**HOMEWORK #3 SARAH MENTEL 2/20**

1. (5) Write a SELECT statement that returns these four columns where the balance due is less than 200 and greater than 0.

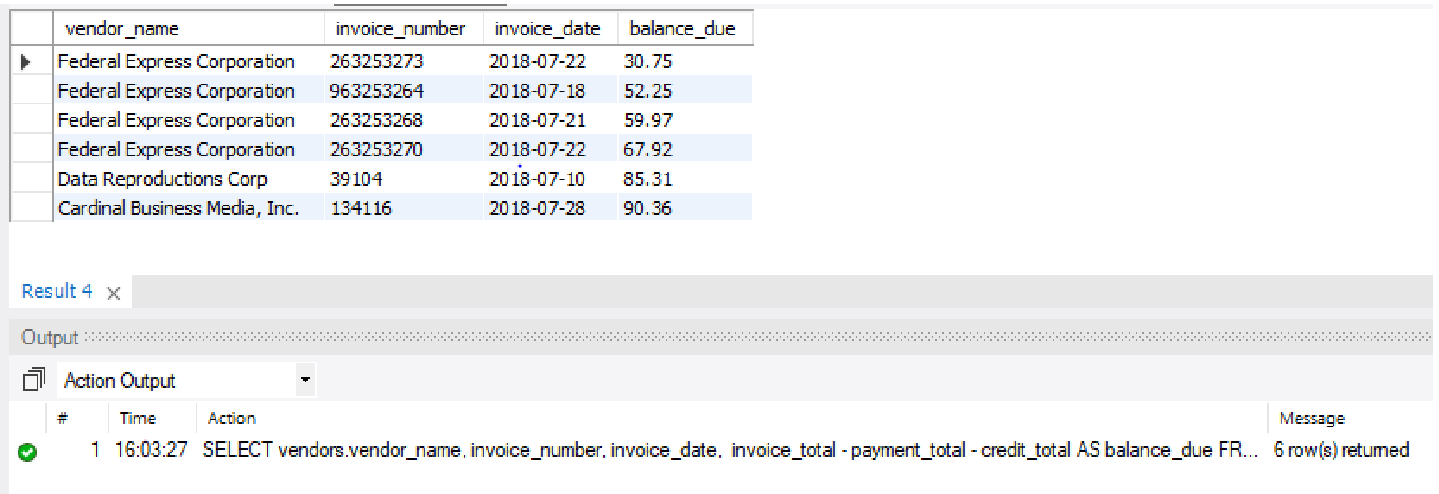
| | |
|---|---|
| vendor_name | The vendor_name column from the Vendors table |
| invoice_number | The invoice_number column from the Invoices table |
| invoice_date | The invoice_date column from the Invoices table |
| balance_due | The invoice_total column minus the payment_total and credit_total columns from the Invoices table |

Sort the result by balance_due.

**ANSWER 1:**

```
SELECT vendors.vendor_name, invoice_number, invoice_date,  invoice_total - payment_total -
                              credit_total AS balance_due

FROM invoices

JOIN vendors ON vendors.vendor_id = invoices.vendor_id

WHERE (invoice_total - payment_total - credit_total) > 0

AND (invoice_total - payment_total - credit_total) < 200

ORDER BY (invoice_total - payment_total - credit_total)
```

**SCREENSHOT 1:**



| vendor_name | invoice_number | invoice_date | balance_due |
|---|---|---|---|
| Federal Express Corporation | 263253273 | 2018-07-22 | 30.75 |
| Federal Express Corporation | 963253264 | 2018-07-18 | 52.25 |
| Federal Express Corporation | 263253268 | 2018-07-21 | 59.97 |
| Federal Express Corporation | 263253270 | 2018-07-22 | 67.92 |
| Data Reproductions Corp | 39104 | 2018-07-10 | 85.31 |
| Cardinal Business Media, Inc. | 134116 | 2018-07-28 | 90.36 |

| # | Time | Action | Message |
|---|---|---|---|
| 1 | 16:03:27 | SELECT vendors.vendor_name, invoice_number, invoice_date,  invoice_total - payment_total - credit_total AS balance_due FR... | 6 row(s) returned |

2. (5) Write a SELECT statement that returns three columns:

vendor_id The vendor_id column from the Vendors table

vendor_name The vendor_name column from the Vendors table

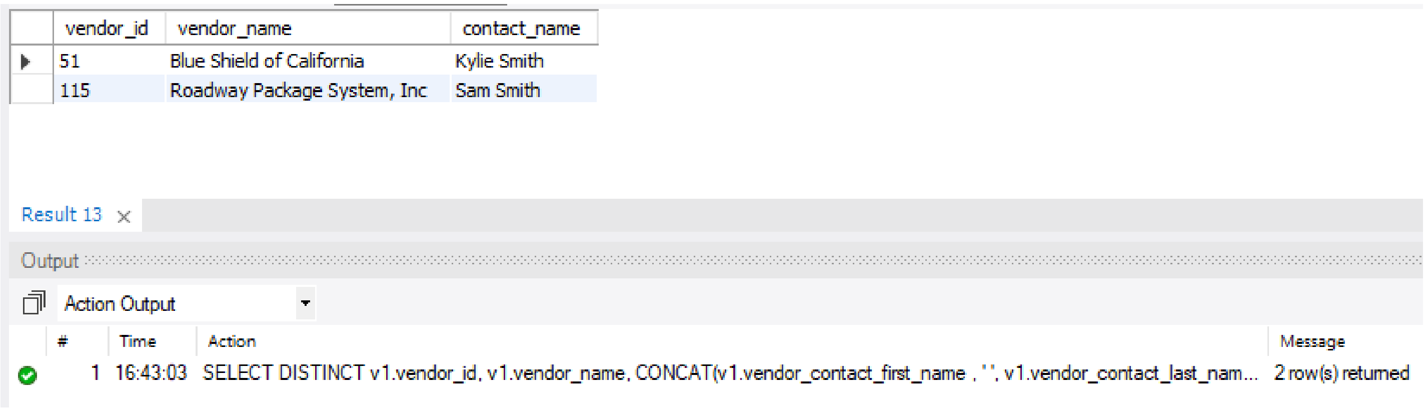
contact_name A concatenation of the vendor_contact_first_name and vendor_contact_last_name columns with a space between

Return one row for each vendor whose contact has the same last name as another vendor's contact. Sort the result set by vendor_contact_last_name.

