

# MYSQL LAB 4 JOIN (Select –From –Join –On–Where–Order By)

## 1) INNER JOIN

**SELECT** invoice\_number, vendor\_name

**FROM** vendors **INNER JOIN** invoices

**ON** vendors.vendor\_id = invoices.vendor\_id

**ORDER BY** invoice\_number;

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' tab with a tree view of databases: ap, ex, and sys. The 'ap' database is selected, showing its tables, views, stored procedures, and functions. The main editor window displays a SQL query in 'Query 1':

```
1 • use ap;
2 • SELECT invoice_number, vendor_name
3 • FROM vendors INNER JOIN invoices
4 • ON vendors.vendor_id = invoices.vendor_id
5 • ORDER BY invoice_number;
```

Below the query editor, the 'Result Grid' shows the results of the query. It has two columns: 'invoice\_number' and 'vendor\_name'. The results are as follows:

invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation
1-202-2978	Federal Express Corporation
10843	Yesmed, Inc
109596	Coffee Break Service
111-070-10007	Pacific Bell

At the bottom left, the 'Connection Details' are shown:

**Connection Details**  
Name: navneet  
Host: 127.0.0.1  
Port: 55515  
Login User: root  
Current User: root@localhost  
SSL cipher: DHE-RSA-AES128-GCM-

At the bottom right, the 'Output' tab shows the 'Action Output' log:

#	Time	Action	Message
1	17:59:24	use ap	0 row(s) affected
2	17:59:24	SELECT invoice_number, vendor_name FROM vendors INNER JOIN invoices ON vendors.vendor_id = i...	114 row(s) returned

## 2)

i) **SELECT** invoice\_number, vendor\_name, invoice\_due\_date,

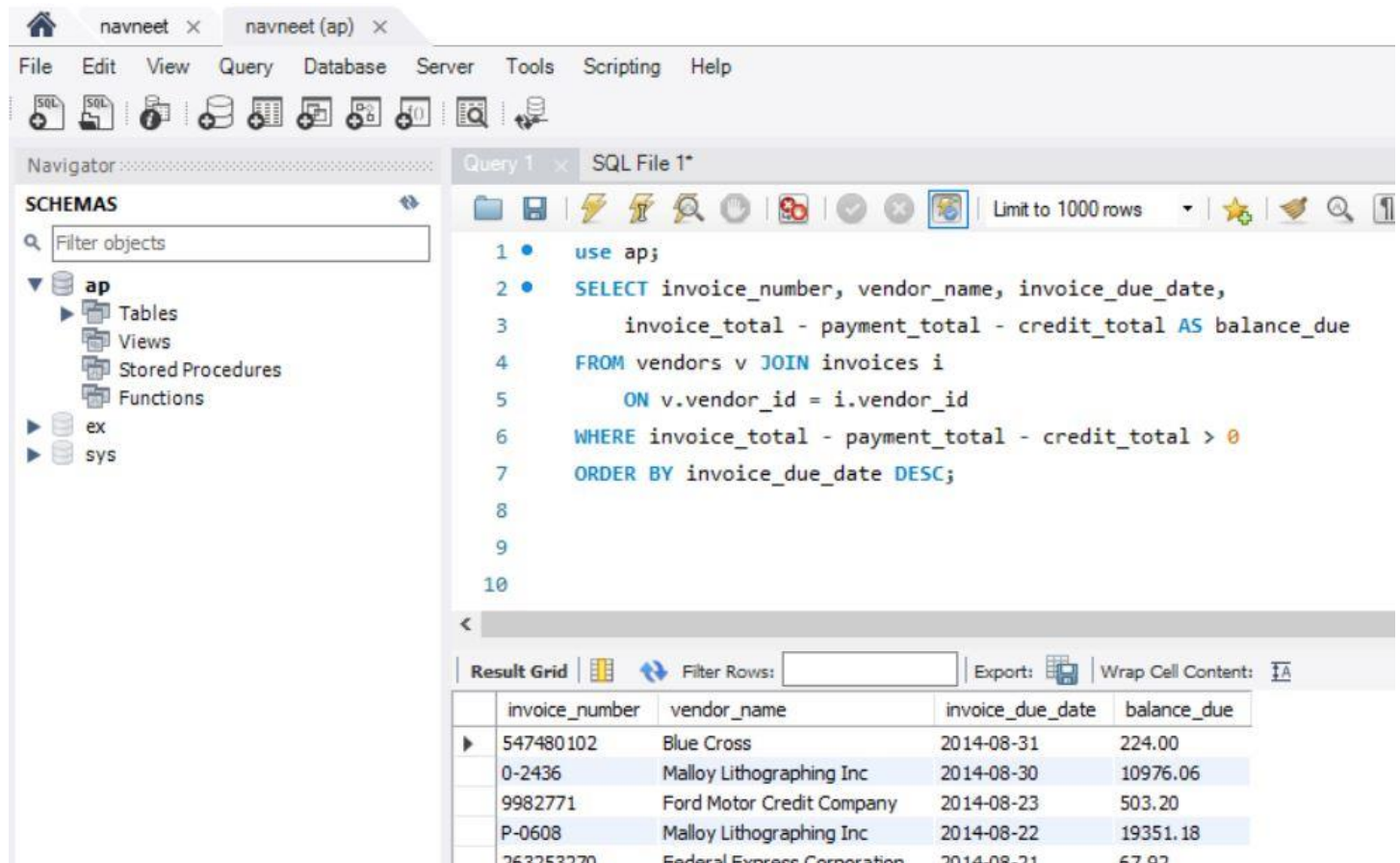
invoice\_total - payment\_total - credit\_total AS balance\_due

**FROM** vendors v **JOIN** invoices i

**ON** v.vendor\_id = i.vendor\_id

**WHERE** invoice\_total - payment\_total - credit\_total > 0

**ORDER BY** invoice\_due\_date **DESC**;



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left pane displays a 'SCHEMAS' tree with 'ap' expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main editor shows a SQL query in 'Query 1':

```
1 use ap;
2 SELECT invoice_number, vendor_name, invoice_due_date,
3        invoice_total - payment_total - credit_total AS balance_due
4 FROM vendors v JOIN invoices i
5      ON v.vendor_id = i.vendor_id
6 WHERE invoice_total - payment_total - credit_total > 0
7 ORDER BY invoice_due_date DESC;
```

