

# Data Analysis project on diabetes in women in india using SQL



- Previewing all the data:

The screenshot shows a database management interface with a left sidebar and a main workspace. The sidebar, titled 'SCHEMAS', lists databases: diabetes, new\_schema, sakila, sys, and world. The 'diabetes' database is selected, showing its structure (Tables, Views, Stored Procedures, Functions). Below the sidebar, there are tabs for 'Administration' and 'Schemas', and a status message 'No object selected'.

The main workspace contains a SQL editor with the following code:

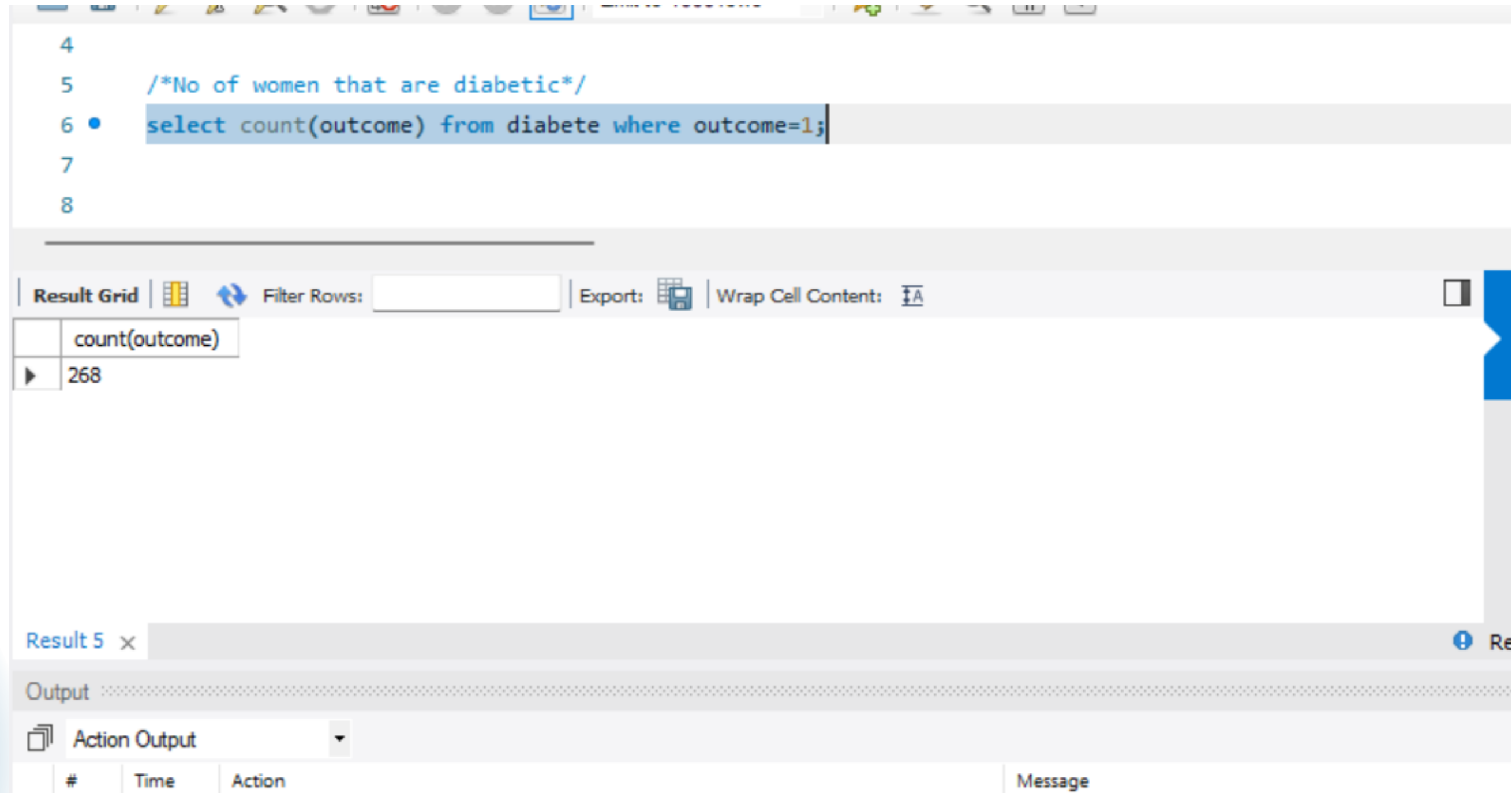
```
1  /*previewing all the data*/
2  • select * from diabete;
3
4
5  /*No of women that are diabetic*/
```