**ANSWER 2:**

```
 SELECT DISTINCT v1.vendor_id, v1.vendor_name, CONCAT(v1.vendor_contact_first_name , ' ',
v1.vendor_contact_last_name) AS contact_name

FROM vendors v1 JOIN vendors v2 ON (v1.vendor_id <>  v2.vendor_id)

WHERE (v1.vendor_contact_last_name = v2.vendor_contact_last_name)

ORDER BY contact_name
```

**SCREENSHOT 2:**

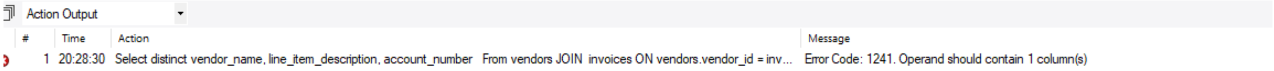


| vendor_id | vendor_name | contact_name |
|---|---|---|
| 51 | Blue Shield of California | Kylie Smith |
| 115 | Roadway Package System, Inc | Sam Smith |

```
#   Time      Action                                                                                              Message
1   16:43:03  SELECT DISTINCT v1.vendor_id, v1.vendor_name, CONCAT(v1.vendor_contact_first_name , ' ', v1.vendor_contact_last_nam...   2 row(s) returned
```

3. (5) Show the vendor name, line_item_description, and account number for each line item. Show vendors which use just one account number.
   **ANSWER you gave in class**

```
Select distinct vendor_name, line_item_description, account_number

From vendors JOIN  invoices ON vendors.vendor_id = invoices.vendor_id

JOIN invoice_line_items ON invoices.invoice_id = invoice_line_items.invoice_id

WHERE vendor_name IN (

Select vendor_name, line_item_description, count(account_number)

From vendors JOIN  invoices ON vendors.vendor_id = invoices.vendor_id

JOIN invoice_line_items ON invoices.invoice_id = invoice_line_items.invoice_id

GROUP BY vendor_name

HAVING count(distinct account_number) =1);
```



```
Action Output
  #  Time      Action                                                                                               Message
  1  20:28:30  Select distinct vendor_name, line_item_description, account_number  From vendors JOIN  invoices ON vendors.vendor_id = inv...  Error Code: 1241. Operand should contain 1 column(s)
```

4. (5) Show two columns from the Vendors table: vendor_name and vendor_phone. If the vendor has a phone number, the vendor_phone value should be its phone number. Otherwise, the vendor_phone value should be “No Phone.” Just show the vendors whose initial starts with ‘A’ through ‘K’ and sort the result set by vendor_name.

**ANSWER 4:**

**UPDATE vendors**

**SET vendor_phone = 'No Phone'**

**WHERE vendor_phone IS NULL;**

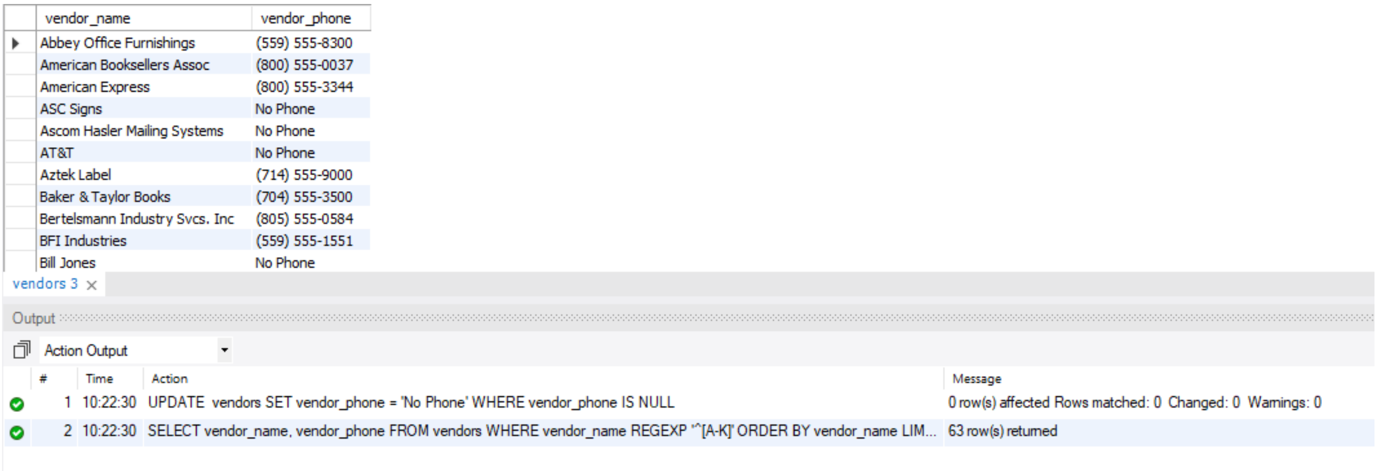
**SELECT vendor_name, vendor_phone**

**FROM vendors**

**WHERE vendor_name REGEXP '^[A-K]'**

**ORDER BY vendor_name;**

**SCREENSHOT 4:**



| vendor_name | vendor_phone |
|---|---|
| Abbey Office Furnishings | (559) 555-8300 |
| American Booksellers Assoc | (800) 555-0037 |
| American Express | (800) 555-3344 |
| ASC Signs | No Phone |
| Ascom Hasler Mailing Systems | No Phone |
| AT&T | No Phone |
| Aztek Label | (714) 555-9000 |
| Baker & Taylor Books | (704) 555-3500 |
| Bertelsmann Industry Svcs. Inc | (805) 555-0584 |
| BFI Industries | (559) 555-1551 |
| Bill Jones | No Phone |

```
#  Time      Action                                                                                     Message
1  10:22:30  UPDATE  vendors SET vendor_phone = 'No Phone' WHERE vendor_phone IS NULL                   0 row(s) affected Rows matched: 0  Changed: 0  Warnings: 0
2  10:22:30  SELECT vendor_name, vendor_phone FROM vendors WHERE vendor_name REGEXP '^[A-K]' ORDER BY vendor_name LIM...   63 row(s) returned
```

