1. Top Performers
   Identify caregivers with the highest number of completed visits. Clearly define 'completed' and briefly explain your reasoning.
   Answer:
   Definition:
   A 'completed' visit is the one satisfying the following conditions:
   clock_in_actual_datetime<=start_datetime, (1)
   clock_out_actual_datetime>=end_datetime. (2)

   Since start_datetime/end_datetime are scheduled shift start and end. Inequality (1) means the caregivers does not come late. Inequality (2) means the caregivers does not leave early. Two inequalities assures that the visiting time is enough and meeting the requirement of scheduled shift. So when a carelog satisfies inequalities (1) and (2), it is defined as a 'completed' visit.

   SQL Query:

```
SELECT
  "caregiverId",
  COUNT(*) AS completed_visit_count
FROM carelogs
WHERE
  "clockInActualDatetime" <= "startDatetime"
  AND "clockOutActualDatetime" >= "endDatetime"
GROUP BY "caregiverId"
ORDER BY completed_visit_count DESC;
```

   Sample Outputs:

```
-[ RECORD 1 ]---------+-----------
caregiverId           | 56f5cc4b85
completed_visit_count | 294
-[ RECORD 2 ]---------+-----------
caregiverId           | 98bceaf3fc
completed_visit_count | 171
-[ RECORD 3 ]---------+-----------
caregiverId           | ff48a39a63
completed_visit_count | 170
-[ RECORD 4 ]---------+-----------
caregiverId           | 04e64191b5
completed_visit_count | 154
-[ RECORD 5 ]---------+-----------
caregiverId           | 24b30a5ab1
completed_visit_count | 150
-[ RECORD 6 ]---------+-----------
caregiverId           | baad624bf5
completed_visit_count | 144
-[ RECORD 7 ]---------+-----------
caregiverId           | 02faee4de6
completed_visit_count | 125
-[ RECORD 8 ]---------+-----------
caregiverId           | a18631d720
completed_visit_count | 105
-[ RECORD 9 ]---------+-----------
caregiverId           | e14b12c5d1
completed_visit_count | 99
-[ RECORD 10 ]--------+-----------
caregiverId           | e10e011814
completed_visit_count | 93
-[ RECORD 11 ]--------+-----------
caregiverId           | a667ce9032
completed_visit_count | 93
-[ RECORD 12 ]--------+-----------
caregiverId           | 75344d8077
completed_visit_count | 92
```

2. Reliability Issues:
Highlight caregivers showing frequent reliability issues(e.g., late arrivals, cancellations, missed visits). Clearly explain your criteria for identifying these caregivers.
Answer:
Definition:
A caregiver is assumed to have reliability issues, if any of their carelogs meets one of the following conditions
(1) One of start_datetime, end_datetime, clock_in_actual_datetime and clock_out_actual_datetime is null
(2) (end_datetime-start_datetime)>(clock_out_actual_datetime-clock_in_actual_datetime)
(3) clock_in_actual_datetime > start_datetime + 10 minutes

SQL Query:

```
WITH flagged_logs AS (
  SELECT
    "caregiverId",
    "carelogId",
     (
       "startDatetime" IS NULL OR
       "endDatetime" IS NULL OR
       "clockInActualDatetime" IS NULL OR
       "clockOutActualDatetime" IS NULL
     )  AS has_missing_fields,
    (
      "startDatetime" IS NOT NULL AND
      "endDatetime" IS NOT NULL AND
      "clockInActualDatetime" IS NOT NULL AND
      "clockOutActualDatetime" IS NOT NULL AND
      ("clockOutActualDatetime" -  "clockInActualDatetime") < ( "endDatetime" -
"startDatetime")
    ) AS left_early,
    (
      "startDatetime" IS NOT NULL AND
      "clockInActualDatetime" IS NOT NULL AND
      "clockInActualDatetime" > "startDatetime" + INTERVAL '10 minutes'
    ) AS late_arrival
  FROM carelogs
),
caregiver_flags AS (
  SELECT
    "caregiverId",
    COUNT(*) FILTER (
      WHERE has_missing_fields OR left_early OR late_arrival
```

```
  ) AS reliability_issue_count
 FROM flagged_logs
 GROUP BY "caregiverId"
)
SELECT *
FROM caregiver_flags
WHERE reliability_issue_count > 0
ORDER BY reliability_issue_count DESC;
```

