🧩 Scenario

You have a table `DAILY_TEMP`:

DATE_RECORDED	TEMPERATURE
2025-10-01	31
2025-10-02	33
2025-10-03	32
2025-10-04	34
2025-10-05	35

You want to find:

- **Temperature difference from the previous day (LAG)**
- **Temperature difference to the next day (LEAD)**
  without showing **NULL** for the first or last row.

🧠 Step 1: The Base Query (with NULLs)

```
SELECT
  date_recorded,
  temperature,
  LAG(temperature) OVER (ORDER BY date_recorded) AS prev_temp,
  LEAD(temperature) OVER (ORDER BY date_recorded) AS next_temp
FROM daily_temp;
```

🔍 Output (with NULLs)

DATE_RECORDED	TEMP	PREV_TEMP	NEXT_TEMP
2025-10-01	31	NULL	33
2025-10-02	33	31	32

2025-10-03	32	33	34
2025-10-04	34	32	35
2025-10-05	35	34	NULL

⚙️ Step 2: Handle NULLs Gracefully

We can replace the `NULL` values using the `NVL()` or `COALESCE()` function in Oracle.

✅ Option 1: Replace with Current Day's Temperature

That way, the difference becomes `0` for the first or last day.

```
SELECT
  date_recorded,
  temperature,
  NVL(LAG(temperature) OVER (ORDER BY date_recorded), temperature) AS
prev_temp,
  NVL(LEAD(temperature) OVER (ORDER BY date_recorded), temperature) AS
next_temp
FROM daily_temp;
```

Output

DATE_RECORDED	TEMP	PREV_TEMP	NEXT_TEMP
2025-10-01	31	31	33
2025-10-02	33	31	32
2025-10-03	32	33	34
2025-10-04	34	32	35
2025-10-05	35	34	35

⚙️ Step 3: Compute Temperature Differences

You can now calculate:

- `temp_diff_prev` = `temperature - prev_temp`
- `temp_diff_next` = `next_temp - temperature`

✅ Final Query

```
SELECT
  date_recorded,
  temperature,
  temperature - NVL(LAG(temperature) OVER (ORDER BY date_recorded),
temperature) AS diff_prev_day,
  NVL(LEAD(temperature) OVER (ORDER BY date_recorded), temperature) -
temperature AS diff_next_day
FROM daily_temp
ORDER BY date_recorded;
```

📊 Output

DATE_RECORDED	TEMP	DIFF_PREV_DAY	DIFF_NEXT_DAY
2025-10-01	31	0	2
2025-10-02	33	2	-1
2025-10-03	32	-1	2
2025-10-04	34	2	1
2025-10-05	35	1	0

💡 Alternative Options to Handle NULLs

Approach	How it Works	Output Effect
`NVL()`	Replaces NULL with current temp	First/last difference = 0

`NVL2()`	Conditionally handle NULL	Can assign 'NA' for first/last day
`IGNORE NULLS` (in some DBs)	Skips nulls in LAG/LEAD	Not always available in Oracle

Example:

```
LAG(temperature IGNORE NULLS) OVER (ORDER BY date_recorded)
```

🧮 Optional: Show "NA" Instead of Zero

If you prefer textual output for first/last days:

```
SELECT
  date_recorded,
  temperature,
  CASE
    WHEN LAG(temperature) OVER (ORDER BY date_recorded) IS NULL THEN
'NA'
    ELSE TO_CHAR(temperature - LAG(temperature) OVER (ORDER BY
date_recorded))
  END AS diff_prev_day,
  CASE
    WHEN LEAD(temperature) OVER (ORDER BY date_recorded) IS NULL THEN
'NA'
    ELSE TO_CHAR(LEAD(temperature) OVER (ORDER BY date_recorded) -
temperature)
  END AS diff_next_day
FROM daily_temp;
```

✅ Key Takeaways

Problem	Solution
LAG/LEAD produce NULL for first/last rows	Use `NVL()` or `COALESCE()`

Want zero difference for missing value	Replace NULL with current row value
Want text "NA"	Use `CASE WHEN … IS NULL THEN …`
Want to skip NULLs completely	Use `IGNORE NULLS` (if DB supports)

Ready-to-run Oracle SQL script for the **temperature trend problem using LAG and LEAD** — fully practical and easy to test in SQL Developer.

🌡️ Temperature Difference Analysis with LAG and LEAD

🏗️ Step 1: Create Table

```
CREATE TABLE daily_temp (
  date_recorded DATE PRIMARY KEY,
  temperature NUMBER
);
```

