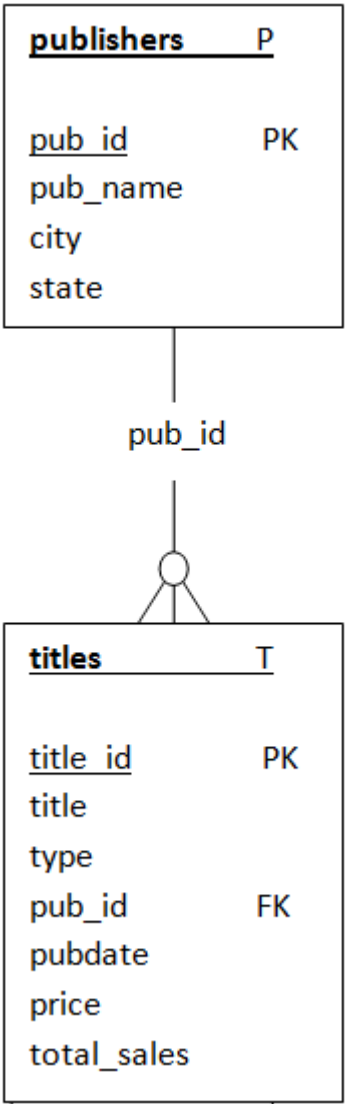


Quel est le livre le plus cher édité par « New Age Books » ?



Query - pubs2 on nediart@10.0.0.45:5432 *

File Edit Query Favouirites Macros View Help

pubs2 on nediart@10.0.0.45:5432

SQL Editor Graphical Query Builder Scratch pad

```
SELECT t.title
FROM titles t
WHERE t.price >= (
    SELECT MAX(t.price)
    FROM titles t, publishers p
    WHERE t.pub_id=p.pub_id
    AND p.pub_name='New Age Books'
)
```

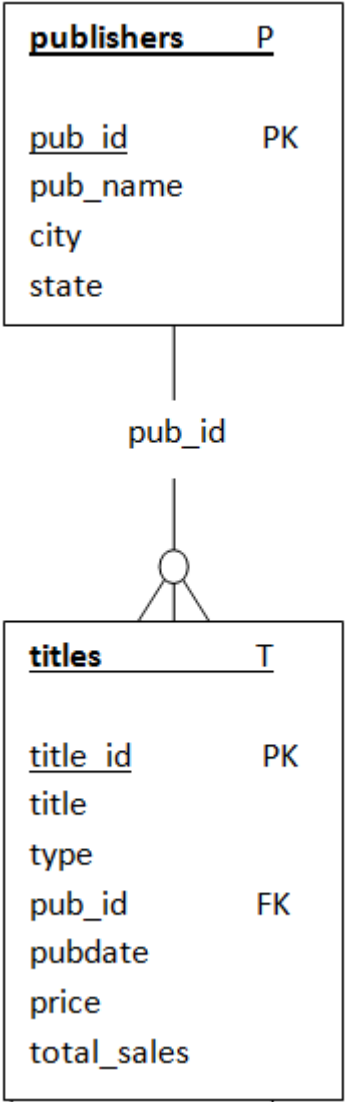
Output pane

Data Output Explain Messages History

	title character varying(80)
1	Secrets of Silicon Valley
2	The Busy Executive's Data
3	Prolonged Data Deprivation
4	Silicon Valley Gastronomic T
5	But Is It User Friendly?
6	Onions, Leeks, and Garlic:
7	Straight Talk About Compu
8	Computer Phobic and Non-

OK. Unix Ln 3 Col 17 Ch 48 8 rows. 11 ms

Quel est le livre le plus cher édité par « New Age Books » ?



Query - pubs2 on nediart@10.0.0.45:5432 *

File Edit Query Favourites Macros View Help

pubs2 on nediart@10.0.0.45:5432

SQL Editor Graphical Query Builder Scratch pad

```
SELECT t.title
FROM titles t, publishers p
WHERE t.pub_id=p.pub_id
AND p.pub_name='New Age Books'
AND t.price >= (
    SELECT MAX(t.price)
    FROM titles t, publishers p
    WHERE t.pub_id=p.pub_id
    AND p.pub_name='New Age Books'
)
```

Output pane

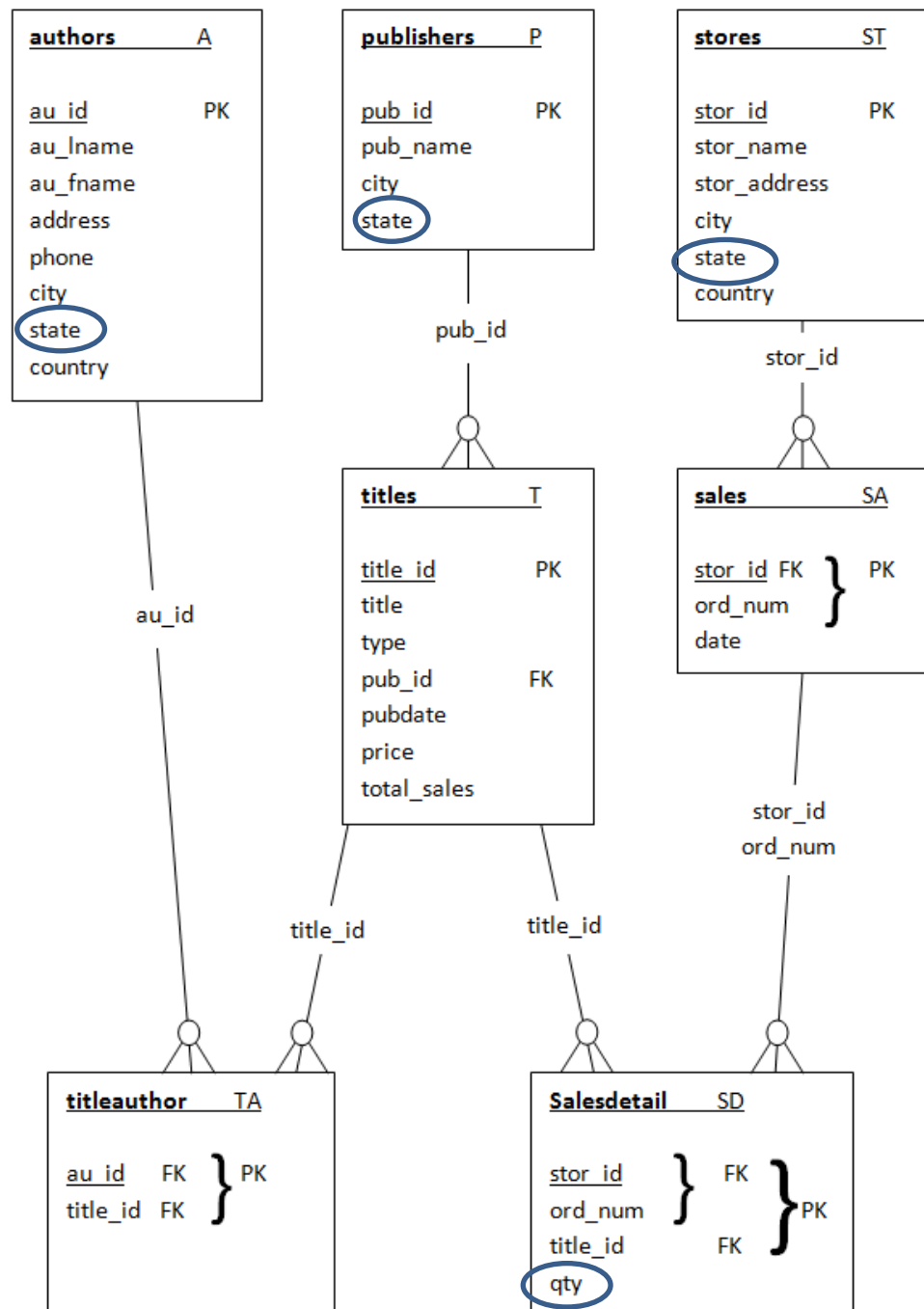
Data Output Explain Messages History

	title
	character varying(80)
1	Prolonged Data Deprivation

OK. Unix Ln 9 Col 42 Ch 237 1 row. 11 ms

Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

Etape 1 : Identifier les champs dont on a besoin

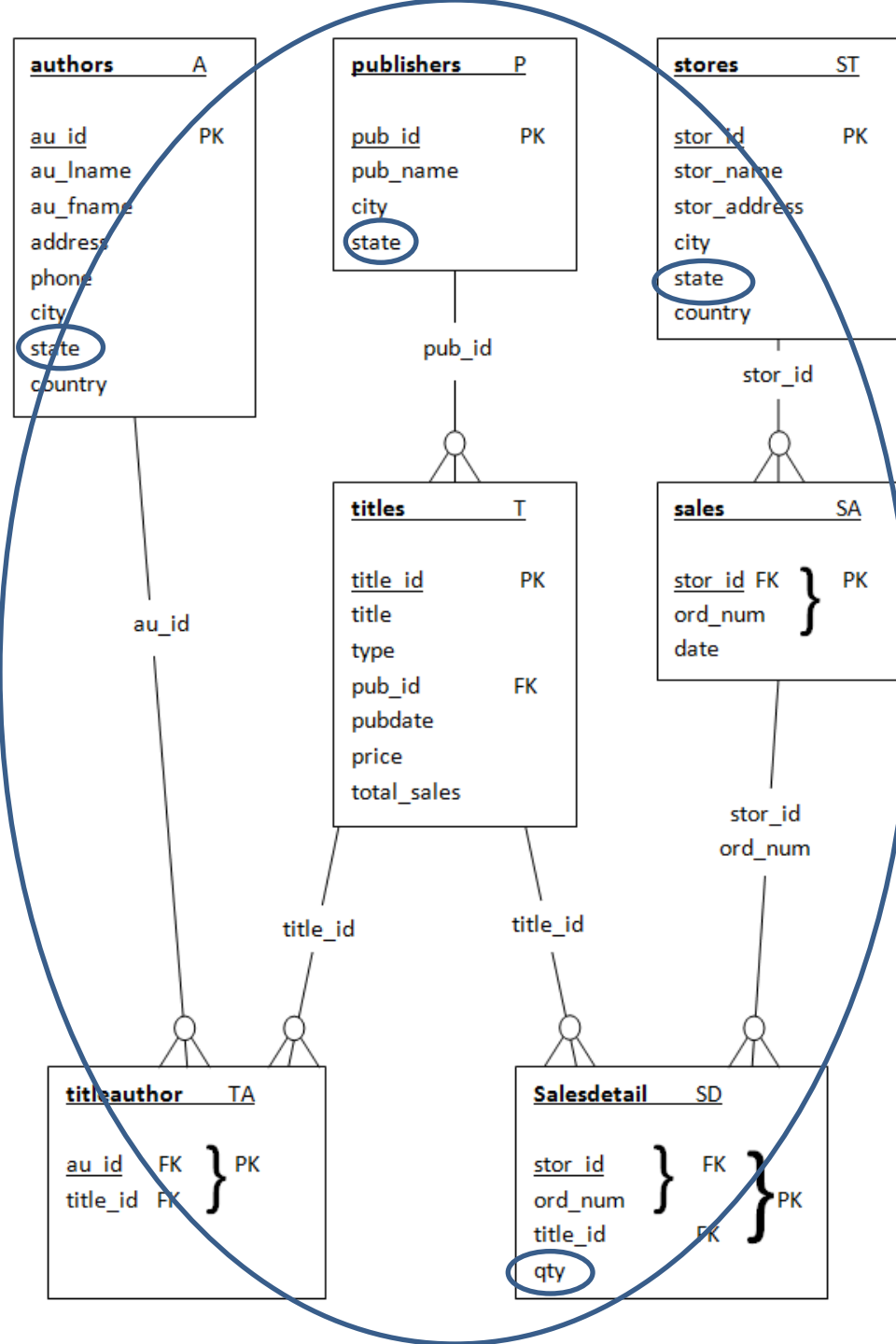


Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

Etape 1 : Identifier les champs

Etape 2 : Sélectionner les tables dont on a besoin

- Dédoubler si nécessaire
- Transitivité permet de savoir quoi dédoubler



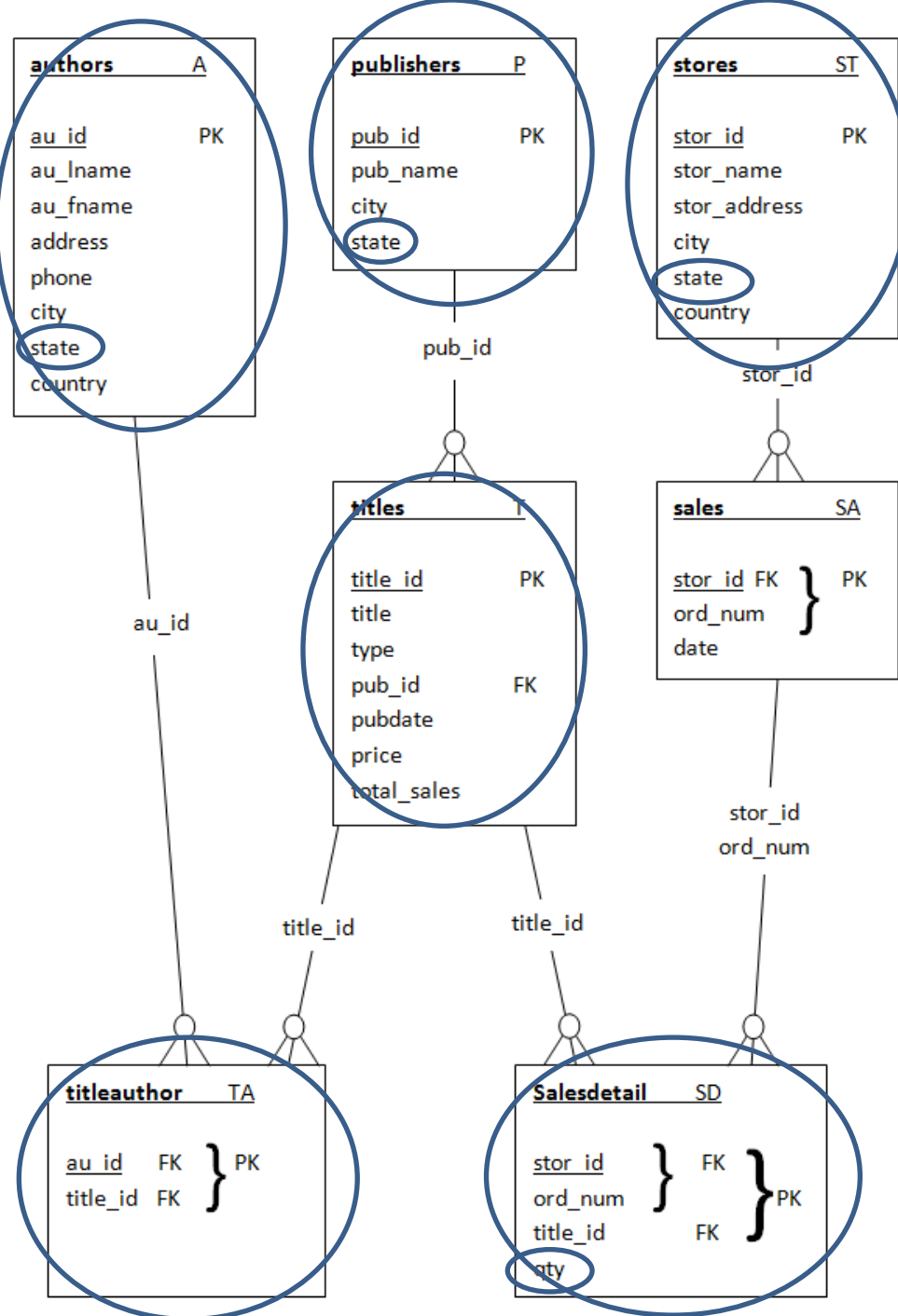
Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

