



SQL FOR EXPLORATORY DATA ANALYSIS

What's in the database?

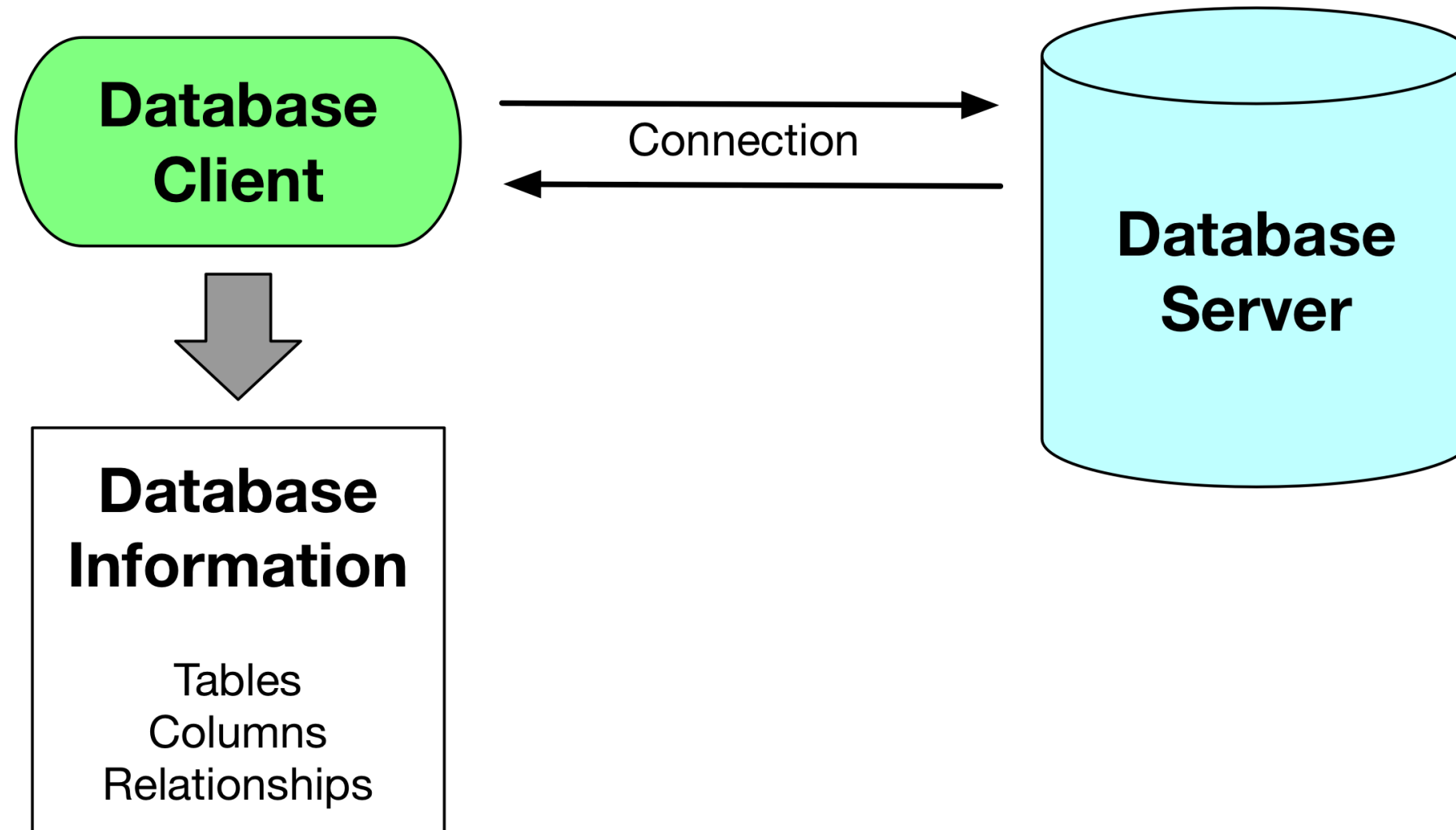
Christina Maimone
Data Scientist

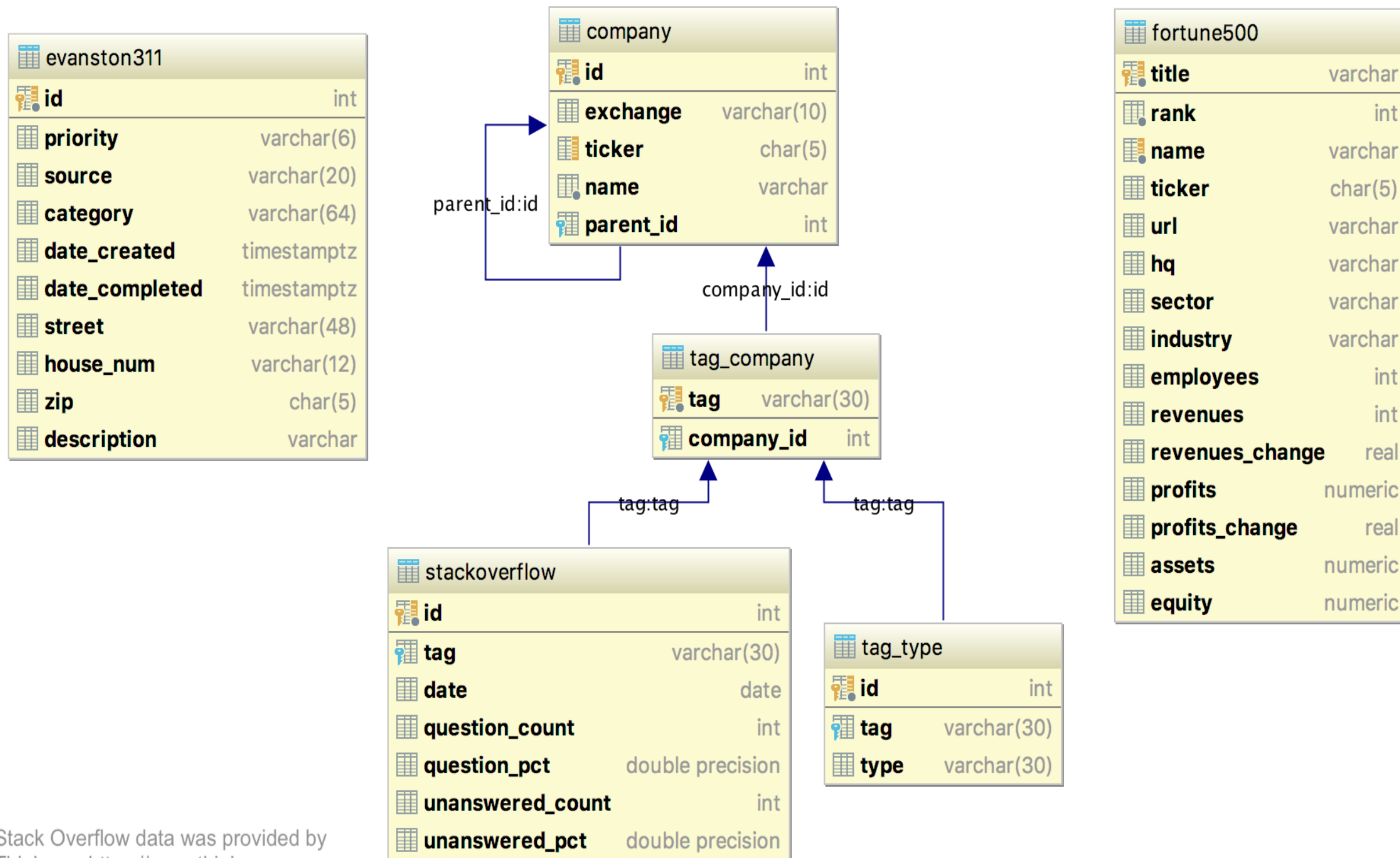
PostgreSQL





