CHAPTER 5

1. Table is Given:

CREATE TABLE dbo.Employee

(

name NVARCHAR(200) ,

Salary INT

);

Write a SQL query to find the 3th highest employee salary from an Employee table. Explain your answer.

1. Given the following table named A:  
    x

------

2

-2

4

-4

-3

0

2

Write a single query to calculate the sum of all positive values of x and sum of all negative values of x.

1. What are the clauses that you can use to define multiple grouping sets in the same query?
2. What is the difference between PIVOT and UNPIVOT?
3. Table is Given:  
     
   CREATE TABLE #BookSales

(BookType VARCHAR(20), SalesYear INT, BookSales MONEY);

INSERT INTO #BookSales VALUES('Fiction', 2014, 11201);

INSERT INTO #BookSales VALUES('Fiction', 2014, 12939);

INSERT INTO #BookSales VALUES('Fiction', 2013, 10436);

INSERT INTO #BookSales VALUES('Fiction', 2013, 9346);

INSERT INTO #BookSales VALUES('Nonfiction', 2014, 7214);

INSERT INTO #BookSales VALUES('Nonfiction', 2014, 5800);

INSERT INTO #BookSales VALUES('Nonfiction', 2013, 8922);

INSERT INTO #BookSales VALUES('Nonfiction', 2013, 7462);

Write a query that use the operator to turn the **SalesYear** values into columns and to pivot and aggregate the sales totals(By Year):

1. Point out the wrong statement :

a) The ROW\_NUMBER function simply assigns sequential numbering to the records of a result-set or to the records within groups of a result-set  
b) OVER clause is not required in all the ranking functions  
c) SQL Server introduced four different ranking functions  
d) All of the mentioned

1. Which of the clause is not mandatory ?

a) OVER clause  
b) ORDER BY clause  
c) PARTITION BY clause  
d) All of the mentioned

1. Given the following table:

create table #WinTest(

Col1 int,

Col2 int,

Col3 int

)

insert into #WinTest

values (1 , 2 , 2) ,

(2 , 3 , 4),

(4 , 3 , 3),

(6 ,5 ,2) ,

(7 , 6 , 2)

What will be the output result of these queries:

SELECT SUM(Col1) OVER ( PARTITION BY Col3 ) AS sm1 ,

SUM(Col1) OVER ( PARTITION BY Col2 ORDER BY Col1, Col2 ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW ) AS sm2 ,

SUM(Col1) OVER ( PARTITION BY Col2 ORDER BY Col1, Col2 ) AS sm3 ,

ROW\_NUMBER() OVER ( PARTITION BY Col2 ORDER BY Col2 ) AS rnmb ,

RANK() OVER ( PARTITION BY Col3 ORDER BY Col1 ) AS rnk ,

DENSE\_RANK() OVER ( PARTITION BY Col3 ORDER BY Col1 ) AS dn\_rnk ,

LAG(Col3, 1, 222) OVER ( PARTITION BY Col3 ORDER BY Col2 ) AS lg ,

LEAD(Col3, 1, -222) OVER ( PARTITION BY Col3 ORDER BY Col2 ) AS ld ,

FIRST\_VALUE(Col3) OVER ( PARTITION BY Col3 ORDER BY Col2 ) AS fv ,

LAST\_VALUE(Col3) OVER ( PARTITION BY Col3 ORDER BY Col2 ) AS lv

FROM #WinTest;

1. Find a gaped values for Col3