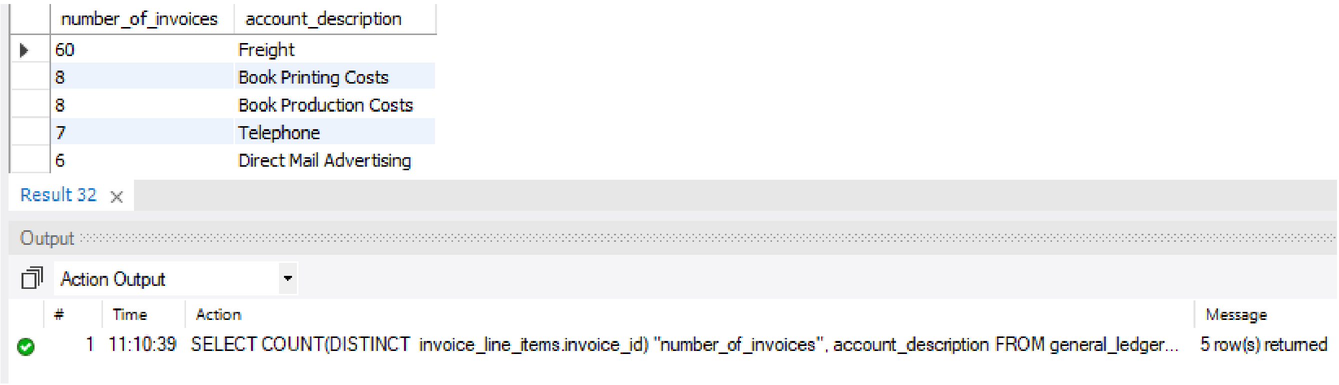
5. (40) Consider the following ER diagram.

   a. (10) Show the account_description and the number of invoices for each account_description. Show the top five results in terms of the number of invoices.

**ANSWER 5a:**

```
SELECT COUNT(DISTINCT  invoice_line_items.invoice_id) "number_of_invoices", account_description

FROM general_ledger_accounts

JOIN invoice_line_items ON general_ledger_accounts.account_number =
invoice_line_items.account_number

GROUP BY account_description

ORDER BY COUNT(DISTINCT  invoice_line_items.invoice_id) DESC LIMIT 5;
```

**SCREENSHOT 5a:**



| number_of_invoices | account_description |
|---|---|
| 60 | Freight |
| 8 | Book Printing Costs |
| 8 | Book Production Costs |
| 7 | Telephone |
| 6 | Direct Mail Advertising |

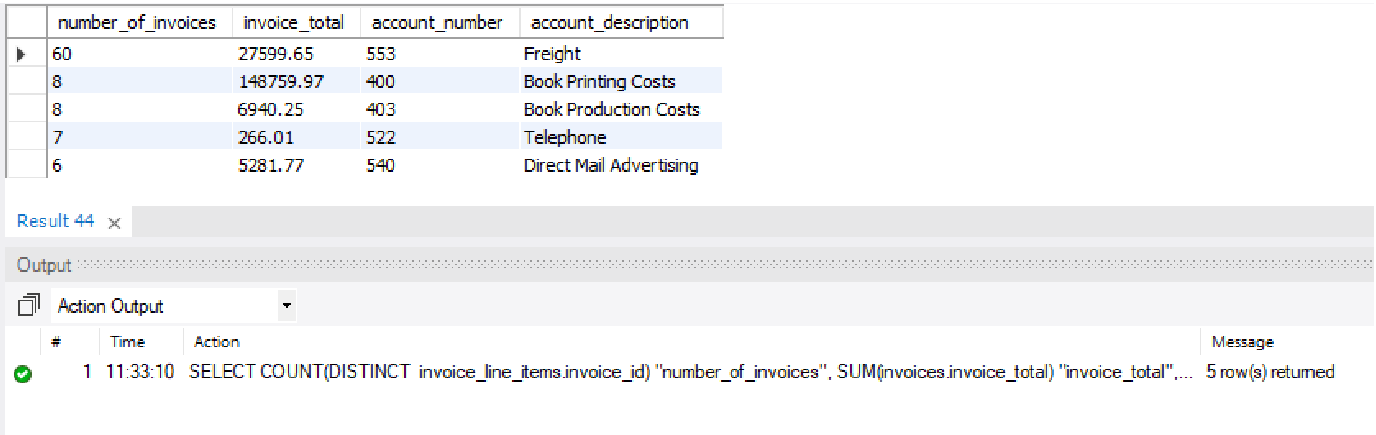
```
#   Time      Action                                                                                                      Message
1   11:10:39  SELECT COUNT(DISTINCT  invoice_line_items.invoice_id) "number_of_invoices", account_description FROM general_ledger...   5 row(s) returned
```

b. (10) What is the total invoice for the account from the above question? Show the query that finds the answer. The result should show the account_number, account_description, the number of invoices, and the total invoice for each account_number.

