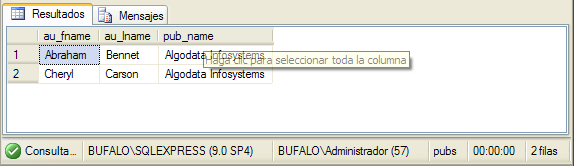
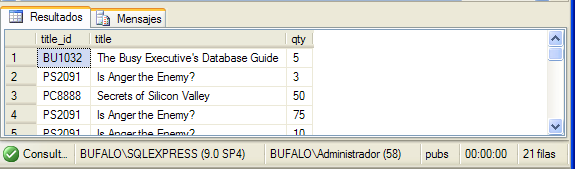
**PUBS**

6.Recuperar los autores ordenados alfabéticamente, que viven en la misma ciudad y estado en la que tiene sede alguna editorial



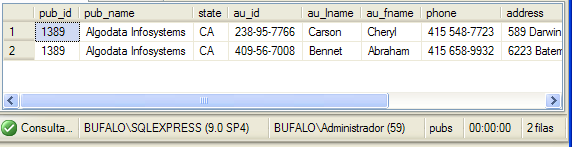
|  |
| --- |
| USE pubs  GO  SELECT au\_fname, au\_lname, pub\_name  FROM authors  INNER JOIN publishers ON authors.city = publishers.city  WHERE (authors.city = publishers.city)  ORDER by au\_fname;  GO |

7.Listar los identificadores de títulos, el título y el número de libros vendidos (qty)



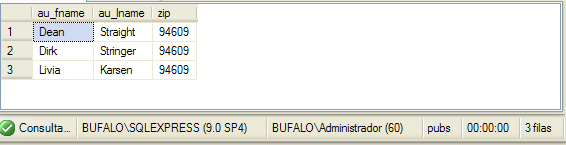
|  |
| --- |
| USE pubs  GO  SELECT titles.title\_id, title, sum(qty)  FROM titles  INNER JOIN sales ON titles.title\_id = sales.title\_id  WHERE titles.title\_id = sales.title\_id  GROUP BY titles.title\_id, title;  GO |

8.Listar el identificativo, el nombre y el estado de la editorial con todos los datos del autor que sean de la misma ciudad. (Se trata de obtener el id de editorial, nombre y estado de las editoriales que tengan sede o sucursal en ciudades en las que resida algún autor. Y obtener además todos los datos de dicho autor)



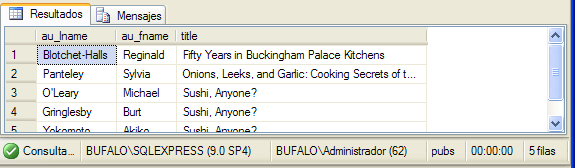
|  |
| --- |
| USE pubs  GO  SELECT pub\_id, pub\_name, publishers.state, authors.\*  FROM publishers  INNER JOIN authors ON publishers.state = authors.state  WHERE publishers.state = authors.state and publishers.city = authors.city;  GO |

9.Listar los autores de Oakland, que viven en el mismo área de código postal(zip) ordenados por código postal. (En definitiva se trata de obtener los autores de Oakland con colegas de igual zip)

****

|  |
| --- |
| USE pubs  GO  SELECT au\_fname, au\_lname, zip  FROM authors  WHERE zip in (  SELECT zip  FROM authors  WHERE city = 'Oakland'  GROUP BY zip  Having COUNT(\*)>1);  GO |

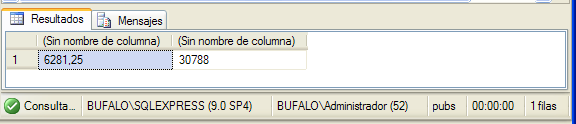
10. Listar los títulos ordenados de manera alfabética y los nombres de sus autores, de todos los libros del tipo 'trad\_cook'.

****

|  |
| --- |
| USE pubs  GO  SELECT au\_lname, au\_fname, titles.title  FROM authors  INNER JOIN titleauthor ON authors.au\_id = titleauthor.au\_id  INNER JOIN titles ON titleauthor.title\_id = titles.title\_id  WHERE type = 'trad\_cook'  ORDER BY titles.title;  GO |

11.Calcular el promedio del adelanto(advance) y la suma de las ventas

(ytd\_sales), sólo de los libros de empresa(business).



