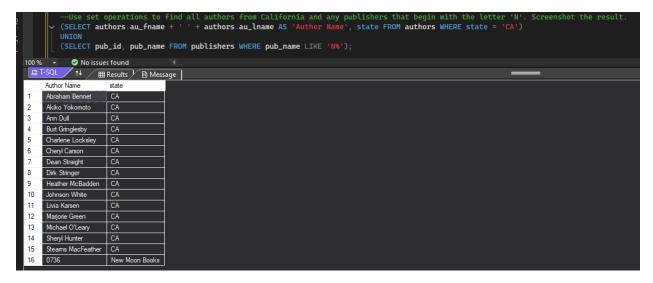
8. Use set operations to find all authors from California and any publishers that begin with the letter 'N'. Screenshot the result.



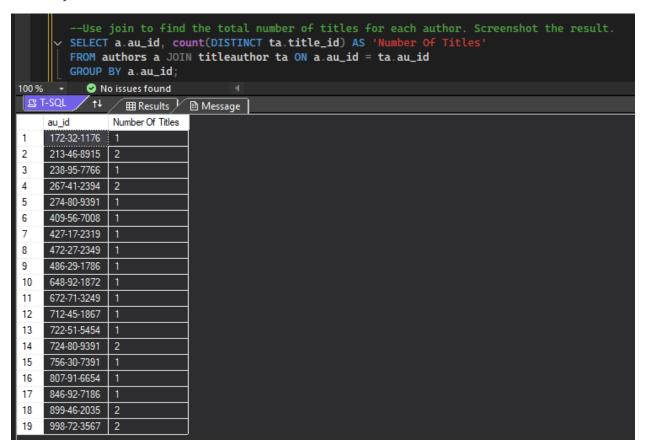
9. Use nested queries to find the average price of titles from the New Moon Books publisher. Screenshot the result.

```
--Use nested queries to find the average price of titles from the New Moon Books publisher. Screenshot the result.

SELECT avg(price) AS 'Average Price'
FROM titles
WHERE pub_id IN(
SELECT pub_id
FROM publishers
WHERE pub_name = 'New Moon Books');

100%  No issues found
Average Price
1 9.784
```

10. Use join to find the total number of titles for each author. Screenshot the result.



11. Create the relation discounts. The data types of the relation attributes are:

```
dis_id int

discounttype varchar(40)

stor_id char(4)

title_id varchar(6)

discount decimal(4,2)
```

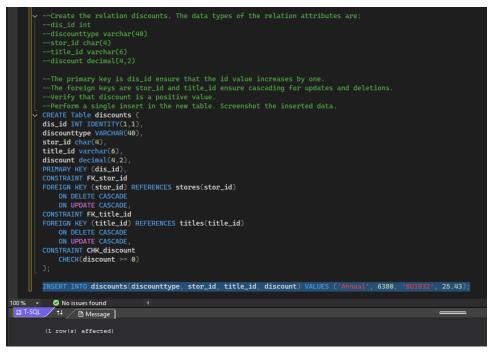
The primary key is dis_id ensure that the id value increases by one. The foreign keys are stor_id and title_id ensure cascading for updates and deletions. Verify that discount is a positive value.

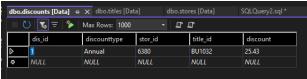
Perform a single insert in the new table. Screenshot the inserted data.

```
--Create the relation discounts. The data types of the relation attributes are:
--dis_id int
--discounttype varchar(40)
--stor_id char(4)
--title_id varchar(6)
--discount decimal(4,2)

--The primary key is dis_id ensure that the id value increases by one.
--The foreign keys are stor_id and title_id ensure cascading for updates and deletions.
--Verify that discount is a positive value.
--Perform a single insert in the new table. Screenshot the inserted data.

| CREATE Table discounts (| dis_id INT IDENTITY(1,1), | discount decimal(4,2), | discount decimal(4,2), | discount decimal(4,2), | discount decimal(4,2), | primary KEY (dis_id). | CONSTRAINT FK_stor_id | CONSTRAINT FK_stor_id | CONSTRAINT FK_ttite_id | FOREIGN KEY (stor_id) REFERENCES stores(stor_id) | ON DELETE CASCADE, ON UPDATE CASCADE, ON UPDATE CASCADE, ON UPDATE CASCADE, CONSTRAINT FK_ttite_id | REFERENCES titles(title_id) | ON DELETE CASCADE, CONSTRAINT GKL_discount | CHECK(discount >= 0) | 10% | N DELETE CASCADE, CONSTRAINT GKL_discount | CHECK(discount >= 0) | 10% | N DELETE CASCADE, CONSTRAINT GKL_discount | CHECK(discount >= 0) | 10% | N DELETE CASCADE, CONSTRAINT GKL_discount | CHECK(discount >= 0) | 10% | N DELETE CASCADE, CONSTRAINT GKL_discount | CHECK(discount >= 0) | 10% | N DELETE CASCADE, CONSTRAINT GKL_discount | CHECK(discount >= 0) | 10% | N DELETE CASCADE, CONSTRAINT GKL_discount | N DELETE CASCADE, CONSTRAINT GKL_discount | CHECK(discount >= 0) | 10% | N DELETE CASCADE, CONSTRAINT GKL_discount | N DELETE CASCADE, C
```





12. Create a view named TitleDetails that lists the title, price, and AuthorFullName for titles greater or equal to \$10.00. Screenshot all entries in the view displayed from least expensive to most expensive.

```
--Create a view named TitleDetails that lists the title, price, and AuthorFullName for titles greater or equal to $10.00.
--Screenshot all entries in the view displayed from least expensive to most expensive.

go

CREATE VIEW [TitleDetails] AS

SELECT title, t.price, a.au_fname + ' ' + a.au_lname AS 'Author Full Name'

FROM authors a JOIN titleauthor ta ON a.au_id = ta au_id

JOIN titles t ON ta.title_id = t.title_id

WHERE t.price >= 10;

go

100%    ON issues found

100%    ON issues found
```

