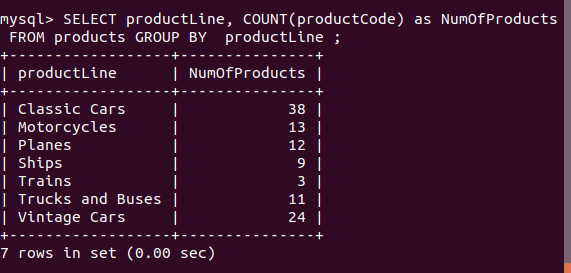
**Assignment**

**use classicmodels database**

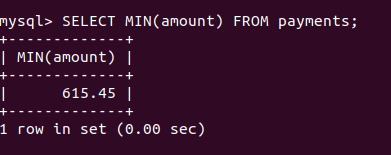
**1. How many products in each product line?**

SELECT productLine, COUNT(productCode) as NumOfProducts

FROM products GROUP BY productLine ;

****

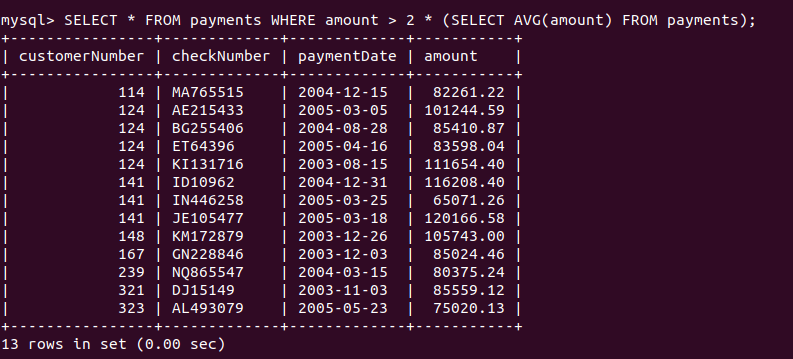
**2. What is the minimum payment received?**



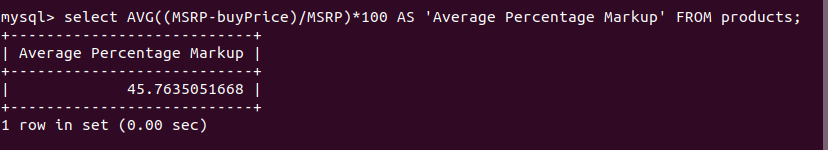
**3. List all payments greater than twice the average payment.**

SELECT \* FROM payments

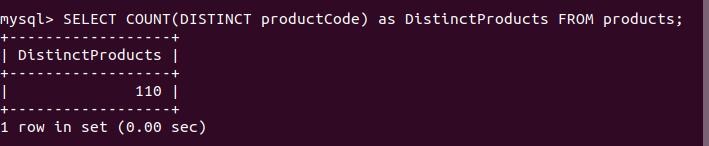
WHERE amount > 2 \* (SELECT AVG(amount) FROM payments);

