

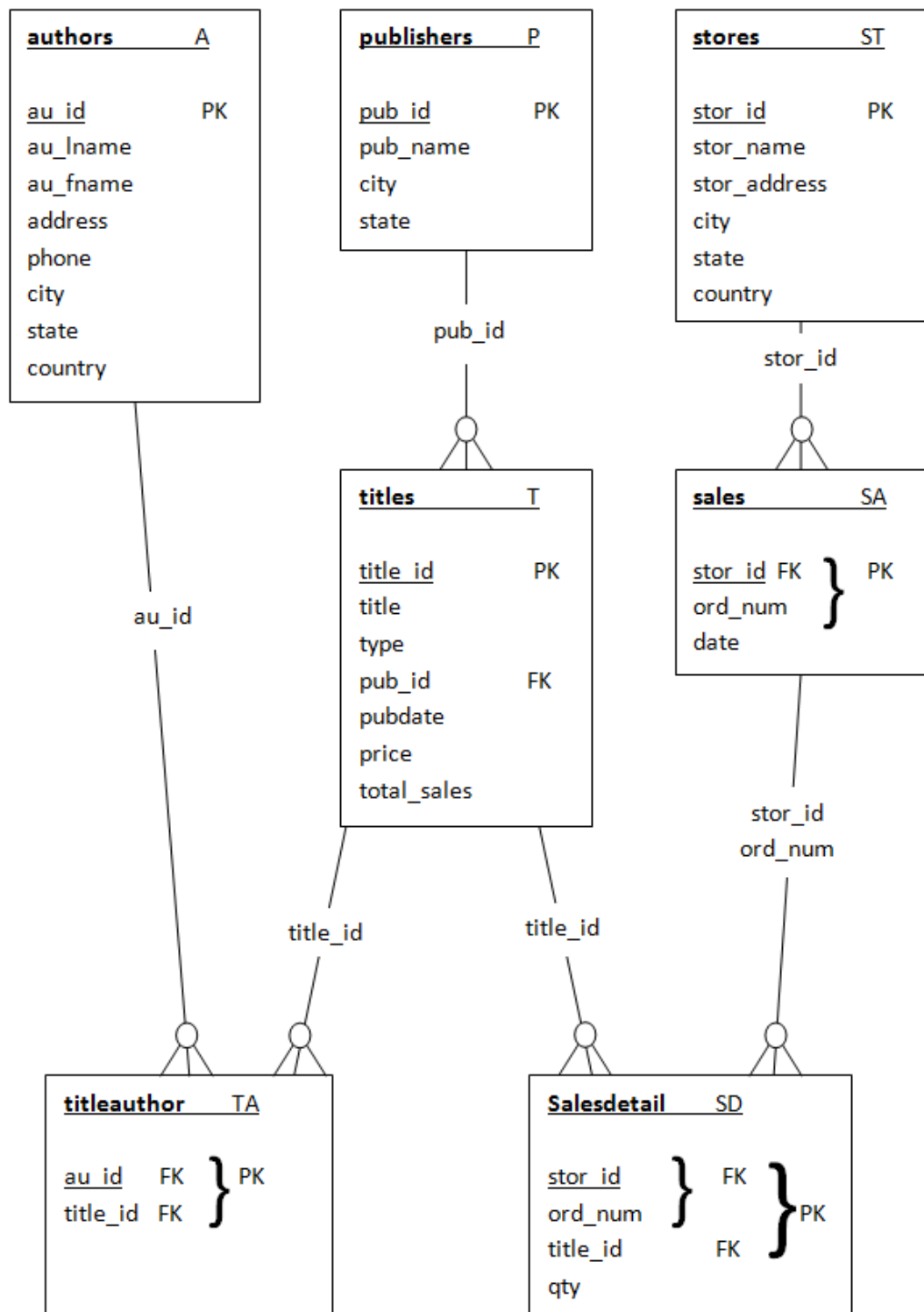
Les auteurs sans téléphone

SELECT *

FROM authors a

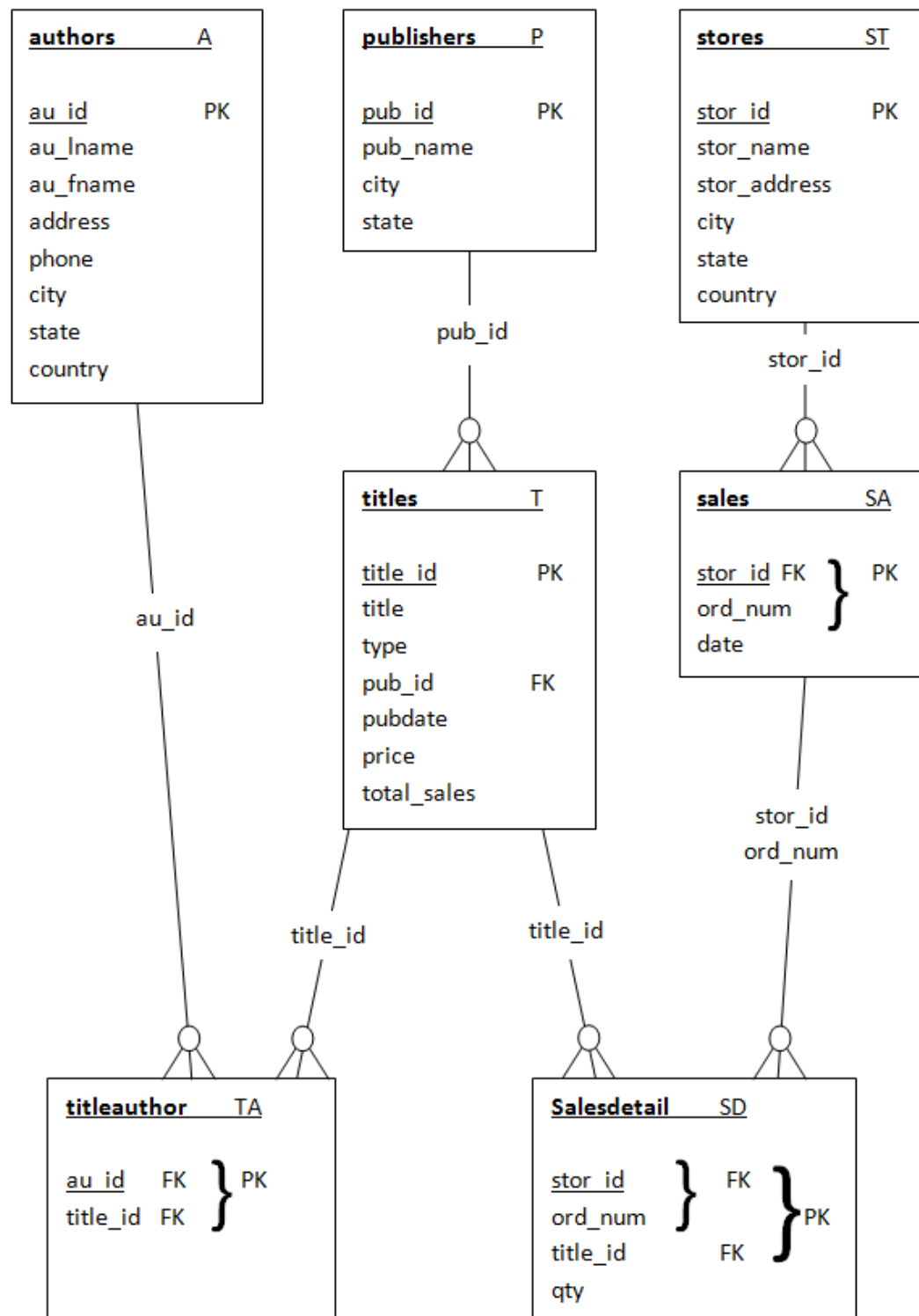
WHERE a.phone IS NULL

NULL s'utilise toujours par **IS NULL** ou **IS NOT NULL**,
jamais par **= NULL** ou **<> NULL**



Les auteurs en Californie, Oregon et Michigan

```
SELECT *  
FROM authors a  
WHERE a.state IN  
('CA', 'OR', 'MI')
```



Les ventes en 1991

```
SELECT *
```

```
FROM sales s
```

```
WHERE s.date>='1991-01-01'
```

```
AND s.date<'1992-01-01'
```

```
SELECT *
```

```
FROM sales s
```

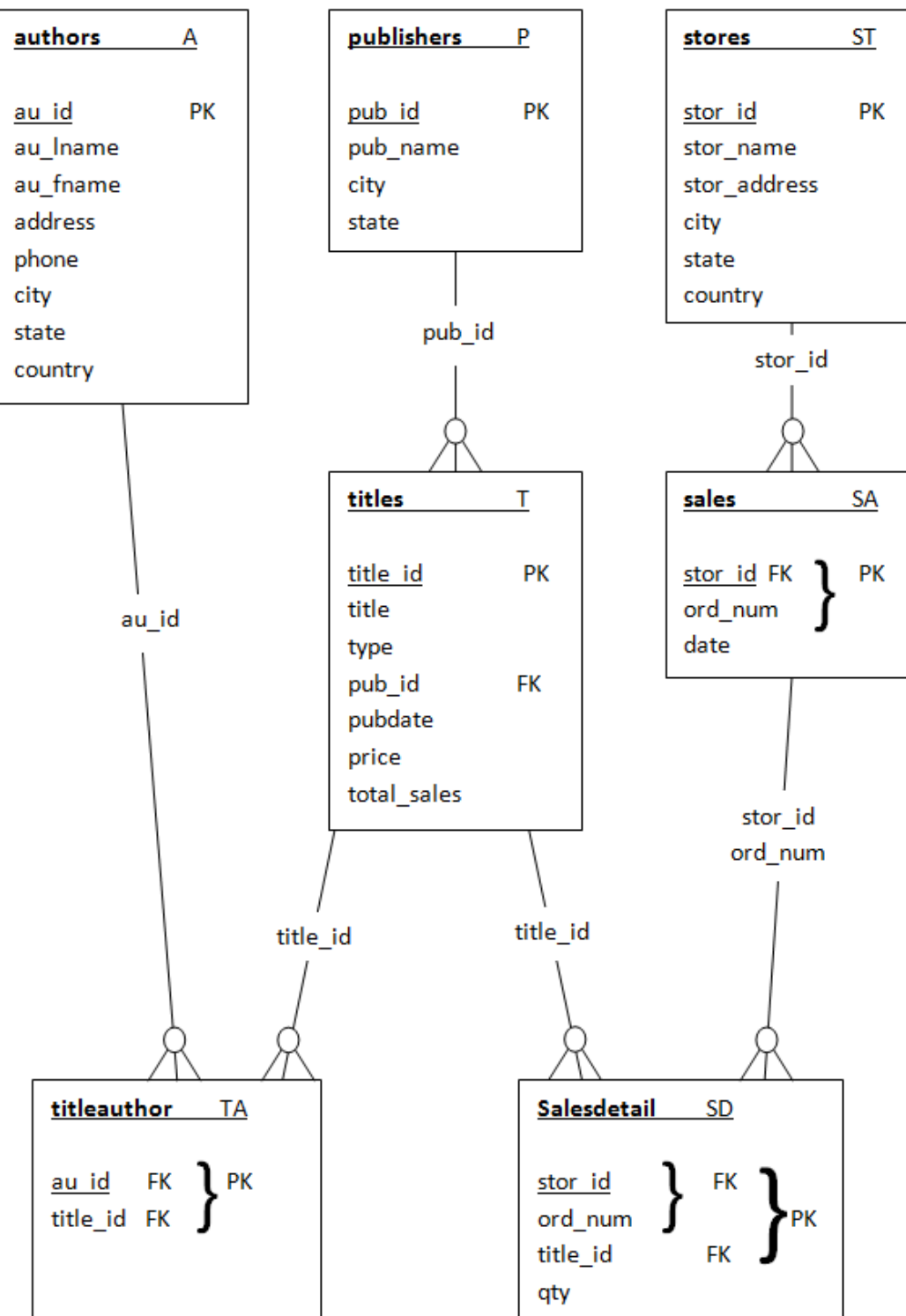
```
WHERE s.date BETWEEN '1991-01-01'
AND '1991-12-31'
```

```
SELECT *
```

```
FROM sales s
```

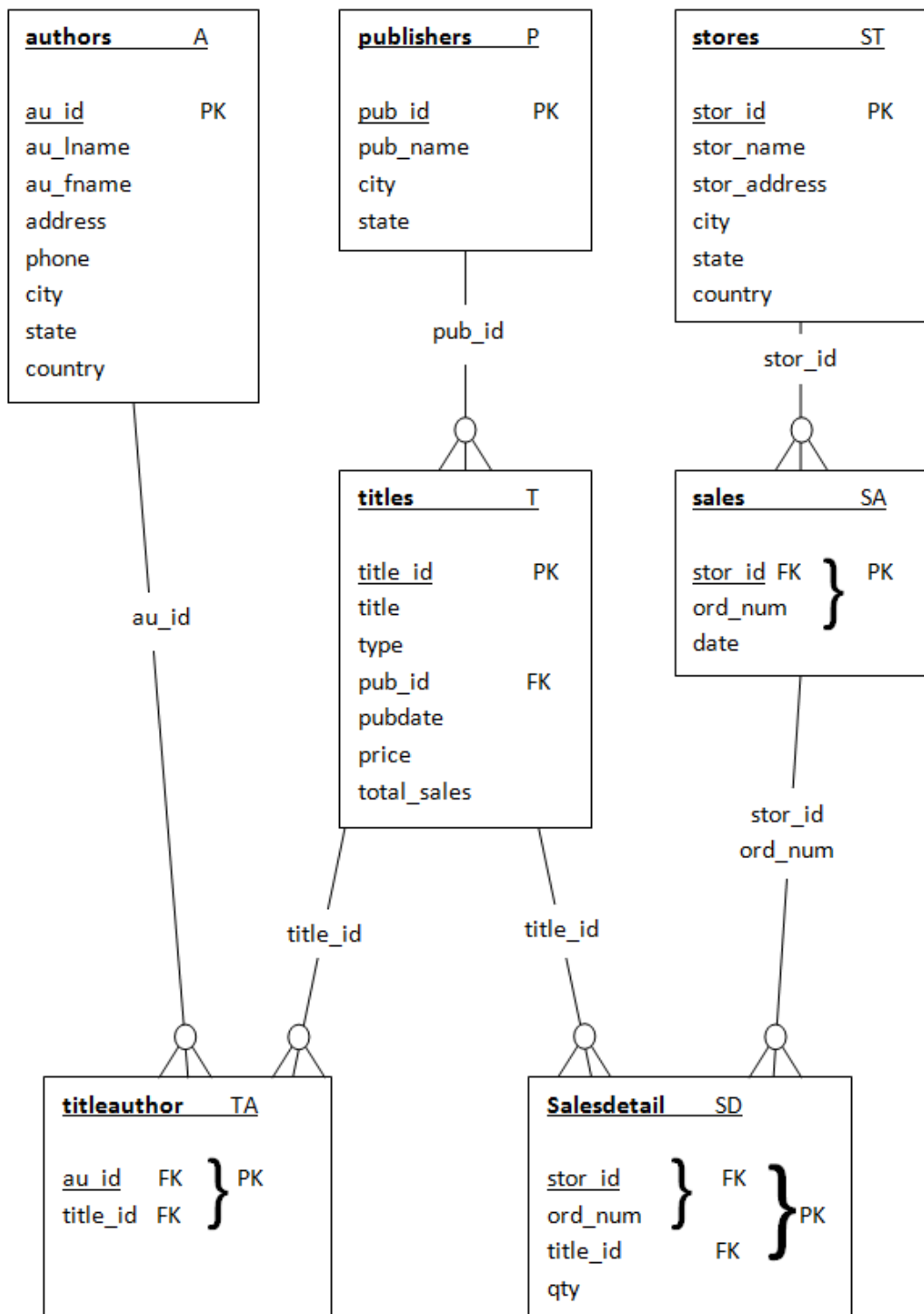
```
WHERE
```

```
date_part('YEAR',s.date)=1991
```



Magasins ayant vendu livre contenant cook ?

```
SELECT DISTINCT
    st.stor_id,st.stor_name
FROM stores st,
    salesdetail sd, sales
    sa, titles t
WHERE
    st.stor_id=sa.stor_id
AND sa.stor_id=sd.stor_id
AND sa.ord_num=sd.ord_num
AND sd.title_id=t.title_id
AND t.title SIMILAR TO
    '%[cC]ook%'
```



