```
-----------------------------------------------------------------------
-- Identifica los elementos no relacionales
-----------------------------------------------------------------------

-- 2.

USE CURSOSQL;

SELECT custid, YEAR(orderdate)
FROM Sales.Orders
ORDER BY 1, 2;

-----------------------------------------------------------------------
-- Convierte la query en relacional
-----------------------------------------------------------------------

-- 1.

SELECT DISTINCT custid, YEAR(orderdate) AS orderyear
FROM Sales.Orders;

-----------------------------------------------------------------------
--  Arregla el problema con la agrupación
-----------------------------------------------------------------------

-- 2.

-- Esta query produce error, arreglala
SELECT custid, orderid
FROM Sales.Orders
GROUP BY custid;

-- 3.

SELECT custid, MAX(orderid) AS maxorderid
FROM Sales.Orders
GROUP BY custid;
```

```
-------------------------------------------------------------------------
- Soluciona los problemas con los alias

-------------------------------------------------------------------------

-- 1. Nombra columnas sin nombre:

SELECT shipperid, SUM(freight)
FROM Sales.Orders
WHERE freight > 20000.00
GROUP BY shipperid;

-- 2. Esta query falla, arreglala:
SELECT shipperid, SUM(freight) AS totalfreight
FROM Sales.Orders
GROUP BY shipperid
HAVING totalfreight > 20000.00;

-- 3. Solución

SELECT shipperid, SUM(freight) AS totalfreight
FROM Sales.Orders
GROUP BY shipperid
HAVING SUM(freight) > 20000.00;


-------------------------------------------------------------------------
-- La clausula FROM
-------------------------------------------------------------------------

USE CURSOSQL;

-- Ejecuta una query que nos de el id, nombre y apellido de todos los
empleados:

SELECT empid, firstname, lastname
FROM HR.Employees;

-- Pon un alias a la tabla empleados de la anterior query

SELECT E.empid, firstname, lastname
FROM HR.Employees AS E;

-------------------------------------------------------------------------
-- La clausula SELECT
-------------------------------------------------------------------------

-- Arregla esta query:

SELECT empid, firstname lastname
FROM HR.Employees;
```

```
-- Renombra la columna empid como employeeid:

SELECT empid AS employeeid, firstname, lastname
FROM HR.Employees;


-- 2. Arregla esta query:

SELECT shipperid, SUM(freight) AS totalfreight
FROM Sales.Orders
GROUP BY shipperid
HAVING totalfreight > 20000.00;

-- 3. Solución

SELECT shipperid, SUM(freight) AS totalfreight
FROM Sales.Orders
GROUP BY shipperid
HAVING SUM(freight) > 20000.00;


-- Elimina los duplicados de esta query:

SELECT country, region, city
FROM HR.Employees;

-- Solución:

SELECT DISTINCT country, region, city
FROM HR.Employees;

-- Ejecuta un ejemplo de SELECT sin clausula FROM:

SELECT 10 AS col1, 'ABC' AS col2;

---------------------------------------------------------------------
-- Tipos de datos y funciones
---------------------------------------------------------------------


-- Esta query falla, arreglala:
SELECT CAST('abc' AS INT);

-- ¿Porque devuelve NULL esta query?

SELECT TRY_CAST('abc' AS INT);

---------------------------------------------------------------------
-- Funciones de fecha y hora
---------------------------------------------------------------------

---------------------------------------------------------------------
-- Hora actual:
---------------------------------------------------------------------

SELECT
  GETDATE()           AS [GETDATE],
  CURRENT_TIMESTAMP   AS [CURRENT_TIMESTAMP],
  GETUTCDATE()        AS [GETUTCDATE],
  SYSDATETIME()       AS [SYSDATETIME],
  SYSUTCDATETIME()    AS [SYSUTCDATETIME],
```

```
  SYSDATETIMEOFFSET() AS [SYSDATETIMEOFFSET];

SELECT
  CAST(SYSDATETIME() AS DATE) AS [current_date],
  CAST(SYSDATETIME() AS TIME) AS [current_time];

-------------------------------------------------------------------
-- Funciones de partes de fecha y hora
-------------------------------------------------------------------

-- ¿puedes devolver unicamente el mes de la fecha 12/02/2012?

