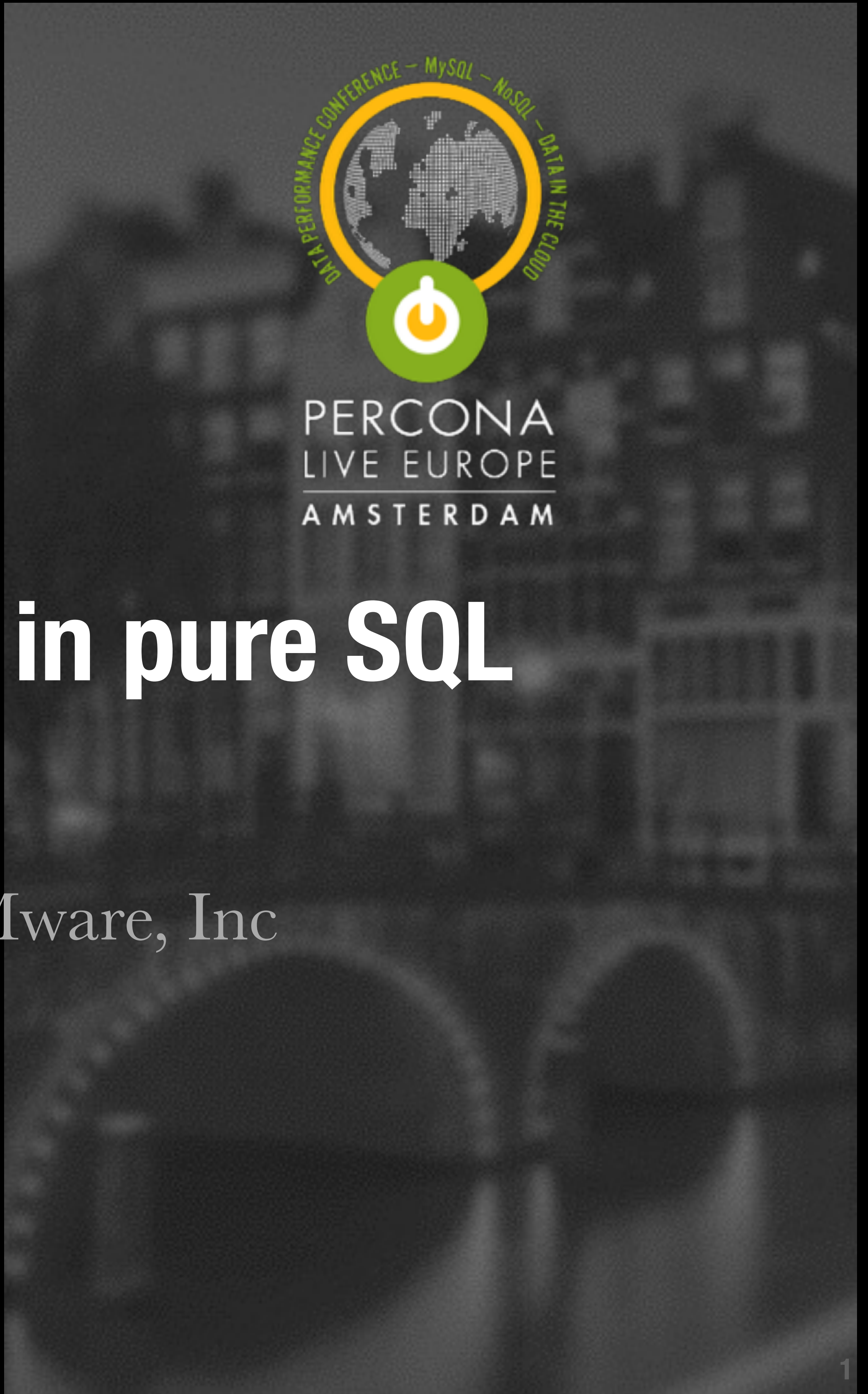


DATA PERFORMANCE CONFERENCE – MySQL – NoSQL – DATA IN THE CLOUD

PERCONA
LIVE EUROPE
AMSTERDAM

Pivot tables: Analytics in pure SQL

Giuseppe Maxia

Quality Assurance Architect at VMware, Inc

About me

Who’s this guy?

- Giuseppe Maxia, a.k.a. "The Data Charmer"
  - QA Architect at VMware
  - 25+ years development and DB experience
  - Long timer MySQL community member.
  - Oracle ACE Director
  - Blog: http://datacharmer.blogspot.com
  - Twitter: @datacharmer
- Curious fact: I learned SQL before learning English

Agenda: pivot tables, ak.a. cross tabs

- The story so far
- What are pivot tables
- Why pivot tables in SQL
- Components of mono-dimensional pivot tables
- Advanced: multi-dimension pivot tables
- More advanced: multi-operation, multi-dimension pivot tables.

The story so far

This is an old topic, which seems to be still popular

- 2001: an article - MySQL Wizardry
- 2003: a Perl module - DBIx::SQLCrosstab
- 2003: an article on advanced cross tabs
- 2005: hackish implementation in SQL
- 2009: implementations in Lua

What's a pivot table

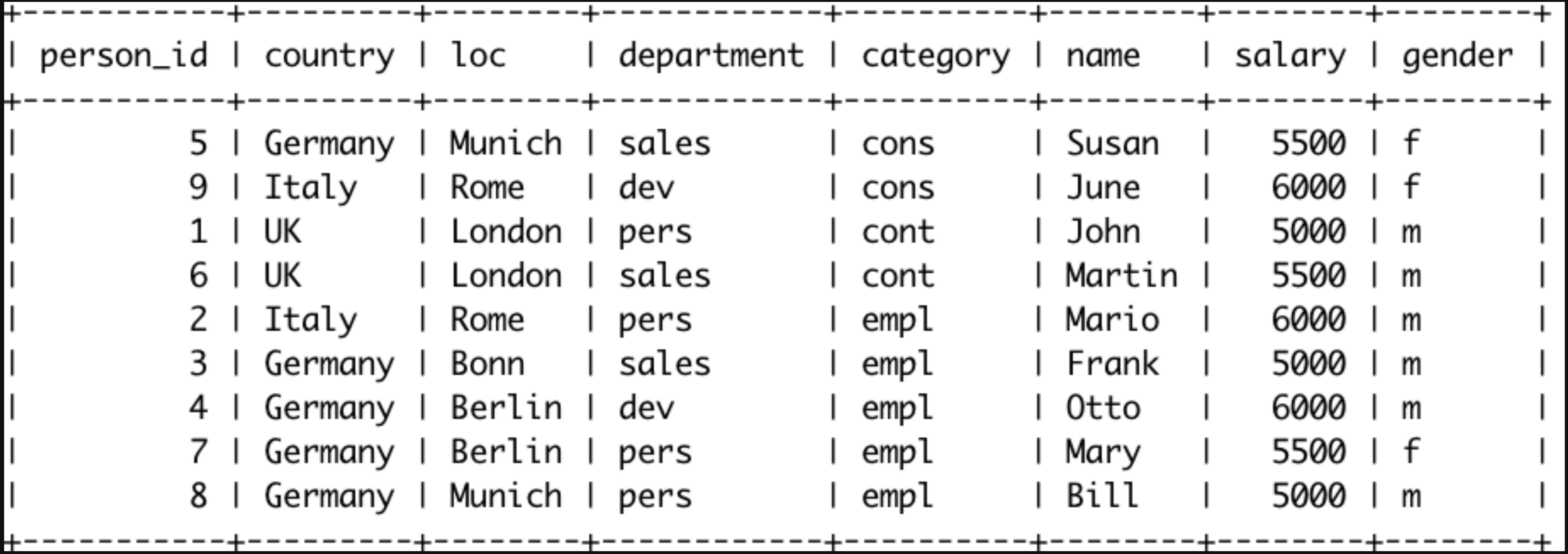
A bit of formality

- A statistical report
- Data grouped by one field
- And a column created for each distinct value of another field

Yes, but what is it, really?

Let's see it in practice

- **Suppose you have this table**



```
+-----------+---------+--------+------------+----------+--------+--------+--------+
| person_id | country | loc    | department | category | name   | salary | gender |
+-----------+---------+--------+------------+----------+--------+--------+--------+
|         5 | Germany | Munich | sales      | cons     | Susan  |   5500 | f      |
|         9 | Italy   | Rome   | dev        | cons     | June   |   6000 | f      |
|         1 | UK      | London | pers       | cont     | John   |   5000 | m      |
|         6 | UK      | London | sales      | cont     | Martin |   5500 | m      |
|         2 | Italy   | Rome   | pers       | empl     | Mario  |   6000 | m      |
|         3 | Germany | Bonn   | sales      | empl     | Frank  |   5000 | m      |
|         4 | Germany | Berlin | dev        | empl     | Otto   |   6000 | m      |
|         7 | Germany | Berlin | pers       | empl     | Mary   |   5500 | f      |
|         8 | Germany | Munich | pers       | empl     | Bill   |   5000 | m      |
+-----------+---------+--------+------------+----------+--------+--------+--------+
```

- **and you want to know how many male and female employees are in each department**

You could try this:

Group by department and gender

```
select department, gender, count(*) as how_many from
all_personnel group by department,gender;
+------------+--------+----------+
| department | gender | how_many |
+------------+--------+----------+
| dev        | f      |        1 |
| dev        | m      |        1 |
| pers       | f      |        1 |
| pers       | m      |        3 |
| sales      | f      |        1 |
| sales      | m      |        2 |
+------------+--------+----------+
```

But wouldn't be better if you got this?

