



# 君はWindow関数 を知っている か！？

テクテクテック#7 DB勉強会  
2019/01/16

@masayuki14

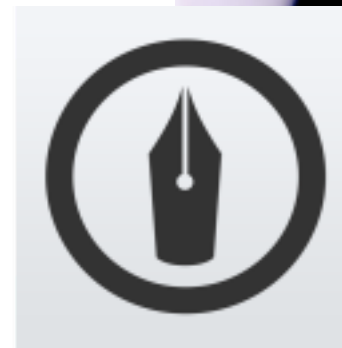
# 目次



- ✓ 自己紹介
- ✓ Window関数とは
- ✓ 実践Window関数
- ✓ まとめ

# 自己紹介

✓ もりさきまさゆき  
✓ @masayuki14



**Follow me !!**

# 自己紹介

✓ プロ主夫

✓ 妻1人 (フルタイム) 幼児2人

✓ フリーランス (Web系パートタイム)

✓ データベーススペシャリスト

✓ コミュニティ

✓ はんなりPython (第3金曜開催@京都)

✓ OSS Gate (京都, 大阪, 東京)

✓ スピーキーズアンバサダー

# スプーキーズ@京都



- ✓ Web系システム
- ✓ ソーシャルゲーム開発
- ✓ ボードゲーム制作



**Webエンジニア積極採用中！！**

# スプーキーズ@京都



勉強会 テクテクテック

✓ 2019/04/10

✓ サーバ監視や負荷テストどうやって  
るの？

✓ もくもく会はじめました

✓ 2/28(木) 19:00～

✓ いつでも来てね!!



# Window関数とは



# Window関数と は

# Window関数とは



標準SQL規格

SQL:2003にて追加された  
標準機能



# Window関数とは



MySQL8.0でサポート  
主要RDBMSで利用可能

✓ Oracle, SQL Server

✓ PostgreSQL, MySQL

# Window関数とは



# Window関数と は

# Window関数とは



Window + 関  
数 ??

# Window関数とは



## Window の意味は？

---

✓ × 窓

✓ ○ 範囲 幅

# Window関数とは



## 範囲を指定して

- ✓ 違う行を自分の行にもってくる
- ✓ 集約結果を自分の行にもってくる

# Window関数とは



どうやって範囲  
を指定する？

# Window関数とは



# Window関数を使っ

# Window関数とは



SQLを見てみよう



# Window関数とは



## Window関数を使ったSQLの例

```
SELECT
    customer_id,
    amount,
    RANK() OVER (
        PARTITION BY customer_id
        ORDER BY amount DESC) AS rk
FROM payment;
```

# Window関数とは



実はこれの省略形(名前付き構文)

```
SELECT
    customer_id,
    amount,
    RANK() OVER w AS rk
FROM payment
WINDOW w AS (PARTITION BY customer_id
              ORDER BY amount DESC);
```

# Window関数とは



## Windowの使い回し

```
SELECT
  customer_id,
  amount,
  RANK()          OVER w AS rk,
  MAX(amount)     OVER w AS max
FROM payment
WINDOW w AS (PARTITION BY customer_id
              ORDER BY amount DESC);
```