Etape 1 : Identifier les champs

Etape 2 : Sélectionner les tables

Etape 3 : Simplifier

- Transitivité
- PK suffisante



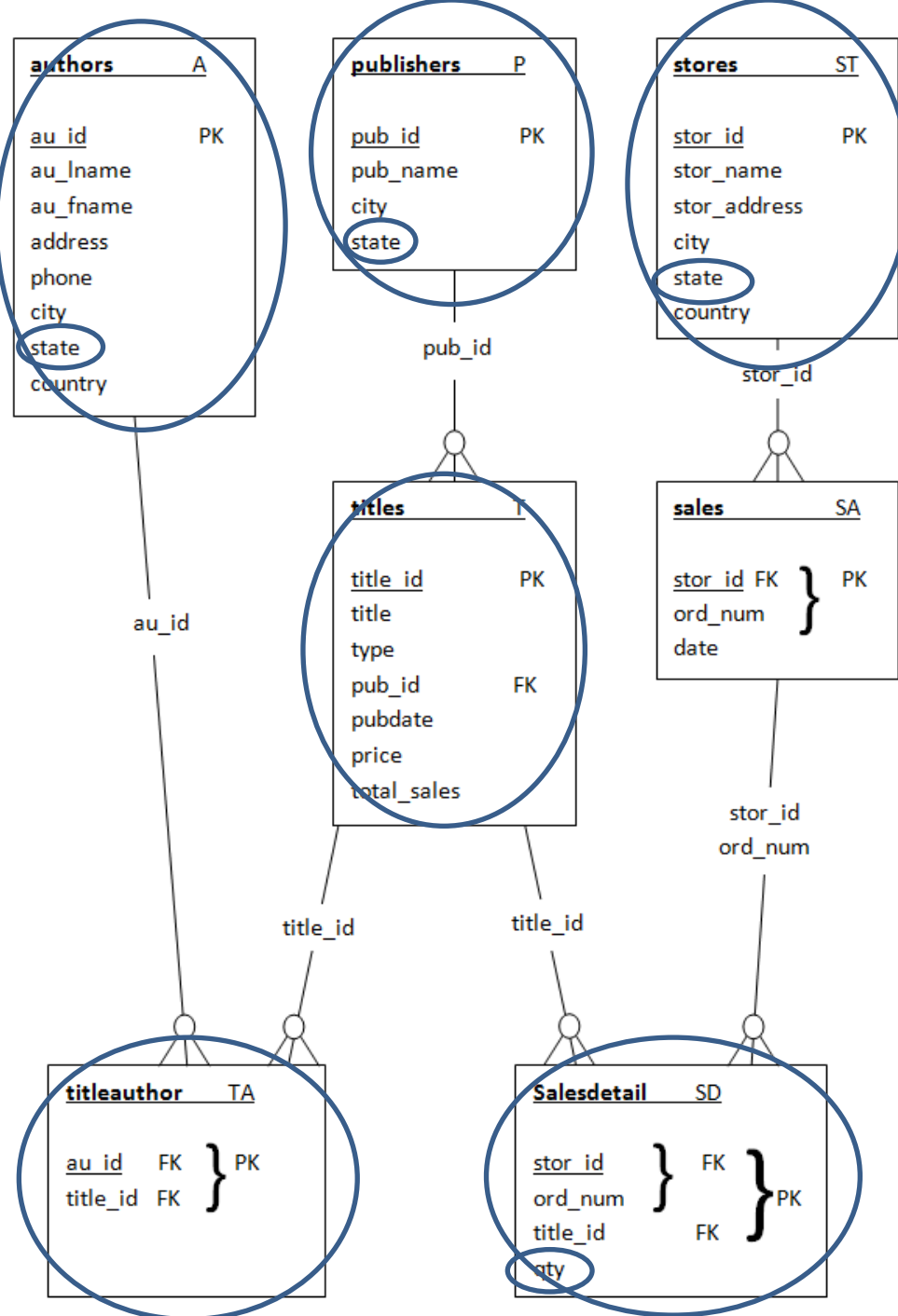
Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

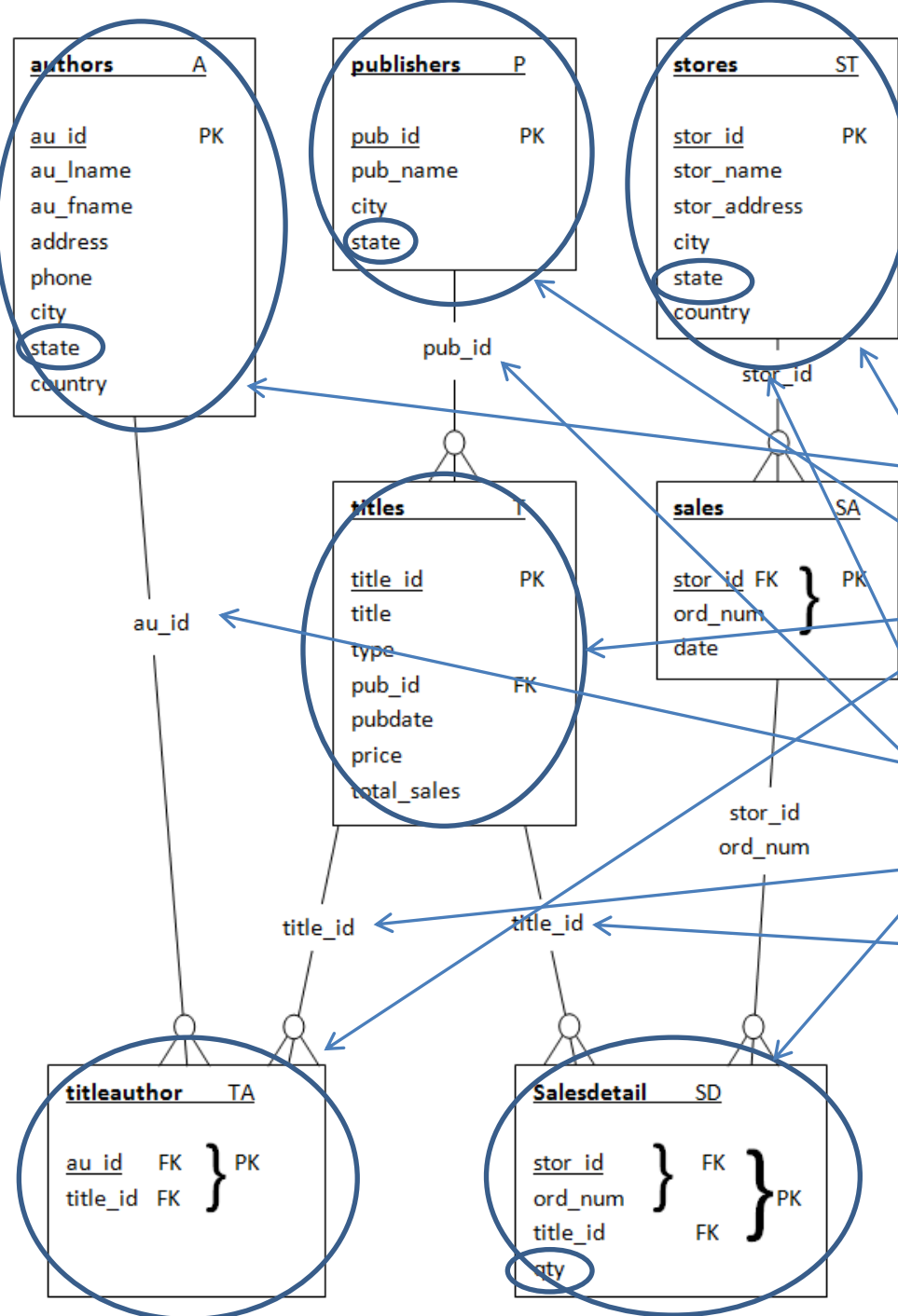
Etape 1 : Identifier les champs

Etape 2 : Sélectionner les tables

Etape 3 : Simplifier

Etape 4 : Ecrire la requête





Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

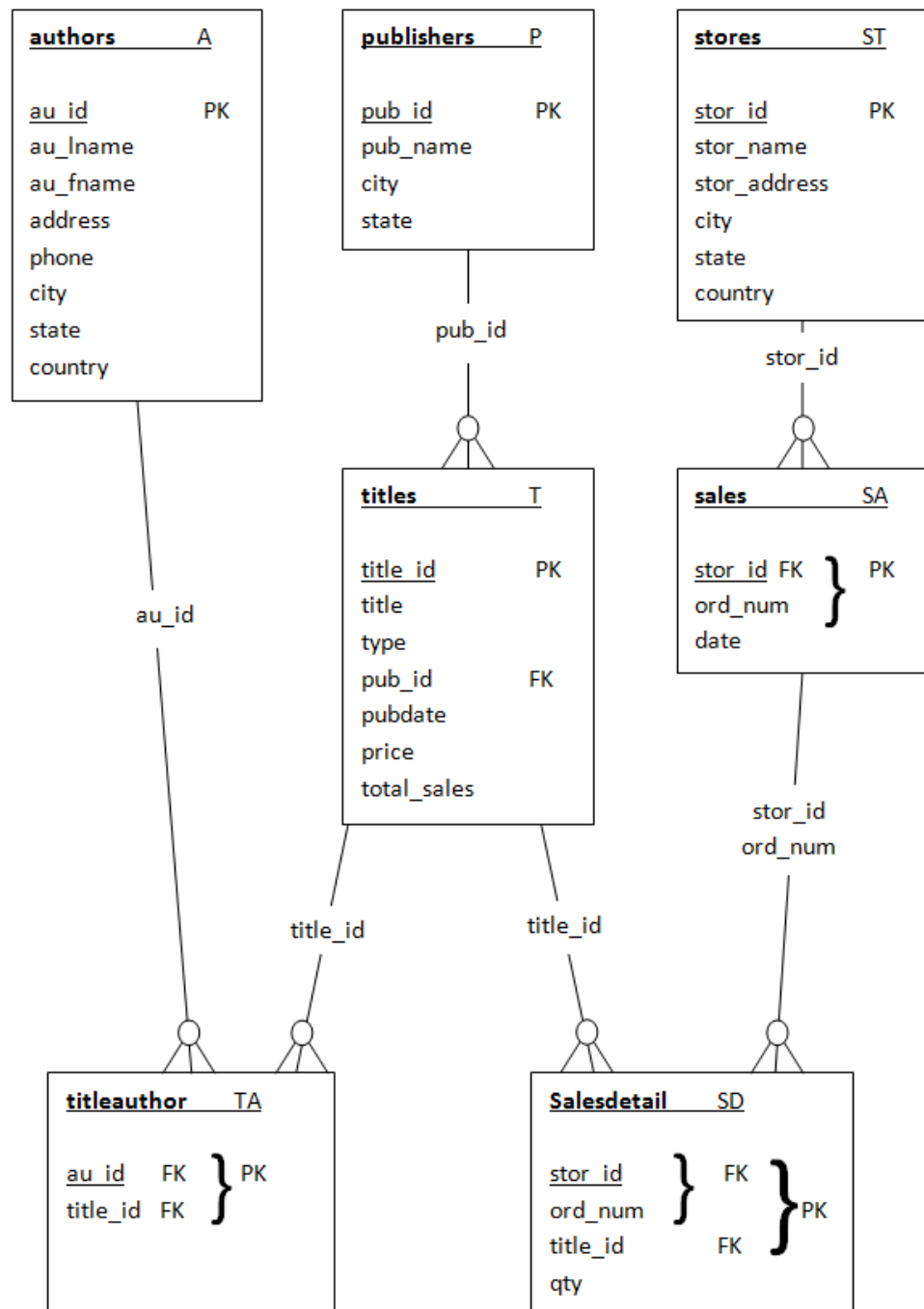
Etape 4 : Ecrire la requête

```
SELECT SUM(sd.qty)
FROM authors a,
titleauthor ta,
titles t,
publishers p,
salesdetail sd,
stores st
WHERE a.au_id=ta.au_id
AND ta.title_id=t.title_id
AND t.pub_id=p.pub_id
AND t.title_id=sd.title_id
AND sd.stor_id=st.stor_id
AND a.state='CA'
AND p.state='CA'
AND st.state='CA'
```

Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

Etape 4 : Ecrire la requête

```
SELECT SUM(sd.qty)
FROM authors a,
titleauthor ta,
titles t,
publishers p,
salesdetail sd,
stores st
WHERE a.au_id=ta.au_id
AND ta.title_id=t.title_id
AND t.pub_id=p.pub_id
AND t.title_id=sd.title_id
AND sd.stor_id=st.stor_id
AND a.state='CA'
AND p.state='CA'
AND st.state='CA'
```



Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

Etape 1 : Identifier les champs

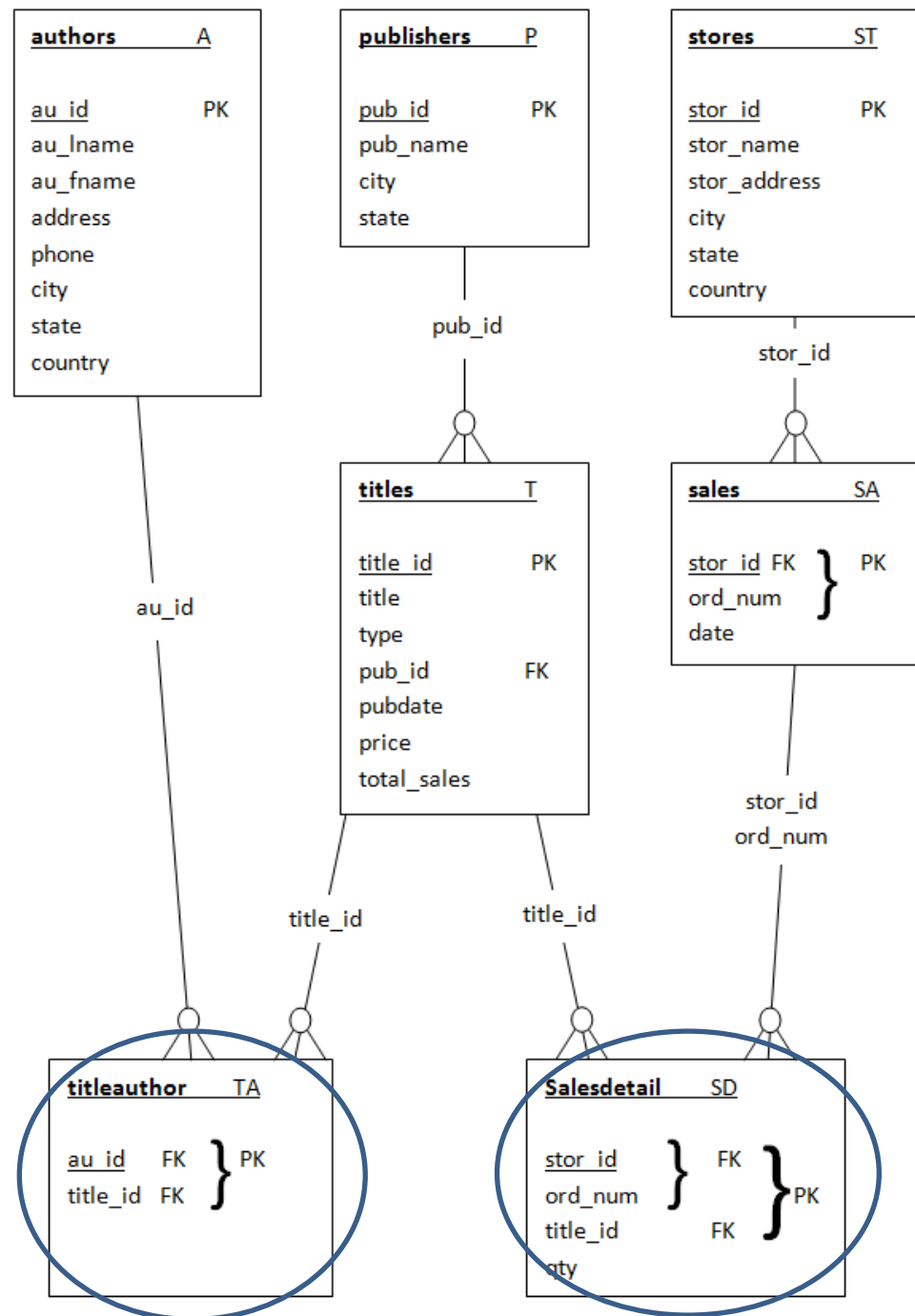
Etape 2 : Sélectionner les tables

Etape 3 : Simplifier

Etape 4 : Ecrire la requête

Etape 5 : Doublons ?