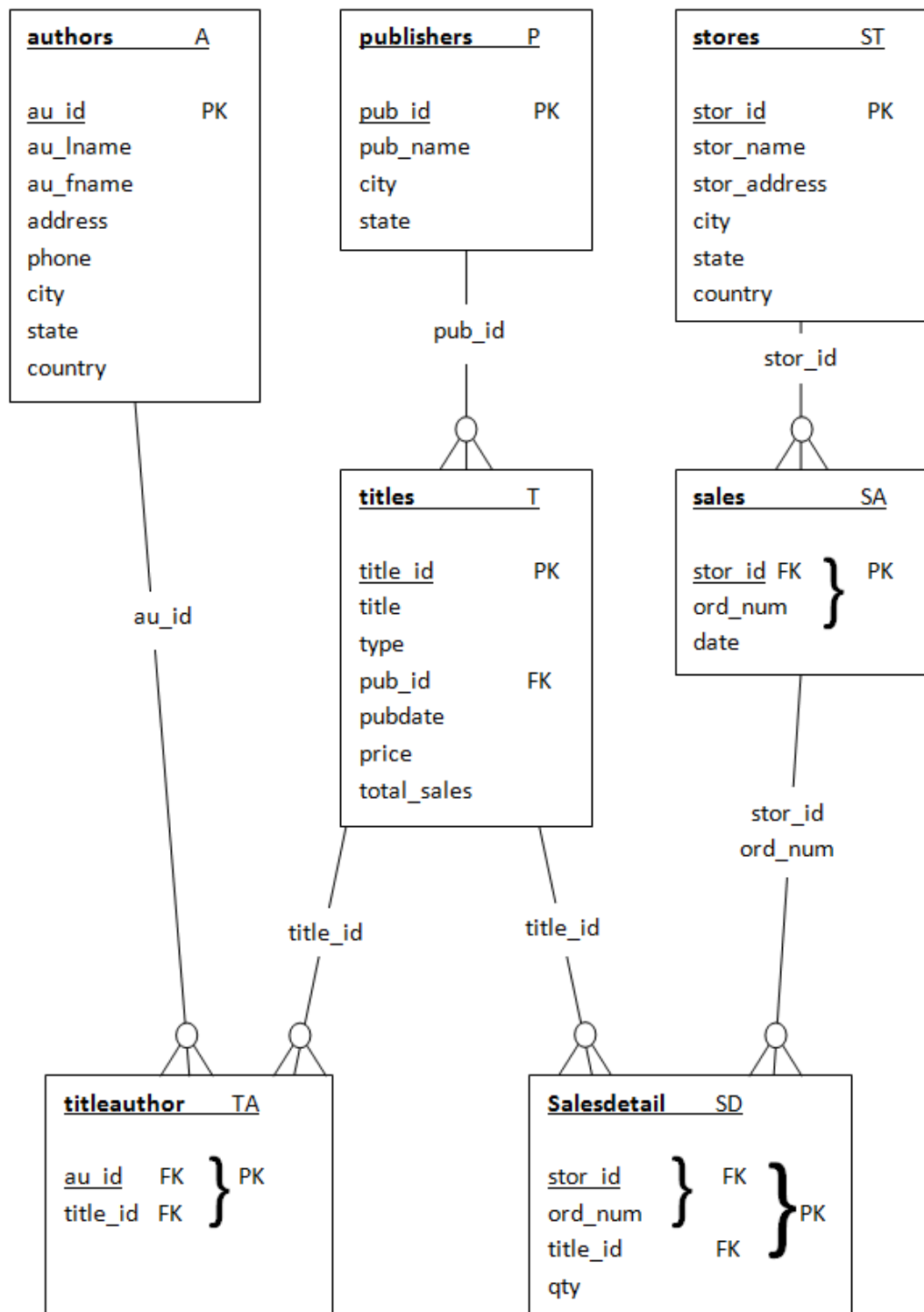
RAPPEL MATHÉMATIQUE : LA TRANSITIVITÉ

Si $A=B$ et $B=C$

- **Alors $A=C$!**

Magasins ayant vendu livre contenant cook ?

```
SELECT DISTINCT
    st.stor_id, st.stor_name
FROM stores st,
    salesdetail sd, titles t
WHERE
    st.stor_id=sd.stor_id
AND sd.title_id=t.title_id
AND t.title SIMILAR TO
    '%[cC]ook%'
```

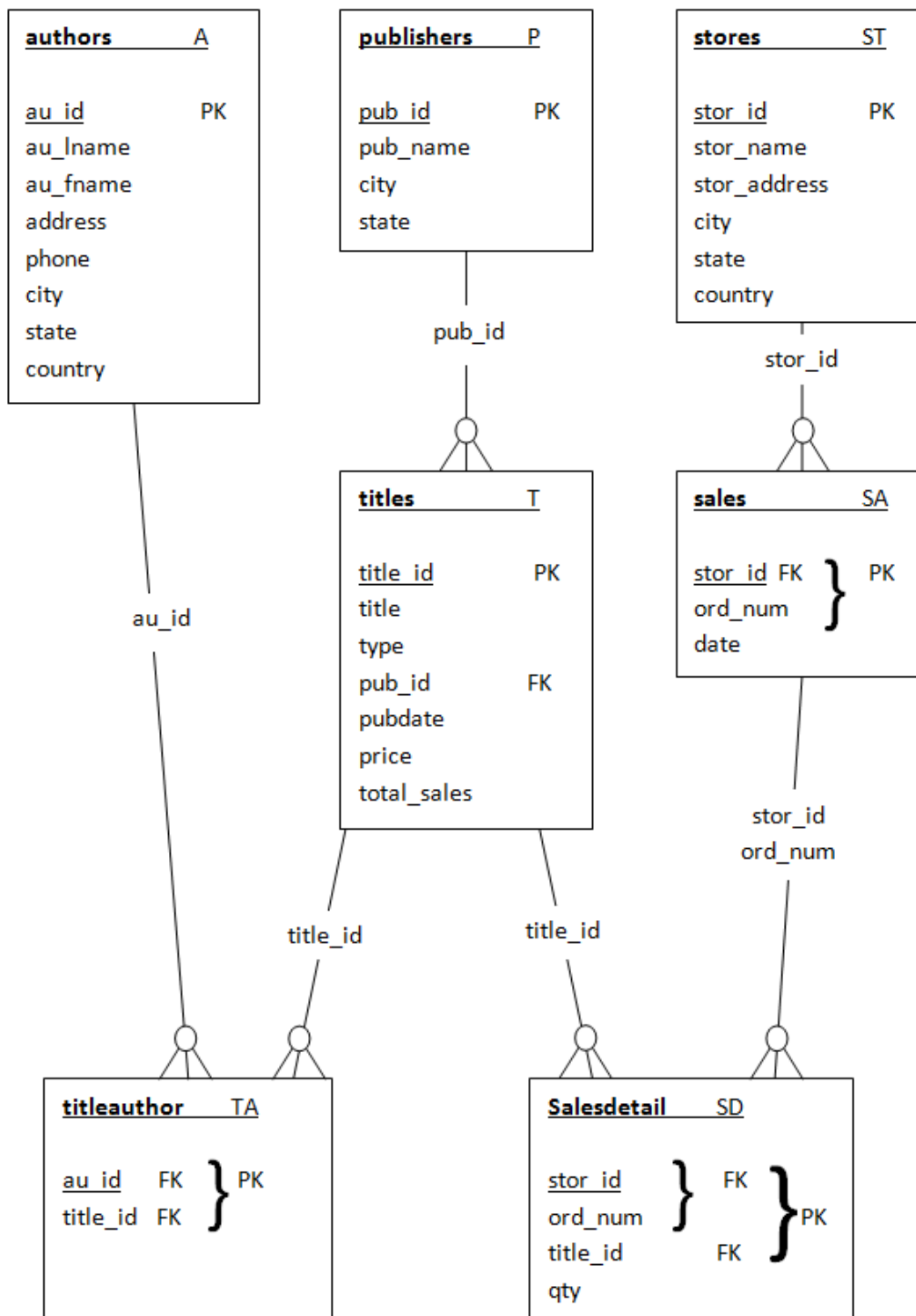


Auteurs n'ayant pas tout publié au même éditeur ?

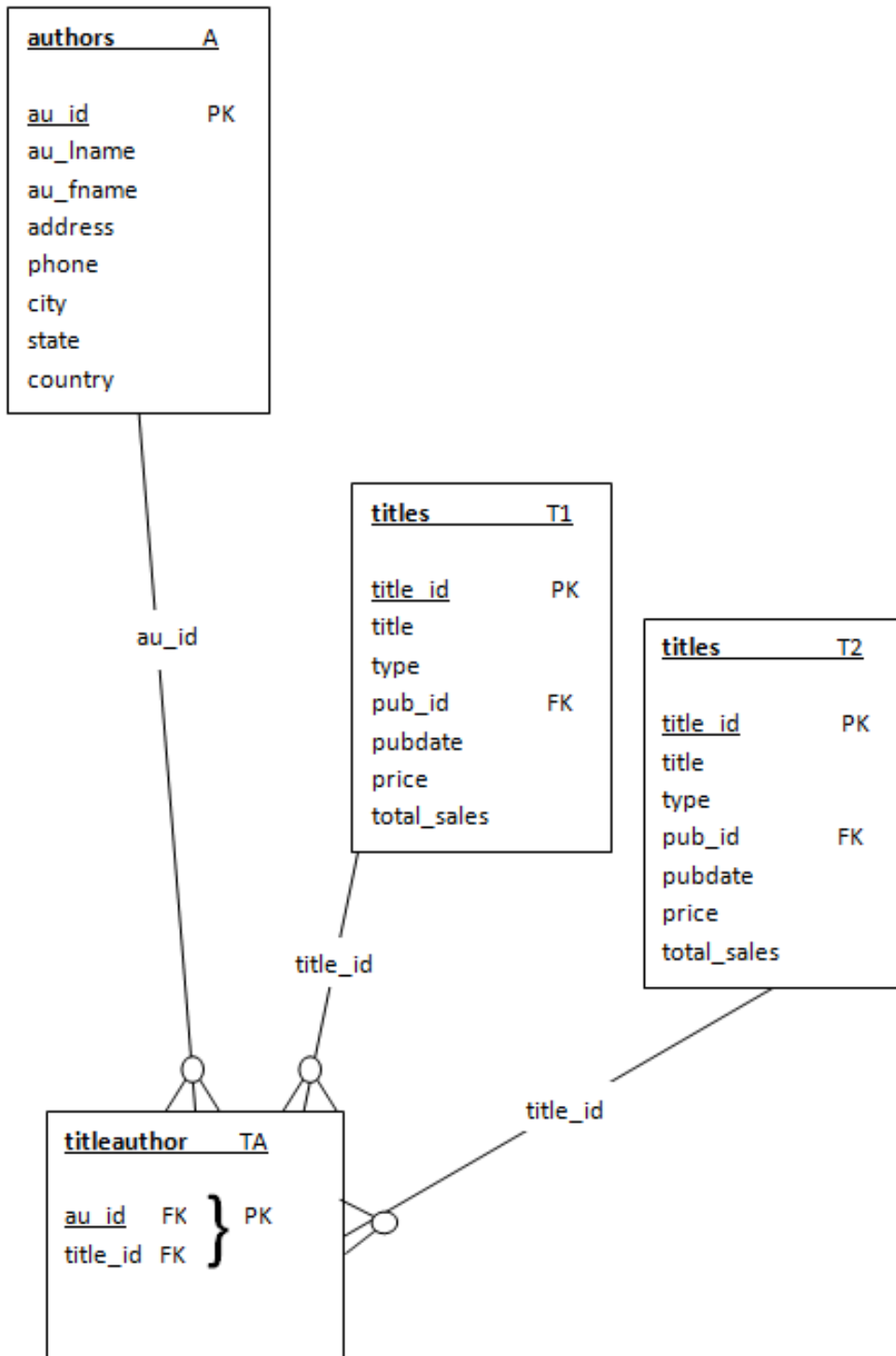
On ne peut pas répondre à cette question directement sous cette forme.

Reformulation :

Auteurs ayant publié au moins deux livres chez deux éditeurs différents.



Auteurs avec 2 livres chez 2 éditeurs différents ?



```
SELECT DISTINCT a.*
```

```
FROM authors a,  
     titleauthor ta, titles  
     t1, titles t2
```

```
WHERE  
      t1.title_id<>t2.title_id
```

```
AND  
      t1.title_id=ta.title_id
```

```
AND  
      t2.title_id=ta.title_id
```

```
AND ta.au_id=a.au_id
```

```
AND t1.pub_id<>t2.pub_id
```


Auteurs avec 2 livres chez 2 éditeurs différents ?

SELECT DISTINCT a.*

FROM authors a, titleauthor
ta, titles t1, titles t2

WHERE

t1.title_id<>t2.title_id

AND t1.title_id=ta.title_id

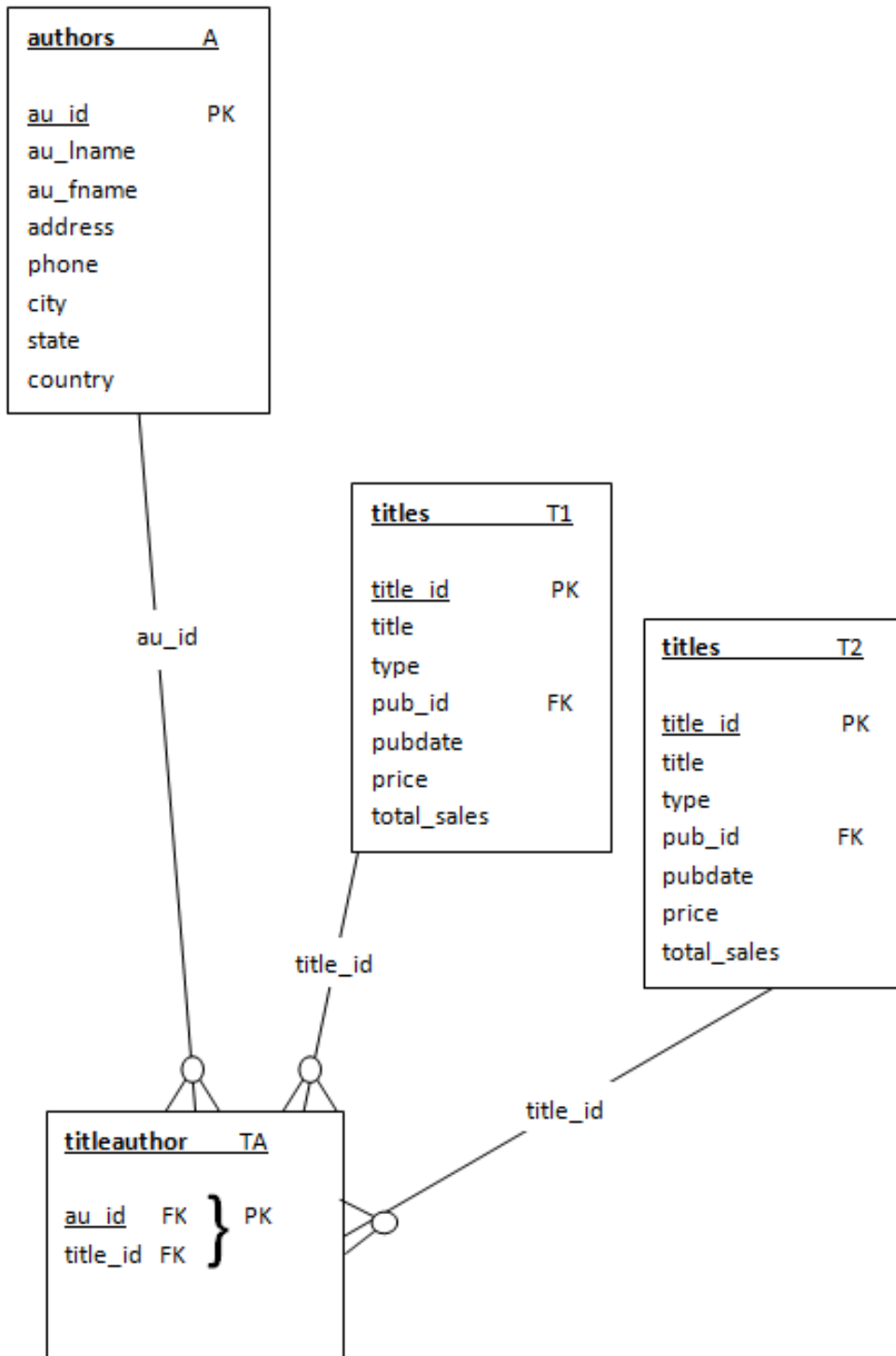
AND t2.title_id=ta.title_id

AND ta.au_id=a.au_id

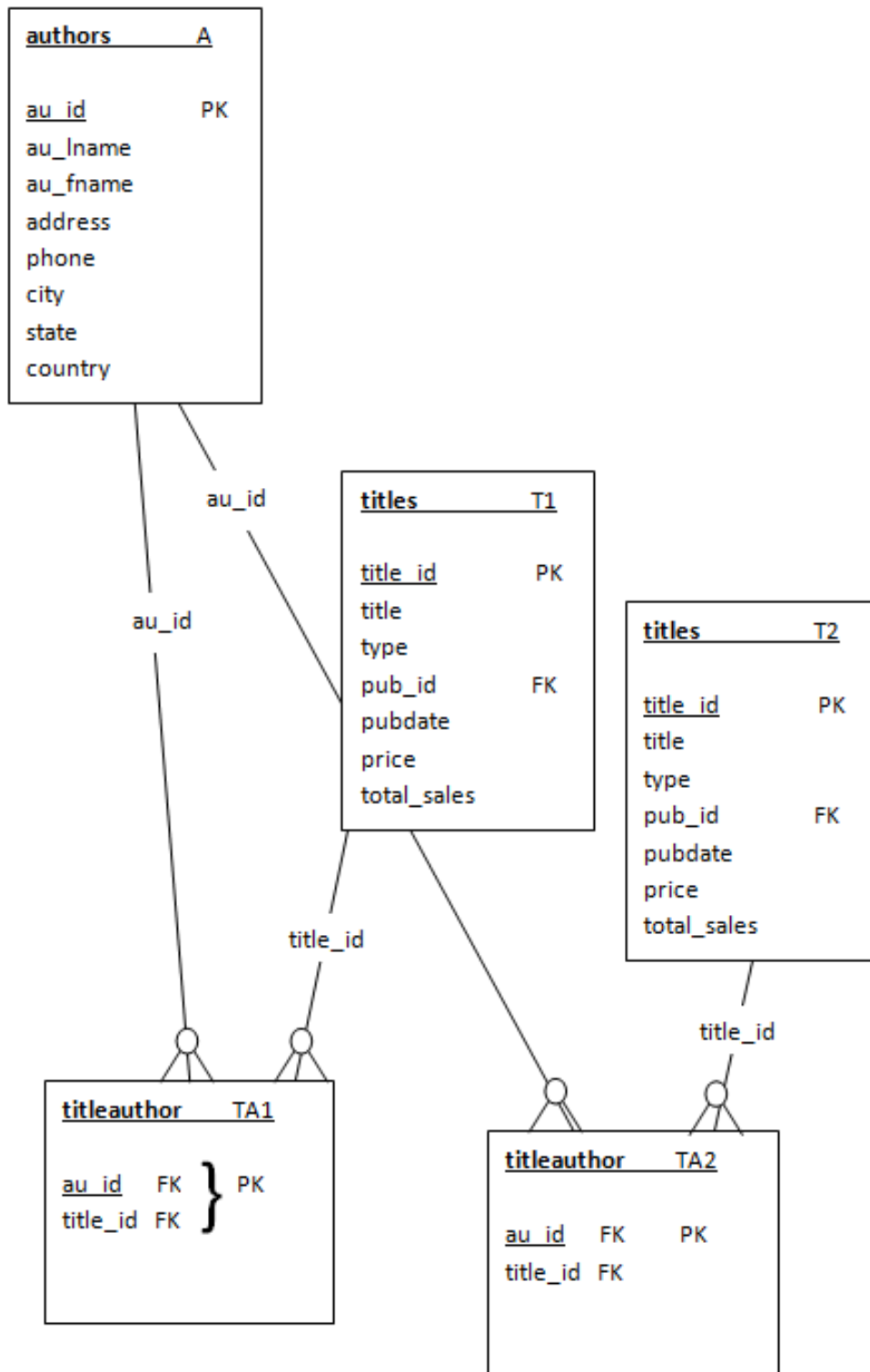
AND t1.pub_id<>t2.pub_id

Incorrect !

