

```

98
99
100 • UPDATE students SET phone_number = '7777777777' WHERE student_id = 1;
101
102 • DELETE FROM students WHERE student_id = 3;
103
104
105
106 • SELECT * FROM students WHERE department_id = 1;
107

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
▶	1	Rahul	2002-05-10	Male	NULL	7777777777	Delhi	2021-07-10	1
	2	Anjali	2001-08-15	Female	anjali@gmail.com	9123456789	Jaipur	2020-07-10	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Limit to 1000 rows

```

104
105
106 • SELECT * FROM students WHERE department_id = 1;
107
108
109 • SELECT * FROM grades ORDER BY marks_obtained DESC LIMIT 10;
110
111
112
113

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	grade_id	student_id	course_id	marks_obtained
▶	1	1	1	95
	2	2	1	70
*	NULL	NULL	NULL	NULL

grades 2 x Apply

Limit to 1000 rows

```

110
111
112 • SELECT student_id,
113      COUNT(CASE WHEN status='Present' THEN 1 END)/COUNT(*)*100 AS attendance_percentage
114 FROM attendance
115 GROUP BY student_id
116 HAVING attendance_percentage < 75;
117
118
119

```

Result Grid

	student_id	attendance_percentage
▶ 1		50.0000

Filter Rows: Export: Wrap Cell Content:

Result 3 × Read Only

Output

#	Time	Action	Message
✓ 20	17:14:23	SELECT * FROM students WHERE department_id = 1 LIMIT 0, 1000	2 row(s) returned
✓ 21	17:15:36	SELECT * FROM grades ORDER BY marks_obtained DESC LIMIT 10	2 row(s) returned
✓ 22	17:16:11	SELECT student_id, COUNT(CASE WHEN status='Present' THEN 1 END)/COUN...	1 row(s) returned

Limit to 1000 rows

```

116     HAVING attendance_percentage < 75;
117
118
119 •   SELECT * FROM grades
120     WHERE marks_obtained < 40 AND student_id IN (
121         SELECT student_id FROM attendance
122         GROUP BY student_id
123         HAVING COUNT(CASE WHEN status='Present' THEN 1 END)/COUNT(*)*100 < 50
124     );
125

```

Result Grid

grade_id	student_id	course_id	marks_obtained
NULL	NULL	NULL	NULL

grades 4 x

Output:

Action Output

#	Time	Action	Message
21	17:15:36	SELECT * FROM grades ORDER BY marks_obtained DESC LIMIT 10	2 row(s) returned
22	17:16:11	SELECT student_id, COUNT(CASE WHEN status='Present' THEN 1 END)/COUN...	1 row(s) returned
23	17:16:57	SELECT * FROM grades WHERE marks_obtained < 40 AND student_id IN (SE...	0 row(s) returned

SQL File 6" SQL File 8" DataDigger DATADigger DATATransformer Final"

Limit to 1000 rows

```

131     HAVING COUNT(CASE WHEN status='Present' THEN 1 END) = COUNT(*)
132 ); Execute the selected portion of the script or everything, if there is no selection
133
134
135 • SELECT * FROM faculty
136 WHERE faculty_id NOT IN (SELECT faculty_id FROM courses WHERE faculty_id IS NOT NULL);
137
138
139
140

```

Result Grid

	faculty_id	name	email	phone_number	department_id	experience
▶	2	Mr Verma	verma@gmail.com	8888888888	2	3
*	NULL	NULL	NULL	NULL	NULL	NULL

faculty 6 x

Output

Action Output

#	Time	Action	Message
✓ 23	17:16:57	SELECT * FROM grades WHERE marks_obtained < 40 AND student_id IN (SE...	0 row(s) returned
✓ 24	17:17:33	SELECT * FROM grades WHERE marks_obtained > 90 OR student_id IN (SEL...	2 row(s) returned
✓ 25	17:18:07	SELECT * FROM faculty WHERE faculty_id NOT IN (SELECT faculty_id FROM co...	1 row(s) returned

