SQLで移動平均を算出

2020/01/08
平均値格納行がおかしかったので修正。

2020/01/08
職場のTRSHM君に煽られたんでちょっと書く。

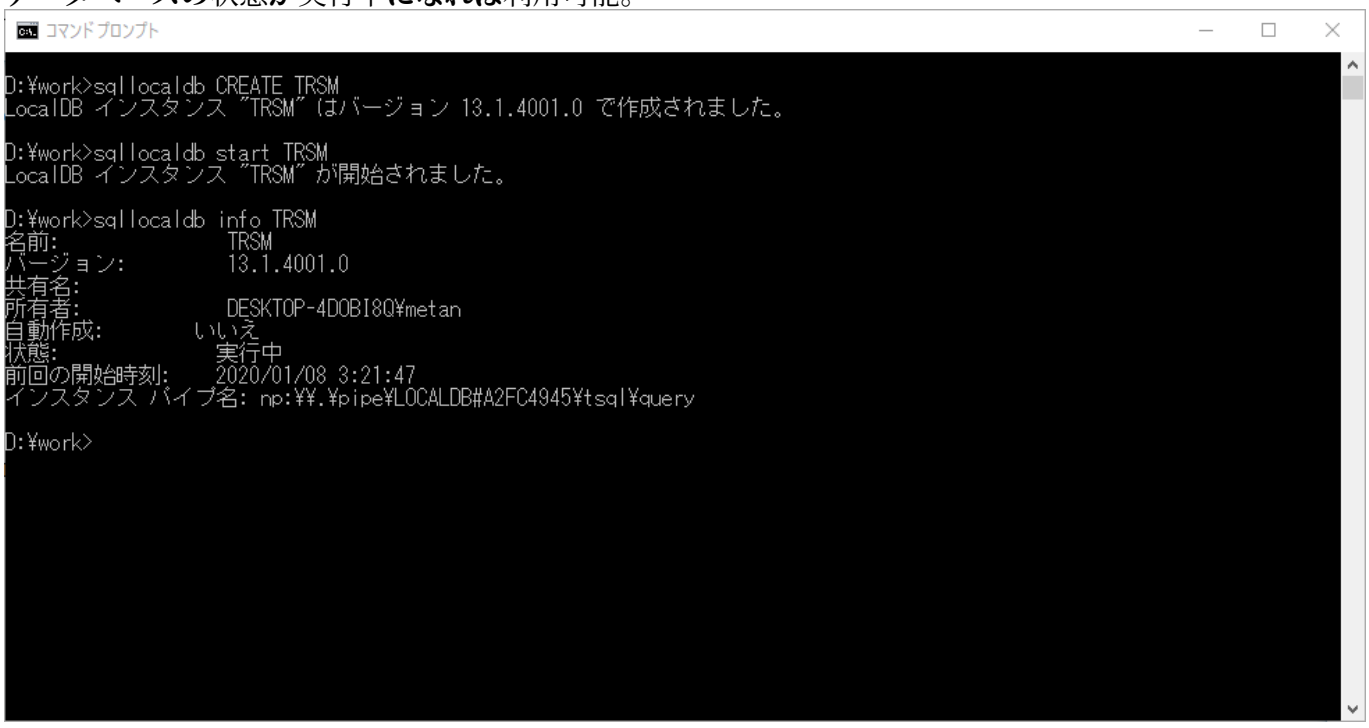
概要

- 直近3日の移動平均を出すサンプルコードを掲示する。
- ウインドウ関数を利用する。サブクエリを連ねるのはさすがに古い。

データベース準備

Visual Studioをインストールしていれば多分SQLServer LocalDBが使えると思うので、こちらを利用する。

データベースの状態が実行中になれば利用可能。

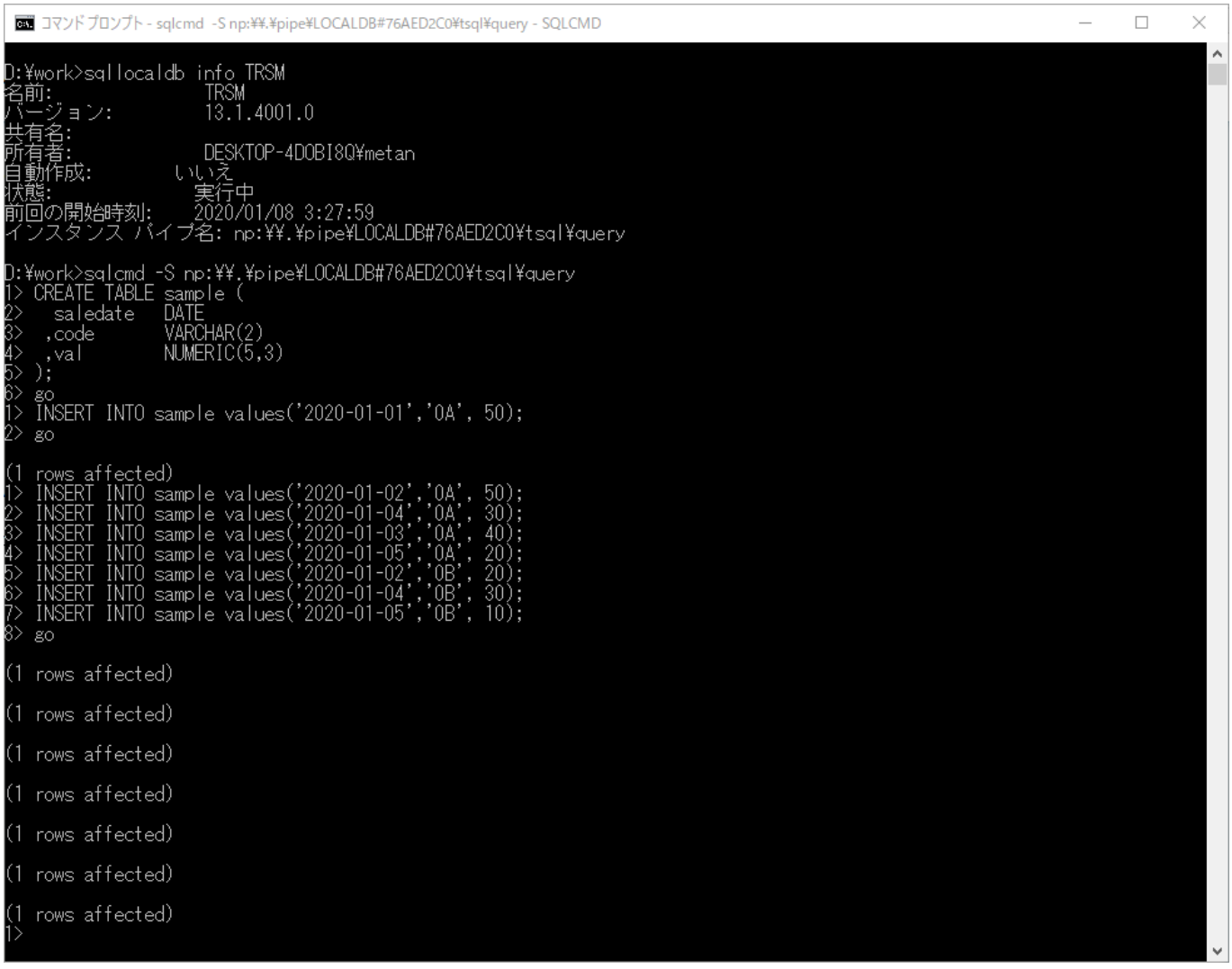


テーブルとデータの用意

sqlcmdでインスタンスパイプ名を指定して接続。あとは普通にSQLを実行してデータを定義する。

```
CREATE TABLE sample (
  saledate   DATE
 ,code       VARCHAR(2)
```

```
 ,val         NUMERIC(5,3)
);
GO
INSERT INTO sample VALUES('2020-01-01','0A', 50);
GO
INSERT INTO sample VALUES('2020-01-02','0A', 50);
INSERT INTO sample VALUES('2020-01-04','0A', 30);
INSERT INTO sample VALUES('2020-01-03','0A', 40);
INSERT INTO sample VALUES('2020-01-05','0A', 20);
INSERT INTO sample VALUES('2020-01-02','0B', 20);
INSERT INTO sample VALUES('2020-01-04','0B', 30);
INSERT INTO sample VALUES('2020-01-05','0B', 10);
GO
```