A cross-tabulation query, a.k.a. pivot table

```
+------------+---+---+
| department | f | m |
+------------+---+---+
| dev        | 1 | 1 |
| pers       | 1 | 3 |
| sales      | 1 | 2 |
+------------+---+---+
```

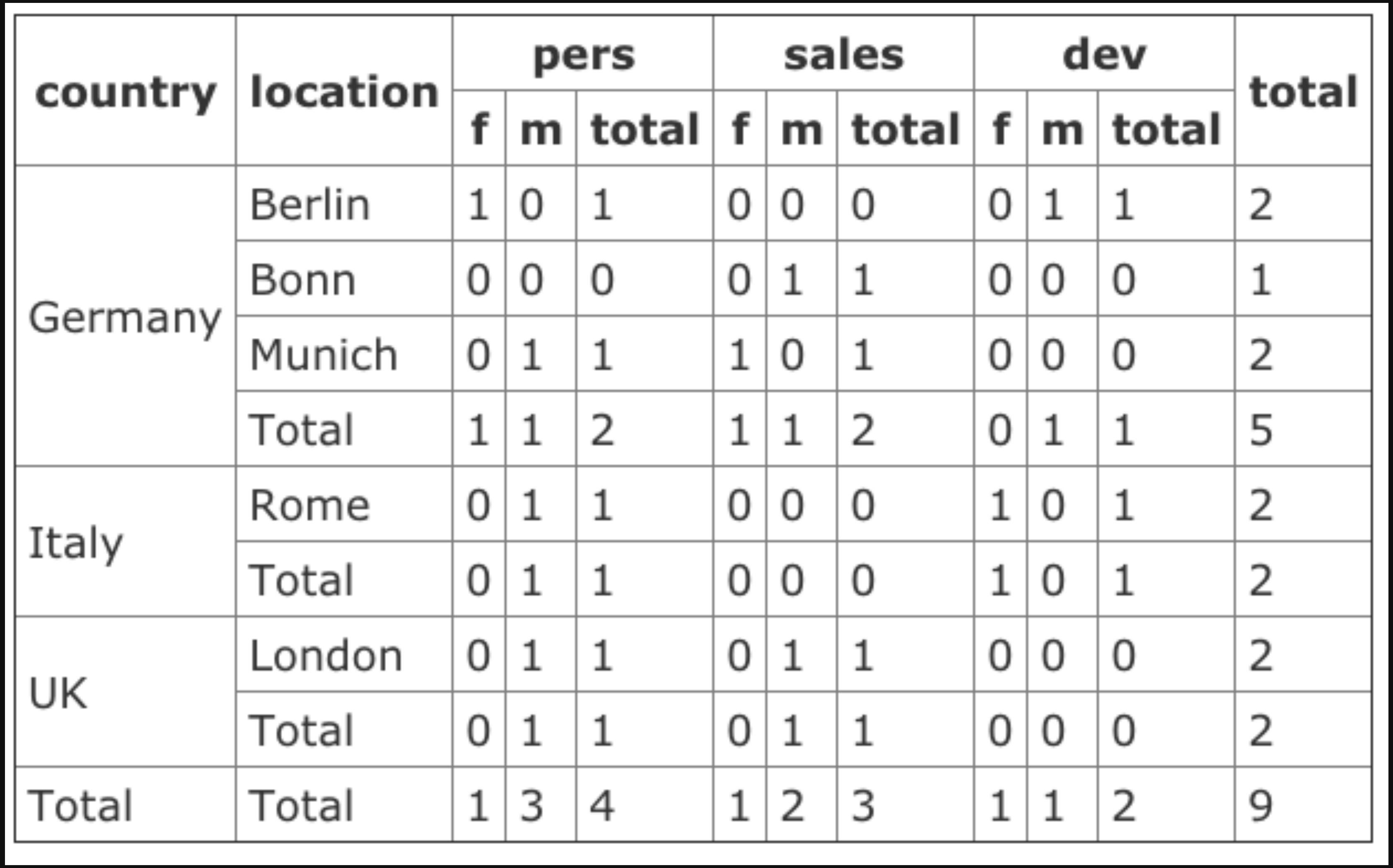
Or even better, this:

A cross-tabulation query with summary

```
+------------+---+---+-------+
| department | f | m | total |
+------------+---+---+-------+
| dev        | 1 | 1 |     2 |
| pers       | 1 | 3 |     4 |
| sales      | 1 | 2 |     3 |
| TOTAL      | 3 | 6 |     9 |
+------------+---+---+-------+
```

Or what if you could do this ...

A multi-level cross-tabulation query

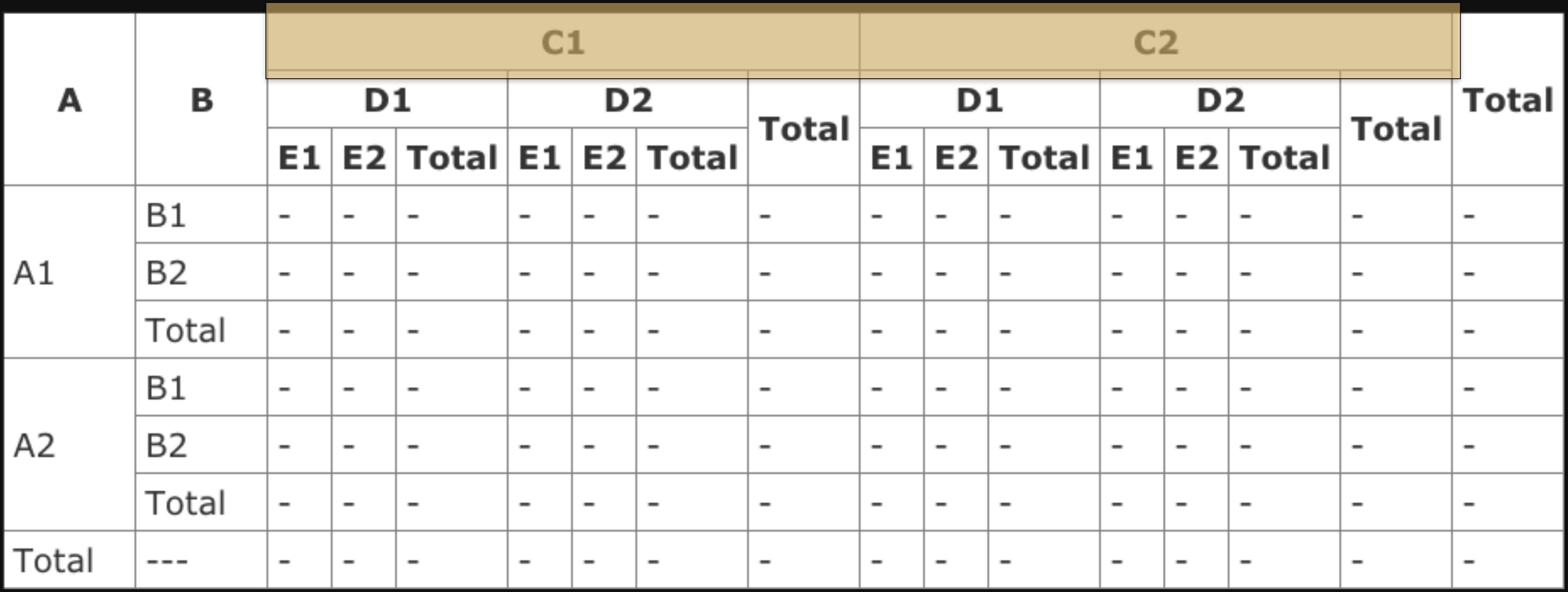


country	location	pers			sales			dev			total
		f	m	total	f	m	total	f	m	total	
Germany	Berlin	1	0	1	0	0	0	0	1	1	2
	Bonn	0	0	0	0	1	1	0	0	0	1
	Munich	0	1	1	1	0	1	0	0	0	2
	Total	1	1	2	1	1	2	0	1	1	5
Italy	Rome	0	1	1	0	0	0	1	0	1	2
	Total	0	1	1	0	0	0	1	0	1	2
UK	London	0	1	1	0	1	1	0	0	0	2
	Total	0	1	1	0	1	1	0	0	0	2
Total	Total	1	3	4	1	2	3	1	1	2	9