Below the editor is a 'Result Grid' showing the results of the query. The grid has columns: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, and Outcome. The results are as follows:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
▶	6	148	72	35	0	33.6	0.627	50	1
	1	85	66	29	0	26.6	0.351	31	0
	8	183	64	0	0	23.3	0.672	32	1
	1	89	66	23	94	28.1	0.167	21	0
	0	137	40	35	168	43.1	2.288	33	1
	5	116	74	0	0	25.6	0.201	30	0
	3	78	50	32	88	31	0.248	26	1
	10	115	0	0	0	35.3	0.134	29	0
	2	107	70	45	542	30.5	0.150	52	1

Below the result grid, there is a tab labeled 'diabete 4 x' and an 'Output' section. At the bottom, there is an 'Action Output' dropdown menu and a table with columns: #, Time, Action, and Message.

- No of women that are diabetic:



The screenshot shows a SQL IDE interface. The top pane contains a SQL query: `/*No of women that are diabetic*/` followed by `select count(outcome) from diabete where outcome=1;`. The bottom pane displays the result of the query in a table with one row and one column, showing the value 268. The interface includes a toolbar with icons for filtering, exporting, and wrapping cell content. The bottom status bar shows the output of the action.

```
4
5  /*No of women that are diabetic*/
6  • select count(outcome) from diabete where outcome=1;
7
8
```

count(outcome)
268

Result 5 x

Output

Action Output

#	Time	Action	Message
---	------	--------	---------

- No of women that are Non-diabetic:

The screenshot shows a SQL IDE interface with a script editor and a results pane. The script editor contains a comment and a SQL query. The results pane shows a single row with the count of non-diabetic women.

**SQL Script:**

```
7  
8  
9  /*No of women that are Non-diabetic*/  
10 • select count(outcome) from diabete where outcome=0;  
11
```