- Jointures avec relations multiples
- DISTINCT ?
- Ici : bug

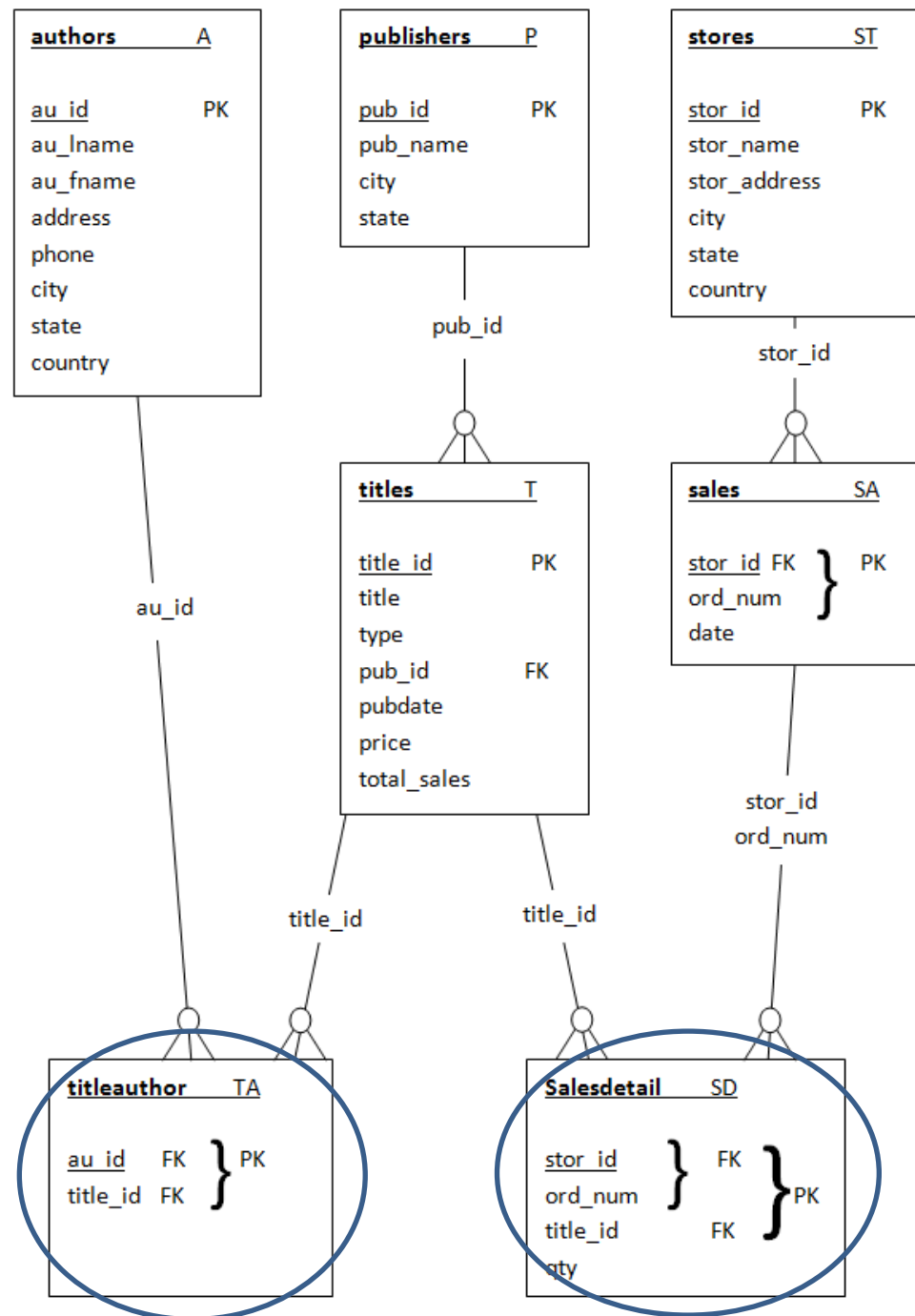


Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

Etape 5 : Doublons ?

Solution : ne pas joindre via
titleauthor

A la place : sous-select

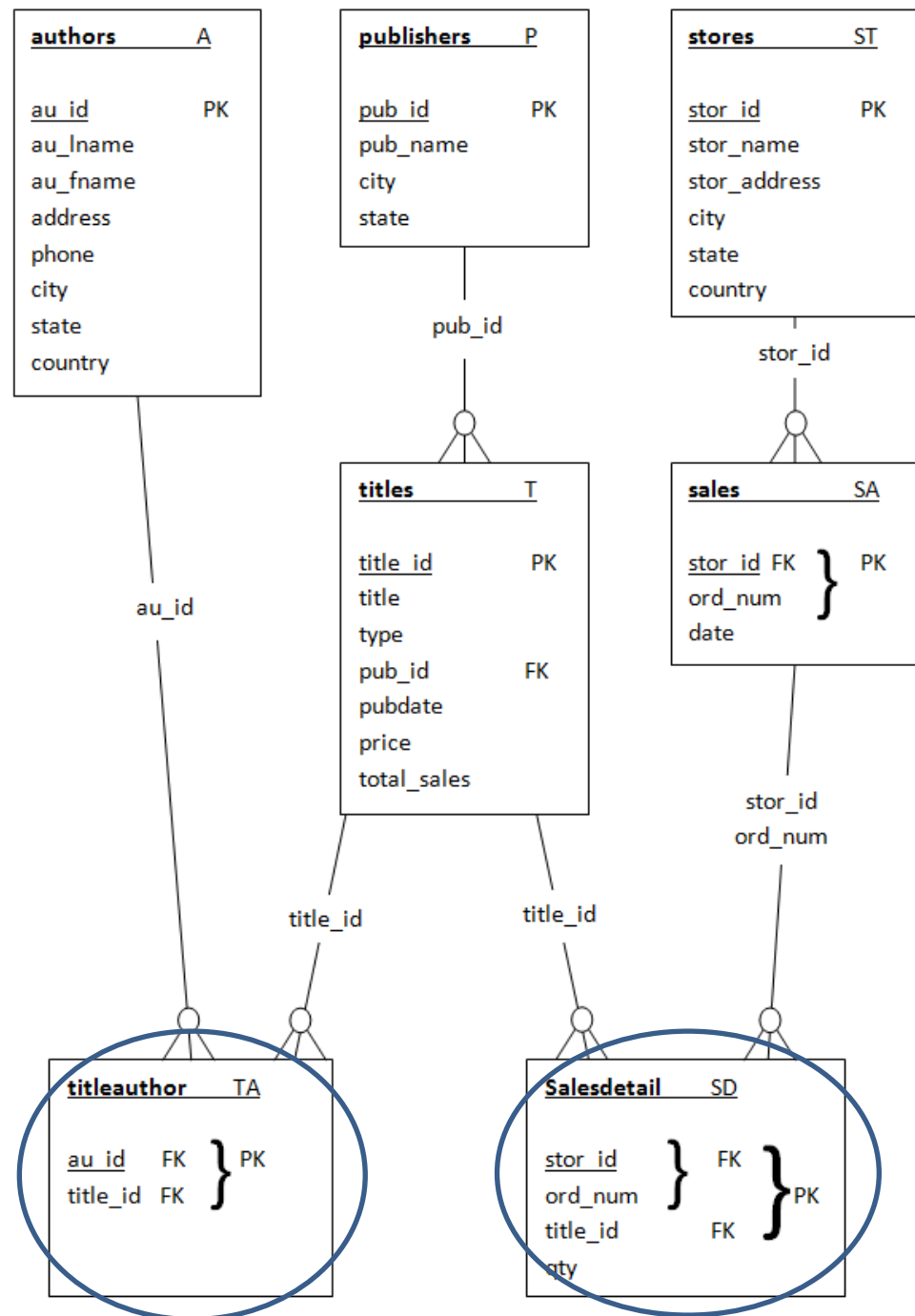


Somme des livres d'auteurs californiens & éditeurs californiens & vendus en Californie

```

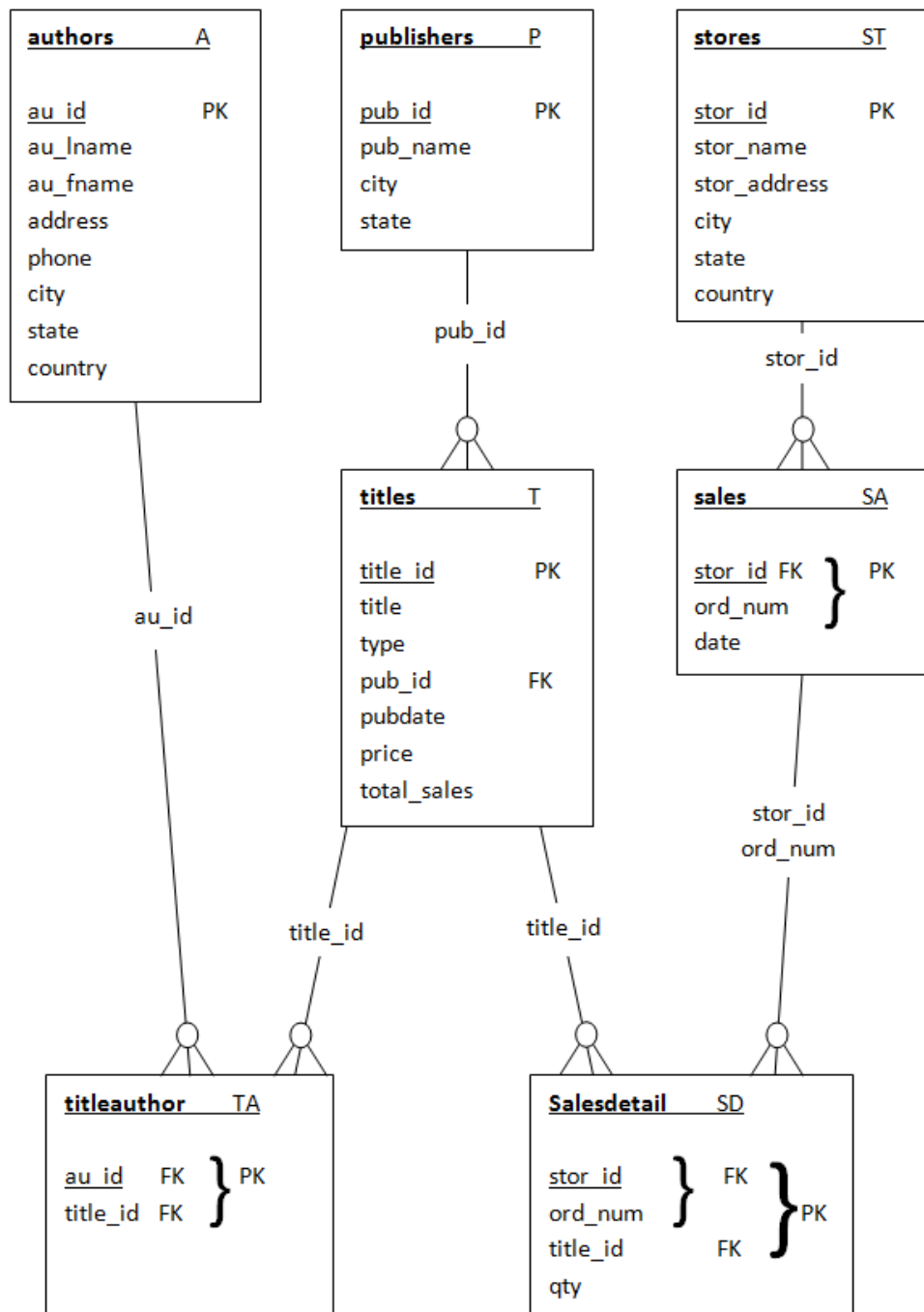
SELECT SUM(sd.qty)
FROM   titles t,
        publishers p,
        salesdetail sd,
        stores st
WHERE  t.pub_id=p.pub_id
AND    t.title_id=sd.title_id
AND    sd.stor_id=st.stor_id
AND    p.state='CA'
AND    st.state='CA'
AND    t.title_id IN (
        SELECT ta.title_id
        FROM authors a,
             titleauthor ta
        WHERE a.au_id=ta.au_id
              AND a.state='CA'
    )

```



Les magasins qui vendent
tous les livres édités par
Algodata ?

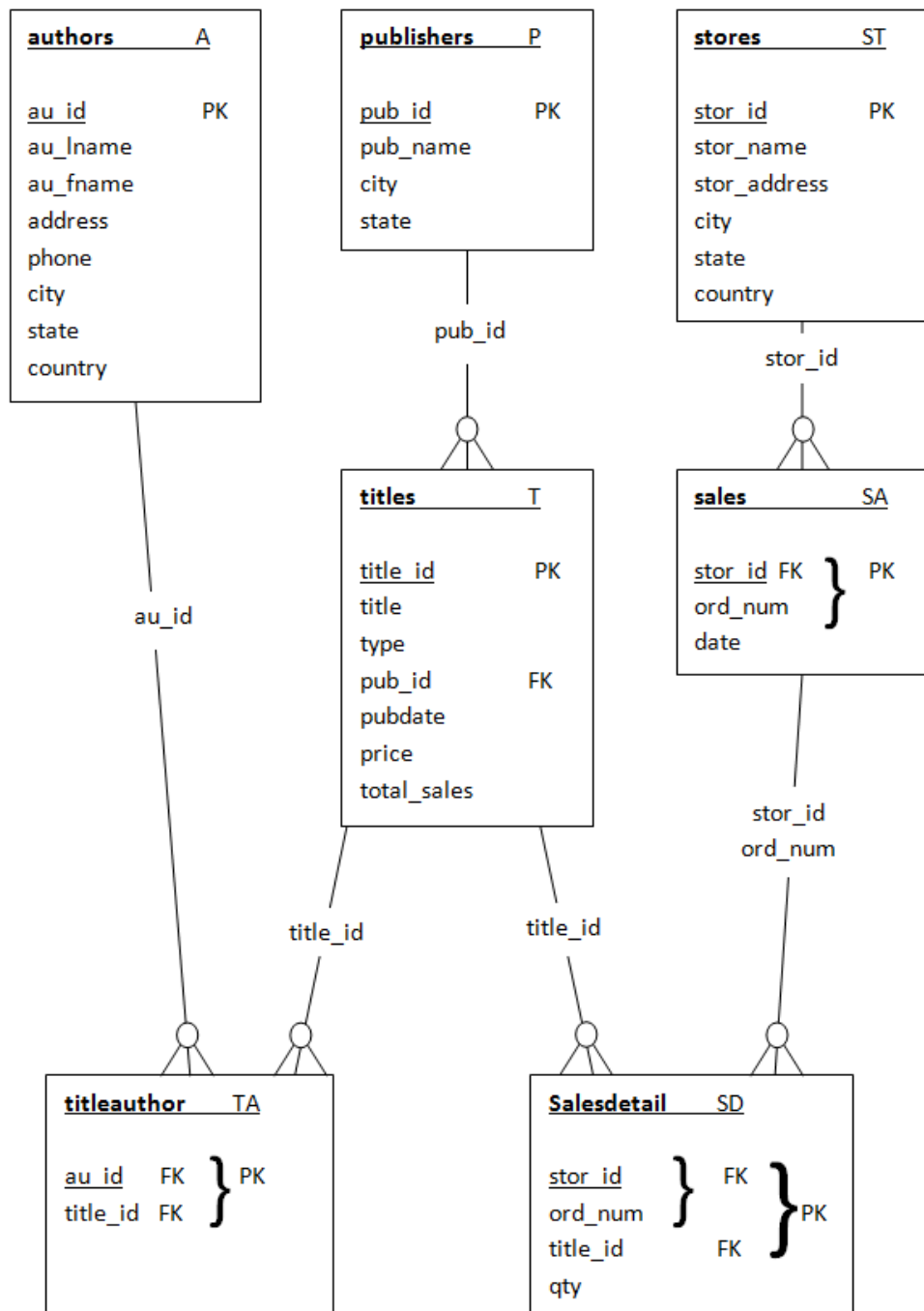
On ne peut pas traduire
directement en SQL !



Les magasins qui vendent
tous les livres édités par
Algodata ?

