

Chapter 12, 14

- **Factless fact tables (pages 291-297)**
- **Pivoted fact tables (pages 335-337)**

Factless fact table

- Measure the occurrence of an event or activity
- No metrics
- Table could be all key
- (pages 291-297)

Factless fact table

Suppose that a business wants to measure the amount of contact that takes place with customers

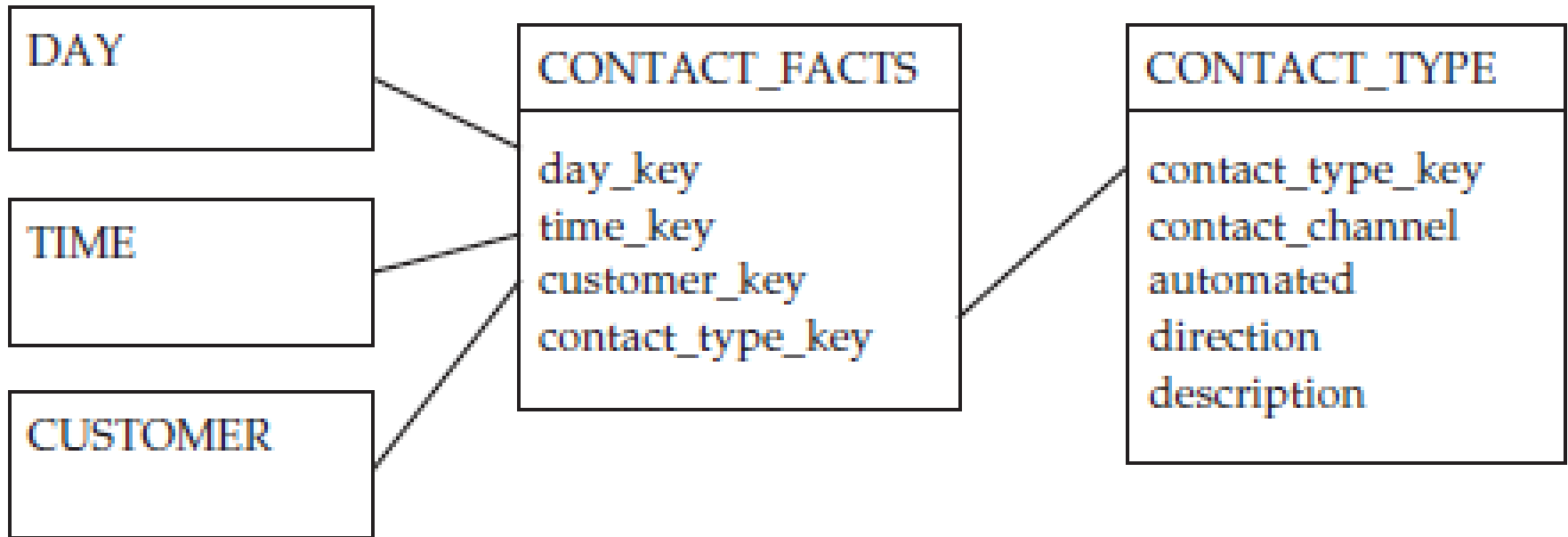
assume there are two channels for customer contact: e-mail and telephone.

A given contact may be initiated by the customer (inbound) or by the business (outbound).

Some outbound contacts may be automated, such as an e-mail newsletter, while others are not.

Factless fact table

The fact table `contact_facts` below measures customer contact activities. The grain is defined as one row for each contact event with a customer. Each row includes foreign keys that refer to dimension tables, and nothing else.