The bottom pane shows the 'Result Grid' with 5 rows of data:

invoice_number	vendor_name	invoice_due_date	balance_due
547480102	Blue Cross	2014-08-31	224.00
0-2436	Malloy Lithographing Inc	2014-08-30	10976.06
9982771	Ford Motor Credit Company	2014-08-23	503.20
P-0608	Malloy Lithographing Inc	2014-08-22	19351.18
263253270	Federal Express Corporation	2014-08-21	67.92

ii) **SELECT** invoice\_number, line\_item\_amount, line\_item\_description

**FROM** invoices **JOIN** invoice\_line\_items line\_items

**ON** invoices.invoice\_id = line\_items.invoice\_id

**WHERE** account\_number = 540

**ORDER BY** invoice\_date;

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'ap' expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The right pane shows a query window with the following SQL code:

```

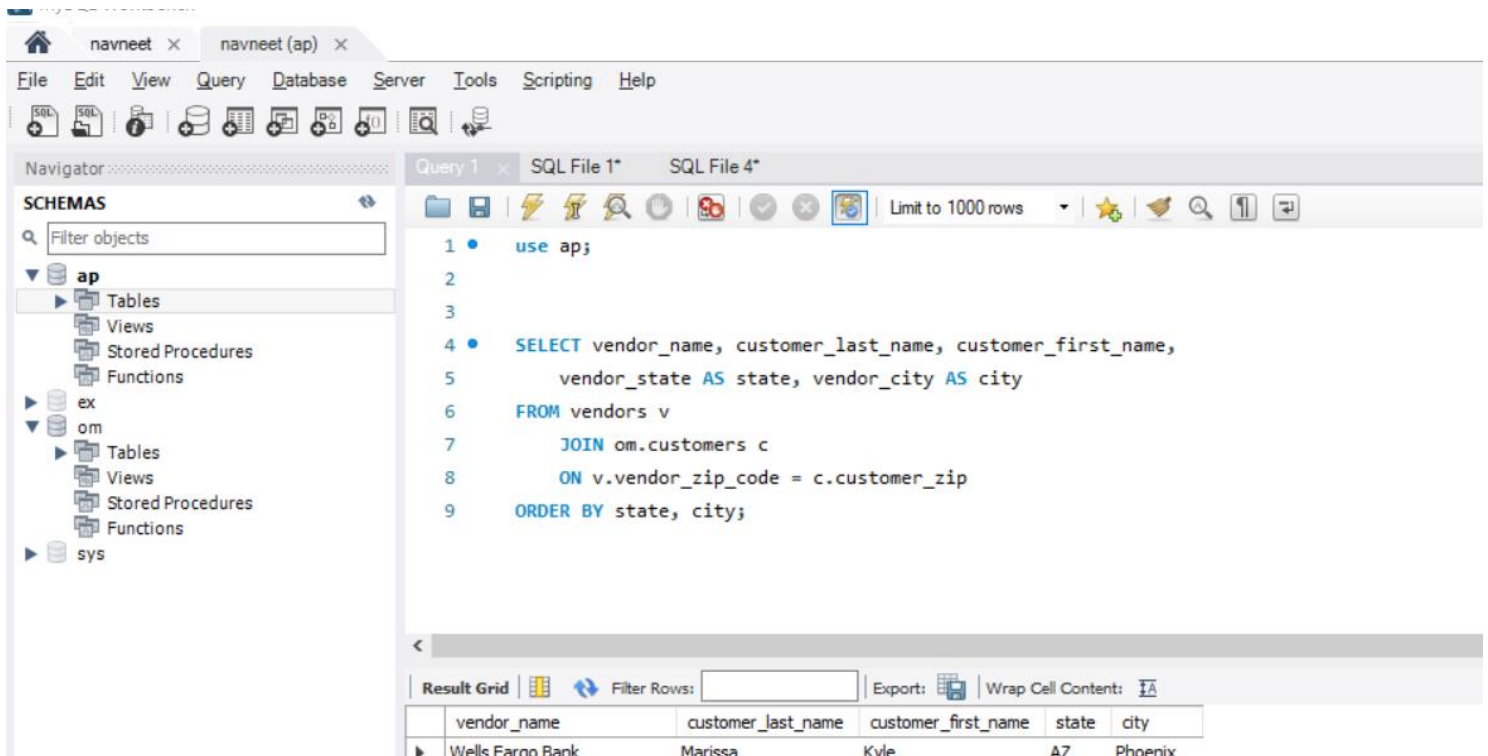
1 use ap;
2
3
4 SELECT invoice_number, line_item_amount, line_item_description
5 FROM invoices JOIN invoice_line_items line_items
6     ON invoices.invoice_id = line_items.invoice_id
7 WHERE account_number = 540
8 ORDER BY invoice_date;

```

Below the query window, the 'Result Grid' is displayed, showing the following data:

invoice_number	line_item_amount	line_item_description
177271-001	478.00	Publishers Marketing
972110	207.78	Prospect list
133560	175.00	Card deck advertising
97/522	765.13	Catalog design

3) **SELECT** vendor\_name, customer\_last\_name, customer\_first\_name,  
 vendor\_state AS state, vendor\_city AS city  
**FROM** vendors v  
**JOIN** om.customers c  
**ON** v.vendor\_zip\_code = c.customer\_zip  
**ORDER BY** state, city;



