DATABASE SYSTEMS IMPLEMENTATION

CA PART 2 Toy Store Query Design

APRIL 18, 2016

K00203642

Joe O’Regan

Question 1

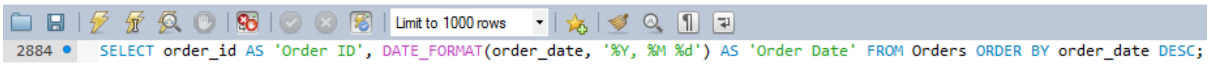
List all order numbers, with the date the order was placed, where date is formatted like the following:-

Order ID	Order Date
4	2007, June 17
….	….

for all orders, sort latest order first.

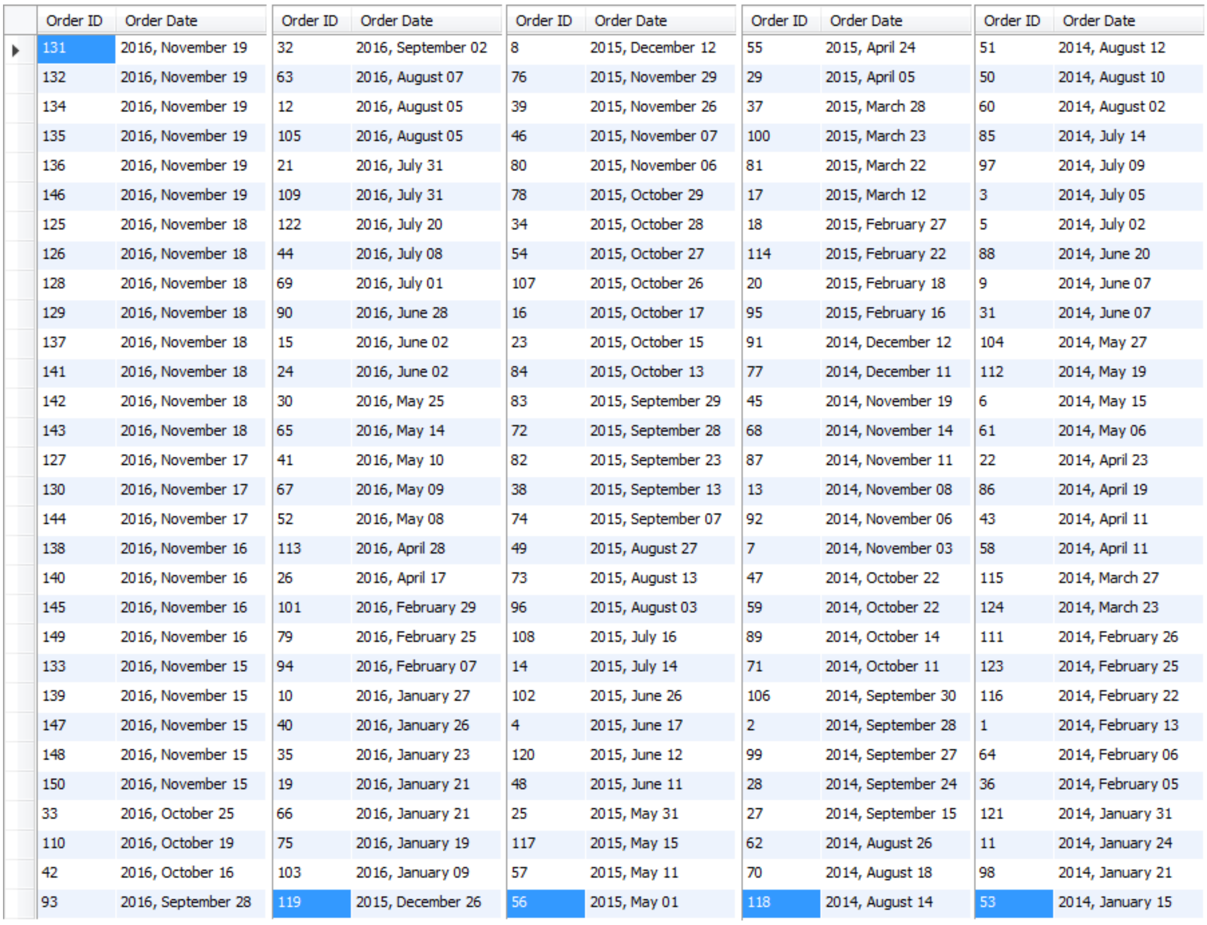
SELECT order_id AS 'Order ID', DATE_FORMAT(order_date, '%Y, %M %d') AS 'Order Date' FROM Orders ORDER BY order_date DESC;

(use aliases to rename the columns, DATE_FORMAT() to format the the date)



```
SELECT order_id AS 'Order ID', DATE_FORMAT(order_date, '%Y, %M %d') AS 'Order Date' FROM Orders ORDER BY order_date DESC;
```

Figure 1 Question 1 Select Statement



Order ID	Order Date	Order ID	Order Date	Order ID	Order Date	Order ID	Order Date	Order ID	Order Date
131	2016, November 19	32	2016, September 02	8	2015, December 12	55	2015, April 24	51	2014, August 12
132	2016, November 19	63	2016, August 07	76	2015, November 29	29	2015, April 05	50	2014, August 10
134	2016, November 19	12	2016, August 05	39	2015, November 26	37	2015, March 28	60	2014, August 02
135	2016, November 19	105	2016, August 05	46	2015, November 07	100	2015, March 23	85	2014, July 14
136	2016, November 19	21	2016, July 31	80	2015, November 06	81	2015, March 22	97	2014, July 09
146	2016, November 19	109	2016, July 31	78	2015, October 29	17	2015, March 12	3	2014, July 05
125	2016, November 18	122	2016, July 20	34	2015, October 28	18	2015, February 27	5	2014, July 02
126	2016, November 18	44	2016, July 08	54	2015, October 27	114	2015, February 22	88	2014, June 20
128	2016, November 18	69	2016, July 01	107	2015, October 26	20	2015, February 18	9	2014, June 07
129	2016, November 18	90	2016, June 28	16	2015, October 17	95	2015, February 16	31	2014, June 07
137	2016, November 18	15	2016, June 02	23	2015, October 15	91	2014, December 12	104	2014, May 27
141	2016, November 18	24	2016, June 02	84	2015, October 13	77	2014, December 11	112	2014, May 19
142	2016, November 18	30	2016, May 25	83	2015, September 29	45	2014, November 19	6	2014, May 15
143	2016, November 18	65	2016, May 14	72	2015, September 28	68	2014, November 14	61	2014, May 06
127	2016, November 17	41	2016, May 10	82	2015, September 23	87	2014, November 11	22	2014, April 23
130	2016, November 17	67	2016, May 09	38	2015, September 13	13	2014, November 08	86	2014, April 19
144	2016, November 17	52	2016, May 08	74	2015, September 07	92	2014, November 06	43	2014, April 11
138	2016, November 16	113	2016, April 28	49	2015, August 27	7	2014, November 03	58	2014, April 11
140	2016, November 16	26	2016, April 17	73	2015, August 13	47	2014, October 22	115	2014, March 27
145	2016, November 16	101	2016, February 29	96	2015, August 03	59	2014, October 22	124	2014, March 23
149	2016, November 16	79	2016, February 25	108	2015, July 16	89	2014, October 14	111	2014, February 26
133	2016, November 15	94	2016, February 07	14	2015, July 14	71	2014, October 11	123	2014, February 25
139	2016, November 15	10	2016, January 27	102	2015, June 26	106	2014, September 30	116	2014, February 22
147	2016, November 15	40	2016, January 26	4	2015, June 17	2	2014, September 28	1	2014, February 13
148	2016, November 15	35	2016, January 23	120	2015, June 12	99	2014, September 27	64	2014, February 06
150	2016, November 15	19	2016, January 21	48	2015, June 11	28	2014, September 24	36	2014, February 05
33	2016, October 25	66	2016, January 21	25	2015, May 31	27	2014, September 15	121	2014, January 31
110	2016, October 19	75	2016, January 19	117	2015, May 15	62	2014, August 26	11	2014, January 24
42	2016, October 16	103	2016, January 09	57	2015, May 11	70	2014, August 18	98	2014, January 21
93	2016, September 28	119	2015, December 26	56	2015, May 01	118	2014, August 14	53	2014, January 15

