Comp1168 Lab5 – MySQL Built-in Functions

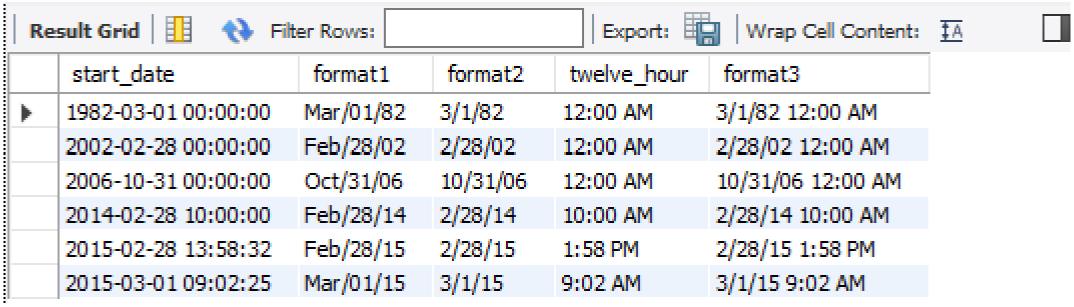
1. Write a SELECT statement that returns these columns from the Date_Sample table in the EX database:

The start_date column

A column that uses the DATE_FORMAT function to return the start_date column with its month name abbreviated and its month, day, and two-digit year separated by slashes

A column that uses the DATE_FORMAT function to return the start_date column with its month and day returned as integers with no leading zeros, a two-digit year, and all date parts separated by slashes

A column that uses the DATE_FORMAT function to return the start_date column with only the hours and minutes on a 12-hour clock with an am/pm indicator



start_date	format1	format2	twelve_hour	format3
1982-03-01 00:00:00	Mar/01/82	3/1/82	12:00 AM	3/1/82 12:00 AM
2002-02-28 00:00:00	Feb/28/02	2/28/02	12:00 AM	2/28/02 12:00 AM
2006-10-31 00:00:00	Oct/31/06	10/31/06	12:00 AM	10/31/06 12:00 AM
2014-02-28 10:00:00	Feb/28/14	2/28/14	10:00 AM	2/28/14 10:00 AM
2015-02-28 13:58:32	Feb/28/15	2/28/15	1:58 PM	2/28/15 1:58 PM
2015-03-01 09:02:25	Mar/01/15	3/1/15	9:02 AM	3/1/15 9:02 AM

2. Write a SELECT statement that returns these columns from the Invoices table:

The invoice_number column

The invoice_date column

The invoice_date column plus 30 days

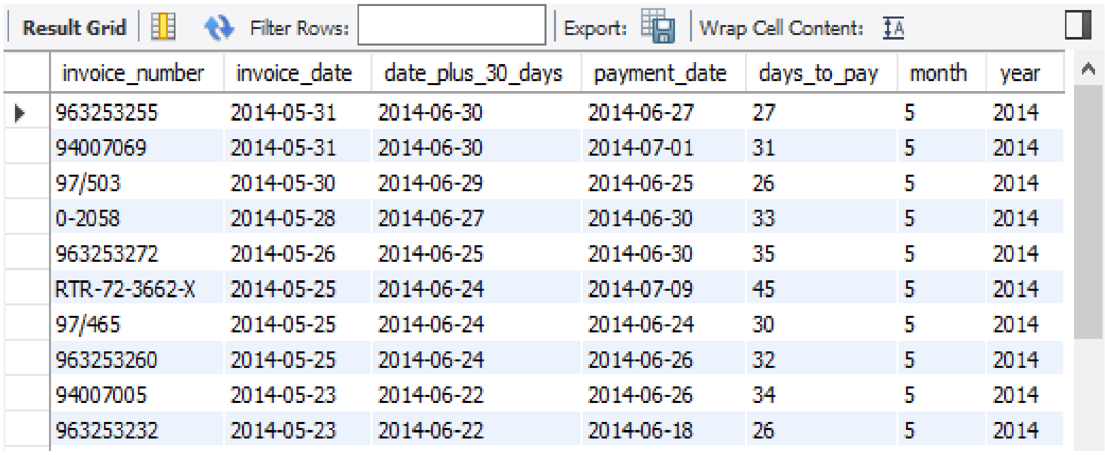
The payment_date column

A column named days_to_pay that shows the number of days between the invoice date and the payment date

The number of the invoice date's month

The four-digit year of the invoice date

When you have this working, add a WHERE clause that retrieves just the invoices for the month of May based on the invoice date, not the number of the invoice month.