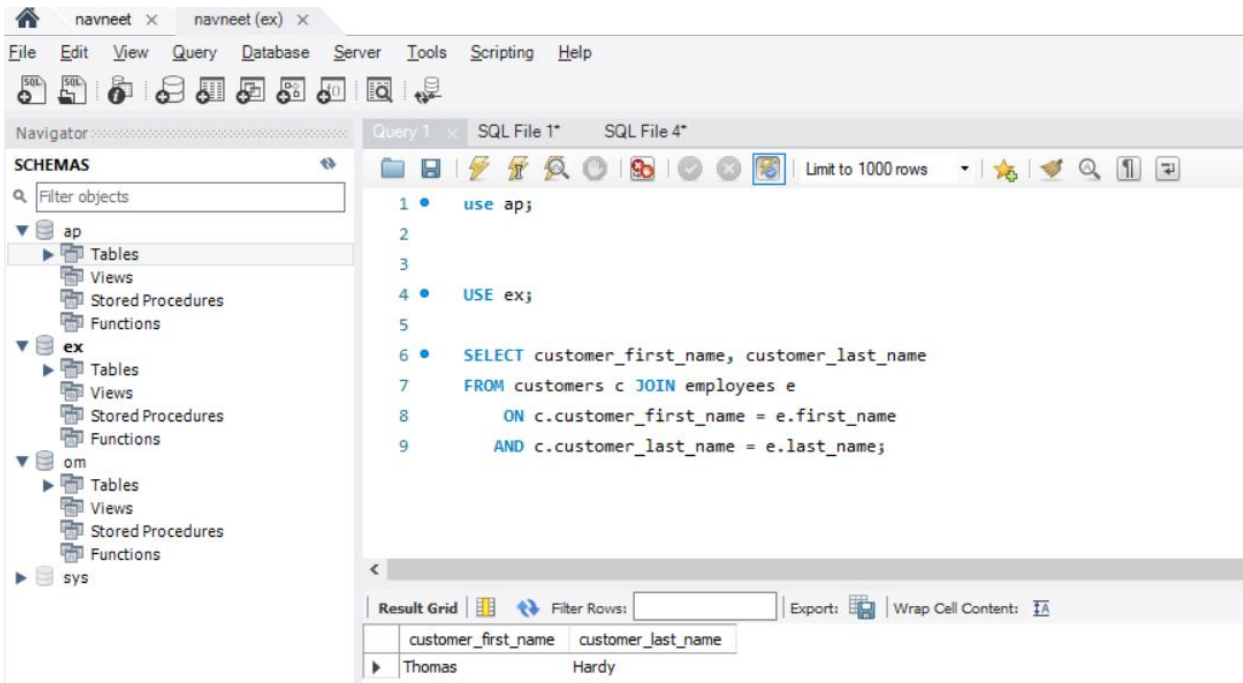
#### 4) JOIN

**SELECT** customer\_first\_name, customer\_last\_name

**FROM** customers c **JOIN** employees e

**ON** c.customer\_first\_name = e.first\_name

**AND** c.customer\_last\_name = e.last\_name;



5) **SELECT DISTINCT v1.vendor\_name, v1.vendor\_city,**

**v1.vendor\_state**

**FROM vendors v1 JOIN vendors v2**

**ON v1.vendor\_city = v2.vendor\_city AND**

**v1.vendor\_state = v2.vendor\_state AND**

**v1.vendor\_name <> v2.vendor\_name**

**ORDER BY v1.vendor\_state, v1.vendor\_city;**

## Answer:

The screenshot shows the SQL Enterprise Manager interface. The left pane displays the 'Schemas' tree with folders for 'ap', 'ex', 'om', and 'sys'. The 'ap' folder is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The right pane shows a query window with the following SQL code:

```
1 use ap;
2
3
4
5 SELECT DISTINCT v1.vendor_name, v1.vendor_city,
6                 v1.vendor_state
7 FROM vendors v1 JOIN vendors v2
8     ON v1.vendor_city = v2.vendor_city AND
9        v1.vendor_state = v2.vendor_state AND
10        v1.vendor_name <> v2.vendor_name
11 ORDER BY v1.vendor_state, v1.vendor_city;
```

Below the query window, the 'Result Grid' is displayed with the following data:

vendor_name	vendor_city	vendor_state
Computer Library	Phoenix	AZ
AT&T	Phoenix	AZ
Wells Fargo Bank	Phoenix	AZ
Aztek Label	Anaheim	CA
Blue Shield of California	Anaheim	CA

6) **SELECT** vendor\_name, invoice\_number, invoice\_date,

line\_item\_amount, account\_description

**FROM** vendors v

**JOIN** invoices i

**ON** v.vendor\_id = i.vendor\_id

**JOIN** invoice\_line\_items li

**ON** i.invoice\_id = li.invoice\_id

**JOIN** general\_ledger\_accounts gl

**ON** li.account\_number = gl.account\_number

**WHERE** invoice\_total - payment\_total - credit\_total > 0

**ORDER BY** vendor\_name, line\_item\_amount **DESC**;

The screenshot shows a SQL Enterprise Manager interface with a query editor and a results grid. The query editor contains the following SQL code:

```

1 • use ap;
2
3
4
5 • SELECT vendor_name, invoice_number, invoice_date,
6         line_item_amount, account_description
7 FROM vendors v
8     JOIN invoices i
9         ON v.vendor_id = i.vendor_id
10    JOIN invoice_line_items li
11        ON i.invoice_id = li.invoice_id
12    JOIN general_ledger_accounts gl
13        ON li.account_number = gl.account_number
14 WHERE invoice_total - payment_total - credit_total > 0
15 ORDER BY vendor_name, line_item_amount DESC;

```

The results grid shows the following data:

vendor_name	invoice_number	invoice_date	line_item_amount	account_description
Blue Cross	547480102	2014-08-01	224.00	Group Insurance

## 7)TABLE REFERENCE

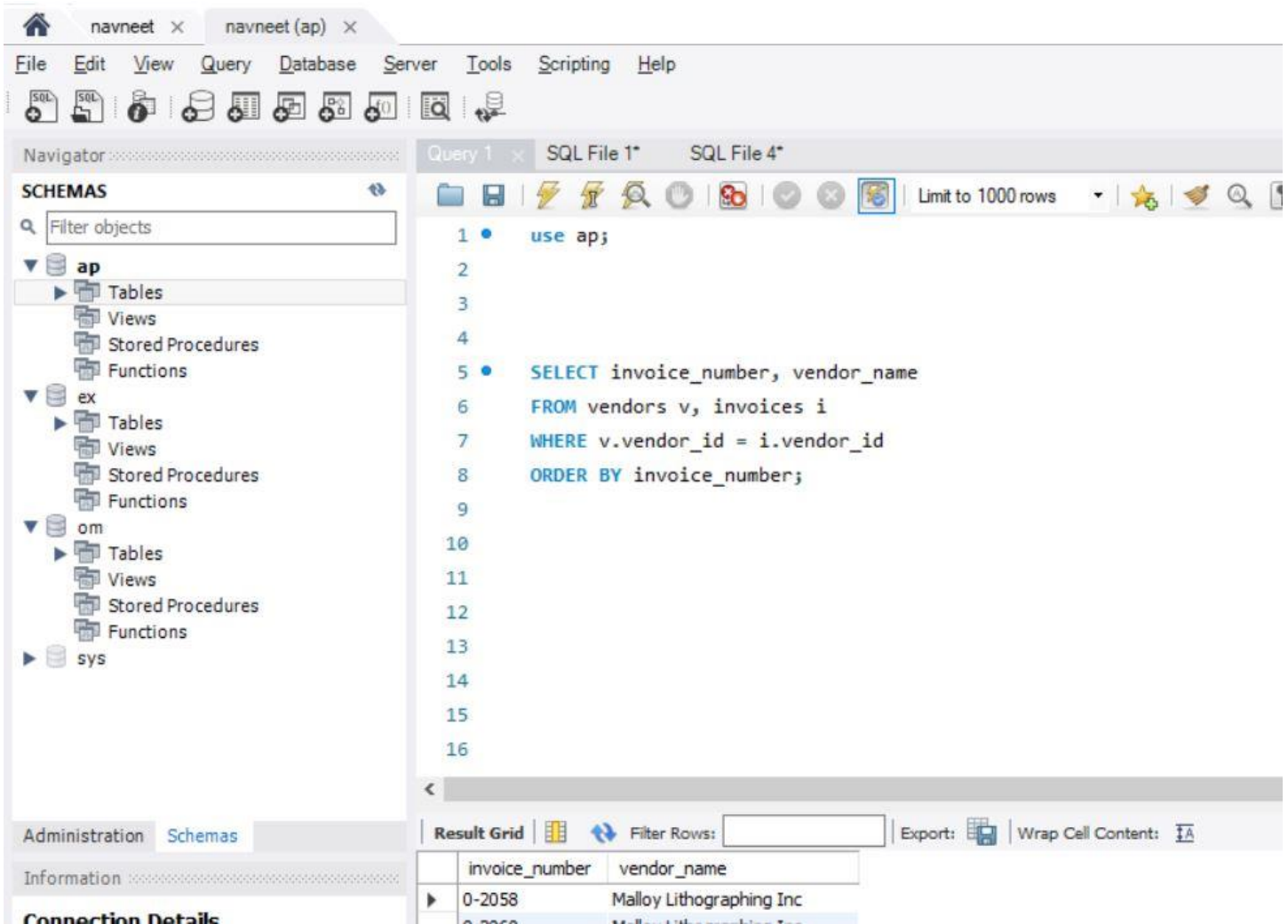
i)SELECT invoice\_number, vendor\_name

FROM vendors v, invoices i

WHERE v.vendor\_id = i.vendor\_id

ORDER BY invoice\_number;





ii) **SELECT** vendor\_name, invoice\_number, invoice\_date,  
line\_item\_amount, account\_description  
**FROM** vendors **v**, invoices **i**, invoice\_line\_items **li**,  
general\_ledger\_accounts **gl**  
**WHERE** v.vendor\_id = i.vendor\_id  
**AND** i.invoice\_id = li.invoice\_id  
**AND** li.account\_number = gl.account\_number  
**AND** invoice\_total - payment\_total - credit\_total > 0  
**ORDER BY** vendor\_name, line\_item\_amount **DESC**;