Par transitivité t1.title_id=t2.title_id !



Auteurs avec 2 livres chez 2 éditeurs différents ?



```
SELECT DISTINCT a.*
```

```
FROM authors a, titleauthor  
ta1, titleauthor ta2,  
titles t1, titles t2
```

```
WHERE
```

```
t1.title_id<>t2.title_id
```

```
AND t1.title_id=ta1.title_id
```

```
AND t2.title_id=ta2.title_id
```

```
AND ta1.au_id=a.au_id
```

```
AND ta2.au_id=a.au_id
```

```
AND t1.pub_id<>t2.pub_id
```

CORRECT ! Est-ce que la partie en
bleu est nécessaire ?

SOUS SELECT

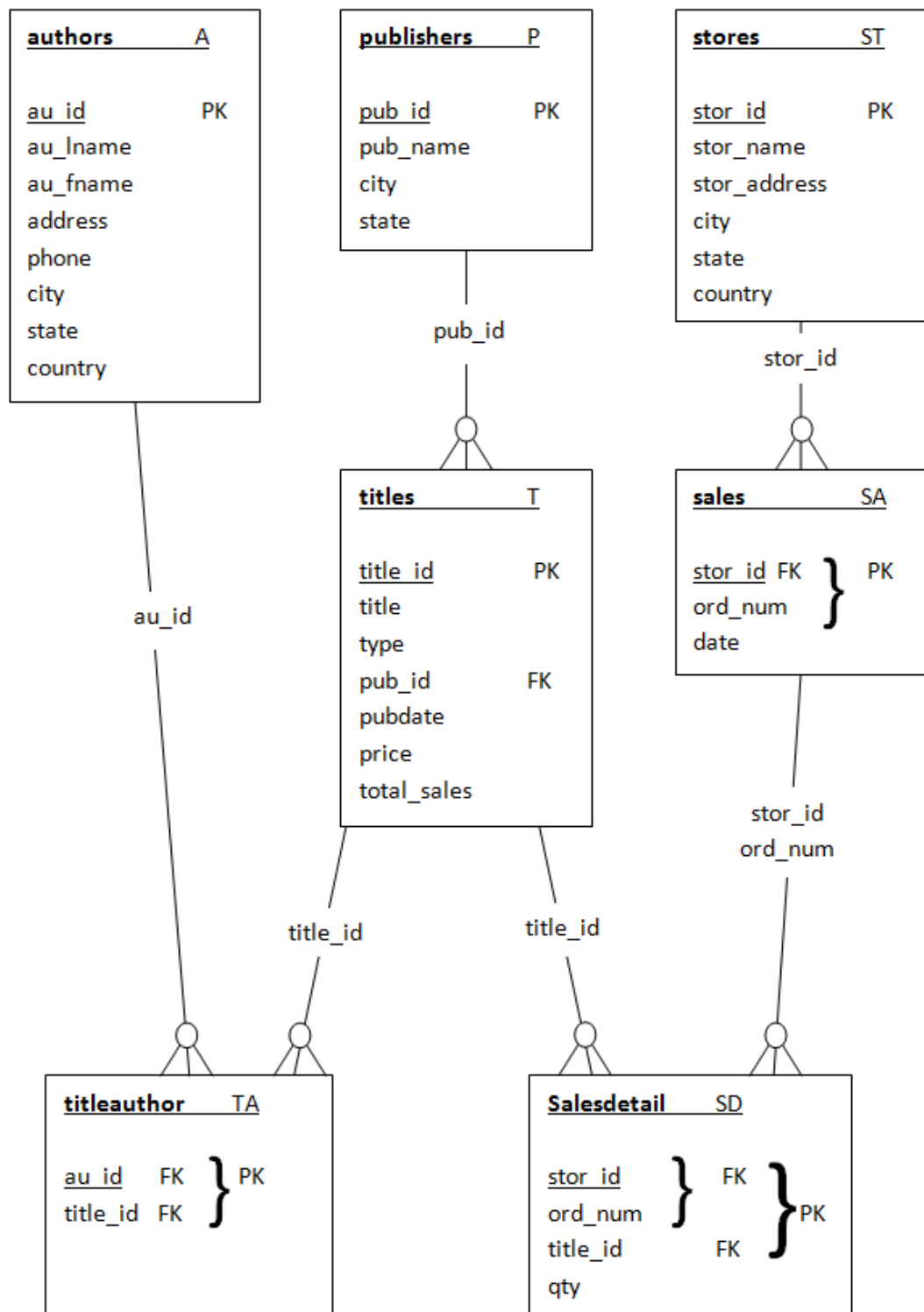
Dans la partie condition :

- `conditionelementaire =`
`expression (= | <> | < | > | <= | >=) (SELECT ...)`

où (SELECT ...) est un SELECT retournant un seul tuple d'une seule colonne.

Livre le plus cher

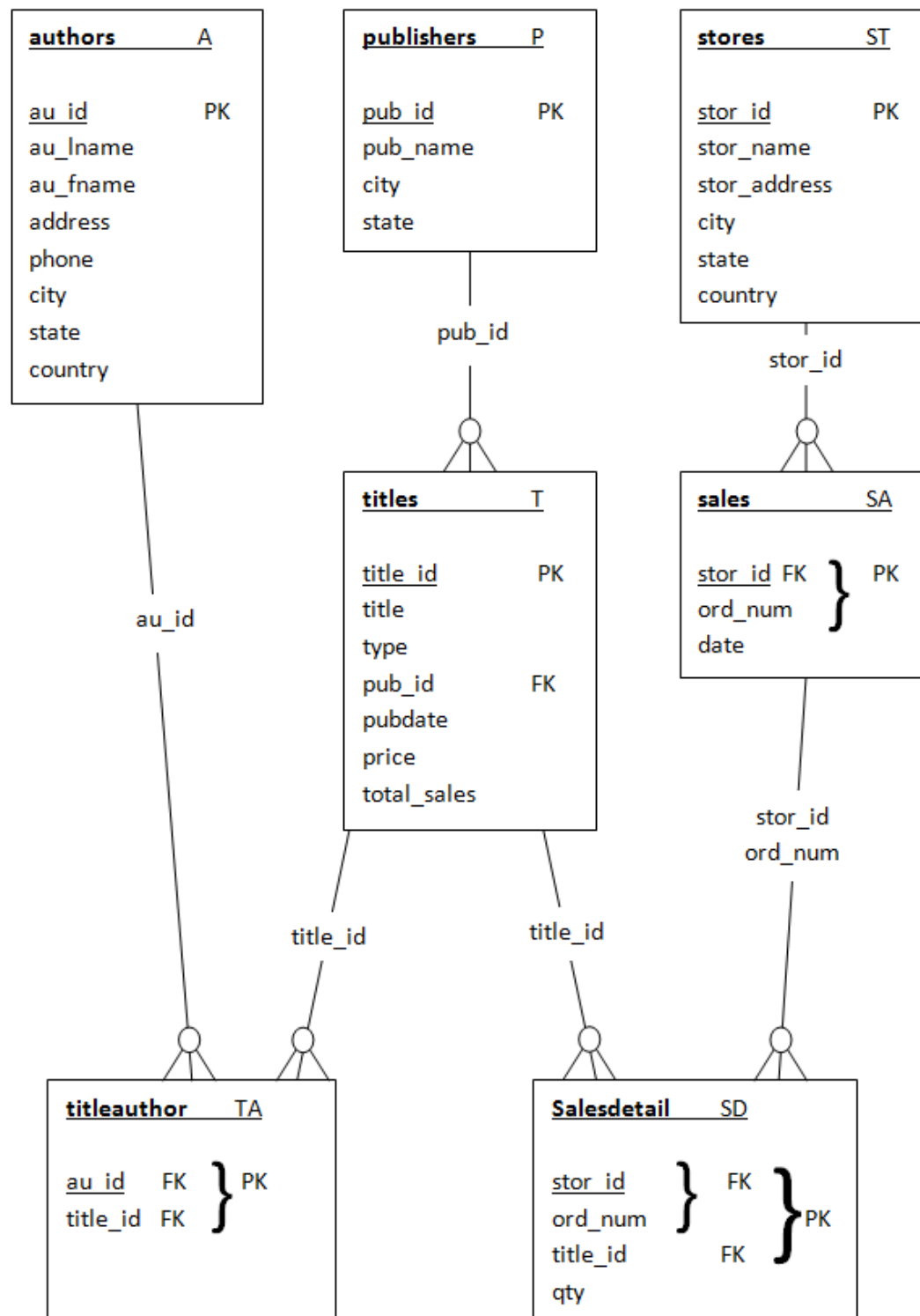
```
SELECT t.title_id,  
       t.title  
FROM titles t  
WHERE t.price =  
       (SELECT MAX(price)  
        FROM titles) ;
```



Quels sont les auteurs qui ont écrit exactement 2 livres ?

```
SELECT a.au_id, a.au_lname,  
       a.au_fname  
FROM authors (a)  
WHERE 2 = (SELECT COUNT(*)  
           FROM titleauthor ta  
           WHERE ta.au_id = (a.au_id));
```

Sous-select fait référence au select principal



SOUS SELECT

Dans la partie condition :

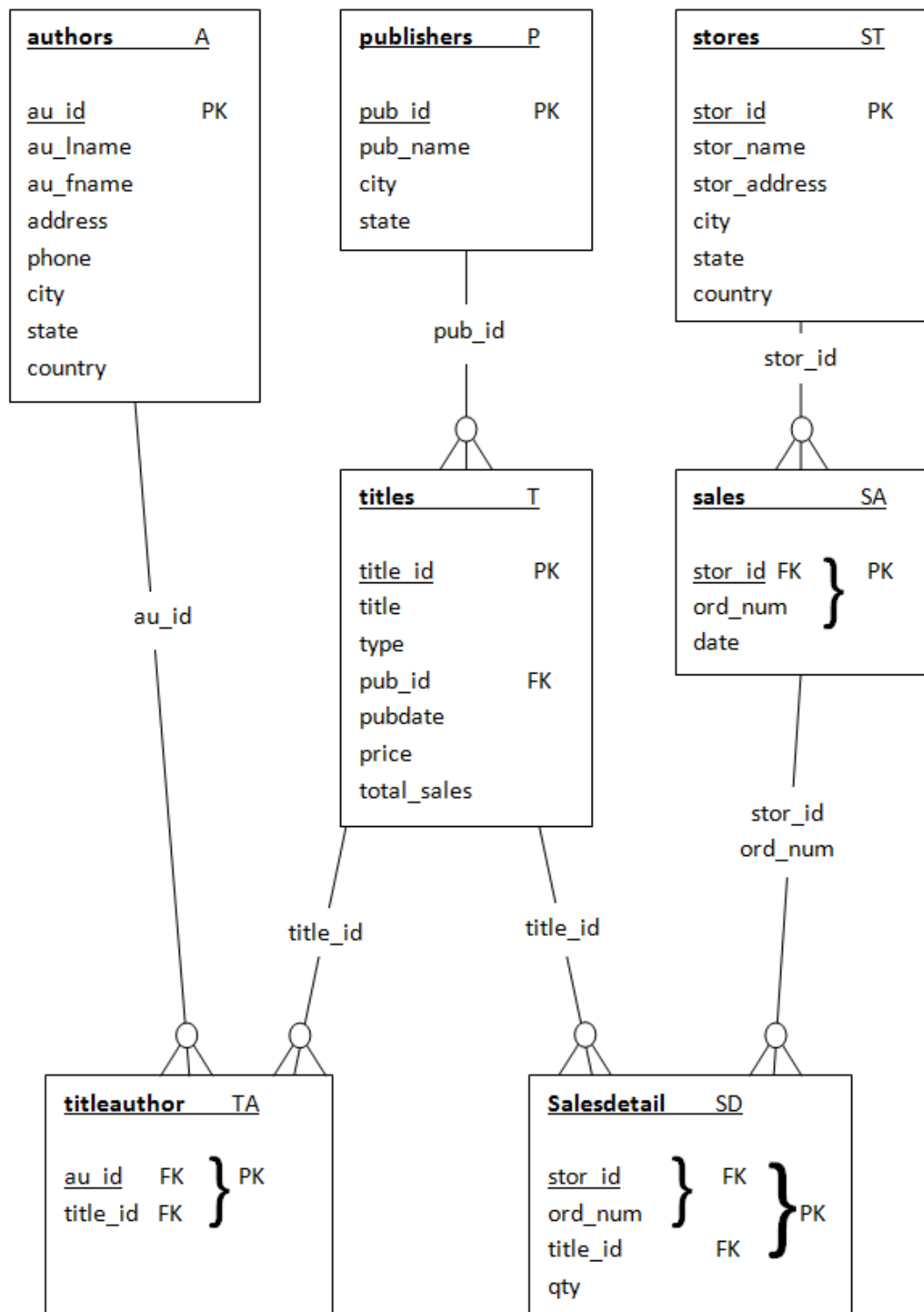
- `nom_attribut IN (SELECT ...)`
- `nom_attribut NOT IN (SELECT ...)`

où **(SELECT ...)** est un **SELECT** retournant une seule colonne

- la condition teste si `nom_attribut` se trouve dans un des tuples du **SELECT**

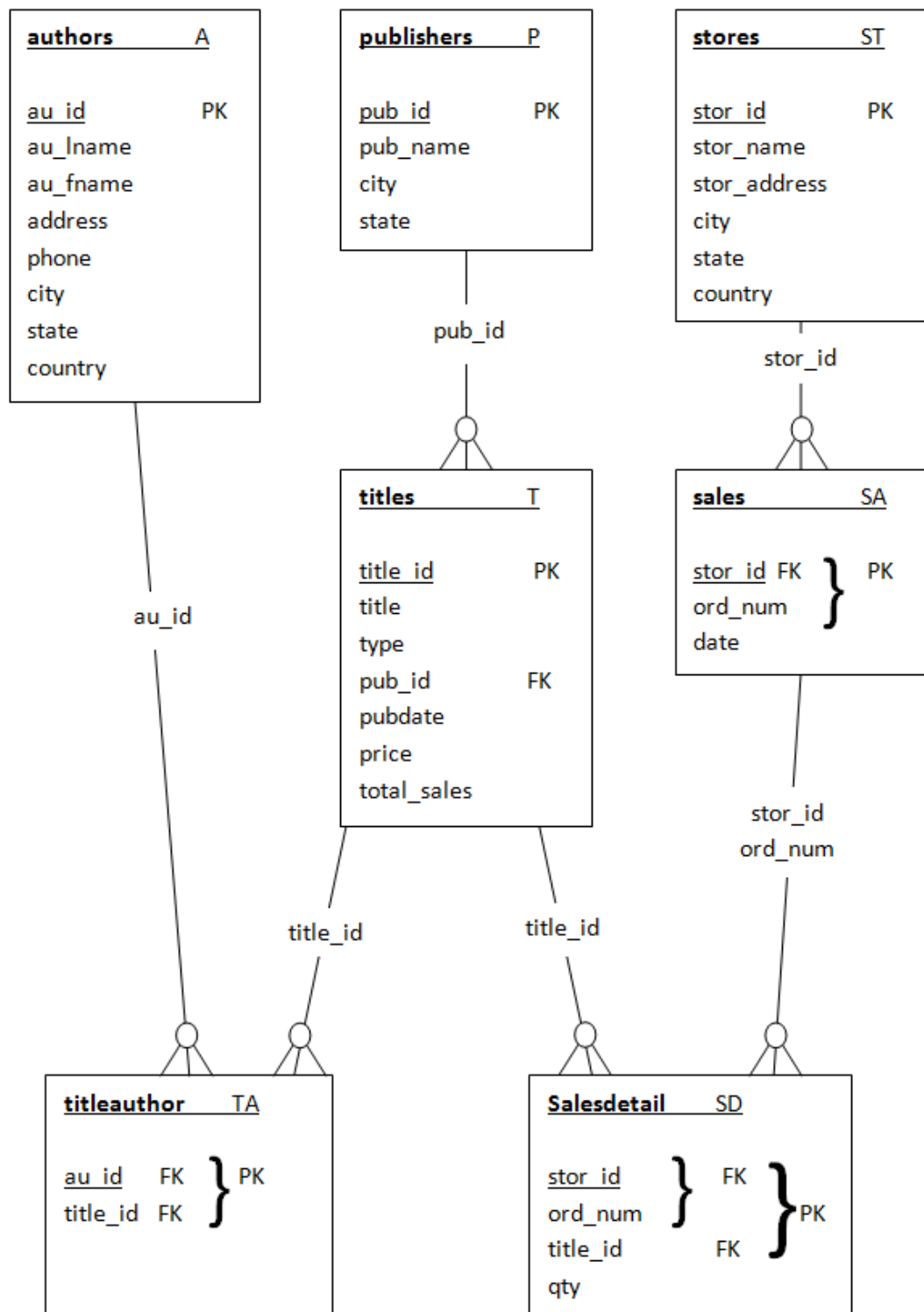
Les auteurs habitant dans le même état qu'un éditeur

```
SELECT DISTINCT  
  a.au_id, a.au_lname,  
  a.au_fname  
FROM authors a,  
  publishers p  
WHERE p.state=a.state
```