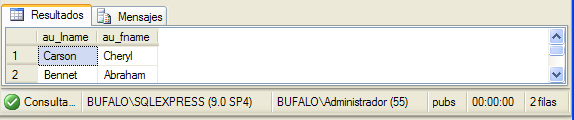
|  |
| --- |
| USE pubs  GO  SELECT AVG(advance), SUM(ytd\_sales)  FROM titles  WHERE type = 'business'  GO |

12. Contar el nº de adelantos (advance) de la tabla titles. En otra consulta contar los nombres de título. ¿Por qué tienen distintos resultados?.

|  |
| --- |
| USE pubs  GO  SELECT count(advance)  FROM titles  GO |
| USE pubs  GO  SELECT count(title)  FROM titles  GO |

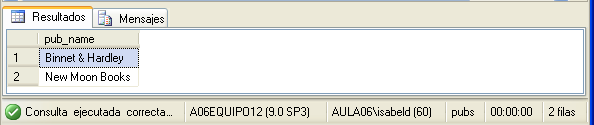
En advance hay dos líneas cuyos valores es NULL por lo tanto no lo cuenta mientras que en haciendo el Count de title, al no haber ningún null, cuenta dos elementos más que el otro.

13.Listar los autores que viven en la misma ciudad en la que hay alguna editorial.



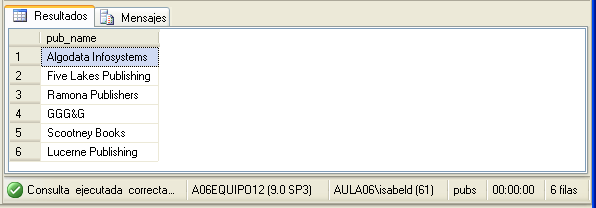
|  |
| --- |
| USE pubs  GO  SELECT au\_lname, au\_fname  FROM authors  INNER JOIN publishers ON authors.city = publishers.city  WHERE authors.city = publishers.city;  GO |

14.Listar los nombres de los editores que han publicado libros de psychology



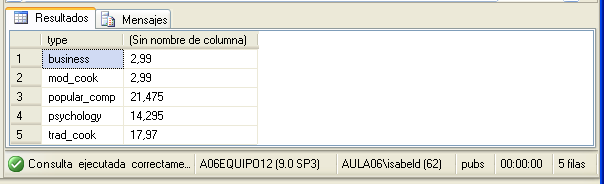
|  |
| --- |
| USE pubs  GO  SELECT pub\_name  FROM publishers  INNER JOIN titles ON publishers.pub\_id = titles.pub\_id  WHERE type = 'psychology'  GROUP BY pub\_name;  GO |

15.Listar el nombre de los editores que no han publicado libros de psychology.



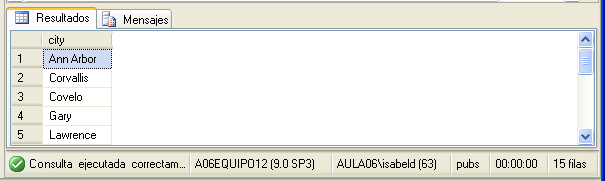
|  |
| --- |
| USE pubs  GO  SELECT pub\_name  FROM publishers  WHERE pub\_name not in  (  SELECT pub\_name  FROM publishers INNER JOIN titles ON publishers.pub\_id = titles.pub\_id  WHERE type = 'psychology'  GROUP BY pub\_name  )  ORDER BY pub\_name;  GO |

16. Para cada tipo de titulo obtener el promedio del precio de todos los títulos con adelantos (advance) mayor de 5000.



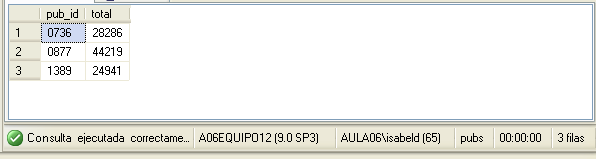
|  |
| --- |
| USE pubs  GO  SELECT type, AVG(price)  FROM titles  WHERE advance > 5000  GROUP BY type;  GO |

17.Listar las ciudades en las que viven autores pero no editores



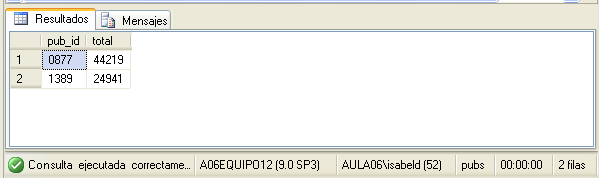
|  |
| --- |
| USE pubs  GO  SELECT city  FROM authors  WHERE city not in (  SELECT publishers.city  FROM publishers  INNER JOIN authors ON publishers.city = authors.city  )  GROUP BY city;  GO |

18.Listar los identificativos de los editores que hayan tenido ventas (ytd\_sales) hasta la fecha superiores a 4000 dólares.