The screenshot shows the SQL Developer interface with a query window open. The query is as follows:

```

1  use ap;
2
3
4  SELECT vendor_name, invoice_number, invoice_date,
5         line_item_amount, account_description
6  FROM   vendors v, invoices i, invoice_line_items li,
7         general_ledger_accounts gl
8  WHERE  v.vendor_id = i.vendor_id
9         AND i.invoice_id = li.invoice_id
10         AND li.account_number = gl.account_number
11         AND invoice_total - payment_total - credit_total > 0
12  ORDER BY vendor_name, line_item_amount DESC;

```

The results grid shows the following data:

vendor_name	invoice_number	invoice_date	line_item_amount	account_description
Blue Cross	547480102	2014-08-01	224.00	Group Insurance

## 8) LEFT JOIN

**SELECT** vendor\_name, invoice\_number, invoice\_total

**FROM** vendors **LEFT JOIN** invoices

**ON** vendors.vendor\_id = invoices.vendor\_id

**ORDER BY** vendor\_name;

Answer:

The screenshot shows the SQL Developer interface with a query window and a results grid. The query is as follows:

```
1 • use ap;
2
3 • SELECT vendor_name, invoice_number, invoice_total
4 FROM vendors LEFT JOIN invoices
5     ON vendors.vendor_id = invoices.vendor_id
6 ORDER BY vendor_name;
```

The results grid displays the following data:

vendor_name	invoice_number	invoice_total
Abbey Office Furnishings	203339-13	17.50
American Booksellers Assoc	NULL	NULL
American Express	NULL	NULL

## 9)JOIN ,LEFT JOIN

i)SELECT department\_name, d.department\_number, last\_name

FROM departments d

LEFT JOIN employees e

ON d.department\_number = e.department\_number

ORDER BY department\_name;

ii)SELECT department\_name, e.department\_number, last\_name

```
FROM departments d  
  
    RIGHT JOIN employees e  
  
        ON d.department_number = e.department_number  
  
ORDER BY department_name;
```

```
iii)SELECT department_name, last_name, project_number  
  
FROM departments d  
  
    LEFT JOIN employees e  
  
        ON d.department_number = e.department_number  
  
    LEFT JOIN projects p  
  
        ON e.employee_id = p.employee_id  
  
ORDER BY department_name, last_name;
```

```
iv)SELECT department_name, last_name, project_number  
  
FROM departments d  
  
    JOIN employees e  
  
        ON d.department_number = e.department_number  
  
    LEFT JOIN projects p  
  
        ON e.employee_id = p.employee_id  
  
ORDER BY department_name, last_name;
```

The screenshot shows the Navicat for MySQL interface. On the left is the 'SCHEMAS' navigator with a tree view showing databases 'ap', 'ex', 'om', and 'sys'. The 'ex' database is selected, showing its tables, views, stored procedures, and functions. The main window displays a SQL query in 'Query 1':

```

1 • use ex;
2
3 • SELECT department_name, last_name, project_number
4 FROM departments d
5     JOIN employees e
6     ON d.department_number = e.department_number
7     LEFT JOIN projects p
8     ON e.employee_id = p.employee_id
9 ORDER BY department_name, last_name;
10

```

Below the query editor is the 'Result Grid' showing the output of the query. It has columns for 'department\_name', 'last\_name', and 'project\_number'. The results are as follows:

department_name	last_name	project_number
Accounting	Hernandez	P1011
Maintenance	Hardy	NULL
Payroll	Aaronsen	P1012
Payroll	Simonian	P1012