****

**4. What is the average percentage markup of the MSRP on buyPrice?**

****

**5. How many distinct products does ClassicModels sell?**

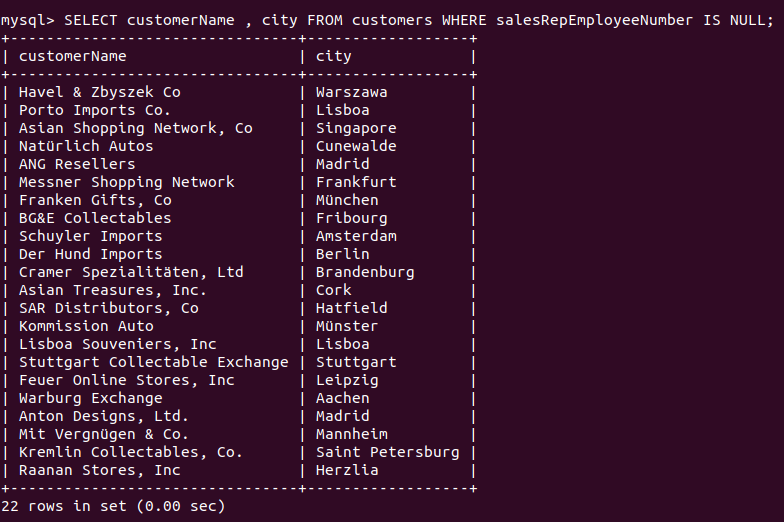
****

**6. Report the name and city of customers who don't have sales representatives?**

SELECT customerName , city

FROM customers

WHERE salesRepEmployeeNumber IS NULL;

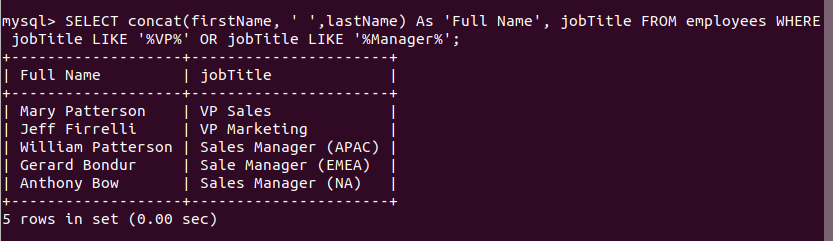
****

**7. What are the names of executives with VP or Manager in their title? Use the CONCAT function to combine the employee's first name and last name into a single field for reporting.**

SELECT concat(firstName, ' ',lastName) As 'Full Name', jobTitle

FROM employees

WHERE jobTitle LIKE '%VP%' OR jobTitle LIKE '%Manager%';



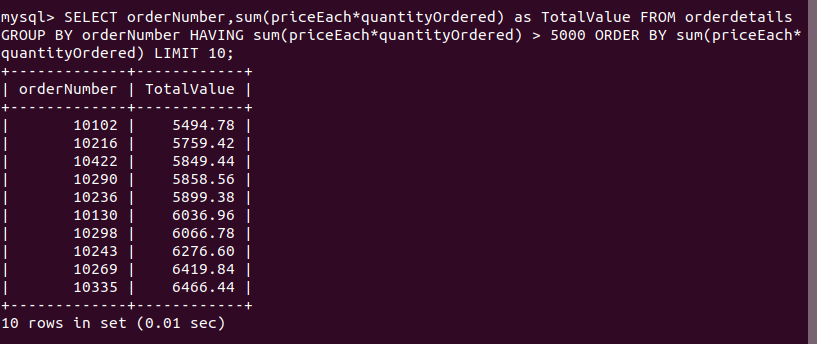
**8. Which orders have a value greater than $5,000?**

SELECT orderNumber,sum(priceEach\*quantityOrdered) as TotalValue

FROM orderdetails GROUP BY orderNumber

HAVING sum(priceEach\*quantityOrdered) > 5000

ORDER BY sum(priceEach\*quantityOrdered);

****

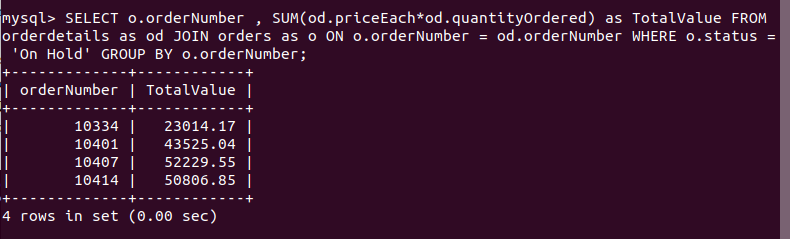
**9. List the value of 'On Hold' orders.**

SELECT o.orderNumber, SUM(od.priceEach\*od.quantityOrdered) as TotalValue

FROM orderdetails as od JOIN orders as o ON o.orderNumber = od.orderNumber

WHERE o.status = ‘On Hold’