Les auteurs habitant dans le même état qu'un éditeur

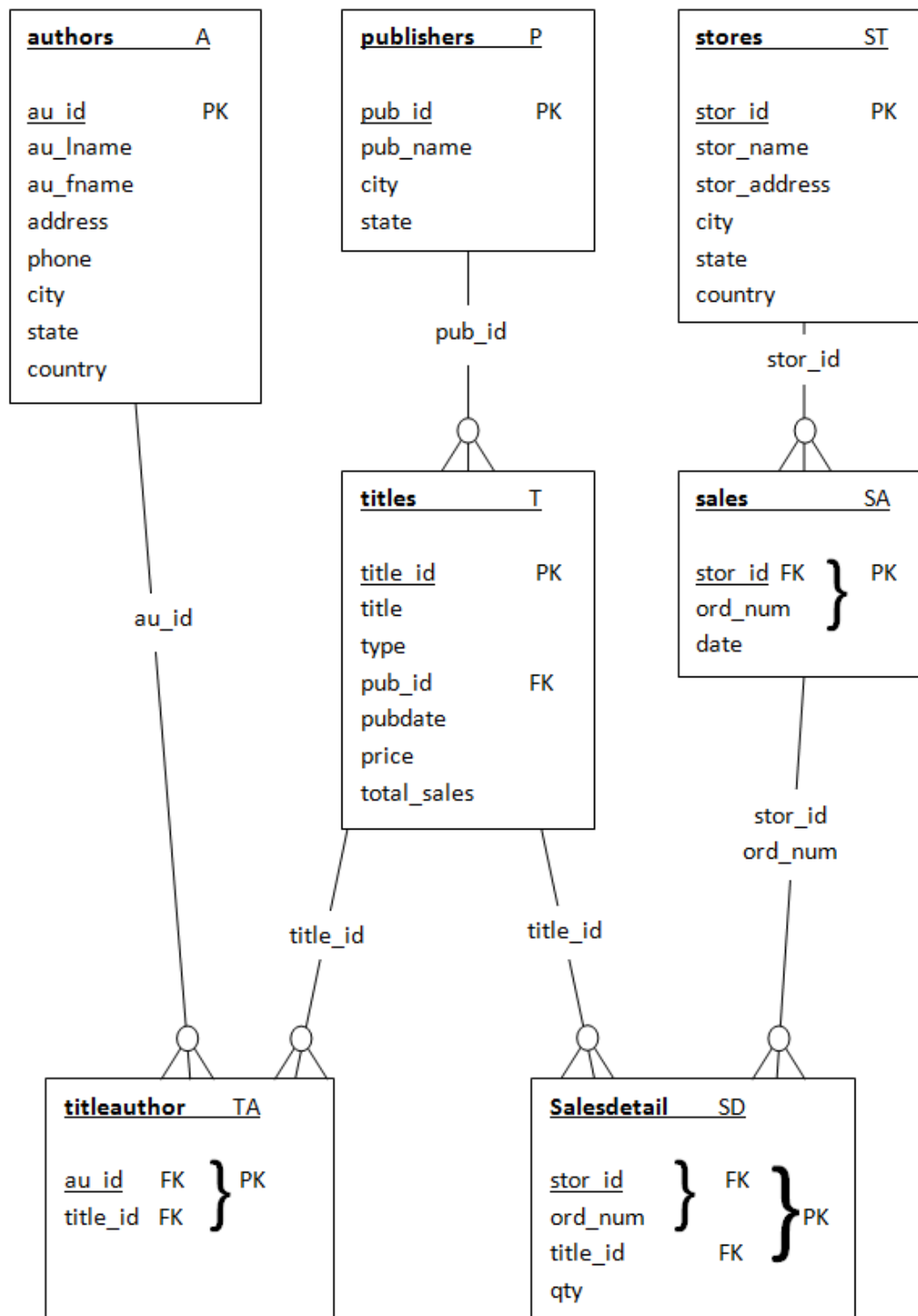
```
SELECT a.au_id,  
       a.au_lname, a.au_fname  
FROM authors a  
WHERE a.state IN  
      ( SELECT p.state  
        FROM publishers p);
```



Les auteurs habitant dans le même état qu'un éditeur

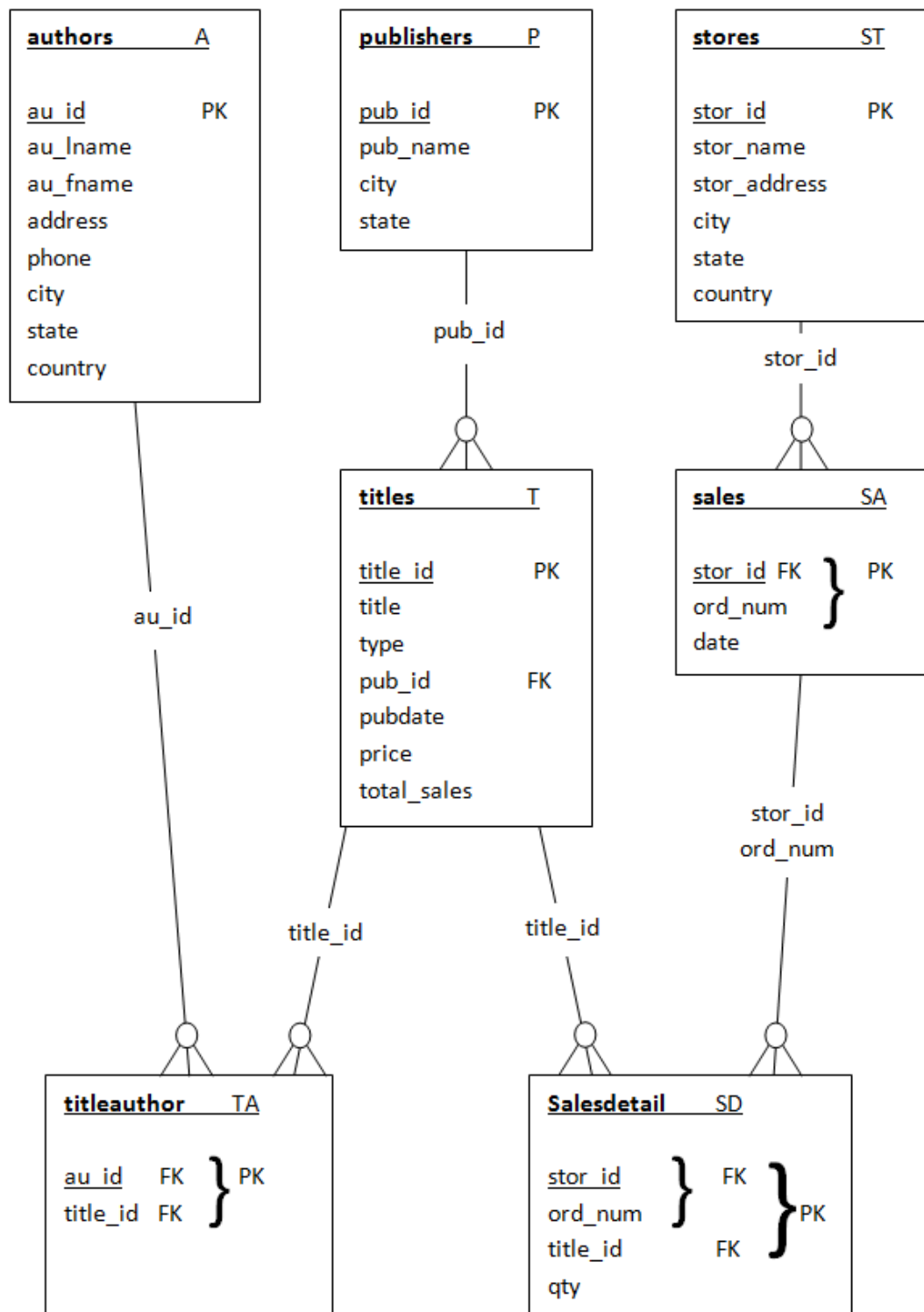
```
SELECT p.state  
FROM publishers p;  
⇒ MA, DC, CA
```

```
SELECT a.au_id,  
       a.au_lname,  
       a.au_fname  
FROM authors a  
WHERE a.state IN  
      ( 'MA' , 'DC' , 'CA' );
```



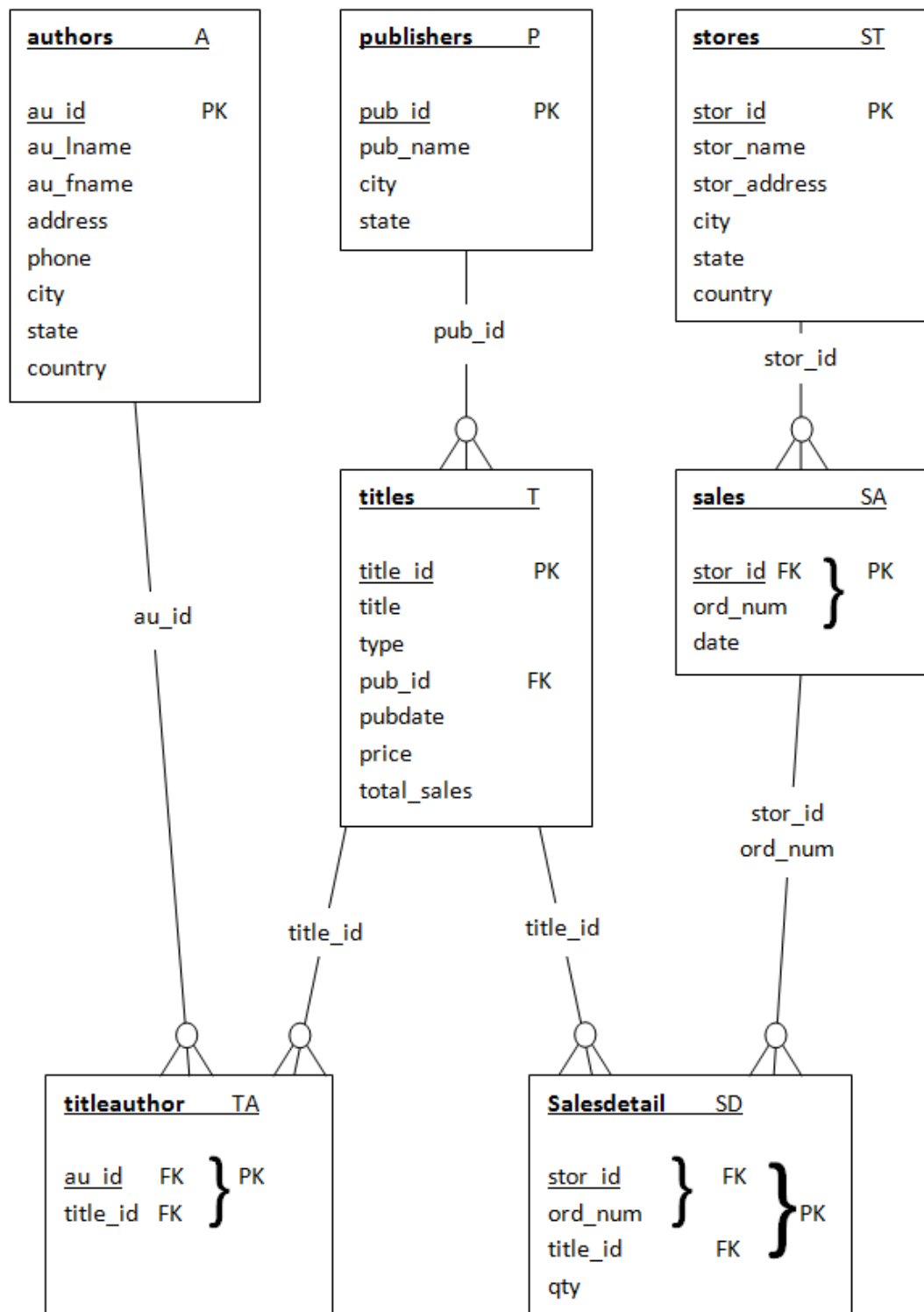
Les auteurs habitant dans le même état qu'un éditeur ayant publié un livre de business

```
SELECT DISTINCT a.au_id,  
                 a.au_lname, a.au_fname  
FROM authors a, publishers p, titles t  
WHERE p.pub_id = t.pub_id  
      AND a.state = p.state  
      AND t.type = 'business';
```



Les auteurs habitant dans le même état qu'un éditeur ayant publié un livre de business

```
SELECT a.au_id, a.au_lname,  
       a.au_fname  
FROM authors a  
WHERE a.state IN  
      ( SELECT p.state  
        FROM publishers p, titles t  
        WHERE p.pub_id = t.pub_id  
              AND t.type = 'business');
```



Les auteurs habitant dans le même état qu'un éditeur ayant publié un livre de business

```
SELECT a.au_id, a.au_lname,  
       a.au_fname
```

```
FROM authors a
```

```
WHERE a.state IN (
```

```
    SELECT p.state
```

```
    FROM publishers p
```

```
    WHERE p.pub_id IN (
```

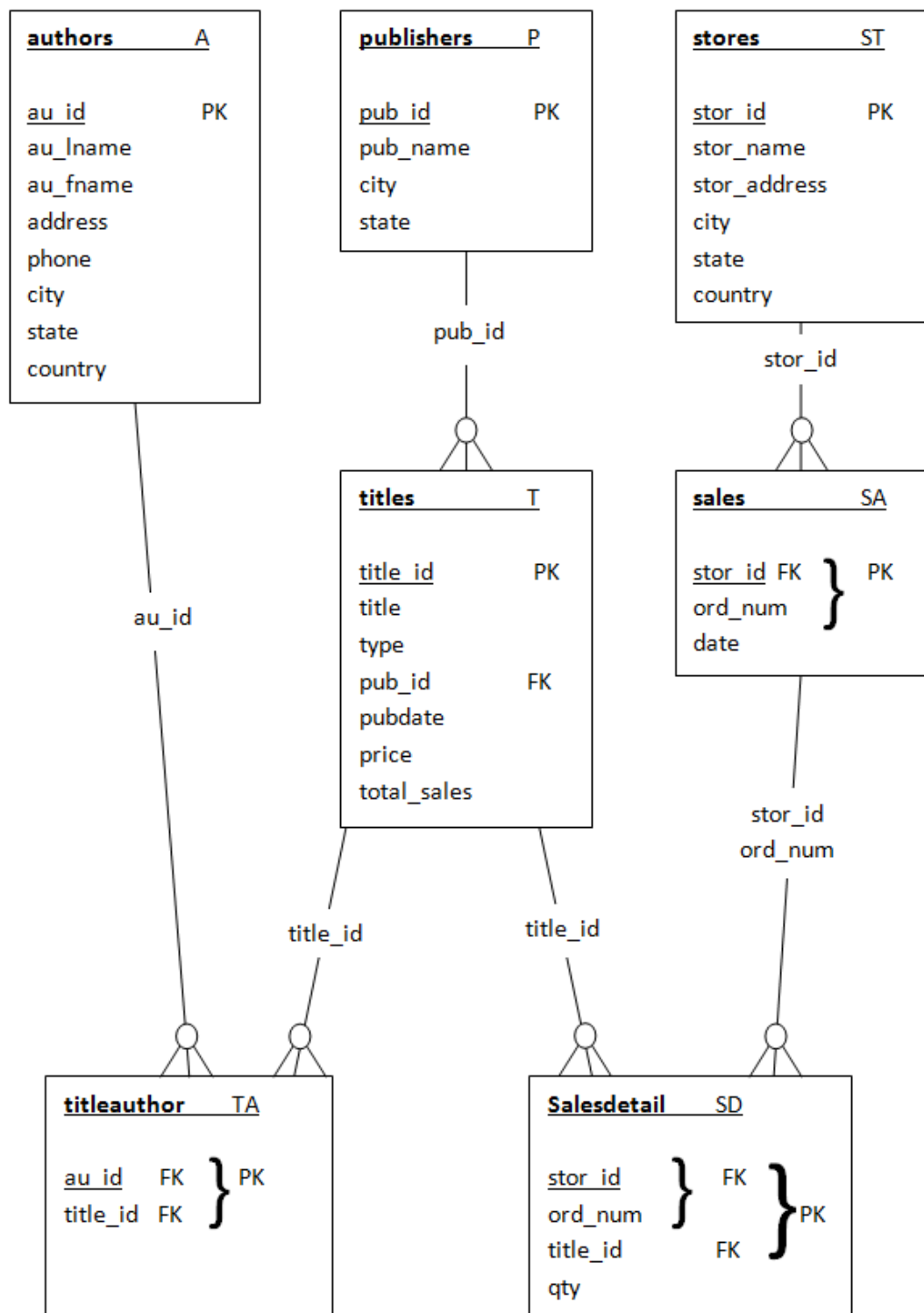
```
        SELECT t.pub_id
```

```
        FROM titles t
```

```
        WHERE t.type = 'business')
```

```
);
```