# Window関数とは



## Window関数の書き方

```
SELECT  
    集約関数() OVER (範囲指定)  
FROM table
```

OVER() があればWindow関数

# Window関数とは



## Window関数の書き方

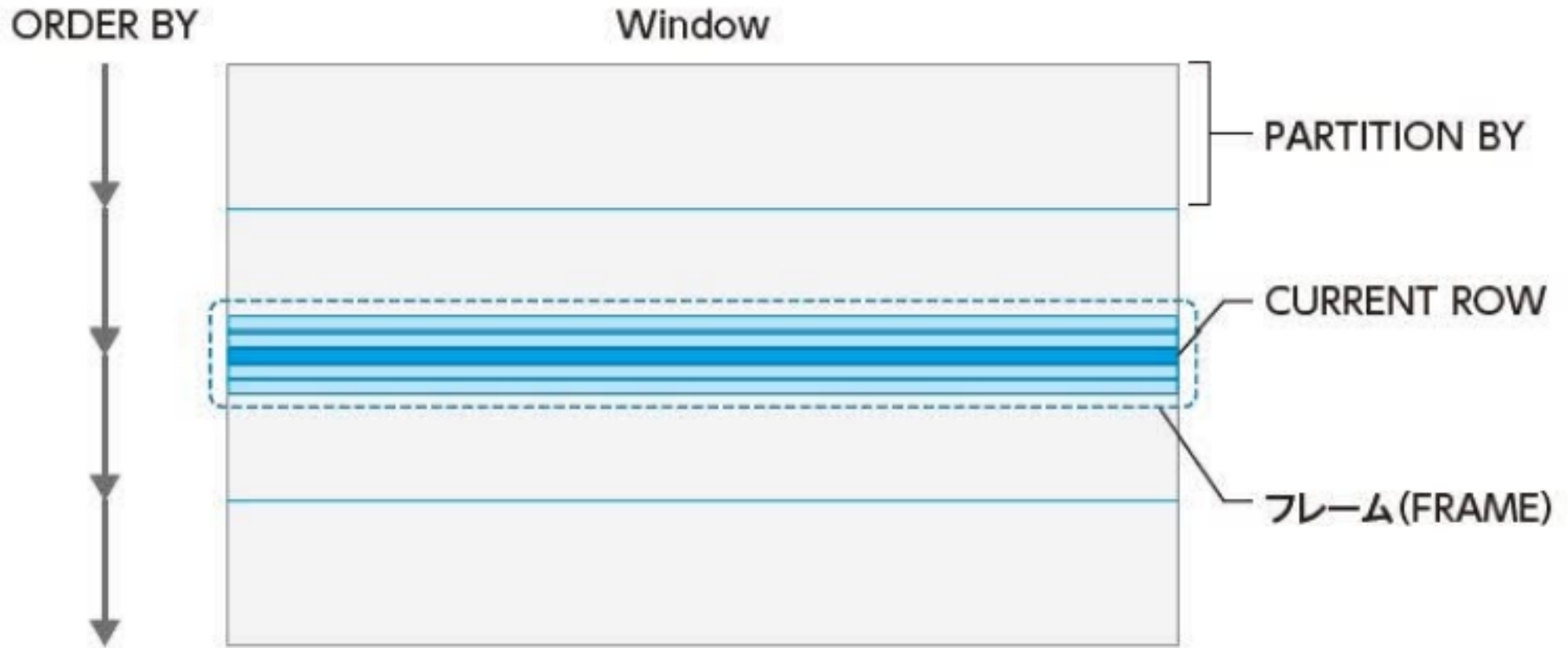
### 集約関数

- ✓ ROW\_NUMBER(), FIRST\_VALUE()
- ✓ MAX(), MIN(), AVG()

### 範囲指定

- ✓ PARTITION BY, ORDER BY
- ✓ フレーム句

# Window関数とは



1枚でわかる Window関数 - <https://codezine.jp/article/detail/11115>

# Window関数とは



## Window関数の3つの機能

- ✓ PARTITION BY 句によるレコード集合の分割
- ✓ ORDER BY 句によるレコードの順序づけ
- ✓ フレーム句によるカレントレコードを中心としたサブセット定義

# 実践Window関数



# 実践Window関 数



# 実践Window関数



まずは  
PARTITION BY  
を  
おさえよう

# 実践Window関数



## 単純な例

teams table

member	team	age
富士崎	A	39
西塚	A	45
西崎	A	24
逸見	B	26
岡村	B	18
東野	C	50
各務原	D	27
犬山	D	28
鳥羽	D	33
桃山	D	28

# 実践Window関数



team ごとに分割して行番号を取得する

```
SELECT
    member,
    team,
    age,
    ROW_NUMBER() OVER (
        PARTITION BY team
        ORDER BY age) AS num
FROM teams;
```

# 実践Window関数



team ごとに分割して行番号を取得する

member	team	age	num
西崎	A	24	1
富士崎	A	39	2
西塚	A	45	3
岡村	B	18	1
逸見	B	26	2
東野	C	50	1
各務原	D	27	1
桃山	D	28	2
犬山	D	28	3
鳥羽	D	33	4

# 実践Window関数



team ごとに分割して順位を取得する

```
SELECT
    member,
    team,
    age,
    RANK() OVER (
        PARTITION BY team
        ORDER BY age) AS rk
FROM teams;
```