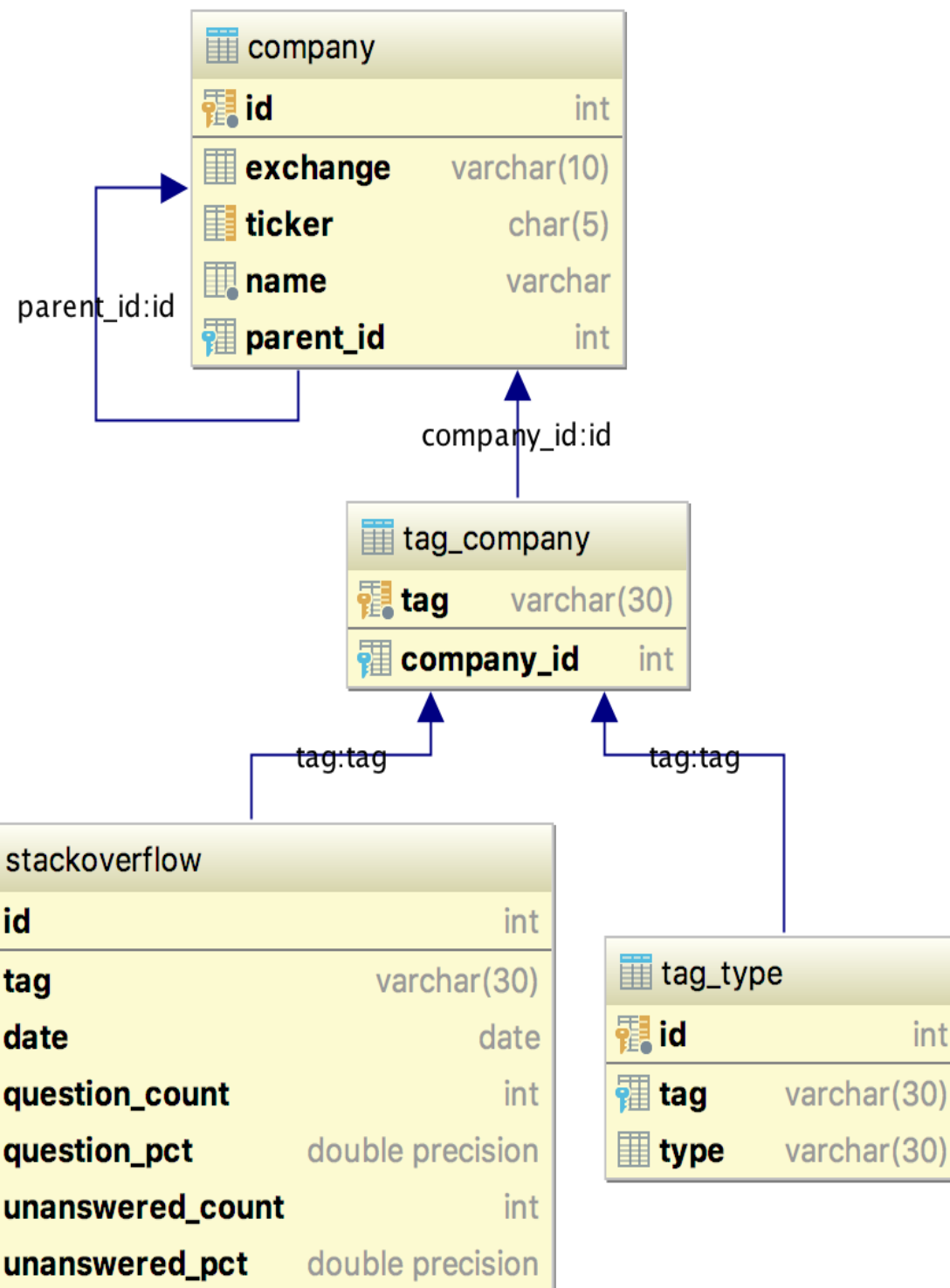
Database client





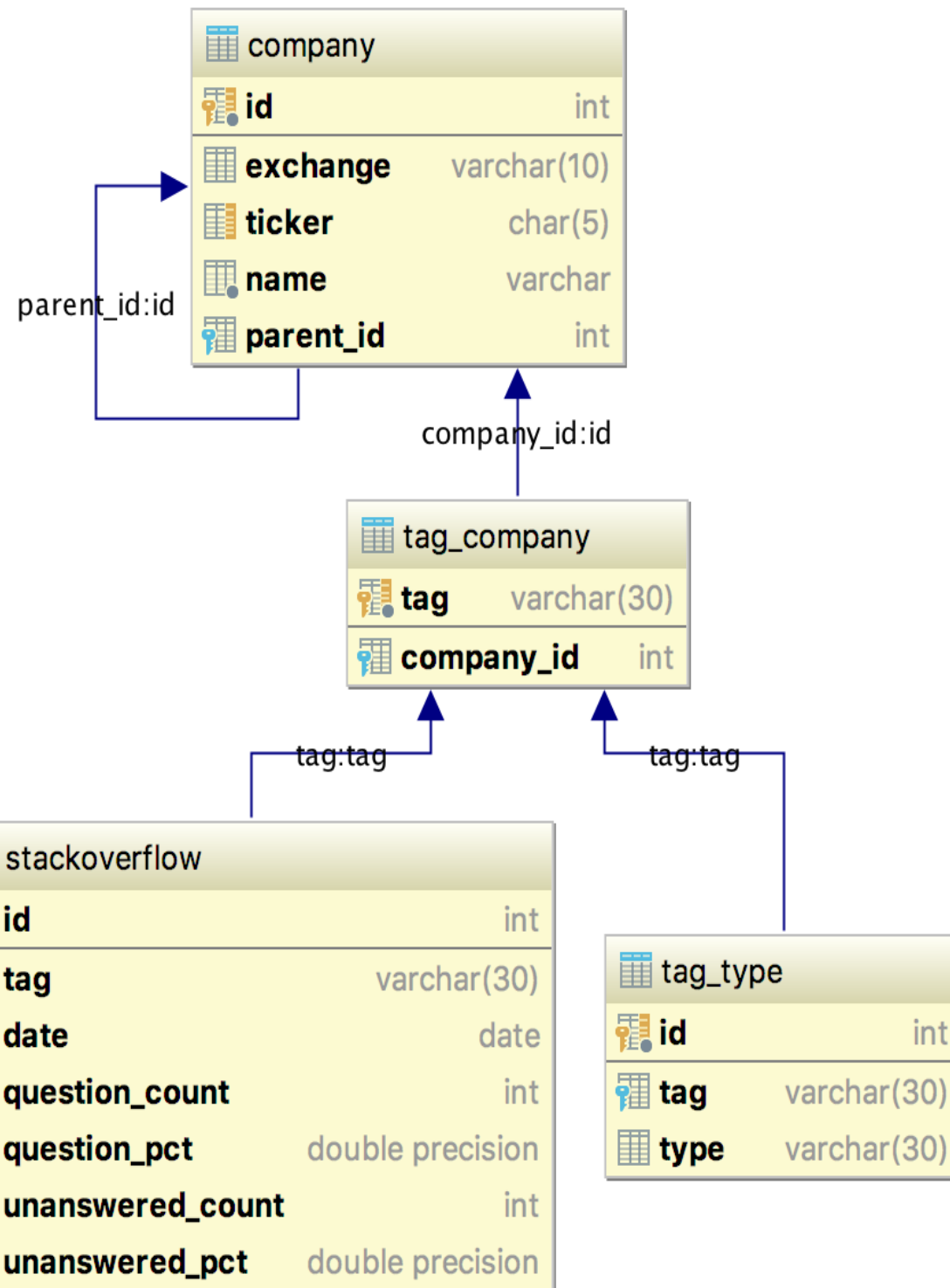
Stack Overflow data was provided by
Thinknum <https://www.thinknum.com>

evanston311	
id	int
priority	varchar(6)
source	varchar(20)
category	varchar(64)
date_created	timestampz
date_completed	timestampz
street	varchar(48)
house_num	varchar(12)
zip	char(5)
description	varchar