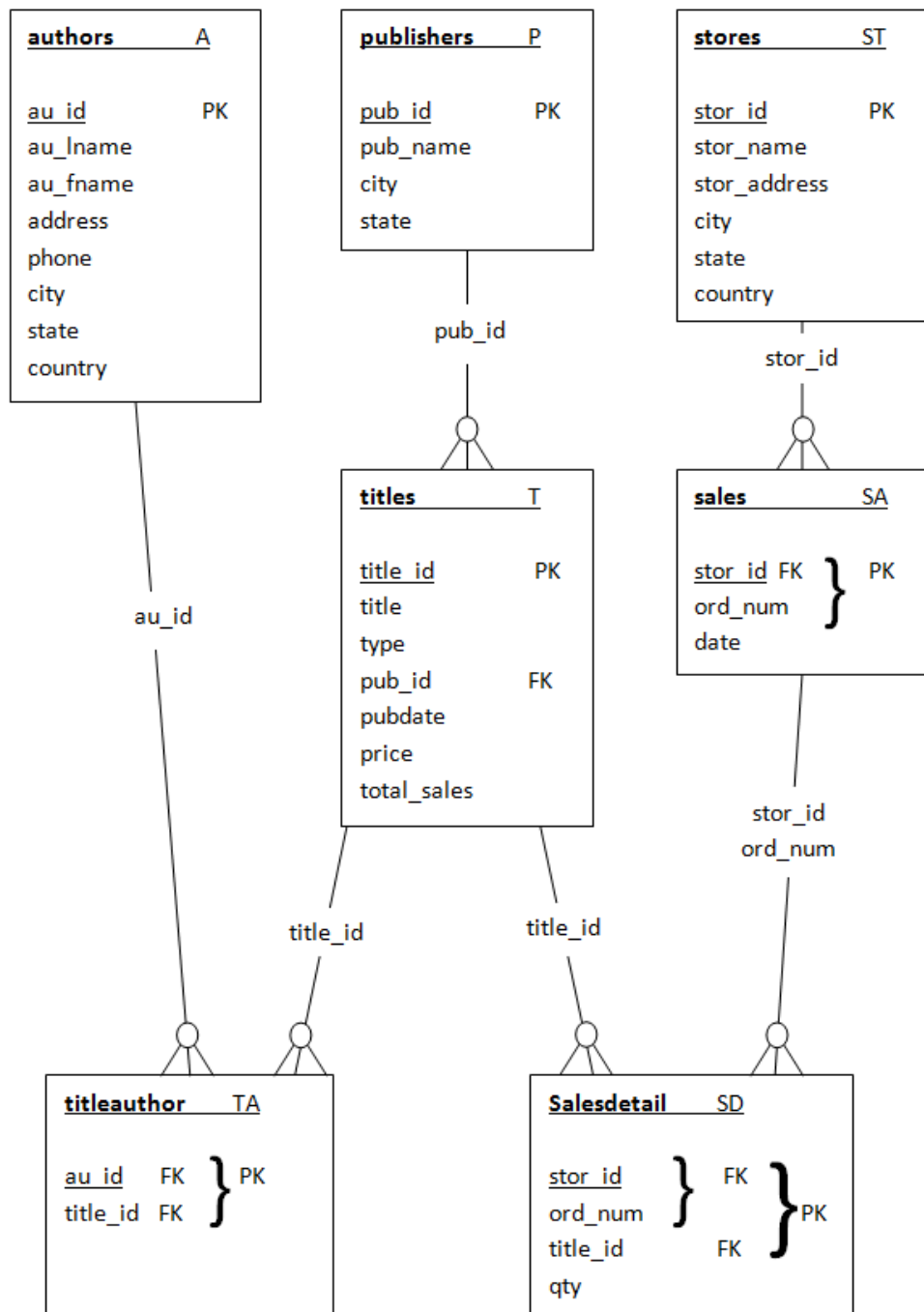
Reformulation :

Les magasins tels qu'il
n'existe pas de livre
édité par Algodata qui
n'y soit pas vendu



Les magasins tels qu'il
n'existe pas de livre édité
par Algodata qui n'y soit
pas vendu

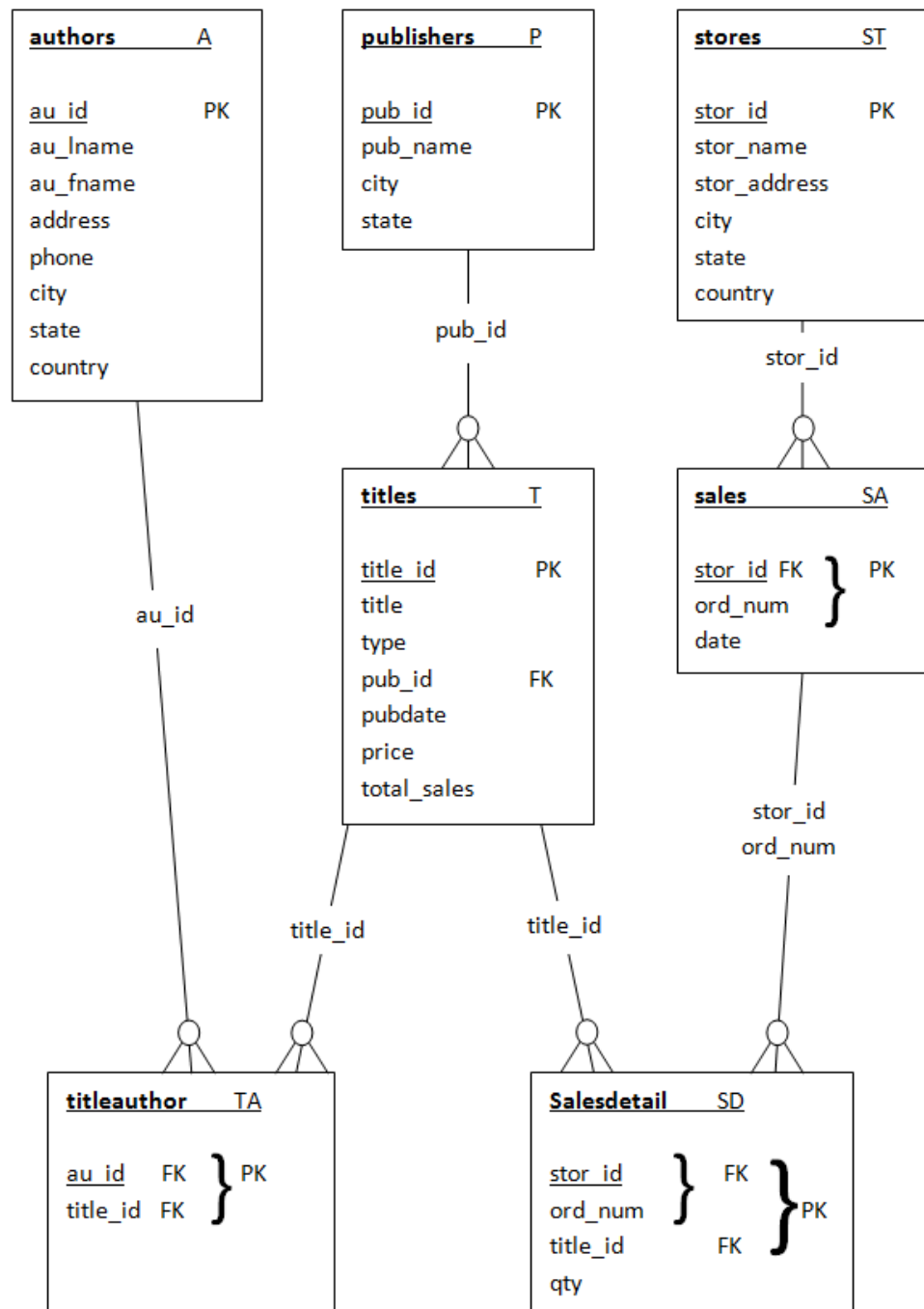
```
SELECT st.stor_id, st.stor_name
FROM stores st
WHERE NOT EXISTS (
    SELECT *
    FROM titles t, publishers p
    WHERE t.pub_id = p.pub_id
    AND p.pub_name LIKE 'Algodata%'
    AND NOT EXISTS (
        SELECT *
        FROM salesdetail sd
        WHERE sd.title_id = t.title_id
        AND sd.stor_id = st.stor_id ));
```



Les magasins qui vendent
tous les livres édités par
Algodata ?

Autre reformulation :

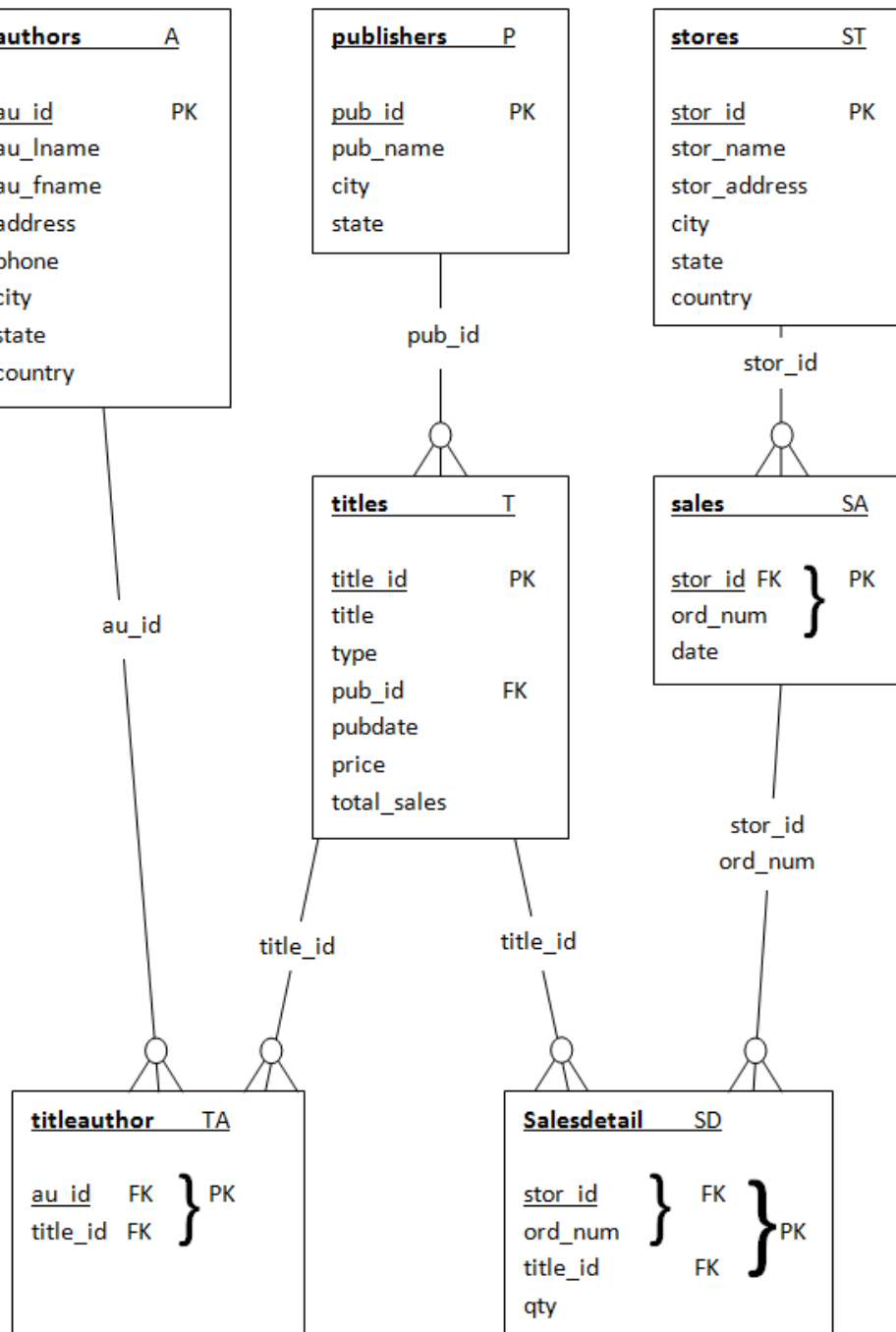
Les magasins tels que le
nombre de livres édités
par Algodata qu'ils
vendent est le même
que le nombre de livre
édité par Algodata



Les magasins qui vendent tous les livres édités par Algodata ?

```

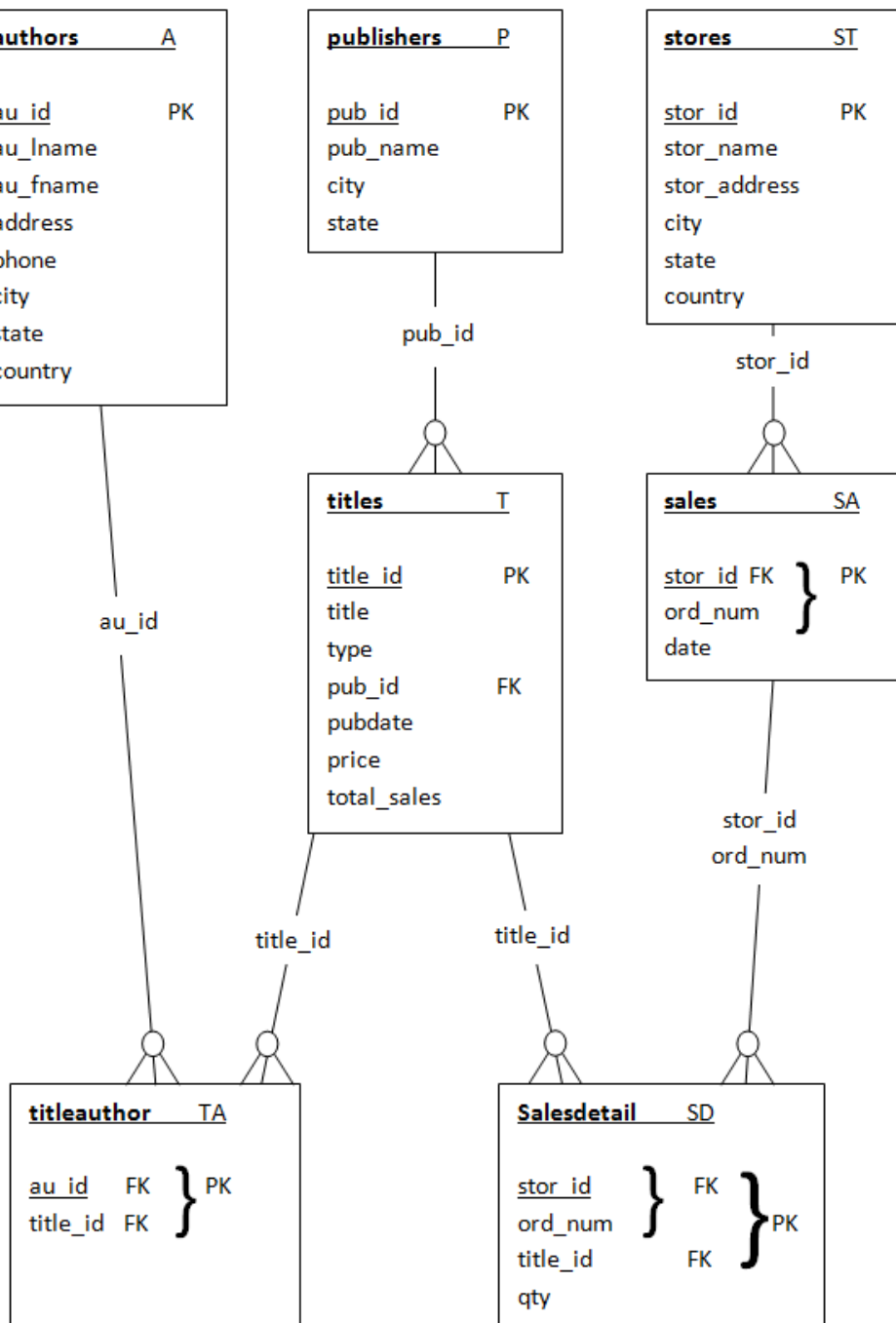
SELECT st.stor_name
FROM   stores st,
       titles t,
       publishers p,
       salesdetail sd
WHERE  st.stor_id=sd.stor_id
AND    sd.title_id=t.title_id
AND    t.pub_id=p.pub_id
AND    p.pub_name LIKE 'Algodata%'
GROUP BY st.stor_id
HAVING COUNT(DISTINCT t.title_id) =
        (SELECT COUNT(*)
         FROM   titles t, publishers p
         WHERE  p.pub_id=t.pub_id
         AND    p.pub_name LIKE 'Algodata%')
    
```



Les magasins qui vendent tous les livres édités par Algodata ?

```

SELECT st.stor_name
FROM   stores st
WHERE  (SELECT COUNT(DISTINCT t.title_id)
        FROM   titles t,
              publishers p,
              salesdetail sd
        WHERE  st.stor_id=sd.stor_id
        AND    sd.title_id=t.title_id
        AND    t.pub_id=p.pub_id
        AND    p.pub_name LIKE 'Algodata%')=
        (SELECT COUNT(*)
        FROM   titles t, publishers p
        WHERE  p.pub_id=t.pub_id
        AND    p.pub_name LIKE 'Algodata%')
    
```



Mais quel est la requête la plus performante ?

- Cela dépend de la manière dont PostgreSQL les réalisera.
 - Spécifique à son implémentation.
 - Très difficile de deviner ce qui sera réellement fait et le temps nécessaire à l'accomplir.
 - Autant demander à PostgreSQL lui-même !
 - Menu Query/Explain (F7)

Query - pubs2 on nedar@10.0.0.45:5432 *

File Edit Query Favourites Macros View Help

pubs2 on nedar@10.0.0.45:5432

SQL Editor Graphical Query Builder

```
SELECT st.stor_id, st.stor_name
FROM stores st
WHERE NOT EXISTS (
  SELECT *
  FROM titles t, publishers p
  WHERE t.pub_id = p.pub_id
  AND p.pub_name LIKE 'Algodata%'
  AND NOT EXISTS (
    SELECT *
    FROM salesdetail sd
    WHERE sd.title_id = t.title_id
    AND sd.stor_id = st.stor_id ));
```

Scratch pad

Output pane

Data Output Explain Messages History

The diagram illustrates the execution plan for the provided SQL query. It starts with three input tables: 'titles', 'publishers', and 'stores'. The 'titles' table is processed by a 'Hash' operator. The 'publishers' table is processed by a 'Hash Join' operator, which also receives input from the 'Hash' operator. The output of the 'Hash Join' is then processed by a 'Materialize' operator. The 'stores' table is processed by a 'Nested Loop Anti Join' operator, which also receives input from the 'Materialize' operator. The final output is a 'Nested Loop Anti Join' operator, which also receives input from the 'Materialize' operator. The diagram shows the flow of data from the input tables through the operators to the final output.