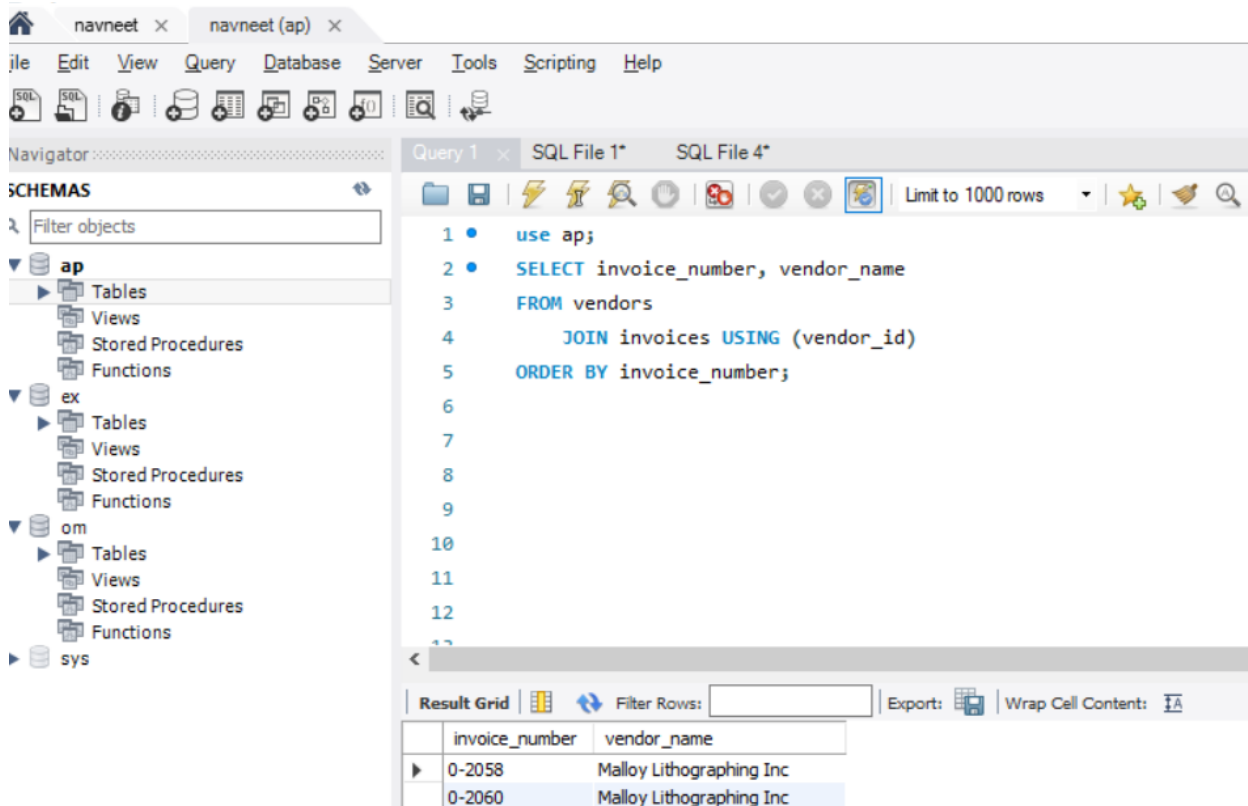
## 10) JOIN,LEFT JOIN,USING,ORDER BY

i) **SELECT** invoice\_number, vendor\_name

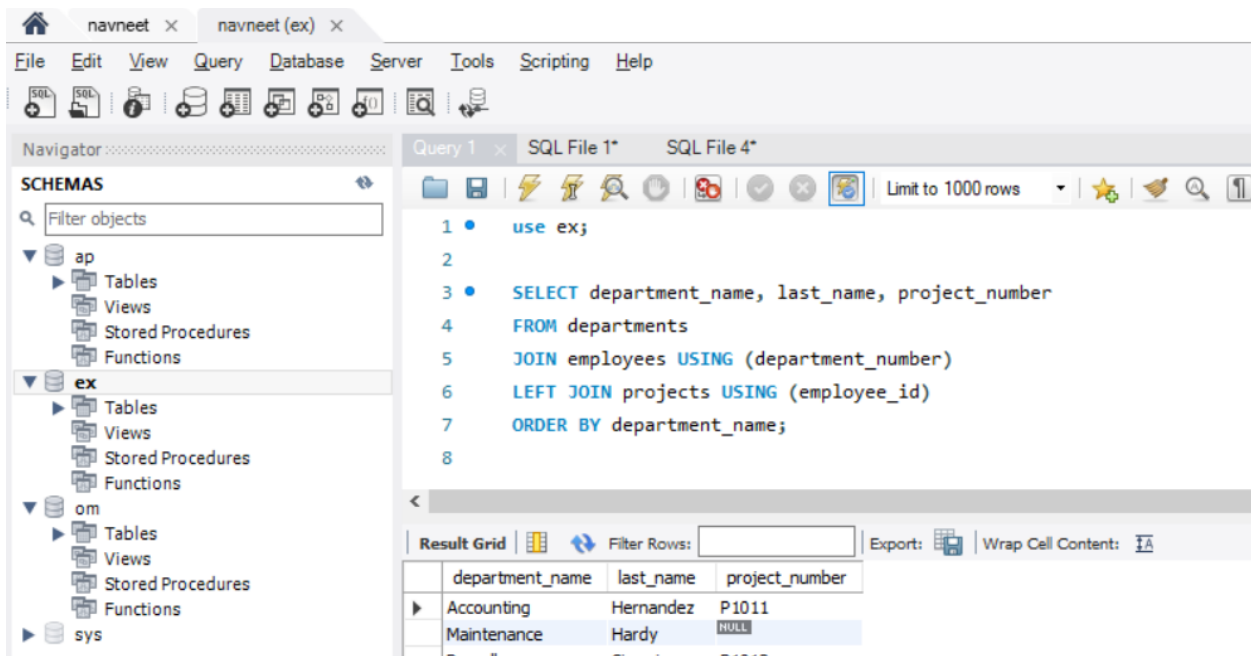
**FROM** vendors

**JOIN** invoices **USING** (vendor\_id)

**ORDER BY** invoice\_number;



ii)SELECT department\_name, last\_name, project\_number  
FROM departments  
JOIN employees USING (department\_number)  
LEFT JOIN projects USING (employee\_id)  
ORDER BY department\_name;



## 11) NATURAL JOIN

i) USE ap;

**SELECT** invoice\_number, vendor\_name

**FROM** vendors

**NATURAL JOIN** invoices

**ORDER BY** invoice\_number;



The screenshot shows a SQL IDE with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The Navigator pane on the left shows a tree structure of schemas: ap, ex, om, and sys. The main editor displays a SQL query in 'Query 1'.

```

1 • USE ap;
2
3 • SELECT invoice_number, vendor_name
4 FROM vendors
5     NATURAL JOIN invoices
6 ORDER BY invoice_number;

```

Below the query editor is a 'Result Grid' showing the results of the query. It has columns 'invoice\_number' and 'vendor\_name'.

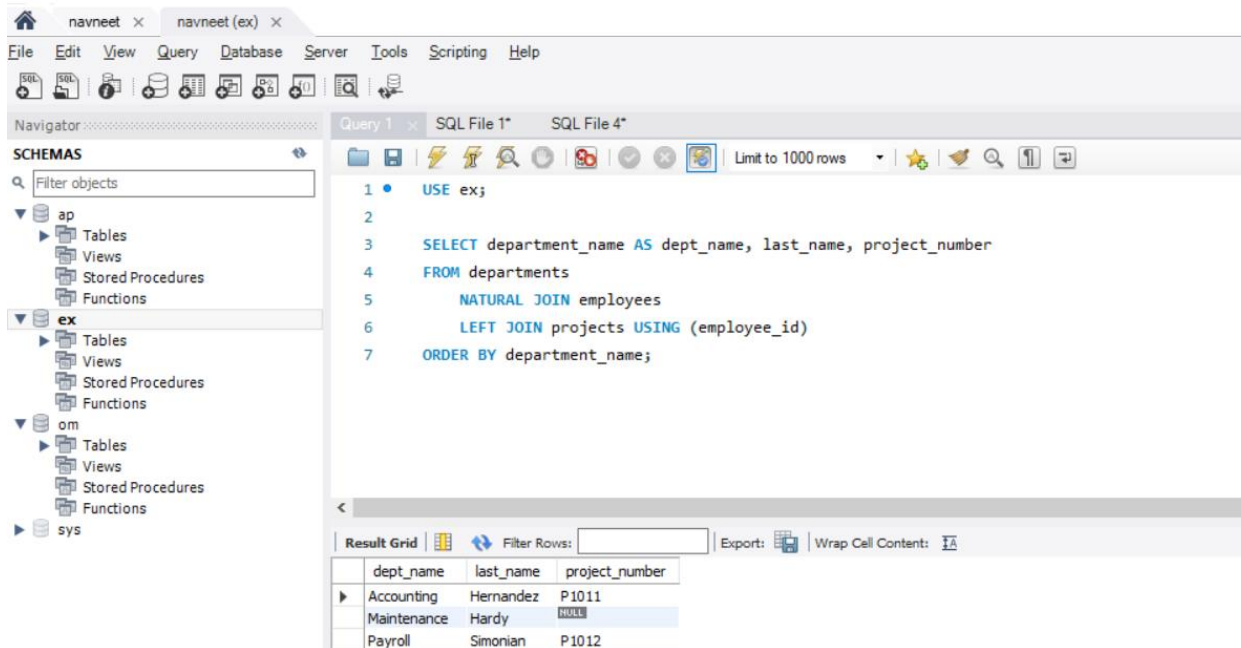
invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc

ii)USE ex;

```

SELECT department_name AS dept_name, last_name, project_number
FROM departments
    NATURAL JOIN employees
    LEFT JOIN projects USING (employee_id)
ORDER BY department_name;