SELECT DATEPART(month, '20120212');

-- ¿puedes devolver el día, mes y año de la fecha 12/02/2012?
SELECT
  DAY('20120212') AS theday,
  MONTH('20120212') AS themonth,
  YEAR('20120212') AS theyear;

-- ¿puedes devolver el nombre del mes de la fecha 12/02/2009?
SELECT DATENAME(month, '20090212');

-- ¿puedes devolver la fecha completa partiendo de las partes, día 12,
mes 2 y año 2012?

SELECT
  DATEFROMPARTS(2012, 02, 12),
  DATETIME2FROMPARTS(2012, 02, 12, 13, 30, 5, 1, 7),
  DATETIMEFROMPARTS(2012, 02, 12, 13, 30, 5, 997),
  DATETIMEOFFSETFROMPARTS(2012, 02, 12, 13, 30, 5, 1, -8, 0, 7),
  SMALLDATETIMEFROMPARTS(2012, 02, 12, 13, 30),
  TIMEFROMPARTS(13, 30, 5, 1, 7);

-- ¿Puedes devolver el ultimo día del mes actual?

SELECT EOMONTH(SYSDATETIME());

-------------------------------------------------------------------
-- Funciones de adición de fechas
-------------------------------------------------------------------

-- Añade un año a la fecha 12/02/2012:
SELECT DATEADD(year, 1, '20120212');

-- Devuelve la diferencia en días entre las fechas 12/02/2012 y
12/02/2011

SELECT DATEDIFF(day, '20110212', '20120212');
```

```
-----------------------------------------------------------------------
-- Funciones de cadena
-----------------------------------------------------------------------

-----------------------------------------------------------------------
-- Concatenacion
-----------------------------------------------------------------------

-- Devuelve el id y los campos country, region, city concatenados con
espacios:


SELECT empid, country, region, city,
  country + N',' + region + N',' + city AS location
FROM HR.Employees;

-- En la anterior query, elimina los nulos provocados por el campo
región:


SELECT empid, country, region, city,
  country + COALESCE( N',' + region, N'') + N',' + city AS location
FROM HR.Employees;

-- Utiliza una function de concatenación con la query anterior:


SELECT empid, country, region, city,
  CONCAT(country, N',' + region, N',' + city) AS location
FROM HR.Employees;

-----------------------------------------------------------------------
-- Subcadenas y posición de caracteres y cadenas
-----------------------------------------------------------------------


-- Devuelve las tres primeras letras de la cadena abcde empezando en
la letra b:

SELECT SUBSTRING('abcde', 2, 3); -- 'bcd'

-- Devuelve las tres primeras letras de la cadena abcde:

SELECT LEFT('abcde', 3); -- 'abc'

-- Devuelve las tres ultimas letras de la cadena abcde:

SELECT RIGHT('abcde', 3); -- 'cde'

-- Encuentra la posición de la primera ocurrencia del carácter espacio
en la cadena Itzik Ben-Gan:

SELECT CHARINDEX(' ','Itzik Ben-Gan'); -- 6

-- Encuentra la posición del primer carácter numérico en la cadena
abcd123efgh:

SELECT PATINDEX('%[0-9]%', 'abcd123efgh'); -- 5
```

```
-------------------------------------------------------------------------
-- Longitud de cadena
-------------------------------------------------------------------------

-- Devuelve el número de caracteres de la cadena xyz:

SELECT LEN(N'xyz'); -- 3

-- Devuelve el tamaño en bytes de la cadena xyz:

SELECT DATALENGTH(N'xyz'); -- 6

-------------------------------------------------------------------------
-- Alteración de cadenas
-------------------------------------------------------------------------

--Reemplaza el carcter . por / en la cadena .1.2.3.