We need some definitions first

A multi-level pivot table has several components

Column header #1

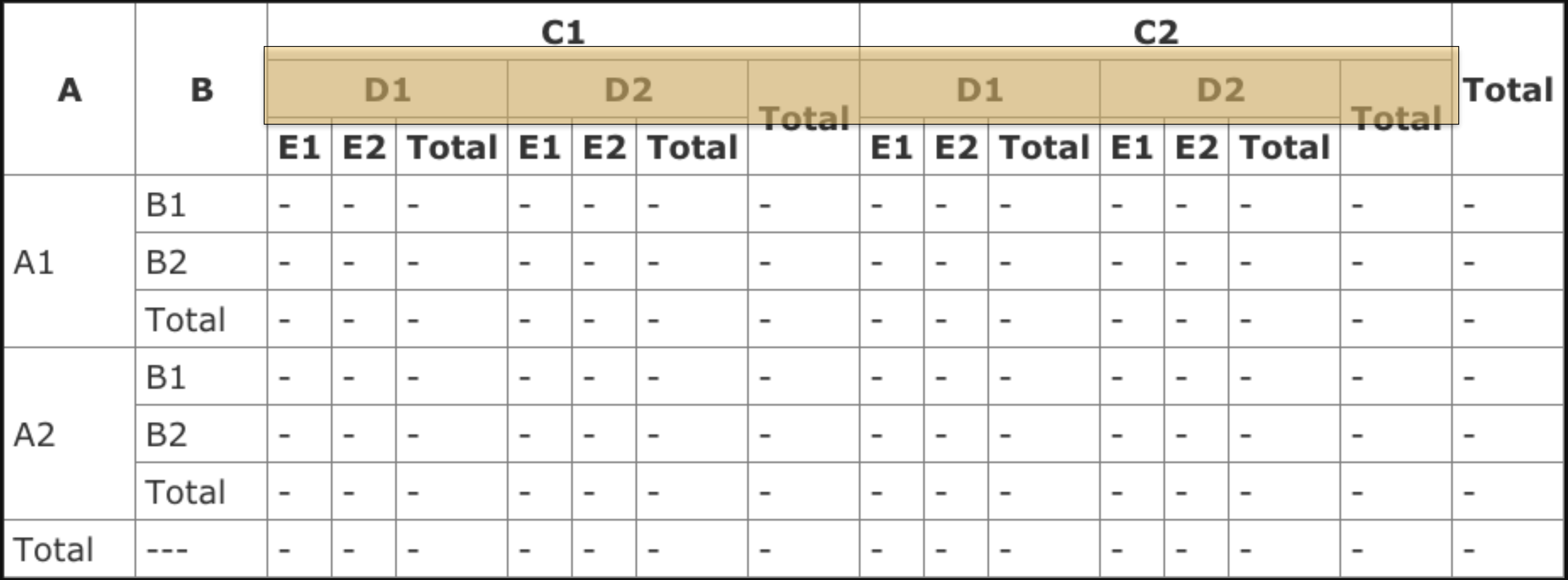


A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

We need some definitions first

A multi-level pivot table has several components

Column header #2

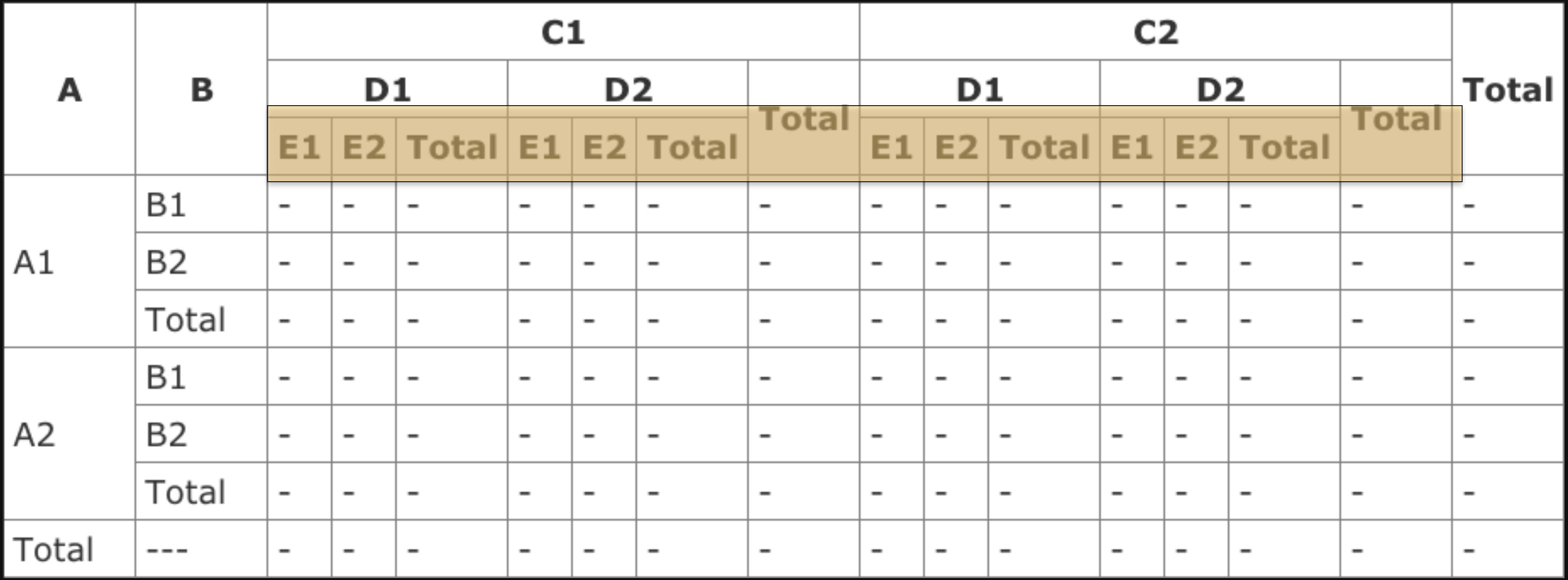


A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

We need some definitions first

A multi-level pivot table has several components

Column header #3

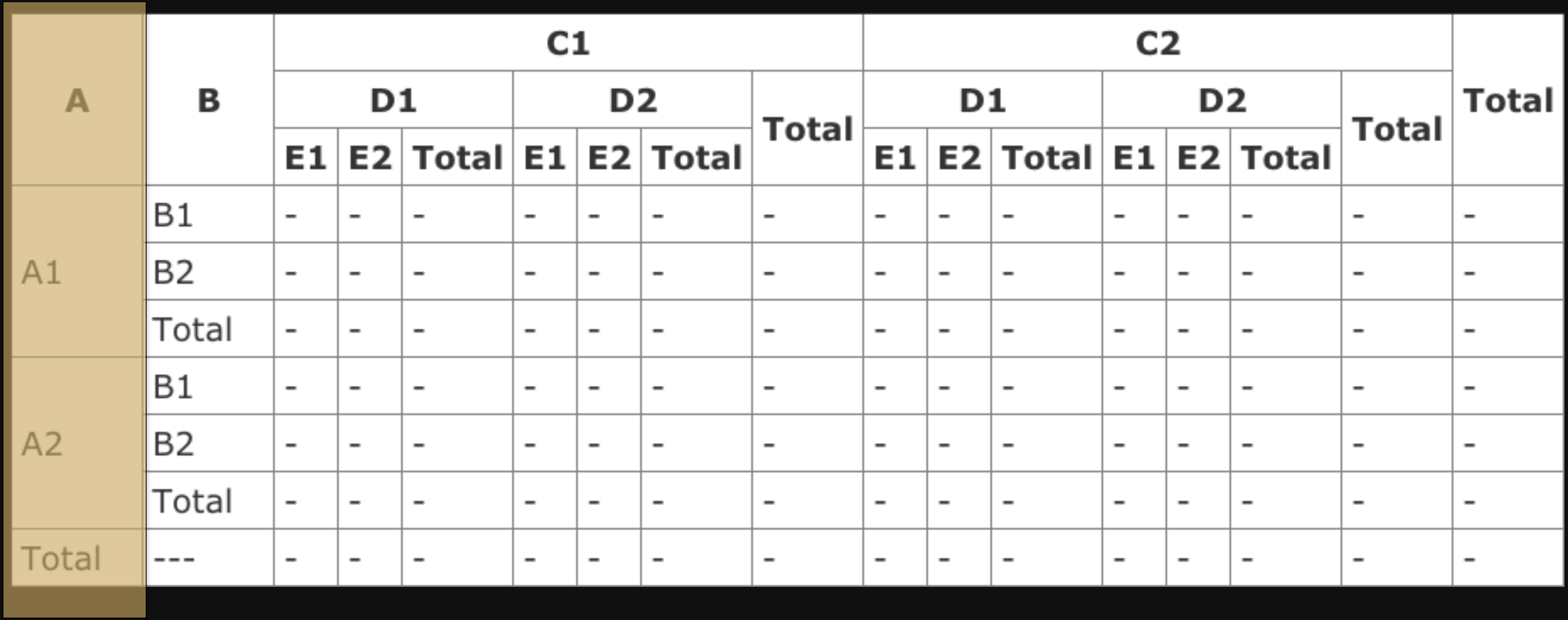


A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

We need some definitions first

A multi-level pivot table has several components

Row header #1

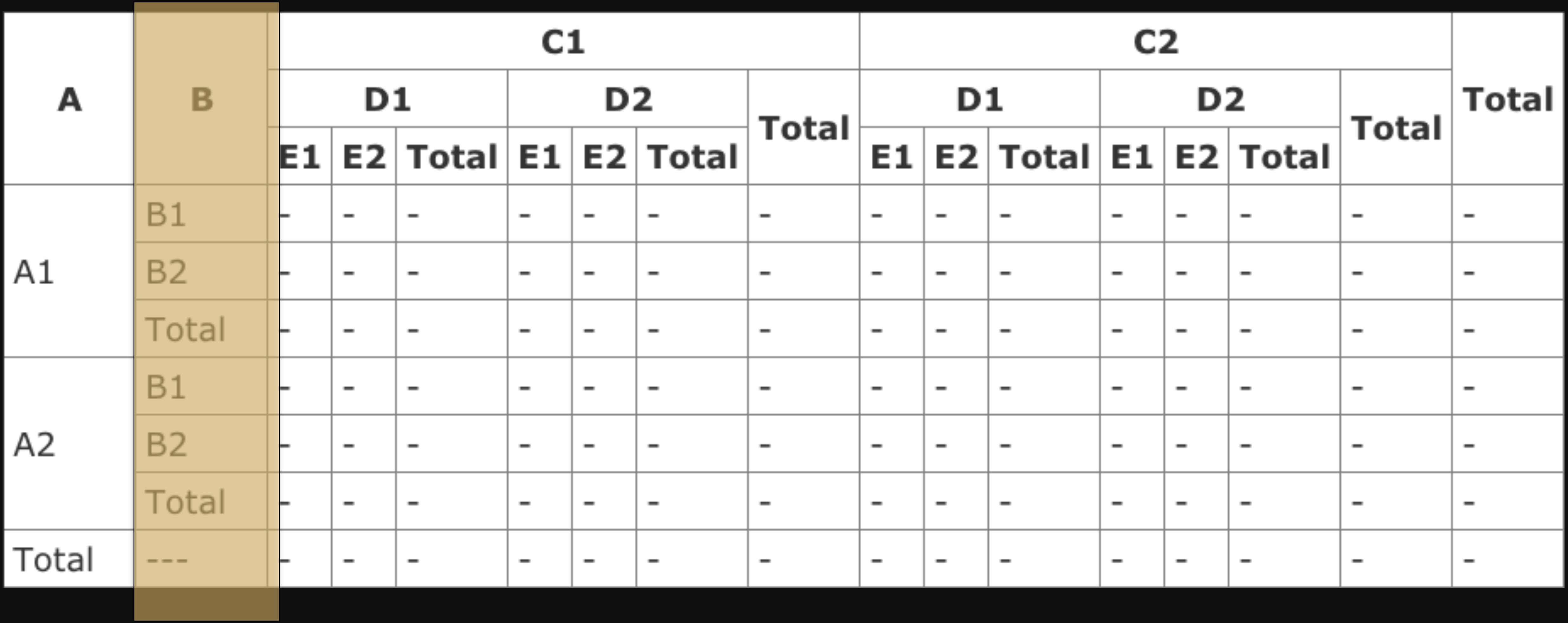


A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

We need some definitions first

A multi-level pivot table has several components

Row header #2

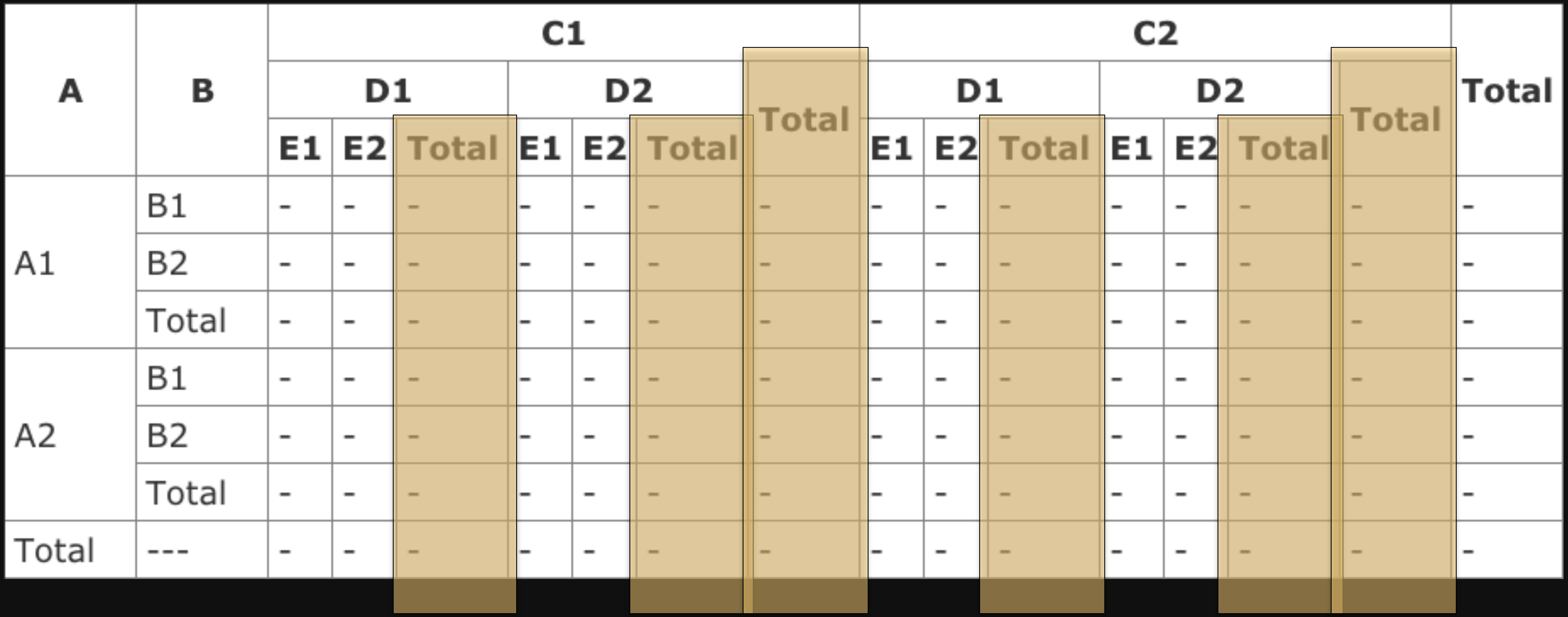


A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

We need some definitions first

A multi-level pivot table has several components

Column sub totals

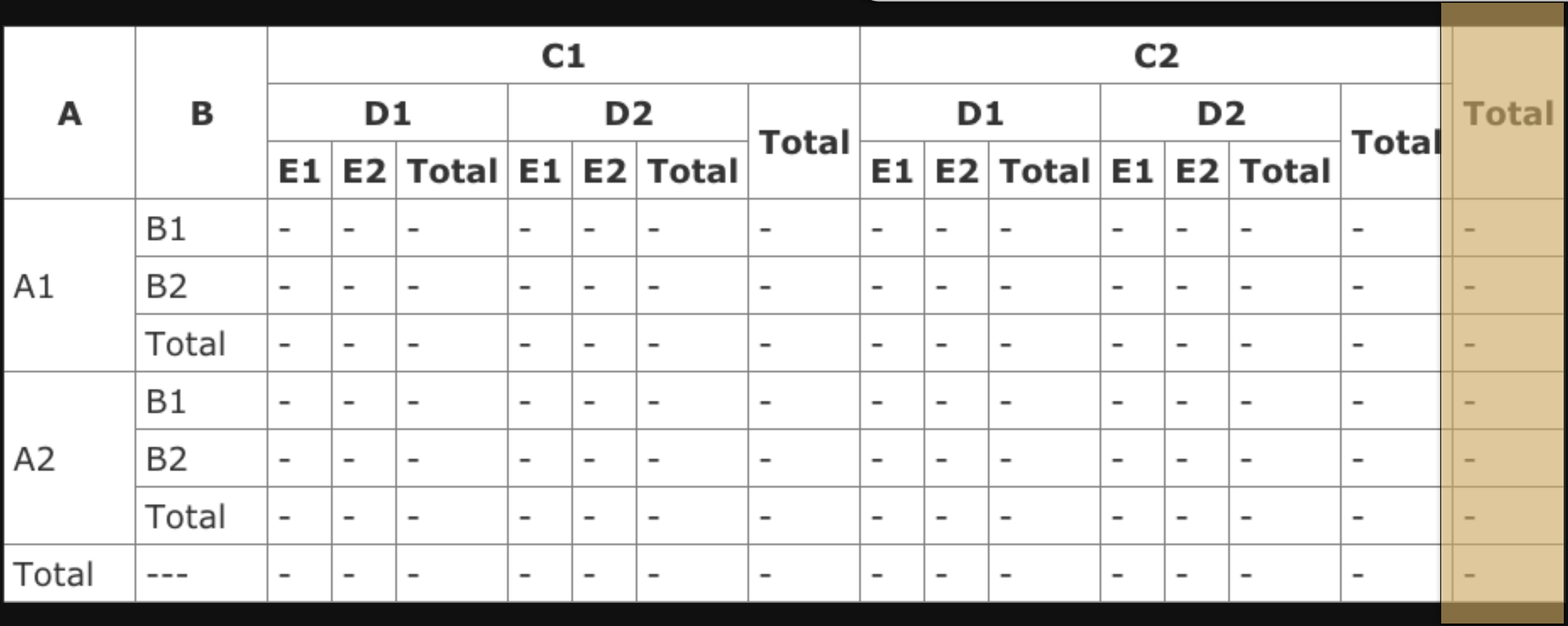


A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

We need some definitions first

A multi-level pivot table has several components

Column grand total

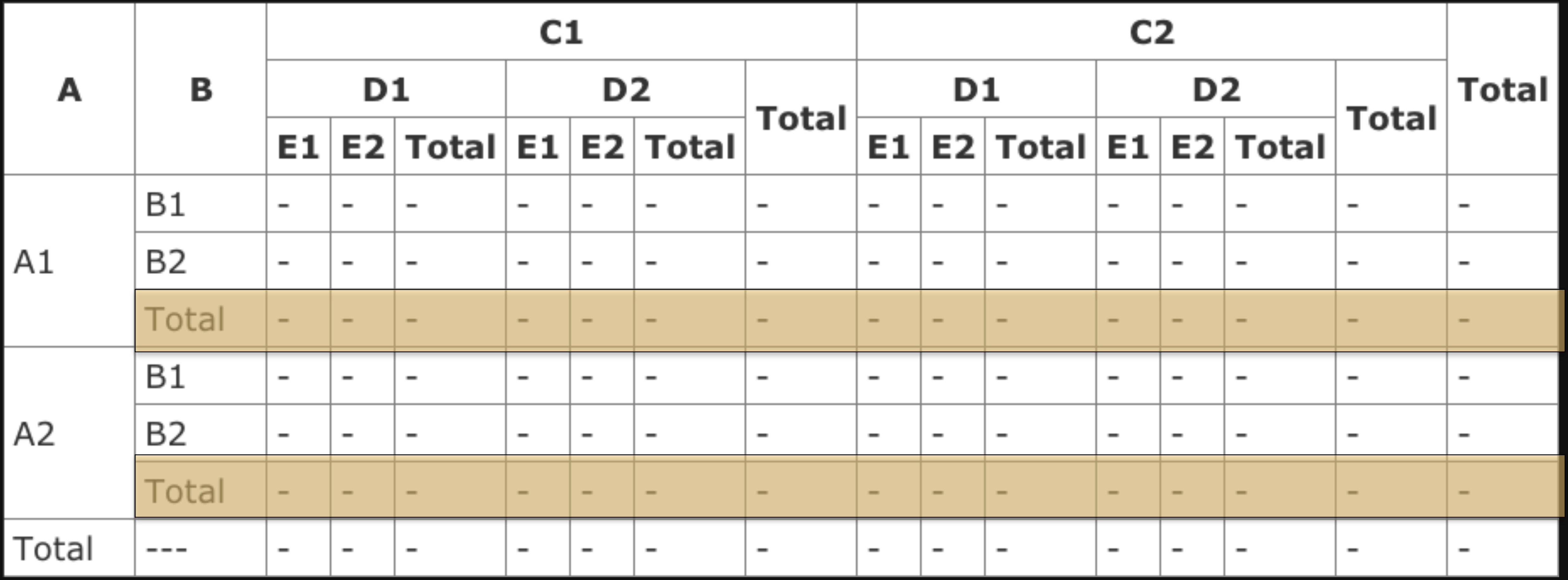


A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