Figure 2 Question 1: Output for select statement (ordered left to right)

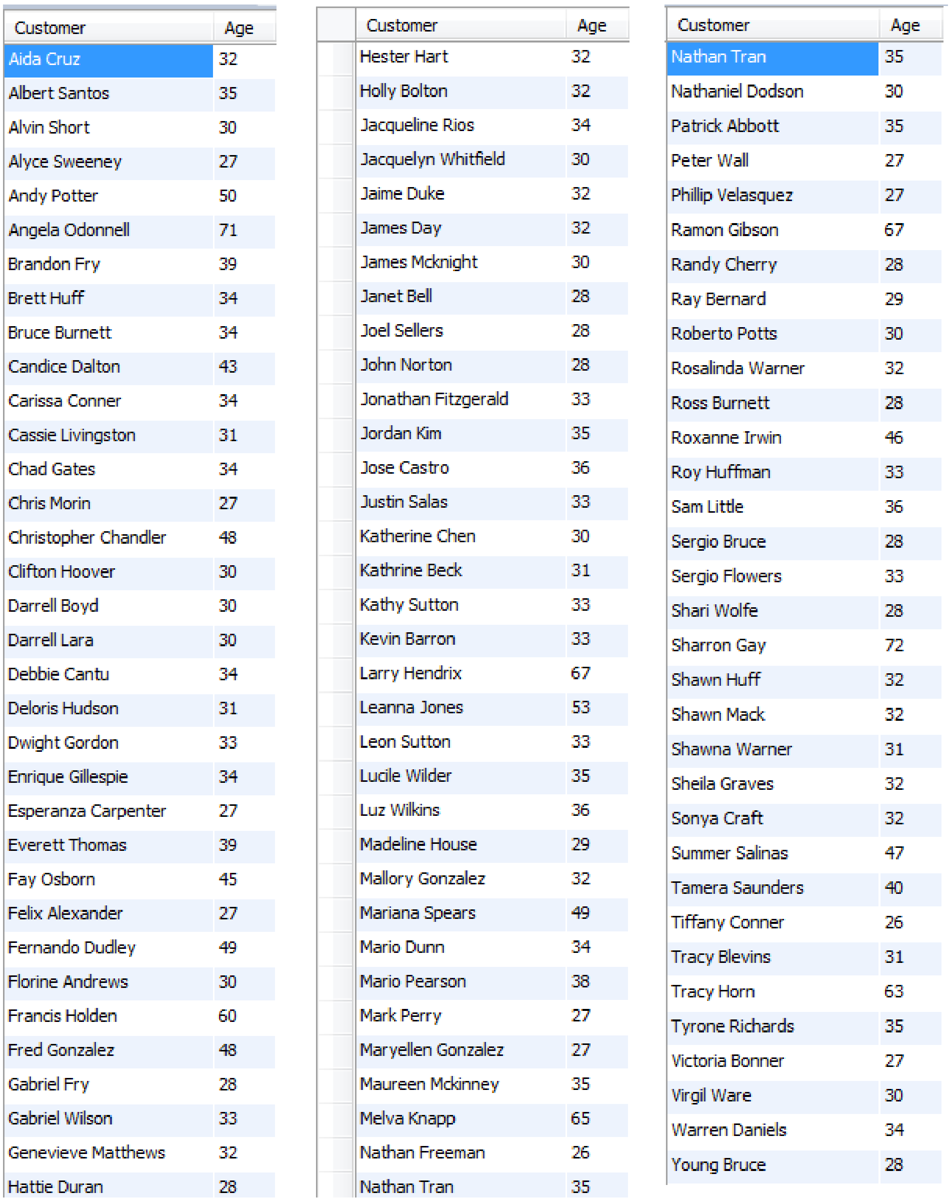
Question 2

List all account holders (account.last_name and account.firstname) with their age in years, where the customers name is concatentated as a string. Sort in alphabetical order E.g.

Customer	Age
Dwight Gordon	34
….	….

SELECT CONCAT(first_name, ' ', last_name) AS 'Customer', TIMESTAMPDIFF(YEAR, birthday, CURDATE()) AS Age FROM account ORDER BY Customer;

I googled the TIMESTAMPDIFF() command, subtracting dates didn't work too good if someone's birthday had yet to happen this year.



Customer	Age
Aida Cruz	32
Albert Santos	35
Alvin Short	30
Alyce Sweeney	27
Andy Potter	50
Angela Odonnell	71
Brandon Fry	39
Brett Huff	34
Bruce Burnett	34
Candice Dalton	43
Carissa Conner	34
Cassie Livingston	31
Chad Gates	34
Chris Morin	27
Christopher Chandler	48
Clifton Hoover	30
Darrell Boyd	30
Darrell Lara	30
Debbie Cantu	34
Deloris Hudson	31
Dwight Gordon	33
Enrique Gillespie	34
Esperanza Carpenter	27
Everett Thomas	39
Fay Osborn	45
Felix Alexander	27
Fernando Dudley	49
Florine Andrews	30
Francis Holden	60
Fred Gonzalez	48
Gabriel Fry	28
Gabriel Wilson	33
Genevieve Matthews	32
Hattie Duran	28
Hester Hart	32
Holly Bolton	32
Jacqueline Rios	34
Jacquelyn Whitfield	30
Jaime Duke	32
James Day	32
James Mcknight	30
Janet Bell	28
Joel Sellers	28
John Norton	28
Jonathan Fitzgerald	33
Jordan Kim	35
Jose Castro	36
Justin Salas	33
Katherine Chen	30
Kathrine Beck	31
Kathy Sutton	33
Kevin Barron	33
Larry Hendrix	67
Leanna Jones	53
Leon Sutton	33
Lucile Wilder	35
Luz Wilkins	36
Madeline House	29
Mallory Gonzalez	32
Mariana Spears	49
Mario Dunn	34
Mario Pearson	38
Mark Perry	27
Maryellen Gonzalez	27
Maureen Mckinney	35
Melva Knapp	65
Nathan Freeman	26
Nathan Tran	35
Nathan Tran	35
Nathaniel Dodson	30
Patrick Abbott	35
Peter Wall	27
Phillip Velasquez	27
Ramon Gibson	67
Randy Cherry	28
Ray Bernard	29
Roberto Potts	30
Rosalinda Warner	32
Ross Burnett	28
Roxanne Irwin	46
Roy Huffman	33
Sam Little	36
Sergio Bruce	28
Sergio Flowers	33
Shari Wolfe	28
Sharron Gay	72
Shawn Huff	32
Shawn Mack	32
Shawna Warner	31
Sheila Graves	32
Sonya Craft	32
Summer Salinas	47
Tamera Saunders	40
Tiffany Conner	26
Tracy Blevins	31
Tracy Horn	63
Tyrone Richards	35
Victoria Bonner	27
Virgil Ware	30
Warren Daniels	34
Young Bruce	28

