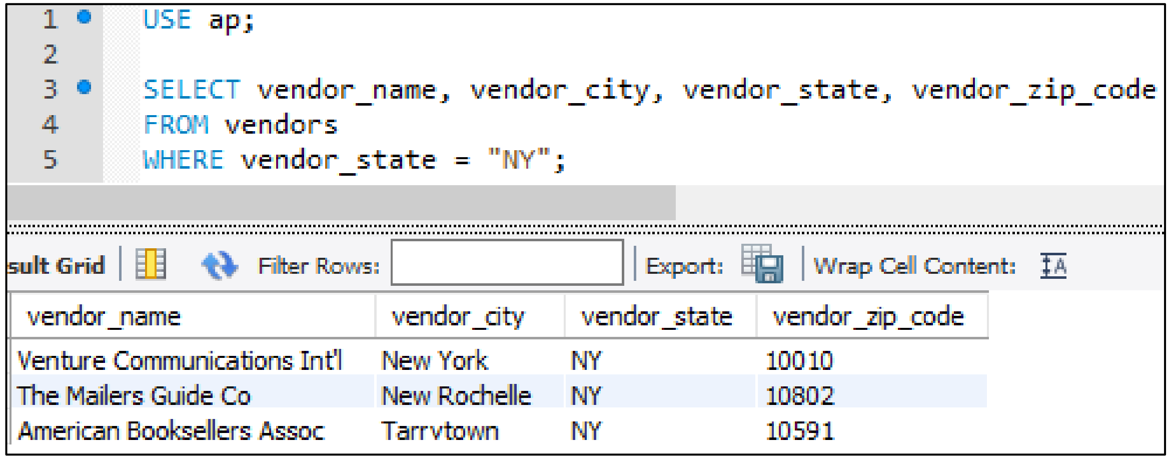
SQL: WHERE Statements, Cont.

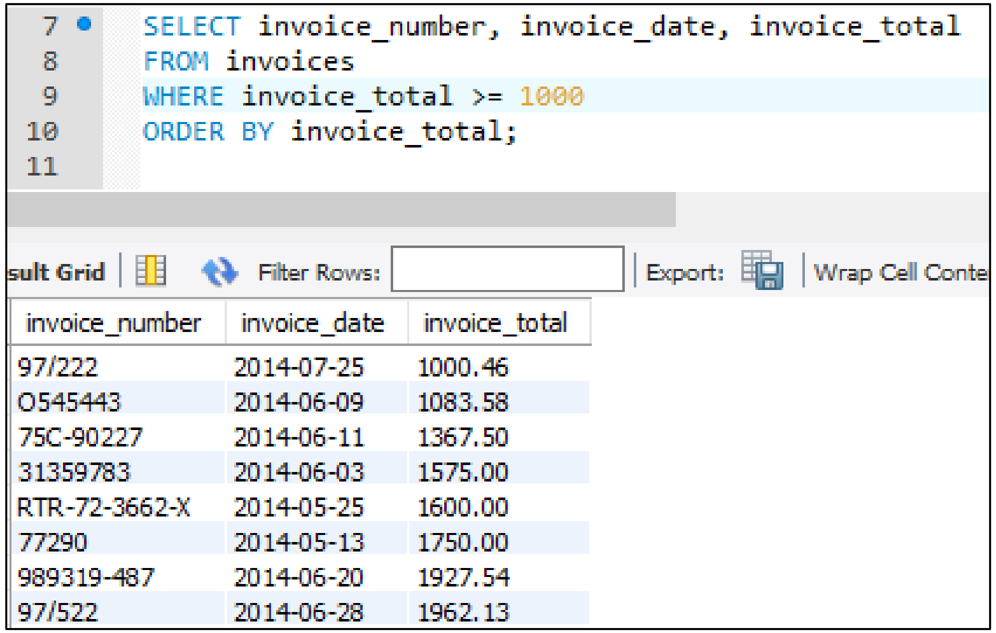
- Chapter 3 in Murach text
- SQL for advanced filters
  - Operators
  - Logicals
  - Working with data types – strings, dates, etc. (check Chap 8 for more references)
  - Functions (check Chap 9 for more references)
- Chapter 8 in Murach for more on data types
- Chapter 9 in Murach for more on functions, especially string and date functions

Recall, WHERE Statements



```
USE ap;

SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code
FROM vendors
WHERE vendor_state = "NY";
```

vendor_name	vendor_city	vendor_state	vendor_zip_code
Venture Communications Int'l	New York	NY	10010
The Mailers Guide Co	New Rochelle	NY	10802
American Booksellers Assoc	Tarrytown	NY	10591



```
SELECT invoice_number, invoice_date, invoice_total
FROM invoices
WHERE invoice_total >= 1000
ORDER BY invoice_total;
```

invoice_number	invoice_date	invoice_total
97/222	2014-07-25	1000.46
0545443	2014-06-09	1083.58
75C-90227	2014-06-11	1367.50
31359783	2014-06-03	1575.00
RTR-72-3662-X	2014-05-25	1600.00
77290	2014-05-13	1750.00
989319-487	2014-06-20	1927.54
97/522	2014-06-28	1962.13