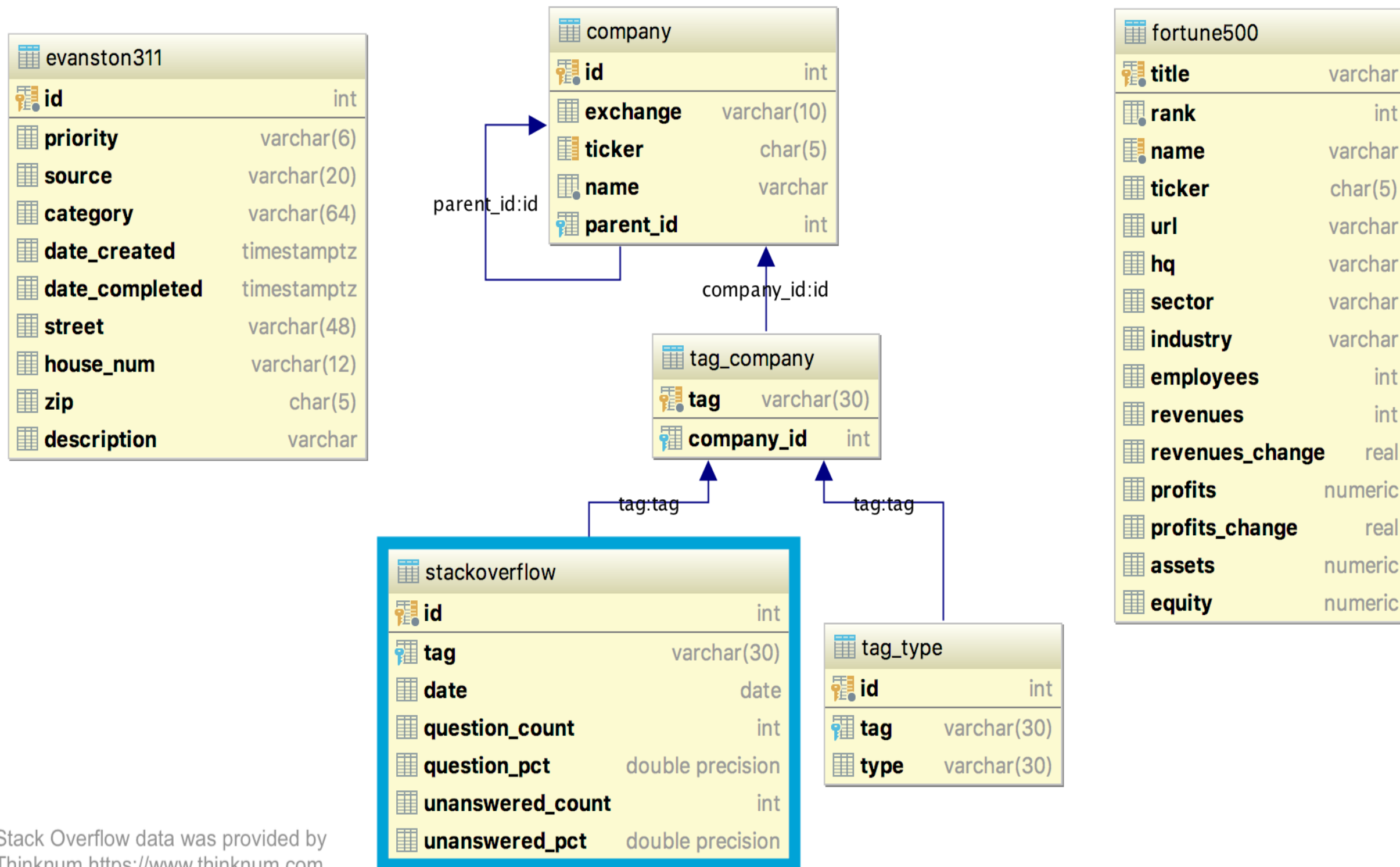


fortune500	
title	varchar
rank	int
name	varchar
ticker	char(5)
url	varchar
hq	varchar
sector	varchar
industry	varchar
employees	int
revenues	int
revenues_change	real
profits	numeric
profits_change	real
assets	numeric
equity	numeric

evanston311	
id	int
priority	varchar(6)
source	varchar(20)
category	varchar(64)
date_created	timestampz
date_completed	timestampz
street	varchar(48)
house_num	varchar(12)
zip	char(5)
description	varchar



fortune500	
title	varchar
rank	int
name	varchar
ticker	char(5)
url	varchar
hq	varchar
sector	varchar
industry	varchar
employees	int
revenues	int
revenues_change	real
profits	numeric
profits_change	real
assets	numeric
equity	numeric



evanston311	
id	int
priority	varchar(6)
source	varchar(20)
category	varchar(64)
date_created	timestampz
date_completed	timestampz
street	varchar(48)
house_num	varchar(12)
zip	char(5)
description	varchar

company	
id	int
exchange	varchar(10)
ticker	char(5)
name	varchar
parent_id	int

parent_id:id

company_id:id

tag_company	
tag	varchar(30)
company_id	int

tag:tag

tag:tag

stackoverflow	
id	int
tag	varchar(30)
date	date
question_count	int
question_pct	double precision
unanswered_count	int
unanswered_pct	double precision

tag_type	
id	int
tag	varchar(30)
type	varchar(30)

fortune500	
title	varchar
rank	int
name	varchar
ticker	char(5)
url	varchar
hq	varchar
sector	varchar
industry	varchar
employees	int
revenues	int
revenues_change	real
profits	numeric
profits_change	real
assets	numeric
equity	numeric



Select a few rows

```
SELECT *  
  FROM company  
 LIMIT 5;
```

id	exchange	ticker	name	parent_id
1	nasdaq	PYPL	PayPal Holdings, Inc.	
2	nasdaq	AMZN	Amazon.com, Inc.	
3	nasdaq	MSFT	Microsoft Corporation	
4	nasdaq	MDB	MongoDB Inc.	
5	nasdaq	DBX	Dropbox, Inc.	

(5 rows)

A few reminders

Code	Note
NULL	missing



A few reminders

Code	Note
<code>NULL</code>	missing
<code>IS NULL, IS NOT NULL</code>	don't use <code>= NULL</code>



A few reminders

Code	Note
<code>NULL</code>	missing
<code>IS NULL, IS NOT NULL</code>	don't use <code>= NULL</code>
<code>count (*)</code>	number of rows



A few reminders

Code	Note
<code>NULL</code>	missing
<code>IS NULL, IS NOT NULL</code>	don't use <code>= NULL</code>
<code>count (*)</code>	number of rows
<code>count(column_name)</code>	number of non-NULL values



A few reminders

Code	Note
<code>NULL</code>	missing
<code>IS NULL, IS NOT NULL</code>	don't use <code>= NULL</code>
<code>count (*)</code>	number of rows
<code>count (column_name)</code>	number of non-NULL values
<code>count (DISTINCT column_name)</code>	number of different non-NULL values



A few reminders

Code	Note
<code>NULL</code>	missing
<code>IS NULL, IS NOT NULL</code>	don't use <code>= NULL</code>
<code>count (*)</code>	number of rows
<code>count(column_name)</code>	number of non-NULL values
<code>count(DISTINCT column_name)</code>	number of different non-NULL values
<code>SELECT DISTINCT column_name ...</code>	distinct values, including NULL