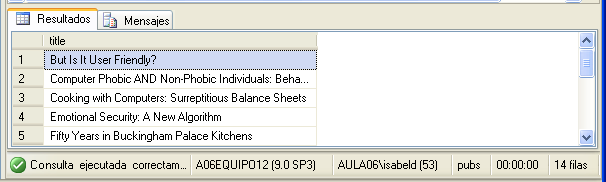
|  |
| --- |
| USE pubs  GO  SELECT publishers.pub\_id, SUM(ytd\_sales) as total  FROM publishers  INNER JOIN titles ON publishers.pub\_id = titles.pub\_id  WHERE ytd\_sales > 4000  GROUP BY publishers.pub\_id;  GO |

19.Listar los identificativos de los editores y la suma de ventas (ytd\_sales) que como mínimo tengan seis libros involucrados en los cálculos para cada editor. *(Se trata de obtener el id de la editorial y el total de ventas de aquellas editoriales que tengan publicados al menos 6 libros)*



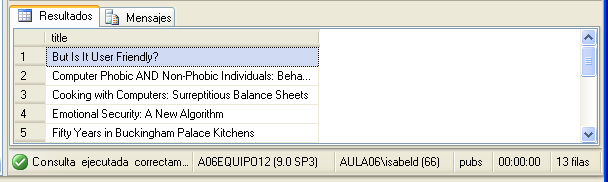
|  |
| --- |
| USE pubs  GO  SELECT publishers.pub\_id, SUM(ytd\_sales) as total  FROM publishers  INNER JOIN titles ON publishers.pub\_id = titles.pub\_id  GROUP BY publishers.pub\_id  HAVING COUNT(title) >= 6;  GO |

20. Busca los nombres de todos los libros con un precio superior al precio mínimo actual.



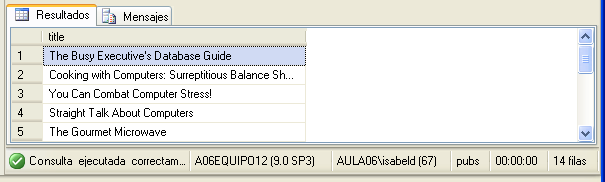
|  |
| --- |
| USE pubs  GO  SELECT title  FROM titles  WHERE price > (  SELECT MIN(price)  FROM titles)  ORDER BY title;  GO |

21.Listar los nombres de todos los libros con un precio superior al precio mínimo del tipo psychology.



|  |
| --- |
| USE pubs  GO  SELECT title  FROM titles  WHERE price > (  SELECT MIN(price)  FROM titles  WHERE type = 'psychology')  ORDER BY title;  GO |

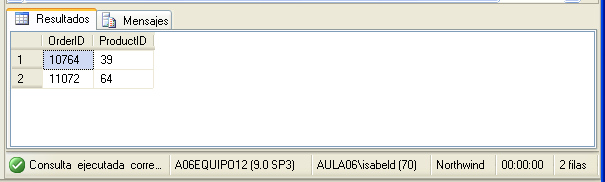
22.Busca los títulos que recibieron un adelanto superior a alguno de los adelantos recibidos por los títulos publicados por 'New Moon Books'



|  |
| --- |
| USE pubs  GO  SELECT title  FROM titles  WHERE advance > (  SELECT MIN(advance)  FROM titles  INNER JOIN publishers ON titles.pub\_id = publishers.pub\_id  WHERE pub\_name = 'New Moon Books')  ORDER BY title;  GO |

**NORTHWIND**

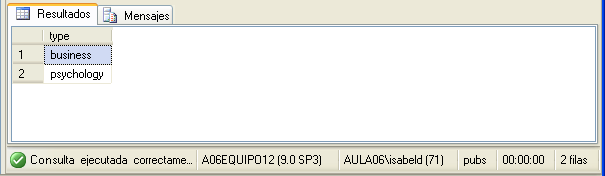
23.Listar los identificadores de los pedidos y de los productos de la tabla Order Details donde la cantidad del producto enviado sea mayor que todas las cantidades enviadas de producto de la categoría 2.



|  |
| --- |
| USE Northwind  GO  SELECT OrderID, ProductID  FROM [Order Details]  WHERE Quantity > (  SELECT MAX(Quantity)  FROM [Order Details]  INNER JOIN Products ON [Order Details].ProductID = Products.ProductID  WHERE CategoryID = 2  );  GO |

**PUBS**

24. Listar los tipos de libros publicados por más de un editor.



|  |
| --- |
| USE pubs  GO  SELECT type  FROM (  SELECT type, publishers.pub\_id  FROM titles  INNER JOIN publishers ON titles.pub\_id = publishers.pub\_id  GROUP BY type, publishers.pub\_id  ) as TablaNueva  GROUP BY type  HAVING COUNT(type) > 1;  GO |