**ANSWER 5b:**

```
SELECT COUNT(DISTINCT  invoice_line_items.invoice_id) "number_of_invoices",
SUM(invoices.invoice_total) "invoice_total", general_ledger_accounts.account_number,
account_description

FROM general_ledger_accounts

JOIN invoice_line_items ON general_ledger_accounts.account_number =
invoice_line_items.account_number

JOIN invoices ON invoice_line_items.invoice_id = invoices.invoice_id

GROUP BY account_number

ORDER BY COUNT(DISTINCT  invoice_line_items.invoice_id) DESC LIMIT 5;
```

**SCREENSHOT 5b:**



| number_of_invoices | invoice_total | account_number | account_description |
|---|---|---|---|
| 60 | 27599.65 | 553 | Freight |
| 8 | 148759.97 | 400 | Book Printing Costs |
| 8 | 6940.25 | 403 | Book Production Costs |
| 7 | 266.01 | 522 | Telephone |
| 6 | 5281.77 | 540 | Direct Mail Advertising |

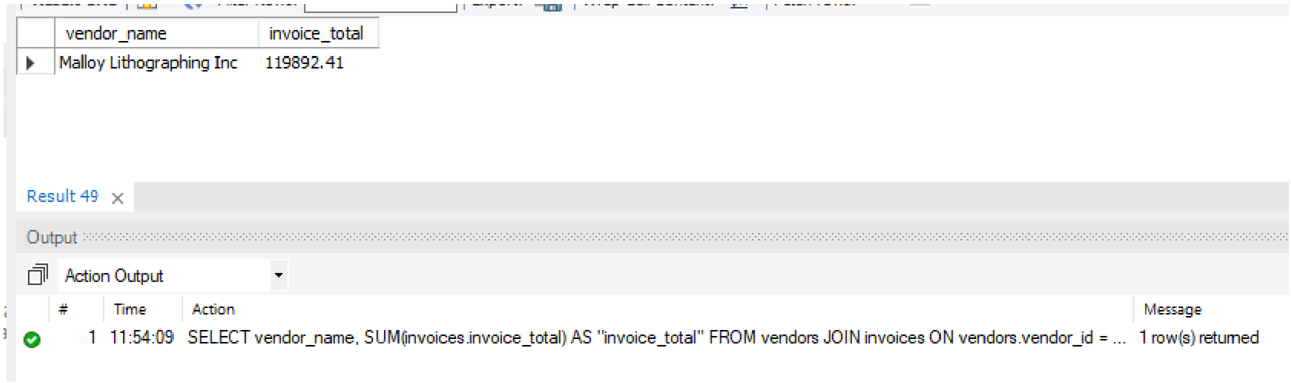
```
#   Time      Action                                                                                                Message
1   11:33:10  SELECT COUNT(DISTINCT  invoice_line_items.invoice_id) "number_of_invoices", SUM(invoices.invoice_total) "invoice_total",...  5 row(s) returned
```

c.  (10) Find the vendor which has the biggest amount of total invoices. Show the query that shows the vendor name and the total invoice amount.

**ANSWER  5c:**

```
SELECT vendor_name, SUM(invoices.invoice_total) AS "invoice_total"

FROM vendors

JOIN invoices ON vendors.vendor_id = invoices.vendor_id

GROUP BY vendor_name

ORDER BY SUM(invoices.invoice_total) DESC LIMIT 1
```

**SCREENSHOT  5c:**



| vendor_name | invoice_total |
|---|---|
| Malloy Lithographing Inc | 119892.41 |

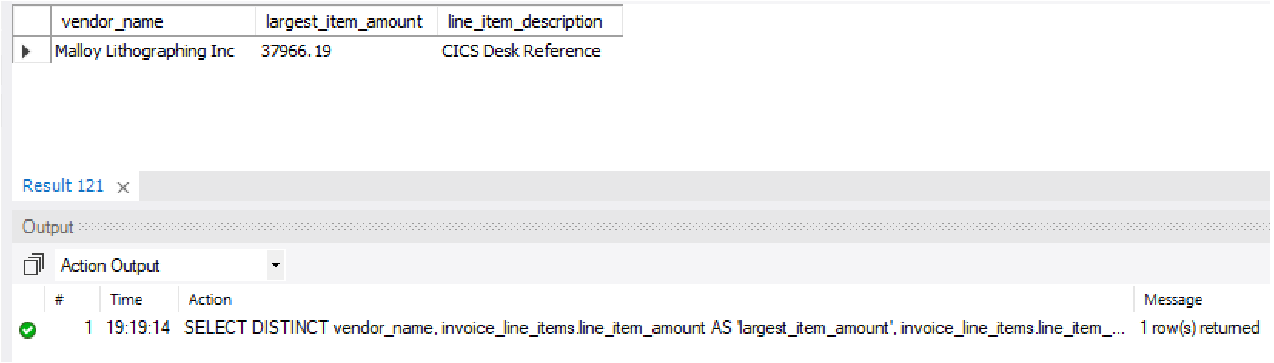
```
#   Time      Action                                                                                         Message
1   11:54:09  SELECT vendor_name, SUM(invoices.invoice_total) AS "invoice_total" FROM vendors JOIN invoices ON vendors.vendor_id = ...  1 row(s) returned
```

d. (10) Find the vendor which has the single most expensive order for an item. Show the query that shows the vendor name, the item description, and the amount.

**ANSWER 5d:**

```
SELECT DISTINCT vendor_name, invoice_line_items.line_item_amount AS 'largest_item_amount',
                                invoice_line_items.line_item_description

FROM vendors

JOIN invoices ON vendors.vendor_id = invoices.vendor_id

JOIN invoice_line_items ON invoices.invoice_id  = invoice_line_items.invoice_id

WHERE invoice_line_items.line_item_amount <= (

SELECT MAX(line_item_amount)

FROM invoice_line_items

)

ORDER BY invoice_line_items.line_item_amount DESC LIMIT 1;
```

**SCREENSHOT 5c:**



| vendor_name | largest_item_amount | line_item_description |
|---|---|---|
| Malloy Lithographing Inc | 37966.19 | CICS Desk Reference |

```
#   Time      Action                                                                                                    Message
1   19:19:14  SELECT DISTINCT vendor_name, invoice_line_items.line_item_amount AS 'largest_item_amount', invoice_line_items.line_item_...   1 row(s) returned
```

6. 5) Create a table called “new_terms” from the terms table. Then, write an INSERT statement that adds this row to the new_terms table:

