Table of Contents

[1. Unmatched queries using outer join 1](#_Toc363654377)

[2. Queries using the Altgeld mart tables 2](#_Toc363654378)

[3. Unmatched queries using subqueries 4](#_Toc363654379)

[4. What can go wrong? 5](#_Toc363654380)

Inner joins are great for finding customers with Orders and for finding Products that have been ordered. But we often want to find customers who have no orders or products that no one has ordered. These are sometimes called unmatched queries since we are looking for customers in the customer table who have no matching rows in the order headers table. We will see several ways to do this. For now we will look at two approaches: (1) using the outer join and (2) using subqueries.

1. Unmatched queries using outer join

In the previous document we used the outer join to find employees with and without an assigned department. A variation on the outer join is a query to display only those employees who have no assigned department. Be careful to select the proper column for testing against null. With these tests you do not want to use the join with the Using (col) syntax because you have to specify the exact column you are looking for. Compare the following two queries. We want departments with no employees.

1. Unmatched rows. Departments which do not have any employees

Select z\_em\_emp.d\_id as "em\_emp.d\_id"

, z\_em\_dept.d\_id as "em\_dept.d\_id"

, d\_name

From z\_em\_dept

Left join z\_em\_emp on z\_em\_dept.d\_id = z\_em\_emp.d\_id

Where z\_em\_emp.d\_id is null

;

+-------------+--------------+-----------+

| em\_emp.d\_id | em\_dept.d\_id | d\_name |

+-------------+--------------+-----------+

| NULL | 200 | Marketing |

+-------------+--------------+-----------+

1. Unmatched rows. Take care which attribute you test. Since we are retrieving all data from the department table we will not have nulls in the department table id attribute.

Select z\_em\_emp.d\_id as "em\_emp.d\_id"

, z\_em\_dept.d\_id as "em\_dept.d\_id"

, d\_name

From z\_em\_dept

Left join z\_em\_emp on z\_em\_dept.d\_id = z\_em\_emp.d\_id

Where z\_em\_dept.d\_id IS NULL

;

Empty set (0.00 sec)

Still looking for departments which do not have any employees. Compare the following two demos.

1. Using a column name join. In the standard you cannot qualify the column used in this join and we get back no departments without employees. Note that this does not present as an error; we simply get no rows returned.

Select d\_id

, d\_name

From z\_em\_dept

Left join z\_em\_emp using(d\_id)

Where d\_id is null;

Empty set (0.00 sec)

1. MySQL allows qualification of the joining column D\_ID to specify the table name. Since this is a non-standard extension you might with to stay with the condition join

Select z\_em\_emp.d\_id as "em\_emp.d\_id"

, z\_em\_dept.d\_id as "em\_dept.d\_id"

, d\_name

From z\_em\_dept

Left join z\_em\_emp using(d\_id)

Where z\_em\_emp.d\_id is null;

+-------------+--------------+-----------+

| em\_emp.d\_id | em\_dept.d\_id | d\_name |

+-------------+--------------+-----------+

| NULL | 200 | Marketing |

+-------------+--------------+-----------+

1. If you have troubles with setting up this type of query, run the query without the null filter first and examine the columns for the rows you want to return. Here we can see that we want to test the z\_em\_emp.d\_id column for nulls.

Select z\_em\_emp.d\_id as "em\_emp.d\_id"

, z\_em\_dept.d\_id as "em\_dept.d\_id"

, d\_name

From z\_em\_dept

Left join z\_em\_emp on z\_em\_dept.d\_id = z\_em\_emp.d\_id;

+-------------+--------------+---------------+

| em\_emp.d\_id | em\_dept.d\_id | d\_name |

+-------------+--------------+---------------+

| 100 | 100 | Manufacturing |

| 150 | 150 | Accounting |

| 150 | 150 | Accounting |

| NULL | 200 | Marketing |

| 250 | 250 | Research |

+-------------+--------------+---------------+

1. Queries using the Altgeld mart tables
2. Customers without orders. We can use the Using (col) syntax here since the filter column is not the join column.

Select cust\_id

, cust\_name\_last

, ord\_id

From a\_oe.customers

Left join a\_oe.order\_headers using(cust\_id)

Where cust\_id between 404900 and 409030

and ord\_id IS NULL

Order by cust\_id;

+---------+----------------+--------+

| cust\_id | cust\_name\_last | ord\_id |

+---------+----------------+--------+

| 409010 | Morris | NULL |

| 409020 | Max | NULL |

+---------+----------------+--------+

2 rows in set (0.00 sec)

1. If we try to find orders with no customers, we have no rows returned. Our database is set up to reject any order that is not associated with a customer. This would be a good query to run on poorly designed databases to locate orphaned rows.

