Manufacturing processes for any product is like putting together a puzzle. Products are pieced together step by step, and keeping a close eye on the process is important.

For this project, you're supporting a team that wants to improve how they monitor and control a manufacturing process. The goal is to implement a more methodical approach known as statistical process control (SPC). SPC is an established strategy that uses data to determine whether the process works well. Processes are only adjusted if measurements fall outside of an acceptable range.

This acceptable range is defined by an upper control limit (UCL) and a lower control limit (LCL), the formulas for which are:

$$ucl = avg\_height + 3 * \frac{stddev\_height}{\sqrt{5}}$$

$$lcl = avg\_height - 3 * \frac{stddev\_height}{\sqrt{5}}$$

The UCL defines the highest acceptable height for the parts, while the LCL defines the lowest acceptable height for the parts. Ideally, parts should fall between the two limits.

Using SQL window functions and nested queries, you'll analyze historical manufacturing data to define this acceptable range and identify any points in the process that fall outside of the range and therefore require adjustments. This will ensure a smooth running manufacturing process consistently making high-quality products.

The data

The data is available in the `manufacturing_parts` table which has the following fields:

- `item_no` : the item number
- `length` : the length of the item made
- `width` : the width of the item made
- `height` : the height of the item made
- `operator` : the operating machine

Projects Data DataFrame as alerts

```
WITH calculated_heights AS (
SELECT
    operator,
    ROW_NUMBER() OVER (
        PARTITION BY operator ORDER BY item_no) AS row_number,
    height,
    AVG(height) OVER (
        ORDER BY item_no ASC
        ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) AS avg_height,
    stddev_samp(height) OVER (
        ORDER BY item_no ASC
        ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) as stddev_height
FROM manufacturing_parts
),
control_limits AS (
SELECT
    *,
    avg_height + 3 * stddev_height/SQRT(5) AS ucl,
    avg_height - 3 * stddev_height/SQRT(5) AS lcl
FROM calculated_heights
WHERE row_number >= 5
)
SELECT
    *,
    CASE
        WHEN (height NOT BETWEEN lcl AND ucl) THEN TRUE
        ELSE FALSE
    END AS alert
FROM control_limits
```

index	operator	row_number	height	avg_height	stddev_height	ucl
0	Op-1	5	19.46	19.778	1.062812307	
1	Op-1	6	20.36	19.912	1.0908116244	
2	Op-1	7	20.22	20.03	1.084573649	
3	Op-1	8	21.03	19.934	0.9312249997	
4	Op-1	9	19.78	20.17	0.5988321969	
5	Op-1	10	20.71	20.42	0.4768123321	
6	Op-1	11	20.62	20.472	0.4827732387	
7	Op-1	12	19.51	20.33	0.6506535176	
8	Op-1	13	20.06	20.136	0.5215649528	
9	Op-1	14	20.3	20.24	0.4832701108	
10	Op-1	15	20.25	20.148	0.4095973633	
11	Op-1	16	20.52	20.128	0.3823218539	
12	Op-1	17	19.33	20.092	0.4563660811	
13	Op-1	18	19.12	19.904	0.6324792487	
14	Op-1	19	19.37	19.718	0.6235944195	
15	Op-1	20	18.8	19.428	0.6508993778	

Rows: 420 Expand