Factless fact table

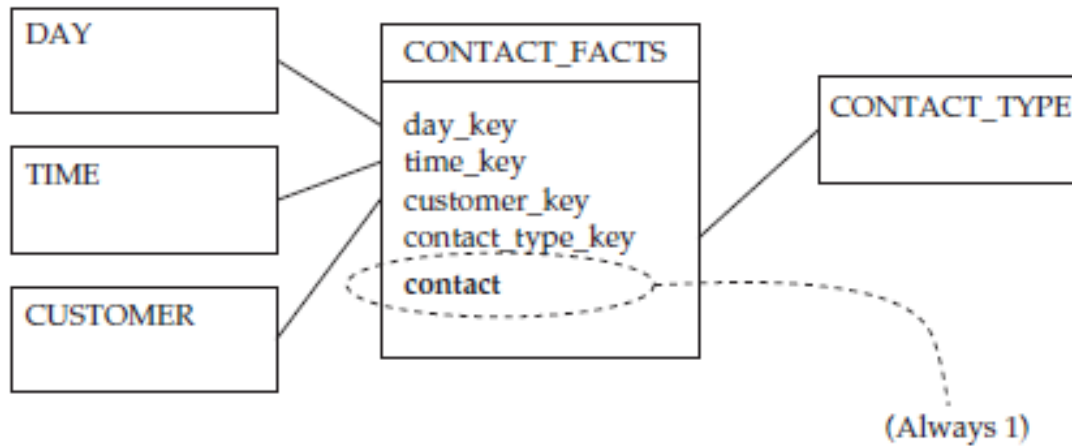
CONTACT_TYPE

contact_type_key	contact_channel	automated	direction	description
100	E-Mail	Automated	Outbound	Weekly Coupons
101	E-Mail	Automated	Outbound	eNews Mailing
102	E-Mail	Not Automated	Inbound	Tech Support
103	E-Mail	Not Automated	Outbound	Tech Support
104	Telephone	Not Automated	Outbound	Promotional Offer
105	Telephone	Not Automated	Inbound	Tech Support
106	Telephone	Not Automated	Outbound	Tech Support

Using the table involves statements like
Select count(*) from Contact_Type

Factless fact table

Add a fact ...



Minor change.... But simplifies SQL in some cases

Factless fact table

Add a fact ...

(Always 1)

CONTACT_FACTS

day_key	time_key	customer_key	contact_type_key	contact
27221	121	4622	101	1
27221	121	3722	101	1
27221	121	2828	102	1
27221	123	1911	102	1
27221	123	3922	103	1
27221	123	3811	101	1
27221	124	8201	101	1

Now, there's a field (a metric) that can be treated like any other

Using the table involves statements like

Select sum(contact) from Contact_Type

Factless fact table

From the text, page 297

Adding a fact that always contains the value 1 makes SQL readable, allowing queries to use the SQL function SUM() rather than COUNT().

However this may impact performance:

... the important part: since all that is needed is a count, it is not actually necessary for the RDBMS to read these rows; it simply counts the number of “hits” in the index.

... the database is asked to SUM() a fact table column called contact, it has no way of knowing that this column always contains the value 1.

...must read these rows from disk to get the contact values, and then add up the values.

...good news is you can have it both ways. Adding the additional fact does not stop anyone from writing a query that counts rows. COUNT() and SUM() are both available options

Pivoted fact table

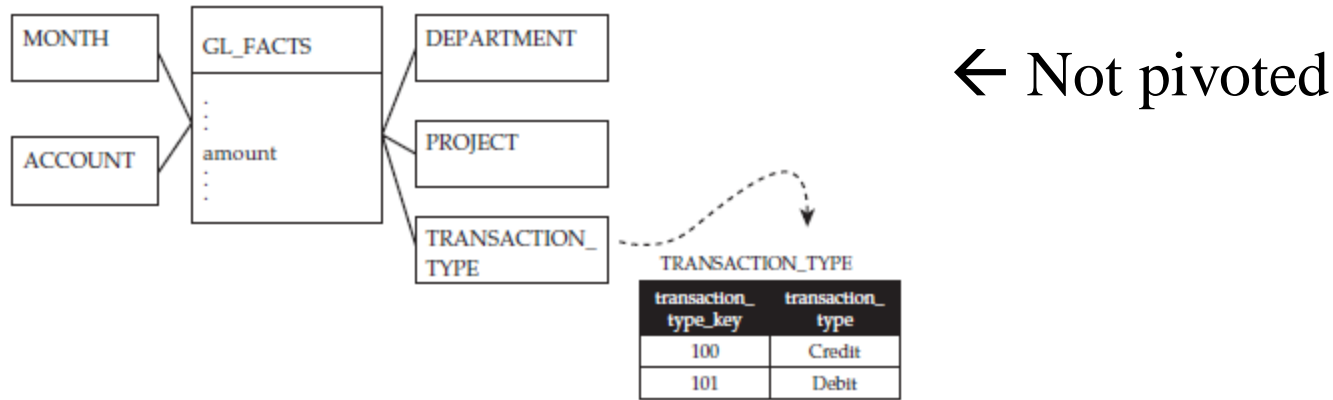
Pages 335-337

Data is transposed from a row-wise to a column-wise orientation

Useful to simplify data access

(recall Access cross-tab queries?)