Figure 3 Question 2 Output

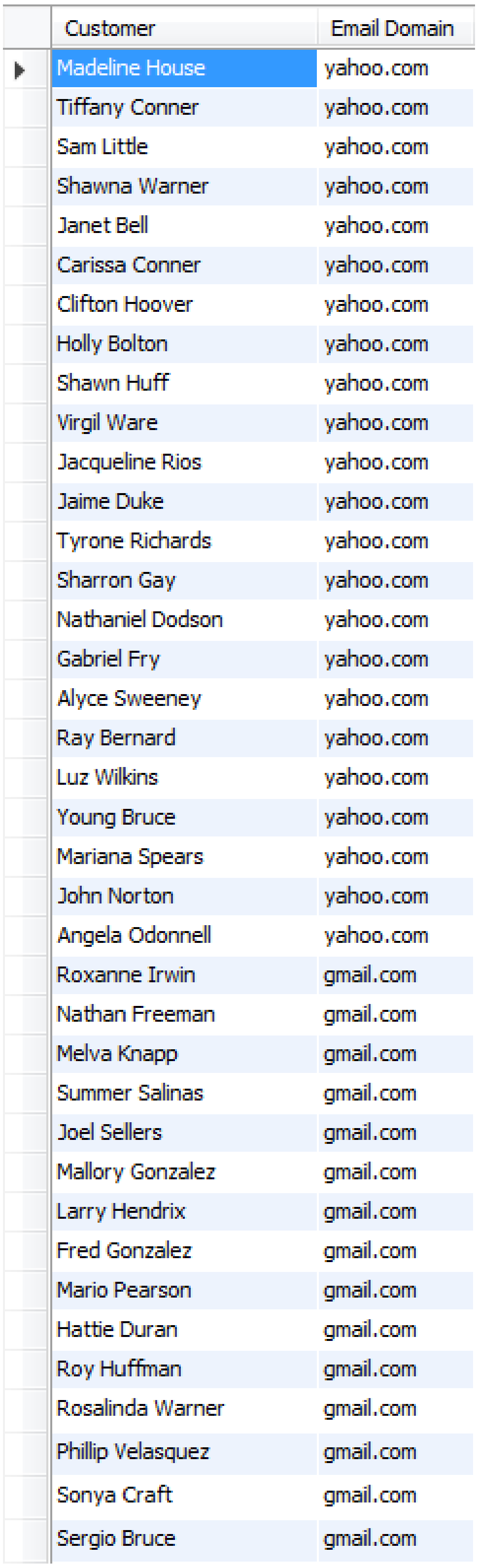
Question 3

List all account holders who are using either a yahoo.com or a gmail.com email account. Where all yahoo.com accounts are listed first, followed by all gmail.com accounts, e.g.

Customer	Email Domain
Madeline House	yahoo.com
….	….

```
SELECT CONCAT(first_name, ' ',  last_name)
AS Customer,
RIGHT(email,9) AS 'Email Domain'
FROM account
WHERE EMAIL LIKE ('%yahoo.com')
OR email LIKE ('%gmail.com')
ORDER BY right(email,9) DESC;
```

Using an alias for email with an underscore (AS Email_Domain … ORDER BY Email_Domain) worked too for order by, but not two words in quotes like the alias above for ordering.



Customer	Email Domain
Madeline House	yahoo.com
Tiffany Conner	yahoo.com
Sam Little	yahoo.com
Shawna Warner	yahoo.com
Janet Bell	yahoo.com
Carissa Conner	yahoo.com
Clifton Hoover	yahoo.com
Holly Bolton	yahoo.com
Shawn Huff	yahoo.com
Virgil Ware	yahoo.com
Jacqueline Rios	yahoo.com
Jaime Duke	yahoo.com
Tyrone Richards	yahoo.com
Sharron Gay	yahoo.com
Nathaniel Dodson	yahoo.com
Gabriel Fry	yahoo.com
Alyce Sweeney	yahoo.com
Ray Bernard	yahoo.com
Luz Wilkins	yahoo.com
Young Bruce	yahoo.com
Mariana Spears	yahoo.com
John Norton	yahoo.com
Angela Odonnell	yahoo.com
Roxanne Irwin	gmail.com
Nathan Freeman	gmail.com
Melva Knapp	gmail.com
Summer Salinas	gmail.com
Joel Sellers	gmail.com
Mallory Gonzalez	gmail.com
Larry Hendrix	gmail.com
Fred Gonzalez	gmail.com
Mario Pearson	gmail.com
Hattie Duran	gmail.com
Roy Huffman	gmail.com
Rosalinda Warner	gmail.com
Phillip Velasquez	gmail.com
Sonya Craft	gmail.com
Sergio Bruce	gmail.com

Figure 4 Question 3 Output

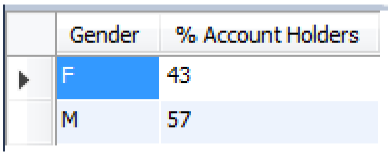
Question 4 (10 Marks)

Calculate the percentage split of female and male account holders, e.g.