SQL FOR EXPLORATORY DATA ANALYSIS

Let's start exploring



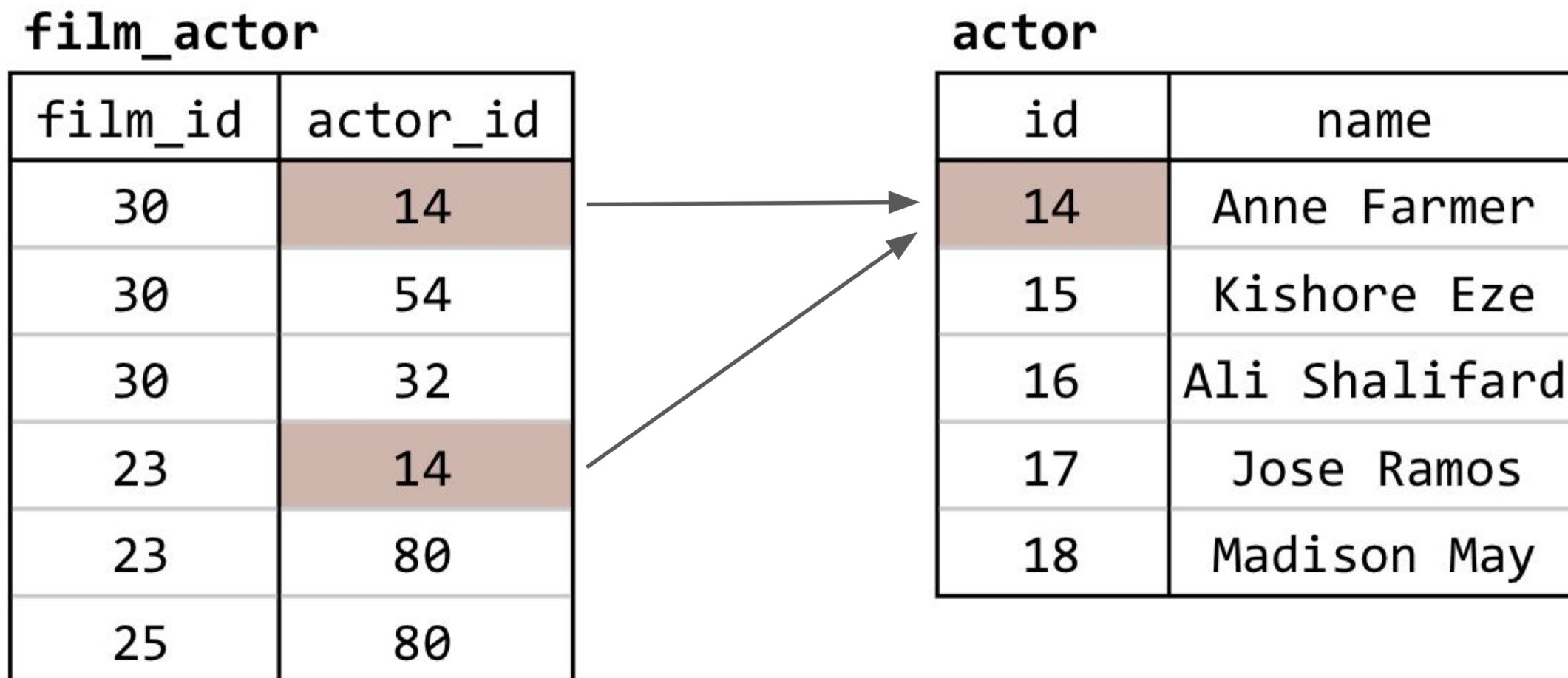
SQL FOR EXPLORATORY DATA ANALYSIS

The keys to the database

Christina Maimone
Data Scientist