Pivoted fact table



If you want to generate totals for credits and debits by account:

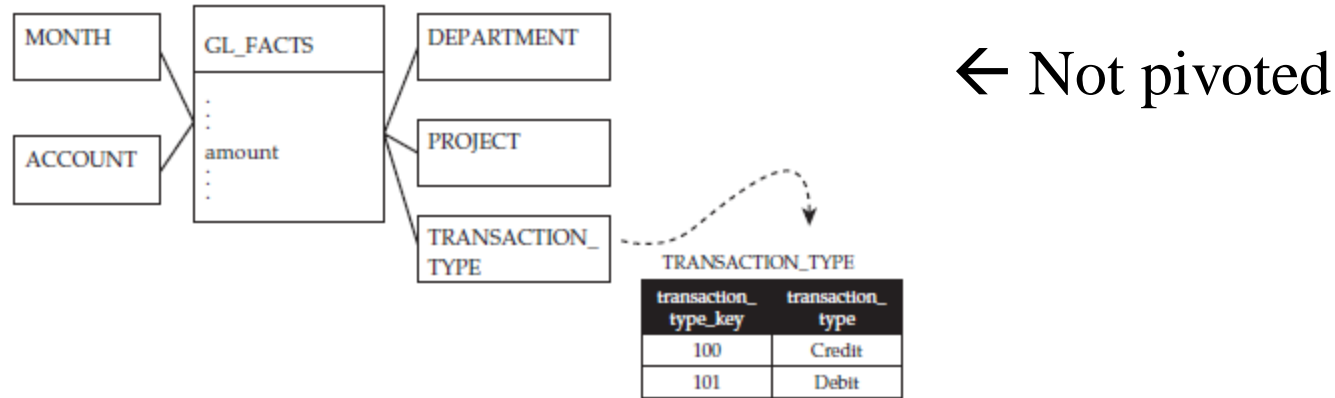
Select account, transaction_type, sum(amount)

From Gl_Facts inner join Transaction_Type on (...)

Group by account, transaction_type

But the amounts are on different lines

Pivoted fact table



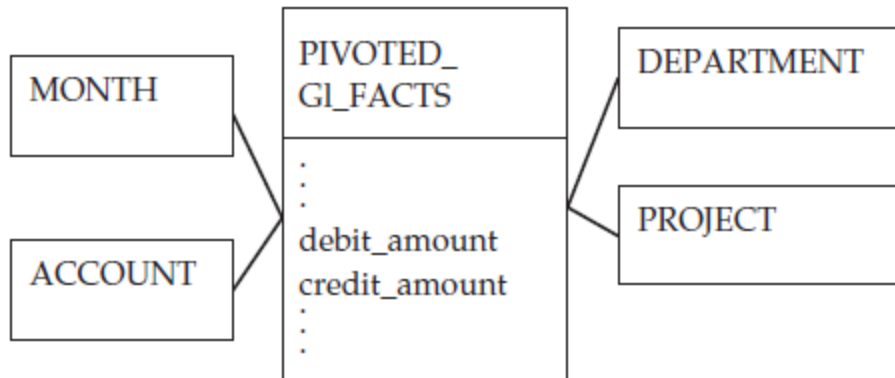
Select account, transaction_type, sum(amount)
 From Gl_Facts inner join Transaction_Type on (...)
 Group by account, transaction_type

ACCOUNT	TRANSACTION_TYPE	SUM(AMOUNT)
01-2211	Credits	20,301.00
01-2211	Debits	- 17,691.30
07-4499	Credits	1,221.23
07-4499	Debits	- 2,220.01
.		
.		

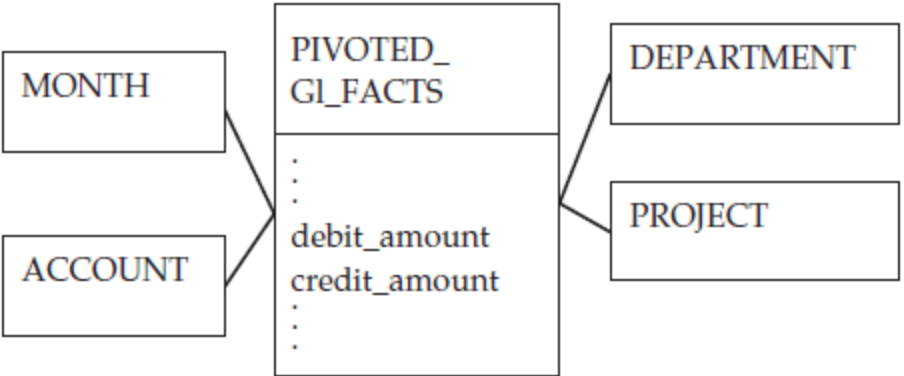
Pivoted fact table

Change the design ... Pivot the fact table

Removed Transaction type/ modified fact table for 2 metrics



Pivoted fact table



Now, to generate totals for credits and debits by account:

Select account, sum(debit_amount), sum(credit_amount)
From Pivoted_Gl_Facts
Group by account

And both totals are on one line

ACCOUNT	DEBITS	CREDITS
01-2211	17,691.30	20,301.00
07-4499	2,220.01	1,221.23