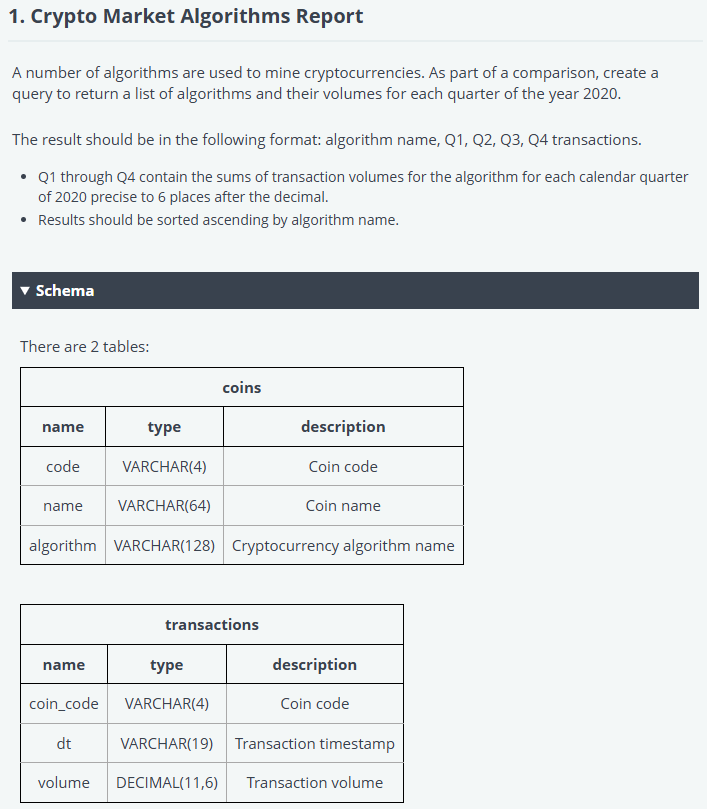
# Crypto Market Algorithms Report



/\*

Enter your query below.

Please append a semicolon ";" at the end of the query

\*/

SELECT

    c.algorithm,

    -- Q1: January to March

    COALESCE(SUM(CASE WHEN EXTRACT(MONTH FROM t.dt) BETWEEN 1 AND 3 THEN t.volume ELSE 0 END), 0) AS transactions\_Q1,

    -- Q2: April to June

    COALESCE(SUM(CASE WHEN EXTRACT(MONTH FROM t.dt) BETWEEN 4 AND 6 THEN t.volume ELSE 0 END), 0) AS transactions\_Q2,

    -- Q3: July to September

    COALESCE(SUM(CASE WHEN EXTRACT(MONTH FROM t.dt) BETWEEN 7 AND 9 THEN t.volume ELSE 0 END), 0) AS transactions\_Q3,

    -- Q4: October to December

    COALESCE(SUM(CASE WHEN EXTRACT(MONTH FROM t.dt) BETWEEN 10 AND 12 THEN t.volume ELSE 0 END), 0) AS transactions\_Q4

FROM

    coins c

JOIN

    transactions t ON c.code = t.coin\_code

WHERE

    EXTRACT(YEAR FROM t.dt) = 2020

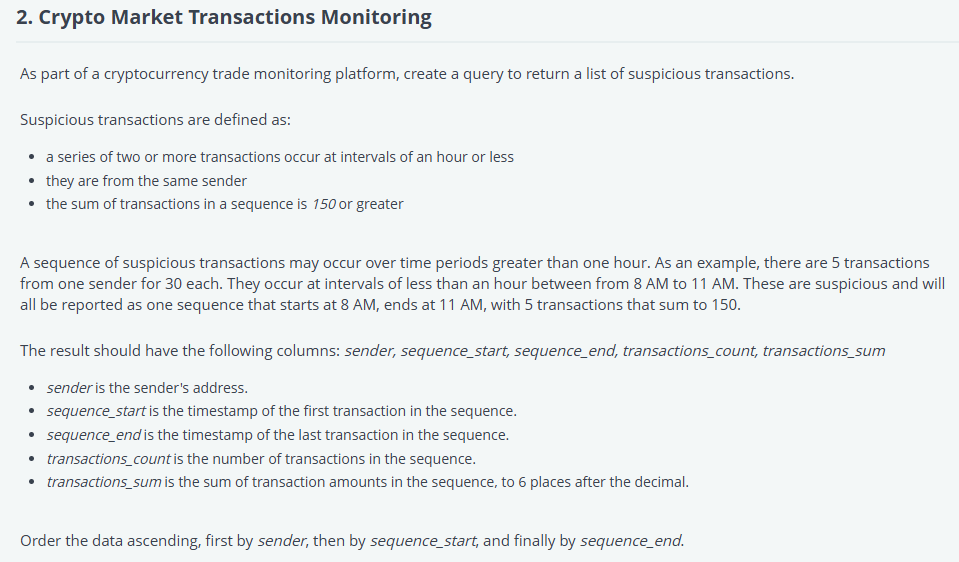
GROUP BY

    c.algorithm

ORDER BY

    c.algorithm ASC;