SQL Operators

Relational comparison operators

- =
- <
- >
- <=
- >=
- <>
- !=

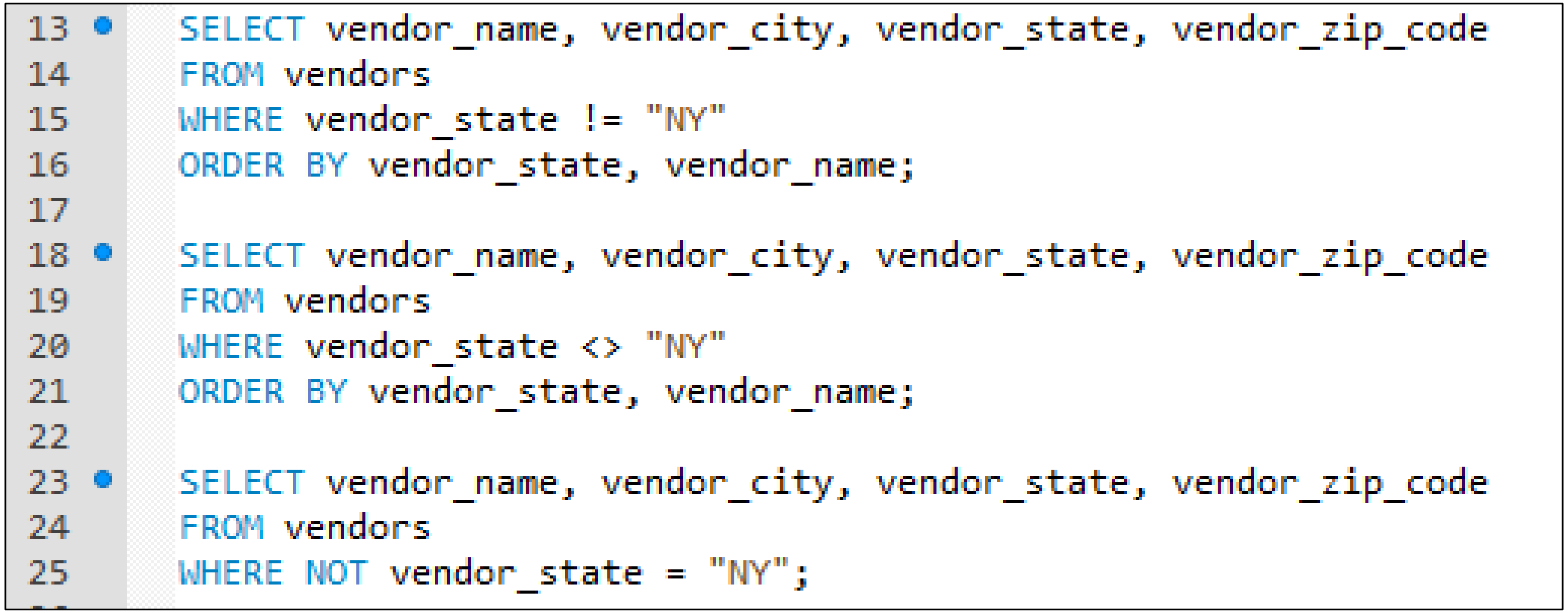
Comparison operators

- BETWEEN
- IN
- LIKE
- IS
- GREATEST/LEAST
- COALESCE

Logical operators

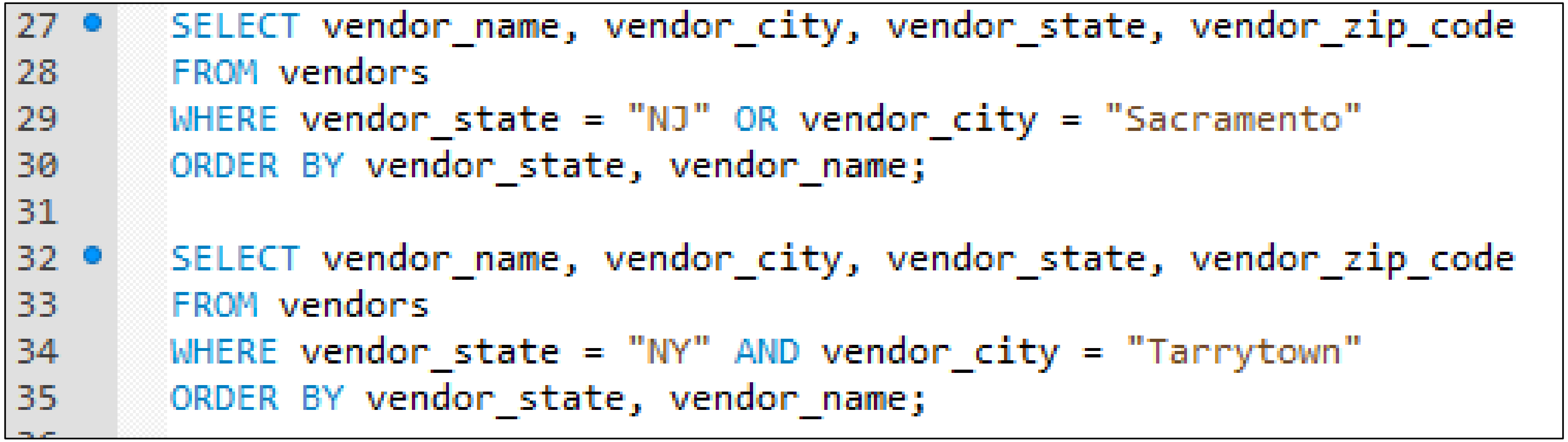
- NOT
- AND
- OR

“Not Equal” Operators



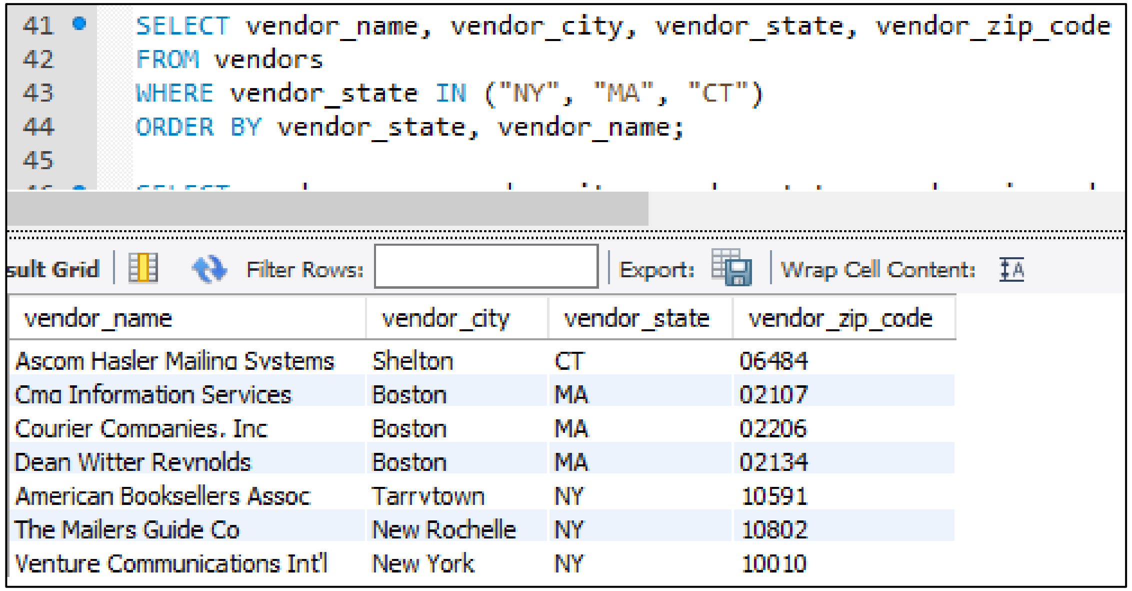
```
SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code
FROM vendors
WHERE vendor_state != "NY"
ORDER BY vendor_state, vendor_name;

SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code
FROM vendors
WHERE vendor_state <> "NY"
ORDER BY vendor_state, vendor_name;

SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code
FROM vendors
WHERE NOT vendor_state = "NY";
```

Multiple Conditions in WHERE Clauses



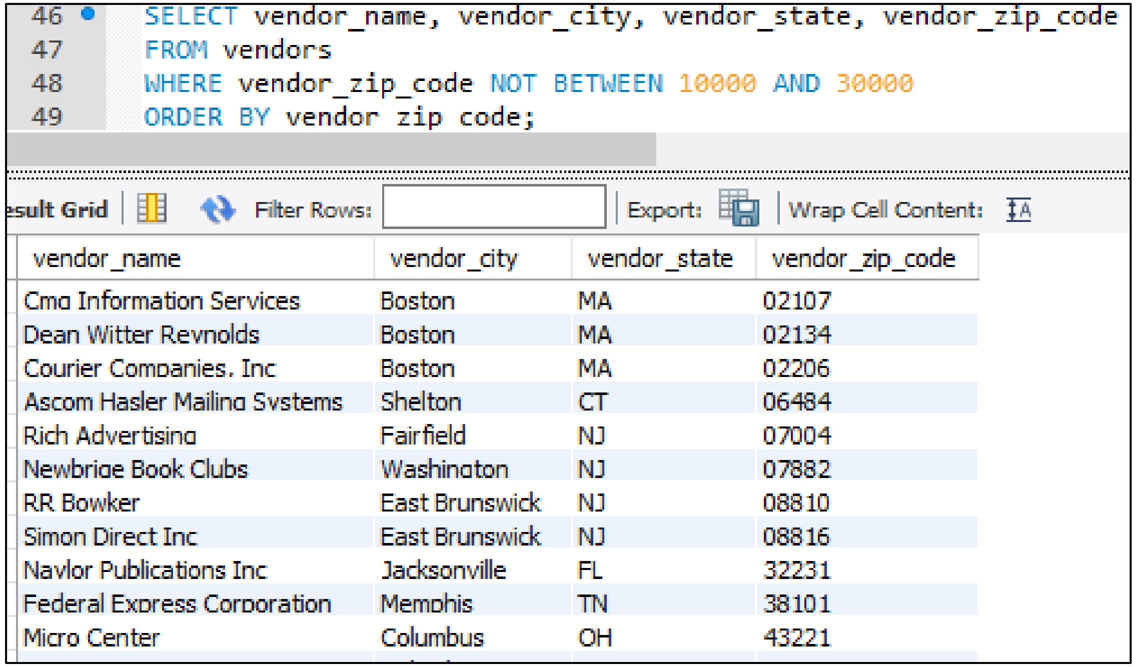
```
SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code
FROM vendors
WHERE vendor_state = "NJ" OR vendor_city = "Sacramento"
ORDER BY vendor_state, vendor_name;

SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code
FROM vendors
WHERE vendor_state = "NY" AND vendor_city = "Tarrytown"
ORDER BY vendor_state, vendor_name;
```