Gender	% Account Holders
G	52
M	48

SELECT gender AS Gender,
FORMAT(COUNT(*) / (SELECT COUNT(*) FROM account) * 100,0)
AS '% Account Holders'
FROM ACCOUNT WHERE gender = 'M' OR gender = 'F' GROUP BY GENDER;

There was no decimal places in the e.g. so I used FORMAT(N,D) to format the output to have no decimal places, could also use ROUND()



Gender	% Account Holders
F	43
M	57

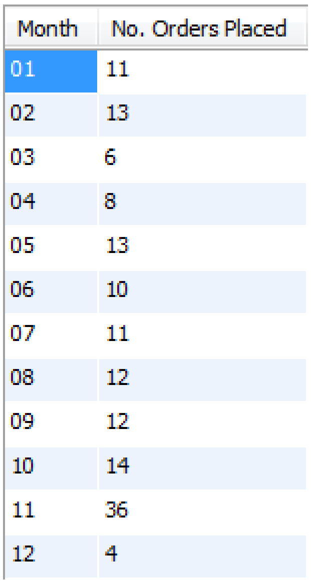
Figure 5 Question 4 Output

Question 5

Calculate how many orders are placed in each month of the year. You do not need to differentiate by year, i.e. count all the orders placed in January, all placed in February etc e.g.

Month	No. Orders Placed
1	67
2	31
3	45
4	66

SELECT DATE_FORMAT(order_date, '%m') AS 'Month',
COUNT(*) AS 'No. Orders Placed'
FROM orders GROUP BY DATE_FORMAT(order_date, '%m')
ORDER BY DATE_FORMAT(order_date, '%m');



Month	No. Orders Placed
01	11
02	13
03	6
04	8
05	13
06	10
07	11
08	12
09	12
10	14
11	36
12	4

Figure 6 Question 5 Output

Grouping by DATE_FORMAT(order_date,'%m') gives the count for each month

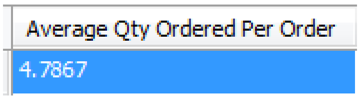
Question 6

Calculate the average number of items placed across all orders, .e.g

Average Qty Ordered Per Order
12

SELECT (SELECT SUM(quantity) FROM lineitem) /
(SELECT COUNT(DISTINCT order_id) FROM lineitem)
AS 'Average Qty Ordered Per Order';

Total number of items ordered (sum of the quantity field) 718, divided by the total (unique) number of orders 150



Average Qty Ordered Per Order
4.7867

Figure 7 Question 6 Output

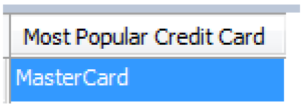
Question 7

List the most popular credit card to pay an order with. E.g

Most Popular Credit Card
Visa

SELECT cc_type AS 'Most Popular Credit Card' FROM orders
GROUP BY cc_type ORDER BY COUNT(cc_type) DESC LIMIT 1;

Grouping by the cc_type, to get the count for each cc_type, and ordering by the count for each cc_type, then limiting the answer to the highest 1



Most Popular Credit Card
MasterCard

Figure 8 Question 7 Output

Question 8

List the name of account holders, the order number, order date, and productId for all orders for any products in the Fate/Stay Night series.

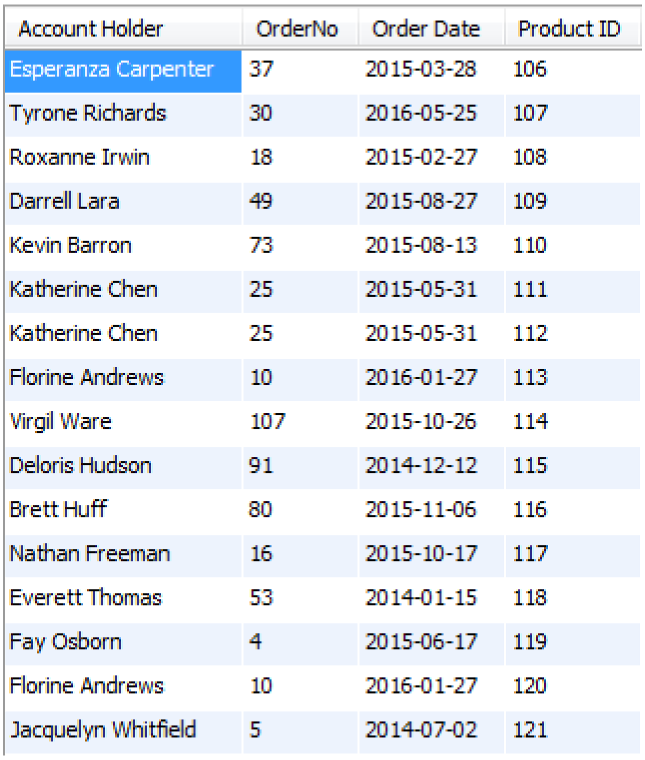
Account Holder	OrderNo	Order Date	ProductID
Lara Daryl	147	2008-11-15	115
….	….	….	….

```
SELECT CONCAT(first_name, " ", last_name) AS 'Account Holder',
o.order_id AS OrderNo, order_date AS 'Order Date', p.product_id AS 'Product ID'
FROM account a
JOIN orders o ON a.user_id = o.user_id
JOIN lineitem l ON l.order_id = o.order_id
JOIN item i ON i.item_id = l.item_id
JOIN product p ON p.product_id = i.product_id
WHERE series LIKE 'Fate/Stay Night' GROUP BY p.product_id;
```

The first and last name are concatenated as one column from account joining orders on user_id

Lineitem joins orders on order_id to get item_id. Item_id then gets the product_id from item. To get the series from product with the product_id.

Order_id, order_no, and order_date are from orders, and then grouped by product_id and compared to series from the product table



Account Holder	OrderNo	Order Date	Product ID
Esperanza Carpenter	37	2015-03-28	106
Tyrone Richards	30	2016-05-25	107
Roxanne Irwin	18	2015-02-27	108
Darrell Lara	49	2015-08-27	109
Kevin Barron	73	2015-08-13	110
Katherine Chen	25	2015-05-31	111
Katherine Chen	25	2015-05-31	112
Florine Andrews	10	2016-01-27	113
Virgil Ware	107	2015-10-26	114
Deloris Hudson	91	2014-12-12	115
Brett Huff	80	2015-11-06	116
Nathan Freeman	16	2015-10-17	117
Everett Thomas	53	2014-01-15	118
Fay Osborn	4	2015-06-17	119
Florine Andrews	10	2016-01-27	120
Jacquelyn Whitfield	5	2014-07-02	121