🌞 Step 2: Insert Sample 30 Days of Data

```
INSERT ALL
INTO daily_temp VALUES (TO_DATE('2025-10-01', 'YYYY-MM-DD'), 31)
INTO daily_temp VALUES (TO_DATE('2025-10-02', 'YYYY-MM-DD'), 33)
INTO daily_temp VALUES (TO_DATE('2025-10-03', 'YYYY-MM-DD'), 32)
INTO daily_temp VALUES (TO_DATE('2025-10-04', 'YYYY-MM-DD'), 34)
INTO daily_temp VALUES (TO_DATE('2025-10-05', 'YYYY-MM-DD'), 35)
INTO daily_temp VALUES (TO_DATE('2025-10-06', 'YYYY-MM-DD'), 36)
INTO daily_temp VALUES (TO_DATE('2025-10-07', 'YYYY-MM-DD'), 34)
INTO daily_temp VALUES (TO_DATE('2025-10-08', 'YYYY-MM-DD'), 33)
INTO daily_temp VALUES (TO_DATE('2025-10-09', 'YYYY-MM-DD'), 32)
INTO daily_temp VALUES (TO_DATE('2025-10-10', 'YYYY-MM-DD'), 31)
INTO daily_temp VALUES (TO_DATE('2025-10-11', 'YYYY-MM-DD'), 32)
INTO daily_temp VALUES (TO_DATE('2025-10-12', 'YYYY-MM-DD'), 33)
INTO daily_temp VALUES (TO_DATE('2025-10-13', 'YYYY-MM-DD'), 34)
INTO daily_temp VALUES (TO_DATE('2025-10-14', 'YYYY-MM-DD'), 35)
INTO daily_temp VALUES (TO_DATE('2025-10-15', 'YYYY-MM-DD'), 36)
INTO daily_temp VALUES (TO_DATE('2025-10-16', 'YYYY-MM-DD'), 37)
INTO daily_temp VALUES (TO_DATE('2025-10-17', 'YYYY-MM-DD'), 35)
INTO daily_temp VALUES (TO_DATE('2025-10-18', 'YYYY-MM-DD'), 34)
INTO daily_temp VALUES (TO_DATE('2025-10-19', 'YYYY-MM-DD'), 32)
INTO daily_temp VALUES (TO_DATE('2025-10-20', 'YYYY-MM-DD'), 31)
INTO daily_temp VALUES (TO_DATE('2025-10-21', 'YYYY-MM-DD'), 32)
INTO daily_temp VALUES (TO_DATE('2025-10-22', 'YYYY-MM-DD'), 33)
INTO daily_temp VALUES (TO_DATE('2025-10-23', 'YYYY-MM-DD'), 34)
INTO daily_temp VALUES (TO_DATE('2025-10-24', 'YYYY-MM-DD'), 33)
INTO daily_temp VALUES (TO_DATE('2025-10-25', 'YYYY-MM-DD'), 35)
INTO daily_temp VALUES (TO_DATE('2025-10-26', 'YYYY-MM-DD'), 36)
INTO daily_temp VALUES (TO_DATE('2025-10-27', 'YYYY-MM-DD'), 37)
INTO daily_temp VALUES (TO_DATE('2025-10-28', 'YYYY-MM-DD'), 36)
INTO daily_temp VALUES (TO_DATE('2025-10-29', 'YYYY-MM-DD'), 34)
INTO daily_temp VALUES (TO_DATE('2025-10-30', 'YYYY-MM-DD'), 32)
SELECT * FROM dual;

COMMIT;
```

🔍 Step 3: Analyze Temperature Difference (Handling NULLs)

```
SELECT
  TO_CHAR(date_recorded, 'YYYY-MM-DD') AS date_recorded,
  temperature,
  temperature - NVL(LAG(temperature) OVER (ORDER BY date_recorded),
temperature) AS diff_prev_day,
  NVL(LEAD(temperature) OVER (ORDER BY date_recorded), temperature) -
temperature AS diff_next_day
FROM daily_temp
ORDER BY date_recorded;
```

📊 Sample Output (first 10 rows)

DATE_RECORDED	TEMP	DIFF_PREV_DAY	DIFF_NEXT_DAY

2025-10-01	31	0	2
2025-10-02	33	2	-1
2025-10-03	32	-1	2
2025-10-04	34	2	1
2025-10-05	35	1	1
2025-10-06	36	1	-2
2025-10-07	34	-2	-1
2025-10-08	33	-1	-1
2025-10-09	32	-1	-1
2025-10-10	31	-1	1

(continues for all 30 days)

🧮 Step 4: Optional — Show ‘NA’ for Boundary Days

If you prefer **text output** instead of numeric zeros:

```
SELECT
  TO_CHAR(date_recorded, 'YYYY-MM-DD') AS date_recorded,
  temperature,
  CASE
    WHEN LAG(temperature) OVER (ORDER BY date_recorded) IS NULL THEN
'NA'
    ELSE TO_CHAR(temperature - LAG(temperature) OVER (ORDER BY
date_recorded))
  END AS diff_prev_day,
  CASE
    WHEN LEAD(temperature) OVER (ORDER BY date_recorded) IS NULL THEN
'NA'
    ELSE TO_CHAR(LEAD(temperature) OVER (ORDER BY date_recorded) -
temperature)
  END AS diff_next_day
FROM daily_temp
```

```
ORDER BY date_recorded;
```

🧠 Explanation

Concept	What It Does
`LAG(temp)`	Fetches previous day's temperature
`LEAD(temp)`	Fetches next day's temperature
`NVL(..., temperature)`	Replaces NULL with current day's value
`CASE WHEN`	Replaces NULL with 'NA' for clarity
`ORDER BY date_recorded`	Defines time sequence for window

✅ End Result

- **First day:** Difference from previous = `0` (or "NA")
- **Last day:** Difference to next = `0` (or "NA")
- **All others:** True difference values

Would you like me to extend this with a **rolling 3-day average temperature trend** using window functions (`AVG() OVER (...)`) to visualize how temperature fluctuates over time?