```



Query 1

```

1 • USE ex;
2
3 SELECT department_name AS dept_name, last_name, project_number
4 FROM departments
5     NATURAL JOIN employees
6     LEFT JOIN projects USING (employee_id)
7 ORDER BY department_name;

```

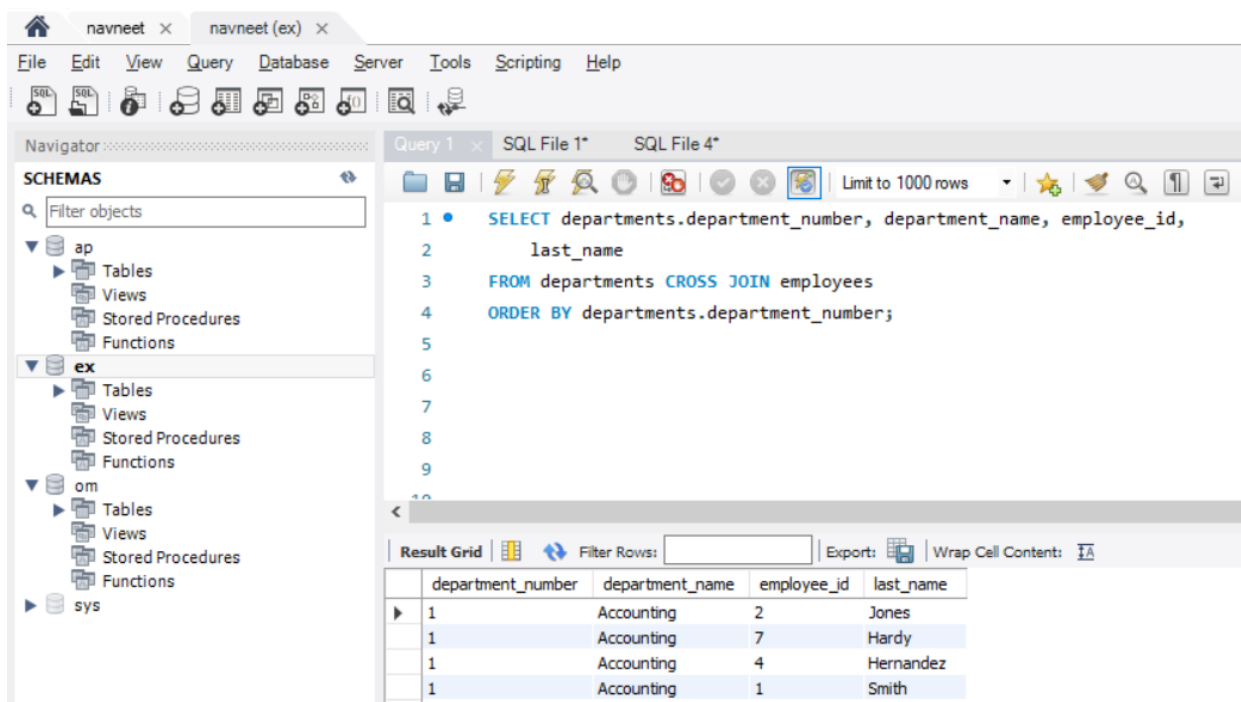
dept_name	last_name	project_number
Accounting	Hernandez	P1011
Maintenance	Hardy	NULL
Payroll	Simonian	P1012

## 12)CROSS JOIN

i)SELECT departments.department\_number, department\_name, employee\_id,  
last\_name

FROM departments CROSS JOIN employees

ORDER BY departments.department\_number;



Query 1

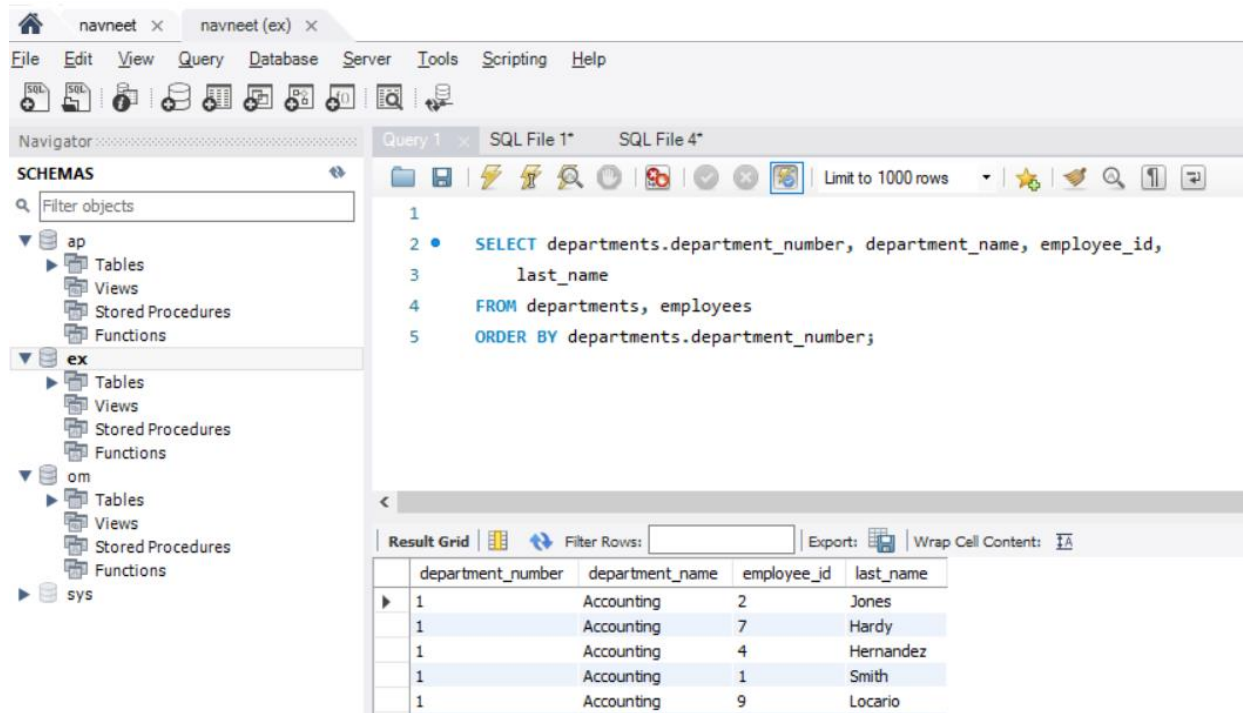
```

1 • SELECT departments.department_number, department_name, employee_id,
2     last_name
3 FROM departments CROSS JOIN employees
4 ORDER BY departments.department_number;
5
6
7
8
9
10