IN and BETWEEN Operators



```sql
SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code
FROM vendors
WHERE vendor_state IN ("NY", "MA", "CT")
ORDER BY vendor_state, vendor_name;
```

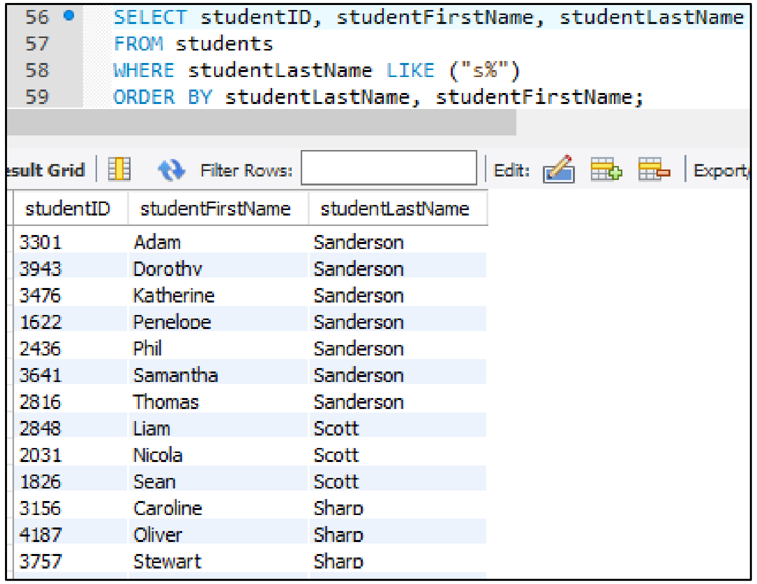
vendor_name	vendor_city	vendor_state	vendor_zip_code
Ascom Hasler Mailing Systems	Shelton	CT	06484
Cmg Information Services	Boston	MA	02107
Courier Companies, Inc	Boston	MA	02206
Dean Witter Reynolds	Boston	MA	02134
American Booksellers Assoc	Tarrytown	NY	10591
The Mailers Guide Co	New Rochelle	NY	10802
Venture Communications Int'l	New York	NY	10010



```sql
SELECT vendor_name, vendor_city, vendor_state, vendor_zip_code
FROM vendors
WHERE vendor_zip_code NOT BETWEEN 10000 AND 30000
ORDER BY vendor zip code;
```

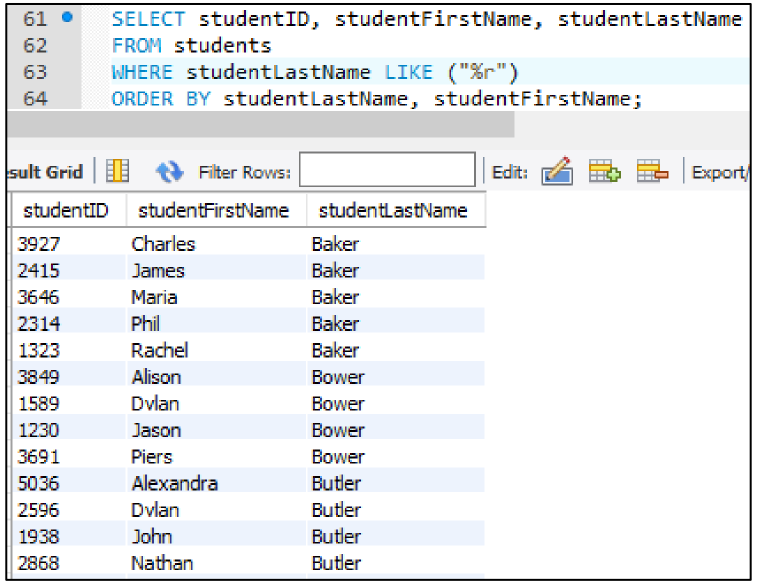
vendor_name	vendor_city	vendor_state	vendor_zip_code
Cmg Information Services	Boston	MA	02107
Dean Witter Reynolds	Boston	MA	02134
Courier Companies, Inc	Boston	MA	02206
Ascom Hasler Mailing Systems	Shelton	CT	06484
Rich Advertising	Fairfield	NJ	07004
Newbrige Book Clubs	Washington	NJ	07882
RR Bowker	East Brunswick	NJ	08810
Simon Direct Inc	East Brunswick	NJ	08816
Naylor Publications Inc	Jacksonville	FL	32231
Federal Express Corporation	Memphis	TN	38101
Micro Center	Columbus	OH	43221

LIKE Operator



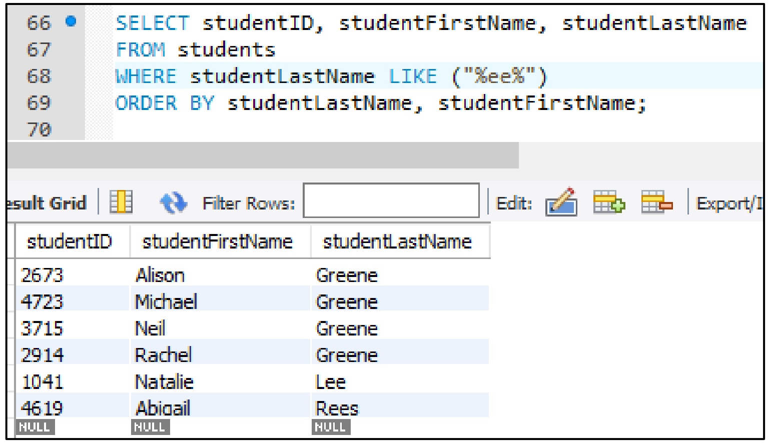
```
SELECT studentID, studentFirstName, studentLastName
FROM students
WHERE studentLastName LIKE ("s%")
ORDER BY studentLastName, studentFirstName;
```

studentID	studentFirstName	studentLastName
3301	Adam	Sanderson
3943	Dorothy	Sanderson
3476	Katherine	Sanderson
1622	Penelope	Sanderson
2436	Phil	Sanderson
3641	Samantha	Sanderson
2816	Thomas	Sanderson
2848	Liam	Scott
2031	Nicola	Scott
1826	Sean	Scott
3156	Caroline	Sharp
4187	Oliver	Sharp
3757	Stewart	Sharp



```
SELECT studentID, studentFirstName, studentLastName
FROM students
WHERE studentLastName LIKE ("%r")
ORDER BY studentLastName, studentFirstName;
```