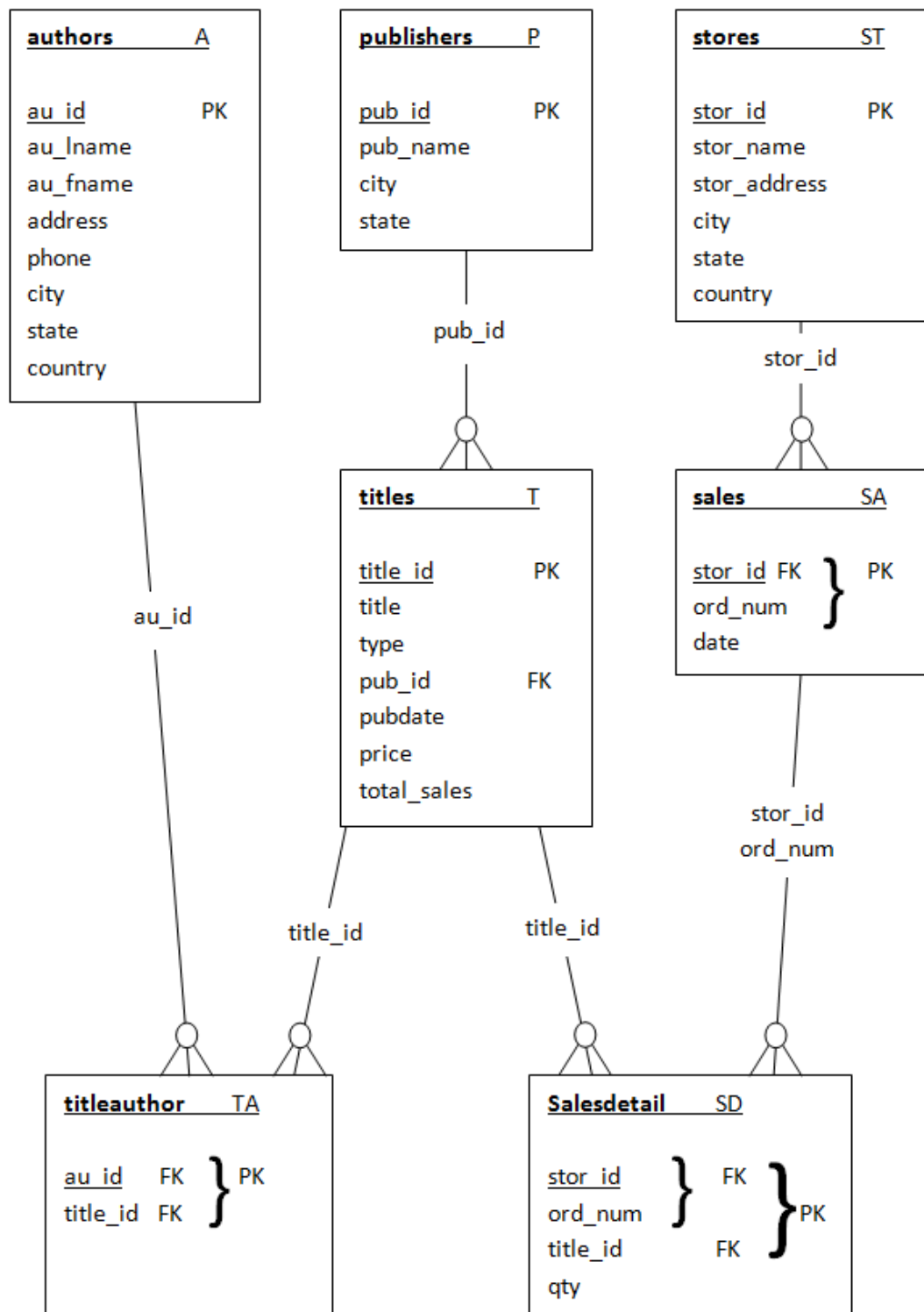
IN et jointures sont interchangeables



Les auteurs habitant dans un état sans éditeur

```
SELECT a.au_id,  
       a.au_lname, a.au_fname  
FROM authors a  
WHERE a.state NOT IN (  
      SELECT p.state  
      FROM publishers p)  
;
```

Impossible avec des
jointures !



SOUS SELECT

Dans la partie condition :

- conditionelementaire =
expression (= | <> | < | > | <= | >=) **ANY**(SELECT ...)
- conditionelementaire =
expression (= | <> | < | > | <= | >=) **ALL**(SELECT ...)

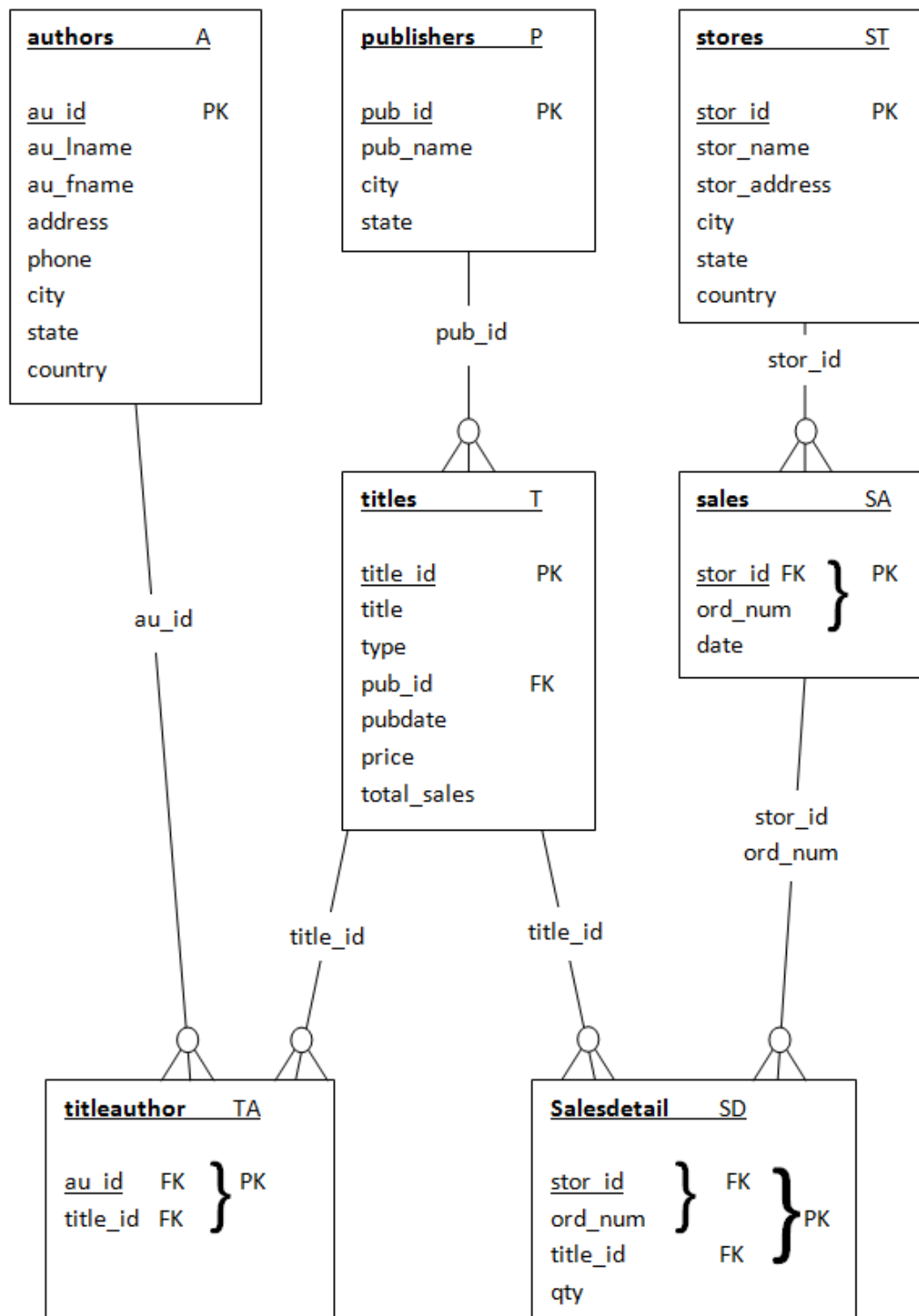
où (SELECT ...) est un SELECT retournant une seule colonne

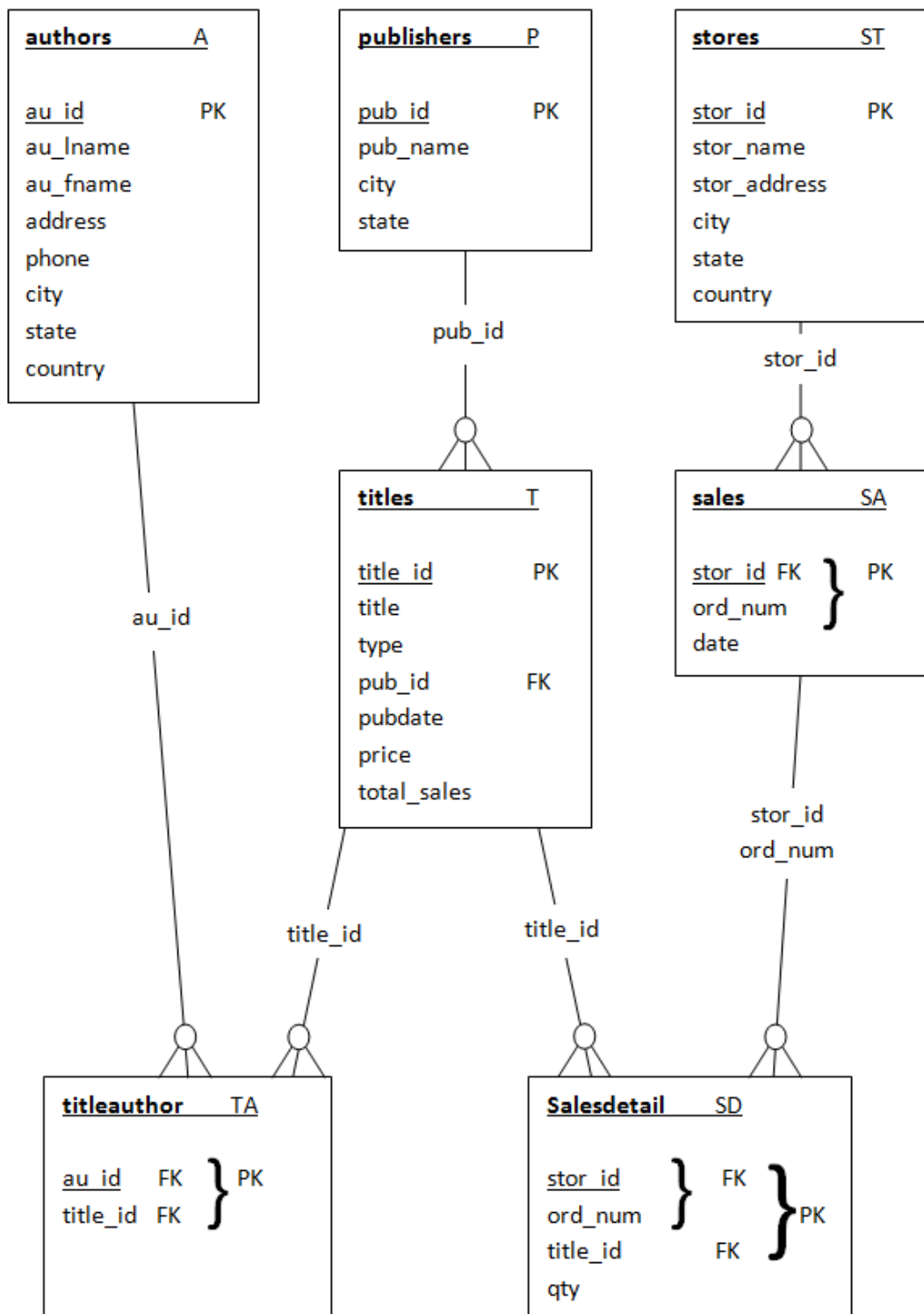
- la condition teste si l'opérateur est vrai pour au moins un (ANY) ou tous (ALL) les éléments du sous-select.

Les auteurs habitant dans le même état qu'un éditeur

```
SELECT a.au_id,  
       a.au_lname, a.au_fname  
FROM authors a  
WHERE a.state = ANY (  
      SELECT p.state  
      FROM publishers p)  
;
```

IN d'un sous-select est équivalent
à = ANY





Livre le plus cher

```

SELECT t1.title_id, t1.title
FROM titles t1
WHERE t1.price =
      ( SELECT MAX(t2.price)
        FROM titles t2) ;
  
```

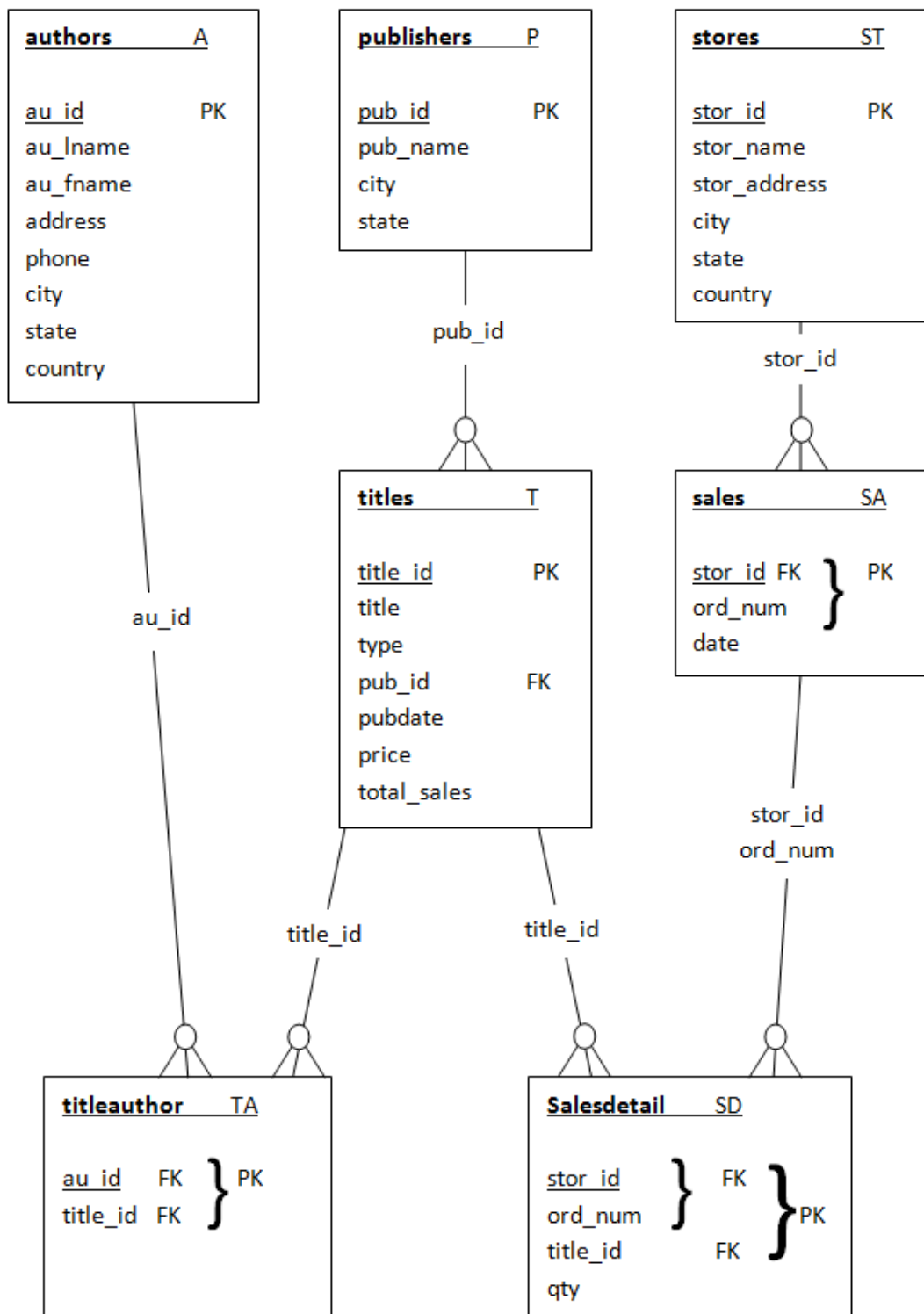
Résultat : PC1035

```

SELECT t1.title_id, t1.title
FROM titles t1
WHERE t1.price >=
      ALL (SELECT t2.price
           FROM titles t2)
;
  
```

Résultat : Ø

Pourquoi ? null ...