Sample Outputs:

```
-[ RECORD 1 ]-----------+-----------
caregiverId             | cd6bd8d5f1
reliability_issue_count | 214
-[ RECORD 2 ]-----------+-----------
caregiverId             | 7bfbfda241
reliability_issue_count | 202
-[ RECORD 3 ]-----------+-----------
caregiverId             | f864a0fb90
reliability_issue_count | 183
-[ RECORD 4 ]-----------+-----------
caregiverId             | 78735ef0b9
reliability_issue_count | 168
-[ RECORD 5 ]-----------+-----------
caregiverId             | 5b8dae6f05
reliability_issue_count | 166
-[ RECORD 6 ]-----------+-----------
caregiverId             | b9ab60b9bf
reliability_issue_count | 147
-[ RECORD 7 ]-----------+-----------
caregiverId             | 2129ff7024
reliability_issue_count | 134
-[ RECORD 8 ]-----------+-----------
caregiverId             | 4cfbacf756
reliability_issue_count | 122
-[ RECORD 9 ]-----------+-----------
caregiverId             | d295fe6f11
reliability_issue_count | 117
-[ RECORD 10 ]----------+-----------
caregiverId             | dbf8bf7cf9
reliability_issue_count | 86
-[ RECORD 11 ]----------+-----------
caregiverId             | a2c8b4391c
reliability_issue_count | 85
-[ RECORD 12 ]----------+-----------
caregiverId             | 83bdde6140
reliability_issue_count | 82
-[ RECORD 13 ]----------+-----------
caregiverId             | a1d6c5ef13
reliability_issue_count | 82
-[ RECORD 14 ]----------+-----------
caregiverId             | 1209c9695d
reliability_issue_count | 77
-[ RECORD 15 ]----------+-----------
caregiverId             | 8222655a46
reliability_issue_count | 75
```

3. Visit Duration Analysis:
   Calculate and clearly present the average actual duration of caregiver visits. Clearly handle potential anomalies such as missing or inconsistent timestamps.
   Answer:
   Duration time:
   clock_out_actual_datetime- clock_in_actual_datetime
   Average duration time:
   sum (clock_out_actual_datetime - clock_in_actual_datetime) over all visiting carelogs of the caregiver, and then divide it by the total number of visiting carelogs of this caregiver

SQL Query:

```
SELECT
  "caregiverId",
  COUNT(*) AS valid_visit_count,
  ROUND(
    AVG(
      EXTRACT(EPOCH FROM "clockOutActualDatetime" - "clockInActualDatetime") /
60
    )::numeric,
    1
  ) AS avg_visit_duration_minutes
FROM carelogs
WHERE
  "clockInActualDatetime" IS NOT NULL
  AND "clockOutActualDatetime" IS NOT NULL
  AND "clockOutActualDatetime" > "clockInActualDatetime"
GROUP BY "caregiverId"
ORDER BY avg_visit_duration_minutes DESC;
```