```

department_number	department_name	employee_id	last_name
1	Accounting	2	Jones
1	Accounting	7	Hardy
1	Accounting	4	Hernandez
1	Accounting	1	Smith

```
ii)SELECT departments.department_number, department_name, employee_id,  
  
last_name  
  
FROM departments, employees  
  
ORDER BY departments.department_number;
```

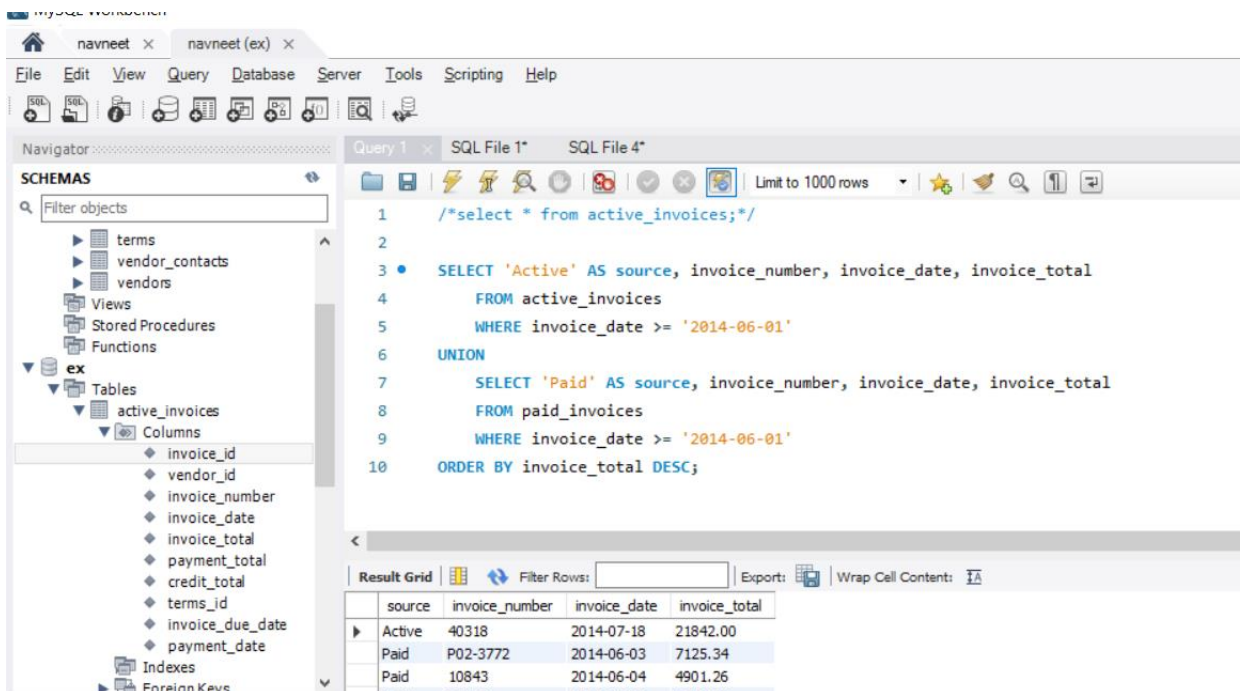


### 13) UNION

```
SELECT 'Active' AS source, invoice_number, invoice_date, invoice_total  
  
FROM active_invoices  
  
WHERE invoice_date >= '2014-06-01'
```

### UNION

```
SELECT 'Paid' AS source, invoice_number, invoice_date, invoice_total  
  
FROM paid_invoices  
  
WHERE invoice_date >= '2014-06-01'  
  
ORDER BY invoice_total DESC;
```



## 14) JOIN, UNION

**SELECT 'Active' AS source, invoice\_number, invoice\_date, invoice\_total**

**FROM** invoices

**WHERE** invoice\_total - payment\_total - credit\_total > 0

**UNION**

**SELECT 'Paid' AS source, invoice\_number, invoice\_date, invoice\_total**

**FROM** invoices

**WHERE** invoice\_total - payment\_total - credit\_total <= 0

**ORDER BY invoice\_total DESC;**

The screenshot shows a SQL IDE interface. On the left, a 'SCHEMAS' pane displays a tree view with 'ap' and 'ex' databases. Under 'ap', there are 'Tables' (general\_ledger\_accounts, invoice\_archive, invoice\_line\_items, invoices, terms, vendor\_contacts, vendors) and 'Views' (Stored Procedures, Functions). The 'active\_invoices' table is selected under 'ex'. The main pane shows a SQL query in 'Query 1' with the following code:

```
1 use ap;
2 SELECT 'Active' AS source, invoice_number, invoice_date, invoice_total
3 FROM invoices
4 WHERE invoice_total - payment_total - credit_total > 0
5 UNION
6 SELECT 'Paid' AS source, invoice_number, invoice_date, invoice_total
7 FROM invoices
8 WHERE invoice_total - payment_total - credit_total <= 0
9 ORDER BY invoice_total DESC;
10
11
```

Below the query, the 'Result Grid' shows the following data:

	source	invoice_number	invoice_date	invoice_total
▶	Paid	0-2058	2014-05-28	37966.19
	Paid	P-0259	2014-07-19	26881.40
	Paid	0-2060	2014-07-24	23517.58
	Paid	40318	2014-06-01	21842.00