OK.

Unix Ln 13 Col 1 Ch 290 15 rows. 21 ms

Query - pubs2 on nediari@10.0.0.45:5432 *

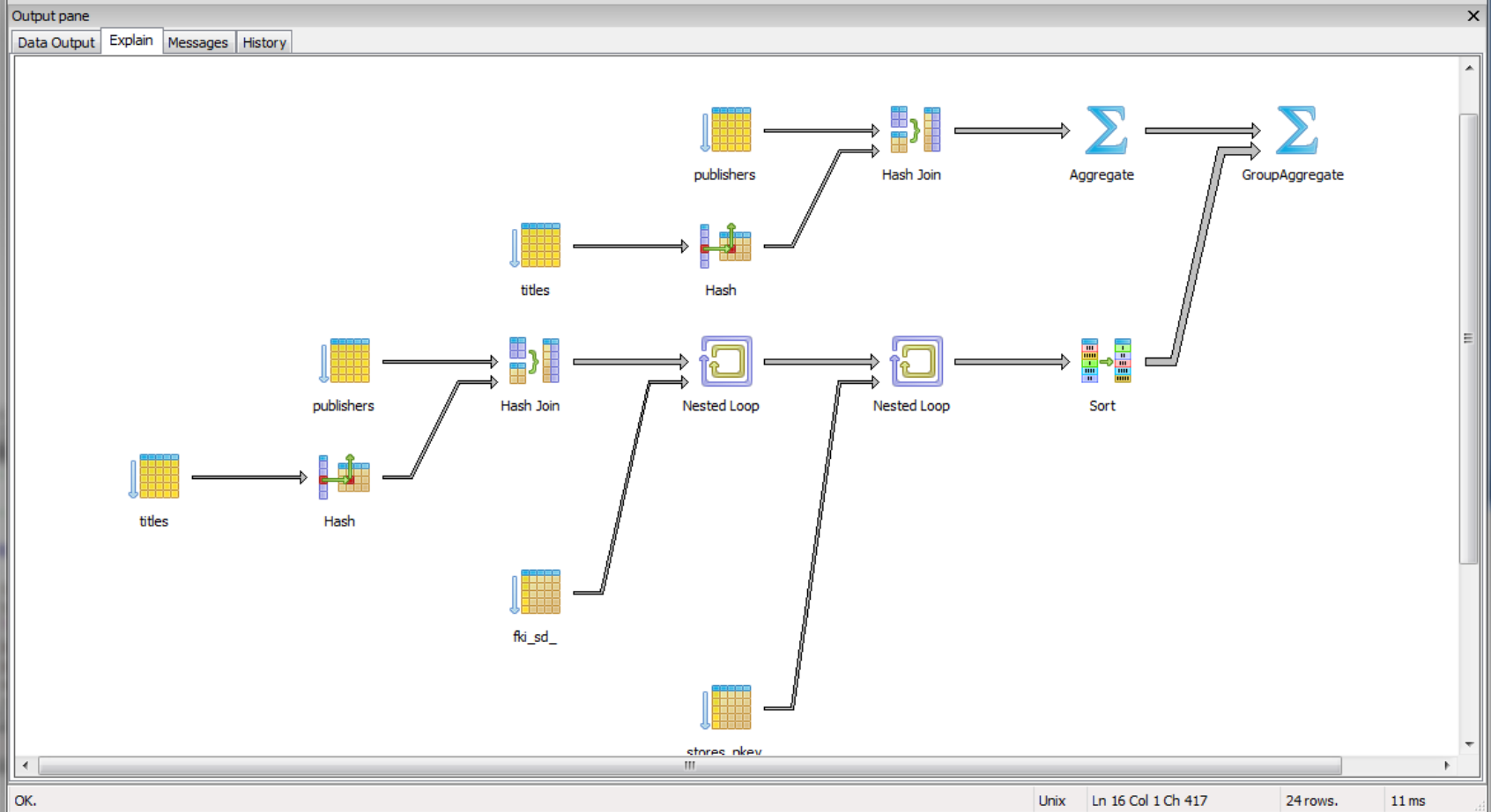
File Edit Query Favgrites Macros View Help

pubs2 on nediari@10.0.0.45:5432

SQL Editor Graphical Query Builder

```
SELECT st.stor_name
FROM   stores st,
       titles t,
       publishers p,
       salesdetail sd
WHERE  st.stor_id=sd.stor_id
AND    sd.title_id=t.title_id
AND    t.pub_id=p.pub_id
AND    p.pub_name LIKE 'Algodata%'
GROUP BY st.stor_id, st.stor_name
HAVING COUNT(DISTINCT t.title_id) =
        (SELECT COUNT(*)
         FROM   titles t, publishers p
         WHERE  p.pub_id=t.pub_id
         AND    p.pub_name LIKE 'Algodata%')
```

Scratch pad



Query - pubs2 on nediard@10.0.0.45:5432 *

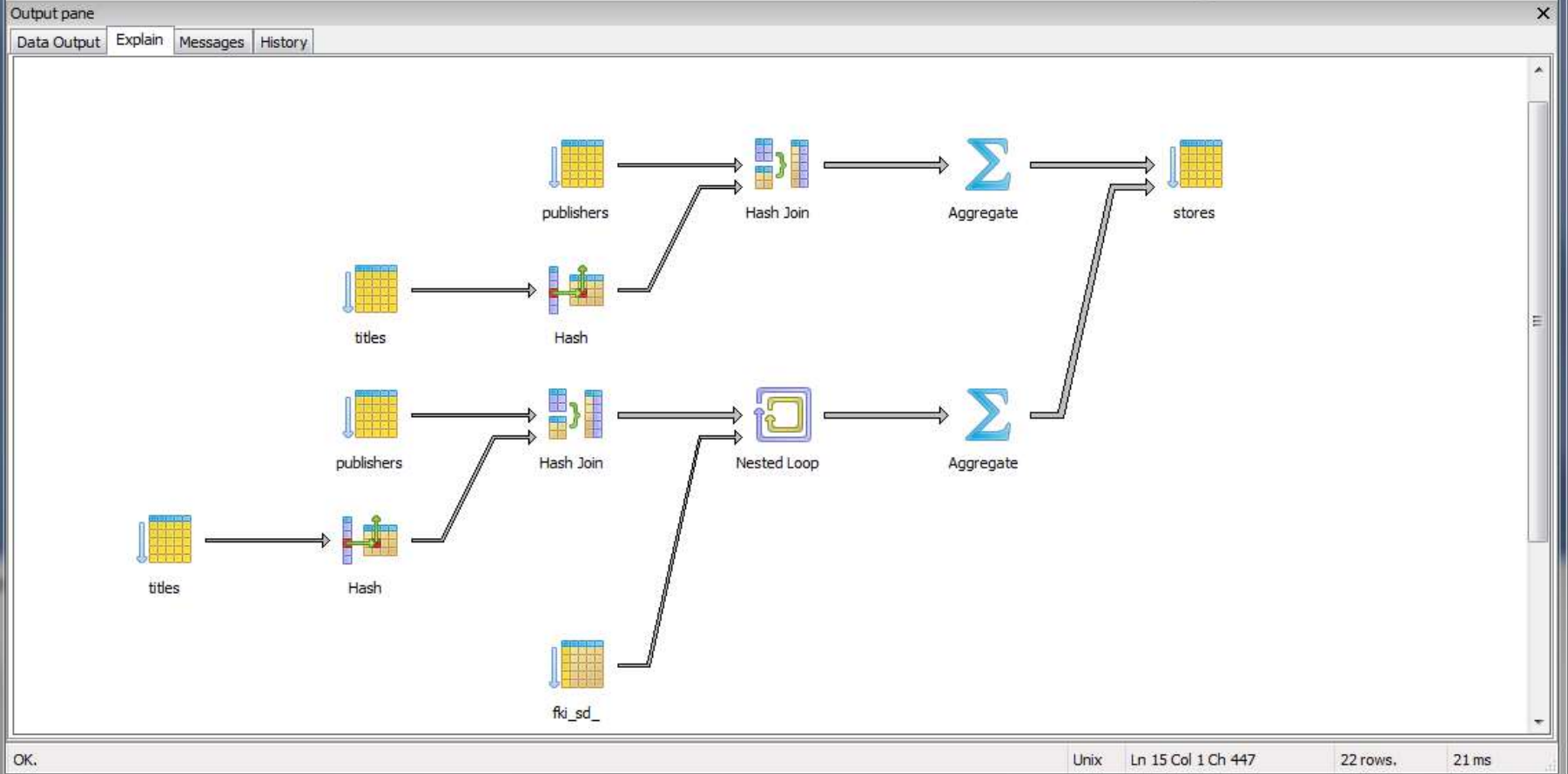
File Edit Query Favgrites Macros View Help

pubs2 on nediard@10.0.0.45:5432

SQL Editor Graphical Query Builder

```
SELECT st.stor_name
FROM stores st
WHERE (SELECT COUNT(DISTINCT t.title_id)
FROM titles t,
publishers p,
salesdetail sd
WHERE st.stor_id=sd.stor_id
AND sd.title_id=t.title_id
AND t.pub_id=p.pub_id
AND p.pub_name LIKE 'Algodata%')=
(SELECT COUNT(*)
FROM titles t, publishers p
WHERE p.pub_id=t.pub_id
AND p.pub_name LIKE 'Algodata%')
```

Scratch pad



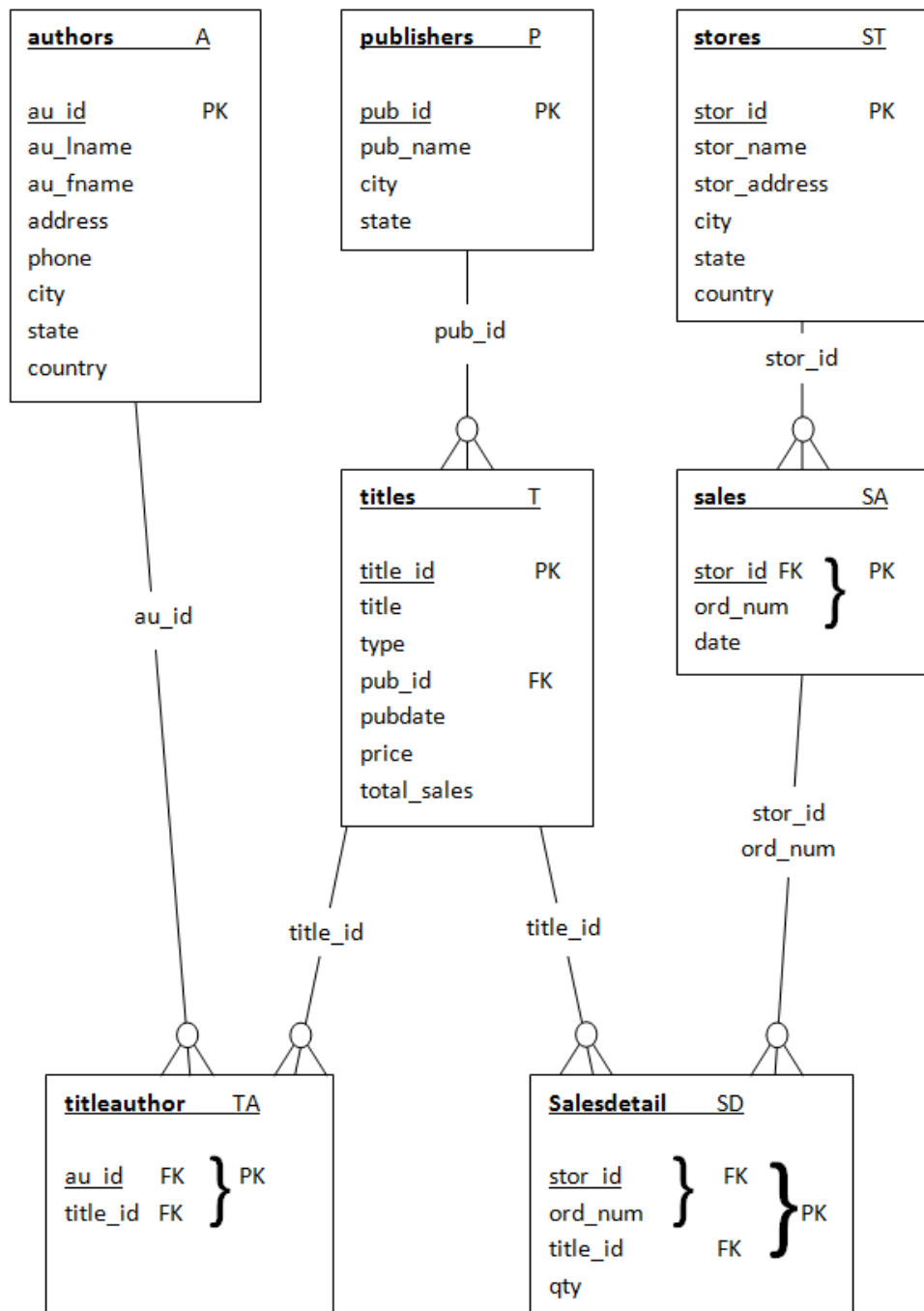
Les coûts sont dans l'onglet Data Output

- Deux nombres sont affichés
 - Le temps de démarrage, écoulé avant que la première ligne soit renvoyée.
 - Le temps d'exécution total, nécessaire au renvoi de toutes les lignes.
- 1. Double négation : 16.06 / 846.86
- 2. GROUP BY : 43.74 / 44.07
- 3. 2 sous SELECT : 16.08 / 3306.75
- Conclusion : group by est beaucoup plus efficace qu'un sous-select !

Quel est l'éditeur qui a
publié le plus grand
nombre de livres ?

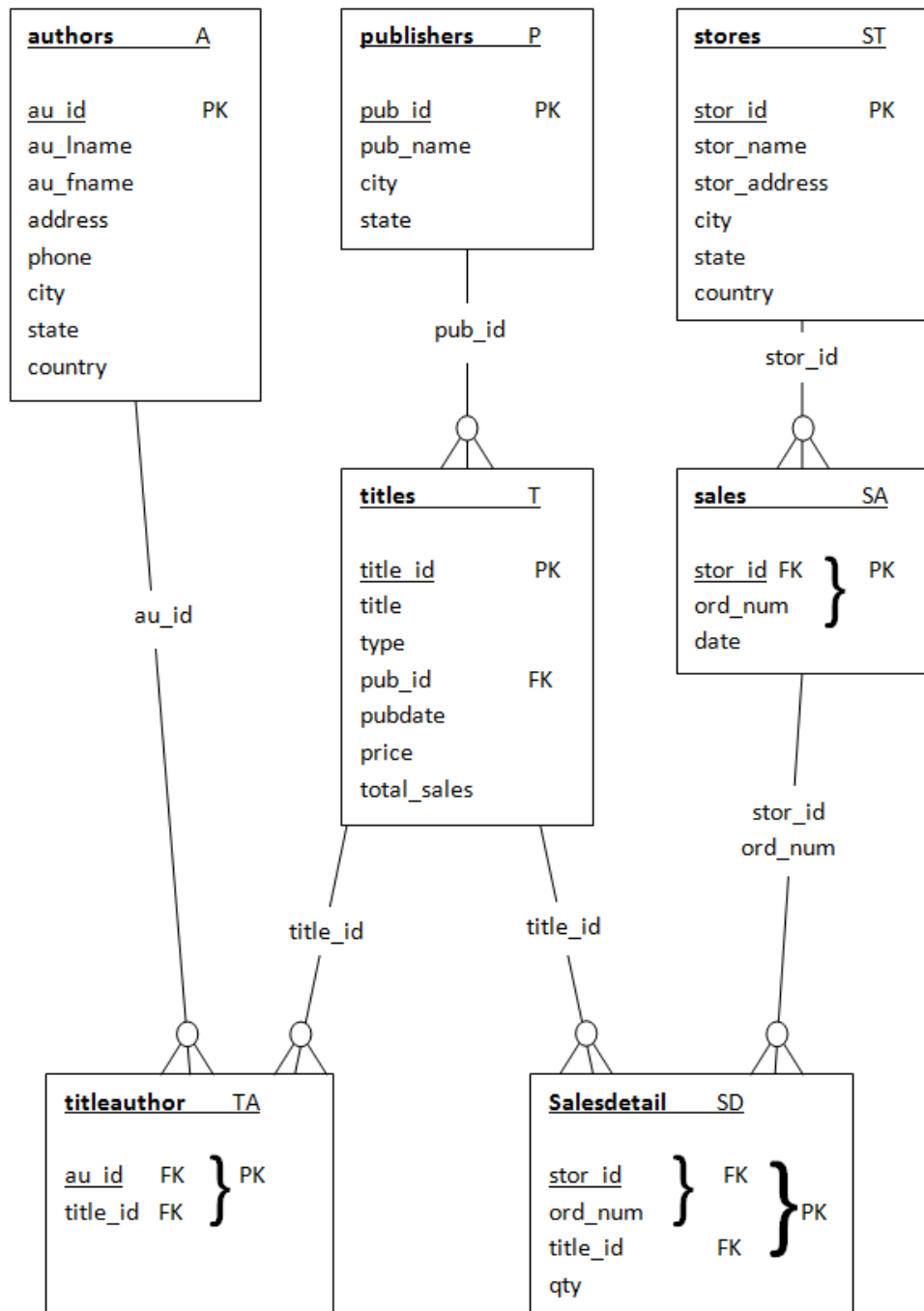
```
SELECT p.pub_name
FROM publishers p, titles t
WHERE p.pub_id=t.pub_id
GROUP BY p.pub_id
HAVING COUNT(*)=(
    SELECT MAX(COUNT(*))
    FROM publishers p, titles t
    WHERE p.pub_id=t.pub_id
    GROUP BY p.pub_id)
```

On ne peut pas faire MAX(COUNT(*)) !