Select CS.cust\_id

, CS.cust\_name\_last

, OH.ord\_id

From a\_oe.order\_headers OH

Left Join a\_oe.customers CS on CS.cust\_id = OH.cust\_id

Where OH.cust\_id is null

;

Empty set (0.00 sec)

1. What is the product name and list price for the products that are not selling? These would be products in the products table that do not appear on any order.

Select catg\_id as catg

, prod\_id as p\_id

, prod\_desc as product

, prod\_list\_price as price

From a\_prd.products

Left Join a\_oe.order\_details using (prod\_id)

Where ord\_id is null

Order by catg\_id, prod\_id

;

+------+------+----------------------------------------------+--------+

| catg | p\_id | product | price |

+------+------+----------------------------------------------+--------+

| APL | 1126 | Low Energy washer Dryer combo | 850.00 |

| APL | 4569 | Sized for the apartment | 349.95 |

| GFD | 5000 | Cello bag of mixed fingerling potatoes | 12.50 |

| GFD | 5001 | Dundee Ginger Preserve 12 oz jar | 5.00 |

| HW | 1160 | Stand Mixer with attachments | 149.99 |

| HW | 4575 | GE model 34PG98 | 49.95 |

| MUS | 2234 | Charles Mingus - Pithecanthropus Erectus | 15.88 |

| MUS | 2337 | John Coltrane - Blue Train | 15.87 |

| MUS | 2487 | Stanley Turrentine - Don't Mess With Mr. T | 9.45 |

| MUS | 2933 | David Newman - I Remember Brother Ray | 12.45 |

| MUS | 2987 | Stanley Turrentine - Ballads | 15.87 |

| PET | 1142 | Bird seed mix with sunflowers | 2.50 |

| PET | 1143 | Bird seed mix with more sunflower seeds | 2.50 |

| PET | 4567 | Our highest end cat tree- you gotta see this | 549.99 |

| PET | 4568 | Satin four-poster cat bed | 549.99 |

+------+------+----------------------------------------------+--------+

15 rows in set (0.00 sec)

1. Do we have any products with no inventory? This is an example of a question that needs clarification. For this query we will consider this to be either no inventory row at all or an inventory level of zero.

Select catg\_id

, a\_prd.products.prod\_id

, prod\_name as product

, quantity\_on\_hand

From a\_prd.products

Left Join a\_prd.inventory on a\_prd.products.prod\_id = a\_prd.inventory.prod\_id

Where a\_prd.inventory.prod\_id is null

or quantity\_on\_hand = 0

Order by quantity\_on\_hand , catg\_id, prod\_id

;

+---------+---------+---------------------+------------------+

| catg\_id | prod\_id | product | quantity\_on\_hand |

+---------+---------+---------------------+------------------+

| APL | 1126 | WasherDryer | NULL |

| APL | 4569 | Mini Dryer | NULL |

| GFD | 5000 | Fingerling Potatoes | NULL |

| GFD | 5001 | Ginger Preserve | NULL |

. . . rows omitted

| PET | 4567 | Deluxe Cat Tree | NULL |

| PET | 4568 | Deluxe Cat Bed | NULL |

| PET | 4576 | Cosmo cat nip | NULL |

| PET | 4577 | Cat leash | NULL |

| SPG | 1050 | Stationary bike | 0 |

+---------+---------+---------------------+------------------+

26 rows in set (0.00 sec)

1. Unmatched queries using subqueries

Some people find this syntax easier to understand. We are looking for data where we have a value in one table and we do not have that value in another table.

1. This is the Customers without orders query done using a subquery. ( This is not including the cust\_id range filter we had before) This filters for customer id values that are not in the order headers table- that would be customers with no orders.

Select cust\_id, cust\_name\_last

From a\_oe.customers

Where cust\_id NOT IN (

Select cust\_id

From a\_oe.order\_headers

)

;

+---------+----------------+

| cust\_id | cust\_name\_last |

+---------+----------------+

| 400801 | Washington |

| 402110 | Coltrane |

| 402120 | McCoy |

| 402500 | Jones |

| 403500 | Stevenson |

| 403750 | O'Leary |

| 403760 | O'Leary |

| 404150 | Dancer |

| 404180 | Shay |

| 404890 | Kelley |

| 409010 | Morris |

| 409020 | Max |

+---------+----------------+

12 rows in set (0.00 sec)

1. What is the product name and list price for the products that are not selling? These would be products in the products table that do not appear on any order.   
   This query does not need table aliases since each part of the query is referencing a single table

