

FSDA ASSIGNMENT-1

TASK-1

- ✓ Creating a new warehouse (ASSIGNMENT_1).

ACCOUNTADMIN • ASSIGNMENT_1 Share Updated 6 minutes ago

```
SQL_PROJECT.PUBLIC
1 CREATE DATABASE SQL_PROJECT;
2
3 USE SQL_PROJECT;
4 SHOW DATABASES;
5 SHOW TABLES;
6
7 -- Task 1
8 CREATE TABLE IF NOT EXISTS shopping_history
9 (
10     product VARCHAR(50) NOT NULL,
11     quantity INT NOT NULL,
12     unit_price INT NOT NULL
13 );
14
15 INSERT INTO shopping_history VALUES('milk', 3, 10);
16 INSERT INTO shopping_history VALUES('bread', 7, 3);
17 INSERT INTO shopping_history VALUES('bread', 5, 2);
18 INSERT INTO shopping_history VALUES('cakes', 10, 20);
19 INSERT INTO shopping_history VALUES('pizza', 2, 100);
20 INSERT INTO shopping_history VALUES('biscuit', 20, 5);
21 INSERT INTO shopping_history VALUES('cakes', 5, 40);
22 INSERT INTO shopping_history VALUES('muffins', 4, 80);
23 INSERT INTO shopping_history VALUES('pizza', 3, 150);
24 INSERT INTO shopping_history VALUES('cakes', 1, 300);
25
26
27 SELECT * FROM shopping_history;
28
```

ACCOUNTADMIN • ASSIGNMENT_1 Share Updated 26 seconds ago

SQL_PROJECT.PUBLIC

```
25
26
27 SELECT * FROM shopping_history;
28
29 SELECT PRODUCT, SUM(total_price) FROM
30 (SELECT PRODUCT, (QUANTITY * UNIT_PRICE) AS total_price FROM shopping_history) GROUP BY PRODUCT ORDER BY PRODUCT DESC;
31
32
```

Objects Editor Results Chart

	PRODUCT	SUM(TOTAL_PRICE)
1	pizza	650
2	muffins	320
3	milk	30
4	cakes	700
5	bread	31
6	biscuit	100

Query Details
Query duration 2.3s
Rows 6
PRODUCT Aa
100% filled

ACCOUNTADMIN • ASSIGNMENT_1 Share Updated 2 minutes ago

```
SQL_PROJECT.PUBLIC
32
33 -- Task2
34
35 CREATE OR REPLACE TABLE phones
36 (
37     name VARCHAR(50) NOT NULL UNIQUE,
38     phone_number INT NOT NULL UNIQUE
39 );
40
41 CREATE OR REPLACE TABLE calls
42 (
43     id INT NOT NULL,
44     caller INT NOT NULL,
45     callee INT NOT NULL,
46     duration INT NOT NULL,
47     UNIQUE(id)
48 );
49
```

```
SQL_PROJECT.PUBLIC
49
50
51 -- Phone numbers are randomly entered
52 INSERT INTO phones VALUES
53 ('Prashantha', 9738964435),
54 ('Sabari', 9380969613),
55 ('Naveen', 8095578486),
56 ('Shubhash', 6280345975),
57 ('Venkatesh', 9948756245),
58 ('Ramesh', 7845258965),
59 ('Suresh', 8578964572),
60 ('Sathish', 9875486215),
61 ('Sudeep', 6548795412),
62 ('Darshan', 9852156324);
63
64 SELECT * FROM phones;
65
66 INSERT INTO calls VALUES
67 (101, 9738964435, 9380969613, 5),
68 (102, 8095578486, 6280345975, 3),
69 (103, 9948756245, 7845258965, 4),
70 (104, 8578964572, 9875486215, 7),
71 (105, 6548795412, 9852156324, 6),
72 (106, 9738964435, 6548795412, 4),
73 (107, 9380969613, 8578964572, 6),
74 (108, 9738964435, 9875486215, 2),
75 (109, 9852156324, 8095578486, 2),
76 (110, 9852156324, 6280345975, 2);
--
```

```
SQL_PROJECT.PUBLIC
77
78
79 SELECT * FROM calls;
80
81
82 SELECT name FROM
83 (SELECT p.name, p.phone_number, c.total_duration, c.caller
84 FROM phones p
85 LEFT OUTER JOIN
86 (SELECT sum(duration) AS total_duration, caller FROM
87 (SELECT SUM(duration) AS duration, caller FROM calls GROUP BY caller
88 UNION ALL
89 SELECT SUM(duration) AS duration, callee FROM calls GROUP BY callee) AS NEW_TABLE GROUP BY caller) AS c
90 ON p.phone_number = c.caller) WHERE total_duration >= 10 ORDER BY name;
91
92
```

Objects

Editor

Results

Chart

	NAME	
1	Darshan	
2	Prashantha	
3	Sabari	
4	Sudeep	
5	Suresh	

Query Details

Query duration327ms

Rows5

NAMEAs

100% filled

Updated 1 minute ago

```
SQL_PROJECT.PUBLIC
92
93 -- TASK 3
94 CREATE TABLE IF NOT EXISTS transaction
95 (
96     amount INTEGER NOT NULL,
97     date DATE NOT NULL
98 );
99
100 INSERT INTO transaction
101 VALUES (1000, '2020-01-06'),
102 (-10, '2020-01-14'),
103 (-75, '2020-01-20'),
104 (-5, '2020-01-25'),
105 (-4, '2020-01-29'),
106 (2000, '2020-03-10'),
107 (-75, '2020-03-12'),
108 (-20, '2020-03-15'),
109 (40, '2020-03-15'),
110 (-50, '2020-03-17'),
111 (200, '2020-10-10'),
112 (-200, '2020-10-10');
113
114 SELECT * FROM transaction;
115
```