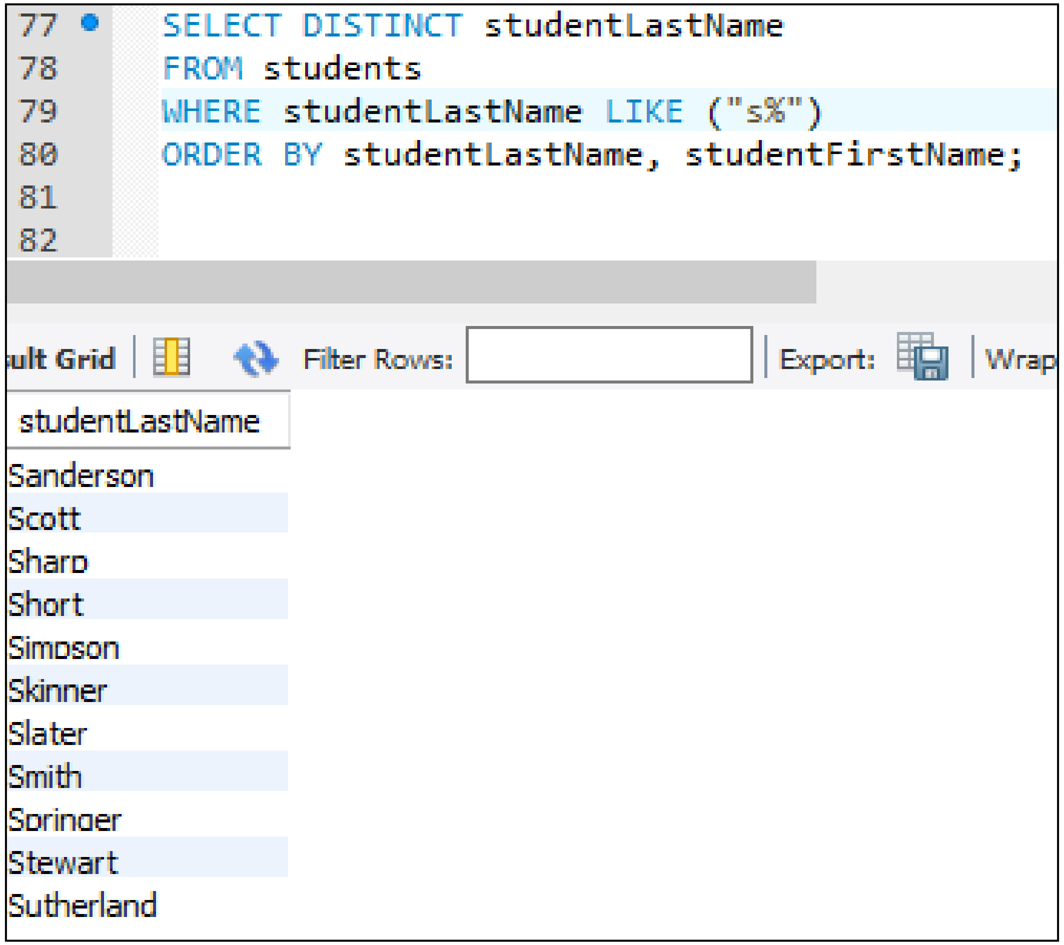
studentID	studentFirstName	studentLastName
3927	Charles	Baker
2415	James	Baker
3646	Maria	Baker
2314	Phil	Baker
1323	Rachel	Baker
3849	Alison	Bower
1589	Dylan	Bower
1230	Jason	Bower
3691	Piers	Bower
5036	Alexandra	Butler
2596	Dylan	Butler
1938	John	Butler
2868	Nathan	Butler



```
SELECT studentID, studentFirstName, studentLastName
FROM students
WHERE studentLastName LIKE ("%ee%")
ORDER BY studentLastName, studentFirstName;
```

studentID	studentFirstName	studentLastName
2673	Alison	Greene
4723	Michael	Greene
3715	Neil	Greene
2914	Rachel	Greene
1041	Natalie	Lee
4619	Abigail	Rees
NULL	NULL	NULL

SELECT DISTINCT

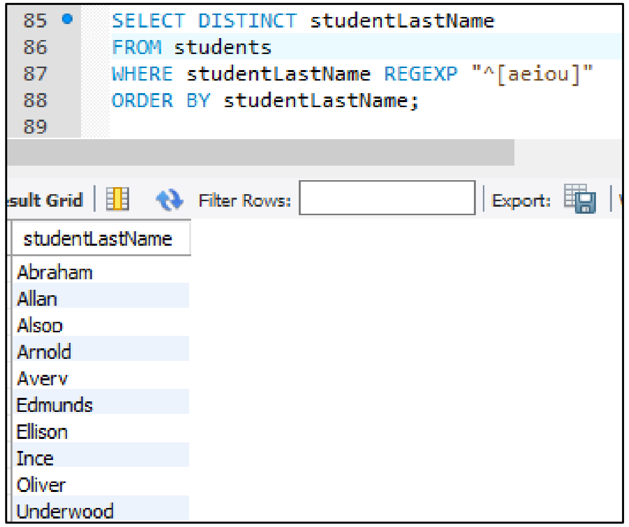


```
SELECT DISTINCT studentLastName
FROM students
WHERE studentLastName LIKE ("s%")
ORDER BY studentLastName, studentFirstName;
```

```
studentLastName
Sanderson
Scott
Sharp
Short
Simpson
Skinner
Slater
Smith
Springer
Stewart
Sutherland
```

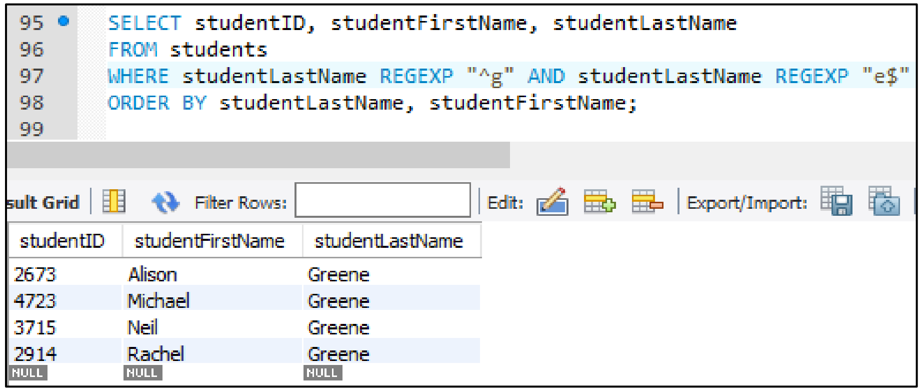
REGEXP Statements

https://www.tutorialspoint.com/mysql/mysql-regexps.htm



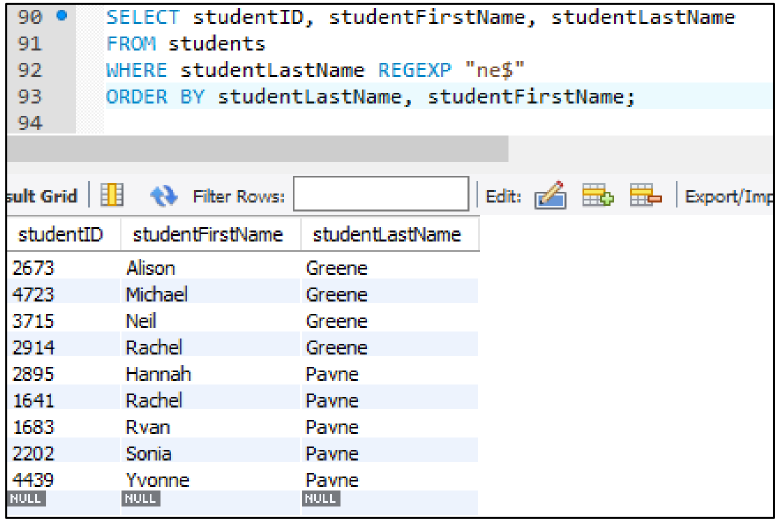
```
SELECT DISTINCT studentLastName
FROM students
WHERE studentLastName REGEXP "^[aeiou]"
ORDER BY studentLastName;
```

studentLastName
Abraham
Allan
Alsop
Arnold
Avery
Edmunds
Ellison
Ince
Oliver
Underwood



