

SQL for Single-table Query

BY,

BHARATH VENKATESH SRINIVASAN (BXS210006)

1. Provide a list of students whose first name starts with 'B' or who live in Seattle

Input

Select the list of students from students table whose first name starts with 'B' or who live in Seattle // **Translation**

Select the StudFirstName, StudLastName from students where StudFistName starts with 'B' or Studcity is Seattle // **Clean up**

SQL Query: select * from students where Studcity = 'Seattle' or StudFirstName like 'B%' order by StudFirstName;

Output

The screenshot shows a database management tool interface. On the left, there's a 'SCHEMAS' pane with a tree view showing databases like 'recipedb', 'sakila', 'salesorden', and 'schoolscheduli'. The 'schoolscheduli' database is selected, and its tables are listed. The main area displays a SQL query in a text editor:

```
1  
2 # Select the list of students from students table whose first name starts with 'B' or who live in Seattle // Translati  
3 # Select the StudFirstName, StudLastName from students where StudFistName starts with 'B' or Studcity is Seattle // Cle  
4  
5 • select * from students  
6 where Studcity = 'Seattle' or StudFirstName like 'B%'  
7 order by StudFirstName;
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has columns for StudentID, StudFirstName, StudLastName, StudStreetAddress, StudCity, StudState, StudZipCode, StudAreaCode, StudPhoneNumber, and StudBirth. The results are as follows:

StudentID	StudFirstName	StudLastName	StudStreetAddress	StudCity	StudState	StudZipCode	StudAreaCode	StudPhoneNumber	StudBirth
1003	Betsy	Stadick	611 Alpine Drive	Palm Springs	CA	92263	760	555-2696	1989-03-
1015	Brannon	Jones	777 Fenexet Blvd	Long Beach	CA	90809	562	555-0399	1992-05-
1005	Doris	Hartwig	4726 - 11th Ave. N.E.	Seattle	WA	98105	206	555-2671	1992-08-
1014	Kendra	Bonnicksen	12330 Larchmont Lane	Seattle	WA	98105	206	555-2716	1988-10-
1018	Richard	Lum	754 Fourth Ave	Seattle	WA	98115	206	555-2296	1995-04-

At the bottom, there's an 'Output' pane showing the execution log with the following entries:

#	Time	Action	Message
15	15:55:25	select RecipeID, RecipeTitle from recipes where upper(Preparation) like ("%EGGS%") LIMIT ...	1 row(s) returned
16	15:58:06	select concat(StfFirstName, ' ', StfLastName) as 'Full Name', Salary from staff where Salary be...	14 row(s) returned
17	16:05:17	select * from students where Studcity = 'Seattle' or StudFirstName like 'B%' order by StudFirst...	5 row(s) returned

2. Show me an alphabetical list of all the staff members and their salaries if they make between \$40,000 and \$50,000 a year

Input

Select the list of all the staff members and their salaries from staff table if they make between \$40,000 and \$50,000 // **Translation**

Select StfFirstName, StfLastName from staff where salary is between 40000 and 50000
// **cleanup**

SQL Query: select concat(StfFirstName,' ',StfLastname) as 'Full Name', Salary from staff where Salary between 40000 and 50000 order by 'StfFirstName';

Output

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane lists databases: 'recipedb', 'sakila', 'salesorden', 'schoolscheduli', and 'Administrative'. The 'schoolscheduli' database is selected, showing its tables and views. The main area displays a SQL query in a text editor:

```
1  
2 # Select the list of all the staff members and their salaries from staff table if they make between $40,000 and $50,000  
3 # Select StfFirstName, StfLastName from staff where salary is between 40000 and 50000 // cleanup  
4  
5 • select concat(StfFirstName,' ',StfLastname) as 'Full Name', Salary from staff  
6 where Salary between 40000 and 50000 order by StfFirstName;  
7
```

Below the query editor, a 'Result Grid' shows the results of the query. The grid has two columns: 'Full Name' and 'Salary'. The results are as follows:

Full Name	Salary
Ann Patterson	45000.00
Caleb Viescas	45000.00
Carol Viescas	50000.00
Deb Waldal	44000.00
Jim Glynn	45000.00
Jim Wilson	50000.00
Katherine Ehrlich	45000.00
Kirk DeGraff	45000.00

At the bottom, an 'Output' pane shows the execution log. It includes the following entries:

- 17 16:05:17 select * from students where Studcity = 'Seattle' or StudFirstName like 'B%' order by StudFirst... 5 row(s) returned
- 18 16:09:16 select concat(StfFirstName,' ',StfLastname) as 'Full Name', Salary from staff where Salary be... Error Code: 1054. Unknown column 'StfFirstName' in 'order clause'
- 19 16:09:34 select concat(StfFirstName,' ',StfLastname) as 'Full Name', Salary from staff where Salary be... 14 row(s) returned

3. Show me the engagements that run for at least 3 days

Input

Select all columns of engagements from engagements table that run for at least 3 days //

translation

Select all columns from engagements where difference in StartDate and EndDate is 3 days //

clean up

SQL Query: SELECT * FROM engagements where EndDate-StartDate >=3

and StopTime>=StartTime

order by EngagementNumber;

Output

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane lists various databases including 'recipedb', 'sakila', 'salesorden', and 'schoolscheduli'. The main area displays a SQL query in a text editor:

```
1 # Select all columns of engagements from engagements table that run for at least 3 days // translation
2 # Select all columns from engagements where difference in StartDate and EndDate is 3 days // clean up
3
4 SELECT * FROM engagements where EndDate-StartDate >=3
5 and StopTime>=StartTime
6 order by EngagementNumber;
7
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has columns: EngagementNumber, StartDate, EndDate, StartTime, StopTime, ContractPrice, CustomerID, AgentID, and EntertainerID. The results are as follows:

EngagementNumber	StartDate	EndDate	StartTime	StopTime	ContractPrice	CustomerID	AgentID	EntertainerID
2	2012-09-01	2012-09-05	13:00:00	15:00:00	200.00	10006	4	1004
3	2012-09-10	2012-09-15	13:00:00	15:00:00	590.00	10001	3	1005
5	2012-09-11	2012-09-14	16:00:00	19:00:00	1130.00	10006	5	1003
6	2012-09-10	2012-09-14	15:00:00	21:00:00	2300.00	10014	7	1008
7	2012-09-11	2012-09-18	17:00:00	20:00:00	770.00	10004	4	1002
8	2012-09-18	2012-09-25	20:00:00	23:00:00	1850.00	10006	3	1007
9	2012-09-18	2012-09-28	19:00:00	21:00:00	1370.00	10010	2	1010
10	2012-09-17	2012-09-26	13:00:00	17:00:00	3650.00	10005	3	1003

At the bottom, an 'Output' pane shows a list of actions and their messages. The first action is successful, returning 5 rows. The second action is an error (Error Code: 1054, Unknown column 'StfFirstName' in 'order clause'). The third action is successful, returning 14 rows.

4. Provide a list of staff members and their salaries if they make at least \$35000 a year

Input

Select list of staff members and their salaries from staff table if they make at least \$35000 a year // **translation**

Select staffID, Stffirstname, Stflastname and salary from staff table where salary >=35000 //**cleanup**

SQL Query: select StaffID, StffirstName, Stflastname, Salary from staff where Salary >= 35000 order by StffirstName;

Output

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane lists databases: 'recipedb', 'sakila', 'salesorder', and 'schoolsch'. The 'schoolsch' database is selected, showing its tables and views. The main area displays a SQL query in a text editor:

```
1 # Select list of staff members and their salaries from staff table if they make at least $35000 a year // translation
2 # Select staffID, Stffirstname, Stflastname and salary from staff table where salary >=35000 //cleanup
3
4
5 • select StaffID, StffirstName, Stflastname, Salary from staff where Salary >= 35000 order by StffirstName;
6
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has columns: StaffID, StffirstName, Stflastname, and Salary. The results are as follows:

StaffID	StffirstName	Stflastname	Salary
98028	Alaina	Hallmark	57000.00
98055	Alastair	Black	60000.00
98011	Ann	Patterson	45000.00
98053	Caleb	Viescas	45000.00
98025	Carol	Viescas	50000.00
98062	Caroline	Cole	52000.00
98042	David	Smith	52000.00
98013	Deh	Waldal	44000.00

Below the result grid is an 'Output' pane showing the execution log. It contains three entries:

#	Time	Action	Message
20	16:11:38	SELECT * FROM engagements where EndDate-StartDate >=3 and StopTime>=StartTime or...	79 row(s) returned
21	16:12:58	select StaffID, StffirstName, Stflastname, Salary from staff where Salary >= 35000 order by St...	Error Code: 1146. Table 'entertainmentagencydb.staff' doesn't exist
22	16:13:13	select StaffID, StffirstName, Stflastname, Salary from staff where Salary >= 35000 order by St...	26 row(s) returned

5. Provide a list of recipes that contain eggs in their preparation

Input

Select list of recipes from recipes table that contain eggs in their preparation //translation

Select recipeID, recipetitle from recipes where preparation includes egg // cleanup

SQL Query: select RecipeID, RecipeTitle from recipes where upper(Preparation) like ("%EGGS%");

Output

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' panel lists databases including 'recipedb'. The main area displays a SQL query editor with the following text:

```
1 # Select list of recipes from recipes table that contain eggs in their preparation //translation
2 # Select recipeID, recipetitle from recipes where preparation includes egg // cleanup
3
4 • select RecipeID, RecipeTitle from recipes where upper(Preparation) like ("%EGGS%");
5
```

Below the query editor, a 'Result Grid' shows the results of the query. The grid has two columns: 'RecipeID' and 'RecipeTitle'. The first row shows '10' and 'Yorkshire Pudding'.

RecipeID	RecipeTitle
10	Yorkshire Pudding

At the bottom, an 'Output' panel shows a log of actions. The third action (23) is highlighted, showing the execution of the SQL query and the message '1 row(s) returned'.

#	Time	Action	Message
21	16:12:58	select StaffID, StfFirstName, Stflastname, Salary from staff where Salary >= 35000 order by St...	Error Code: 1146. Table 'entertainmentagencydb.staff' doesn't exist
22	16:13:13	select StaffID, StfFirstName, Stflastname, Salary from staff where Salary >= 35000 order by St...	26 row(s) returned
23	16:14:29	select RecipeID, RecipeTitle from recipes where upper(Preparation) like ("%EGGS%") LIMIT ...	1 row(s) returned