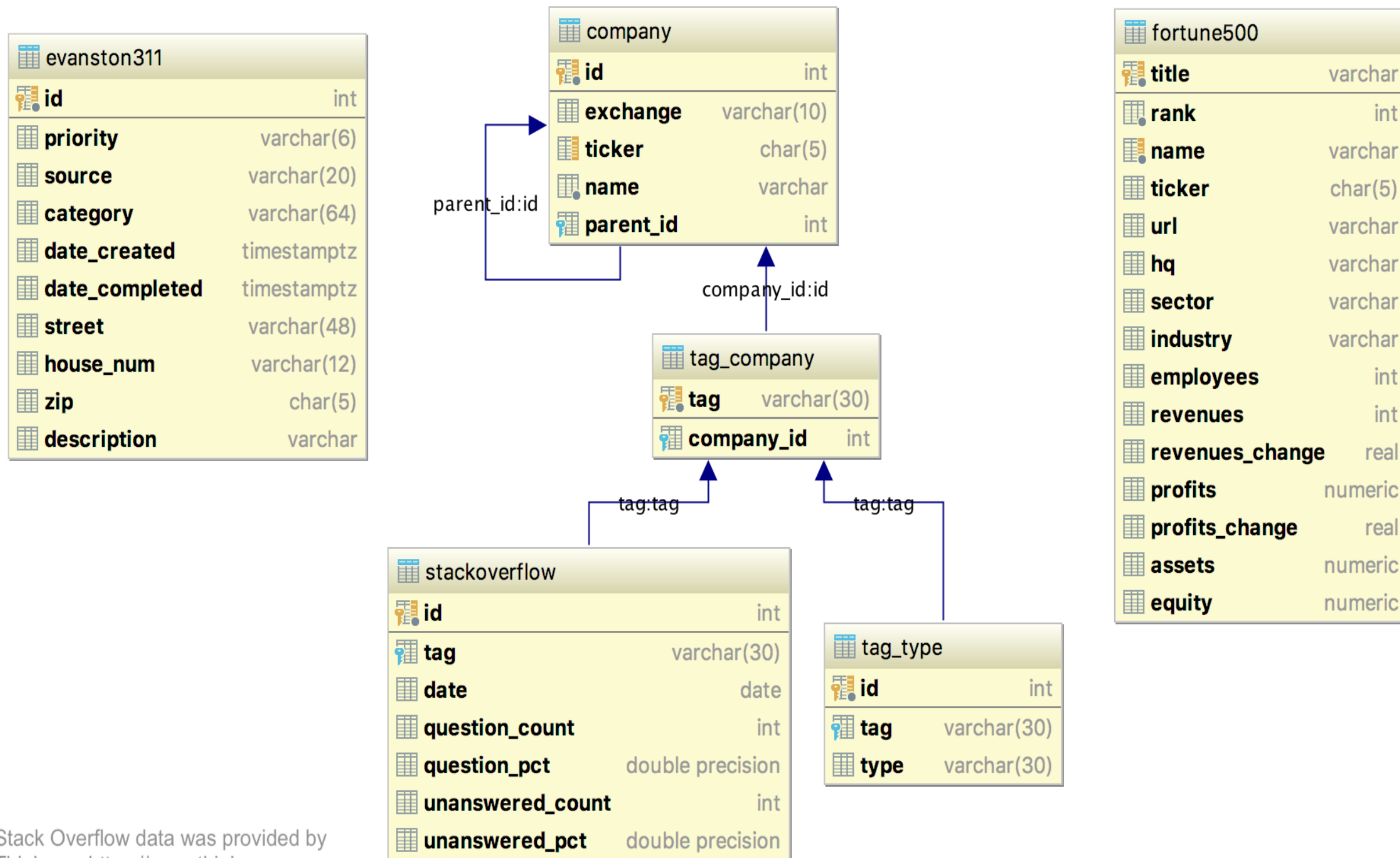
Foreign keys



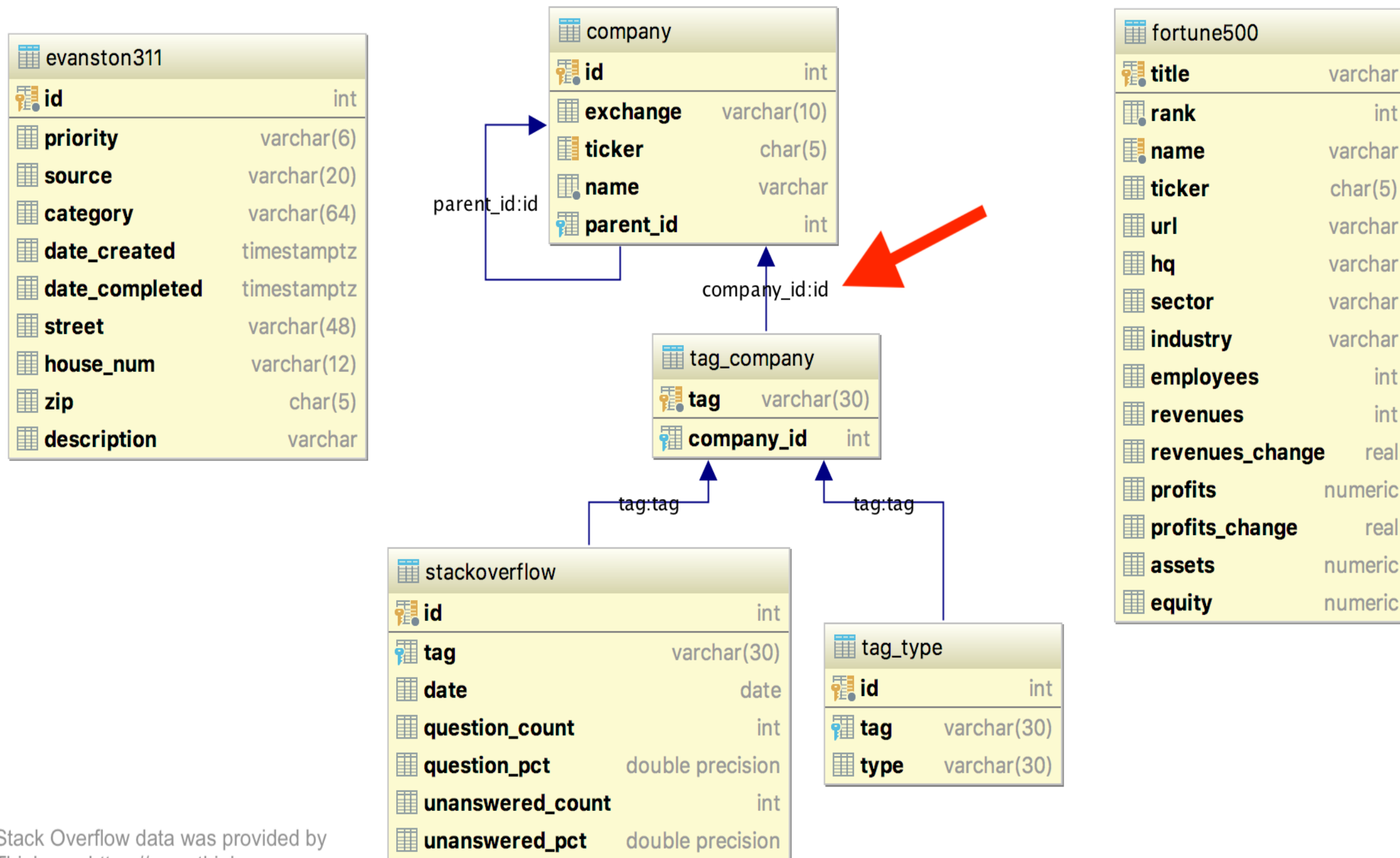


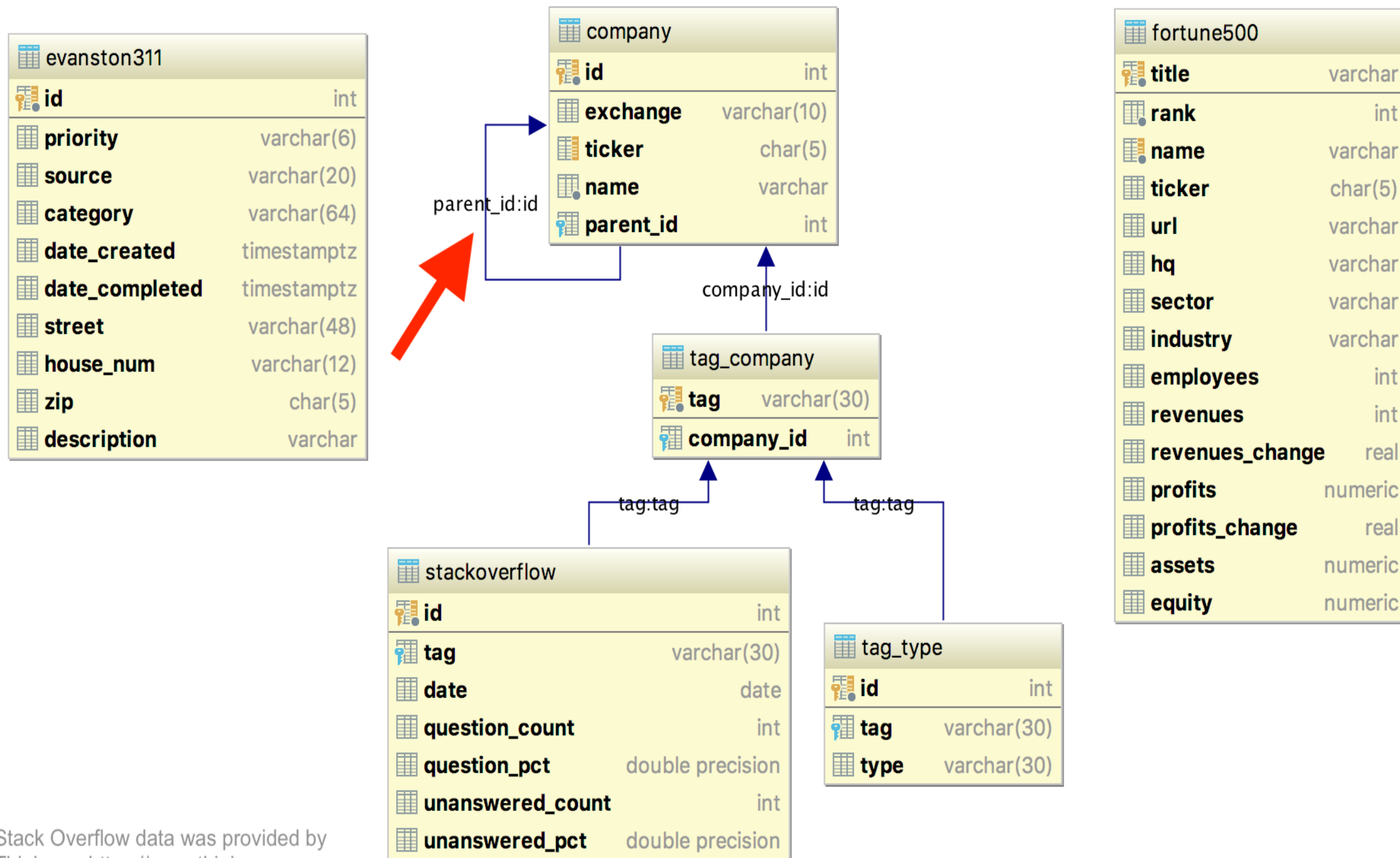
Foreign keys

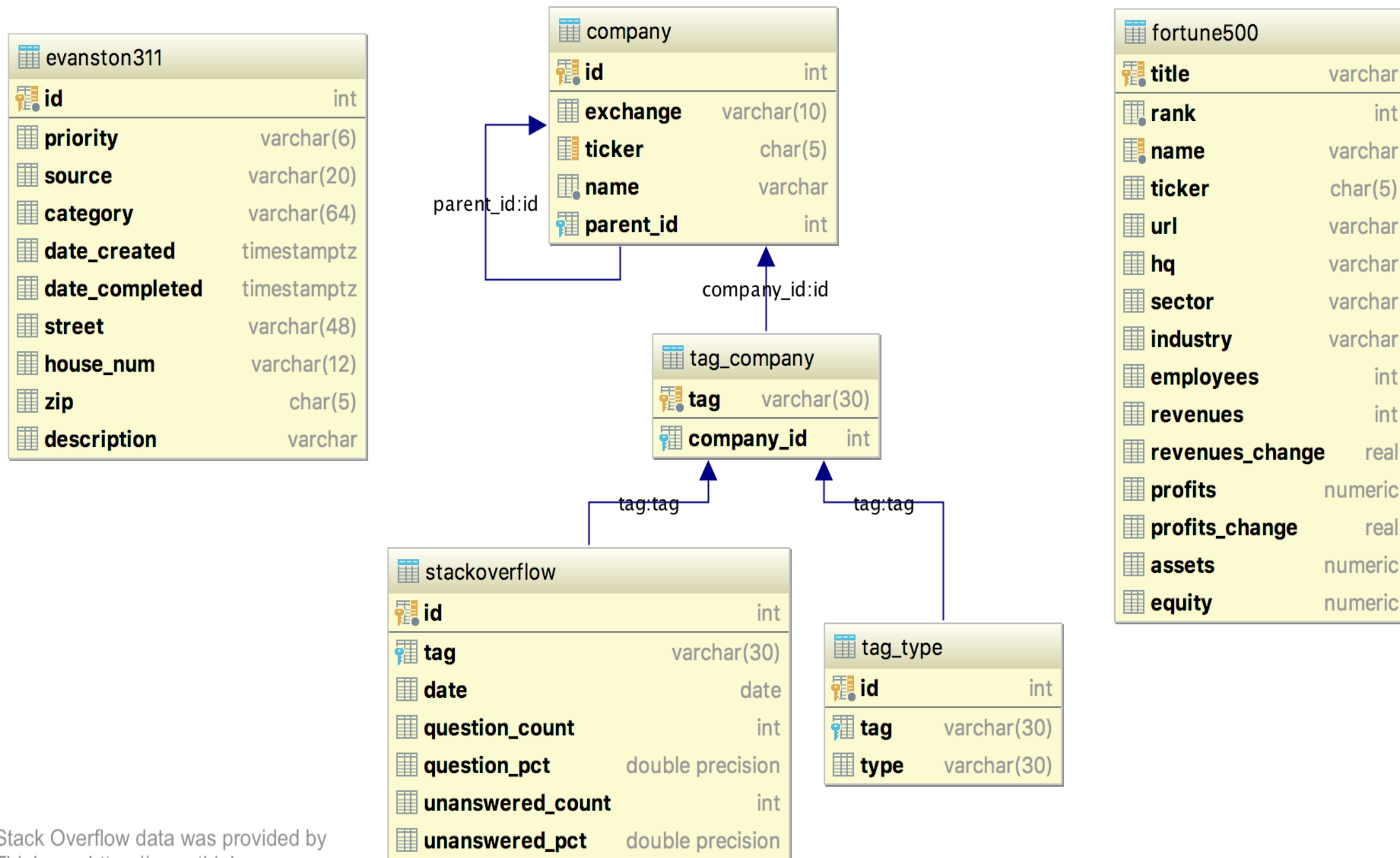
- Reference another row
 - In a different table or the same table
 - Via a unique ID
 - >> Primary key column containing unique, non-NULL values
- Values restricted to values in referenced column OR `NULL`



