

## Medium

**M-1) Compare each employee's salary with the average salary of the corresponding department. Output the department, first name, and salary of employees along with the average salary of that department.**

`employee(id:int, first_name:vchar, last_name:vchar, age:int, sex:vchar, employee_title:vchar, department:vchar, salary:int, target:int, bonus:int, email:vchar, city:vchar, address:vchar, manager_id:int)`

**M-2) What were the top 10 ranked songs in 2010? Output the rank, group name, and song name but do not show the same song twice. Sort the result based on the year\_rank in ascending order.**

`billboard_top_100_year_end(id:int, year:int, year_rank:int, group_name:vchar, artist:vchar, song_name:vchar)`

**M-3) Find the number of apartments per nationality that are owned by people under 30 years old. Output the nationality along with the number of apartments. Sort records by the apartments count in descending order.**

`airbnb_hosts(host_id:int, nationality:vchar, gender:vchar, age:int)`

`airbnb_units(host_id:int, unit_id:vchar, unit_type:vchar, n_beds:int, n_bedrooms:int, country:vchar, city:vchar)`

# Hint: unit\_type = 'Apartment'

**M-4) Find the average total compensation based on employee titles and gender. Total compensation is calculated by adding both the salary and bonus of each employee. However, not every employee receives a bonus so disregard employees without bonuses in your calculation. Employee can receive more than one bonus. Output the employee title, gender (i.e., sex), along with the average total compensation.**

`sf_employee(id:int, first_name:vchar, last_name:vchar, age:int, sex:vchar, employee_title:vchar, department:vchar, salary:int, target:int, email:vchar, city:vchar, address:vchar, manager_id:int)`

`sf_bonus(worker_ref_id:int, bonus:int, bonus_date:datetime)`

**M-5) Find the Olympics with the highest number of athletes. The Olympics game is a combination of the year and the season, and is found in the 'games' column. Output the Olympics along with the corresponding number of athletes.**

`olympics_athletes_events(id:int, name:vchar, sex:vchar, age:float, height:float, weight:datetime, team:vchar, noc:vchar, games:vchar, year:int, season:vchar, city:vchar, sport:vchar, event:vchar, medal:vchar)`

**M-6) Find the date with the highest total energy consumption from the Meta/Facebook data centers. Output the date along with the total energy consumption across all data centers.**

`fb_eu_energy(date:datetime, consumption:int)`

`fb_asia_energy(date:datetime, consumption:int)`

`fb_na_energy(date:datetime, consumption:int)`

**M-7) Find songs that have ranked in the top position. Output the track name and the number of times it ranked at the top. Sort your records by the number of times the song was in the top position in descending order.**

`spotify_worldwide_daily_song_ranking(id:int, position:int, trackname:vchar, artist:vchar, streams:int, url:vchar, date:datetime, region:vchar)`

**M-8) Rank guests based on the number of messages they've exchanged with the hosts. Guests with the same number of messages as other guests should have the same rank. Do not skip rankings if the preceding rankings are identical. Output the rank, guest id, and number of total messages they've sent. Order by the highest number of total messages first.**

`airbnb_contacts(id_guest:vchar, id_host:vchar, id_listing:vchar, ts_contact_at:datetime, ts_reply_at:datetime, ts_accepted_at:datetime, ts_booking_at:datetime, ds_checkin:datetime, ds_checkout:datetime, n_guests:int, n_messages:int)`

**M-9) Find the 3 most profitable companies in the entire world. Output the result along with the corresponding company name. Sort the result based on profits in descending order.**

forbes\_global\_2010\_2014(company:vchar, sector:vchar, industry:vchar, continent:vchar, country:vchar, marketvalue:float, sales:float, profits:float, assets:float, rank:int, forbeswebpage:vchar)

**M-10) Write a query that will identify returning active users. A returning active user is a user that has made a second purchase within 7 days of any other of their purchases. Output a list of user\_ids of these returning active users.**

amazon\_transactions(id:int, user\_id:int, item:vchar, created\_at:datetime, revenue:int)

**M-11) Calculate the total revenue from each customer in March 2019. Include only customers who were active in March 2019. Output the revenue along with the customer id and sort the results based on the revenue in descending order.**

orders(id:int, cust\_id:int, order\_date:datetime, order\_details:vchar, total\_order\_cost:int)

**M-12) You're given a dataset of health inspections. Count the number of violation in an inspection in 'Roxanne Cafe' for each year. If an inspection resulted in a violation, there will be a value in the 'violation\_id' column. Output the number of violations by year in ascending order.**

sf\_restaurant\_health\_violations(business\_id:int, business\_name:vchar, business\_address:vchar, business\_city:vchar, business\_state:vchar, business\_postal\_code:float, business\_latitude:float, business\_longitude:float, business\_location:vchar, business\_phone\_number:float, inspection\_id:vchar, inspection\_date:datetime, inspection\_score:float, inspection\_type:vchar, violation\_id:vchar, violation\_description:vchar, risk\_category:vchar)

**M-13) Classify each business as either a restaurant, cafe, school, or other. A restaurant should have the word 'restaurant' in the business name. For cafes, either 'cafe', 'café', or 'coffee' can be in the business name. 'School' should be in the business name for schools. All other businesses should be classified as 'other'. Output the business name and the calculated classification.**

sf\_restaurant\_health\_violations(business\_id:int, business\_name:vchar, business\_address:vchar, business\_city:vchar, business\_state:vchar, business\_postal\_code:float, business\_latitude:float, business\_longitude:float, business\_location:vchar, business\_phone\_number:float, inspection\_id:vchar, inspection\_date:datetime, inspection\_score:float, inspection\_type:vchar, violation\_id:vchar, violation\_description:vchar, risk\_category:vchar)

