COALESCE ( orders_query.product, shipments_query.product),
orders_query.quantity_ordered,
shipments_query.quantity_shipped,
orders_query.quantity_ordered / shipments_query.quantity_shipped as "Ratio"
FROM
(
SELECT
product.product,
SUM (order_facts.quantity_ordered) as quantity_ordered
FROM
day,
product,
order_facts
WHERE
... joins and constraints on date ...
```

Sólo con SQL (cont.)

```
) orders_query
FULL OUTER JOIN
(
SELECT
product.product,
SUM (shipment_facts.quantity_shipped) as quantity_shipped
FROM
day,
product,
shipment_facts
WHERE
... joins and constraints on date ...
) shipments_query
ON
orders_query.product = shipments_query.product
```

Referencias

Adamson, C.
Star Schema - The Complete Reference.
McGraw-Hill, 2010.