OVER句に対象レコードの範囲を指定する

PARTITION BY句で範囲を決め、ORDER BY句でレコードの並びを決めたのちROWS句で処理対象レコードを明示する。
以下のクエリだと、

- ~~2つ前のレコード~~ 1つ前のレコード

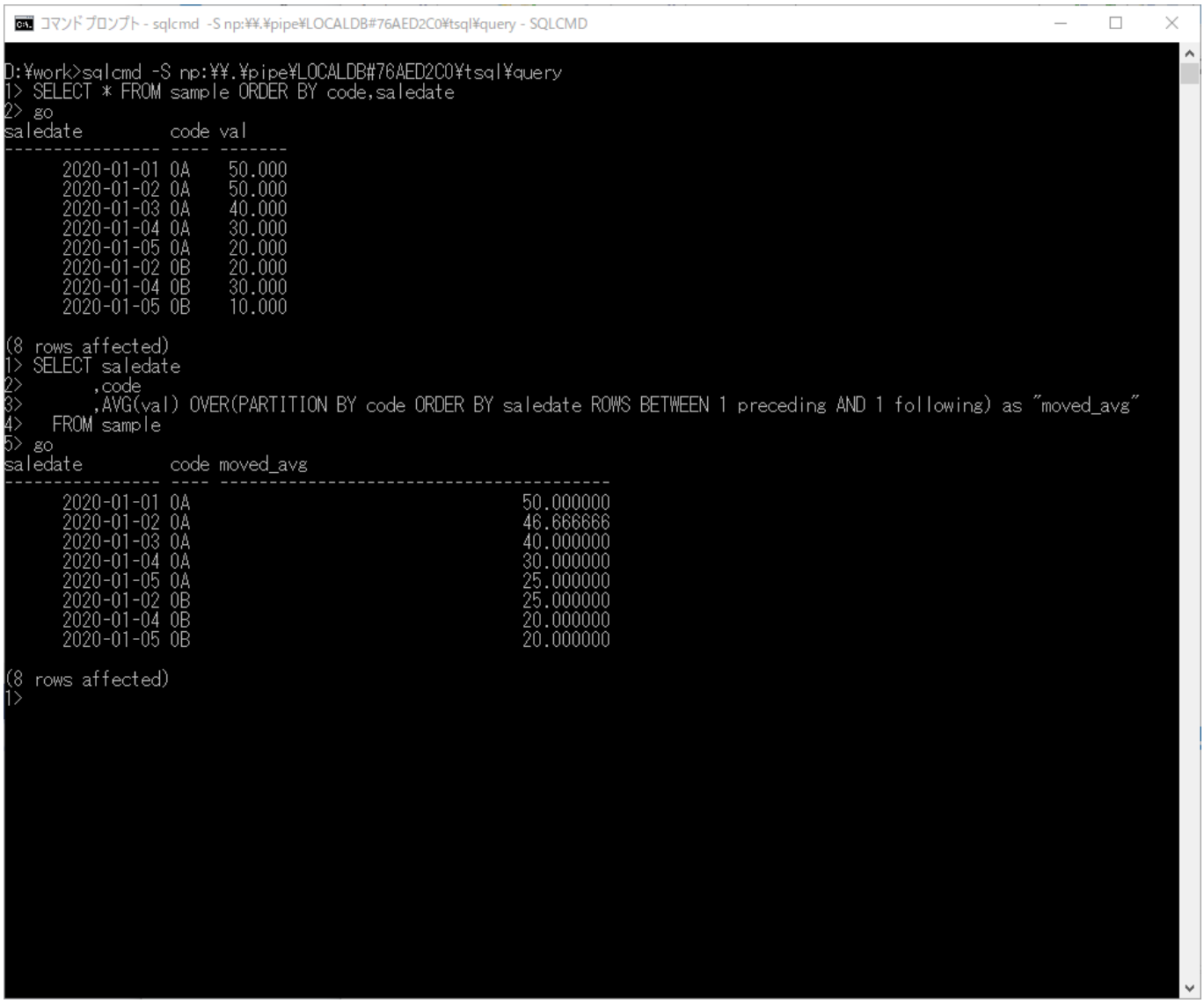
- ~~1つ前のレコード~~ 今選択されているレコード
- 今選択~~されているレコード~~ 1つ後のレコード

の3レコードが処理対象になる。日付でソートして“□日前”や“□日後”をとれるようにすること。

```
SELECT saledate
      ,code
      ,AVG(val) OVER(PARTITION BY code ORDER BY saledate ROWS BETWEEN 1
preceding AND 1 following) AS "moved_avg"
  FROM sample
GO
```