Updated 14 seconds ago

```
SQL_PROJECT.PUBLIC
112
113
114 SELECT * FROM transaction;
115
116
117 SELECT sum(Balance) AS balance FROM
118 (SELECT SUM(amount) AS Balance FROM transaction
119 UNION
120 SELECT ((12 - no_month) * -5) AS credit_card_charge FROM
121 (SELECT COUNT(month) AS no_month FROM
122 (SELECT s.month, s.sum_of_months, c.no_transaction FROM
123 (SELECT MONTH(date) AS month, sum(iff(amount < 0, amount, 0)) AS sum_of_months FROM transaction GROUP BY month) AS s
124 LEFT OUTER JOIN
125 (SELECT MONTH(date) AS month, count(amount) AS no_transaction FROM transaction WHERE amount < 0 GROUP BY month) AS C
126 ON s.month = c.month) AS no_month_satisfied_both_cond WHERE no_transaction >= 3 AND sum_of_months >= -100) AS credit_card_charge_table) AS final;
127
```

Objects	Editor	Results	Chart
...		BALANCE	Query Details
1		2,746	Query duration 331ms

Updated 1 minute ago

```
SQL_PROJECT.PUBLIC
127
128
129 -- TASK-3 EXAMPLE-2
130 CREATE TABLE IF NOT EXISTS transaction1
131 (
132     amount INTEGER NOT NULL,
133     date DATE NOT NULL
134 );
135
136 INSERT INTO transaction1
137 VALUES
138 (1, '2020-06-29'),
139 (35, '2020-02-20'),
140 (-50, '2020-02-02'),
141 (-1, '2020-02-26'),
142 (-200, '2020-08-01'),
143 (-44, '2020-02-07'),
144 (-5, '2020-02-25'),
145 (1, '2020-06-29'),
146 (1, '2020-06-29'),
147 (-100, '2020-12-29'),
148 (-100, '2020-12-30'),
149 (-100, '2020-12-31');
150
```

Updated 21 seconds ago

```
SQL_PROJECT.PUBLIC
150
151 SELECT sum(Balance) AS balance FROM
152 (SELECT SUM(amount) AS Balance FROM transaction1
153 UNION
154 SELECT ((12 - no_month) * -5) AS credit_card_charge FROM
155 (SELECT COUNT(month) AS no_month FROM
156 (SELECT s.month, s.sum_of_months, c.no_transaction FROM
157 (SELECT MONTH(date) AS month, sum(iff(amount < 0, amount, 0)) AS sum_of_months FROM transaction1 GROUP BY month) AS s
158 LEFT OUTER JOIN
159 (SELECT MONTH(date) AS month, count(amount) AS no_transaction FROM transaction1 WHERE amount < 0 GROUP BY month) AS C
160 ON s.month = c.month) AS no_month_satisfied_both_cond WHERE no_transaction >= 3 AND sum_of_months <= -100) AS credit_card_charge_table) AS final;
161
```

Objects	Editor	Results	Chart
...		BALANCE	Query Details
1		-612	Query duration 60ms
		Rows	1

SQL_PROJECT.PUBLIC

161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
--

-- TASK-3 EXAMPLE-3
CREATE TABLE IF NOT EXISTS transaction2
(
 amount INTEGER NOT NULL,
 date DATE NOT NULL
);

INSERT INTO transaction2
VALUES
(6000, '2020-04-03'),
(5000, '2020-04-02'),
(4000, '2020-04-01'),
(3000, '2020-03-01'),
(2000, '2020-02-01'),
(1000, '2020-01-01');

SELECT * FROM transaction2;

ACCOUNTADMIN • ASSIGNMENT_1

Share

Updated 29 seconds ago

Draft

SQL_PROJECT.PUBLIC

178
179
180
181
182
183
184
185
186
187
188
189
190
191
--

SELECT * FROM transaction2;

SELECT sum(Balance) AS balance FROM
(SELECT SUM(amount) AS Balance FROM transaction2
UNION
SELECT ((12 - no_month) * -5) AS credit_card_charge FROM
(SELECT COUNT(month) AS no_month FROM
(SELECT s.month, s.sum_of_months, c.no_transaction FROM
(SELECT MONTH(date) AS month, sum(IFF(amount < 0, amount, 0)) AS sum_of_months FROM transaction2 GROUP BY month) AS s
LEFT OUTER JOIN
(SELECT MONTH(date) AS month, count(amount) AS no_transaction FROM transaction2 WHERE amount < 0 GROUP BY month) AS C
ON s.month = c.month) AS no_month_satisfied_both_cond WHERE no_transaction >= 3 AND sum_of_months <= -100) AS credit_card_charge_table) AS final;

ACCOUNTADMIN • ASSIGNMENT_1

Share

Updated 29 seconds ago

Draft

Objects

Editor

Results

Chart

...	BALANCE	Query Details
1	20,940	Query duration 67ms