32.32.1SQL File 6*SQL File 8*DataDiggerDATADiggerDATATransformerFinal*SQL File

Limit to 1000 rows

137
138
139
140 • `SELECT c.*`
141 `FROM faculty f`
142 `RIGHT JOIN courses c ON f.faculty_id = c.faculty_id`
143 `WHERE f.faculty_id IS NULL;`
144
145
146

Result Grid
Filter Rows:
Export:
Wrap Cell Content:

	course_id	course_name	faculty_id
▶ 2	Thermodynamics	NULL	

Result 7 ×
Read

Output

Action Output

#	Time	Action	Message
✓ 24	17:17:33	SELECT * FROM grades WHERE marks_obtained > 90 OR student_id IN (SEL...	2 row(s) returned
✓ 25	17:18:07	SELECT * FROM faculty WHERE faculty_id NOT IN (SELECT faculty_id FROM co...	1 row(s) returned
✓ 26	17:19:06	SELECT c.* FROM faculty f RIGHT JOIN courses c ON f.faculty_id = c.faculty_id ...	1 row(s) returned

Limit to 1000 rows

143 WHERE f.faculty_id IS NULL;

144

145

146 • SELECT s.*

147 FROM students s

148 LEFT JOIN grades g ON s.student_id = g.student_id

149 WHERE g.student_id IS NULL;

150

151

152

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
------------	------	-----	--------	-------	--------------	---------	----------------	---------------

Result 8 x

Output

Action Output

#	Time	Action	Message
✓ 25	17:18:07	SELECT * FROM faculty WHERE faculty_id NOT IN (SELECT faculty_id FROM co...	1 row(s) returned
✓ 26	17:19:06	SELECT c.* FROM faculty f RIGHT JOIN courses c ON f.faculty_id = c.faculty_id ...	1 row(s) returned
✓ 27	17:20:26	SELECT s.* FROM students s LEFT JOIN grades g ON s.student_id = g.student_id ...	0 row(s) returned

SQL File 0 SQL File 0 DataTrigger DATATrigger DATA

Limit to 1000 rows

```
149 WHERE g.student_id IS NULL;
150
151
152 • SELECT * FROM grades
153 WHERE marks_obtained > (SELECT AVG(marks_obtained) FROM grades);
154
155
156
157
158 • SELECT * FROM courses
```

<

Result Grid

Filter Rows:

Edit: Export/Import:

	grade_id	student_id	course_id	marks_obtained
▶	1	1	1	95
*	NULL	NULL	NULL	NULL

grades 9 x

Output

3

2.1

SQL File 6*

SQL File 8*

DataDigger

DATADigger

DATATransformer

Final

Limit to 1000 rows

152 •

SELECT * FROM grades

153

WHERE marks_obtained > (SELECT AVG(marks_obtained) FROM grades);

154

155 •

SELECT * FROM courses

156

WHERE faculty_id IN (

157

SELECT faculty_id FROM faculty WHERE experience >= 5

158

);

159

160

161 •

SELECT student id FROM attendance

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	course_id	course_name	faculty_id
▶	1	Database Systems	1
*	NULL	NULL	NULL

courses 11 ×

Output

Action Output

#	Time	Action	Message
✓ 28	17:20:59	SELECT * FROM grades WHERE marks_obtained > (SELECT AVG(marks_obtaine...	1 row(s) returned
✓ 29	17:22:46	SELECT * FROM grades WHERE marks_obtained > (SELECT AVG(marks_obtaine...	1 row(s) returned
✓ 30	17:23:02	SELECT * FROM courses WHERE faculty_id IN (SELECT faculty_id FROM fac...	1 row(s) returned

1.3
2.1
SQL File 6*
SQL File 8*
DataDigger
DATADigger
DATATransformer
Final*

Limit to 1000 rows

```

158 );
159
160
161 • SELECT student_id FROM attendance
162 WHERE status = 'Absent'
163 GROUP BY student_id
164 HAVING COUNT(*) > 10;
165
166
167

```

Result Grid
Filter Rows:
Export:
Wrap Cell Content:

student_id

attendance 12

Output

Action Output

#	Time	Action	Message
✓ 29	17:22:46	SELECT * FROM grades WHERE marks_obtained > (SELECT AVG(marks_obtaine...	1 row(s) returned
✓ 30	17:23:02	SELECT * FROM courses WHERE faculty_id IN (SELECT faculty_id FROM fac...	1 row(s) returned
✓ 31	17:23:45	SELECT student_id FROM attendance WHERE status = 'Absent' GROUP BY stud...	0 row(s) returned