Livre le plus cher

```
SELECT t1.title_id, t1.title
FROM titles t1
WHERE t1.price =
```

```
( SELECT MAX(t2.price)
  FROM titles t2) ;
```

Résultat : PC1035

```
SELECT t1.title_id, t1.title
FROM titles t1
WHERE t1.price >=
```

```
ALL (SELECT t2.price
     FROM titles t2
```

```
WHERE t2.price IS NOT NULL) ;
```

Résultat : PC1035

SOUS SELECT

WITH nom_table_temporaire AS (SELECT ...)

SELECT ...

où (SELECT ...) est un SELECT sans condition particulière

- **L'objectif du WITH est de décomposer des requêtes complexes en requêtes plus simples**
- **La clause WITH permet de créer des tables temporaires utilisables dans la requête principale**

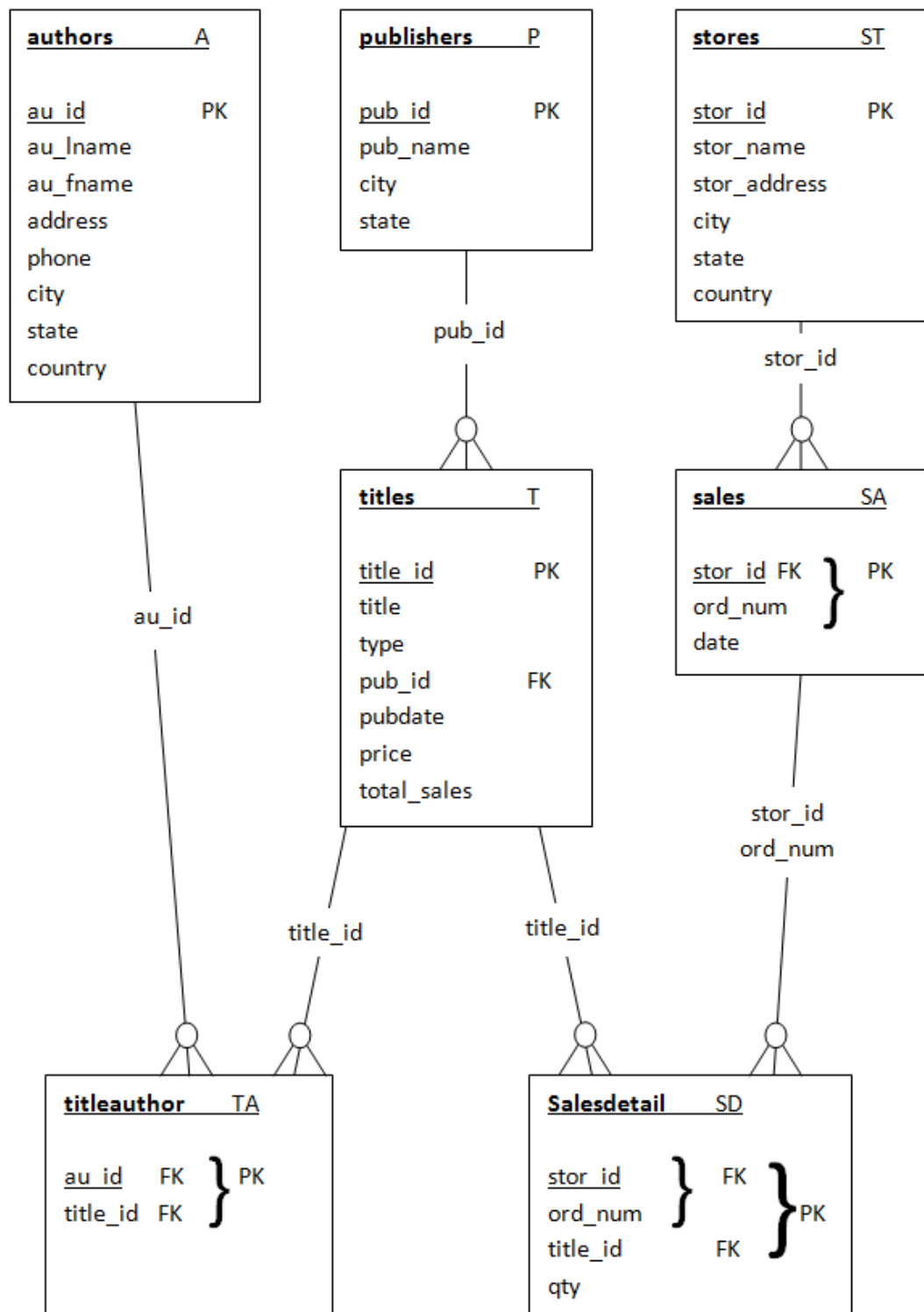
Quelles sont les villes
de Californie où l'on
peut trouver un
auteur, mais aucun
magasin ?

WITH californiaStores AS
(SELECT st.* FROM stores
st WHERE st.state = 'CA')

SELECT DISTINCT a.city

FROM authors a

WHERE a.state = 'CA' AND
a.city NOT IN (SELECT
c.city FROM
californiaStores c);



SOUS SELECT

Dans la partie condition :

- EXISTS (SELECT ...)
- NOT EXISTS (SELECT ...)

où (SELECT ...) est un SELECT sans condition particulière

- la condition vérifie si le sous-select est vide ou pas.
- comme la présence d'un tuple est suffisante, inutile de SELECT des colonnes en particulier : SELECT *

Les auteurs habitant dans le même état qu'un éditeur

```
SELECT a.au_id,  
       a.au_lname,  
       a.au_fname
```

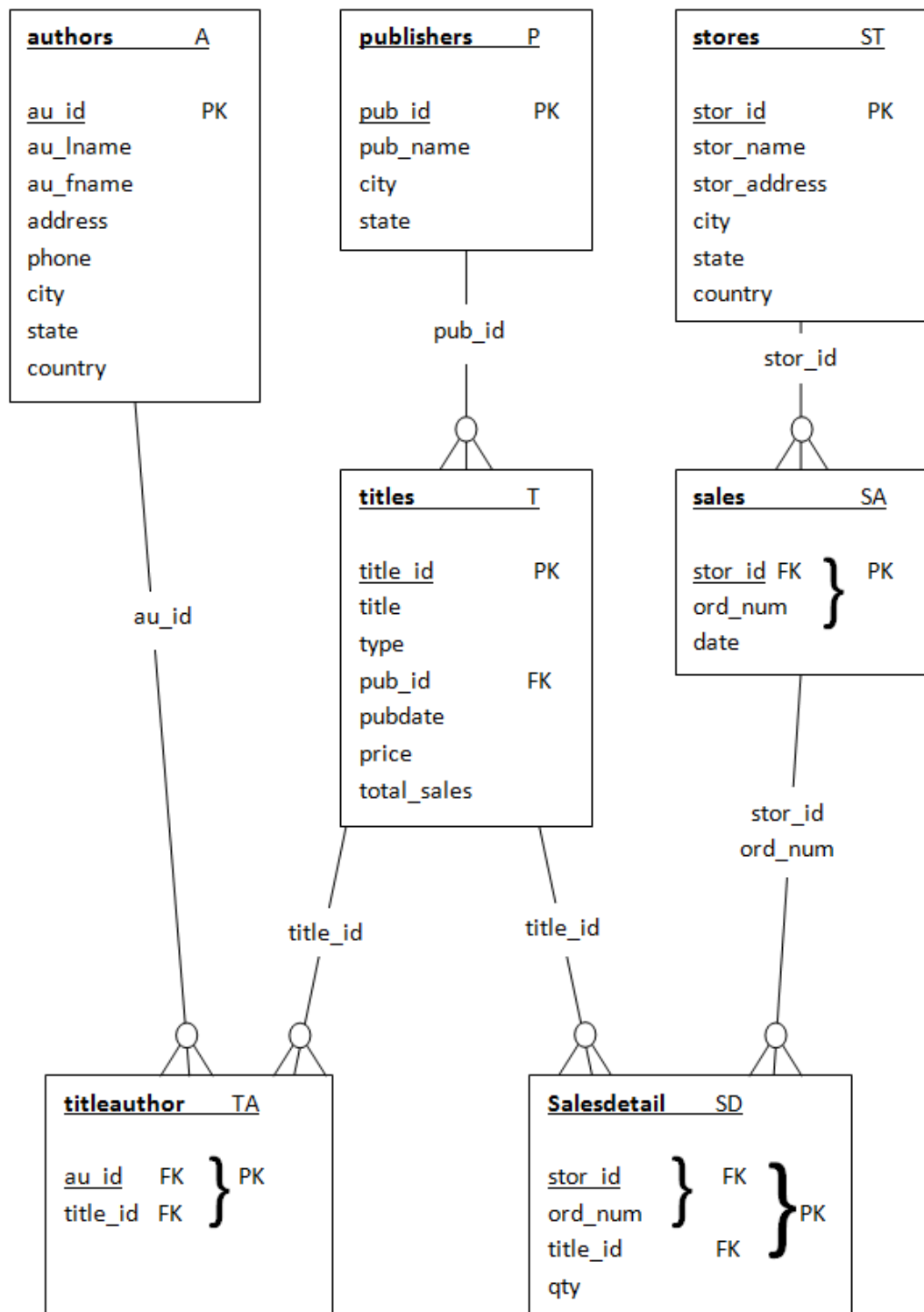
```
FROM authors a
```

```
WHERE EXISTS (
```

```
  SELECT *
```

```
  FROM publishers p
```

```
  WHERE p.state =  
         a.state)
```



Les auteurs habitant dans un état sans éditeur

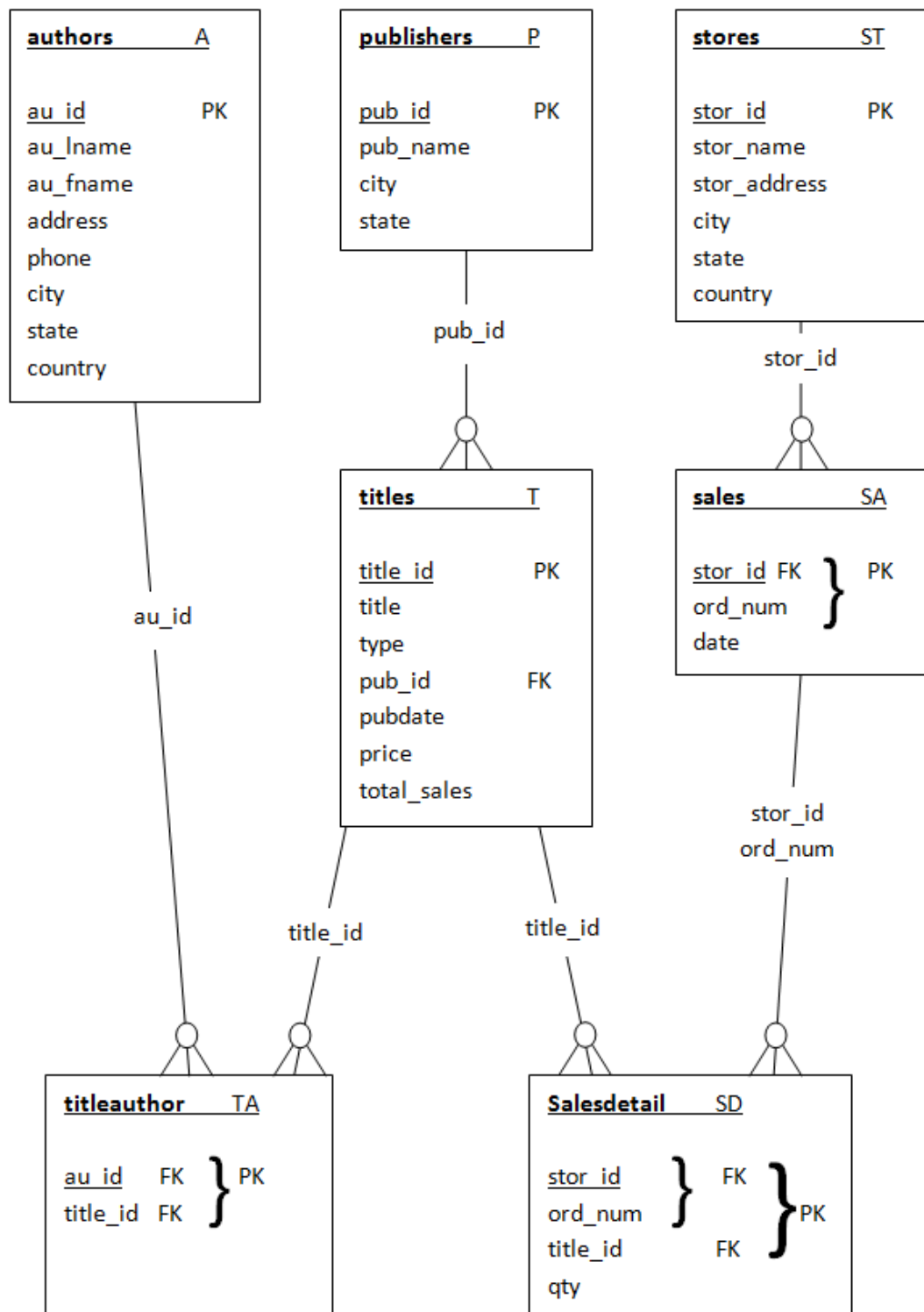
```
SELECT a.au_id,  
       a.au_lname,  
       a.au_fname
```

```
FROM authors a  
WHERE NOT EXISTS (
```

```
  SELECT *
```

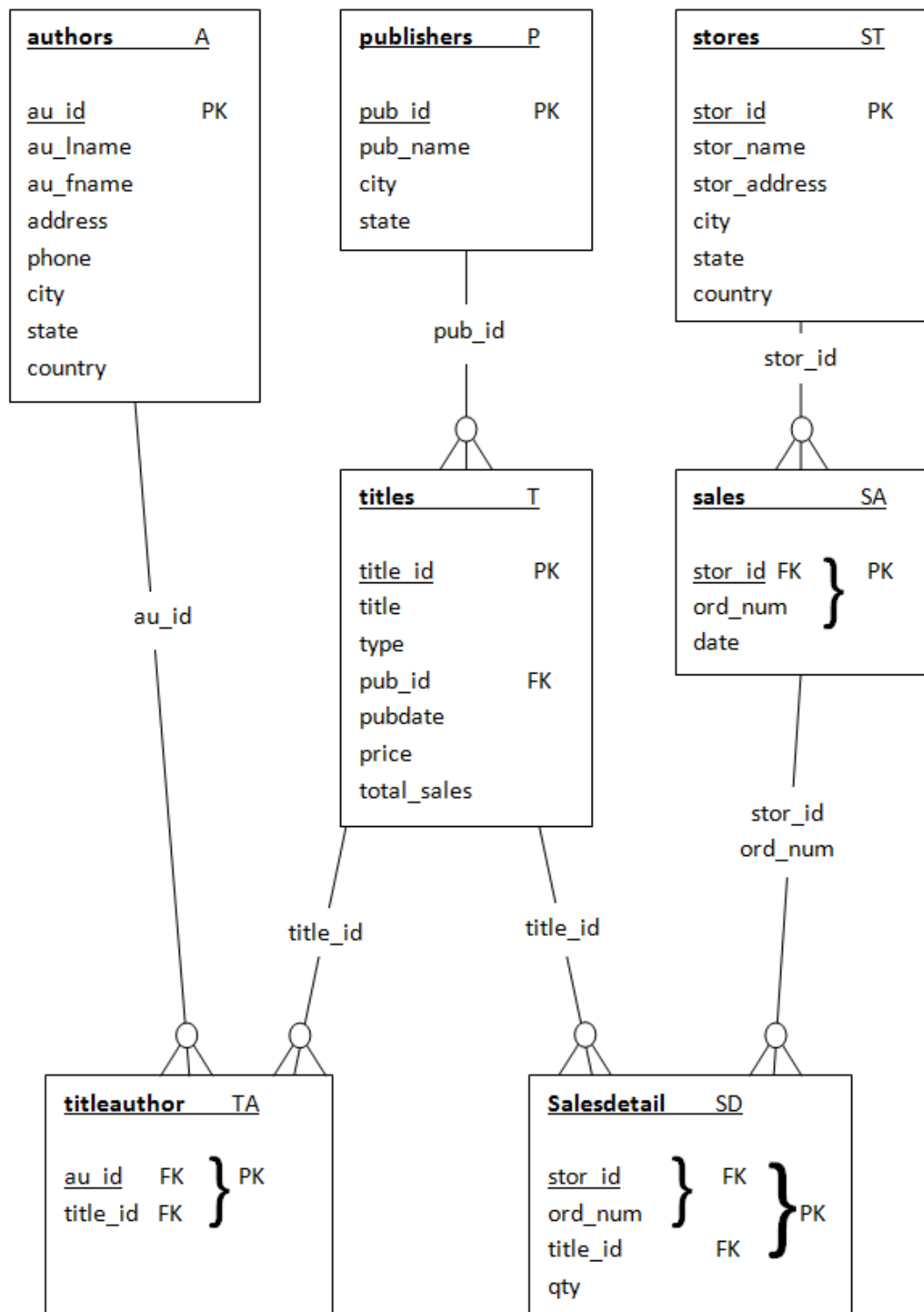
```
  FROM publishers p
```

```
  WHERE p.state =  
         a.state)
```



Les magasins qui vendent tous les livres édités par Algodata ?