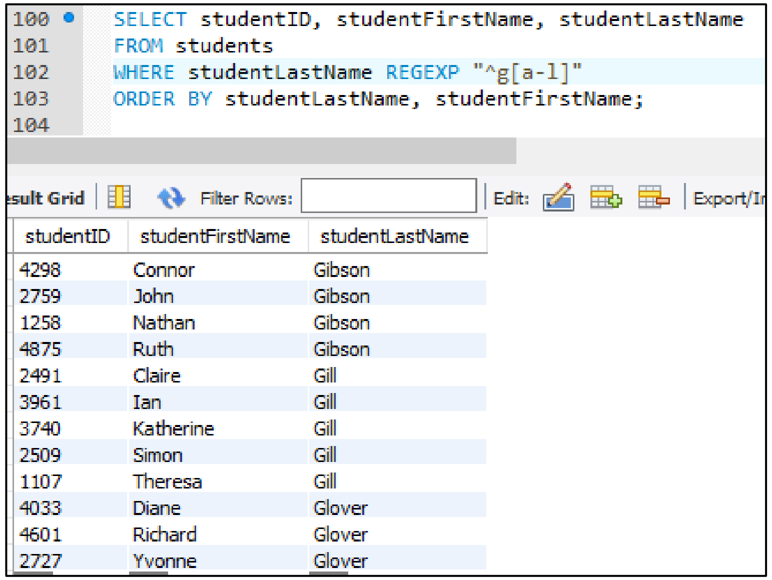
```
SELECT studentID, studentFirstName, studentLastName
FROM students
WHERE studentLastName REGEXP "^g" AND studentLastName REGEXP "e$"
ORDER BY studentLastName, studentFirstName;
```

studentID	studentFirstName	studentLastName
2673	Alison	Greene
4723	Michael	Greene
3715	Neil	Greene
2914	Rachel	Greene
NULL	NULL	NULL



```
SELECT studentID, studentFirstName, studentLastName
FROM students
WHERE studentLastName REGEXP "ne$"
ORDER BY studentLastName, studentFirstName;
```

studentID	studentFirstName	studentLastName
2673	Alison	Greene
4723	Michael	Greene
3715	Neil	Greene
2914	Rachel	Greene
2895	Hannah	Payne
1641	Rachel	Payne
1683	Ryan	Payne
2202	Sonia	Payne
4439	Yvonne	Payne
NULL	NULL	NULL

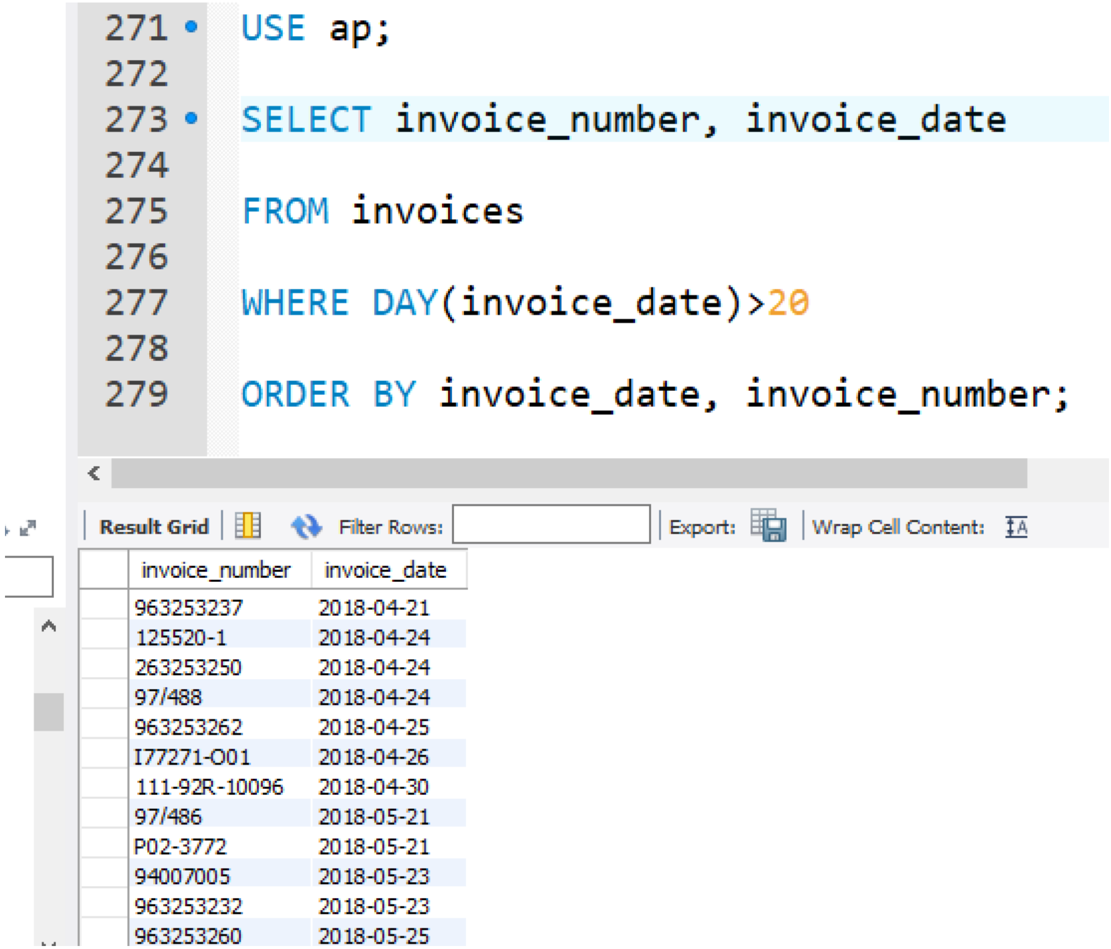


```
SELECT studentID, studentFirstName, studentLastName
FROM students
WHERE studentLastName REGEXP "^g[a-l]"
ORDER BY studentLastName, studentFirstName;
```

studentID	studentFirstName	studentLastName
4298	Connor	Gibson
2759	John	Gibson
1258	Nathan	Gibson
4875	Ruth	Gibson
2491	Claire	Gill
3961	Ian	Gill
3740	Katherine	Gill
2509	Simon	Gill
1107	Theresa	Gill
4033	Diane	Glover
4601	Richard	Glover
2727	Yvonne	Glover

Date Operators

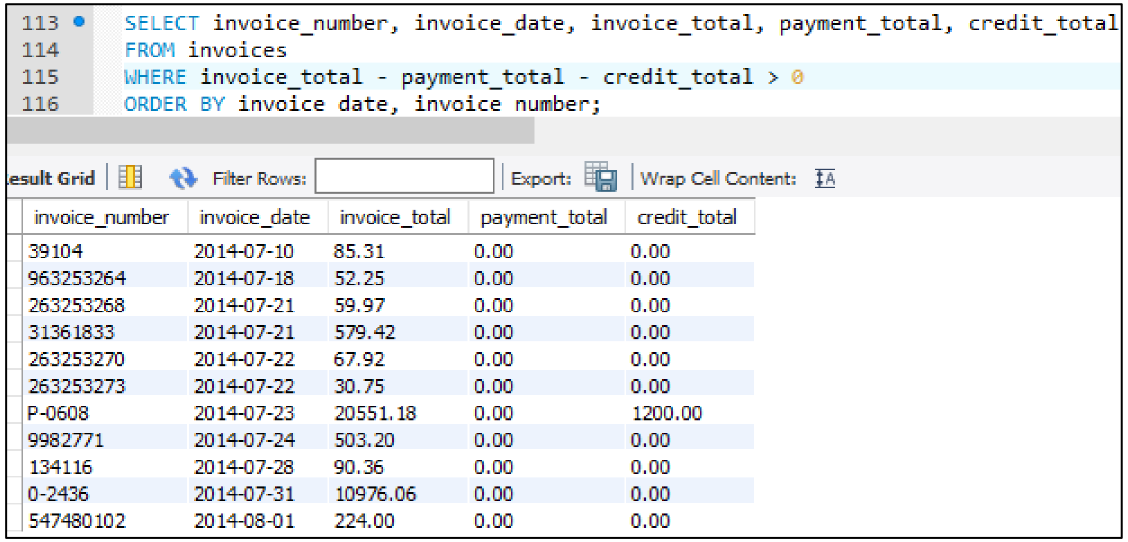
http://www.tutorialspoint.com/mysql/mysql-date-time-functions.htm



```sql
USE ap;

SELECT invoice_number, invoice_date

FROM invoices

WHERE DAY(invoice_date)>20

ORDER BY invoice_date, invoice_number;
```