Sample Outputs:

```
-[ RECORD 1 ]--------------+-----------
caregiverId                | 4d923d62d0
valid_visit_count          | 13
avg_visit_duration_minutes | 1956.9
-[ RECORD 2 ]--------------+-----------
caregiverId                | d10ad48e4a
valid_visit_count          | 10
avg_visit_duration_minutes | 1941.0
-[ RECORD 3 ]--------------+-----------
caregiverId                | 80bd335c43
valid_visit_count          | 12
avg_visit_duration_minutes | 1925.6
-[ RECORD 4 ]--------------+-----------
caregiverId                | fffac188bd
valid_visit_count          | 5
avg_visit_duration_minutes | 1728.0
-[ RECORD 5 ]--------------+-----------
caregiverId                | 9685da0500
valid_visit_count          | 8
avg_visit_duration_minutes | 1687.5
-[ RECORD 6 ]--------------+-----------
caregiverId                | 211115fea1
valid_visit_count          | 10
avg_visit_duration_minutes | 1610.9
-[ RECORD 7 ]--------------+-----------
caregiverId                | 9a21240ae0
valid_visit_count          | 9
avg_visit_duration_minutes | 1573.7
-[ RECORD 8 ]--------------+-----------
caregiverId                | 6b952adc33
valid_visit_count          | 6
avg_visit_duration_minutes | 1445.0
-[ RECORD 9 ]--------------+-----------
caregiverId                | 271704d430
valid_visit_count          | 6
avg_visit_duration_minutes | 1444.6
-[ RECORD 10 ]-------------+-----------
caregiverId                | 1c7cb68c6b
valid_visit_count          | 8
avg_visit_duration_minutes | 1441.2
```

4. Identifying Outliers
   Identify and clearly present visits significantly shorter or longer than typical durations. Explain your criteria and reasoning clearly but succinctly. Briefly suggest potential operational causes or implications of these anomalies.
   Answer:
   Criteria:
   Calculate the mean value and standard deviation(std) of all the duration times. For a given duration time, if it is less than mean minus one std, then it is considered significantly shorter. If it is larger than mean plus three times std, then it is considered significantly longer.

   Suggestion:

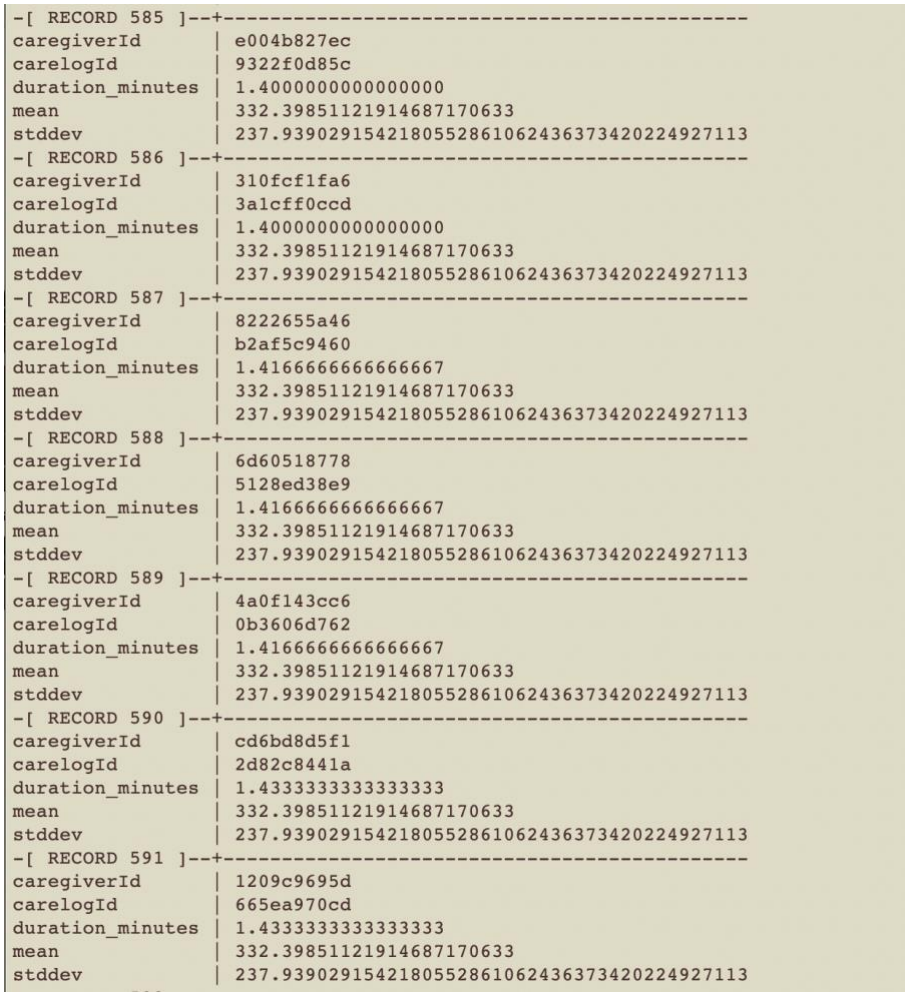
Operationally, a significantly shorter duration time implies that the patient may be not satisfied with the current caregiver. And the patient may want to change for another caregiver. On the other hand, a significantly longer duration time suggests that some emergent events may have occurred, which requires the caregiver much more time than usual to deal with.

