

POST-LAB

1. Suppose that a website contains two tables, the **Customers** table and the **Orders** table. Write a SQL query to find all customers who never order anything.

Table: **Customers**

Id	Name
1	Joe
2	Henry
3	Sam
4	Max

Table: **Orders**

Id	CustomerId
1	3
2	1

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, a toolbar with 'Limit to 1000 rows', and a 'SQLAdditions' panel on the right with a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

The SQL editor contains the following query:

```

1 • SELECT * FROM database1.customers;
2 • SELECT Name AS Customers FROM Customers WHERE Id NOT IN
3   (SELECT CustomerId FROM Orders);

```

Below the editor, the 'Result Grid' is displayed, showing the results of the query. The first result is 'Customers' with 4 rows: Henry, Max, and two empty rows. The second result is 'customers 3' with 2 rows: Henry and Max.

The 'Output' panel at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
304	15:13:59	SELECT * FROM database1.customers LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
305	15:13:59	SELECT Name AS Customers FROM Customers WHERE Id NOT IN (SELECT CustomerId F...	2 row(s) returned	0.078 sec / 0.000 sec

2. Given a **Weather** table, write a SQL query to find all dates' Ids with higher temperature compared to its previous (yesterday's) dates.

Id(INT)	RecordDate(DATE)	Temperature(INT)
1	01-01-2015	10
2	02-01-2015	25
3	03-01-2015	20
4	04-01-2015	30

The screenshot shows a SQL IDE interface with multiple tabs. The active tab is 'weather'. The SQL editor contains the following query:

```

1 • SELECT * FROM database2.weather;
2 • SELECT t.Id FROM Weather AS t, Weather AS y
3 WHERE DATEDIFF(t.RecordDate, y.RecordDate) = 1 AND t.Temperature > y.Temperature;
4 |

```

Below the editor, the 'Result Grid' is displayed, showing the results of the query:

Id
2
4

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
299	15:11:40	SELECT * FROM database2.weather LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
300	15:11:40	SELECT t.Id FROM Weather AS t, Weather AS y WHERE DATEDIFF(t.RecordDate, y.RecordDate) = 1 AND t.Temperature > y.Temperature	2 row(s) returned	0.031 sec / 0.000 sec

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

3. There is a table `World`

name	continent	area	population	gdp
Afghanistan	Asia	652230	25500100	20343000
Albania	Europe	28748	2831741	12960000
Algeria	Africa	2381741	37100000	188681000
Andorra	Europe	468	78115	3712000
Angola	Africa	1246700	20609294	100990000

A

country is big if it has an area of bigger than 3 million square km or a population of more than 25 million. Write a SQL solution to output big countries' name, population and area.

The screenshot shows a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query editor contains the following SQL query:

```
SELECT name, population, area FROM World WHERE area > 3000000 OR population > 25000000;
```

The results pane displays the output of the query in a table format:

name	population	area
Afghanistan	25500100	652230
Algeria	37100000	2381741

The results pane also shows the execution details of the query, including the time taken and the number of rows returned.

4.A city built a new stadium, each day many people visit it and the stats are saved as these columns: id, visit_date, people. Write a query to display the records which have 3 or more consecutive rows and the amount of people more than 100. Each day only have one row record, and the dates are increasing with id increasing. table stadium:

id	visit_date	people
1	01-01-2017	10
2	02-01-2017	109
3	03-01-2017	150
4	04-01-2017	99
5	05-01-2017	145
6	06-01-2017	1455
7	07-01-2017	199
8	08-01-2017	188

SQL File 44* SQL File 46* SQL File 47* SQL File 49* SQL File 50* SQL File 51* customers weather customers stadium

Limit to 1000 rows

```

1 • SELECT * FROM database4.stadium;
2 • select s1.* from stadium as s1, stadium as s2, stadium as s3
3   where
4   ((s1.id = s2.id - 1 and s1.id = s3.id - 2) or
5   (s1.id = s2.id + 1 and s1.id = s3.id - 1) or
6   (s1.id = s2.id + 2 and s1.id = s3.id + 1)) and
7   (s1.people >= 100 and s2.people >= 100 and s3.people >= 100)
8   group by s1.id;

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

id	visit_date	people
7	2017-01-07	199
5	2017-01-05	145
6	2017-01-06	1455
8	2017-01-08	188

stadium 2 Result 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
307	15:17:48	SELECT * FROM database4.stadium LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
308	15:17:49	select s1.* from stadium as s1, stadium as s2, stadium as s3 where ((s1.id = s2.id - 1 and s1.id ...	4 row(s) returned	0.000 sec / 0.000 sec

5.The **Employee** table holds all employees. Every employee has an Id, a salary, and there is also a column for the department Id. Write a SQL query to find employees who have the highest salary in each of the departments

Id	Name	Salary	DepartmentId
1	Joe	70000	1
2	Jim	90000	1
3	Henry	80000	2
4	Sam	60000	2
5	Max	90000	1

Department

Id	Name
1	IT
2	Sales

The screenshot shows the SQL Developer interface with a query window titled 'department'. The query is as follows:

```

1 • SELECT * FROM database5.department;
2 • SELECT d.Name AS Department, e.Name AS Employee, e.Salary
3 FROM Employee AS e, Department AS d
4 WHERE e.DepartmentId = d.Id AND
5 e.Salary = (SELECT MAX(Salary) FROM Employee AS t WHERE t.DepartmentId = e.DepartmentId);

```

The results are displayed in the 'Result Grid' tab, showing the following data:

Department	Employee	Salary
IT	Jim	90000
Sales	Henry	80000
IT	Max	90000

The 'Output' window shows the execution of the query, indicating that 2 row(s) were returned for the first statement and 3 row(s) were returned for the second statement.