GROUP BY o.orderNumber;



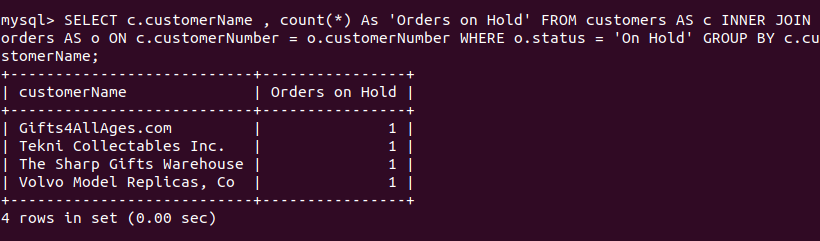
**10. Report the number of orders 'On Hold' for each customer.**

SELECT c.customerName , count(\*) As 'Orders on Hold'

FROM customers AS c INNER JOIN orders AS o

ON c.customerNumber = o.customerNumber

WHERE o.status = 'On Hold' GROUP BY c.customerName;

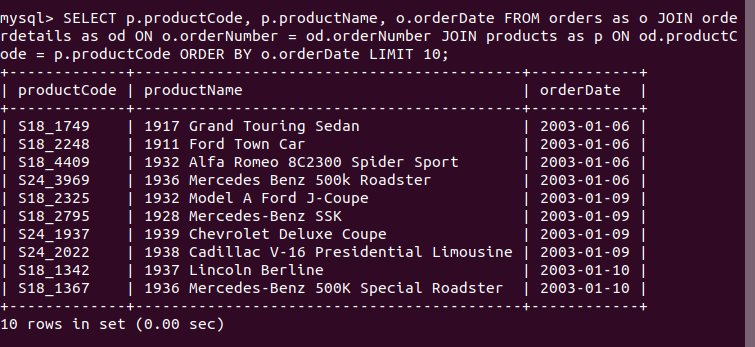
****

**11. List products sold by order date.**

SELECT p.productCode, p.productName, o.orderDate

FROM orders as o JOIN orderdetails as od ON o.orderNumber = od.orderNumber

JOIN products as p ON od.productCode = p.productCode ORDER BY o.orderDate;

****

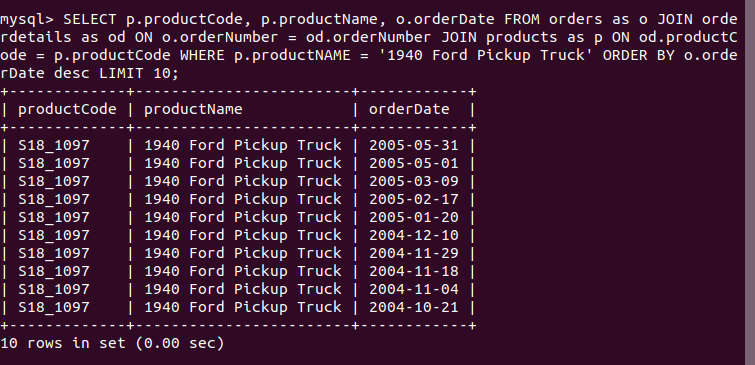
**12. List the order dates in descending order for orders for the 1940 Ford Pickup Truck.**

SELECT p.productCode, p.productName, o.orderDate

FROM orders as o JOIN orderdetails as od ON o.orderNumber = od.orderNumber

JOIN products as p ON od.productCode = p.productCode

WHERE p.productNAME = '1940 Ford Pickup Truck' ORDER BY o.orderDate desc;



**13. List the names of customers and their corresponding order number where a particular order from that customer has a value greater than $25,000?**