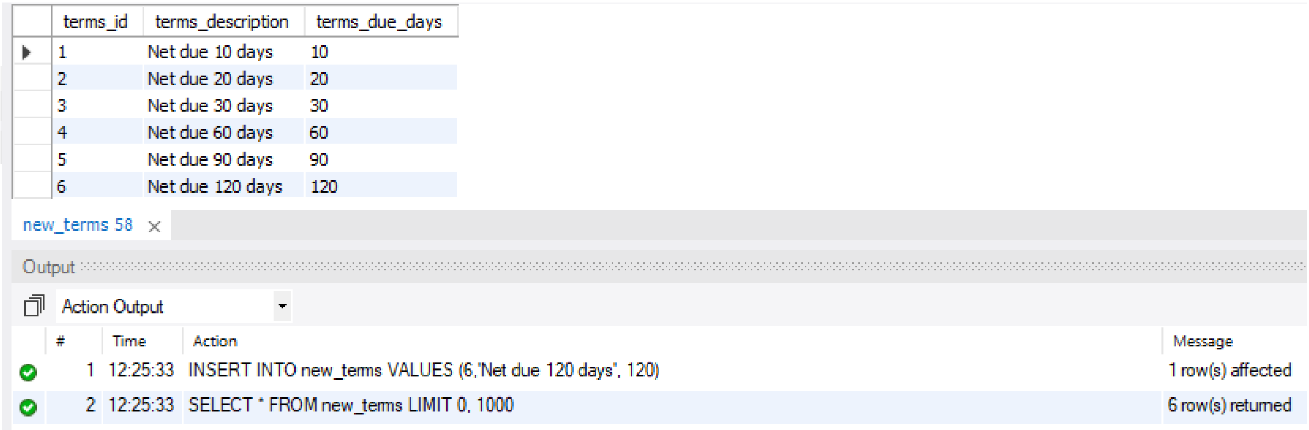
|  |  |
|---|---|
| terms_id: | 6 |
| terms_description: | Net due 120 days |
| terms_due_days: | 120 |

Then, show the result of the following query: SELECT * FROM new_terms;

**ANSWER 6:**

```
CREATE TABLE new_terms AS SELECT *

FROM terms;


INSERT INTO new_terms VALUES

(6,'Net due 120 days', 120);


SELECT *

FROM new_terms;
```

**SCREENSHOT 6:**



| terms_id | terms_description | terms_due_days |
|---|---|---|
| 1 | Net due 10 days | 10 |
| 2 | Net due 20 days | 20 |
| 3 | Net due 30 days | 30 |
| 4 | Net due 60 days | 60 |
| 5 | Net due 90 days | 90 |
| 6 | Net due 120 days | 120 |

| # | Time | Action | Message |
|---|---|---|---|
| 1 | 12:25:33 | INSERT INTO new_terms VALUES (6,'Net due 120 days', 120) | 1 row(s) affected |
| 2 | 12:25:33 | SELECT * FROM new_terms LIMIT 0, 1000 | 6 row(s) returned |

7.(5) Then, write an UPDATE statement that modifies the row you just added to the Terms table. This statement should change the terms_description column to “Net due 100 days”, and it should change the terms_due_days column to 100.

Then, show the result of the following query: SELECT * FROM new_terms;

**ANSWER 7:**

**UPDATE new_terms**

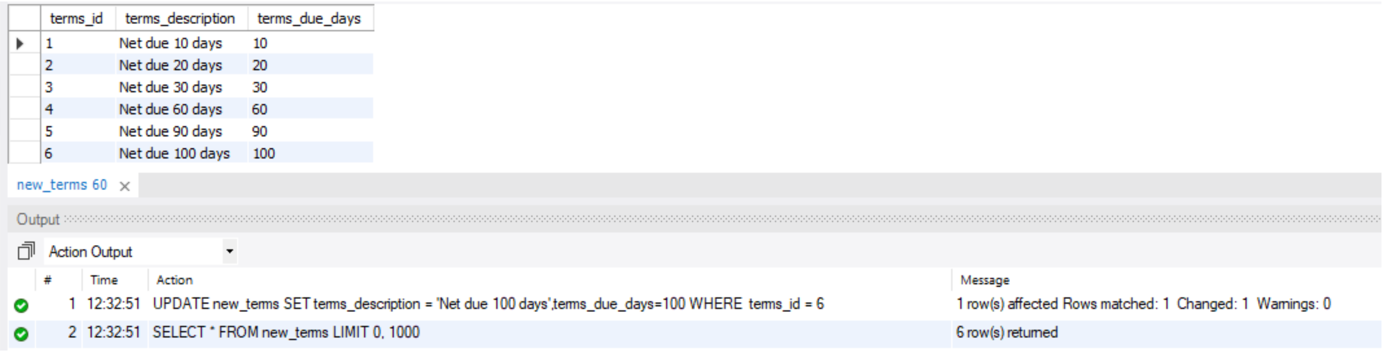
**SET terms_description = 'Net due 100 days',terms_due_days=100**

**WHERE terms_id = 6;**

**SELECT ***

**FROM new_terms;**

**SCREENSHOT 7:**



| terms_id | terms_description | terms_due_days |
|---|---|---|
| 1 | Net due 10 days | 10 |
| 2 | Net due 20 days | 20 |
| 3 | Net due 30 days | 30 |
| 4 | Net due 60 days | 60 |
| 5 | Net due 90 days | 90 |
| 6 | Net due 100 days | 100 |

```
#  Time      Action                                                                                   Message
1  12:32:51  UPDATE new_terms SET terms_description = 'Net due 100 days',terms_due_days=100 WHERE  terms_id = 6   1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0
2  12:32:51  SELECT * FROM new_terms LIMIT 0, 1000                                                    6 row(s) returned
```

8. (5) Then, write a DELETE statement that deletes the row you added to the new Terms table (that is with the terms_id 6).
Then, show the result of the following query: SELECT * FROM new_terms;

