【ウィンドウ関数を極める（その２）】

④順位付けをする（その４）

前回までの順位付けは、直接順位を付けていましたが、今回は、相対的な位置を求めるPERCENT_RANK関数と累積分布値を求めるCUME_DIST関数（cumeは累積、distはdistributionで分布）を説明します。

最初に、PERCENT_RANK関数は、上位何%に位置付けられているかを求める関数であり、最高位が0、最下位が1になるようにするため、以下の式から求めます。

PERCENT_RANK＝（順位から１を引く）÷（グループ化した行の総数から１を引く）

次に、CUME_DIST関数ですが、最下位は1ですが、最高位が0にならないように、以下の式により求めます。こちらの方が馴染みやすいと思います。

CUME_DIST＝（順位）÷（グループ化した行の総数）

そして、先の投稿の再掲になりますが、RANK関数を使用した以下のSELECT文に対して、赤字部分を緑字部分に変えれば、PERCENT_RANK関数とCUME_DIST関数を使うことができます。まず、左の下表（勤怠表）に対して、PERCENT_RANK関数を使用した結果が右の下表（実行結果）です。だいぶ慣れてきたと思うので、詳細は省略します。

SELECT 係、職員、労働時間、RANK（　）OVER（PARTITION BY 係 ORDER BY 労働時間 DESC）AS 順位
FROM 勤怠表　ORDER BY 係 DESC、労働時間 DESC

PERCENT_RANK関数を使用するためには、**RANK（　）をPERCENT_RANK（　）、順位を相対位置**
CUME_DIST関数を使用するためには、**RANK（　）をCUME_DIST（　）、順位を累積分布**

表：勤怠表

係	職員	労働時間
総務係	A	150
総務係	B	140
総務係	C	140
総務係	D	100
総務係	E	120
総務係	F	200
人事係	G	90
人事係	H	130
人事係	I	140

表：実行結果

係	職員	労働時間	相対順位	
総務係	F	200	0	総務係の行の総数は6
総務係	A	150	0.2	← **(1-1) ÷ (6-1)**
総務係	B	140	0.4	← **(2-1) ÷ (6-1)**
総務係	C	140	0.4	← **(3-1) ÷ (6-1)**
総務係	E	120	0.8	← **(3-1) ÷ (6-1)**
総務係	D	100	1	← **(5-1) ÷ (6-1)**
人事係	I	140	0	← **(6-1) ÷ (6-1)**
人事係	H	130	0.5	
人事係	G	90	1	

本資料は正確性に欠く部分及び誤字脱字等も多いと思います。そのため、本資料に起因した損害等については、管理人として責任を負いかねますので御了承ください。
© 2025 spine-team.com

次に、左の下表（勤怠表）に対して、 CUME_DIST関数を使用した結果が右の下表（実行結果）です。 PERCENT_RANK関数との違いは、ピア行の措置です。

表：勤怠表

係	職員	労働時間
総務係	A	150
総務係	B	140
総務係	C	140
総務係	D	100
総務係	E	120
総務係	F	200
人事係	G	90
人事係	H	130
人事係	I	140

表：実行結果

係	職員	労働時間	相対順位	
総務係	F	200	0.1666…	← 1 ÷ 6
総務係	A	150	0.3333…	← 2 ÷ 6
総務係	B	140	0.6666…	← 4 ÷ 6
総務係	C	140	0.6666…	← 4 ÷ 6
総務係	E	120	0.8333…	← 5 ÷ 6
総務係	D	100	1	← 6 ÷ 6
人事係	I	140	0	
人事係	H	130	0.5	
人事係	G	90	1	

総務係の行の総数は6

ピア行がある場合の措置

本資料は正確性に欠く部分及び誤字脱字等も多いと思います。そのため、本資料に起因した損害等については、管理人として責任を負いかねますので御了承ください。
© 2025 spine-team.com

⑤他の行の値を利用する

複数回にわたり説明してきた順位付けに関するウィンドウ関数は、終わりにして、今回は、他の行との比較を行うために、他の行の値を真横に表示するLAG関数とLEAD関数について説明します。