SELECT c.customerName, o.orderNumber

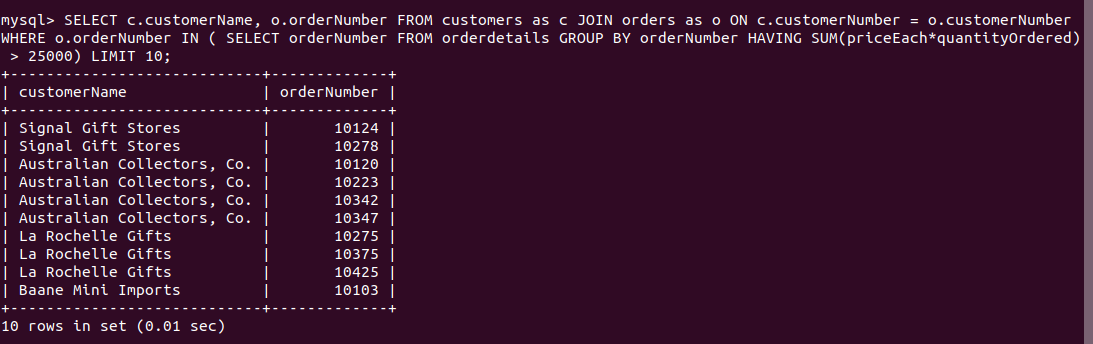
FROM customers as c JOIN orders as o ON c.customerNumber = o.customerNumber

WHERE o.orderNumber IN

( SELECT orderNumber

FROM orderdetails

GROUP BY orderNumber HAVING SUM(priceEach\*quantityOrdered) > 25000 );



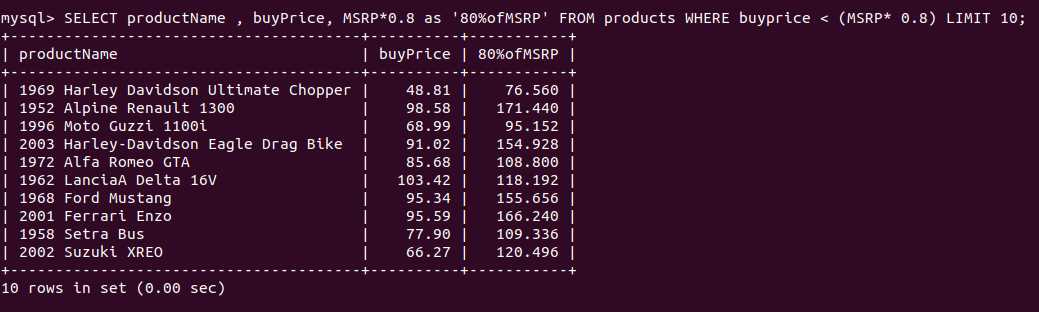
**14. Are there any products that appear on all orders?**

‘No such Products’

**15. List the names of products sold at less than 80% of the MSRP**

SELECT productName , buyPrice, MSRP\*0.8 as ‘80%ofMSRP’

FROM products WHERE buyprice < (MSRP\* 0.8);



**16.Compute the commission for each sales representative, assuming the commission is 5% of the value of an order. Sort by employee last name and first name.**

SELECT e.employeeNumber, e.firstName, e.lastName, CAST(0.05 \*SUM(od.priceEach\*od.quantityOrdered) AS DECIMAL(8,2)) AS Commission