SQL Query:

```
WITH durations AS (
  SELECT
    "caregiverId",
     "carelogId",
    EXTRACT(EPOCH FROM "clockOutActualDatetime" - "clockInActualDatetime") / 60
AS duration_minutes
  FROM carelogs
  WHERE
    "clockOutActualDatetime" IS NOT NULL
    AND "clockInActualDatetime" IS NOT NULL
    AND "clockOutActualDatetime" > "clockInActualDatetime"
),
stats AS (
  SELECT
    AVG(duration_minutes) AS mean,
    STDDEV(duration_minutes) AS stddev
  FROM durations
),
outliers AS (
  SELECT d.*, s.mean, s.stddev
  FROM durations d
  CROSS JOIN stats s
  WHERE d.duration_minutes < (s.mean -  s.stddev)
     OR d.duration_minutes > (s.mean + 3 * s.stddev)
)
SELECT *
FROM outliers
ORDER BY duration_minutes;
```

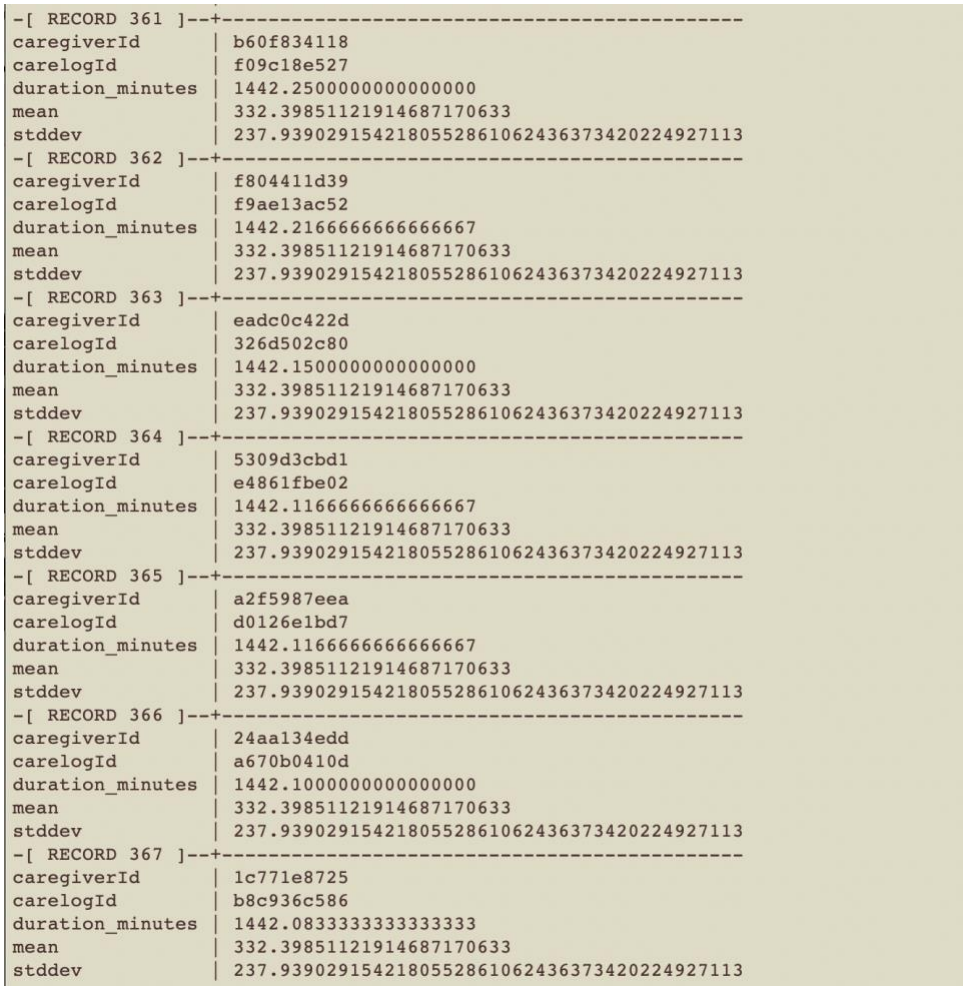
Sample Outputs:

Significantly shorter duration



```
-[ RECORD 585 ]--+--------------------------------------------
caregiverId      | e004b827ec
carelogId        | 9322f0d85c
duration_minutes | 1.4000000000000000
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 586 ]--+--------------------------------------------
caregiverId      | 310fcf1fa6
carelogId        | 3a1cff0ccd
duration_minutes | 1.4000000000000000
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 587 ]--+--------------------------------------------
caregiverId      | 8222655a46
carelogId        | b2af5c9460
duration_minutes | 1.4166666666666667
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 588 ]--+--------------------------------------------
caregiverId      | 6d60518778
carelogId        | 5128ed38e9
duration_minutes | 1.4166666666666667
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 589 ]--+--------------------------------------------
caregiverId      | 4a0f143cc6
carelogId        | 0b3606d762
duration_minutes | 1.4166666666666667
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 590 ]--+--------------------------------------------
caregiverId      | cd6bd8d5f1
carelogId        | 2d82c8441a
duration_minutes | 1.4333333333333333
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 591 ]--+--------------------------------------------
caregiverId      | 1209c9695d
carelogId        | 665ea970cd
duration_minutes | 1.4333333333333333
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
```

Significantly longer duration



```
-[ RECORD 361 ]--+--------------------------------------------
caregiverId      | b60f834118
carelogId        | f09c18e527
duration_minutes | 1442.2500000000000000
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 362 ]--+--------------------------------------------
caregiverId      | f804411d39
carelogId        | f9ae13ac52
duration_minutes | 1442.2166666666666667
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 363 ]--+--------------------------------------------
caregiverId      | eadc0c422d
carelogId        | 326d502c80
duration_minutes | 1442.1500000000000000
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 364 ]--+--------------------------------------------
caregiverId      | 5309d3cbd1
carelogId        | e4861fbe02
duration_minutes | 1442.1166666666666667
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 365 ]--+--------------------------------------------
caregiverId      | a2f5987eea
carelogId        | d0126e1bd7
duration_minutes | 1442.1166666666666667
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 366 ]--+--------------------------------------------
caregiverId      | 24aa134edd
carelogId        | a670b0410d
duration_minutes | 1442.1000000000000000
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
-[ RECORD 367 ]--+--------------------------------------------
caregiverId      | 1c771e8725
carelogId        | b8c936c586
duration_minutes | 1442.0833333333333333
mean             | 332.39851121914687170633
stddev           | 237.939029154218055286106243637342022492711 3
```

5. Detailed Documentation Providers:
   Clearly identify caregivers consistently leaving detailed comments. Define your own criteria for “consistent” and “detailed”.
   Answer:
   Criteria for detailed:
   general_comment_char_count>=100.
   Criteria for consistent:
   80% of a certain caregiver’s carelogs are detailed.

