```
-- COUNT is a SQL aggregate function for counting the number of rows in a particular column.
/*
SELECT COUNT(*)
  FROM tutorial.aapl_historical_stock_price
   */

/*
SELECT Count(DATE) AS "Count of Date"
FROM tutorial.aapl_historical_stock_price */

-- SUM is a SQL aggregate function. that totals the values in a given column. Aggregators only aggregate vertically. If you want to perform a calculation across rows, you would do this with simple arithmetic.
/*select SUM(volume) from tutorial.aapl_historical_stock_price */

-- MIN and MAX are SQL aggregation functions that return the lowest and highest values in a particular column
/* SELECT min(volume) as "Min Volume",
max(volume) as "Max Volume"
from tutorial.aapl_historical_stock_price */

-- AVG is a SQL aggregate function that calculates the average of a selected group of values.
/* select avg(high) from tutorial.aapl_historical_stock_price
WHERE high is not NULL */

-- GROUP BY allows you to separate data into groups, which can be aggregated independently of one another.
/*select year,count(*)  As Count, month from tutorial.aapl_historical_stock_price GROUP by year, month */
/*SELECT year,
       month,
       COUNT(*) AS count
  FROM tutorial.aapl_historical_stock_price
 GROUP BY 1, 2 */

 -- Using GROUP BY with ORDER BY
/* SELECT year, MONTH, count(*) FROM tutorial.aapl_historical_stock_price GROUP by YEAR, month
 order by month, year */
```

```
-- The HAVING clause in SQL is used to filter groups of rows based on conditions applied to aggregate functions.
-- While the WHERE clause filters individual rows before grouping, HAVING filters groups after they have been formed by the GROUP BY clause and aggregate functions have been calculated.
/* select name, sum(sell) as total_sells from tutorial.animal_crossing_dress_up
GROUP by name
having sum(sell) > 10000*/

-- The CASE expression in SQL is used to implement conditional logic within SQL queries, similar to if-else statements in programming languages.
-- It allows you to return different values or perform different actions based on specified conditions
/*
SELECT
   school_name,
   player_name,
   CASE
      WHEN position = 'RB' THEN 'Running Back'
      WHEN position = 'WR' THEN 'Wide Receiver'
      WHEN position = 'QB' THEN 'Quarterback'
      ELSE 'Other Position'
   END AS player_position
FROM benn.college_football_players; */

/* SELECT
   player_name,
   weight,
   CASE
      WHEN weight > 250 THEN 'over 250'
      WHEN weight > 200 AND weight <= 250 THEN '201-250'
      WHEN weight > 175 AND weight <= 200 THEN '176-200'
      ELSE '175 or under'
   END AS weight_group
FROM benn.college_football_players; */

-- The DISTINCT keyword in SQL is used to eliminate duplicate rows from the result set of a SELECT query, ensuring that only unique values are returned.SELECT DISTINCT month
/*SELECT count(DISTINCT month) as unique_months
  FROM tutorial.aapl_historical_stock_price*/
```