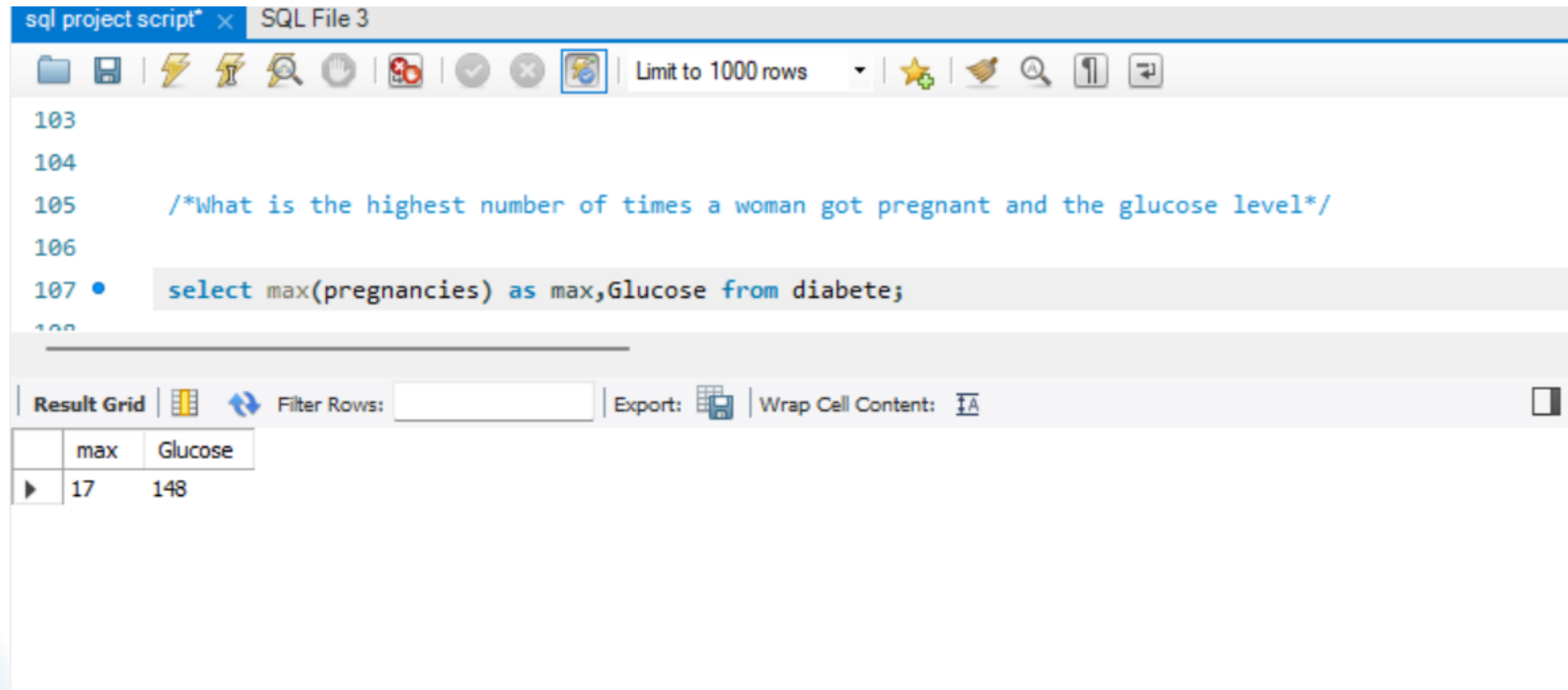
**Result Grid:**

count(outcome)
500

**Output:**

#	Time	Action	Message
1	11:18:15	select * from diabete LIMIT 0, 1000	768 row(s) returned
2	11:18:31	select * from diabete LIMIT 0, 1000	768 row(s) returned

- What is the highest number of times a woman got pregnant and the glucose level



The screenshot shows a SQL IDE window titled "SQL File 3". The editor contains a SQL query with a comment and a SELECT statement. The query is as follows:

```
103  
104  
105      /*What is the highest number of times a woman got pregnant and the glucose level*/  
106  
107 •   select max(pregnancies) as max,Glucose from diabete;  
108
```

Below the editor, the "Result Grid" tab is active, displaying the query results in a table. The table has two columns: "max" and "Glucose". The first row shows the values "17" and "148".

	max	Glucose
▶	17	148

- Alter table in order to update new column of result instead of outcome(int) to result(varchar)

The screenshot shows a SQL IDE window titled "SQL File 3". The script contains the following SQL statements:

```
20  /* alter table in order to update new column of result instead of outcome(int) to result(varchar)*/  
21  • alter table diabete add column result varchar(20);  
22  • update diabete set result = case  
23      when outcome=1 then "diabetic"  
24      else "non-diabetic"  
25  end;
```

Below the script, the "Result Grid" is displayed, showing the results of the update operation. The grid has a single column labeled "result". The data is as follows:

result
diabetic
non-diabetic
diabetic
non-diabetic
diabetic
non-diabetic
diabetic
non-diabetic
diabetic

The IDE interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Read Only" status indicator at the bottom right.