SQL File 6* SQL File 8* DataDigger DATADigger DATATransformer Final* SQL F

Limit to 1000 rows

```

161 • SELECT student_id FROM attendance
162   Execute the selected portion of the script or everything, if there is no selection
163   GROUP BY student_id
164   HAVING COUNT(*) > 10;
165
166
167 • SELECT MONTH(attendance_date) AS month FROM attendance;
168
169
170

```

Result Grid

month
1
1
1

Result 13 x

Output

Action Output

#	Time	Action	Message
✓ 30	17:23:02	SELECT * FROM courses WHERE faculty_id IN (SELECT faculty_id FROM fac...	1 row(s) returned
✓ 31	17:23:45	SELECT student_id FROM attendance WHERE status = 'Absent' GROUP BY stud...	0 row(s) returned
✓ 32	17:24:08	SELECT MONTH(attendance_date) AS month FROM attendance LIMIT 0, 1000	3 row(s) returned

SQL Editor interface showing SQL queries and their results.

SQL Editor:

```
167 • SELECT MONTH(attendance_date) AS month FROM attendance;
168
169
170 • SELECT name, TIMESTAMPDIFF(YEAR, admission_date, CURDATE()) AS years
171 FROM students;
172
173
174
175
176
```

Result Grid:

	name	years
▶	Rahul	4
	Anjali	5

Result 14 × [Read On](#)

Output:

Action Output

#	Time	Action	Message
✓ 31	17:23:45	SELECT student_id FROM attendance WHERE status = 'Absent' GROUP BY stud...	0 row(s) returned
✓ 32	17:24:08	SELECT MONTH(attendance_date) AS month FROM attendance LIMIT 0, 1000	3 row(s) returned
✓ 33	17:24:34	SELECT name, TIMESTAMPDIFF(YEAR, admission_date, CURDATE()) AS years F...	2 row(s) returned

Limit to 1000 rows

SQL Editor

167 •

SELECT MONTH(attendance_date) AS month FROM attendance;

168

169

170 •

SELECT name, TIMESTAMPDIFF(YEAR, admission_date, CURDATE()) AS years

171 FROM students;

172

173

174 •

SELECT DATE_FORMAT(attendance_date, '%d-%m-%Y') FROM attendance;

175

176

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	DATE_FORMAT(attendance_date, '%d-%m-%Y')
▶	01-01-2024
	02-01-2024
	01-01-2024

Result 15

×

!

R

Output

⌵

Action Output

#	Time	Action	Message
✓ 32	17:24:08	SELECT MONTH(attendance_date) AS month FROM attendance LIMIT 0, 1000	3 row(s) returned
✓ 33	17:24:34	SELECT name, TIMESTAMPDIFF(YEAR, admission_date, CURDATE()) AS years F...	2 row(s) returned
✓ 34	17:25:08	SELECT DATE_FORMAT(attendance_date, '%d-%m-%Y') FROM attendance LIMIT...	3 row(s) returned

SQL File 6* SQL File 8* DataDigger DATADigger DATATransformer Final*

Limit to 1000 rows

```

179 • SELECT TRIM(name) FROM students;
180
181
182 • SELECT name, IFNULL(email, 'Email Not Provided') FROM students;
183
184
185 • SELECT student_id, marks_obtained,
186 RANK() OVER (ORDER BY marks_obtained DESC) AS rank_no
187 FROM grades;
188

```

Execute the selected portion of the script or everything, if there is no selection

Result Grid

	student_id	marks_obtained	rank_no
▶	1	95	1
	2	70	2

Result 19 x

Output

Action Output

#	Time	Action	Message
✓ 36	17:26:06	SELECT TRIM(name) FROM students LIMIT 0, 1000	2 row(s) returned
✓ 37	17:26:34	SELECT name, IFNULL(email, 'Email Not Provided') FROM students LIMIT 0, 1000	2 row(s) returned
✓ 38	17:27:07	SELECT student_id, marks_obtained, RANK() OVER (ORDER BY marks_obtained ...	2 row(s) returned