invoice_number	invoice_date	date_plus_30_days	payment_date	days_to_pay	month	year
963253255	2014-05-31	2014-06-30	2014-06-27	27	5	2014
94007069	2014-05-31	2014-06-30	2014-07-01	31	5	2014
97/503	2014-05-30	2014-06-29	2014-06-25	26	5	2014
0-2058	2014-05-28	2014-06-27	2014-06-30	33	5	2014
963253272	2014-05-26	2014-06-25	2014-06-30	35	5	2014
RTR-72-3662-X	2014-05-25	2014-06-24	2014-07-09	45	5	2014
97/465	2014-05-25	2014-06-24	2014-06-24	30	5	2014
963253260	2014-05-25	2014-06-24	2014-06-26	32	5	2014
94007005	2014-05-23	2014-06-22	2014-06-26	34	5	2014
963253232	2014-05-23	2014-06-22	2014-06-18	26	5	2014

3. Write a SELECT statement that returns these columns from the String_Sample table of the EX database:

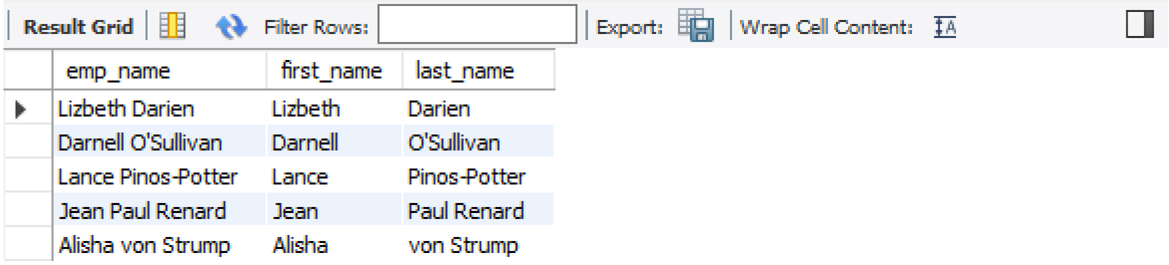
The emp_name column

A column that displays each employee's first name

A column that displays each employee's last name

Use regular expression functions to get the first and last name. If a name contains three parts, everything after the first part should be considered part of the last name. Be sure to provide for last names with hypens and apostrophes.

Hint: To include an apostrophe in a pattern, you can code a \ in front of it or you can enclose the pattern in double quotes.



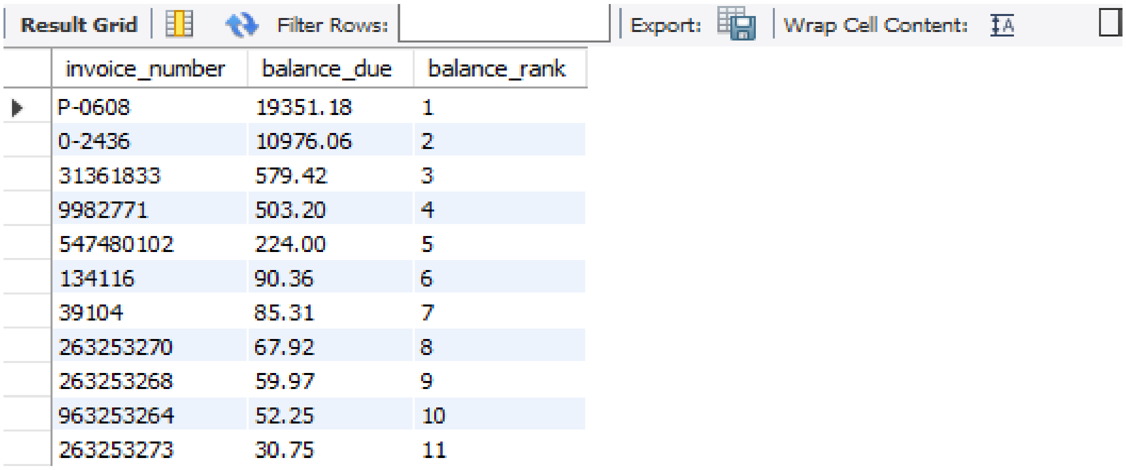
emp_name	first_name	last_name
Lizbeth Darien	Lizbeth	Darien
Darnell O'Sullivan	Darnell	O'Sullivan
Lance Pinos-Potter	Lance	Pinos-Potter
Jean Paul Renard	Jean	Paul Renard
Alisha von Strump	Alisha	von Strump

4. Write a SELECT statement that returns these columns from the Invoice table of the AP database:

The invoice_number column

The balance due for each invoice with a balance due greater than zero

A column that uses the RANK() function to rank the balance due in descending sequence



invoice_number	balance_due	balance_rank
P-0608	19351.18	1
0-2436	10976.06	2
31361833	579.42	3
9982771	503.20	4
547480102	224.00	5
134116	90.36	6
39104	85.31	7
263253270	67.92	8
263253268	59.97	9
963253264	52.25	10
263253273	30.75	11