Figure 9 Question 8 Output

Question 9

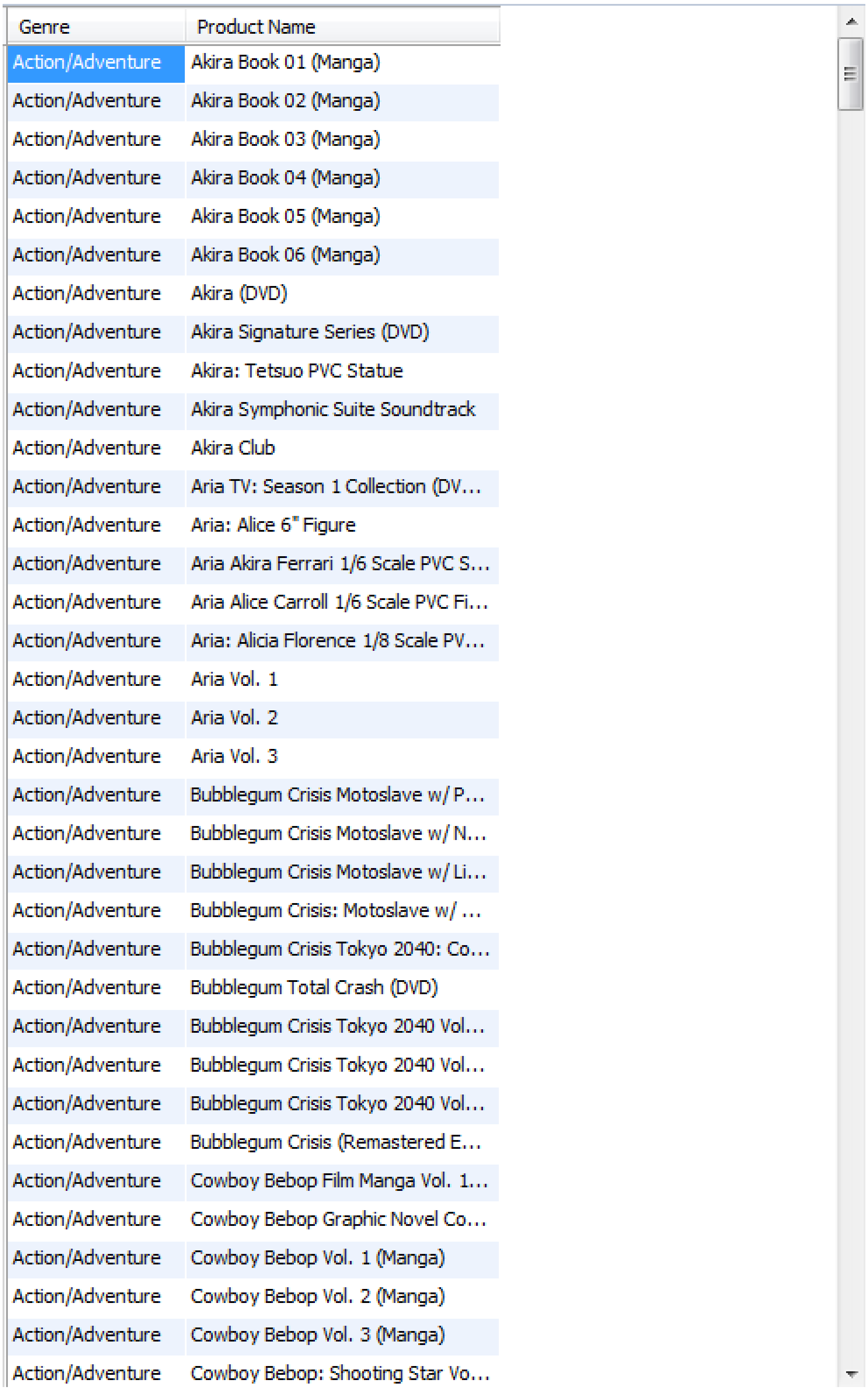
Create a view that lists all products and their descriptions, sorted by genre, e.g.

Genre	Product Name
Action/Adventure	Akira Book 01 (Manga)
….	….

Note that some products will appear more than once in the list as the product series may fall under many genres.

```
CREATE VIEW genre_product_view AS
SELECT genre AS 'Genre', name AS 'Product Name'
FROM product p
JOIN xrefseriesgenre x on p.series = x.series ORDER BY genre;
```

The xrefseriesgenre and product tables join to give the product name and genre using series. Using "SELECT * FROM genre_product_view;" to show output for this view.



Genre	Product Name
Action/Adventure	Akira Book 01 (Manga)
Action/Adventure	Akira Book 02 (Manga)
Action/Adventure	Akira Book 03 (Manga)
Action/Adventure	Akira Book 04 (Manga)
Action/Adventure	Akira Book 05 (Manga)
Action/Adventure	Akira Book 06 (Manga)
Action/Adventure	Akira (DVD)
Action/Adventure	Akira Signature Series (DVD)
Action/Adventure	Akira: Tetsuo PVC Statue
Action/Adventure	Akira Symphonic Suite Soundtrack
Action/Adventure	Akira Club
Action/Adventure	Aria TV: Season 1 Collection (DV…
Action/Adventure	Aria: Alice 6" Figure
Action/Adventure	Aria Akira Ferrari 1/6 Scale PVC S…
Action/Adventure	Aria Alice Carroll 1/6 Scale PVC Fi…
Action/Adventure	Aria: Alicia Florence 1/8 Scale PV…
Action/Adventure	Aria Vol. 1
Action/Adventure	Aria Vol. 2
Action/Adventure	Aria Vol. 3
Action/Adventure	Bubblegum Crisis Motoslave w/ P…
Action/Adventure	Bubblegum Crisis Motoslave w/ N…
Action/Adventure	Bubblegum Crisis Motoslave w/ Li…
Action/Adventure	Bubblegum Crisis: Motoslave w/ …
Action/Adventure	Bubblegum Crisis Tokyo 2040: Co…
Action/Adventure	Bubblegum Total Crash (DVD)
Action/Adventure	Bubblegum Crisis Tokyo 2040 Vol…
Action/Adventure	Bubblegum Crisis Tokyo 2040 Vol…
Action/Adventure	Bubblegum Crisis Tokyo 2040 Vol…
Action/Adventure	Bubblegum Crisis (Remastered E…
Action/Adventure	Cowboy Bebop Film Manga Vol. 1…
Action/Adventure	Cowboy Bebop Graphic Novel Co…
Action/Adventure	Cowboy Bebop Vol. 1 (Manga)
Action/Adventure	Cowboy Bebop Vol. 2 (Manga)
Action/Adventure	Cowboy Bebop Vol. 3 (Manga)
Action/Adventure	Cowboy Bebop: Shooting Star Vo…