**ANSWER 8:**

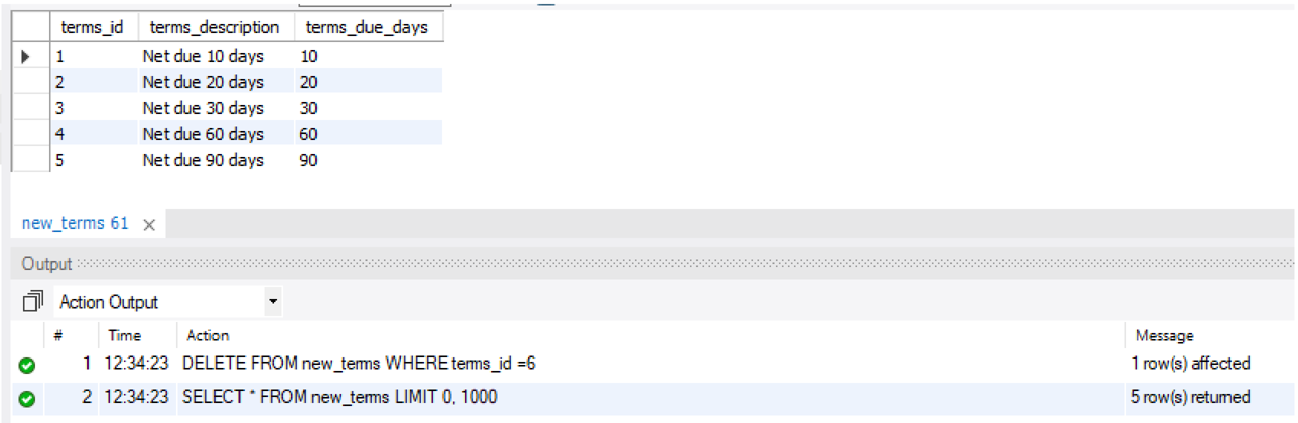
**DELETE FROM new_terms**

**WHERE terms_id =6;**

**SELECT ***

**FROM new_terms;**

**SCREENSHOT 8:**



| terms_id | terms_description | terms_due_days |
|---|---|---|
| 1 | Net due 10 days | 10 |
| 2 | Net due 20 days | 20 |
| 3 | Net due 30 days | 30 |
| 4 | Net due 60 days | 60 |
| 5 | Net due 90 days | 90 |

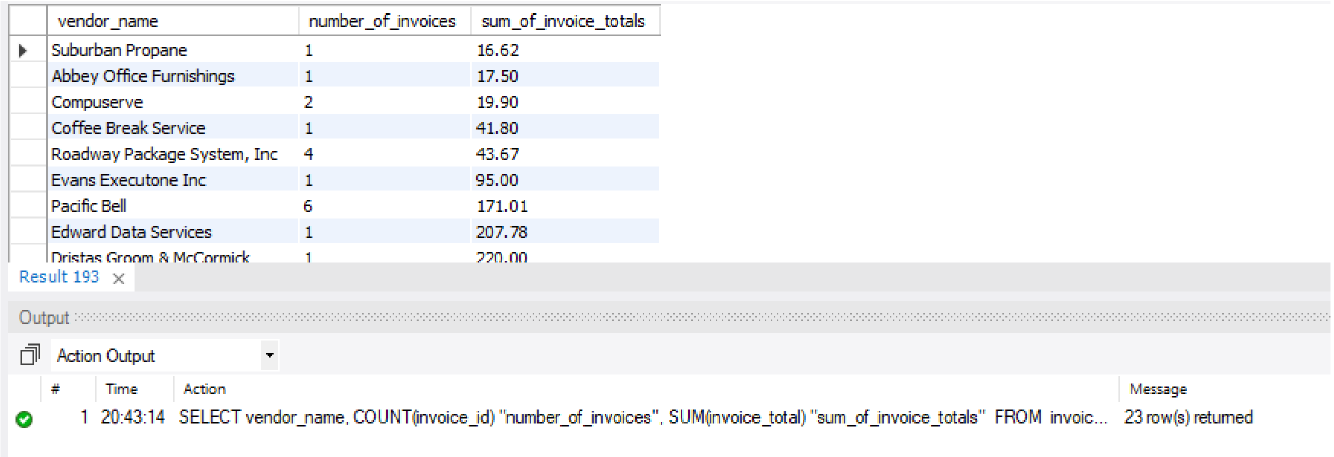
| # | Time | Action | Message |
|---|---|---|---|
| 1 | 12:34:23 | DELETE FROM new_terms WHERE terms_id =6 | 1 row(s) affected |
| 2 | 12:34:23 | SELECT * FROM new_terms LIMIT 0, 1000 | 5 row(s) returned |

9. (5) Write a query that displays the vendor name, the number of invoices, and the sum of invoice totals for each vendor, sorted by the sum of invoice totals. Show only those vendors whose invoice totals are less than the average invoice totals.

**ANSWER 9:**

```
SELECT vendor_name, COUNT(invoice_id) "number_of_invoices", SUM(invoice_total)
                         "sum_of_invoice_totals"

FROM  invoices

JOIN vendors ON invoices.vendor_id = vendors.vendor_id

GROUP BY vendor_name

Having SUM(invoice_total) < (SELECT avg(invoice_total)

FROM invoices)

ORDER BY SUM(invoice_total)
```

**SCREENSHOT 9:**



| vendor_name | number_of_invoices | sum_of_invoice_totals |
|---|---|---|
| Suburban Propane | 1 | 16.62 |
| Abbey Office Furnishings | 1 | 17.50 |
| Compuserve | 2 | 19.90 |
| Coffee Break Service | 1 | 41.80 |
| Roadway Package System, Inc | 4 | 43.67 |
| Evans Executone Inc | 1 | 95.00 |
| Pacific Bell | 6 | 171.01 |
| Edward Data Services | 1 | 207.78 |
| Dristas Groom & McCormick | 1 | 220.00 |

| # | Time | Action | Message |
|---|---|---|---|
| 1 | 20:43:14 | SELECT vendor_name, COUNT(invoice_id) "number_of_invoices", SUM(invoice_total) "sum_of_invoice_totals"  FROM  invoic... | 23 row(s) returned |

10. (5) Consider the following database. Show a query that displays the employee ids, employee names and their manager names, ordered by the employee_id. Also, show the query result.
Note: You should make this table first.