Quel est l'éditeur qui a
publié le plus grand
nombre de livres ?

```
SELECT p.pub_name
FROM publishers p, titles t
WHERE p.pub_id=t.pub_id
GROUP BY p.pub_id
HAVING COUNT(*) >= ALL (
    SELECT COUNT(*)
    FROM publishers p, titles t
    WHERE p.pub_id=t.pub_id
    GROUP BY p.pub_id)
```



Outer Join

Dans la partie FROM d'un SELECT

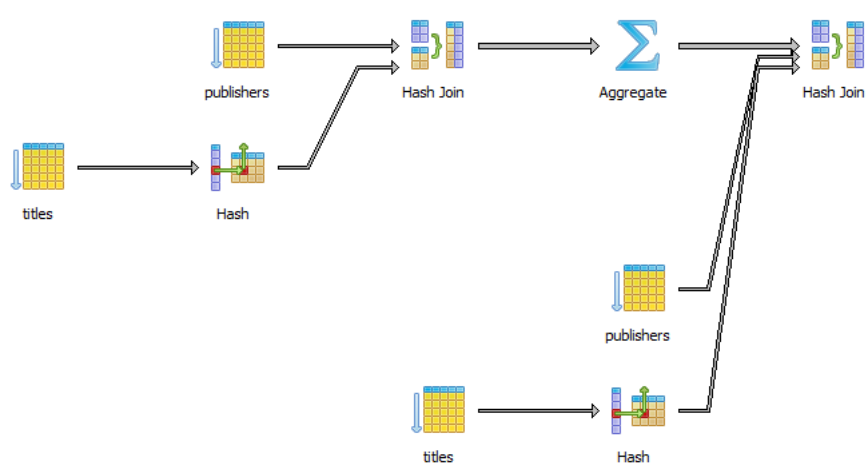
- tableGauche INNER JOIN tableDroite ON condition
=> FROM tableGauche, tableDroite WHERE condition
- tableGauche LEFT OUTER JOIN tableDroite ON condition
retourne tous les champs de tableGauche, joint avec tableDroite quand c'est possible, null au sinon
- tableGauche RIGHT OUTER JOIN tableDroite ON champ
retourne tous les champs de tableDroite, joint avec tableGauche quand c'est possible, null au sinon
- tableGauche FULL OUTER JOIN tableDroite ON champ
retourne tous les champs de tableDroite et de tableGauche, joint avec l'autre table quand c'est possible, null au sinon

INNER JOIN

```

SELECT t.title
FROM   titles t, publishers p
WHERE  t.pub_id=p.pub_id
AND    p.pub_name='New Age Books'
AND    t.price >= (
    SELECT MAX(t.price)
    FROM   titles t, publishers p
    WHERE  t.pub_id=p.pub_id
    AND    p.pub_name='New Age Books'
)

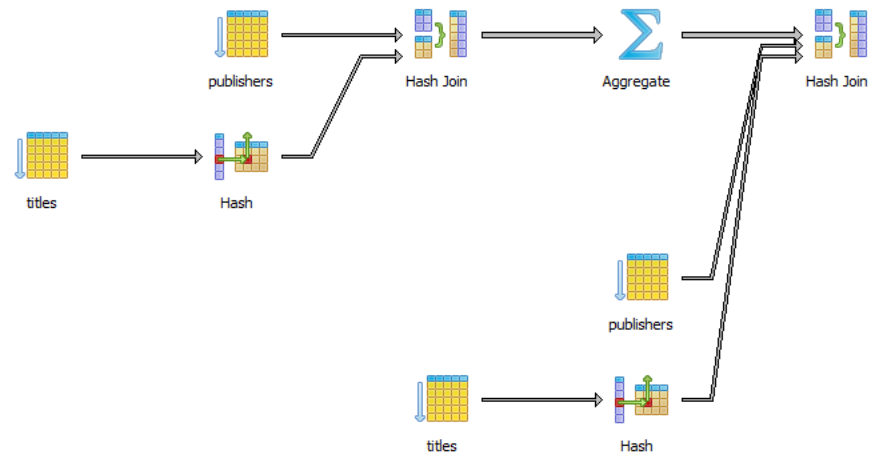
```



```

SELECT t.title
FROM   titles t INNER JOIN publishers p
ON     t.pub_id=p.pub_id
WHERE  p.pub_name='New Age Books'
AND    t.price >= (
    SELECT MAX(t.price)
    FROM   titles t INNER JOIN
    publishers p
    ON     t.pub_id=p.pub_id
    WHERE  p.pub_name='New Age Books'
)

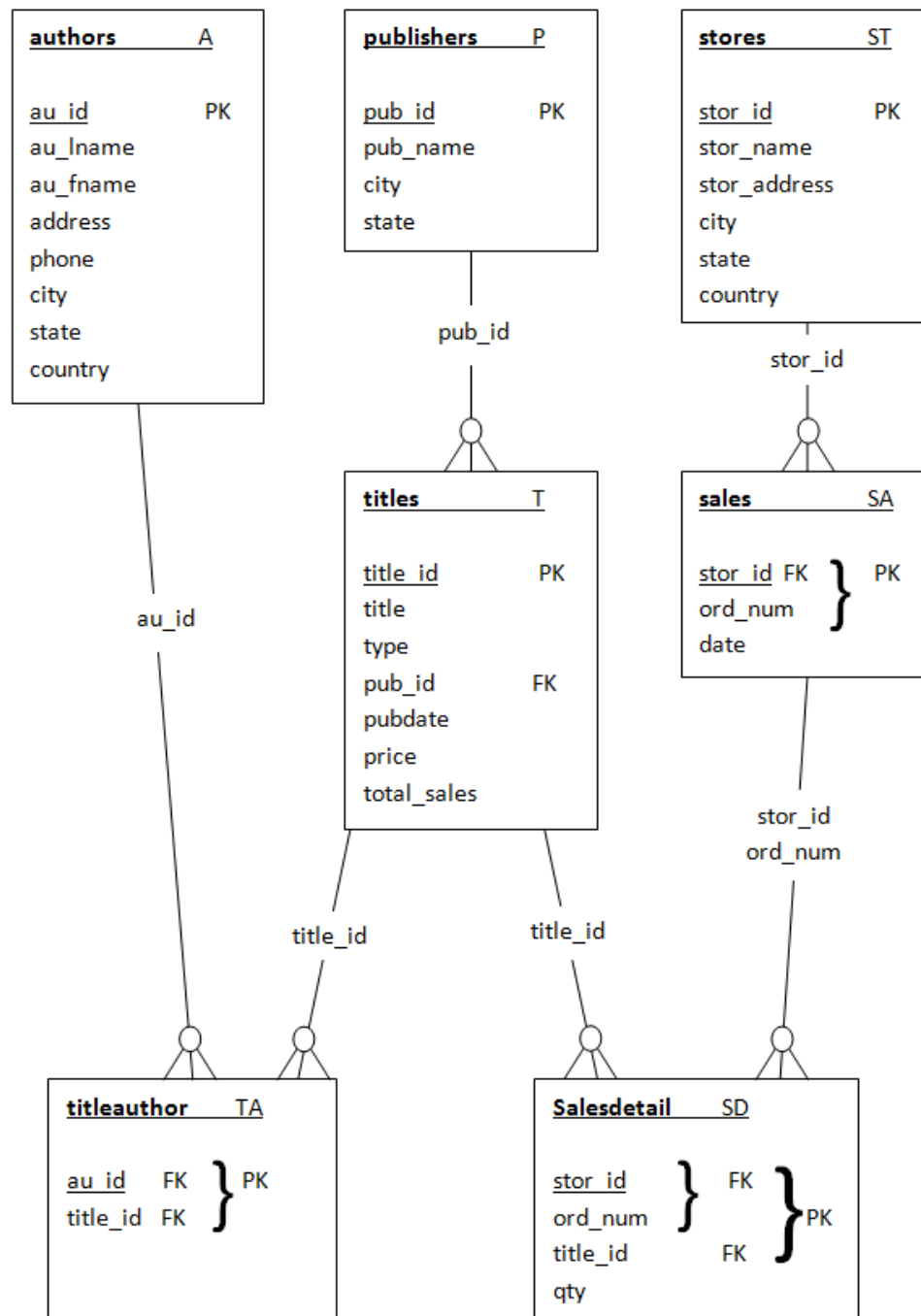
```



Tous les livres avec leurs quantités vendues

```
SELECT t.title, SUM(sd.qty)
FROM titles t, salesdetail sd
WHERE t.title_id=sd.title_id
GROUP BY t.title_id;
```

	title character varying(80)	sum bigint
1	Sushi, Anyone?	4095
2	Cooking with Computers: Surreptitious Balance Sheets	3876
3	Computer Phobic and Non-Phobic Individuals: Behavior Variations	375
4	The Gourmet Microwave	22246
5	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean	375
6	Emotional Security: A New Algorithm	3336
7	Fifty Years in Buckingham Palace Kitchens	15096
8	You Can Combat Computer Stress!	15722
9	Straight Talk About Computers	4095
10	Prolonged Data Deprivation: Four Case Studies	4072
11	But Is It User Friendly?	8780
12	Secrets of Silicon Valley	2095
13	The Busy Executive's Database Guide	4095
14	Silicon Valley Gastronomic Treats	2032
15	Life Without Fear	111
16	Is Anger the Enemy?	2045



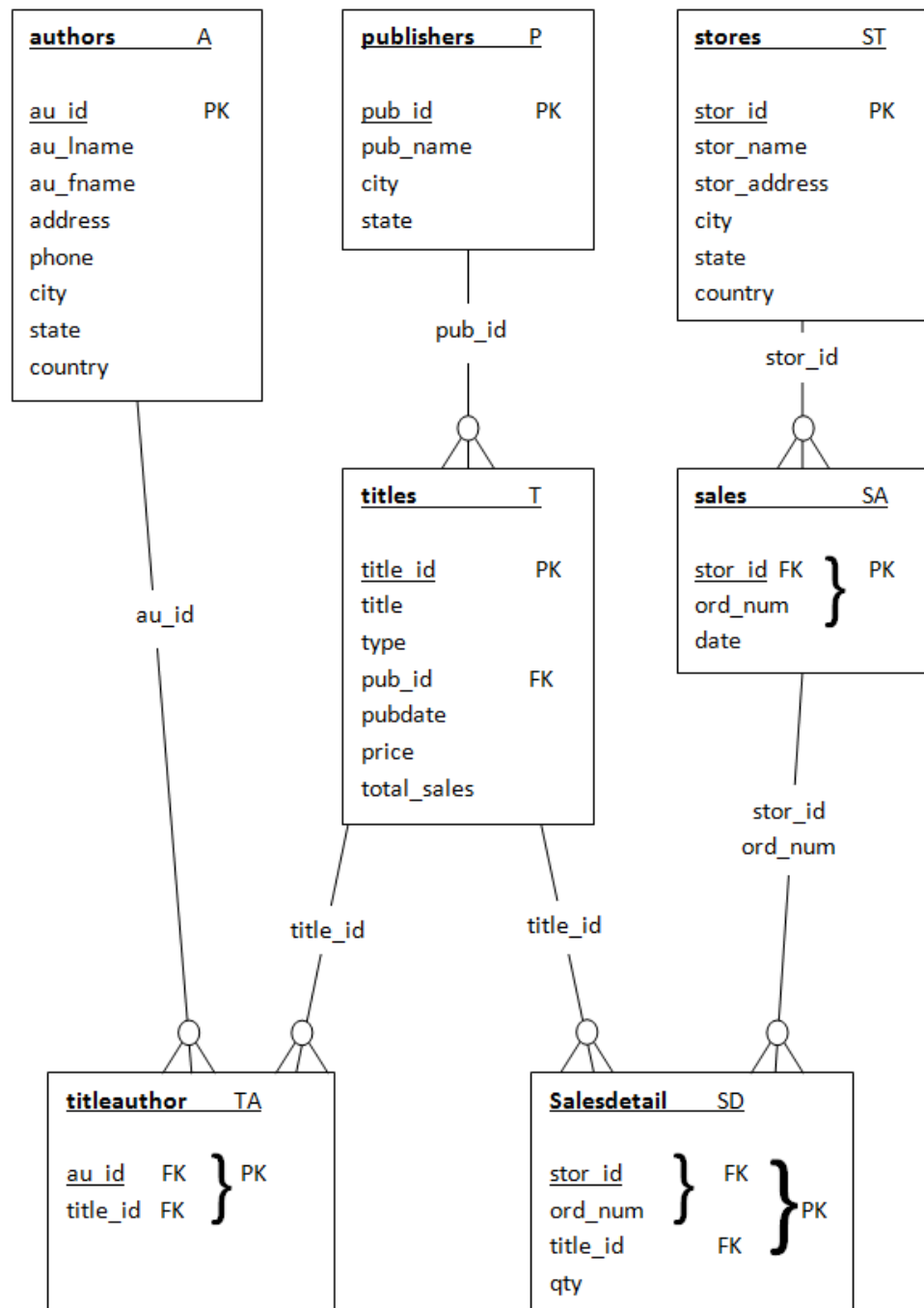
Tous les livres avec leurs quantités vendues

```
SELECT count(t.title_id)
FROM titles t;
```

Retourne 18 !

Où sont les 2 livres manquants ?