We need some definitions first

A multi-level pivot table has several components

Row sub totals

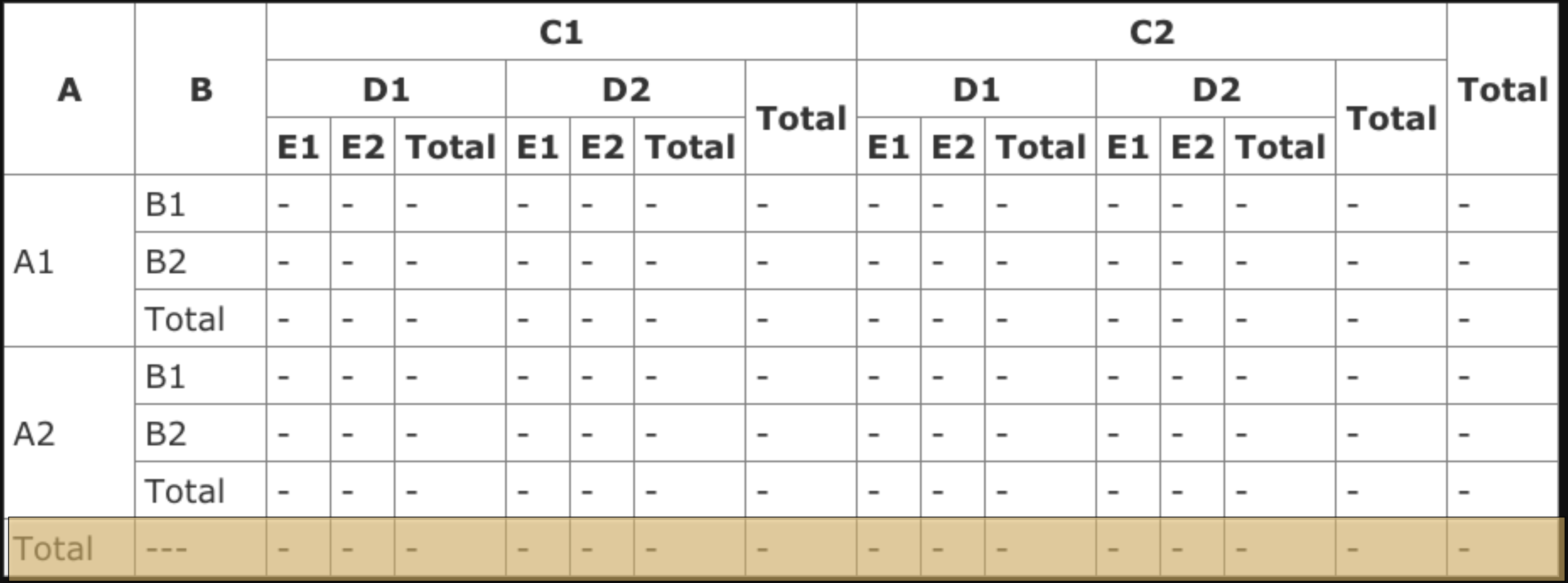


A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

We need some definitions first

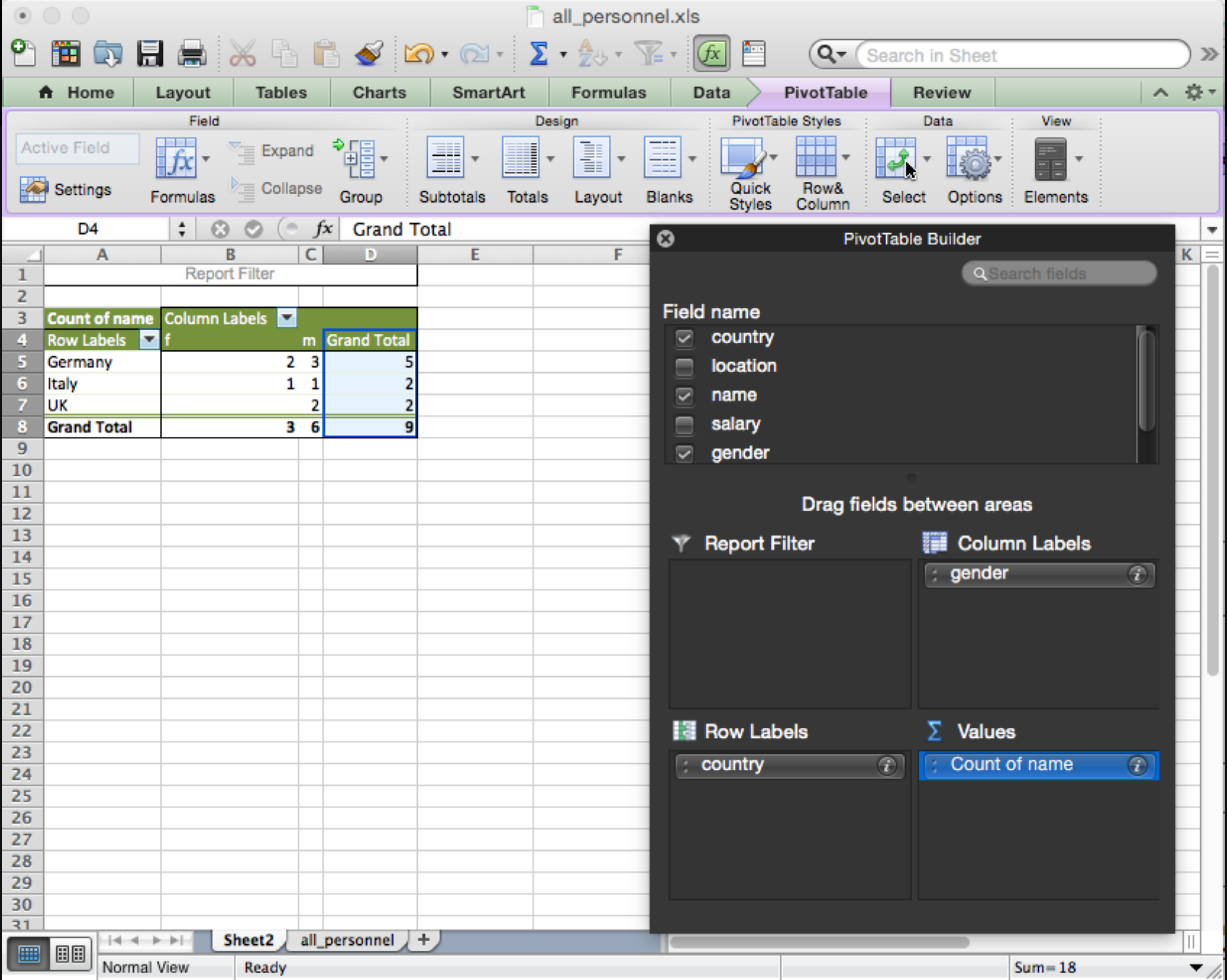
A multi-level pivot table has several components

Row grand total



A	B	C1							C2							Total
		D1			D2			Total	D1			D2			Total	
		E1	E2	Total	E1	E2	Total		E1	E2	Total	E1	E2	Total		
A1	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	B1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	B2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Total	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	---	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Can't we just use Excel?



D4 fx Grand Total

Report Filter

Count of name	Column Labels		
Row Labels	f	m	Grand Total
Germany	2	3	5
Italy	1	1	2
UK		2	2
Grand Total	3	6	9

Sheet2 | all_personnel

Sum= 18

Field name

- ☑ country
- □ location
- ☑ name
- □ salary
- ☑ gender

Report Filter:

Column Labels: gender

Row Labels: country

Values: Count of name

Why not use a spreadsheet

It is tempting, and sometimes it's the right things to do, but ...

- Spreadsheets have limits (32K or 64K rows)
- You need to export your data to a personal computer
- You need to de-normalize the data (your data rarely comes in one table only)