具体的に説明すると、左の下表（職員別月間労働時間）に対して、以下のSELECT文を実行すると、右の下表（実行結果）のとおり、各行に2ヶ月前の労働時間が表示されます。今回は、PARTITION句の後のORDER BY句にはASCやDESCがないため、自動的にASC（昇順：小さい順）になります。また、途中経過を並べ替える必要がないため、2つ目のORDER BY句がありません。

```
SELECT 職員、年月、労働時間、LAG（労働時間,2）OVER（PARTITION BY 職員 ORDER BY 年月）
AS 前々月の労働時間　FROM 職員別月間労働時間
```

LAG（労働時間,2）をの2は2行前の意味なので、前月を表示したい場合は**LAG（労働時間,1）**であるが、1は省略できるため、**LAG（労働時間）**でOK

LEAD関数は、**LAG関数**の逆で、**LEAD（労働時間,3）**ならば、**3行後**になるだけであり、使い方はLAG関数と同じである。

表：職員別月間労働時間

職員	年月	労働時間
A	2024年1月	200
A	2024年2月	150
A	2024年3月	140
A	2024年4月	140
A	2024年5月	120
B	2024年1月	100
B	2024年2月	140
B	2024年3月	130
B	2024年4月	90
B	2024年5月	120

表：実行結果

職員	年月	労働時間	前々月の労働時間	
A	2024年1月	200	NULL	前々月がないのでNULLを設定
A	2024年2月	150	NULL	
A	2024年3月	140	200	
A	2024年4月	140	150	
A	2024年5月	120	140	
B	2024年1月	100	NULL	前々月がないのでNULLを設定
B	2024年2月	140	NULL	
B	2024年3月	130	100	
B	2024年4月	90	140	
B	2024年5月	120	130	

本資料は正確性に欠く部分及び誤字脱字等も多いと思います。そのため、本資料に起因した損害等については、管理人として責任を負いかねますので御了承ください。
© 2025 spine-team.com

⑥移動平均を求める

長かったウィンドウ関数はようやく最後です。今回は、移動平均を求めます。説明を簡単にするため、グループ分けをせずに表全体で考えます。つまり、PARTITION句は使いません。具体的には、左の下表（売上表）に対して、以下のSELECT文を実行すると、3ヶ月の移動平均を得ることができます。また、途中経過を並べ替える必要がないため、2つ目のORDER BY句は不要です。

SELECT 年月、月間売上、AVG（月間売上）OVER（ORDER BY 年月 **ROWS BETWEEN 2 PRECEDING AND CURRENT ROW**）AS 移動平均値　FROM 売上表

ROWS BETWEEN 2 PRECEDING AND CURRENT ROWは、「**行単位**で現在の行より2つ前の行から現在の行まで」の意味であり、2を3にすると、4ヶ月平均になってしまう。また、２をUNBOUNDEDにすると、先頭行から現在の行までの平均となる。

先頭の**ROWS**を**RANGE**にすると「現在の2月前から現在まで」となり、**行単位**ではなく、値（数値）で判断する。つまり、左の下表（売上表）において、7月の行が欠落していて、8月を基準にした場合、ROWSは7月の欠落に気づかずに、2つ前の行を辿り、5月から8月までの4ヶ月平均を算出してしまうが、RANGEは7月を抜かした6月と8月で3ヶ月平均を算出してくれる。

表：売上表

年月	月間売上
2024年1月	200
2024年2月	150
2024年3月	140
2024年4月	140
2024年5月	120
2024年6月	100
2024年7月	140
2024年8月	130
2024年9月	90
2024年10月	120

表：実行結果

年月	月間売上	移動平均値	
2024年1月	200	200	← 3ヶ月分ないため1月の値
2024年2月	150	175	← 3ヶ月分ないため1月と2月の平均値
2024年3月	140	163	← 1月から3月の3ヶ月間の平均値
2024年4月	140	143	← 2月から4月の3ヶ月間の平均値
2024年5月	120	133	
2024年6月	100	120	
2024年7月	140	120	
2024年8月	130	123	
2024年9月	90	120	
2024年10月	120	113	

以下同じ

本資料は正確性に欠く部分及び誤字脱字等も多いと思います。そのため、本資料に起因した損害等については、管理人として責任を負いかねますので御了承ください。
© 2025 spine-team.com