- To ascertain the relationship between blood pressure and diabetes

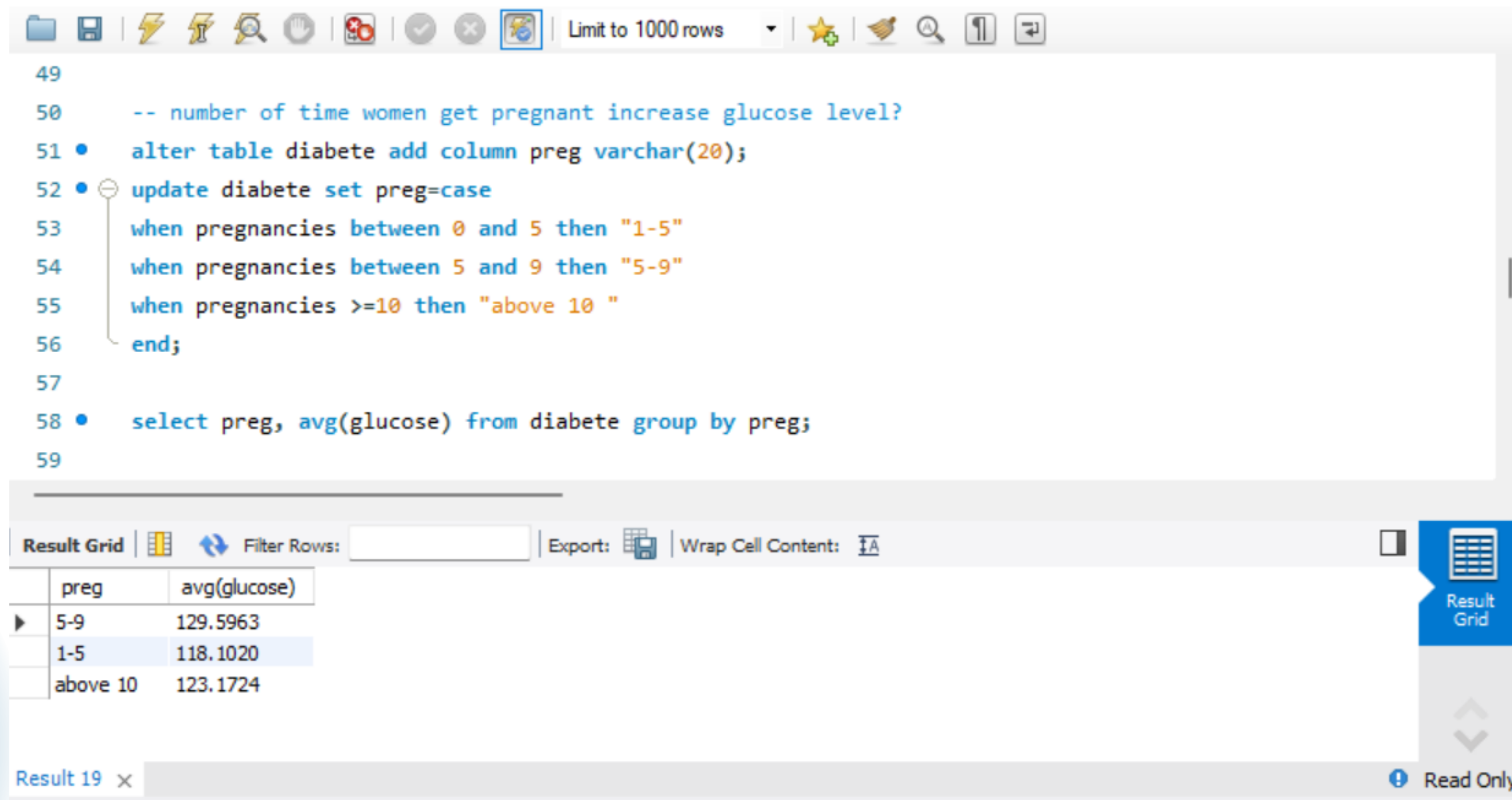
The screenshot shows a SQL IDE window titled "SQL File 3". The script contains the following SQL statements:

```
29  /*to ascertain the relationship between blood pressure and diabetes*/
30  • alter table diabete add column diastolic_blood_pressure varchar(25);
31  • update diabete set diastolic_blood_pressure= CASE
32      WHEN bloodpressure between 60 and 80 THEN 'normal'
33      WHEN bloodpressure BETWEEN 80 AND 89 THEN 'stage_1_hypertension'
34      WHEN bloodpressure BETWEEN 90 AND 120 THEN 'stage_2_hypertension'
35      WHEN bloodpressure >= 120 THEN 'hypertensive_crisis'
36      when bloodpressure <60 then "low hypertension"
37  END;
38  • select diastolic_blood_pressure, count(result) as total
39  from diabete where result="diabetic" group by diastolic_blood_pressure;
```

Below the script, the "Result Grid" is displayed, showing the following data:

diastolic_blood_pressure	total
normal	159
low hypertension	32
stage_2_hypertension	29
stage_1_hypertension	48

- Number of time women get pregnant increase glucose level?