- ... you miss most of the fun!

Back to our "simple" pivot table

A mono-dimension cross-tabulation query, a.k.a. pivot table

```
SELECT department AS department
,count(CASE WHEN gender = 'f' THEN name
       ELSE NULL END) AS 'f'
,count(CASE WHEN gender = 'm' THEN name
       ELSE NULL END) AS 'm'
,count(name) AS total
 FROM all_personnel
 GROUP BY department
```

distinct value becomes column header

```
+------------+---+---+-------+
| department | f | m | total |
+------------+---+---+-------+
| dev        | 1 | 1 |     2 |
| pers       | 1 | 3 |     4 |
| sales      | 1 | 2 |     3 |
+------------+---+---+-------+
```

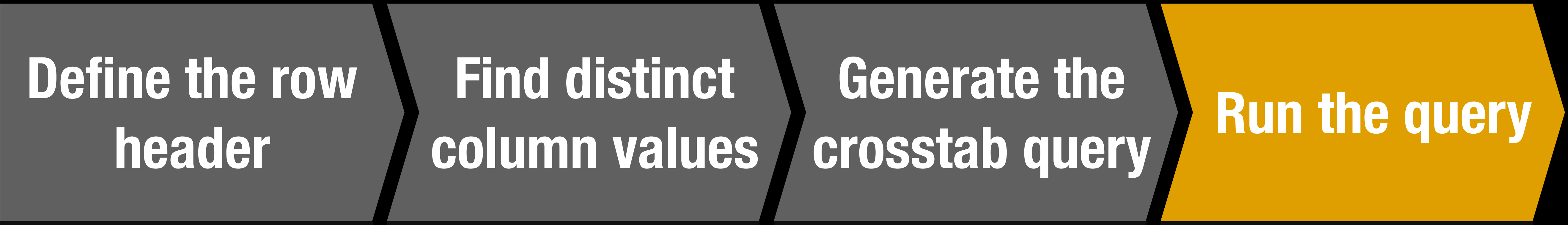
Back to our "simple" pivot table (2)

An alternate syntax

```
SELECT department AS department
,count(if(gender = 'f', name, NULL )) AS 'f'
,count(if(gender = 'm',name, NULL)) AS 'm'
,count(name) AS total
FROM all_personnel
GROUP BY department;
+------------+---+---+-------+
| department | f | m | total |
+------------+---+---+-------+
| dev        | 1 | 1 |     2 |
| pers       | 1 | 3 |     4 |
| sales      | 1 | 2 |     3 |
+------------+---+---+-------+
```

Four Step Pivot table generation

Pivot table queries are defined in several stages



Define the row header

- Simply choose what you want to group by

Find distinct column values

- Run a 'SELECT DISTINCT' query

Generate the crosstab query

- Get the values from the previous step to create the query

Run the query

- Redirect the query to a MySQL client and run it

Step #1 - define the row header

This is (often) simple. Just say which column you want to use

- Your chosen column may not be in the table that has the main data
- You can define your source as a JOIN or a subquery
- (This is the main advantage over spreadsheets)

Step #2 - find the column values

You need to query your data source for distinct values

```
select distinct gender from all_personnel;
+--------+
| gender |
+--------+
| f      |
| m      |
+--------+
```

You can also generate SQL from SQL

But you will find out soon that it is more unpleasant than using a scripting language such as Ruby, Perl, Python.

```
select
 concat(",count(case when gender='", gender,
  "' then name else NULL END) as '", gender,"'")
as q
from ( select distinct gender from all_personnel) as
t;

+-------------------------------------------------------------+
| q                                                           |
+-------------------------------------------------------------+
| ,count(case when gender='f' then name else NULL END) as 'f' |
| ,count(case when gender='m' then name else NULL END) as 'm' |
+-------------------------------------------------------------+
```