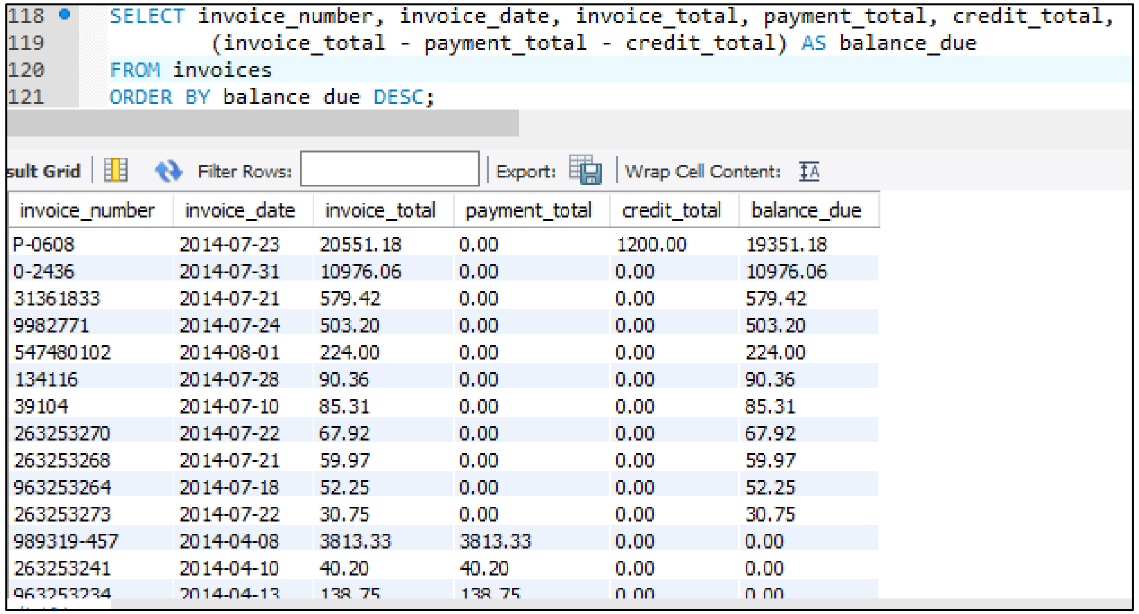
invoice_number	invoice_date
963253237	2018-04-21
125520-1	2018-04-24
263253250	2018-04-24
97/488	2018-04-24
963253262	2018-04-25
I77271-O01	2018-04-26
111-92R-10096	2018-04-30
97/486	2018-05-21
P02-3772	2018-05-21
94007005	2018-05-23
963253232	2018-05-23
963253260	2018-05-25

Calculated Columns (AS Statements)



```
SELECT invoice_number, invoice_date, invoice_total, payment_total, credit_total
FROM invoices
WHERE invoice_total - payment_total - credit_total > 0
ORDER BY invoice date, invoice number;
```

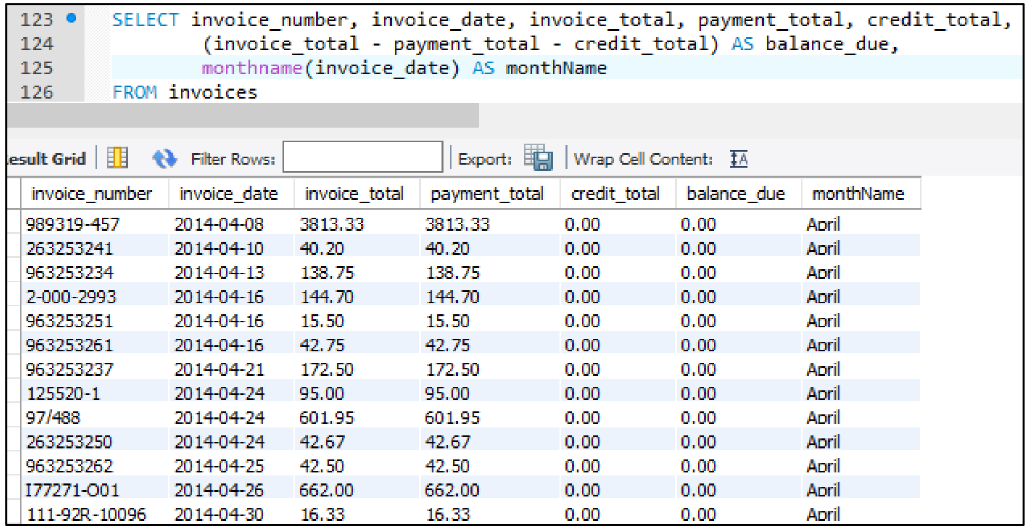
invoice_number	invoice_date	invoice_total	payment_total	credit_total
39104	2014-07-10	85.31	0.00	0.00
963253264	2014-07-18	52.25	0.00	0.00
263253268	2014-07-21	59.97	0.00	0.00
31361833	2014-07-21	579.42	0.00	0.00
263253270	2014-07-22	67.92	0.00	0.00
263253273	2014-07-22	30.75	0.00	0.00
P-0608	2014-07-23	20551.18	0.00	1200.00
9982771	2014-07-24	503.20	0.00	0.00
134116	2014-07-28	90.36	0.00	0.00
0-2436	2014-07-31	10976.06	0.00	0.00
547480102	2014-08-01	224.00	0.00	0.00



```
SELECT invoice_number, invoice_date, invoice_total, payment_total, credit_total,
       (invoice_total - payment_total - credit_total) AS balance_due
FROM invoices
ORDER BY balance due DESC;
```

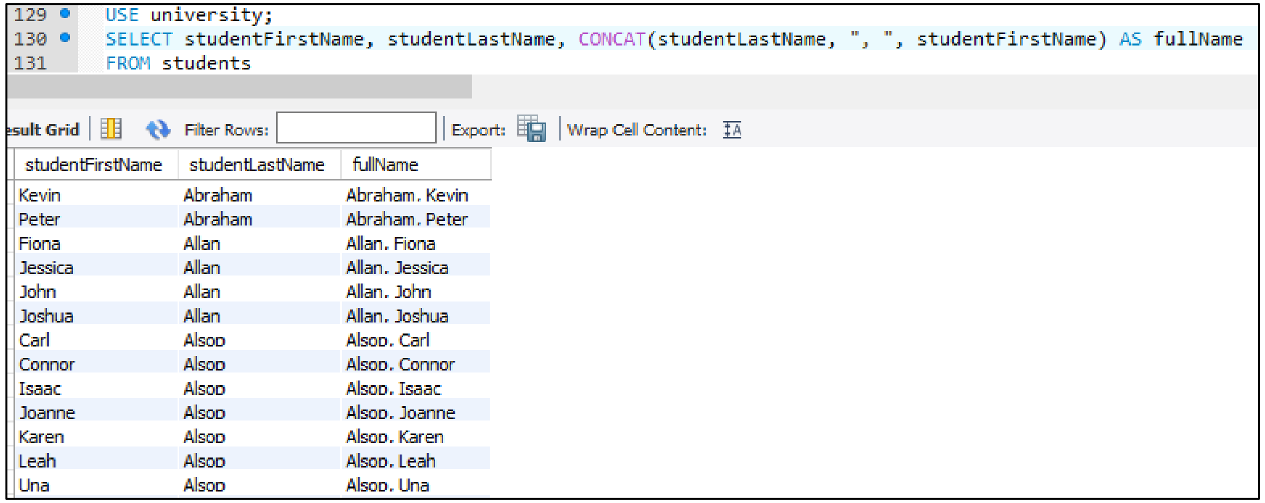
invoice_number	invoice_date	invoice_total	payment_total	credit_total	balance_due
P-0608	2014-07-23	20551.18	0.00	1200.00	19351.18
0-2436	2014-07-31	10976.06	0.00	0.00	10976.06
31361833	2014-07-21	579.42	0.00	0.00	579.42
9982771	2014-07-24	503.20	0.00	0.00	503.20
547480102	2014-08-01	224.00	0.00	0.00	224.00
134116	2014-07-28	90.36	0.00	0.00	90.36
39104	2014-07-10	85.31	0.00	0.00	85.31
263253270	2014-07-22	67.92	0.00	0.00	67.92
263253268	2014-07-21	59.97	0.00	0.00	59.97
963253264	2014-07-18	52.25	0.00	0.00	52.25
263253273	2014-07-22	30.75	0.00	0.00	30.75
989319-457	2014-04-08	3813.33	3813.33	0.00	0.00
263253241	2014-04-10	40.20	40.20	0.00	0.00
963253234	2014-04-13	138.75	138.75	0.00	0.00