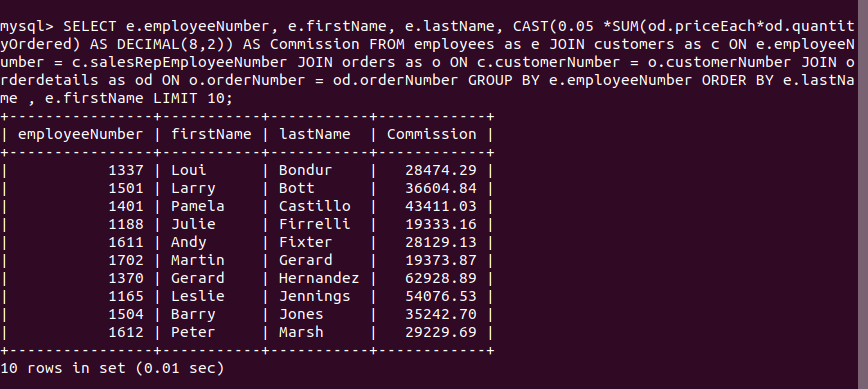
FROM employees as e

JOIN customers as c ON e.employeeNumber = c.salesRepEmployeeNumber

JOIN orders as o ON c.customerNumber = o.customerNumber

JOIN orderdetails as od ON o.orderNumber = od.orderNumber

GROUP BY e.employeeNumber ORDER BY e.lastName , e.firstName;

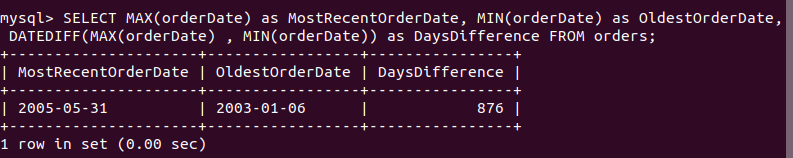


**17. What is the difference in days between the most recent and oldest order date in the Orders file?**

SELECT MAX(orderDate) as MostRecentOrderDate, MIN(orderDate) as OldestOrderDate,

DATEDIFF(MAX(orderDate) , MIN(orderDate)) as DaysDifference

FROM orders;



**18. Compute the ratio of payments for each customer for 2003 versus 2004.**

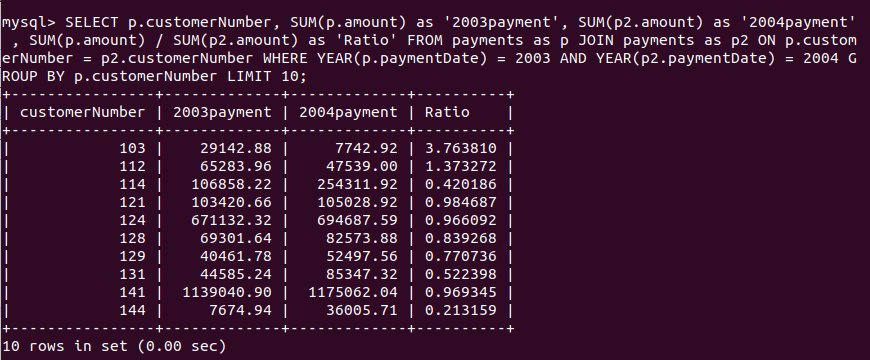
SELECT p.customerNumber, SUM(p.amount) as '2003payment', SUM(p2.amount) as '2004payment' , SUM(p.amount) / SUM(p2.amount) as 'Ratio'

FROM payments as p JOIN payments as p2

ON p.customerNumber = p2.customerNumber

WHERE YEAR(p.paymentDate) = 2003 AND YEAR(p2.paymentDate) = 2004

GROUP BY p.customerNumber LIMIT 10;



**19. Find the products sold in 2003 but not 2004.**

SELECT p.productCode, p.productName, o.orderDate

FROM orders as o JOIN orderdetails as od ON o.orderNumber = od.orderNumber

JOIN products as p ON od.productCode = p.productCode

WHERE YEAR(o.orderDate) = 2003 AND p.productCode NOT IN

(SELECT p.productCode

FROM orders as o JOIN orderdetails as od ON o.orderNumber = od.orderNumber

JOIN products as p ON od.productCode = p.productCode

WHERE YEAR(o.orderDate) = 2004) ;

**20. Find the customers without payments in 2003.**

SELECT DISTINCT p.customerNumber, c.customerName

FROM payments as p JOIN customers as c ON p.customerNumber = c.customerNumber WHERE YEAR(p.paymentDate) <> 2003;