Step #3 - generate the crosstab query

The tricky part

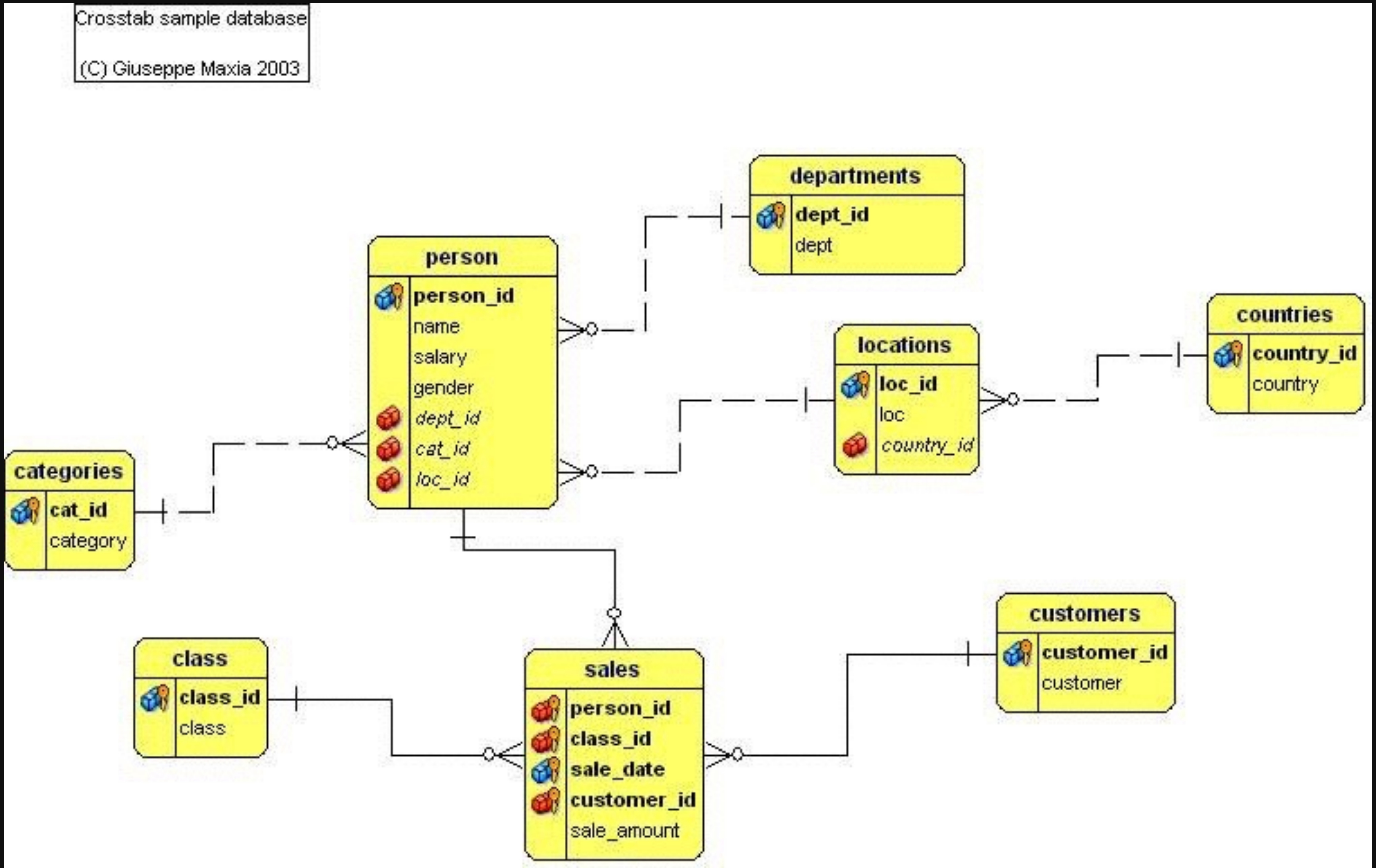
- **You can do it two ways:**
  - Copy the values to an editor, and write the query
  - Use an application that can generate the query for you.

Step #4 - run the final query

- In step #3 you got a SQL query as some text
- You need to pass that text to a database client

When the data source is not just 1 table

What if your row and column headers are not in the same place?



For example:

how do we do a pivot table by country and department?

The data source can be a join

And also the columns can be created with a clever join

```
SELECT country AS country
,sum(CASE WHEN dept_id = '3' THEN salary ELSE NULL
END) AS 'dev'
,sum(CASE WHEN dept_id = '1' THEN salary ELSE NULL
END) AS 'pers'
,sum(CASE WHEN dept_id = '2' THEN salary ELSE NULL
END) AS 'sales'
,sum(salary) AS total
 FROM countries INNER JOIN locations using
(country_id) INNER JOIN person USING (loc_id)
 GROUP BY country
```

BTW: did you notice that we are using SUM instead of COUNT?

You can minimise data scan with FK

When you select the values for a column that has a key in the main table and the real value in a lookup table, you can get both and then combine them in the query

```
SELECT DISTINCT dept_id AS xcol_id, department AS
xcol_alias FROM departments inner join person using
(dept_id);
+---------+------------+
| xcol_id | xcol_alias |
+---------+------------+
|       3 | dev        |
|       1 | pers       |
|       2 | sales      |
+---------+------------+
```

Here's a crosstab from distributed data

It gives you some sort of evil pleasure when you do a query like this, knowing that a spreadsheet could not do it (as easily)

```
+---------+-------+-------+-------+-------+
| country | dev   | pers  | sales | total |
+---------+-------+-------+-------+-------+
| Germany |  6000 | 10500 | 10500 | 27000 |
| Italy   |  6000 |  6000 |  NULL | 12000 |
| UK      |  NULL |  5000 |  5500 | 10500 |
| zzzz    | 12000 | 21500 | 16000 | 49500 |
+---------+-------+-------+-------+-------+
```

Completing the query

Once you got the column values, you still need to create the totals

- **The column subtotal is simply a COUNT(column) after all the column values;**
- **The row subtotal requires A UNION QUERY made of:**
  - a string value to replace the row header ("zzzz" is a good candidate)
  - the same statements for the column values
  - An "ORDER BY header_row" statement, to make zzzz as the last value

This query shows the row total

The grand total will be shown as 'zzzz', which you will need to change either manually or with software

```
SELECT department AS department
,count(CASE WHEN gender = 'f' THEN name ELSE NULL END) AS 'f'
,count(CASE WHEN gender = 'm' THEN name ELSE NULL END) AS 'm'
,count(person_id) AS total
 FROM all_personnel
 GROUP BY department
UNION
SELECT 'zzzz' AS department
,count(CASE WHEN gender = 'f' THEN name ELSE NULL END) AS 'f'
,count(CASE WHEN gender = 'm' THEN name ELSE NULL END) AS 'm'
,count(name) AS total
 FROM all_personnel
 ORDER BY department
```

The grand total ends at the bottom

```
+------------+---+---+-------+
| department | f | m | total |
+------------+---+---+-------+
| dev        | 1 | 1 |     2 |
| pers       | 1 | 3 |     4 |
| sales      | 1 | 2 |     3 |
| zzzz       | 3 | 6 |     9 |
+------------+---+---+-------+
```

Script it!

You don't want to do it manually. The risk of garbling up everything is too high.

- You can do it automatically
- 1. using a Perl module DBIx::SQLCrosstab (http://search.cpan.org)

or

- 2. writing your own software

Sample generation script

Here is a simple case of a mono-dimension cross tab definition to use with DBIx::SQLCrosstab

```
$params = {
    'row_total' => 1,
    'col_total' => 1,
    'use_real_names' => 1,
    'op' => [ [ 'count', 'name' ] ],
    'from' =>    'all_personnel',
    'rows' => [{'col'     => 'department'}],
    'cols' => [
                {
                  'id'    => 'gender',
                  'from' => 'all_personnel'
                }
              ]
};
```

Multi-level crosstab queries

Sometimes, you want to break down a crosstab column by the values of another column. For example, you may want to get salaries by country (row header) and department+gender (columns)

- **To generate multiple level crosstabs, you need to :**
  - 1. get the valid combinations of values from all column
  - 2. generate CASE statements with multiple conditions
  - 3. name the columns with the values of all columns
  - 4. run the query as before

Multi-level column generation in practice

First we get the values for the outer column

```
SELECT DISTINCT dept_id AS xcol_id, department AS
xcol_alias FROM departments inner join  person using
(dept_id);
+---------+-------------+
| xcol_id | xcol_alias |
+---------+-------------+
|       3 | dev         |
|       1 | pers        |
|       2 | sales       |
+---------+-------------+
```

Multi-level column generation in practice

Then we get the values for the inner column

```
SELECT DISTINCT gender from person;
+--------+
| gender |
+--------+
| m      |
| f      |
+--------+
```

Finally, we combine the values

We need to be careful. Not all combinations are always valid

```
SELECT country AS country
,sum(CASE WHEN dept_id = '3' AND gender = 'm' THEN
salary ELSE NULL END) AS 'dev_m'
,sum(CASE WHEN dept_id = '3' AND gender = 'f' THEN
salary ELSE NULL END) AS 'dev_f'
,sum(CASE WHEN dept_id = '3' THEN salary ELSE NULL
END) AS 'dev'
,sum(CASE WHEN dept_id = '1' AND gender = 'm' THEN
salary ELSE NULL END) AS 'pers_m'
,sum(CASE WHEN dept_id = '1' AND gender = 'f' THEN
salary ELSE NULL END) AS 'pers_f'
,sum(CASE WHEN dept_id = '1' THEN salary ELSE NULL
END) AS 'pers'
...
```

Finally, we combine the values (2)

We need to be careful. Not all combinations are always valid

```
# ...
,sum(CASE WHEN dept_id = '2' AND gender = 'm' THEN
salary ELSE NULL END) AS 'sales_m'
,sum(CASE WHEN dept_id = '2' AND gender = 'f' THEN
salary ELSE NULL END) AS 'sales_f'
,sum(CASE WHEN dept_id = '2' THEN salary ELSE NULL
END) AS 'sales'
,sum(salary) AS total
 FROM countries INNER JOIN locations using
(country_id) INNER JOIN person USING (loc_id)
 GROUP BY country
```

When column values are not correlated

You will need to run two queries and combine the values

- **For example, when combining department and location:**
  - you can scan the main datasource with a join to both lookup tables (expensive)
  - or you can scan the lookup tables and the work out all the combinations.