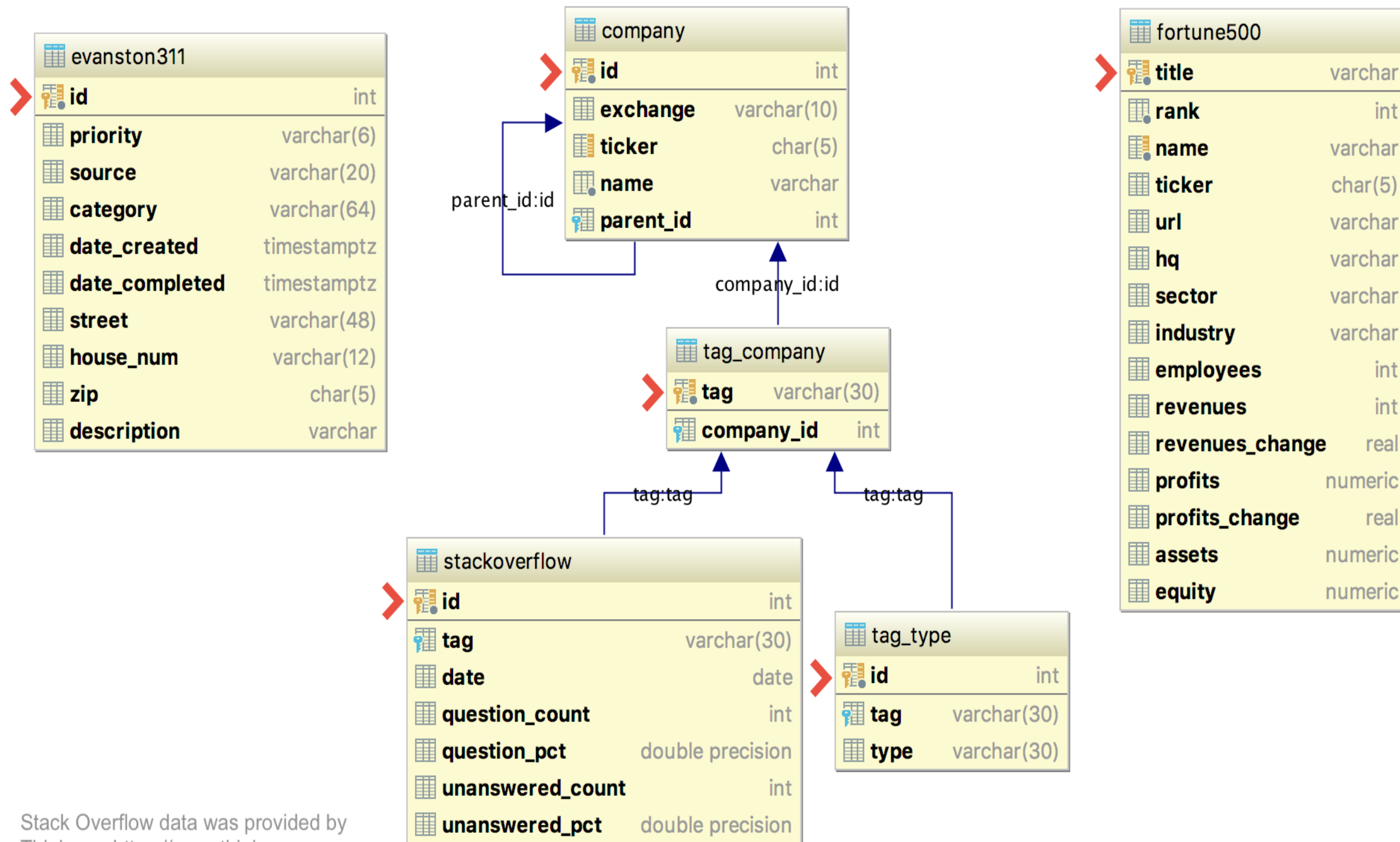
Stack Overflow data was provided by
Thinknum <https://www.thinknum.com>







Stack Overflow data was provided by
Thinknum <https://www.thinknum.com>





Coalesce function

```
coalesce(value_1, value_2 [, ...])
```

- Operates row by row
- Returns first non-NULL value



Coalesce function

```
SELECT *  
FROM prices;
```

column_1	column_2
	10
22	
3	4

```
SELECT coalesce(column_1, column_2)  
FROM prices;
```

coalesce
10
22
3



SQL FOR EXPLORATORY DATA ANALYSIS

Time to keep exploring!



SQL FOR EXPLORATORY DATA ANALYSIS

Column Types and Constraints

Christina Maimone
Data Scientist



Column constraints

- **Foreign key:** value that exists in the referenced column, or `NULL`
- **Primary key:** unique, not `NULL`
- **Unique:** values must all be different except for `NULL`
- **Not null:** `NULL` not allowed: must have a value
- **Check constraints:** conditions on the values
 - `column1 > 0`
 - `columnA > columnB`



Data types

Common

- Numeric
- Character
- Date/Time
- Boolean

Special

- Arrays
- Monetary
- Binary
- Geometric
- Network Address
- XML
- JSON
- and more!
















Numeric types: PostgreSQL documentation

Table 8-2. Numeric Types

Name	Storage Size	Description	Range
<code>smallint</code>	2 bytes	small-range integer	-32768 to +32767
<code>integer</code>	4 bytes	typical choice for integer	-2147483648 to +2147483647
<code>bigint</code>	8 bytes	large-range integer	-9223372036854775808 to +9223372036854775807
<code>decimal</code>	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
<code>numeric</code>	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
<code>real</code>	4 bytes	variable-precision, inexact	6 decimal digits precision
<code>double precision</code>	8 bytes	variable-precision, inexact	15 decimal digits precision
<code>smallserial</code>	2 bytes	small autoincrementing integer	1 to 32767
<code>serial</code>	4 bytes	autoincrementing integer	1 to 2147483647
<code>bigserial</code>	8 bytes	large autoincrementing integer	1 to 9223372036854775807



Types in entity relationship diagrams

fortune500	
 title	varchar
 rank	int ←
 name	varchar
 ticker	char(5)
 url	varchar
 hq	varchar
 sector	varchar
 industry	varchar
 employees	int ←
 revenues	int ←
 revenues_change	real ←
 profits	numeric ←
 profits_change	real ←
 assets	numeric ←
 equity	numeric ←



Casting with CAST()

Format

```
-- With the CAST function  
SELECT CAST (value AS new_type);
```

Examples

```
-- Cast 3.7 as an integer  
SELECT CAST (3.7 AS integer);
```

```
4
```

```
-- Cast a column called total as an integer  
SELECT CAST (total AS integer)  
FROM prices;
```



Casting with ::

Format

```
-- With :: notation  
SELECT value::new_type;
```

Examples

```
-- Cast 3.7 as an integer  
SELECT 3.7::integer;
```

```
-- Cast a column called total as an integer  
SELECT total::integer  
FROM prices;
```



SQL FOR EXPLORATORY DATA ANALYSIS

Time to practice!