Calculated Columns (AS Statements)



```
SELECT invoice_number, invoice_date, invoice_total, payment_total, credit_total,
        (invoice_total - payment_total - credit_total) AS balance_due,
        monthname(invoice_date) AS monthName
FROM invoices
```

invoice_number	invoice_date	invoice_total	payment_total	credit_total	balance_due	monthName
989319-457	2014-04-08	3813.33	3813.33	0.00	0.00	April
263253241	2014-04-10	40.20	40.20	0.00	0.00	April
963253234	2014-04-13	138.75	138.75	0.00	0.00	April
2-000-2993	2014-04-16	144.70	144.70	0.00	0.00	April
963253251	2014-04-16	15.50	15.50	0.00	0.00	April
963253261	2014-04-16	42.75	42.75	0.00	0.00	April
963253237	2014-04-21	172.50	172.50	0.00	0.00	April
125520-1	2014-04-24	95.00	95.00	0.00	0.00	April
97/488	2014-04-24	601.95	601.95	0.00	0.00	April
263253250	2014-04-24	42.67	42.67	0.00	0.00	April
963253262	2014-04-25	42.50	42.50	0.00	0.00	April
I77271-O01	2014-04-26	662.00	662.00	0.00	0.00	April
111-92R-10096	2014-04-30	16.33	16.33	0.00	0.00	April



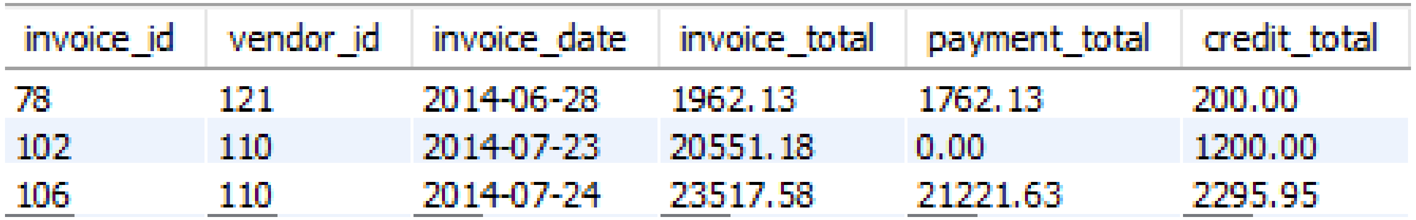
```
USE university;
SELECT studentFirstName, studentLastName, CONCAT(studentLastName, ", ", studentFirstName) AS fullName
FROM students
```

studentFirstName	studentLastName	fullName
Kevin	Abraham	Abraham, Kevin
Peter	Abraham	Abraham, Peter
Fiona	Allan	Allan, Fiona
Jessica	Allan	Allan, Jessica
John	Allan	Allan, John
Joshua	Allan	Allan, Joshua
Carl	Alsop	Alsop, Carl
Connor	Alsop	Alsop, Connor
Isaac	Alsop	Alsop, Isaac
Joanne	Alsop	Alsop, Joanne
Karen	Alsop	Alsop, Karen
Leah	Alsop	Alsop, Leah
Una	Alsop	Alsop, Una

Exercises

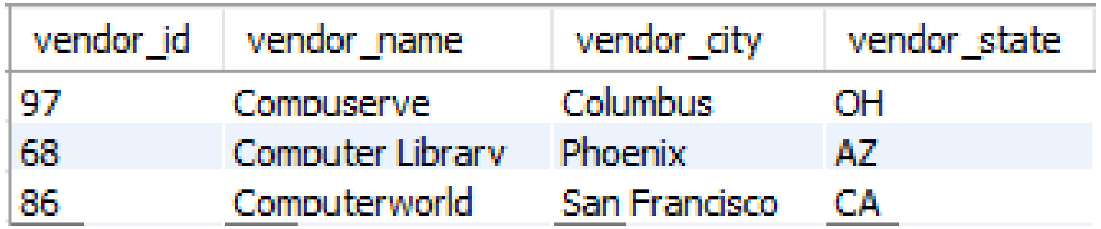
WHERE statements practice

1. USE ap; List of invoices with non-zero credit total.



invoice_id	vendor_id	invoice_date	invoice_total	payment_total	credit_total
78	121	2014-06-28	1962.13	1762.13	200.00
102	110	2014-07-23	20551.18	0.00	1200.00
106	110	2014-07-24	23517.58	21221.63	2295.95

2. USE ap; List of vendors with “compu” in their name.

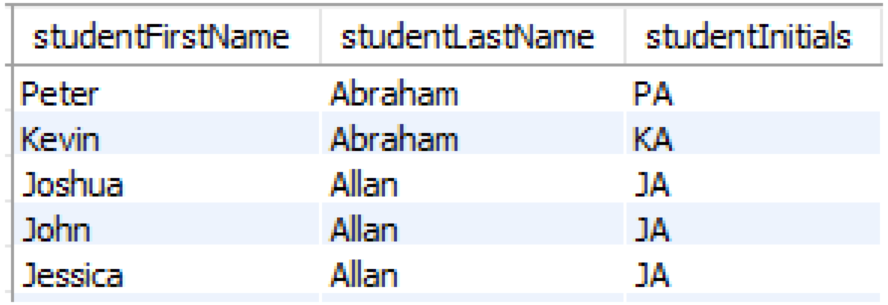


vendor_id	vendor_name	vendor_city	vendor_state
97	Compuserve	Columbus	OH
68	Computer Library	Phoenix	AZ
86	Computerworld	San Francisco	CA

Exercises

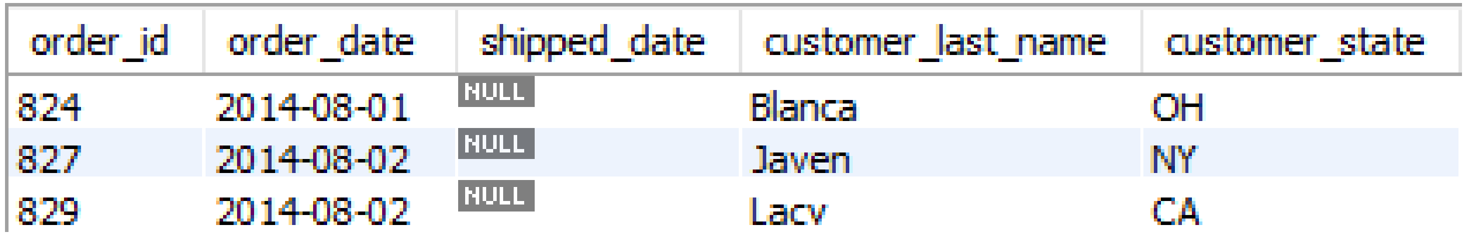
WHERE statements practice

3. USE university; List of students with student initials column



studentFirstName	studentLastName	studentInitials
Peter	Abraham	PA
Kevin	Abraham	KA
Joshua	Allan	JA
John	Allan	JA
Jessica	Allan	JA