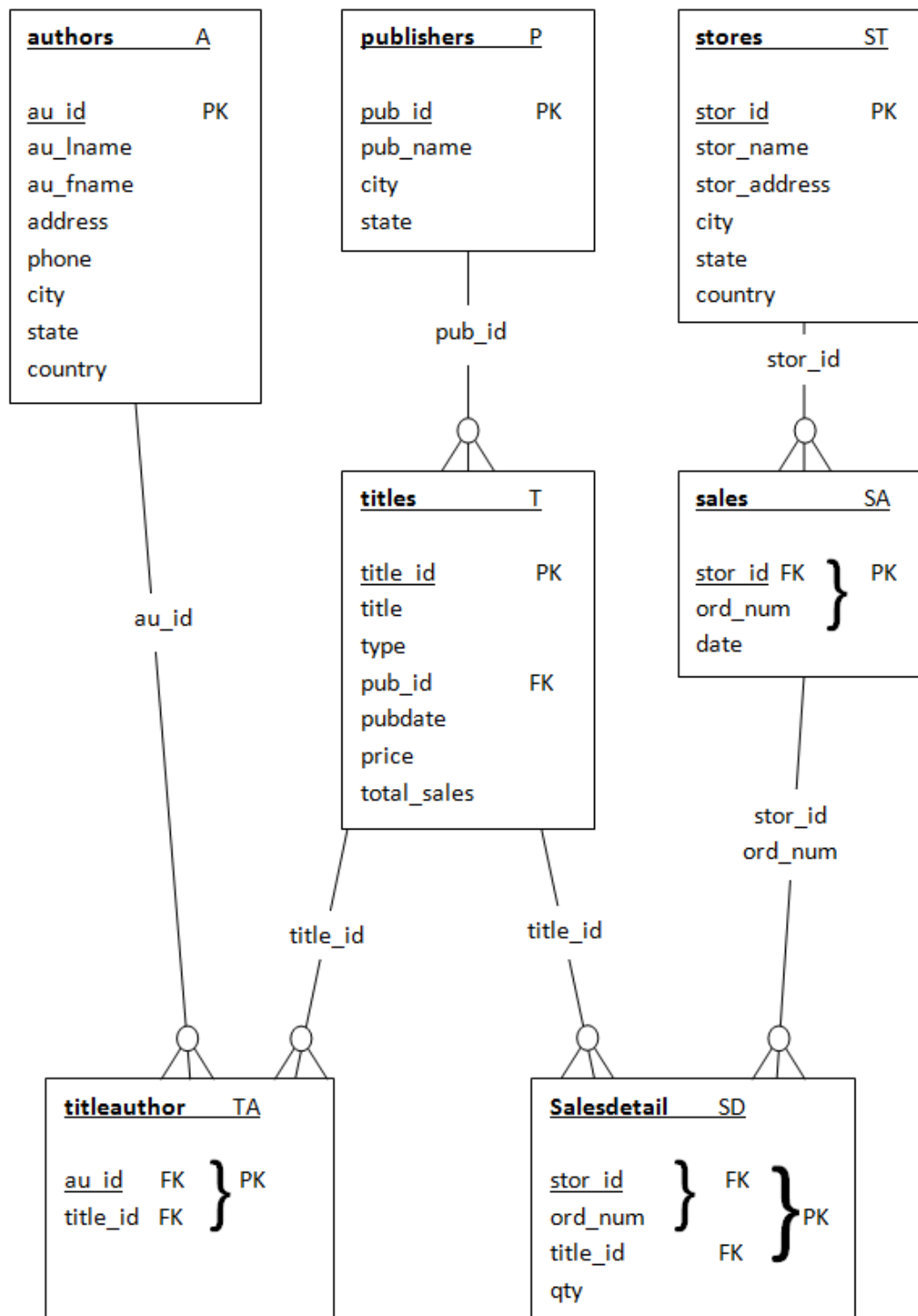
On ne peut pas
traduire directement
en SQL !



Les magasins qui
vendent tous les
livres édités par
Algodata ?

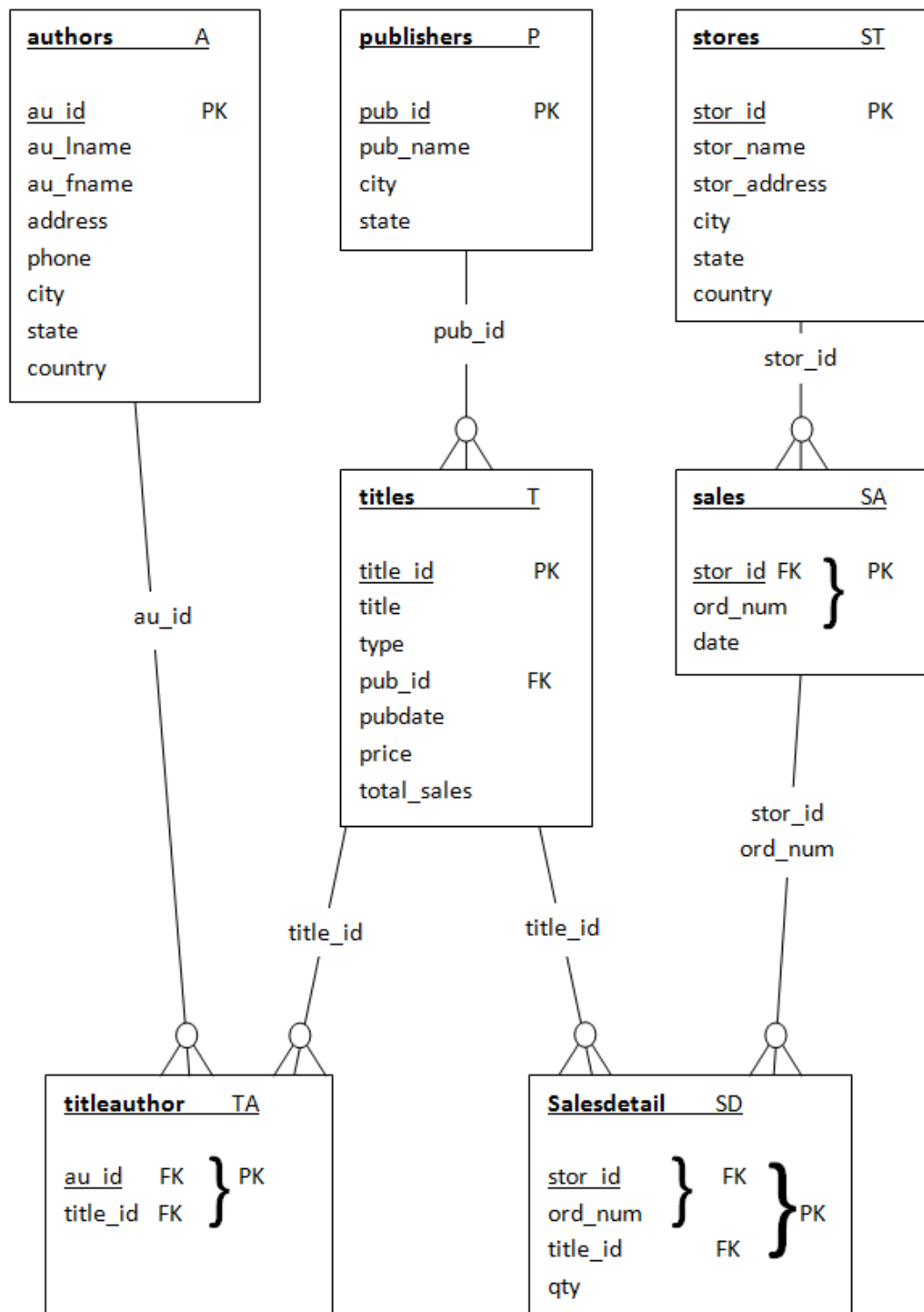
Reformulation :

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
    )
);
```



UNION, INTERSECTION, DIFFERENCE

(SELECT ...) UNION (SELECT ...)

- Les deux selects doivent produire le même nombre de colonnes de types compatibles.
- Effectue l'union ensembliste

(SELECT ...) INTERSECT (SELECT ...)

- Les deux selects doivent produire le même nombre de colonnes de types compatibles.
- Effectue l'intersection ensembliste

(SELECT ...) EXCEPT (SELECT ...)

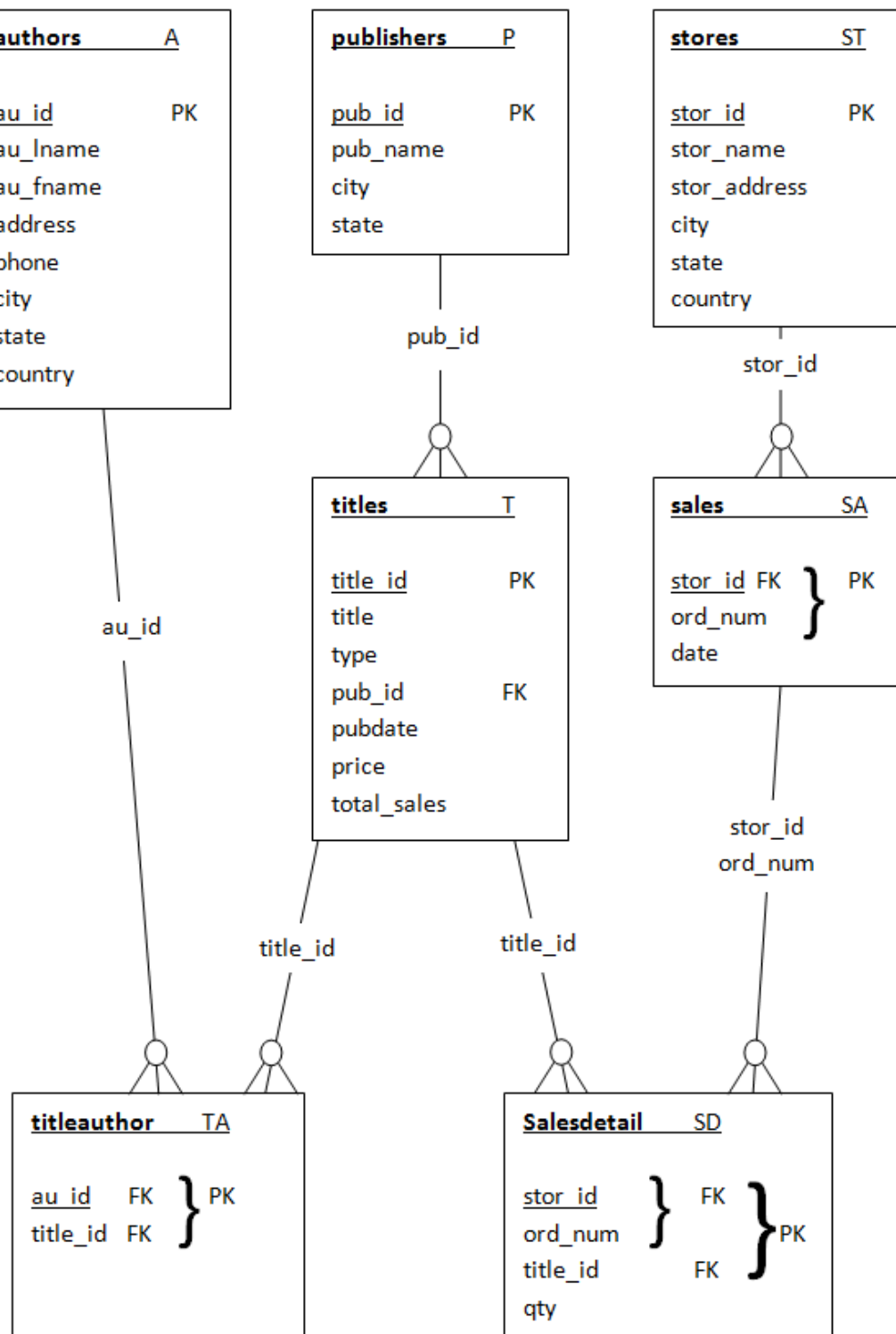
- Les deux selects doivent produire le même nombre de colonnes de types compatibles.
- Effectue la différence

Les villes dans lesquelles il y a un auteur et/ou un éditeur ?

(SELECT DISTINCT a.city
FROM authors a)

UNION

(SELECT DISTINCT p.city
FROM publishers p) ;

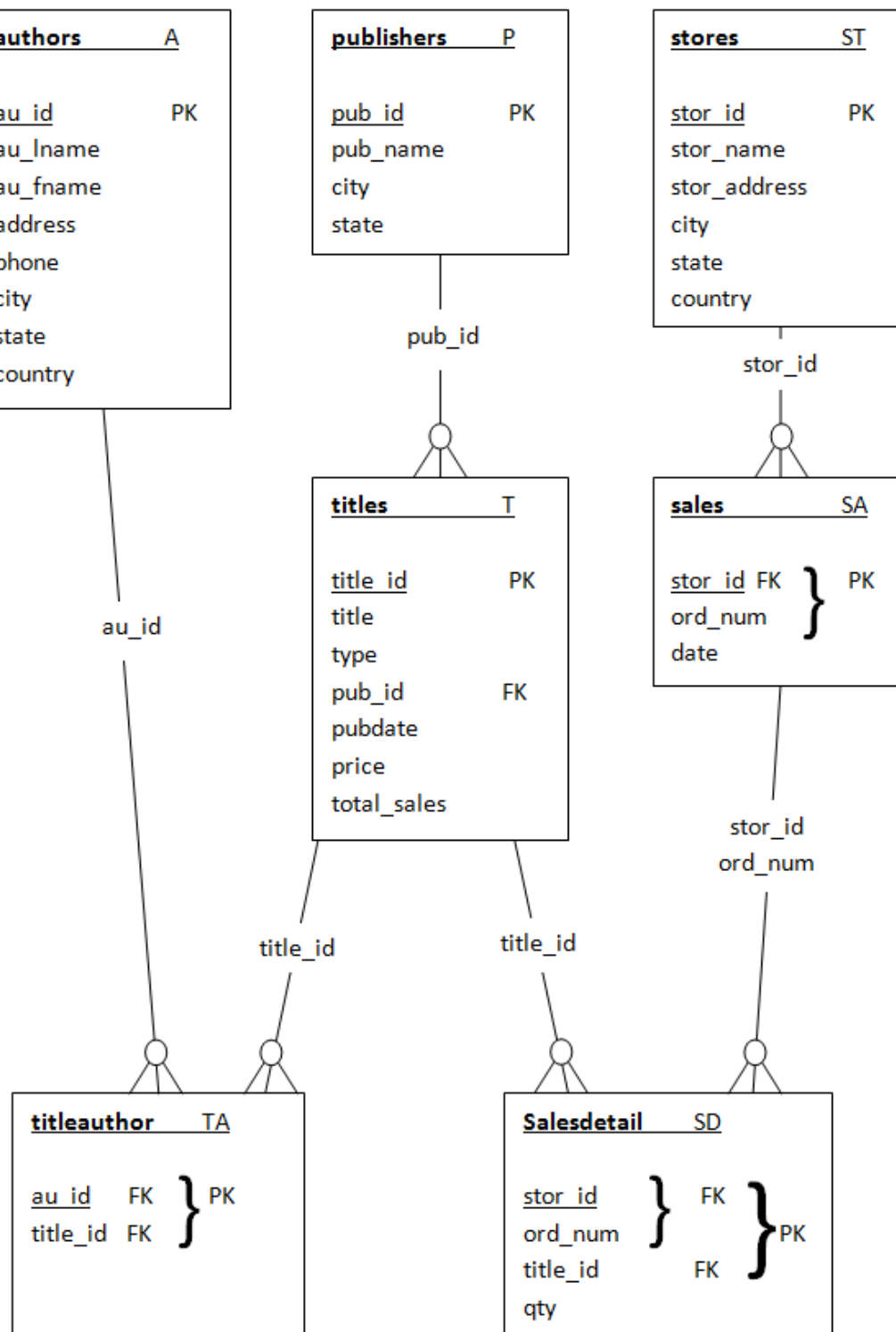


Les villes dans lesquelles il y a un auteur et un éditeur ?

(SELECT DISTINCT a.city
FROM authors a)

INTERSECT

(SELECT DISTINCT p.city
FROM publishers p) ;

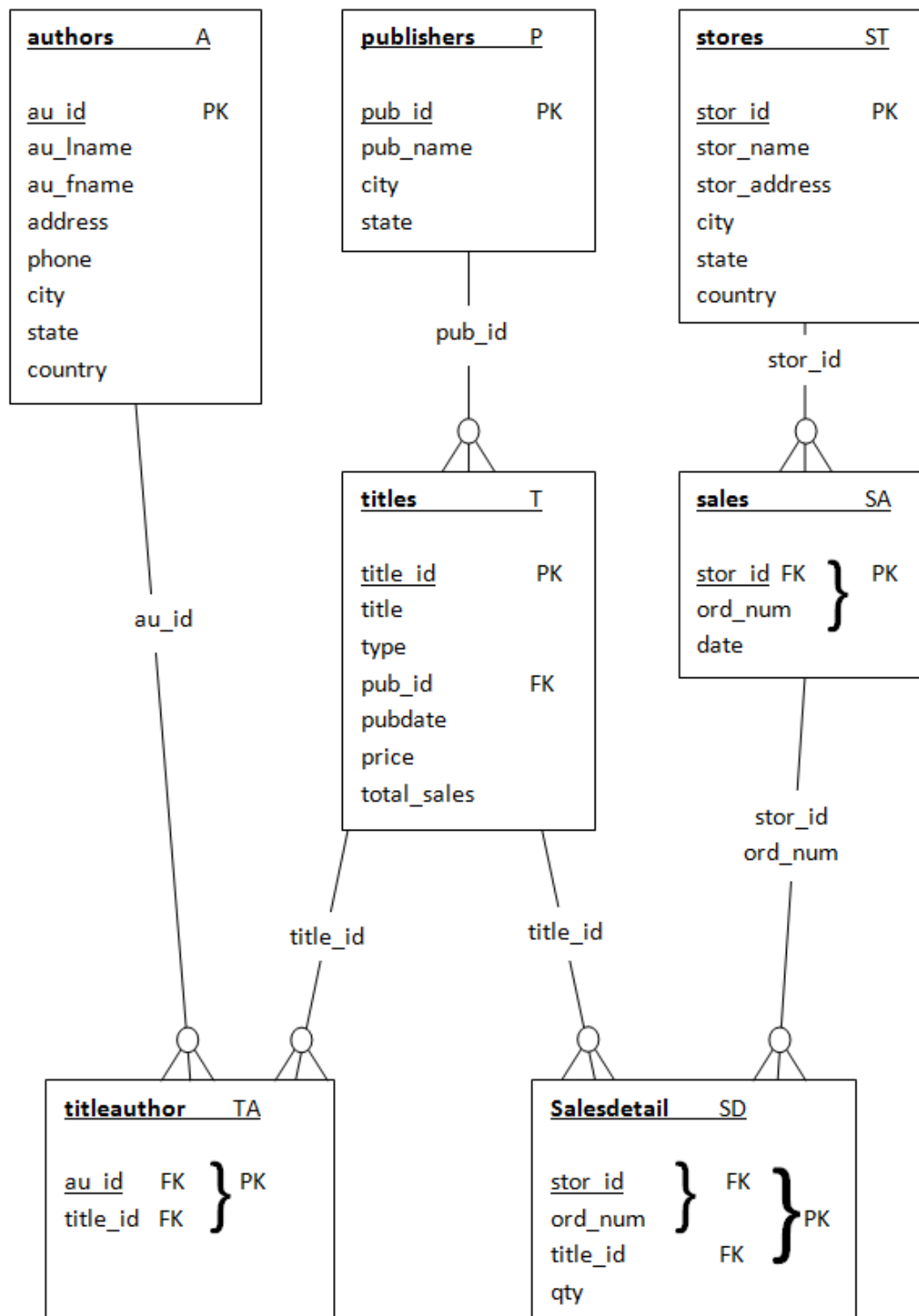


Les villes dans lesquelles il y a un auteur mais pas d'éditeur ?

(SELECT DISTINCT a.city
FROM authors a)

EXCEPT

(SELECT DISTINCT p.city
FROM publishers p) ;



JOIN

- **Dans la partie FROM d'un select**
 - `tableGauche INNER JOIN tableDroite ON condition`
 - `tableGauche LEFT OUTER JOIN tableDroite ON condition`
 - `tableGauche RIGHT OUTER JOIN tableDroite ON condition`

EXEMPLE

- **Afficher tous les titres de livre avec leurs éditeurs**

titre	num_editeur
Les bleus	2
Les rouges	3
Les verts	2
Les blancs	2
Les noirs	5

num_editeur	nom_editeur
1	Bordas
2	Springer
3	Larousse

JOINTURE INTERNE