Select catg\_id as catg

, prod\_id as p\_id

, prod\_desc as product

, prod\_list\_price as price

From a\_prd.products

Where prod\_id NOT IN (

Select prod\_id

From a\_oe.order\_details

)

Order by catg\_id, prod\_id

;

+------+------+----------------------------------------------+--------+

| catg | p\_id | product | price |

+------+------+----------------------------------------------+--------+

| APL | 1126 | Low Energy washer Dryer combo | 850.00 |

| APL | 4569 | Sized for the apartment | 349.95 |

| GFD | 5000 | Cello bag of mixed fingerling potatoes | 12.50 |

| GFD | 5001 | Dundee Ginger Preserve 12 oz jar | 5.00 |

| HW | 1160 | Stand Mixer with attachments | 149.99 |

| HW | 4575 | GE model 34PG98 | 49.95 |

| MUS | 2234 | Charles Mingus - Pithecanthropus Erectus | 15.88 |

| MUS | 2337 | John Coltrane - Blue Train | 15.87 |

| MUS | 2487 | Stanley Turrentine - Don't Mess With Mr. T | 9.45 |

| MUS | 2933 | David Newman - I Remember Brother Ray | 12.45 |

| MUS | 2987 | Stanley Turrentine - Ballads | 15.87 |

| PET | 1142 | Bird seed mix with sunflowers | 2.50 |

| PET | 1143 | Bird seed mix with more sunflower seeds | 2.50 |

| PET | 4567 | Our highest end cat tree- you gotta see this | 549.99 |

| PET | 4568 | Satin four-poster cat bed | 549.99 |

+------+------+----------------------------------------------+--------+

15 rows in set (0.00 sec)

1. What can go wrong?

Suppose we want to find employees who are not associated with any orders. First do a left join to see what the data looks like.

1. Left join Employees to Orders

Select emp\_id, name\_last, ord\_id

From emp\_employees

Left join oe\_order\_headers on emp\_id = sales\_rep\_id

;

+--------+-----------+--------+

| emp\_id | name\_last | Ord\_id |

+--------+-----------+--------+

| 100 | King | NULL |

| 101 | Koch | NULL |

| 102 | D'Haa | NULL |

| 103 | Hunol | NULL |

| 104 | Ernst | NULL |

| 108 | Green | NULL |

| 109 | Fiet | NULL |

| 110 | Chen | NULL |

| 145 | Russ | 112 |

| 145 | Russ | 130 |

| 145 | Russ | 312 |

| 145 | Russ | 405 |

| 145 | Russ | 505 |

| 145 | Russ | 540 |

| 146 | Partne | NULL |

| 150 | Tuck | 105 |

| 150 | Tuck | 106 |

| 150 | Tuck | 107 |

| 150 | Tuck | 111 |

81 rows in set

We could add a filter to find the rows where the Ord\_id is null.

1. Left join Employees to Orders with null order id. These are employees who are not associated with any order.

Select emp\_id, name\_last

From a\_emp.employees

Left join a\_oe.order\_headers on emp\_id = sales\_rep\_id

Where ord\_id is null;

+--------+-----------+

| emp\_id | name\_last |

+--------+-----------+

| 100 | King |

| 101 | Koch |

| 102 | D'Haa |

| 103 | Hunol |

| 104 | Ernst |

| 108 | Green |

| 109 | Fiet |

| 110 | Chen |

| 146 | Partne |

| 160 | Dorna |

| 161 | Dewal |

| 162 | Holme |

| 200 | Whale |

| 201 | Harts |

| 203 | Mays |

| 204 | King |

| 205 | Higgs |

| 206 | Geitz |

| 207 | Russ |

+--------+-----------+

19 rows in set (0.00 sec)

What if we try this with a subquery?

1. Subquery version 1 We are filter for employee id that are not in the appropriate column in the order headers table. This returns no rows at all! Before you read on to the next demo try to figure out why this might happen. (What is the usual villain when a query goes bad?)

Select emp\_id, name\_last

From a\_emp.employees

Where emp\_id NOT IN (

Select sales\_rep\_id

From a\_oe.order\_headers

);

Empty set (0.00 sec)

Remember that a Not In () predicate returns no rows if there is a null in the list.

1. Subquery version 2-

Select emp\_id, name\_last

From a\_emp.employees

Where emp\_id NOT IN (

Select sales\_rep\_id

From a\_oe.order\_headers

Where sales\_rep\_id is not null)

;

*Same result set as with the outer join*

So now the question is: why did the other subqueries work? We were filtering on an attribute that was a not null attribute.