**M-14) Find the rate of processed tickets for each type.**

facebook\_complaints(complaint\_id:int, type:int, processed:bool)

#### ANSWER-M1

```
SELECT department, first_name, salary, AVG(salary) OVER (PARTITION BY department) FROM employee
# We get all records in a table using the PARTITION BY clause. It gives aggregated columns with each record in the specified table.
```

#### ANSWER-M2

```
SELECT DISTINCT year_rank, group_name, song_name
FROM billboard_top_100_year_end
WHERE year=2010 AND year_rank BETWEEN 1 AND 10
ORDER BY year_rank;
```

#### ANSWER-M3

```
SELECT nationality, COUNT(DISTINCT unit_id) AS apartment_count
FROM airbnb_units AS apartment INNER JOIN airbnb_hosts AS host ON apartment.host_id = host.host_id
WHERE host.age < 30 AND unit_type = 'Apartment'
GROUP BY host.nationality ORDER BY apartment_count DESC
```

#### ANSWER-M4

*Approach: each employee has multiple bonus, so first obtain total bonus of each employee; then add it with salary.*

```
SELECT e.employee_title, e.sex, AVG(e.salary + b.ttl_bonus) AS avg_compensation FROM
sf_employee AS e
INNER JOIN
(SELECT worker_ref_id, SUM(bonus) AS ttl_bonus FROM sf_bonus GROUP BY worker_ref_id) AS b
ON e.id = b.worker_ref_id GROUP BY employee_title, sex
```

#### ANSWER-M5

```
WITH subquery AS(
SELECT games, COUNT(DISTINCT id) AS athletes_count
FROM olympics_athletes_events
GROUP BY games ORDER BY athletes_count DESC)

SELECT * FROM subquery
WHERE athletes_count = (SELECT MAX(athletes_count) FROM subquery)
```

#### ANSWER-M6

Approach: UNION ALL combines rows of two or more select statements and allows duplicates.

```
WITH
merged_energy AS (
SELECT * FROM fb_eu_energy eu UNION ALL
SELECT * FROM fb_asia_energy asia UNION ALL
SELECT * FROM fb_na_energy na),

energy_by_date AS (
SELECT date, SUM(consumption) AS total_energy FROM merged_energy GROUP BY date ORDER BY date ASC),

max_energy AS (SELECT MAX(total_energy) AS max_energy FROM energy_by_date)

SELECT ebd.date, ebd.total_energy
FROM energy_by_date AS ebd JOIN max_energy AS me ON ebd.total_energy = me.max_energy
```

ANSWER-M7

```
SELECT trackname, COUNT(*) AS times_top1
FROM spotify_worldwide_daily_song_ranking
WHERE position = 1 GROUP BY trackname
ORDER BY times_top1 DESC
```

ANSWER-M8

```
SELECT DENSE_RANK() OVER(ORDER BY sum(n_messages) DESC) AS ranking,
id_guest, SUM(n_messages) AS sum_n_messages
FROM airbnb_contacts GROUP BY id_guest ORDER BY sum_n_messages DESC
```

ANSWER-M9

```
SELECT company, profit FROM (
SELECT *, DENSE_RANK() OVER (ORDER BY profit DESC) AS rank FROM
(
SELECT company, SUM(profits) AS profit FROM forbes_global_2010_2014 GROUP BY company) AS sq
) AS sq2
WHERE rank <=3
```

ANSWER-M10

```
SELECT DISTINCT(a1.user_id)
FROM amazon_transactions AS a1 JOIN amazon_transactions AS a2 ON
a1.user_id=a2.user_id AND a1.id <> a2.id AND (a2.created_at::DATE - a1.created_at::DATE) BETWEEN 0 AND 7
ORDER BY a1.user_id
```

ANSWER-M11

```
SELECT cust_id, SUM(total_order_cost) AS revenue
FROM orders WHERE
EXTRACT(MONTH FROM order_date :: TIMESTAMP) = 3 AND EXTRACT(YEAR FROM order_date :: TIMESTAMP) = 2019
GROUP BY cust_id ORDER BY revenue DESC
```

ANSWER-M12

```
SELECT EXTRACT (YEAR FROM inspection_date :: DATE) AS year, COUNT(*) AS n_inspections
FROM sf_restaurant_health_violations
WHERE business_name = 'Roxanne Cafe' AND violation_id IS NOT NULL
GROUP BY year ORDER BY year ASC
```

ANSWER-M13

```
SELECT DISTINCT business_name,
CASE
    WHEN business_name ILIKE ANY(ARRAY['%school%']) THEN 'school'
    WHEN LOWER(business_name) LIKE ANY(ARRAY['%restaurant%']) THEN 'restaurant'
    WHEN LOWER(business_name) LIKE ANY(ARRAY['%cafe%', '%café%', '%coffee%']) THEN 'cafe'
    ELSE 'other'
END AS business_type
FROM sf_restaurant_health_violations
```

ANSWER-M14

```
SELECT type, SUM(CASE WHEN processed THEN 1 ELSE 0 END) :: NUMERIC /COUNT(*) AS processed_rate
FROM facebook_complaints GROUP BY type
```