Careful with multiple level column

Sometimes combining values may not be the best option

- **For example, if you want to use two levels, for country and location, you need a correlated query**

```
select distinct country, location from countries join
locations using (country_id)
```

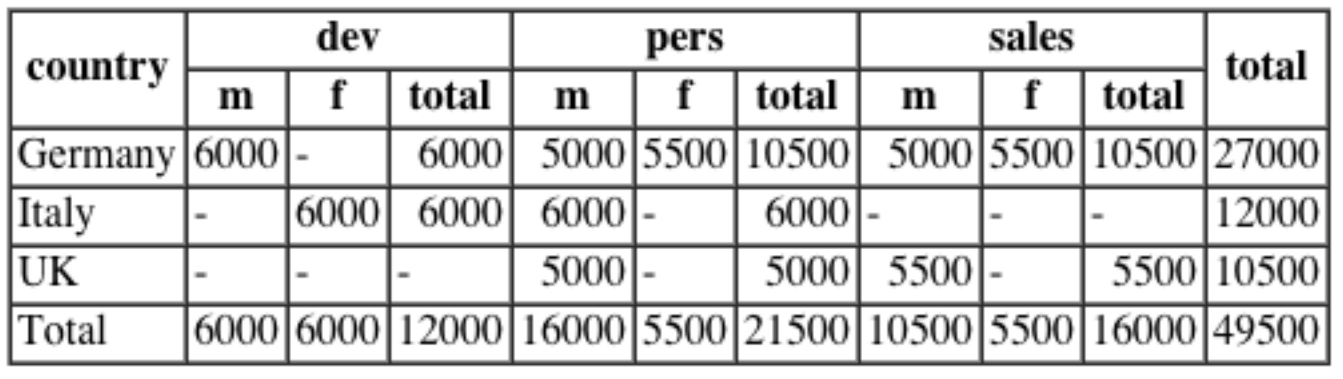
- **If you use two queries and then combine the results, you may end up with unreasonable tuples, such as 'Italy-Paris', 'UK-Berlin', 'USA-Rome'.**

Results may be scary

SQL does not generate hierarchical headers ...

```
+---------+-------+-------+-----+--------+--------+------+---------+---------+-------+-------+
| country | dev_m | dev_f | dev | pers_m | pers_f | pers | sales_m | sales_f | sales | total |
+---------+-------+-------+-----+--------+--------+------+---------+---------+-------+-------+
| Germany |     1 |     0 |   1 |      1 |      1 |    2 |       1 |       1 |     2 |     5 |
| Italy   |     0 |     1 |   1 |      1 |      0 |    1 |       0 |       0 |     0 |     2 |
| UK      |     0 |     0 |   0 |      1 |      0 |    1 |       1 |       0 |     1 |     2 |
| zzzz    |     1 |     1 |   2 |      3 |      1 |    4 |       2 |       1 |     3 |     9 |
+---------+-------+-------+-----+--------+--------+------+---------+---------+-------+-------+
```

But you can format HTML to look the way you want it



country	dev			pers			sales			total
	m	f	total	m	f	total	m	f	total	
Germany	6000	-	6000	5000	5500	10500	5000	5500	10500	27000
Italy	-	6000	6000	6000	-	6000	-	-	-	12000
UK	-	-	-	5000	-	5000	5500	-	5500	10500
Total	6000	6000	12000	16000	5500	21500	10500	5500	16000	49500

Generating XML output

DBIx::SQLCrosstab has also the ability of producing XML output that you can later manipulate on your own

```
<?xml version="1.0"?>
<xtab title="Crosstab"
generator="DBIx::SQLCrosstab::Format version 0.7">
   <country name="Germany">
      <department name="dev">
            <m>6000</m>
            <f/>
            <total>6000</total>
      </department>
      <department name="pers">
            <m>5000</m>
            <f>5500</f>
            <total>10500</total>
      </department>
```

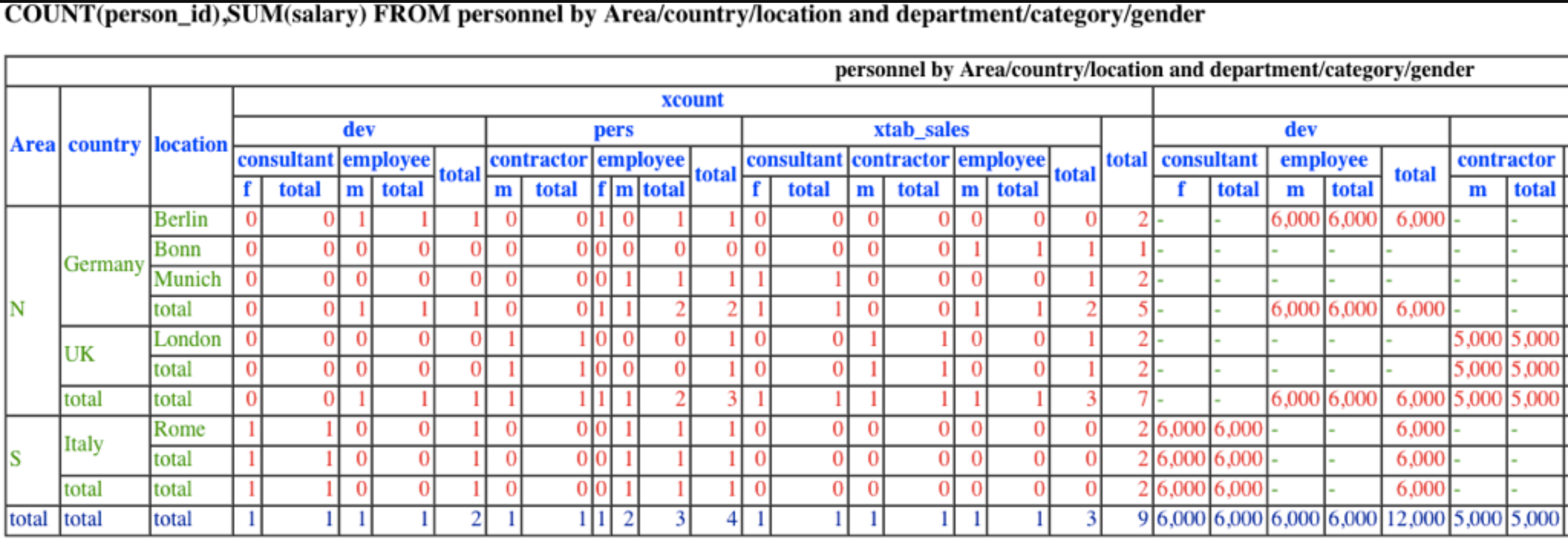
More advanced crosstab features.

What if you want to combine operations?

- **We can generate pivot tables that have COUNT and SUM in the same row**

Some scary stuff ...

Here's a portion of a cross tab with 3 levels by 3 levels, combining COUNT and SUM



COUNT(person_id),SUM(salary) FROM personnel by Area/country/location and department/category/gender

personnel by Area/country/location and department/category/gender

Area	country	location	xcount																									
			dev					pers						xtab_sales							total	dev						
			consultant		employee		total	contractor		employee			total	consultant		contractor		employee		total		consultant		employee		total	contractor	
			f	total	m	total		m	total	f	m	total		f	total	m	total	m	total			f	total	m	total		m	total
N	Germany	Berlin	0	0	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	2	-	-	6,000	6,000	6,000	-	-
		Bonn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	-	-	-	-	-	-	-
		Munich	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	2	-	-	-	-	-	-	-
		total	0	0	1	1	1	0	0	1	1	2	2	1	1	0	0	1	1	2	5	-	-	6,000	6,000	6,000	-	-
	UK	London	0	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	2	-	-	-	-	-	5,000	5,000
		total	0	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	2	-	-	-	-	-	5,000	5,000
	total	total	0	0	1	1	1	1	1	1	1	2	3	1	1	1	1	1	1	3	7	-	-	6,000	6,000	6,000	5,000	5,000
S	Italy	Rome	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	2	6,000	6,000	-	-	6,000	-	-
		total	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	2	6,000	6,000	-	-	6,000	-	-
	total	total	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	2	6,000	6,000	-	-	6,000	-	-
total	total	total	1	1	1	1	2	1	1	1	2	3	4	1	1	1	1	1	1	3	9	6,000	6,000	6,000	6,000	12,000	5,000	5,000

Demo

Slides and example online

- **Slides and examples:**

  http://bit.ly/pivot-samples

  (or check 'datacharmer' on GitHub)

Q&A