```
SELECT titre, num_editeur, nom_editeur  
FROM Livres l, Editeurs e  
WHERE l.num_editeur=e.num_editeur
```

Qui est équivalent à

```
SELECT titre, num_editeur, nom_editeur  
FROM Livres l INNER JOIN Editeurs e  
ON l.num_editeur=e.num_editeur
```


JOINTURE INTERNE

titre	num_editeur
Les bleus	2
Les rouges	3
Les verts	2
Les blancs	2
Les noirs	5

num_editeur	nom_editeur
1	Bordas
2	Springer
3	Larousse

titre	num_editeur	nom_editeur
Les bleus	2	Springer
Les rouges	3	Larousse
Les verts	2	Springer
Les blancs	2	Springer

JOINTURE EXTERNE A GAUCHE

```
SELECT titre, num_editeur, nom_editeur  
FROM Livres l LEFT OUTER JOIN Editeurs e  
ON l.num_editeur=e.num_editeur
```

titre	num_editeur
Les bleus	2
Les rouges	3
Les verts	2
Les blancs	2
Les noirs	5

num_editeur	nom_editeur
1	Bordas
2	Springer
3	Larousse

titre	num_editeur	nom_editeur
Les bleus	2	Springer
Les rouges	3	Larousse
Les verts	2	Springer
Les blancs	2	Springer
Les noirs	5	NULL

JOINTURE EXTERNE A DROITE

```
SELECT titre, num_editeur, nom_editeur  
FROM Livres l RIGHT OUTER JOIN Editeurs e  
ON l.num_editeur=e.num_editeur
```

titre	num_editeur
Les bleus	2
Les rouges	3
Les verts	2
Les blancs	2
Les noirs	5

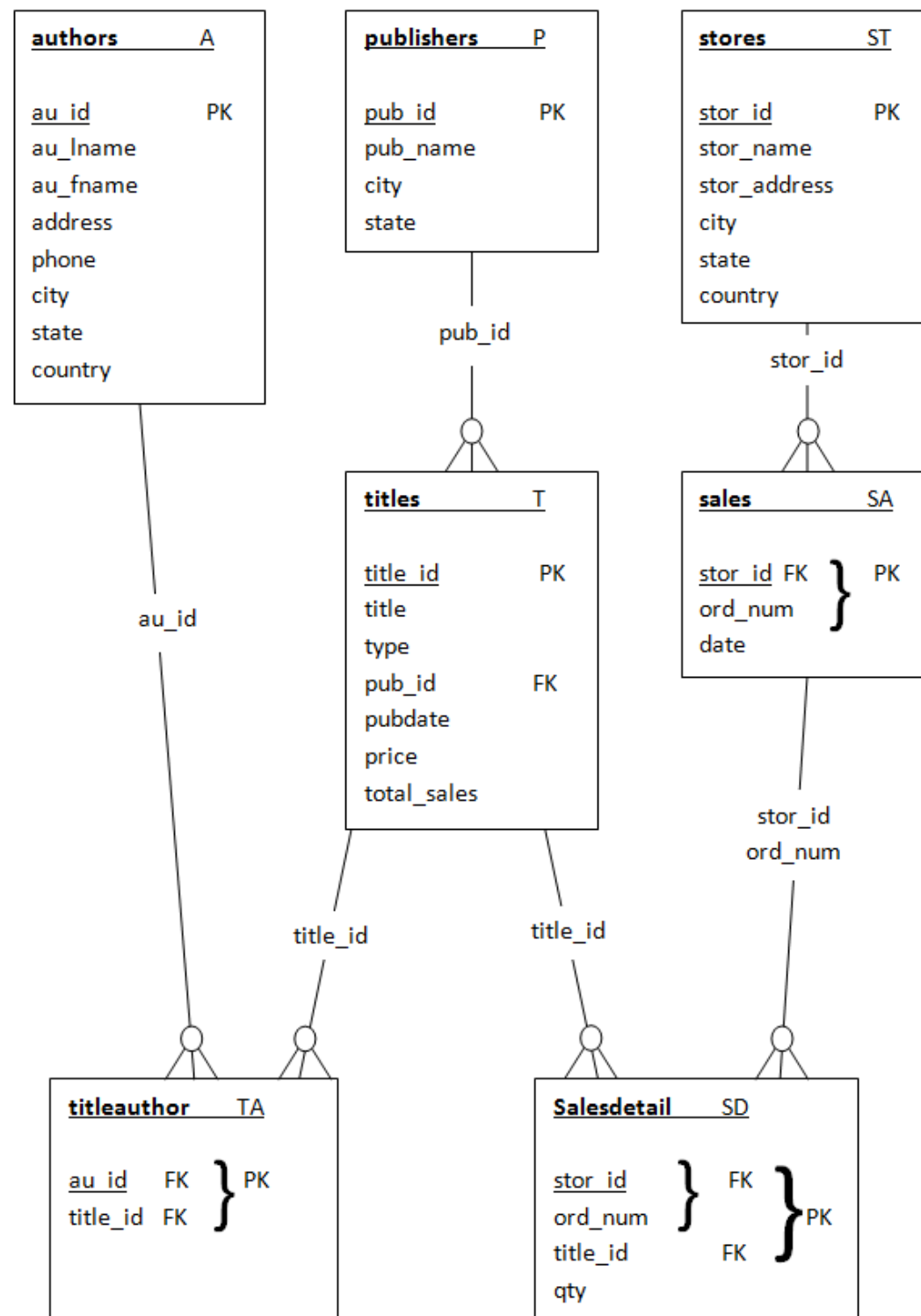
num_editeur	nom_editeur
1	Bordas
2	Springer
3	Larousse

titre	num_editeur	nom_editeur
Les bleus	2	Springer
Les rouges	3	Larousse
Les verts	2	Springer
Les blancs	2	Springer
NULL	1	Bordas

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

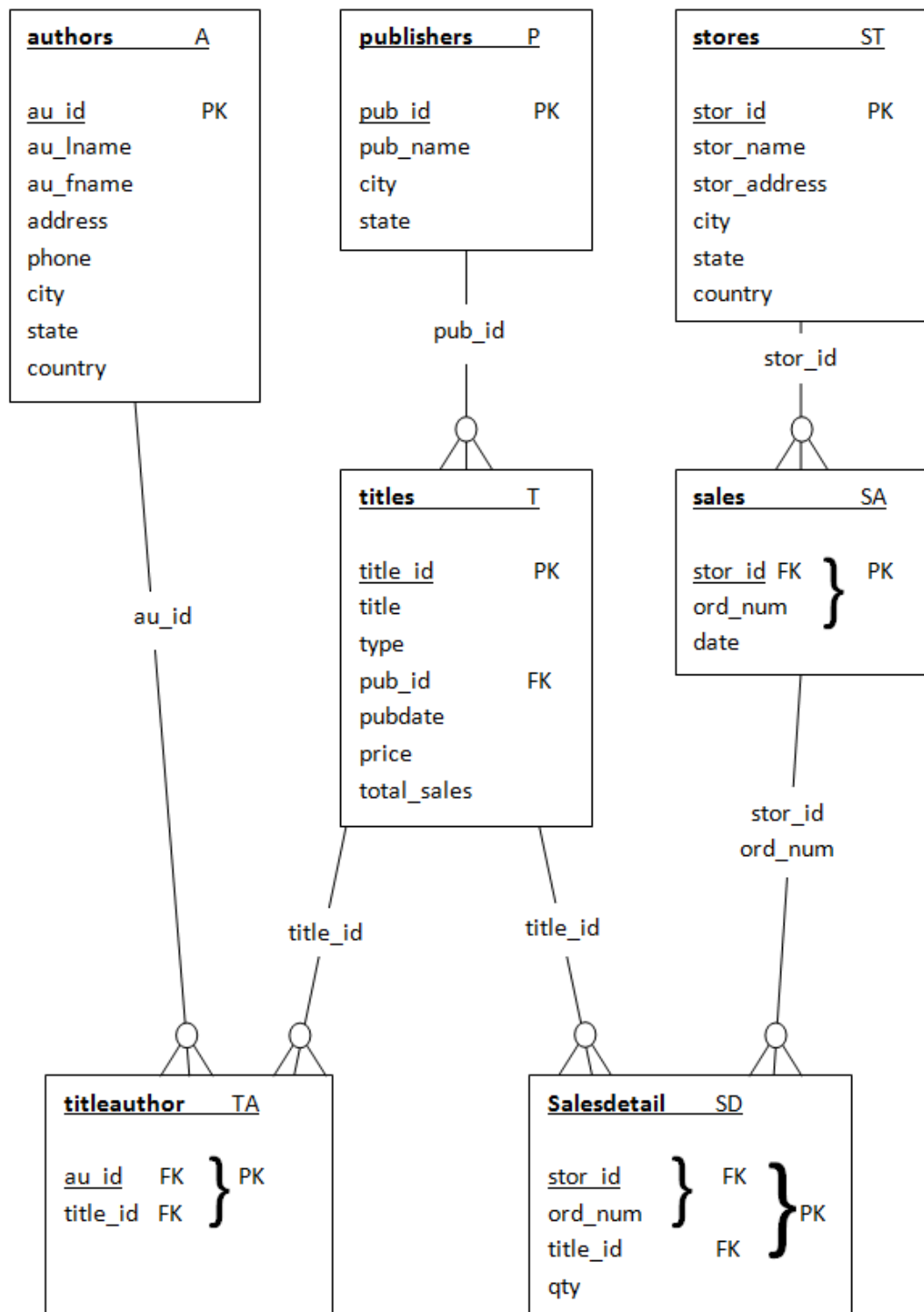


Tous les livres avec leurs quantités vendues

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

Retourne 18 !

Où sont les 2 livres manquants
?



Tous les livres avec leurs quantités vendues

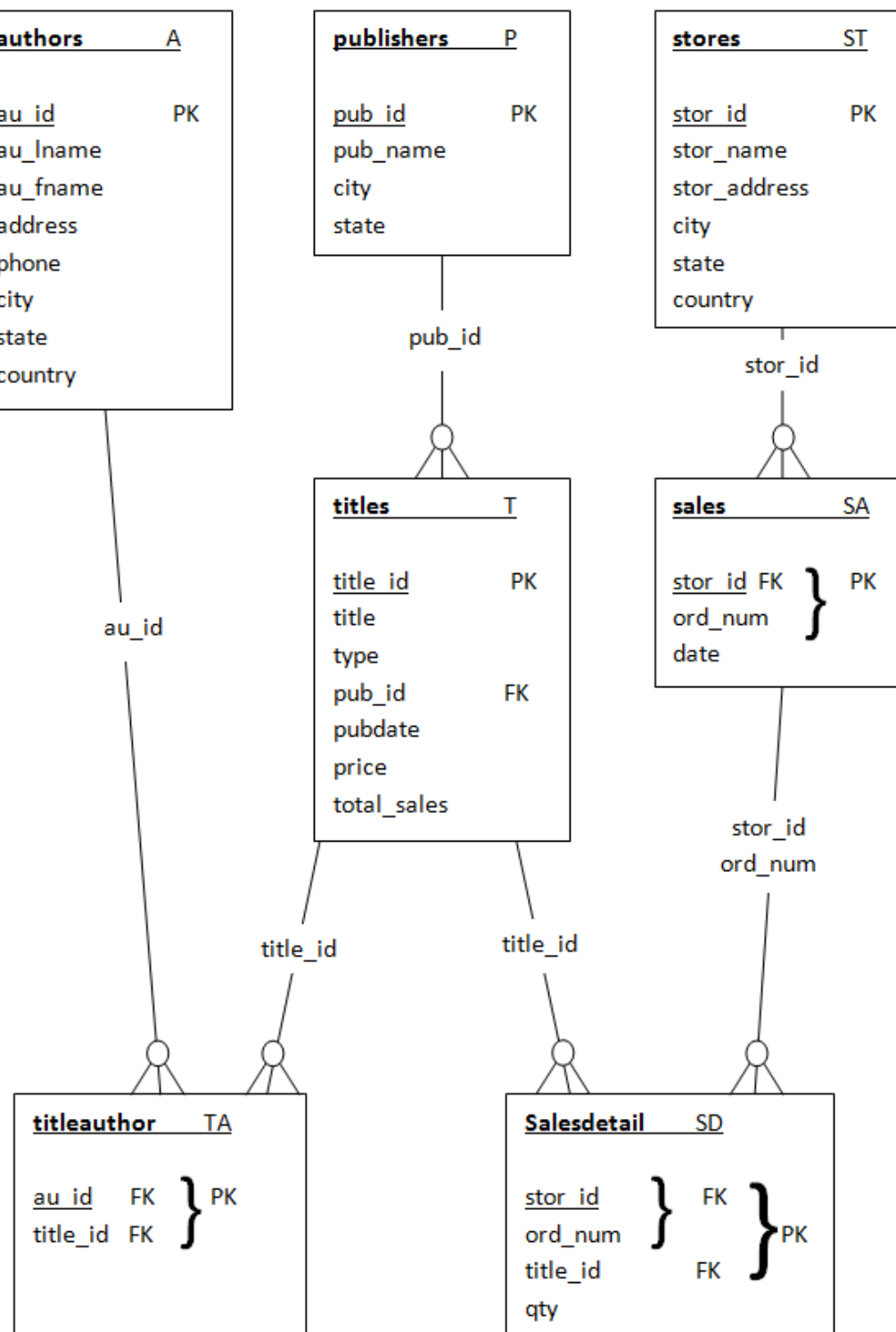
```
SELECT t.title, SUM(sd.qty)
```

```
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



COALESCE

Remplace une valeur NULL par une valeur conventionnelle (par exemple la valeur 0 ou la chaîne vide)

```
SELECT t.title, COALESCE(SUM(sd.qty), 0)
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	0
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