Figure 10 Question 9 Output

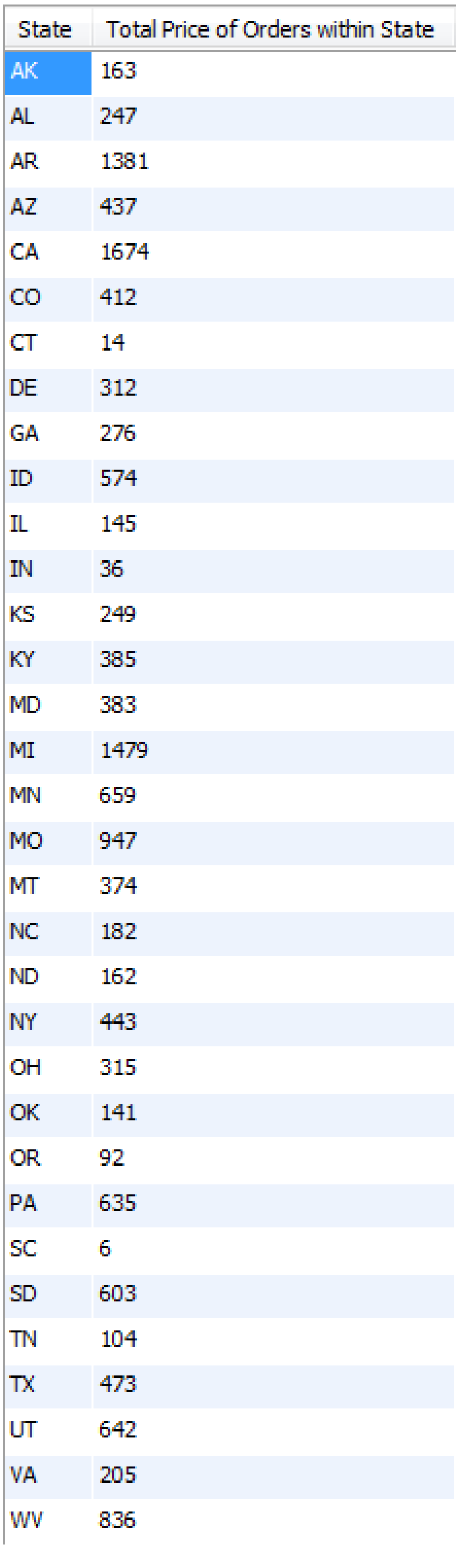
Question 10

List each state (account.state_province) for which account holders are registered, with the total value of orders placed within that State. The total price should be rounded to the nearest integer.

State	Total Price of Orders within State
CA	543
OH	610
….	….

SELECT state_province AS State,
ROUND(SUM(total_price),0)
AS 'Total Price of Orders within State'
FROM account a
JOIN orders o ON a.user_id = o.user_id
GROUP BY state_province;

Account and orders are joined using the user_id, the sum of the orders.total_price field is grouped for each state using account.state_province



State	Total Price of Orders within State
AK	163
AL	247
AR	1381
AZ	437
CA	1674
CO	412
CT	14
DE	312
GA	276
ID	574
IL	145
IN	36
KS	249
KY	385
MD	383
MI	1479
MN	659
MO	947
MT	374
NC	182
ND	162
NY	443
OH	315
OK	141
OR	92
PA	635
SC	6
SD	603
TN	104
TX	473
UT	642
VA	205
WV	836

Figure 10 Question 10

Question 11

List all product numbers, with their list price (selling price), unit price (cost price), and mark-up (% difference profit), where the mark-up between the unit price and what it's sold for is greater than or equal to the average mark-up.

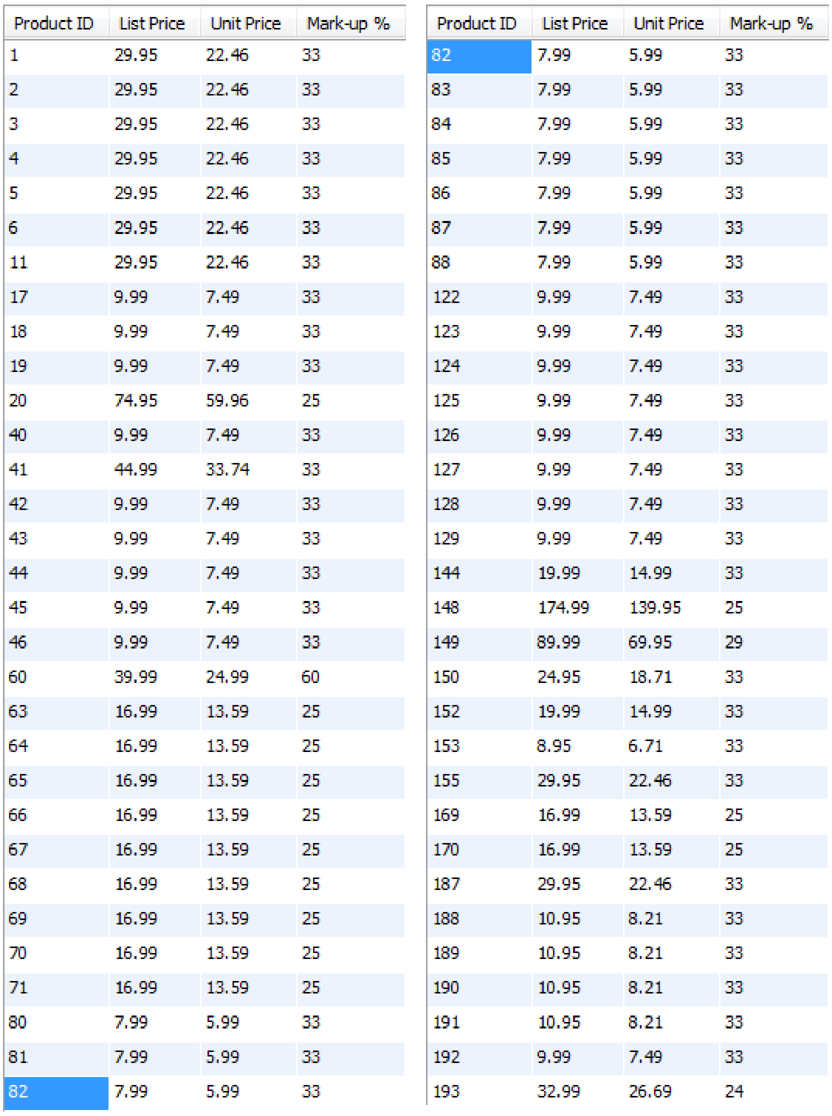
Product ID	List Price	Unit Price	Mark-up %
149	19.99	14.99	33
….	….		

SELECT product_id AS 'Product ID',
listprice AS 'List Price', unitprice AS 'Unit Price',
ROUND(100 * (listprice - unitprice) / unitprice) AS 'Mark-up %' FROM item
WHERE (100 * (listprice - unitprice) / unitprice) >=
(SELECT AVG(100 * (listprice - unitprice) / unitprice) FROM item);

FORMAT() or ROUND() looks like it gives the same mark-up percentage.

Average mark-up 20.65

The difference between item.listprice and item.unitprice is divided by item.unitprice to give the percentage increase from the original unit price.