SELECT REPLACE('.1.2.3.', '.', '/'); -- '/1/2/3/'

-- Genera una cadena con diez ceros usando solo una vez el carácter 0:

SELECT REPLICATE('0', 10); -- '0000000000'

-------------------------------------------------------------------------
-- Formateo de cadenas
-------------------------------------------------------------------------

-- Convierte todos los caracteres de esta cadena aBcD a mayúsculas

SELECT UPPER('aBcD'); -- 'ABCD'

-- Convierte todos los caracteres de esta cadena aBcD a minusculas

SELECT LOWER('aBcD'); -- 'abcd'

-- Elimina los espacios de ambos extremos de la cadena '   xyz   '

SELECT RTRIM(LTRIM('   xyz   ')); -- 'xyz'


-------------------------------------------------------------------------
-- Expresion CASE
-------------------------------------------------------------------------

-- En la tabla Products, mostrando los campos productid, productname,
unitprice, discontinued, genera un nuevo campo que muestre la palabra
Si, No o Desconocido dependiendo del valor del campo discontinued:


SELECT productid, productname, unitprice, discontinued,
  CASE discontinued
    WHEN 0 THEN 'No'
    WHEN 1 THEN 'Yes'
    ELSE 'Unknown'
  END AS discontinued_desc
FROM Production.Products;

-- En la tabla Products, mostrando los campos productid, productname,
unitprice, genera un nuevo campo que muestre los rangos de precio
unitario <20 es Low,<40 es Medium y >=40 es High:
```

```
SELECT productid, productname, unitprice,
  CASE
    WHEN unitprice < 20.00 THEN 'Low'
    WHEN unitprice < 40.00 THEN 'Medium'
    WHEN unitprice >= 40.00 THEN 'High'
    ELSE 'Unknown'
  END AS pricerange
FROM Production.Products;


-----------------------------------------------------------------------
-- Predicados, Logica trivaluada y busqueda
-----------------------------------------------------------------------

USE CURSOSQL;

-- Muestra el contenido de la tabla HR.Employees:
SELECT empid, firstname, lastname, country, region, city
FROM HR.Employees

-- Muestra los empleados de USA:

SELECT empid, firstname, lastname, country, region, city
FROM HR.Employees
WHERE country = N'USA';

-- Muestra los empleados del estado de Washington(región WA):

SELECT empid, firstname, lastname, country, region, city
FROM HR.Employees
WHERE region = N'WA';

-- Muestra los empleados que no son del estado de Washington(región
WA):

SELECT empid, firstname, lastname, country, region, city
FROM HR.Employees
WHERE region <> N'WA';

-- Muestra los empleados que no son del estado de Washington(región
WA) incluyendo los empleados que no sabemos de donde son:

SELECT empid, firstname, lastname, country, region, city
FROM HR.Employees
WHERE region <> N'WA'
   OR region IS NULL;

-- Muestra las ordines de venta con fecha de entrega 12/02/2007:

SELECT orderid, orderdate, empid
FROM Sales.Orders
WHERE shippeddate = '20070212';

-- ¿Es esta consulta SARGABLE?
SELECT orderid, orderdate, empid
FROM Sales.Orders
WHERE
shippeddate = '20070212' OR COALESCE(shippeddate, '19000101') =
'19000101';

-- ¿Puedes crear una consulta alternativa que sea SARGABLE?
```

```
SELECT orderid, orderdate, empid
FROM Sales.Orders
WHERE shippeddate = '20070212'
   OR shippeddate IS NULL;


-----------------------------------------------------------------------
-- Filtrado de cadenas
-----------------------------------------------------------------------

-- En la tabla HR.Employees encuentra todos los empleados que se
apelliden Davis, el literal tiene que estar definido como caracteres
ANSI:

SELECT empid, firstname, lastname
FROM HR.Employees
WHERE lastname = 'Davis';

-- En la tabla HR.Employees encuentra todos los empleados que se
apelliden Davis, el literal tiene que estar definido como caracteres
Unicode:

SELECT empid, firstname, lastname
FROM HR.Employees
WHERE lastname = N'Davis';

-- En la tabla HR.Employees encuentra todos los empleados que su
apellido empiece por la letra D:

SELECT empid, firstname, lastname
FROM HR.Employees
WHERE lastname LIKE N'D%';

-----------------------------------------------------------------------
-- Filtrar fechas
-----------------------------------------------------------------------

-- Filtra la tabla Sales.Orders por la fecha de la orden con un
literal que sea dependiente del lenguaje configurado:

SELECT orderid, orderdate, empid, custid
FROM Sales.Orders
WHERE orderdate = '02/12/07';

-- Filtra la tabla Sales.Orders por la fecha de la orden con un
literal que sea independiente del lenguaje configurado:

SELECT orderid, orderdate, empid, custid
FROM Sales.Orders
WHERE orderdate = '20070212';

-- ¿Puedes hacer de esta consulta una consulta SARGABLE?