   SQL Query:

```
WITH caregiver_logs AS (
  SELECT
   "caregiverId",
    COUNT(*) AS total_logs,
    COUNT(*) FILTER (
      WHERE "generalCommentCharCount" >= 100
    ) AS detailed_logs
  FROM carelogs
  GROUP BY "caregiverId"
```

```
) ,
consistent_documenters AS (
  SELECT
    "caregiverId",
    detailed_logs,
    total_logs,
    ROUND(detailed_logs::numeric / total_logs, 2) AS detailed_ratio
  FROM caregiver_logs
  WHERE total_logs > 0
)
SELECT *
FROM consistent_documenters
WHERE detailed_ratio >= 0.8
ORDER BY detailed_ratio DESC;
```

Sample Outputs:

```
-[ RECORD 254 ]+-----------
caregiverId    | 0307fb213d
detailed_logs  | 7
total_logs     | 8
detailed_ratio | 0.88
-[ RECORD 255 ]+-----------
caregiverId    | 96e2a8ed79
detailed_logs  | 7
total_logs     | 8
detailed_ratio | 0.88
-[ RECORD 256 ]+-----------
caregiverId    | 9e597ca7fc
detailed_logs  | 7
total_logs     | 8
detailed_ratio | 0.88
-[ RECORD 257 ]+-----------
caregiverId    | e6a737866c
detailed_logs  | 7
total_logs     | 8
detailed_ratio | 0.88
-[ RECORD 258 ]+-----------
caregiverId    | 769c628450
detailed_logs  | 7
total_logs     | 8
detailed_ratio | 0.88
-[ RECORD 259 ]+-----------
caregiverId    | cc1da14fa6
detailed_logs  | 7
total_logs     | 8
detailed_ratio | 0.88
-[ RECORD 260 ]+-----------
caregiverId    | b53b48b187
detailed_logs  | 7
total_logs     | 8
detailed_ratio | 0.88
```

6. Data Quality Check:
   Clearly highlight any unusual or suspicious patterns in documentation data. Briefly describe your methodology and explain why these patterns are important operationally.
   Answer:
   Most documentation fields are blank or null in carelogs table. We can regard extreme long documentation as an unusual or suspicious pattern.

Operationally, a blank or null documentation means there is nothing special happened during a scheduled shift. While a very long documentation implies something emergent may have happened during the scheduled shift. So it is reasoning to look at these very long documentations as unusual.

Then what is the threshold for 'long'? This can be investigated by SQL queries. The query we use is

```
SELECT COUNT(*) AS documentation_count FROM carelogs
WHERE LENGTH(documentation)>n;
```

Here *n* can take any positive integer.

The query results are listed in the following table

n	**0**	**100**	**500**	**1000**	**1300**	**1400**	**1500**
documentation_count	58608	58606	43796	12115	3698	731	12

As the threshold *n* becomes larger, the number of documentations whose length is greater than *n* sharply drops. For instance, we can simply take *n*=1400 as the threshold length for unusual documentation. Then the table tells us there are totally 731 unusual documentations in the existing database.

7. Overtime Identification:
Clearly identify caregivers regularly incurring overtime hours. Define clearly how you determine overtime(e.g., number of hours per week exceeding a threshold, such as 40 hours).
Answer:
Definition
(1) overtime=(clock_out_actual_datetime - clock_in_actual_datetime)-(end_datetime - start_datetime)
(2) If 50% of the carelogs of a given caregiver have overtime>1 hour, then this caregiver is considered to regularly incur overtime hours.

SQL Query:

```
WITH overtime_stats AS (
  SELECT
    "caregiverId",
    COUNT(*) AS total_logs,
    COUNT(*) FILTER (WHERE "overtimeMinutes" > 60) AS overtime_logs
  FROM carelogs
  WHERE "overtimeMinutes" IS NOT NULL
  GROUP BY "caregiverId"
),
overtime_flags AS (
  SELECT
    "caregiverId",
    total_logs,
    overtime_logs,
    ROUND((overtime_logs::decimal / total_logs) * 100, 2) AS overtime_percentage
```

```
  FROM overtime_stats
  WHERE (overtime_logs::decimal / total_logs) >= 0.5
)
SELECT
  of."caregiverId",
  a."agencyId",
  of.total_logs,
  of.overtime_logs,
  of.overtime_percentage
FROM overtime_flags of
JOIN caregivers a ON of."caregiverId" = a."caregiverId"
ORDER BY of.overtime_percentage DESC;
```