Product ID	List Price	Unit Price	Mark-up %
1	29.95	22.46	33
2	29.95	22.46	33
3	29.95	22.46	33
4	29.95	22.46	33
5	29.95	22.46	33
6	29.95	22.46	33
11	29.95	22.46	33
17	9.99	7.49	33
18	9.99	7.49	33
19	9.99	7.49	33
20	74.95	59.96	25
40	9.99	7.49	33
41	44.99	33.74	33
42	9.99	7.49	33
43	9.99	7.49	33
44	9.99	7.49	33
45	9.99	7.49	33
46	9.99	7.49	33
60	39.99	24.99	60
63	16.99	13.59	25
64	16.99	13.59	25
65	16.99	13.59	25
66	16.99	13.59	25
67	16.99	13.59	25
68	16.99	13.59	25
69	16.99	13.59	25
70	16.99	13.59	25
71	16.99	13.59	25
80	7.99	5.99	33
81	7.99	5.99	33
82	7.99	5.99	33

Product ID	List Price	Unit Price	Mark-up %
82	7.99	5.99	33
83	7.99	5.99	33
84	7.99	5.99	33
85	7.99	5.99	33
86	7.99	5.99	33
87	7.99	5.99	33
88	7.99	5.99	33
122	9.99	7.49	33
123	9.99	7.49	33
124	9.99	7.49	33
125	9.99	7.49	33
126	9.99	7.49	33
127	9.99	7.49	33
128	9.99	7.49	33
129	9.99	7.49	33
144	19.99	14.99	33
148	174.99	139.95	25
149	89.99	69.95	29
150	24.95	18.71	33
152	19.99	14.99	33
153	8.95	6.71	33
155	29.95	22.46	33
169	16.99	13.59	25
170	16.99	13.59	25
187	29.95	22.46	33
188	10.95	8.21	33
189	10.95	8.21	33
190	10.95	8.21	33
191	10.95	8.21	33
192	9.99	7.49	33
193	32.99	26.69	24

Figure 11 Question 11 Output

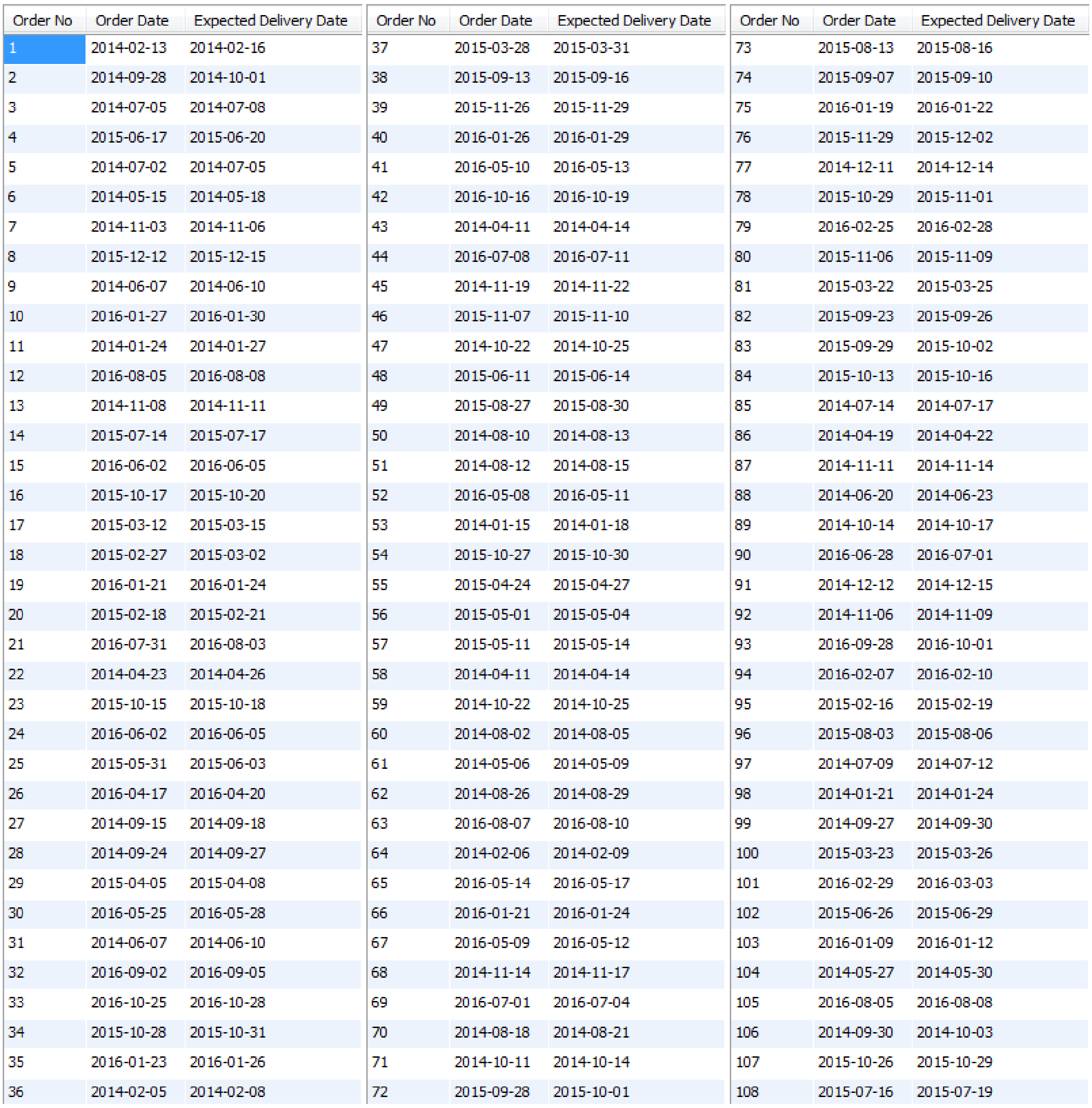
Question 12

List all order numbers, with their order date, and its 'Expected Delivery Date' which is calculated as the order-date plus 3 days.

Order No	Order Date	Expected Delivery Date
12	2008-08-05	2008-08-08
….	….	….

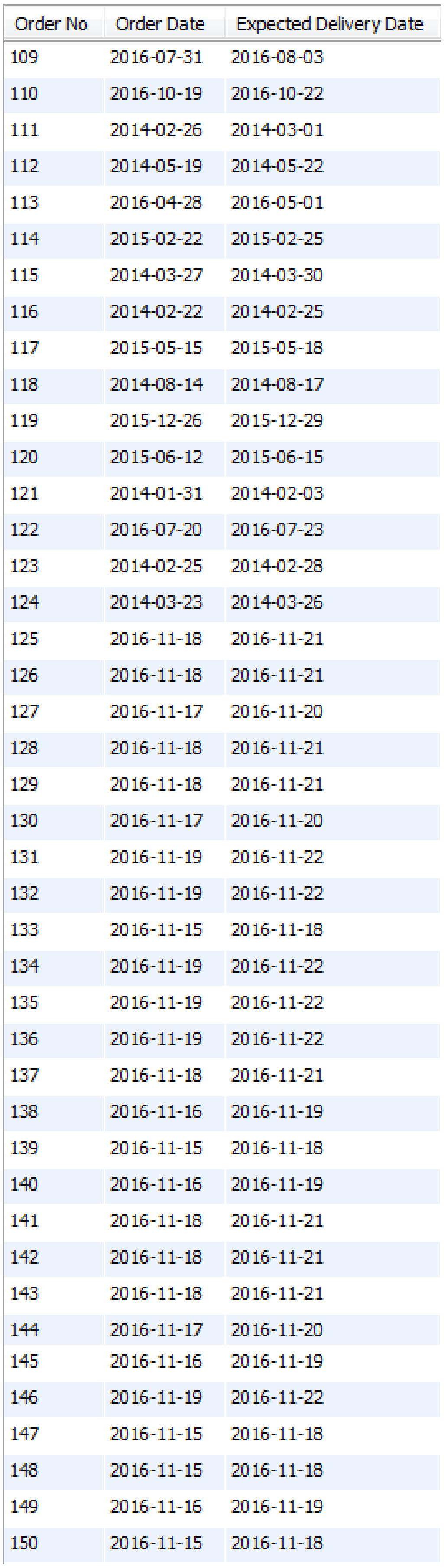
SELECT order_id AS 'Order No', order_date AS 'Order Date', DATE_ADD(order_date, INTERVAL 3 DAY) AS 'Expected Delivery Date' FROM orders;