**ANSWER 10:**

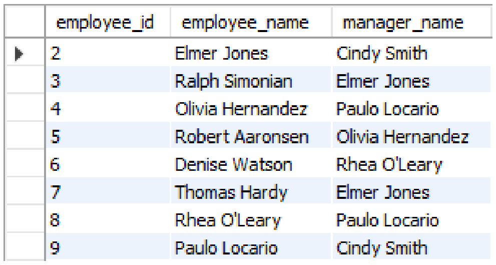
**SELECT DISTINCT e.employee_id, CONCAT(e.first_name, ' ', e.last_name) AS 'employee_name', CONCAT(m.first_name, ' ', m.last_name) AS 'manager_name'**

**FROM employees e , employees m**

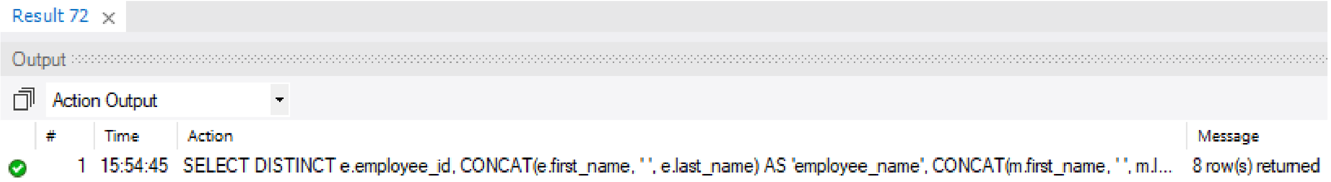
**WHERE e.manager_id = m.employee_id**

**ORDER BY employee_id**

**SCREENSHOT 10:**



| employee_id | employee_name | manager_name |
|---|---|---|
| 2 | Elmer Jones | Cindy Smith |
| 3 | Ralph Simonian | Elmer Jones |
| 4 | Olivia Hernandez | Paulo Locario |
| 5 | Robert Aaronsen | Olivia Hernandez |
| 6 | Denise Watson | Rhea O'Leary |
| 7 | Thomas Hardy | Elmer Jones |
| 8 | Rhea O'Leary | Paulo Locario |
| 9 | Paulo Locario | Cindy Smith |



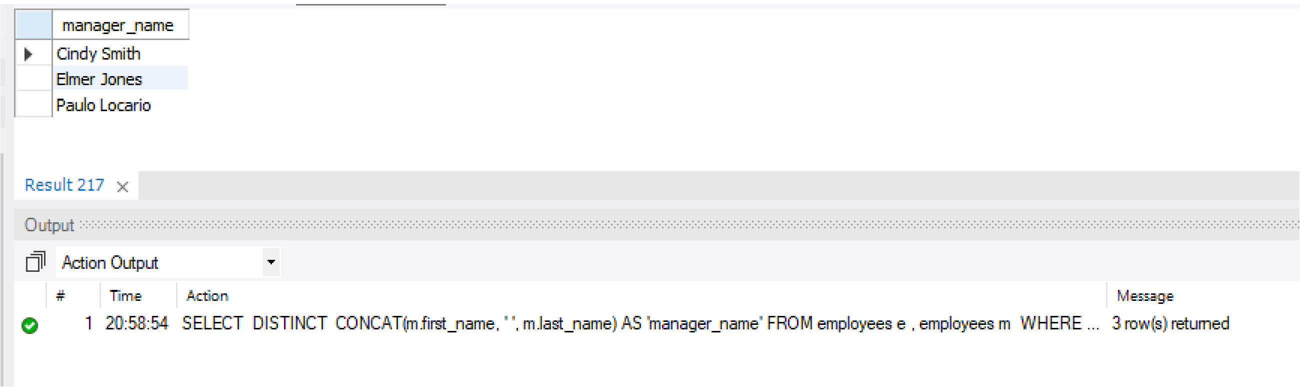
| # | Time | Action | Message |
|---|---|---|---|
| 1 | 15:54:45 | SELECT DISTINCT e.employee_id, CONCAT(e.first_name, ' ', e.last_name) AS 'employee_name', CONCAT(m.first_name, ' ', m.l... | 8 row(s) returned |

11. (5) From the above question, show a query that displays employees who manage more than one employee.

**ANSWER 11:**

```
SELECT  DISTINCT  CONCAT(m.first_name, ' ', m.last_name) AS 'manager_name'

FROM employees e , employees m

WHERE e.manager_id = m.employee_id

GROUP BY CONCAT(m.first_name, ' ', m.last_name)

HAVING COUNT(e.manager_id = m.employee_id) > 9
```

**SCREENSHOT 11:**



| manager_name |
|---|
| Cindy Smith |
| Elmer Jones |
| Paulo Locario |

```
#   Time      Action                                                                                             Message
1   20:58:54  SELECT  DISTINCT  CONCAT(m.first_name, ' ', m.last_name) AS 'manager_name' FROM employees e , employees m  WHERE ...   3 row(s) returned
```