Sample Outputs:

```
-[ RECORD 18 ]------+-----------
caregiverId         | 49e67ec68a
agencyId            | 7c792a8279
total_logs          | 21
overtime_logs       | 19
overtime_percentage | 90.48
-[ RECORD 19 ]------+-----------
caregiverId         | a18631d720
agencyId            | 7c792a8279
total_logs          | 114
overtime_logs       | 101
overtime_percentage | 88.60
-[ RECORD 20 ]------+-----------
caregiverId         | e10e011814
agencyId            | 0678ca2eae
total_logs          | 93
overtime_logs       | 82
overtime_percentage | 88.17
-[ RECORD 21 ]------+-----------
caregiverId         | 5e9cf763c0
agencyId            | b90ba83119
total_logs          | 95
overtime_logs       | 83
overtime_percentage | 87.37
-[ RECORD 22 ]------+-----------
caregiverId         | e14b12c5d1
agencyId            | 0765933456
total_logs          | 110
overtime_logs       | 96
overtime_percentage | 87.27
-[ RECORD 23 ]------+-----------
caregiverId         | 24b30a5ab1
agencyId            | b90ba83119
total_logs          | 154
overtime_logs       | 129
overtime_percentage | 83.77
-[ RECORD 24 ]------+-----------
caregiverId         | 69f03622b6
agencyId            | bfbb56bee6
total_logs          | 12
overtime_logs       | 10
overtime_percentage | 83.33
-[ RECORD 25 ]------+-----------
caregiverId         | 98bceaf3fc
agencyId            | ebd774c929
total_logs          | 197
overtime_logs       | 164
overtime_percentage | 83.25
```

8. Operational Insights:
   Highlight any patterns or insight related overtime:
   Are specific caregivers or agencies disproportionately responsible for overtime?
   Do certain schedules or visit types correlate with higher overtime?
   Answer:
   (1) Yes, there are specific caregivers or agencies disproportionately responsible for overtime.
   These agencies' agency_id are:
   "7c792a8279",
   "a0872cc5b5".

(2) From the carelogs data table, there are no fields related to visit types. From the following query results, we see that when scheduled start time is between 1~4 o'clock, then the corresponding overtime is highest.

SQL Query:

```
SELECT
  EXTRACT(HOUR FROM "startDatetime") AS hour_of_day,
  COUNT(*) AS total_visits,
  ROUND(AVG("overtimeMinutes")::numeric, 2) AS avg_overtime_minutes
FROM carelogs
WHERE
  "startDatetime" IS NOT NULL
  AND "overtimeMinutes" IS NOT NULL
GROUP BY hour_of_day
ORDER BY avg_overtime_minutes DESC;
```

Sample Outputs:

```
-[ RECORD 1 ]--------+------
hour_of_day          | 1
total_visits         | 429
avg_overtime_minutes | 89.34
-[ RECORD 2 ]--------+------
hour_of_day          | 2
total_visits         | 1558
avg_overtime_minutes | 32.20
-[ RECORD 3 ]--------+------
hour_of_day          | 3
total_visits         | 9207
avg_overtime_minutes | 12.92
-[ RECORD 4 ]--------+------
hour_of_day          | 4
total_visits         | 23757
avg_overtime_minutes | 11.89
-[ RECORD 5 ]--------+------
hour_of_day          | 11
total_visits         | 9773
avg_overtime_minutes | 11.85
-[ RECORD 6 ]--------+------
hour_of_day          | 12
total_visits         | 10064
avg_overtime_minutes | 11.21
-[ RECORD 7 ]--------+------
hour_of_day          | 0
total_visits         | 218
avg_overtime_minutes | 9.14
-[ RECORD 8 ]--------+------
hour_of_day          | 5
total_visits         | 35188
avg_overtime_minutes | 7.61
-[ RECORD 9 ]--------+------
hour_of_day          | 10
total_visits         | 10158
avg_overtime_minutes | 7.23
-[ RECORD 10 ]-------+------
hour_of_day          | 15
total_visits         | 8136
avg_overtime_minutes | 6.93
```