RevRoll – SQL

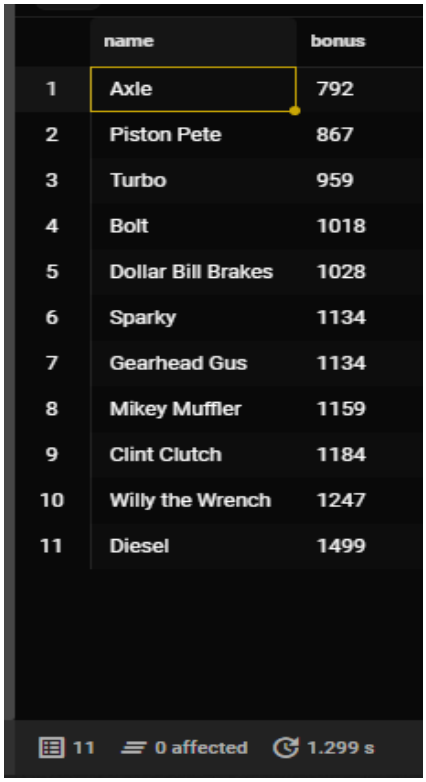
RevRoll is an auto parts dealer and installer. They offer a full range of automotive parts replacement services.

Question #1:

Installers receive performance-based year end bonuses. Bonuses are calculated by taking 10% of the total value of parts installed by the installer.

Calculate the bonus earned by each installer rounded to a whole number. Sort the result by bonus in increasing order.

```
SELECT
   i.name AS name,
   ROUND(SUM(p.price * o.quantity) * 0.1) AS bonus
FROM
   Installers i
JOIN
   Installs ins ON i.installer_id = ins.installer_id
JOIN
   Orders o ON ins.order_id = o.order_id
JOIN
   Parts p ON o.part_id = p.part_id
GROUP BY
   i.name
ORDER BY
   bonus;
```



	name	bonus
1	Axle	792
2	Piston Pete	867
3	Turbo	959
4	Bolt	1018
5	Dollar Bill Brakes	1028
6	Sparky	1134
7	Gearhead Gus	1134
8	Mikey Muffler	1159
9	Clint Clutch	1184
10	Willy the Wrench	1247
11	Diesel	1499

11 0 affected 1.299 s

Question #2:

RevRoll encourages healthy competition. The company holds a “Install Derby” where installers face off to see who can change a part the fastest in a tournament style contest.

Derby points are awarded as follows:

An installer receives three points if they win a match (i.e., Took less time to install the part).

An installer receives one point if they draw a match (i.e., Took the same amount of time as their opponent).

An installer receives no points if they lose a match (i.e., Took more time to install the part).

```
SELECT
   installer_id,
   name,
   SUM(CASE
         WHEN installer_one_time < installer_two_time THEN 3
         WHEN installer_one_time = installer_two_time THEN 1
         ELSE 0
      END
   ) AS num_points
FROM
   Install_Derby
JOIN
   Installers i ON Install_Derby.installer_one_id = i.installer_id
GROUP BY
   installer_id, name

UNION ALL

SELECT
   installer_id,
   name,
   SUM(CASE
         WHEN installer_two_time < installer_one_time THEN 3
         WHEN installer_two_time = installer_one_time THEN 1
```

```
            ELSE 0
        END
    ) AS num_points
FROM
    Install_Derby
JOIN
    Installers i ON Install_Derby.installer_two_id = i.installer_id
GROUP BY
    installer_id, name

ORDER BY
    num_points DESC, installer_id;
```

	installer_id	name	num_points
1	2	Gearhead Gus	21
9	6	Turbo	6
10	8	Piston Pete	6
11	9	Axle	6
12	10	Dollar Bill Brakes	6
13	4	Sparky	3
14	7	Mikey Muffler	3
15	7	Mikey Muffler	3
16	9	Axle	3
17	10	Dollar Bill Brakes	3
18	3	Bolt	0
2	4	Sparky	15
3	6	Turbo	13
4	5	Clint Clutch	12

18 0 affected 1.239 s

Question #3:

Write a query to find the fastest install time with its corresponding derby_id for each installer. In case of a tie, you should find the install with the smallest derby_id.

Return the result table ordered by installer_id in ascending order.

```
WITH RankedInstalls AS (
  SELECT
    derby_id,
    installer_one_id AS installer_id,
    installer_one_time AS install_time,
    ROW_NUMBER() OVER (PARTITION BY installer_one_id ORDER BY installer_one_time, derby_id) AS rnk
  FROM
    Install_Derby

  UNION ALL

  SELECT
    derby_id,
    installer_two_id AS installer_id,
    installer_two_time AS install_time,
    ROW_NUMBER() OVER (PARTITION BY installer_two_id ORDER BY installer_two_time, derby_id) AS rnk
  FROM
    Install_Derby
)
SELECT
  derby_id,
  installer_id,
  install_time
FROM
  RankedInstalls
WHERE  rnk = 1 ORDER BY  installer_id ASC;
```

	derby_id	installer_id	install_time
1	12	2	22
7	28	5	21
8	14	6	27
9	45	6	26
10	48	7	34
11	36	7	47
12	46	8	24
13	52	8	27
14	53	9	32
15	25	9	23
16	33	10	25
17	55	10	28
18	34	11	22
2	26	3	23

18 0 affected 1.236 s

Question #4:

Write a solution to calculate the total parts spending by customers paying for installs on each Friday of every week in November 2023. If there are no purchases on the Friday of a particular week, the parts total should be set to 0.

Return the result table ordered by week of month in ascending order.

```
WITH DateSeries AS (
    SELECT
        generate_series('2023-11-01'::date, '2023-11-30'::date, '1 day'::interval)::date AS date
)
, Fridays AS (
    SELECT
        date
    FROM
        DateSeries
    WHERE
        EXTRACT(DOW FROM date) = 5 -- 5 corresponds to Friday
)
, Weeks AS (
    SELECT
        date,
        EXTRACT(WEEK FROM date) AS week_of_month
    FROM
        Fridays
)
, PartsSpending AS (
    SELECT
        w.date AS november_fridays,
        COALESCE(ROUND(SUM(p.price * o.quantity), 2), 0) AS parts_total
    FROM
        Weeks w
    LEFT JOIN
        Orders o ON w.date = o.install_date
```

```
    LEFT JOIN
        Parts p ON o.part_id = p.part_id
    WHERE
        EXTRACT(MONTH FROM w.date) = 11
        AND EXTRACT(YEAR FROM w.date) = 2023
    GROUP BY
        w.date
)
SELECT
    november_fridays,
    parts_total
FROM
    PartsSpending
ORDER BY
    week_of_month ASC;
```