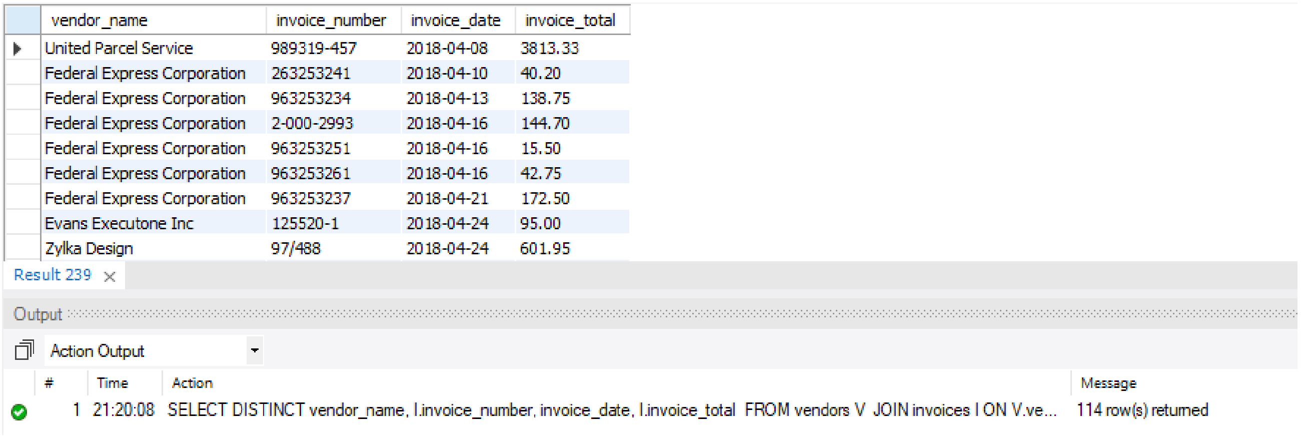
12. (5) Use a correlated subquery to return one row per vendor, representing the vendor’s oldest

invoice (the one with the earliest date). Each row should include these three column: vendor_name, invoice_number, and invoice_total. Sort the result by the invoice_number column.

**ANSWER 12:**

```
SELECT DISTINCT vendor_name, I.invoice_number, invoice_date, I.invoice_total

FROM vendors V

JOIN invoices I ON V.vendor_id = I.vendor_id

WHERE vendor_name <= (SELECT DISTINCT vendor_name

FROM invoices

JOIN vendors

ON V.vendor_id = vendors.vendor_id

HAVING MIN(invoice_date))

GROUP BY vendor_name, I.invoice_number, invoice_date, I.invoice_total
```

**SCREENSHOT 12:**



| vendor_name | invoice_number | invoice_date | invoice_total |
|---|---|---|---|
| United Parcel Service | 989319-457 | 2018-04-08 | 3813.33 |
| Federal Express Corporation | 263253241 | 2018-04-10 | 40.20 |
| Federal Express Corporation | 963253234 | 2018-04-13 | 138.75 |
| Federal Express Corporation | 2-000-2993 | 2018-04-16 | 144.70 |
| Federal Express Corporation | 963253251 | 2018-04-16 | 15.50 |
| Federal Express Corporation | 963253261 | 2018-04-16 | 42.75 |
| Federal Express Corporation | 963253237 | 2018-04-21 | 172.50 |
| Evans Executone Inc | 125520-1 | 2018-04-24 | 95.00 |
| Zylka Design | 97/488 | 2018-04-24 | 601.95 |

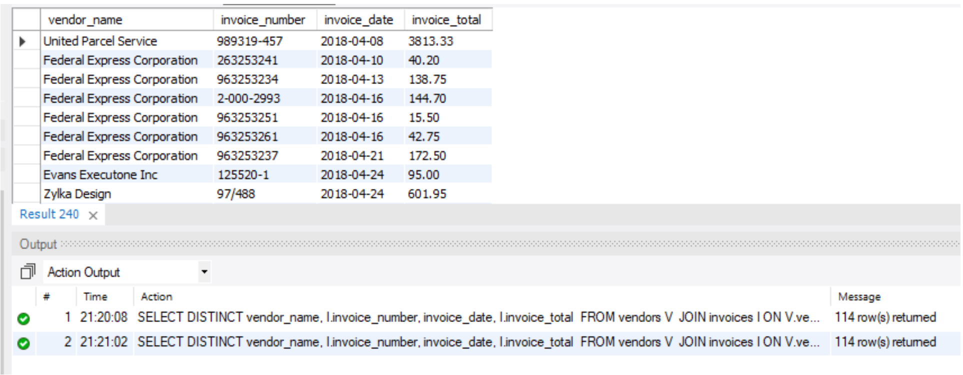
```
#   Time      Action                                                                                          Message
1   21:20:08  SELECT DISTINCT vendor_name, I.invoice_number, invoice_date, I.invoice_total  FROM vendors V  JOIN invoices I ON V.ve...   114 row(s) returned
```

13. (5) Rewrite the query for the above problem that does not use a correlated query.

**ANSWER 13:**

```
 SELECT DISTINCT vendor_name, I.invoice_number, invoice_date, I.invoice_total
FROM vendors V
JOIN invoices I ON V.vendor_id = I.vendor_id
GROUP BY vendor_name, I.invoice_number, invoice_date, I.invoice_total
HAVING MIN(invoice_date)
```

**SCREENSHOT 13:**



| vendor_name | invoice_number | invoice_date | invoice_total |
|---|---|---|---|
| United Parcel Service | 989319-457 | 2018-04-08 | 3813.33 |
| Federal Express Corporation | 263253241 | 2018-04-10 | 40.20 |
| Federal Express Corporation | 963253234 | 2018-04-13 | 138.75 |
| Federal Express Corporation | 2-000-2993 | 2018-04-16 | 144.70 |
| Federal Express Corporation | 963253251 | 2018-04-16 | 15.50 |
| Federal Express Corporation | 963253261 | 2018-04-16 | 42.75 |
| Federal Express Corporation | 963253237 | 2018-04-21 | 172.50 |
| Evans Executone Inc | 125520-1 | 2018-04-24 | 95.00 |
| Zylka Design | 97/488 | 2018-04-24 | 601.95 |

| # | Time | Action | Message |
|---|---|---|---|
| 1 | 21:20:08 | SELECT DISTINCT vendor_name, I.invoice_number, invoice_date, I.invoice_total FROM vendors V JOIN invoices I ON V.ve... | 114 row(s) returned |
| 2 | 21:21:02 | SELECT DISTINCT vendor_name, I.invoice_number, invoice_date, I.invoice_total FROM vendors V JOIN invoices I ON V.ve... | 114 row(s) returned |