4. USE om; List of orders with customer info that have not yet shipped

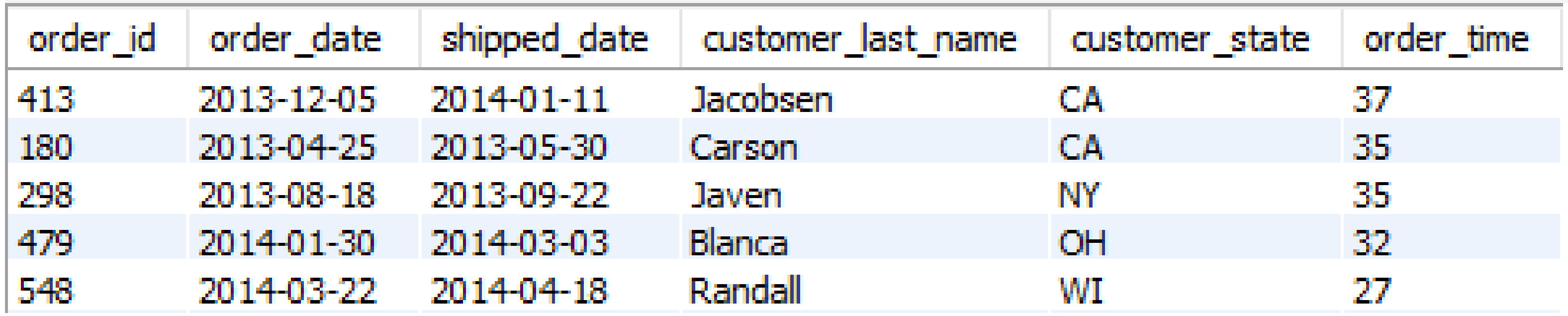


order_id	order_date	shipped_date	customer_last_name	customer_state
824	2014-08-01	NULL	Blanca	OH
827	2014-08-02	NULL	Javen	NY
829	2014-08-02	NULL	Lacv	CA

Exercises

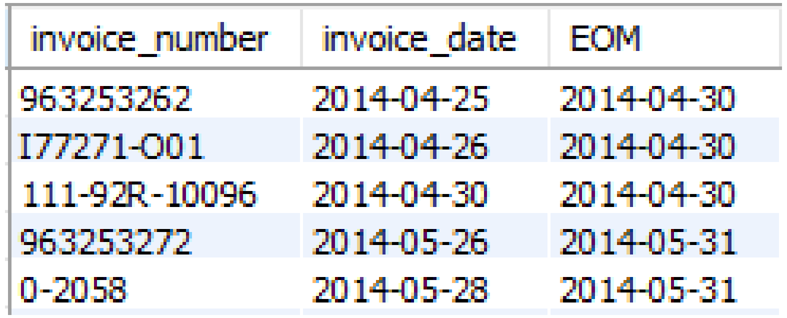
WHERE statements practice

5. USE om; List of orders including days between order and ship date



order_id	order_date	shipped_date	customer_last_name	customer_state	order_time
413	2013-12-05	2014-01-11	Jacobsen	CA	37
180	2013-04-25	2013-05-30	Carson	CA	35
298	2013-08-18	2013-09-22	Javen	NY	35
479	2014-01-30	2014-03-03	Blanca	OH	32
548	2014-03-22	2014-04-18	Randall	WI	27

6. USE ap; List of invoices created in last 5 days of the month

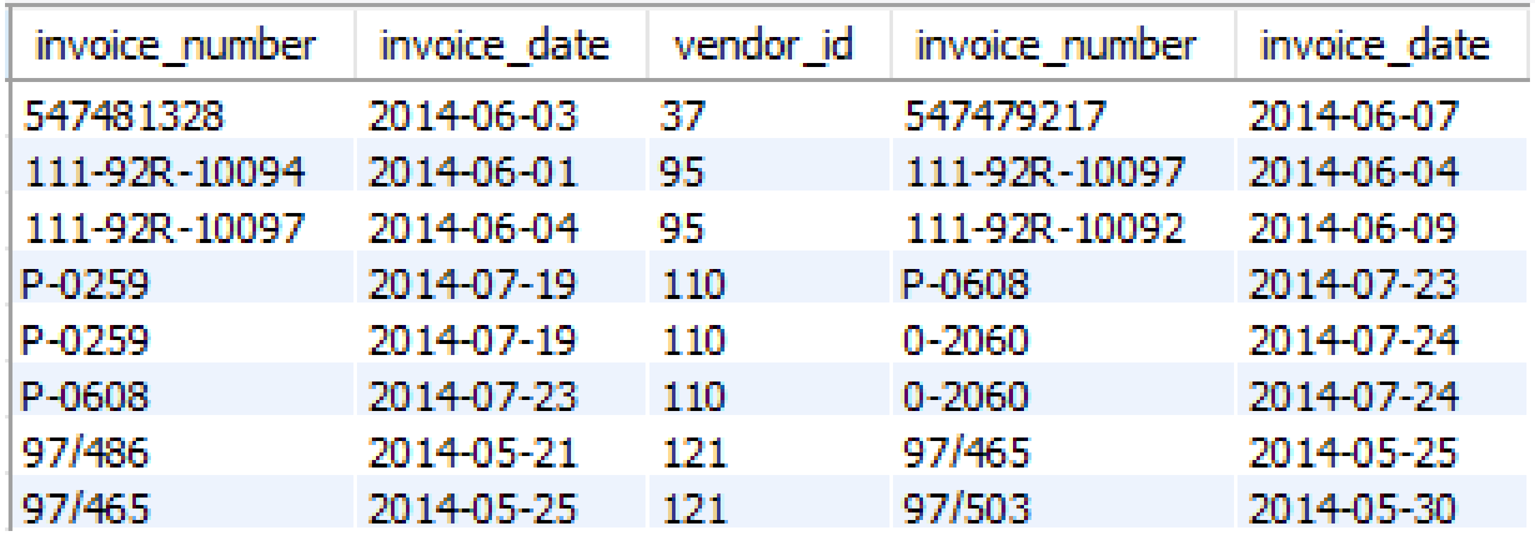


invoice_number	invoice_date	EOM
963253262	2014-04-25	2014-04-30
I77271-O01	2014-04-26	2014-04-30
111-92R-10096	2014-04-30	2014-04-30
963253272	2014-05-26	2014-05-31
0-2058	2014-05-28	2014-05-31

Exercises

WHERE statements practice

7. USE ap; List of pairs of invoices for same vendor within 5 days of each other. (In other words, all invoices which had another invoice created within 5 days.)



invoice_number	invoice_date	vendor_id	invoice_number	invoice_date
547481328	2014-06-03	37	547479217	2014-06-07
111-92R-10094	2014-06-01	95	111-92R-10097	2014-06-04
111-92R-10097	2014-06-04	95	111-92R-10092	2014-06-09
P-0259	2014-07-19	110	P-0608	2014-07-23
P-0259	2014-07-19	110	0-2060	2014-07-24
P-0608	2014-07-23	110	0-2060	2014-07-24
97/486	2014-05-21	121	97/465	2014-05-25
97/465	2014-05-25	121	97/503	2014-05-30