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
49
50 -- number of time women get pregnant increase glucose level?
51 • alter table diabetes add column preg varchar(20);
52 • update diabetes set preg=case
53   when pregnancies between 0 and 5 then "1-5"
54   when pregnancies between 5 and 9 then "5-9"
55   when pregnancies >=10 then "above 10 "
56 end;
57
58 • select preg, avg(glucose) from diabetes group by preg;
59
```

Below the editor is the 'Result Grid' section. It includes a 'Filter Rows' input, 'Export' and 'Wrap Cell Content' buttons, and a 'Result Grid' button. The results are displayed in a table:

preg	avg(glucose)
5-9	129.5963
1-5	118.1020
above 10	123.1724

At the bottom, there is a tab labeled 'Result 19' and a 'Read Only' status indicator.



- To ascertain the relationship between the body mass index and diabetes

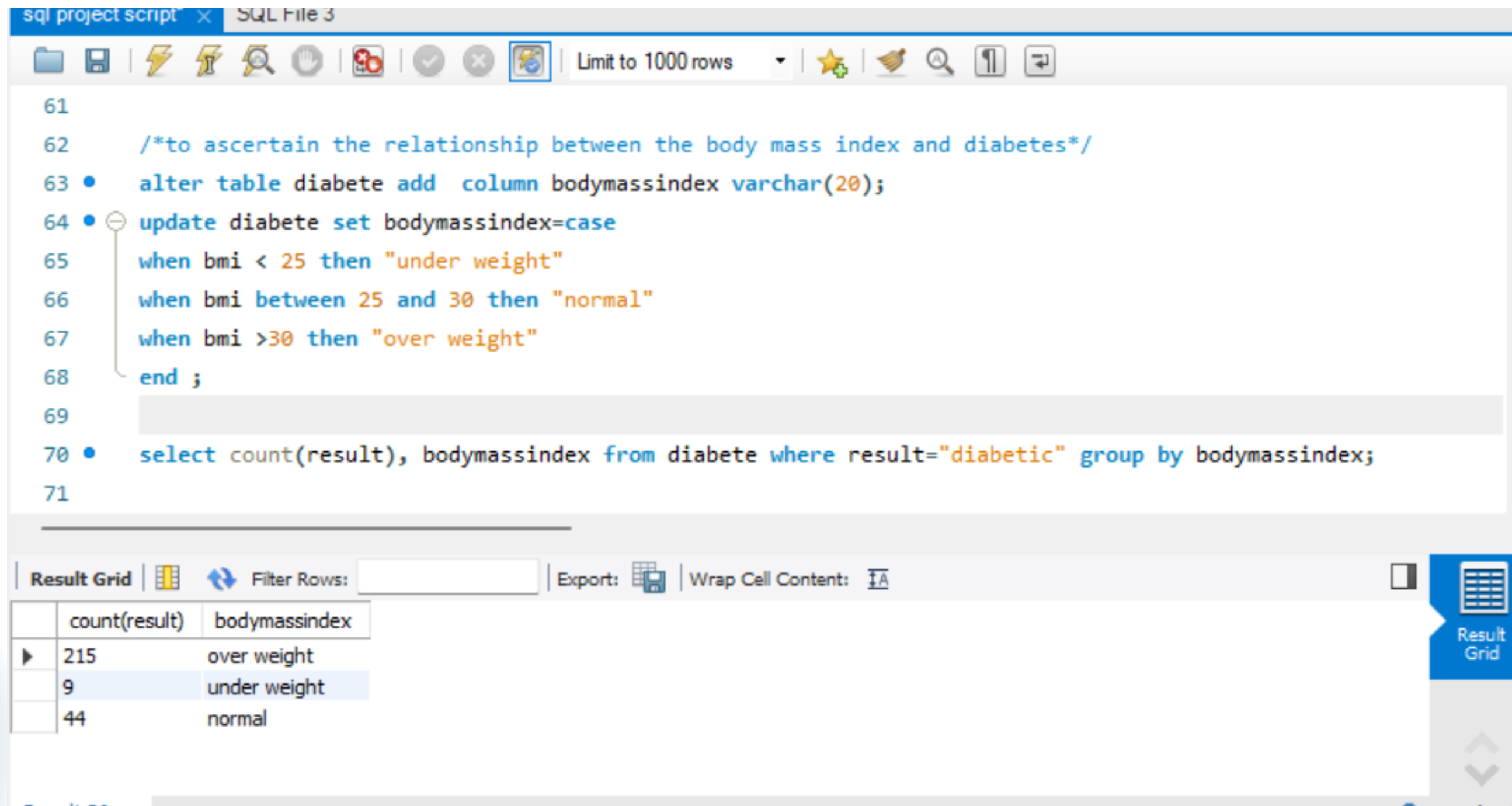
The screenshot shows a SQL IDE window with a script titled "SQL File 3". The script contains the following SQL statements:

```
61
62  /*to ascertain the relationship between the body mass index and diabetes*/
63 • alter table diabete add column bodymassindex varchar(20);
64 • update diabete set bodymassindex=case
65   when bmi < 25 then "under weight"
66   when bmi between 25 and 30 then "normal"
67   when bmi >30 then "over weight"
68 end ;
69
70 • select count(result), bodymassindex from diabete where result="diabetic" group by bodymassindex;
71
```

Below the script, the "Result Grid" is displayed, showing the results of the last query. The grid has two columns: "count(result)" and "bodymassindex".

count(result)	bodymassindex
215	over weight
9	under weight
44	normal

- To ascertain the relationship between age\_range and diabetes



The screenshot shows a SQL IDE window with a script editor and a result grid. The script editor contains the following SQL code:

```
61
62  /*to ascertain the relationship between the body mass index and diabetes*/
63 • alter table diabete add column bodymassindex varchar(20);
64 • update diabete set bodymassindex=case
65   when bmi < 25 then "under weight"
66   when bmi between 25 and 30 then "normal"
67   when bmi >30 then "over weight"
68 end ;
69
70 • select count(result), bodymassindex from diabete where result="diabetic" group by bodymassindex;
71
```

The result grid shows the following data:

	count(result)	bodymassindex
▶	215	over weight
	9	under weight
	44	normal