ii) **SELECT** invoice\_number, vendor\_name, '33% Payment' AS payment\_type,

invoice\_total AS total, invoice\_total \* 0.333 AS payment

**FROM** invoices **JOIN** vendors

**ON** invoices.vendor\_id = vendors.vendor\_id

**WHERE** invoice\_total > 10000

**UNION**

**SELECT** invoice\_number, vendor\_name, '50% Payment' AS payment\_type,

invoice\_total AS total, invoice\_total \* 0.5 AS payment

**FROM** invoices **JOIN** vendors

**ON** invoices.vendor\_id = vendors.vendor\_id

**WHERE** invoice\_total BETWEEN 500 AND 10000

**UNION**

**SELECT** invoice\_number, vendor\_name, 'Full amount' AS payment\_type,

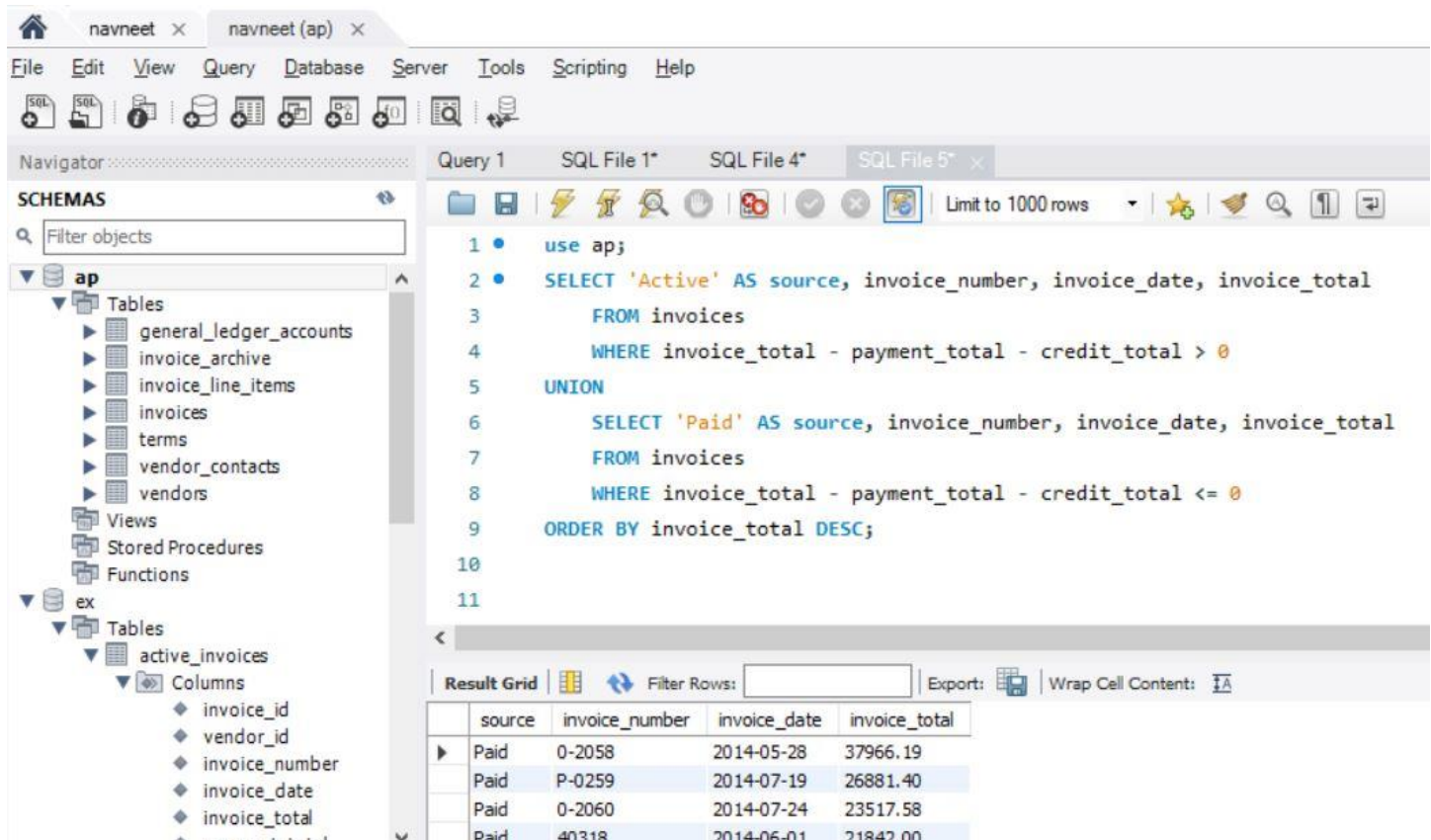
invoice\_total AS total, invoice\_total AS payment

**FROM** invoices **JOIN** vendors

**ON** invoices.vendor\_id = vendors.vendor\_id

**WHERE** invoice\_total < 500

**ORDER BY** payment\_type, vendor\_name, invoice\_number;



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left pane displays a 'SCHEMAS' tree with 'ap' and 'ex' databases. The 'ap' database is expanded, showing tables like 'general\_ledger\_accounts', 'invoice\_archive', 'invoice\_line\_items', 'invoices', 'terms', 'vendor\_contacts', and 'vendors'. The 'ex' database is also expanded, showing 'active\_invoices' with columns 'invoice\_id', 'vendor\_id', 'invoice\_number', 'invoice\_date', and 'invoice\_total'. The main query editor shows a SQL query:   
1 use ap;  
2 SELECT 'Active' AS source, invoice\_number, invoice\_date, invoice\_total  
3 FROM invoices  
4 WHERE invoice\_total - payment\_total - credit\_total > 0  
5 UNION  
6 SELECT 'Paid' AS source, invoice\_number, invoice\_date, invoice\_total  
7 FROM invoices  
8 WHERE invoice\_total - payment\_total - credit\_total <= 0  
9 ORDER BY invoice\_total DESC;  
10  
11  
The bottom pane shows the 'Result Grid' with 4 columns: source, invoice\_number, invoice\_date, and invoice\_total. It contains 4 rows of data: 

source	invoice_number	invoice_date	invoice_total
Paid	0-2058	2014-05-28	37966.19
Paid	P-0259	2014-07-19	26881.40
Paid	0-2060	2014-07-24	23517.58
Paid	40318	2014-06-01	21842.00

## 15) LEFT JOIN, UNION, RIGHT JOIN

**SELECT** department\_name AS dept\_name, d.department\_number AS d\_dept\_no,

e.department\_number AS e\_dept\_no, last\_name

**FROM** departments d

**LEFT JOIN** employees e

**ON** d.department\_number = e.department\_number

**UNION**

**SELECT** department\_name AS dept\_name, d.department\_number AS d\_dept\_no,

e.department\_number AS e\_dept\_no, last\_name



FROM departments d

RIGHT JOIN employees e

ON d.department\_number = e.department\_number

ORDER BY dept\_name;

Answer:

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains a SQL script that uses a UNION to combine three SELECT statements. The first SELECT statement filters for invoices with a total greater than 10000 and a 33% payment. The second SELECT statement filters for invoices with a total between 500 and 10000 and a 50% payment. The third SELECT statement filters for invoices with a total less than 500 and a full amount payment. The result grid shows the first two rows of the query results.

```
1 use ap;
2 SELECT invoice_number, vendor_name, '33% Payment' AS payment_type,
3       invoice_total AS total, invoice_total * 0.333 AS payment
4 FROM invoices JOIN vendors
5       ON invoices.vendor_id = vendors.vendor_id
6 WHERE invoice_total > 10000
7 UNION
8 SELECT invoice_number, vendor_name, '50% Payment' AS payment_type,
9       invoice_total AS total, invoice_total * 0.5 AS payment
10 FROM invoices JOIN vendors
11       ON invoices.vendor_id = vendors.vendor_id
12 WHERE invoice_total BETWEEN 500 AND 10000
13 UNION
14 SELECT invoice_number, vendor_name, 'Full amount' AS payment_type,
15       invoice_total AS total, invoice_total AS payment
16 FROM invoices JOIN vendors
17       ON invoices.vendor_id = vendors.vendor_id
18 WHERE invoice_total < 500
19 ORDER BY payment_type, vendor_name, invoice_number;
```

invoice_number	vendor_name	payment_type	total	payment
40318	Data Reproductions Corp	33% Payment	21842.00	7273.38600
0-2058	Mallory Lithodranhino Inc	33% Payment	37966.19	12642.74127

**Connection Details**  
Name: navneet  
Host: 127.0.0.1  
Port: 55515  
Login User: root  
Current User: root@localhost  
SSL cipher: DHE-RSA-AES128-GCM-