SQL File 6* SQL File 8* DataDigger DATADigger DATAtransformer Final* SQL File 12*

Limit to 1000 rows

188

189 Execute the selected portion of the script or everything, if there is no selection

190

191 • SELECT student_id,

192 SUM(CASE WHEN status='Present' THEN 1 ELSE 0 END) OVER (PARTITION BY student_id) AS total_present

193 FROM attendance;

194

195

196

197

Result Grid

	student_id	total_present
▶	1	1
	1	1
	2	1

Export: Wrap Cell Content:

Result 21 × Read Only

Output

Action Output

#	Time	Action	Message
✓ 38	17:27:07	SELECT student_id, marks_obtained, RANK() OVER (ORDER BY marks_obtained ...	2 row(s) returned
✓ 39	17:30:39	SELECT student_id, marks_obtained, RANK() OVER (ORDER BY marks_obtained ...	2 row(s) returned
✓ 40	17:30:44	SELECT student_id, SUM(CASE WHEN status='Present' THEN 1 ELSE 0 END) O...	3 row(s) returned

1.32.1SQL File 6*SQL File 8*DataDiggerDATADiggerDATATransformerFinal*

Limit to 1000 rows

```

190
191 • SELECT student_id,
192     SUM(CASE WHEN status='Present' THEN 1 ELSE 0 END) OVER (PARTITION BY student_id) AS t
193 FROM attendance;
194
195 • SELECT enrollment_date,
196     COUNT(*) OVER (ORDER BY enrollment_date) AS running_total
197 FROM enrollments;
198
199

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	enrollment_date	running_total
▶	2023-01-10	2
	2023-01-10	2

Result 22 ×

Output

Action Output

#	Time	Action	Message
✓ 39	17:30:39	SELECT student_id, marks_obtained, RANK() OVER (ORDER BY marks_obtained ...	2 row(s) returned
✓ 40	17:30:44	SELECT student_id, SUM(CASE WHEN status='Present' THEN 1 ELSE 0 END) O...	3 row(s) returned
✓ 41	17:31:17	SELECT enrollment_date, COUNT(*) OVER (ORDER BY enrollment_date) AS runni...	2 row(s) returned

1.32.1SQL File 6*SQL File 8*DataDiggerDATADiggerDATATransformerFinal*

Limit to 1000 rows

199

200 • SELECT student_id, marks_obtained,

201 CASE

202 WHEN marks_obtained > 90 THEN 'Excellent'

203 WHEN marks_obtained BETWEEN 75 AND 90 THEN 'Good'

204 ELSE 'Needs Improvement'

205 END AS performance

206 FROM grades;

207

208

<

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	student_id	marks_obtained	performance
▶	1	95	Excellent
	2	70	Needs Improvement

Result 23 ×

Output

Action Output

	#	Time	Action	Message
✓	40	17:30:44	SELECT student_id, SUM(CASE WHEN status='Present' THEN 1 ELSE 0 END) O...	3 row(s) returned
✓	41	17:31:17	SELECT enrollment_date, COUNT(*) OVER (ORDER BY enrollment_date) AS runni...	2 row(s) returned
✓	42	17:33:08	SELECT student_id, marks_obtained, CASE WHEN marks_obtained > 90 THEN...	2 row(s) returned

1.32.1SQL File 6*SQL File 8*DataDiggerDATADiggerDATATransformer

Limit to 1000 rows

200 • SELECT student_id, marks_obtained,

201 CASE

202 WHEN marks_obtained > 90 THEN 'Excellent'

203 WHEN marks_obtained BETWEEN 75 AND 90 THEN 'Good'

204 ELSE 'Needs Improvement'

205 END AS performance

206 FROM grades;

207

208 • SELECT student_id,

209 CASE

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	student_id	attendance_status
▶	1	Irregular
	2	Regular

Result 24 ×

Output

Action Output

#	Time	Action	Message
✓ 41	17:31:17	SELECT enrollment_date, COUNT(*) OVER (ORDER BY enrollment_date) AS runni...	2 row(s) returned