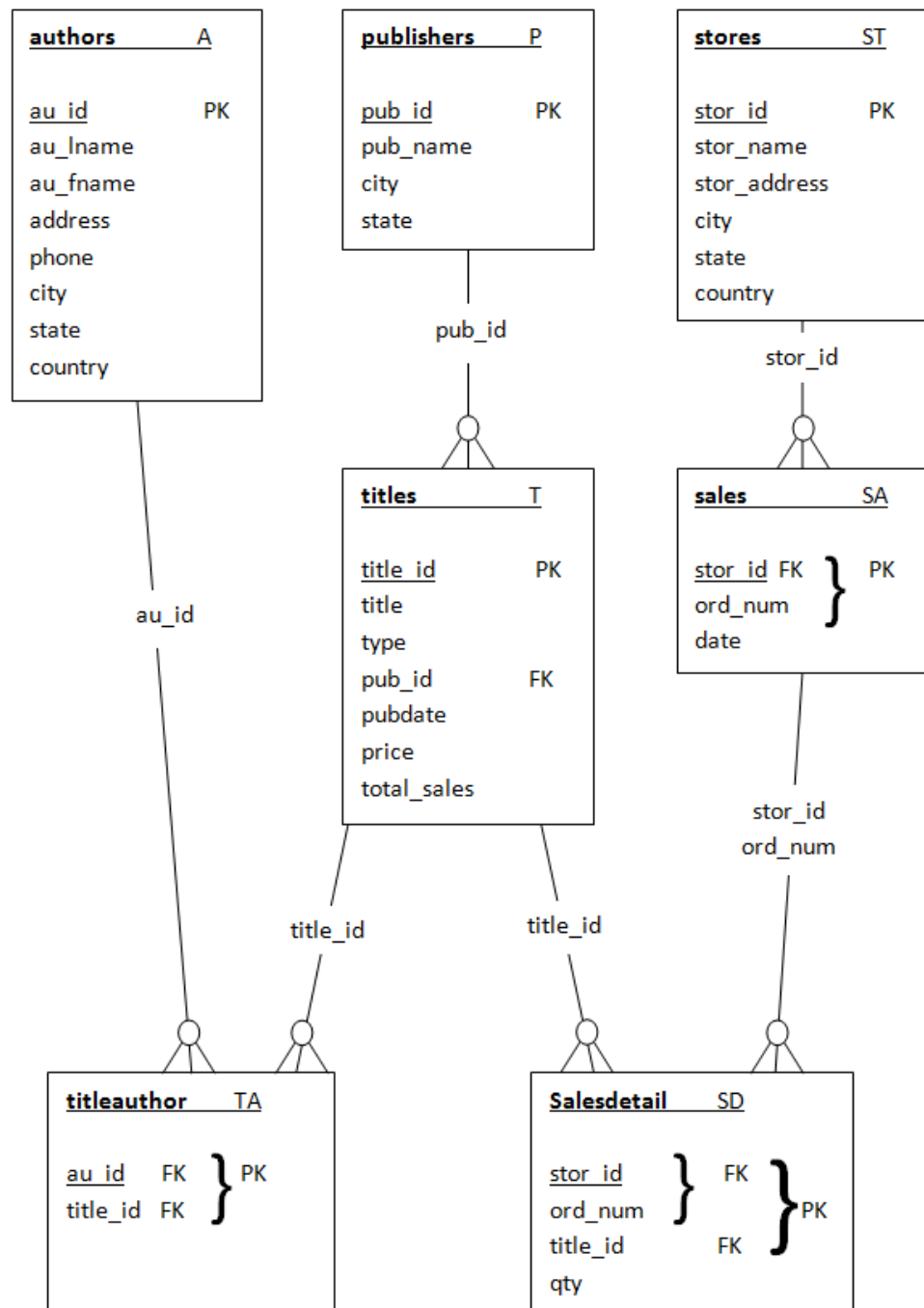
Pas de vente et donc pas de jointure...



Tous les livres avec leurs quantités vendues

```
SELECT t.title, SUM(sd.qty)
FROM   titles t LEFT OUTER JOIN
        salesdetail sd
ON      t.title_id=sd.title_id
GROUP BY t.title_id;
```

	title character varying(80)	sum bigint
1	The Psychology of Computer Cooking	
2	Sushi, Anyone?	4095
3	Cooking with Computers: Surreptitious Balance Sheets	3876
4	Computer Phobic and Non-Phobic Individuals: Behavior Variations	375
5	The Gourmet Microwave	22246
6	Emotional Security: A New Algorithm	3336
7	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean	375
8	Fifty Years in Buckingham Palace Kitchens	15096
9	You Can Combat Computer Stress!	15722
10	Straight Talk About Computers	4095
11	Secrets of Silicon Valley	2095
12	Prolonged Data Deprivation: Four Case Studies	4072
13	But Is It User Friendly?	8780
14	The Busy Executive's Database Guide	4095
15	Net Etiquette	
16	Silicon Valley Gastronomic Treats	2032
17	Life Without Fear	111
18	Is Anger the Enemy?	2045



Autre notation fréquente du LEFT OUTER JOIN

```
SELECT t.title, SUM(sd.qty)
FROM   titles t LEFT OUTER JOIN
       salesdetail sd
ON     t.title_id=sd.title_id
GROUP BY t.title_id;
```

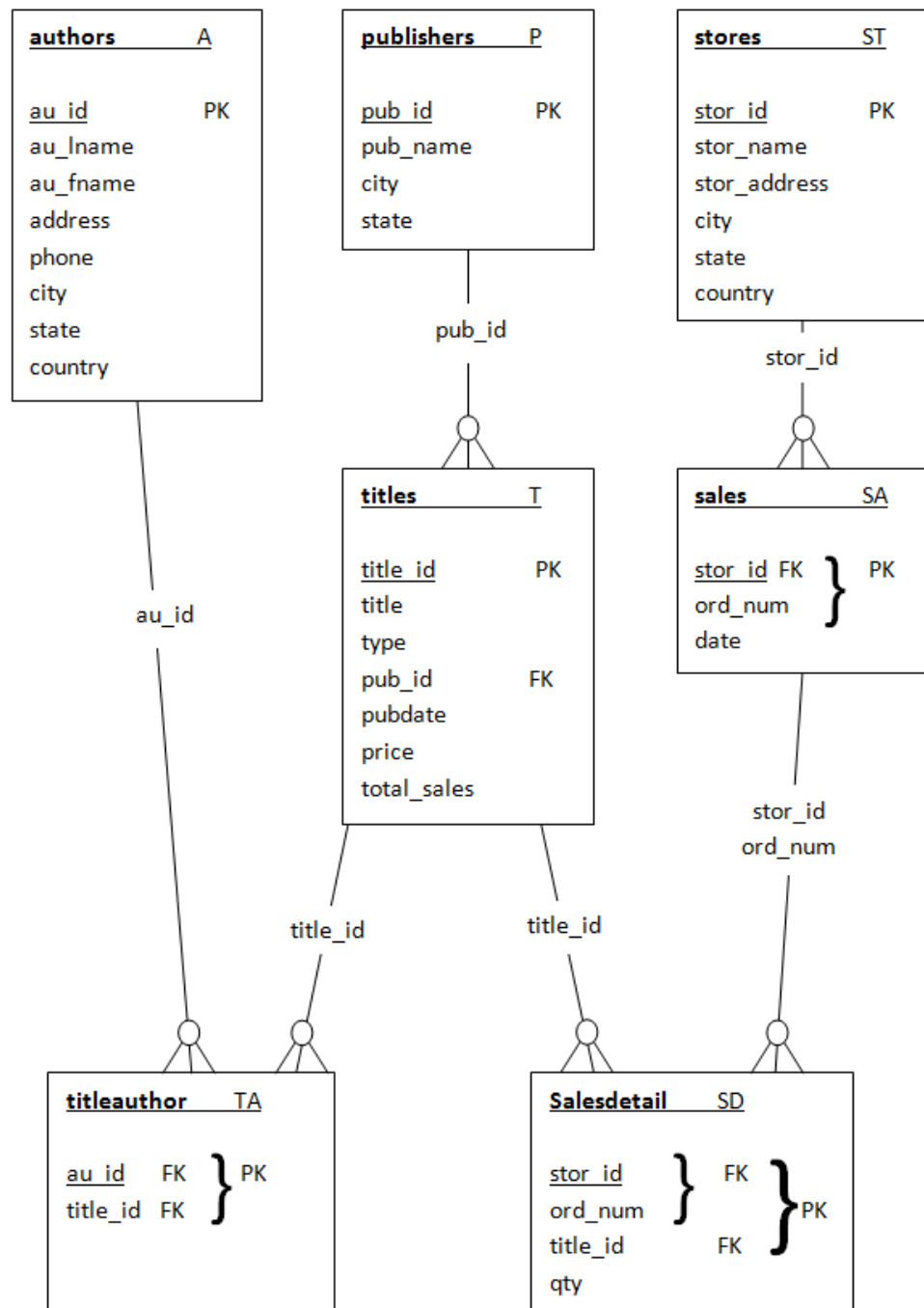
```
SELECT t.title, SUM(sd.qty)
FROM titles t, salesdetail sd
WHERE t.title_id*=sd.title_id
GROUP BY t.title_id;
```

Fonctionne avec Oracle, Sybase, mais **pas** avec PostgreSQL

Tous les livres avec leurs quantités vendues

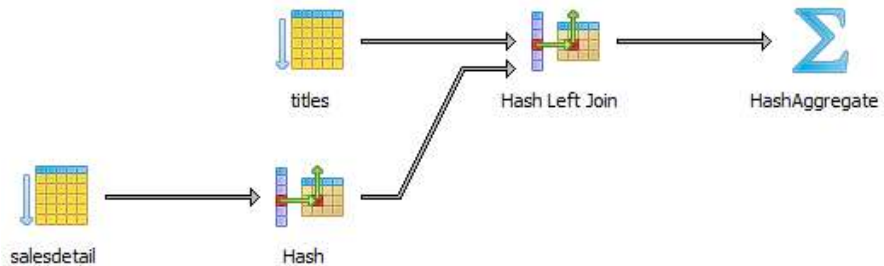
```
SELECT t.title, SUM(sd.qty)
FROM salesdetail sd RIGHT OUTER JOIN
titles t
ON t.title_id=sd.title_id
GROUP BY t.title_id;
```

	title character varying(80)	sum bigint
1	The Psychology of Computer Cooking	
2	Sushi, Anyone?	4095
3	Cooking with Computers: Surreptitious Balance Sheets	3876
4	Computer Phobic and Non-Phobic Individuals: Behavior Variations	375
5	The Gourmet Microwave	22246
6	Emotional Security: A New Algorithm	3336
7	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean	375
8	Fifty Years in Buckingham Palace Kitchens	15096
9	You Can Combat Computer Stress!	15722
10	Straight Talk About Computers	4095
11	Secrets of Silicon Valley	2095
12	Prolonged Data Deprivation: Four Case Studies	4072
13	But Is It User Friendly?	8780
14	The Busy Executive's Database Guide	4095
15	Net Etiquette	
16	Silicon Valley Gastronomic Treats	2032
17	Life Without Fear	111
18	Is Anger the Enemy?	2045

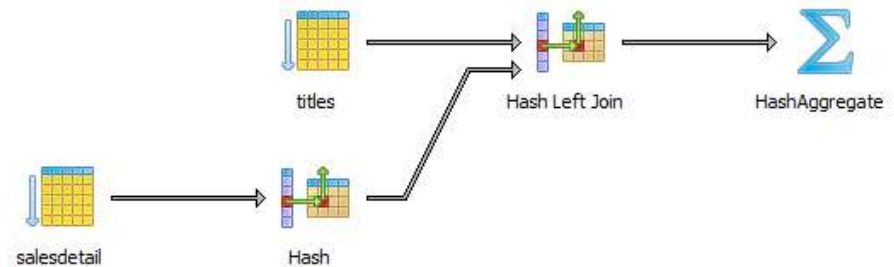


RIGHT vs LEFT

```
SELECT t.title, SUM(sd.qty)
FROM titles t
LEFT OUTER JOIN salesdetail sd
ON t.title_id=sd.title_id
GROUP BY t.title_id;
```



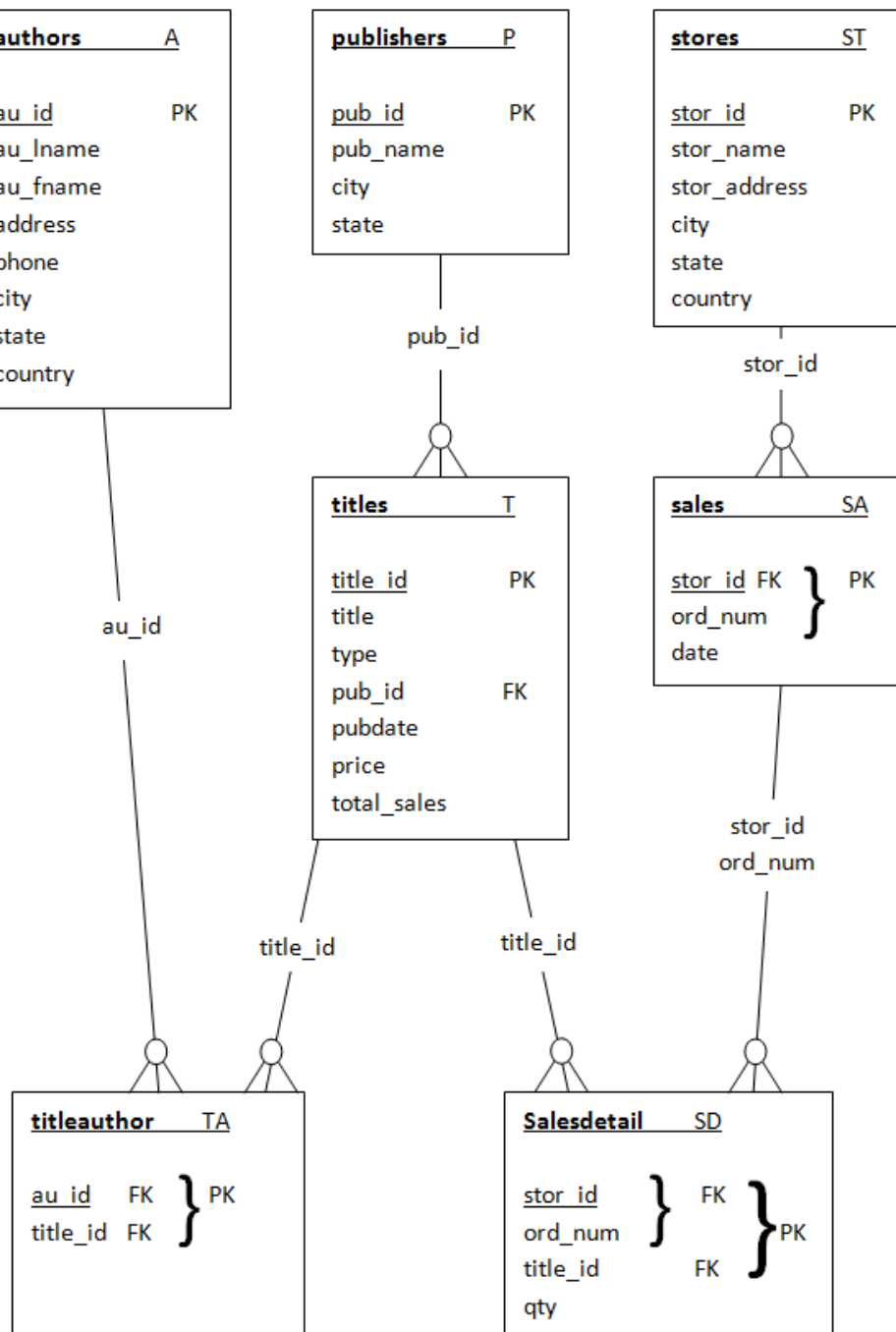
```
SELECT t.title, SUM(sd.qty)
FROM salesdetail sd
RIGHT OUTER JOIN titles t
ON t.title_id=sd.title_id
GROUP BY t.title_id;
```



Tous les auteurs avec tous les livres

```
SELECT a.au_lname, t.title
FROM authors a
FULL OUTER JOIN titleauthor ta
ON a.au_id=ta.au_id
FULL OUTER JOIN titles t
ON ta.title_id=t.title_id
```

8	DeTale	The Gourmet Microwave
9	Ringer	The Gourmet Microwave
10		The Psychology of Comput
11	Carson	But Is It User Friendly?
12	Dull	Secrets of Silicon Valley
13	Hunter	Secrets of Silicon Valley
14	Locksley	Net Etiquette
15	Karsen	Computer Phobic and Non-
16	MacFeather	Computer Phobic and Non-
17	Ringer	Is Anger the Enemy?
18	Ringer	Is Anger the Enemy?
19	Ringer	Life Without Fear
20	White	Prolonged Data Deprivation
21	Locksley	Emotional Security: A New
22	Panteley	Onions, Leeks, and Garlic:
23	Blotch-Halls	Fifty Years in Buckingham F
24	Gringlesby	Sushi, Anyone?
25	O Leary	Sushi, Anyone?
26	Yokomoto	Sushi, Anyone?
27	McBadden	
28	Greene	