# 実践Window関数

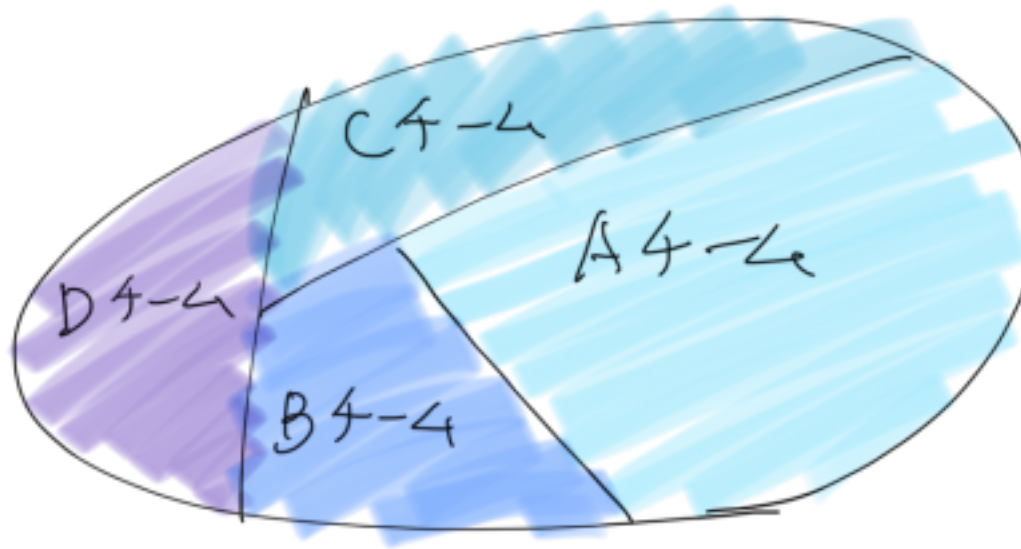


team ごとに分割して順位を取得する

member	team	age	rk
西崎	A	24	1
富士崎	A	39	2
西塚	A	45	3
岡村	B	18	1
逸見	B	26	2
東野	C	50	1
各務原	D	27	1
犬山	D	28	2
桃山	D	28	2
鳥羽	D	33	4

# 実践Window関数

PARTITION BY で部分集合を  
作って  
その中で集約関数使う



# 実践Window関数



少し込み入った例

records table

team	competitor	score	is_win	played_at
A	B	10 - 3	1	2018-11-20
A	C	10 - 8	1	2018-11-24
A	B	8 - 5	1	2018-10-05
A	D	3 - 9	0	2018-09-28
A	C	4 - 5	0	2018-12-04
A	B	7 - 9	0	2018-11-05
A	D	20 - 17	1	2018-12-06
A	D	12 - 11	0	2018-10-14
A	B	8 - 4	1	2018-09-23
A	E	8 - 0	1	2018-12-28
A	E	6 - 0	1	2019-01-06



# 実践Window関数



チームごとに対戦成績をみたい

```
SELECT
```

```
    *
```

```
FROM
```

```
    records
```

```
ORDER BY
```

```
    competitor, played_at;
```

# 実践Window関数



チームごとに対戦成績をみたい

team	competitor	score	is_win	played_at
A	B	8 - 4	1	2018-09-23
A	B	8 - 5	1	2018-10-05
A	B	7 - 9	0	2018-11-05
A	B	10 - 3	1	2018-11-20
A	C	10 - 8	1	2018-11-24
A	C	4 - 5	0	2018-12-04
A	D	3 - 9	0	2018-09-28
A	D	12 - 11	0	2018-10-14
A	D	20 - 17	1	2018-12-06
A	E	8 - 0	1	2018-12-28
A	E	6 - 0	1	2019-01-06

# 実践Window関数



チームごとに最新の対戦成績をみたい

```
SELECT
    team, competitor,
    score, is_win, played_at,
    ROW_NUMBER() OVER (
        PARTITION BY competitor
        ORDER BY played_at DESC
    ) AS rn
FROM
    records;
```

# 実践Window関数



チームごとに**最新の**対戦成績をみたい

team	competitor	score	is_win	played_at	rn
A	B	10 - 3	1	2018-11-20	1
A	B	7 - 9	0	2018-11-05	2
A	B	8 - 5	1	2018-10-05	3
A	B	8 - 4	1	2018-09-23	4
A	C	4 - 5	0	2018-12-04	1
A	C	10 - 8	1	2018-11-24	2
A	D	20 - 17	1	2018-12-06	1
A	D	12 - 11	0	2018-10-14	2
A	D	3 - 9	0	2018-09-28	3
A	E	6 - 0	1	2019-01-06	1
A	E	8 - 0	1	2018-12-28	2

# 実践Window関数



チームごとに**最新の**対戦成績をみたい

```
SELECT * FROM (
    SELECT
        team, competitor,
        score, is_win, played_at,
        ROW_NUMBER() OVER (
            PARTITION BY competitor
            ORDER BY played_at DESC
        ) AS rn
    FROM
        records
) tbl
WHERE rn = 1;
```

# 実践Window関数



チームごとに**最新の**対戦成績をみたい

team	competitor	score	is_win	played_at	rn
A	B	10 - 3	1	2018-11-20	1
A	C	4 - 5	0	2018-12-04	1
A	D	20 - 17	1	2018-12-06	1
A	E	6 - 0	1	2019-01-06	1