# Crypto Market Transactions Monitoring



WITH difference AS (

SELECT \*

, DATEDIFF(MINUTE, LAG(dt) OVER (ORDER BY sender, dt), dt) AS difF\_minute

, ROW\_NUMBER() OVER (ORDER BY sender, dt) AS rownumber

FROM [dbo].[Crypto]

),

rn AS (

SELECT rownumber

FROM difference

WHERE rownumber IN (

SELECT rownumber

FROM [dbo].[Crypto]

WHERE ABS(diff\_minute) < 60

)

),

sequences\_ as (

SELECT \*

FROM difference

WHERE rownumber IN(

SELECT rownumber

FROM rn

UNION

SELECT rownumber - 1 as rownumber

FROM rn

)

)

SELECT sender

, MIN(dt) AS Sequence\_start

, MAX(dt) AS Sequence\_end

, COUNT(rownumber) AS transactions\_count

, SUM(amount) AS transactions\_sum

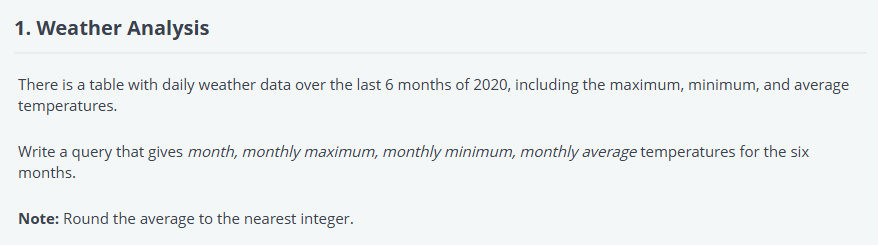
FROM sequences\_

GROUP BY sender

HAVING SUM(amount) >= 150

ORDER BY sender, MIN(dt), MAX(dt)

# Weather Analysis



SELECT MONTH(record\_date), MAX(data\_value) AS max, MIN(data\_value) AS min,

    round(AVG(CASE WHEN data\_type = 'avg' then data\_value END)) AS avg

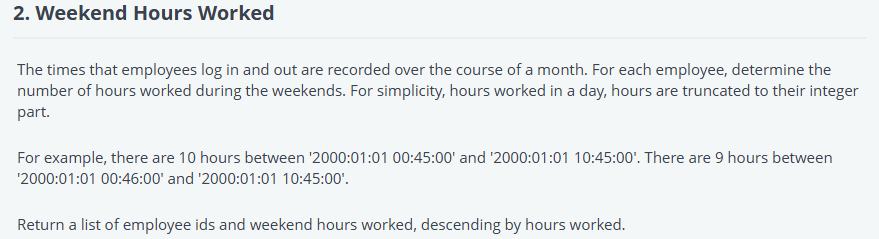
FROM temperature\_records

WHERE MONTH(record\_date) BETWEEN 7 AND 12

GROUP BY MONTH(record\_date)

ORDER BY MONTH(record\_date);

# Weekend Hours Worked



/\*

Enter your query below.

Please append a semicolon ";" at the end of the query

\*/

WITH hours\_worked as (

SELECT

      emp\_id,

       CASE

       WHEN datepart(minute,TIMESTAMP) >= datepart(minute,lag(TIMESTAMP) OVER(PARTITION BY CAST(TIMESTAMP AS date),emp\_id ORDER BY TIMESTAMP)) then datepart(hour,timestamp) - datepart(hour,lag(TIMESTAMP) OVER(PARTITION BY CAST(TIMESTAMP AS date),emp\_id ORDER BY TIMESTAMP))

       ELSE datepart(hour,timestamp) - datepart(hour,lag(TIMESTAMP) OVER(PARTITION BY CAST(TIMESTAMP AS date),emp\_id ORDER BY TIMESTAMP)) - 1

       END AS hours\_worked

FROM   attendance

-- only weekends

WHERE  datepart(weekday,TIMESTAMP) IN(7,1)

)

SELECT

   emp\_id,

      SUM(hours\_worked) as hours\_worked

FROM  hours\_worked

GROUP BY emp\_id

ORDER BY hours\_worked desc