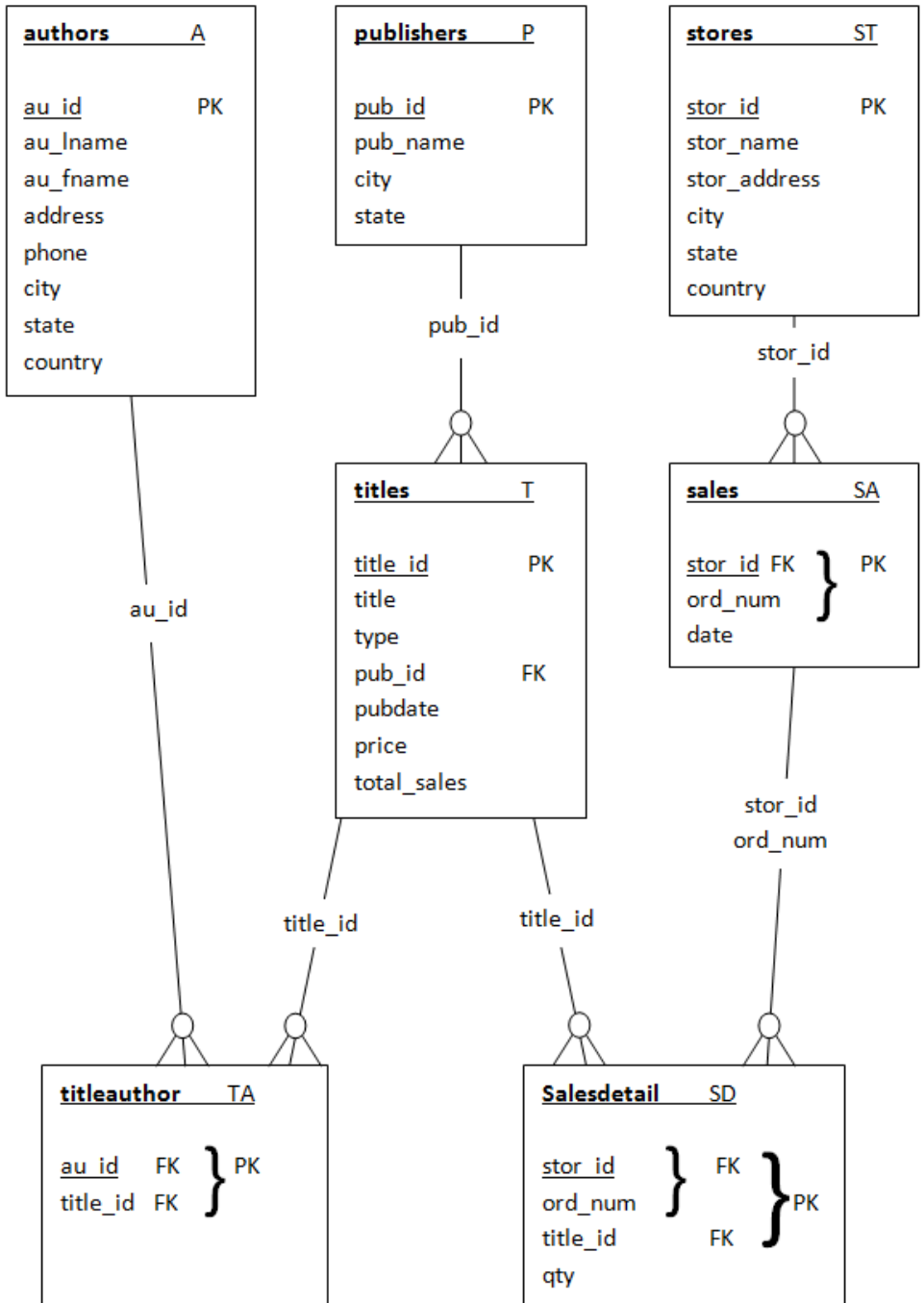


Les livres d'auteurs californiens & éditeurs californiens & vendus en Californie

```

SELECT DISTINCT (t.title)
FROM  authors a,
      titleauthor ta,
      titles t,
      publishers p,
      salesdetail sd,
      stores st
WHERE  a.au_id=ta.au_id
AND    ta.title_id=t.title_id
AND    t.pub_id=p.pub_id
AND    t.title_id=sd.title_id
AND    sd.stor_id=st.stor_id
AND    a.state='CA'
AND    p.state='CA'
AND    st.state='CA'

```



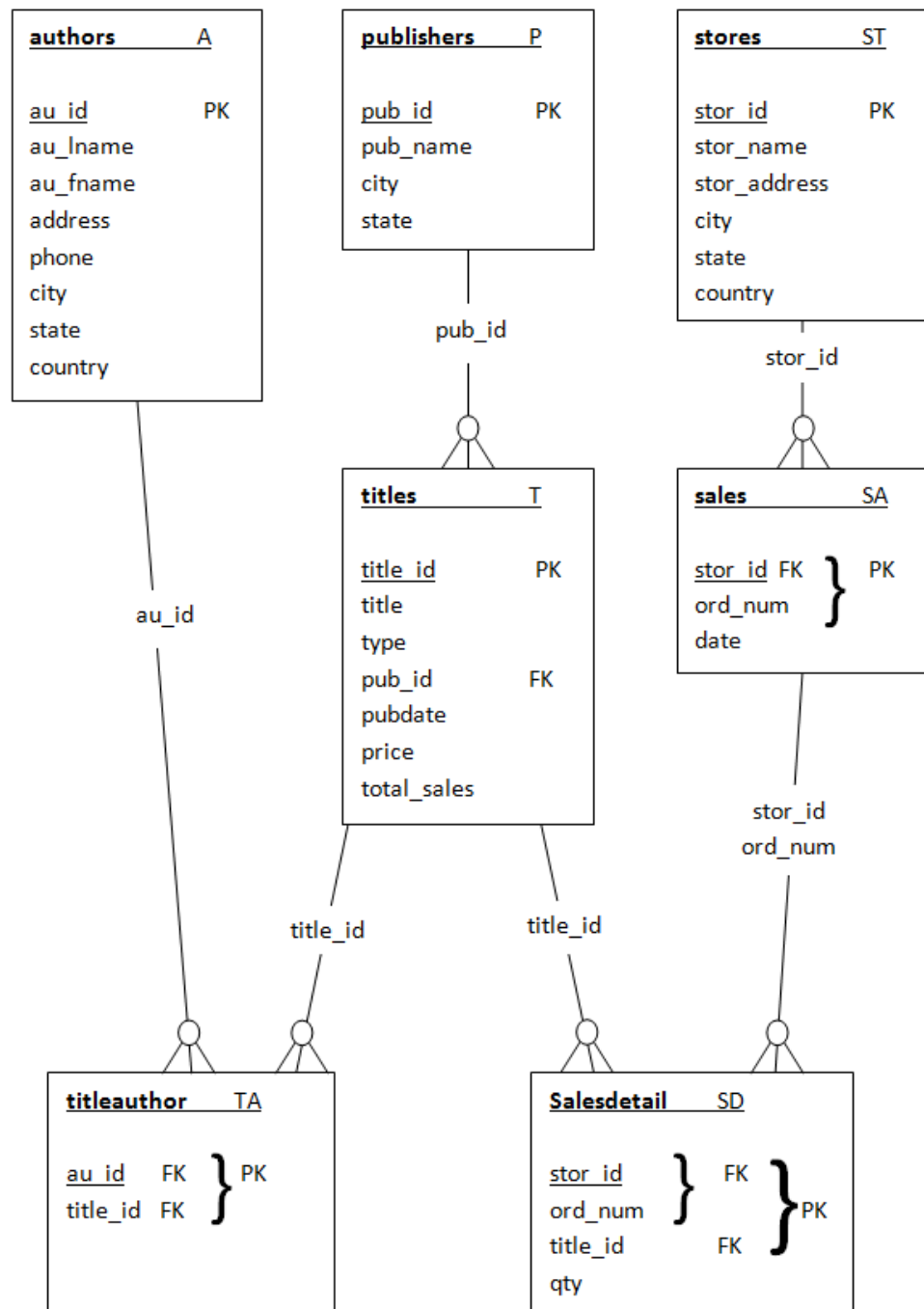
Les livres d'auteurs californiens & éditeurs californiens & vendus **que** en Californie

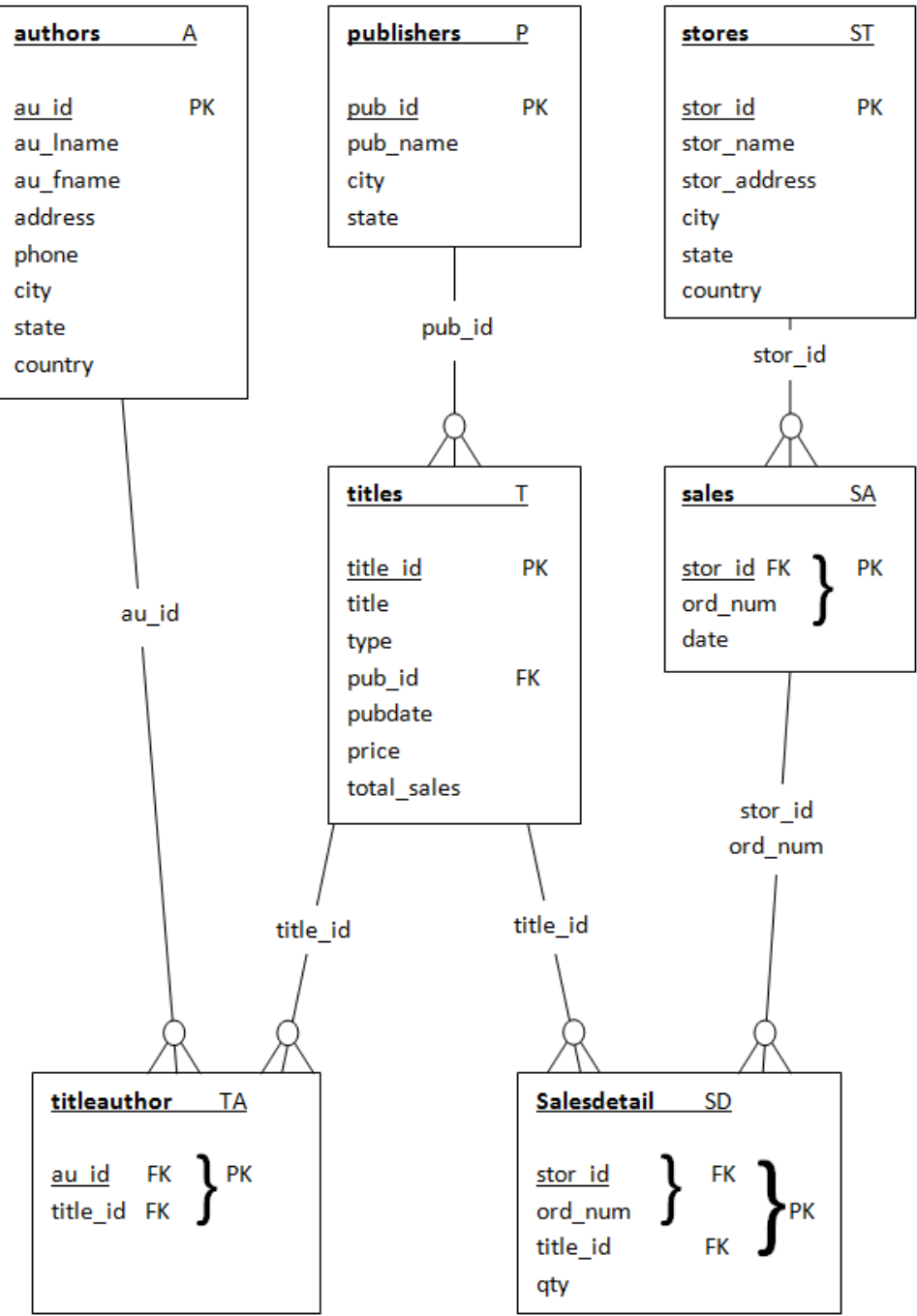
Question différente = requête différente

Reformulation :

Livres d'auteurs californien & d'éditeurs californiens
& pas vendu ailleurs qu'en Californie

```
SELECT DISTINCT (t.title)
FROM  authors a,
      titleauthor ta,
      titles t,
      publishers p
WHERE  a.au_id=ta.au_id
AND    ta.title_id=t.title_id
AND    t.pub_id=p.pub_id
AND    a.state='CA'
AND    p.state='CA'
AND NOT EXISTS (
    SELECT *
    FROM  salesdetail sd, stores st
    WHERE t.title_id=sd.title_id
    AND   sd.stor_id=st.stor_id
    AND   st.state<>'CA')
```

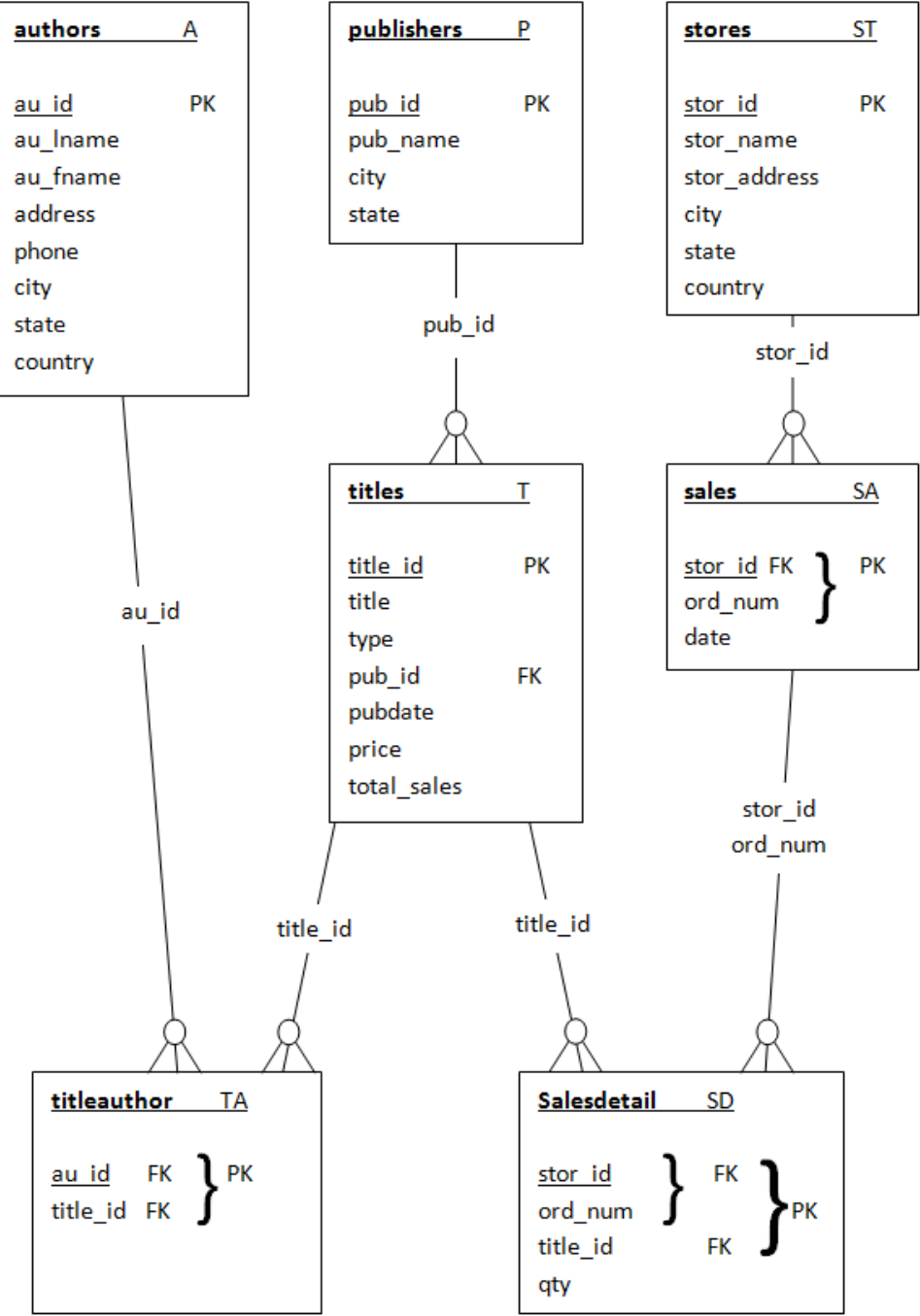




Les livres d'auteurs californiens & éditeurs californiens & vendus **que** en Californie

Reformulation correcte :
Livres d'auteurs californien & d'éditeurs californiens
& vendu & pas vendu ailleurs qu'en Californie

```
SELECT DISTINCT (t.title)
FROM authors a,
titleauthor ta,
titles t,
publishers p,
salesdetail sd
WHERE a.au_id=ta.au_id
AND ta.title_id=t.title_id
AND t.pub_id=p.pub_id
AND t.title_id=sd.title_id
AND a.state='CA'
AND p.state='CA'
AND NOT EXISTS (
  SELECT *
  FROM salesdetail sd, stores st
  WHERE t.title_id=sd.title_id
  AND sd.stor_id=st.stor_id
  AND st.state<>'CA')
```



Les livres d'auteurs californiens & éditeurs
californiens & vendus **que** en Californie

Reformulation correcte :
Livres d'auteurs californien & d'éditeurs
californiens & vendu & tous les magasins
qui l'ont vendu sont en Californie

```
SELECT DISTINCT (t.title)
FROM  authors a,
      titleauthor ta,
      titles t,
      publishers p,
      salesdetail sd
WHERE  a.au_id=ta.au_id
AND    ta.title_id=t.title_id
AND    t.pub_id=p.pub_id
AND    t.title_id=sd.title_id
AND    a.state='CA'
AND    p.state='CA'
AND    'CA'=ALL (
      SELECT st.state
      FROM    salesdetail sd,
             stores st
      WHERE   t.title_id=sd.title_id
      AND     sd.stor_id=st.stor_id)
```