# 実践Window関数



Window関数を使うことで  
部分集合の結果を  
自分の行に持ってくるこ  
とができる

# 実践Window関数



# 実践Window関 数



# 実践Window関数



フレーム句を使  
い  
違う行を持って  
くる

# 実践window関数



## 売上データ

```
sales table
+-----+-----+
| date           | sales |
+-----+-----+
| 2019-01-01     | 1930  |
| 2019-01-02     | 2030  |
| 2019-01-05     | 1720  |
| 2019-01-07     | 2290  |
| 2019-01-08     | 1660  |
| 2019-01-10     | 2450  |
+-----+-----+
```

# 実践Window関数



過去の直近の日付を取得する

```
SELECT
    date AS cur_date,
    sales AS cur_sales,
    MIN(date) OVER (
        ORDER BY date
        ROWS BETWEEN 1 PRECEDING AND 1 PRECEDING
    ) AS latest_date
FROM sales;
```

# 実践Window関数



過去の直近の日付を取得する

cur_date	cur_sales	latest_date
2019-01-01	1930	NULL
2019-01-02	2030	2019-01-01
2019-01-05	1720	2019-01-02
2019-01-07	2290	2019-01-05
2019-01-08	1660	2019-01-07
2019-01-10	2450	2019-01-08

# 実践Window関数



過去の直近の日付と売上を取得する

```
SELECT
    date AS cur_date,
    sales AS cur_sales,
    MIN(date) OVER W AS latest_date,
    MIN(sales) OVER W AS latest_sales
FROM sales
WINDOW W AS (
    ORDER BY date
    ROWS BETWEEN 1 PRECEDING AND 1 PRECEDING
);
```

# 実践Window関数



過去の直近の日付と売上を取得する

cur_date	cur_sales	latest_date	latest_sales
2019-01-01	1930	NULL	NULL
2019-01-02	2030	2019-01-01	1930
2019-01-05	1720	2019-01-02	2030
2019-01-07	2290	2019-01-05	1720
2019-01-08	1660	2019-01-07	2290
2019-01-10	2450	2019-01-08	1660

# 実践Window関数



1日前の売上を取得する

```
SELECT
  date AS cur_date,
  sales AS cur_sales,
  MIN(date) OVER W AS latest_date,
  MIN(sales) OVER W AS latest_sales
FROM sales
WINDOW W AS (
  ORDER BY date
  RANGE BETWEEN INTERVAL 1 day PRECEDING
    AND INTERVAL 1 day PRECEDING
);
```

# 実践Window関数



1日前の売上を取得する

cur_date	cur_sales	latest_date	latest_sales
2019-01-01	1930	NULL	NULL
2019-01-02	2030	2019-01-01	1930
2019-01-05	1720	NULL	NULL
2019-01-07	2290	NULL	NULL
2019-01-08	1660	2019-01-07	2290
2019-01-10	2450	NULL	NULL



# 実践Window関数



2日後の売上を取得する

```
SELECT
  date AS cur_date,
  sales AS cur_sales,
  MIN(date) OVER W AS latest_date,
  MIN(sales) OVER W AS latest_sales
FROM sales
WINDOW W AS (
  ORDER BY date
  RANGE BETWEEN INTERVAL 2 day FOLLOWING
  AND INTERVAL 2 day FOLLOWING
);
```

# 実践Window関数



2日後の売上を取得する

cur_date	cur_sales	latest_date	latest_sales
2019-01-01	1930	NULL	NULL
2019-01-02	2030	NULL	NULL
2019-01-05	1720	2019-01-07	2290
2019-01-07	2290	NULL	NULL
2019-01-08	1660	2019-01-10	2450
2019-01-10	2450	NULL	NULL

# 実践Window関数



フレーム句を使うと  
違う行を現在の行にもっ  
てくることができる

# 実践Window関数



## フレーム句で利用するキーワード

---

- ✓ **ROWS**: 行単位で移動する
- ✓ **RANGE**: 列の値で移動する ORDER BY 句の列
- ✓ **n PRECEDING**: n だけ前へ移動
- ✓ **n FOLLOWING**: n だけ後ろへ移動
- ✓ **CURRENT ROW**: 現在行

# 実践Window関数



## 応用例

- ✓ 前日との売上を比較
- ✓ 3日間の移動平均
- ✓ チームごとの平均勝率

# まとめ



# まとめ

# まとめ



## Window 関数でできること

---

- ✓ 部分集合の値を持ってくる
- ✓ 違う行の値を持ってくる

わかりにくいサブクエリーにさよなら

# まとめ



ありがとうございました