

Dataset A (Question 1 to 4) :

<https://www.db-fiddle.com/f/wFVPv4s89DLRB2fP5tfX8c/8>

1. What is the most purchased item on the menu & how many times was it purchased?
2. Which item was the most popular one for each customer?
3. If each customer \$1 spent equates to 10 points and ramen has 2X points multiplier, how many points would each customer have?
4. In the first week after a customer joins the program, (including their join date) they earn 2x points on ALL items; how many points do customer A and B have at the end of March 2021?

Dataset B (Question 5 to 10)

<https://www.db-fiddle.com/f/jbahqhW5AQwqV1RZ2xExEz/0>

5. What is the monthly distribution of trial plan start_date values for our dataset?
6. What plan start_date values occur after Jan 2021 for our dataset? Show the breakdown by count of events for each plan_name.
7. What are the total number of customers we currently have and the percentage of those customers that have ceased doing business with us (also known as our customer churn rate)?
8. How many customers have churned straight after their initial free trial? — what percentage is this rounded to the nearest whole number?
9. What is the customer count and percentage breakdown of all 5 plan_name values in 2021–12–31?
10. How many customers downgraded from a pro-monthly to a basic monthly plan in 2020?

Dataset C : benn.college_football_players – mode.com (Question 11 to 15)

11. Write a query that includes a column flagged "YES" when a player is from Cincinnati and classify the results with those players first.
12. Write a SQL query to display player names and a height category column. Group players into four height categories based on their height. There are many valid ways to categorize player height, so other answers are possible.
13. Write a SQL query to calculate the total weight of freshman and sophomore (FR/SO) players in New York in one column, and the total weight of junior and senior (JR/SR) players in New York in another column.
14. Write a SQL query to display the number of freshman (FR), sophomore (SO), junior (JR), and senior (SR) players in each state in separate columns. Include an additional column

showing the total number of players per state. Sort the results so that states with the most total players are listed first.

Dataset D: tutorial.city_populations – mode.com (Question 16 to 25)

16.

Write a SQL query to add a new column called "population_density" that categorizes states based on their population. The logic for the column is:

- If a state's population is greater than the average population of all states, categorize it as "High"
- If a state's population is less than or equal to the average, categorize it as "Low"

17. Write a query to add the additional column - percentage_population_state(city population/total population of the state).

18. Find the number of cities where population is more than the population of Texas state

19. Write a query to return all the records where the city population is more than average population of dataset.

20. Get the average, minimum and maximum population across all cities

DATASET D: heather.sfpd_incidents

String Functions and Pivoting data

21. Return only the category and description fields, replacing all occurrences of "AUTO" with "CAR" in the description

22. Concatenate the category, description and resolution fields together

23. Get the day of week name from the date field as a new column

24. Write a query that returns the `description` field, but with the first letter capitalized and the rest of the letters in lower-case.

25. Write a query to that returns the first non-null value in a list.

Datasets used :

benn.college_football_players & benn.college_football_teams

26. Pivoting Rows to Columns for the datasets benn.college_football_players & benn.college_football_teams

Datasets for question 27 to xx

tutorial.sf_crime_incidents_2014_01

tutorial.sf_crime_incidents_cleandate

tutorial.crunchbase_companies

tutorial.crunchbase_acquisitions

tutorial.crunchbase_investments_part1

tutorial.crunchbase_investments_part2

27. Write a query that ranks investors from the combined dataset above by the total number of investments they have

made. Use: tutorial.crunchbase_investments_part1, tutorial.crunchbase_investments_part2 tables

28.

Write a query that ranks investors from the combined dataset above by the total number of investments they have

made. Consider only the companies whose status is operating. Use:

tutorial.crunchbase_investments_part1,

tutorial.crunchbase_investments_part2 tables for investment. Use: tutorial.crunchbase_companies for status.

Dataset tutorial.dc_bikeshare_q1_2012

29. Display the running total (cumulative) of duration_seconds for all the rides per terminal against each row. Partition the data at

start_terminal. Consider the data where start_time is before '2012-12-31'.

30 . Write a query to find the total (sum) of all the durations_seconds for all the trips for each terminal. Display the data against

each row in the dataset. Partition the data at the start_terminal and sort the rows in the window by start_time of the trip.

Consider the data where start_time is before '2012-12-31'.

Dataset : tutorial.dc_bikeshare_q1_2012

31. Write a query to rank all the rides per terminal on the ascending order of the duration taken per ride.

32. Write a query that shows the 5 longest rides from each starting terminal, ordered by terminal, and longest to shortest rides within each terminal. There are limit to rides that occurred before Jan. 10, 2012.

33. Write a query to assign a row number to the entire dataset ordered by the start_time. Consider the data where the start_time is before '2012-01-10'.

34. Write a query that shows only the duration of the trip and the percentile into which that duration falls (across the entire dataset—not partitioned by terminal).

Dataset : Tutorial.sat_scores

35. Write a query to add column - avg_sat_writing. Each row in this column should include average marks in the writing section of the student per school

36. In the above question, add an additional column - count_per_school. Each row of this column should include number of students per school

37. In the above question, add two additional columns - max_per_teacher and min_per_teacher. Each row of this column should include maximum and minimum marks in maths per teacher respectively.

38. For the dataset, write a query to add the two columns cum_hrs_studied and total_hrs_studied. Each row in cum_hrs_studied should display the cumulative sum of hours studied per school. Each row in the total_hrs_studied will display total hours studied per school. Order the data in the ascending order of student id

39. Write a query to rank the students per school on the basis of scores in writing. Use both rank and dense_rank function. Student with the highest marks should get rank 1.

****Note:** see if there is difference in ranking provided by both the functions for teacher = 'Spellman'

40. Write a query to find the top 5 students per teacher who spent maximum hours studying.

41. Write a query to find the worst 5 students per school who got minimum marks in sat_math

42. Write a query to divide the dataset into quartile on the basis of marks in sat_verbal.

43. Write a query to find the worst 5 students per school who got minimum marks in sat_math

44. Write a query to divide the dataset into quartile on the basis of marks in sat_verbal

45. For 'Petersville HS' school, write a query to arrange the students in the ascending order of hours studied. Also, add a column to find the difference in hours studied from the student above(in the row). Exclude the cases where hrs_studied is null.

46.

For 'Washington HS' school, write a query to arrange the students in the descending order of sat_math. Also, add a column to find the difference in sat_math from the student below(in the row).

47. Write a query to return 4 columns - student_id, school, sat_writing, difference in sat_writing and average marks scored in sat_writing in the school.

48. Write a query to return 4 columns - student_id, teacher, sat_verbal, difference in sat_verbal and minimum marks scored in sat_verbal per teacher.

49. Write a query to return the student_id and school who are in bottom 20 in each of sat_verbal, sat_writing, and sat_math for their school.

50. Write a query to find the student_id for the highest mark and lowest mark per teacher for sat_writing.