SELECT orderid, orderdate, empid, custid
FROM Sales.Orders
WHERE YEAR(orderdate) = 2007 AND MONTH(orderdate) = 2;

-- SARGABLE
SELECT orderid, orderdate, empid, custid
FROM Sales.Orders
WHERE orderdate >= '20070201' AND orderdate < '20070301';
```

```
-----------------------------------------------------------------------
-- Ordenaciones
-----------------------------------------------------------------------


-- Muestra los campos empid, firstname, lastname, city y el mes de
nacimiento de la tabla de empleados donde el país sea USA, la región
WA y se ordene por la ciudad de forma ascendente:

SELECT empid, firstname, lastname, city, MONTH(birthdate) AS
birthmonth
FROM HR.Employees
WHERE country = N'USA' AND region = N'WA'
ORDER BY city;

-- Muestra los campos empid, firstname, lastname, city y el mes de
nacimiento de la tabla de empleados donde el país sea USA, la región
WA y se ordene por la ciudad de forma descendente:

SELECT empid, firstname, lastname, city, MONTH(birthdate) AS
birthmonth
FROM HR.Employees
WHERE country = N'USA' AND region = N'WA'
ORDER BY city DESC;

--Muestra los campos empid, firstname, lastname, city y el mes de
nacimiento de la tabla de empleados donde el país sea USA, la región
WA y se ordene por la ciudad y el id de empleado de forma ascendente:

SELECT empid, firstname, lastname, city, MONTH(birthdate) AS
birthmonth
FROM HR.Employees
WHERE country = N'USA' AND region = N'WA'
ORDER BY city, empid;

-- Muestra los campos empid, city de la tabla de empleados donde el
país sea USA, la región WA y se ordene por la fecha de nacimiento de
forma ascendente:

SELECT empid, city
FROM HR.Employees
WHERE country = N'USA' AND region = N'WA'
ORDER BY birthdate;


-- ¿Por que falla esta consulta?

SELECT DISTINCT city
FROM HR.Employees
WHERE country = N'USA' AND region = N'WA'
ORDER BY birthdate;
```

```
-- Devuelve las 3 ordenes de venta más recientes mostrando los campos
orderid, orderdate, custid, empid:

SELECT TOP (3) orderid, orderdate, custid, empid
FROM Sales.Orders
ORDER BY orderdate DESC;


-- Devuelve las primeras 25 ordenes después de saltar 50 mostrando
orderid, orderdate, custid, empid y ordenado por fecha de orden
descendente y id de la orden descendente:

SELECT orderid, orderdate, custid, empid
FROM Sales.Orders
ORDER BY orderdate DESC, orderid DESC
OFFSET 50 ROWS FETCH NEXT 25 ROWS ONLY;

-- Devuelve las primeras 25 ordenes mostrando orderid, orderdate,
custid, empid y ordenado por fecha de orden descendente y id de la
orden descendente:

SELECT orderid, orderdate, custid, empid
FROM Sales.Orders
ORDER BY orderdate DESC, orderid DESC
OFFSET 0 ROWS FETCH FIRST 25 ROWS ONLY;

-- Devuelve todas las ordenes después de saltar las primeras 50
mostrando orderid, orderdate, custid, empid y ordenado por fecha de
orden descendente y id de la orden descendente:

SELECT orderid, orderdate, custid, empid
FROM Sales.Orders
ORDER BY orderdate DESC, orderid DESC
OFFSET 50 ROWS;
```

```
-----------------------------------------------------------------------
-- UNION and UNION ALL
-----------------------------------------------------------------------

-- Une las tablas de empleados y clientes, mostrando los campos
country, city filtrando las ocurrencias repetidas:

SELECT country, city
FROM HR.Employees

UNION

SELECT country, city
FROM Sales.Customers;


-- Une las tablas de empleados y clientes, mostrando los campos
country, city respetando las ocurrencias repetidas:

SELECT country, city
FROM HR.Employees

UNION ALL

SELECT country, city
FROM Sales.Customers;

-----------------------------------------------------------------------
-- INTERSECT Y EXCEPT
-----------------------------------------------------------------------


-- Muestra las combinaciones de país y ciudad que estan en la tabla de
empleados y no estan en la de clientes:
SELECT country, city
FROM HR.Employees

EXCEPT

SELECT country,  city
FROM Sales.Customers;

-- Devuelve los empleados que hicieron ordenes de venta para el
cliente 1 pero no para el 2:

SELECT empid
FROM Sales.Orders
WHERE custid = 1

EXCEPT

SELECT empid
FROM Sales.Orders
WHERE custid = 2;

-- Devuelve los empleados que hicieron ordenes de venta para el
cliente 1 yo para el 2:

SELECT empid
FROM Sales.Orders
WHERE custid = 1
```

```
INTERSECT

SELECT empid
FROM Sales.Orders
WHERE custid = 2;

-- Muestra las combinaciones de país y ciudad que estan en la tabla de
empleados y estan en la de clientes:

SELECT country, city
FROM HR.Employees

INTERSECT

SELECT country, city
FROM Sales.Customers;
```

```
-----------------------------------------------------------------
-- JOINS
-----------------------------------------------------------------

-- Muestra los clientes con sus ordenes de venta, debes mostrar los
siguientes campos, id del cliente, compañía del cliente, id de la
orden de venta, fecha de la orden de venta:


USE CURSOSQL;

SELECT C.custid, C.companyname, O.orderid, O.orderdate
FROM Sales.Customers AS C
  INNER JOIN Sales.Orders AS O
    ON C.custid = O.custid;

-- Muestra los clientes con sus ordenes de venta, debes mostrar los
siguientes campos, id del cliente, compañía del cliente, id de la
orden de venta, fecha de la orden de venta. Se deben mostrar todos los
clientes:

SELECT C.custid, C.companyname, O.orderid, O.orderdate
FROM Sales.Customers AS C
  LEFT OUTER JOIN Sales.Orders AS O
    ON C.custid = O.custid;

-- Muestra los clientes que no tienen ordenes de venta, debes mostrar
los siguientes campos, id del cliente, compañía del cliente:

SELECT C.custid, C.companyname
FROM Sales.Customers AS C
  LEFT OUTER JOIN Sales.Orders AS O
    ON C.custid = O.custid
WHERE O.orderid IS NULL;

-- Muestra todos los clientes, debes mostrar los siguientes campos, id
del cliente, compañía del cliente, id de la orden de venta, fecha de
la orden de venta. Solo se tendrán en cuenta las ordenes de venta con
fecha febrero de 2008:

SELECT C.custid, C.companyname, O.orderid, O.orderdate
FROM Sales.Customers AS C
  LEFT OUTER JOIN Sales.Orders AS O
    ON C.custid = O.custid
   AND O.orderdate >= '20080201'
   AND O.orderdate < '20080301';

-----------------------------------------------------------------
-- CTEs
-----------------------------------------------------------------

-- Crea una consulta que devuelva el precio unitario mas alto de un
producto:

SELECT TOP(1) categoryid, unitprice AS mn
FROM Production.Products
ORDER BY unitprice DESC;

-- Une el resultado anterior con la tabla de productos devolviendo los
campos de id de categoría, id de producto, nombre de producto y precio
unitario:
```

```
WITH CatMin AS
(
  SELECT TOP(1) categoryid, unitprice AS mn
        FROM Production.Products
        ORDER BY unitprice DESC;
)
SELECT P.categoryid, P.productid, P.productname, P.unitprice
FROM Production.Products AS P
  INNER JOIN CatMin AS M
    ON P.categoryid = M.categoryid
    AND P.unitprice = M.mn;
```