

SQL with Multiple Tables

By,

BHARATH VENKATESH SRINIVASAN

Question: Show me the vendors and the products they supply to us for products that cost less than \$100

Translation: List of all the vendor names and product names where product cost is less than \$100

Cleanup: Select vendornames from Vendor table and productnames from Product table where retailprice is less than \$100

Query:

```
select v.VendName , p.ProductName
```

```
from vendors v
```

```
join product_vendors pv on v.VendorID = pv.VendorID
```

```
join products p on p.ProductNumber = pv.ProductNumber
```

```
where p.RetailPrice < 100
```

```
group by 1,2;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1  # Question 2
2  # Show me the vendors and the products they supply to us for products that cost less than $100
3  # Translation: List of all the vendor names and product names where product cost is greater than $100
4  # Cleanup: Select vendornames from Vendor table and productnames from Product table where retailprice is greater than $100
5  select v.VendName , p.ProductName
6  from vendors v
7  join product_vendors pv on v.VendorID = pv.VendorID
8  join products p on p.ProductNumber = pv.ProductNumber
9  where p.RetailPrice < 100
10 group by 1,2;
```

The Results Grid shows the following data:

VendName	ProductName
Dog Ear	Dog Ear Cyclecomputer
Lone Star Bike Supply	Dog Ear Cyclecomputer
Big Sky Mountain Bikes	Victoria Pro All Weather Tires
Lone Star Bike Supply	Victoria Pro All Weather Tires
Dog Ear	Dog Ear Helmet Mount Mirrors
Lone Star Bike Supply	Dog Ear Helmet Mount Mirrors
Viscount	Viscount C-500 Wireless Bike Computer
Sun Sports Suppliers	Kryptonite Advanced 2000 U-Lock
Lone Star Bike Supply	Kryptonite Advanced 2000 U-Lock

The Output tab shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
1	22:56:15	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.Ve...	Error Code: 1146. Table 'recipedb.vendors' doesn't exist	0.000 sec
2	22:56:18	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.Ve...	Error Code: 1146. Table 'recipedb.vendors' doesn't exist	0.000 sec
3	22:56:25	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.Ve...	Error Code: 1146. Table 'recipedb.vendors' doesn't exist	0.000 sec
4	22:56:48	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.Ve...	Error Code: 1146. Table 'recipedb.vendors' doesn't exist	0.000 sec
5	22:57:07	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.Ve...	63 row(s) returned	0.000 sec / 0.000 sec

Question: Find the agents and entertainers who live in the same postal code.

Translation: Select all agents and entertainers who has same postal code

Cleanup: Select agentFirstName, agentLastName from agents and entertainers stage name from entertainers table where zip code is same for both agent and entertainer

Query:

```
select distinct concat (a.AgtFirstName, ' ', a.AgtLastName) as agentname, et.EntStageName, et.EntertainerID
```

```
from agents a
```

```
join entertainers et
```

```
where a.AgtZipCode = et.EntZipCode
```

```
order by 1,2;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'entertainmentagencydb' schema with tables like agents, entertainers, and entertainers_styles. The main editor window contains the following SQL query:

```
# Question 3
# Find the agents and entertainers who live in the same postal code.
# Translation: Select all agents and entertainers who has same postal code
# Cleanup: Select agentFirstName, agentLastName from agents and entertainers stage name from entertainers table where zip code is same for both agent and entertainer

select distinct concat (a.AgtFirstName, ' ', a.AgtLastName) as agentname, et.EntStageName, et.EntertainerID
from agents a
join entertainers et
where a.AgtZipCode = et.EntZipCode
order by 1,2;
```

The 'Result Grid' shows the following data:

agentname	EntStageName	EntertainerID
Caleb Vescas	Carol Peacock Trio	1001
Caleb Vescas	JV & the Deep Six	1003
Carol Vescas	Susan McLain	1012
John Kennedy	Country Feeling	1008
John Kennedy	Saturday Revue	1010
Karen Smith	Country Feeling	1008
Karen Smith	Saturday Revue	1010
Maria Patterson	Susan McLain	1012
Chris Robyn	Country Feeling	1008

The 'Output' pane at the bottom shows the execution log with error messages for tables 'vendors' and 'product_vendors' that do not exist, and a successful result for the main query.

Question: Display all recipe classes and any recipes that might be associated with them.

Translation: Select all recipe classes and the recipes if any associated with recipe classes

Cleanup: Select recipeclasses from recipe_classes table and preparation from recipes table that re present in recipe_classes

Query:

```
select rc.RecipeClassID, RecipeClassDescription , r.RecipeTitle
```

```
from recipe_classes rc
```

```
left outer join recipes r on rc.RecipeClassID = r.RecipeClassID
```

```
order by 1;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' list, including 'recipe_classes' and 'recipes'. The main window shows a query titled 'Query 1' with the following SQL code:

```
# Question 4
#Display all recipe classes and any recipes that might be associated with them.
# Translation: Select all recipe classes and the recipes if any associated with recipe classes
# Cleanup: Select recipeclasses from recipe_classes table and preparation from recipes table that re present in recipe_classes

select rc.RecipeClassID, RecipeClassDescription , r.RecipeTitle
from recipe_classes rc
left outer join recipes r on rc.RecipeClassID = r.RecipeClassID
order by 1;
```

The 'Result Grid' shows the following data:

RecipeClassID	RecipeClassDescription	RecipeTitle
1	Main course	Irish Stew
1	Main course	Fettuccini Alfredo
1	Main course	Pollo Piccolo
1	Main course	Roast Beef
1	Main course	Huachinango Veracruzana (Red Snapper, Veracruz style)
1	Main course	Tourtière (French-Canadian Pork Pie)
1	Main course	Salmon Fillets in Parchment Paper
2	Vegetable	Garlic Green Beans
7	Vegetable	Asparagus

The 'Output' pane shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
2	22:56:18	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.V...	Error Code: 1146. Table 'recipe_classes.vendors' doesn't exist	0.000 sec
3	22:56:26	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.V...	Error Code: 1146. Table 'recipe_classes.vendors' doesn't exist	0.000 sec
4	22:56:48	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.V...	Error Code: 1146. Table 'recipe_classes.vendors' doesn't exist	0.000 sec
5	22:57:07	select v.VendName , p.ProductName from vendors v join product_vendors pv on v.VendorID = pv.V...	63 row(s) returned	0.000 sec / 0.000 sec
6	22:58:38	select distinct concat (a.AgFirst Name, ' ', a.AgLast Name) as agentname, et. EntStageName, et. Ente...	10 row(s) returned	0.000 sec / 0.000 sec
7	23:00:20	select rc.RecipeClassID, RecipeClassDescription , r.RecipeTitle from recipe_classes rc left outer join...	16 row(s) returned	0.000 sec / 0.000 sec

Question: What products have never been ordered?

Translation: Select all the products that are never ordered

Cleanup: Select products from products table where productnumber from order_details table is equal to null after doing left join on products table.

Query:

```
select p.ProductName as Productname
```

```
from (products p
```

```
left outer join order_details od on p.ProductNumber = od.ProductNumber)
```

```
where od.ProductNumber is null;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'salesordersdb' selected. The main editor shows a SQL query for 'Query 1' with the following content:

```
33
34 # Question 5
35 # What products have never been ordered?
36 # Translation: Select all the products that are never ordered
37 # Cleanup: Select products from products table where productnumber from products table is not equal to product number from order_details table
38
39 select p.ProductName as Productname
40 from (products p
41 left outer join order_details od on p.ProductNumber = od.ProductNumber)
42 where od.ProductNumber is null;
43
```

Below the query editor, the 'Result Grid' shows the output of the query:

Productname
Victoria Pro All Weather Tires
Ultra-Pro Rain Jacket

The bottom panel shows the 'Output' tab with a log of database actions and messages, including execution times and row counts for various queries.

Question: List the subjects taught on Wednesday.

Translation: Select all the subjects that are taught on wednesday

Cleanup: Select subjectname from subjects where wednesdayschedule is equal to 1 from classes table

Query:

Select distinct s.SubjectName

from subjects s

join classes c on s.SubjectID = c.SubjectID

where c.WednesdaySchedule = 1

order by 1;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'schoolschedulingdb' schema selected. The main editor window shows a SQL query for 'Query 1' with the following text:

```
43
44 #Question 6
45 # List the subjects taught on Wednesday.
46 # Select all the subjects taught on Wednesday
47 # Select subjectname from subjects where wednesdayschedule is equal to 1 from classes table
48
49 • Select distinct s.SubjectName
50 from subjects s
51 join classes c on s.SubjectID = c.SubjectID
52 where c.WednesdaySchedule = 1
53 order by 1;
54
```

Below the query editor, the 'Result Grid' shows the results of the query. The first column is 'SubjectName' and the results are:

- Advanced English Grammar
- Art History
- Biological Principles
- Chemistry
- Composition - Fundamentals
- Composition - Intermediate
- Design

The bottom panel shows the 'Output' tab with a table of actions and messages:

#	Time	Action	Message	Duration / Fetch
9	23:02:02	select p.ProductName as Productname from (products p left outer join order_details od on p.Product...	2 row(s) returned	0.000 sec / 0.000 sec
10	23:02:49	Select distinct s.SubjectName from subjects s join classes c on s.SubjectID = c.SubjectID where c...	34 row(s) returned	0.000 sec / 0.000 sec
11	23:04:02	select p.ProductName as Productname from (products p left outer join order_details od on p.Product...	2 row(s) returned	0.000 sec / 0.000 sec
12	23:05:23	Select distinct c.CustFirstName, CustLastName, ErtStageName from entertainers et join engagement...	Error Code: 1146. Table 'salesordersdb.entertainers' doesn't exist	0.000 sec
13	23:05:32	Select distinct c.CustFirstName, CustLastName, ErtStageName from entertainers et join engagement...	12 row(s) returned	0.000 sec / 0.000 sec
14	23:14:49	Select distinct s.SubjectName from subjects s join classes c on s.SubjectID = c.SubjectID where c...	34 row(s) returned	0.000 sec / 0.000 sec

Question: Find the entertainers who played engagements for customers Berg or Hallmark

Translation: Select all entertainers who played engagements for customers Berg or Hallmark

Cleanup: Select EntStageName from entertainers who played engagements for customers Berg or Hallmark from customers table.

Query:

Select distinct c.CustFirstName, CustLastName, EntStageName

from entertainers et

join engagements e on et.EntertainerID = e.EntertainerID

join customers c on c.CustomerID = e.CustomerID

where c.CustLastName in ('Berg', 'Hallmark')

order by 1;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'entertainmentagencydb' schema with various tables. The main editor contains the following SQL query:

```
55 # Question 9
56 #Find the entertainers who played engagements for customers Berg or Hallmark
57 # Select all entertainers who played engagements for customers Berg or Hallmark
58 # Select EntStageName from entertainers who played engagements for customers Berg or Hallmark from customers table. Use engagements table for joining
59
60 • Select distinct c.CustFirstName, CustLastName, EntStageName
61 from entertainers et
62 join engagements e on et.EntertainerID = e.EntertainerID
63 join customers c on c.CustomerID = e.CustomerID
64 where c.CustLastName in ('Berg', 'Hallmark')
65 order by 1;
66
```

The 'Result Grid' shows the following data:

CustFirstName	CustLastName	EntStageName
Elizabeth	Hallmark	JV & the Deep Six
Elizabeth	Hallmark	Country Feeling
Elizabeth	Hallmark	Carol Peacock Trio
Elizabeth	Hallmark	Modern Dance
Elizabeth	Hallmark	Susan McLean
Elizabeth	Hallmark	Topazz
Matt	Berg	Jim Glynn

The 'Output' pane at the bottom shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
10	23:02:49	Select distinct s.SubjectName from subjects s join classes c on s.SubjectID = c.SubjectID where c...	34 row(s) returned	0.000 sec / 0.000 sec
11	23:04:02	select p.ProductName as Productname from (products p left outer join order_details od on p.Product...	2 row(s) returned	0.000 sec / 0.000 sec
12	23:05:23	Select distinct c.CustFirstName, CustLastName, EntStageName from entertainers et join engagement...	Error Code: 1146. Table 'salesordersdb.entertainers' doesn't exist	0.000 sec
13	23:05:32	Select distinct c.CustFirstName, CustLastName, EntStageName from entertainers et join engagement...	12 row(s) returned	0.000 sec / 0.000 sec
14	23:14:49	Select distinct s.SubjectName from subjects s join classes c on s.SubjectID = c.SubjectID where c...	34 row(s) returned	0.000 sec / 0.000 sec
15	23:18:27	Select distinct c.CustFirstName, CustLastName, EntStageName from entertainers et join engagement...	12 row(s) returned	0.000 sec / 0.000 sec