The DATE_ADD command stops the date coming out as the 33rd of April if 3 was added to the 30th of April using normal addition



Order No	Order Date	Expected Delivery Date
1	2014-02-13	2014-02-16
2	2014-09-28	2014-10-01
3	2014-07-05	2014-07-08
4	2015-06-17	2015-06-20
5	2014-07-02	2014-07-05
6	2014-05-15	2014-05-18
7	2014-11-03	2014-11-06
8	2015-12-12	2015-12-15
9	2014-06-07	2014-06-10
10	2016-01-27	2016-01-30
11	2014-01-24	2014-01-27
12	2016-08-05	2016-08-08
13	2014-11-08	2014-11-11
14	2015-07-14	2015-07-17
15	2016-06-02	2016-06-05
16	2015-10-17	2015-10-20
17	2015-03-12	2015-03-15
18	2015-02-27	2015-03-02
19	2016-01-21	2016-01-24
20	2015-02-18	2015-02-21
21	2016-07-31	2016-08-03
22	2014-04-23	2014-04-26
23	2015-10-15	2015-10-18
24	2016-06-02	2016-06-05
25	2015-05-31	2015-06-03
26	2016-04-17	2016-04-20
27	2014-09-15	2014-09-18
28	2014-09-24	2014-09-27
29	2015-04-05	2015-04-08
30	2016-05-25	2016-05-28
31	2014-06-07	2014-06-10
32	2016-09-02	2016-09-05
33	2016-10-25	2016-10-28
34	2015-10-28	2015-10-31
35	2016-01-23	2016-01-26
36	2014-02-05	2014-02-08
37	2015-03-28	2015-03-31
38	2015-09-13	2015-09-16
39	2015-11-26	2015-11-29
40	2016-01-26	2016-01-29
41	2016-05-10	2016-05-13
42	2016-10-16	2016-10-19
43	2014-04-11	2014-04-14
44	2016-07-08	2016-07-11
45	2014-11-19	2014-11-22
46	2015-11-07	2015-11-10
47	2014-10-22	2014-10-25
48	2015-06-11	2015-06-14
49	2015-08-27	2015-08-30
50	2014-08-10	2014-08-13
51	2014-08-12	2014-08-15
52	2016-05-08	2016-05-11
53	2014-01-15	2014-01-18
54	2015-10-27	2015-10-30
55	2015-04-24	2015-04-27
56	2015-05-01	2015-05-04
57	2015-05-11	2015-05-14
58	2014-04-11	2014-04-14
59	2014-10-22	2014-10-25
60	2014-08-02	2014-08-05
61	2014-05-06	2014-05-09
62	2014-08-26	2014-08-29
63	2016-08-07	2016-08-10
64	2014-02-06	2014-02-09
65	2016-05-14	2016-05-17
66	2016-01-21	2016-01-24
67	2016-05-09	2016-05-12
68	2014-11-14	2014-11-17
69	2016-07-01	2016-07-04
70	2014-08-18	2014-08-21
71	2014-10-11	2014-10-14
72	2015-09-28	2015-10-01
73	2015-08-13	2015-08-16
74	2015-09-07	2015-09-10
75	2016-01-19	2016-01-22
76	2015-11-29	2015-12-02
77	2014-12-11	2014-12-14
78	2015-10-29	2015-11-01
79	2016-02-25	2016-02-28
80	2015-11-06	2015-11-09
81	2015-03-22	2015-03-25
82	2015-09-23	2015-09-26
83	2015-09-29	2015-10-02
84	2015-10-13	2015-10-16
85	2014-07-14	2014-07-17
86	2014-04-19	2014-04-22
87	2014-11-11	2014-11-14
88	2014-06-20	2014-06-23
89	2014-10-14	2014-10-17
90	2016-06-28	2016-07-01
91	2014-12-12	2014-12-15
92	2014-11-06	2014-11-09
93	2016-09-28	2016-10-01
94	2016-02-07	2016-02-10
95	2015-02-16	2015-02-19
96	2015-08-03	2015-08-06
97	2014-07-09	2014-07-12
98	2014-01-21	2014-01-24
99	2014-09-27	2014-09-30
100	2015-03-23	2015-03-26
101	2016-02-29	2016-03-03
102	2015-06-26	2015-06-29
103	2016-01-09	2016-01-12
104	2014-05-27	2014-05-30
105	2016-08-05	2016-08-08
106	2014-09-30	2014-10-03
107	2015-10-26	2015-10-29
108	2015-07-16	2015-07-19

Figure 13 Question 12 Output 1/2



Order No	Order Date	Expected Delivery Date
109	2016-07-31	2016-08-03
110	2016-10-19	2016-10-22
111	2014-02-26	2014-03-01
112	2014-05-19	2014-05-22
113	2016-04-28	2016-05-01
114	2015-02-22	2015-02-25
115	2014-03-27	2014-03-30
116	2014-02-22	2014-02-25
117	2015-05-15	2015-05-18
118	2014-08-14	2014-08-17
119	2015-12-26	2015-12-29
120	2015-06-12	2015-06-15
121	2014-01-31	2014-02-03
122	2016-07-20	2016-07-23
123	2014-02-25	2014-02-28
124	2014-03-23	2014-03-26
125	2016-11-18	2016-11-21
126	2016-11-18	2016-11-21
127	2016-11-17	2016-11-20
128	2016-11-18	2016-11-21
129	2016-11-18	2016-11-21
130	2016-11-17	2016-11-20
131	2016-11-19	2016-11-22
132	2016-11-19	2016-11-22
133	2016-11-15	2016-11-18
134	2016-11-19	2016-11-22
135	2016-11-19	2016-11-22
136	2016-11-19	2016-11-22
137	2016-11-18	2016-11-21
138	2016-11-16	2016-11-19
139	2016-11-15	2016-11-18
140	2016-11-16	2016-11-19
141	2016-11-18	2016-11-21
142	2016-11-18	2016-11-21
143	2016-11-18	2016-11-21
144	2016-11-17	2016-11-20
145	2016-11-16	2016-11-19
146	2016-11-19	2016-11-22
147	2016-11-15	2016-11-18
148	2016-11-15	2016-11-18
149	2016-11-16	2016-11-19
150	2016-11-15	2016-11-18

Figure 14 Question 12 Output 2/2