ただ、単純にこんなのを書くとはまる。
AVG()で指定する項目がNULLだった場合、そのレコードのカウントが入らないので、例えばcode='0A'の2020-01-01 の“moved_avg”は 33.333 ではなく50.000になってしまう。
※NULL, 50, 50 となるので、100÷3ではなく100÷2になる。

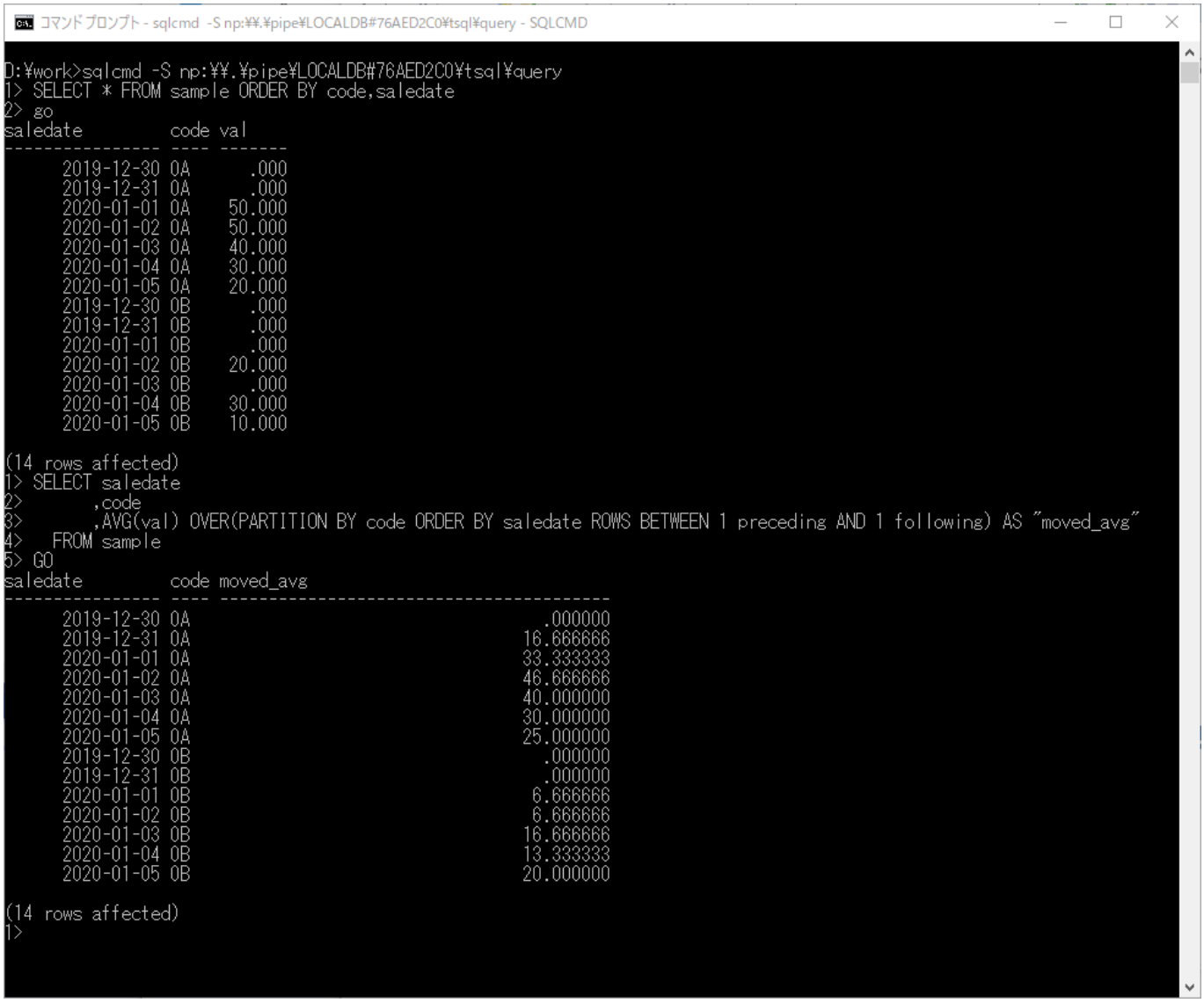
code='0B'なんか歯抜けがあるのでこれも正しくない。



NULL値の対応を入れる

欠損レコードを補いNULL値の影響を消し込む。下手な小細工するより早い。

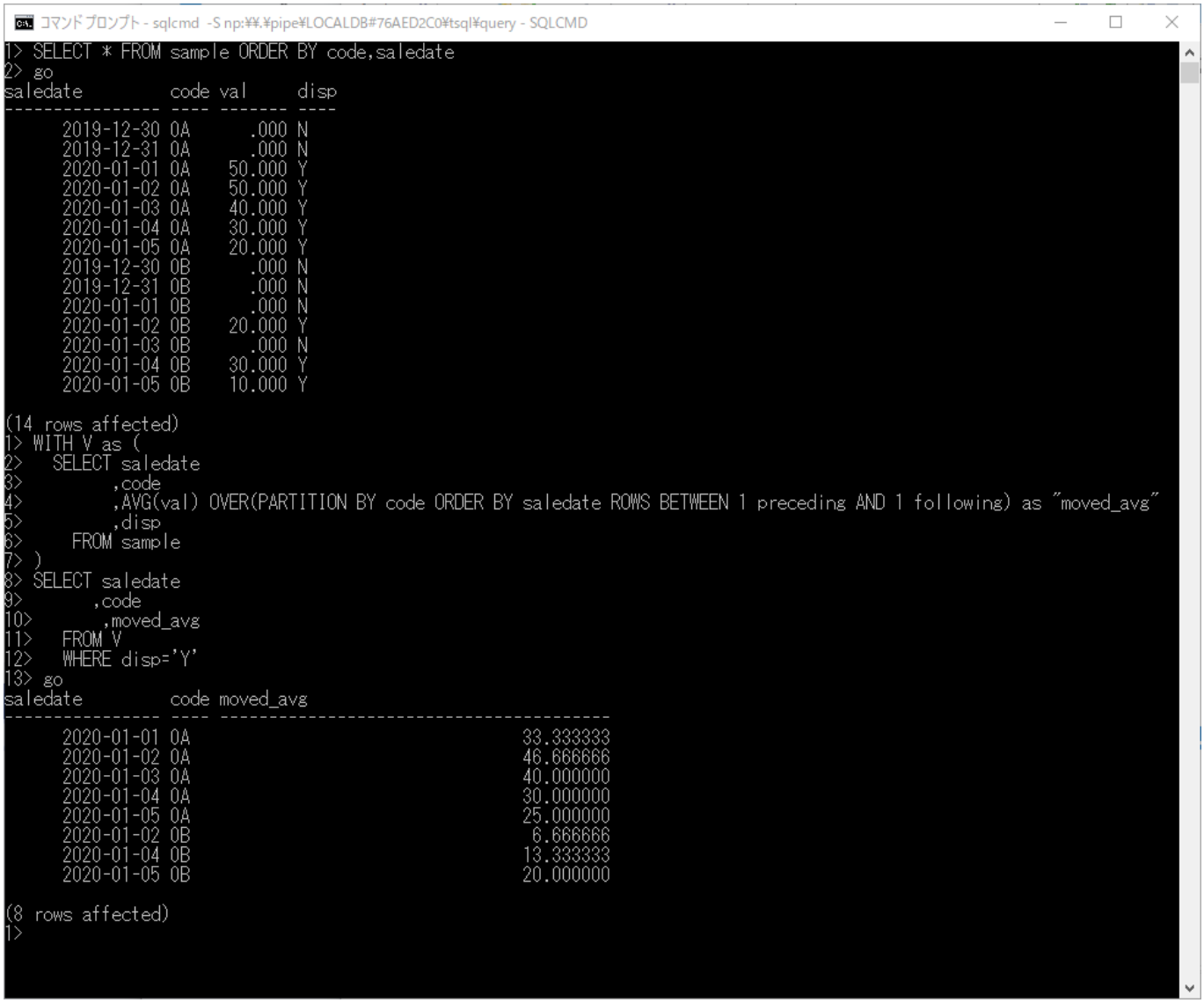
```
INSERT INTO sample VALUES('2019-12-31','0A',0);
INSERT INTO sample VALUES('2019-12-30','0A',0);
INSERT INTO sample VALUES('2019-12-31','0B',0);
INSERT INTO sample VALUES('2019-12-30','0B',0);
INSERT INTO sample VALUES('2020-01-01','0B',0);
INSERT INTO sample VALUES('2020-01-03','0B',0);
GO
SELECT saledate
      ,code
      ,AVG(val) OVER(PARTITION BY code ORDER BY saledate ROWS BETWEEN 1
preceding AND 1 following) AS "moved_avg"
  FROM sample
GO
```



ダミーレコードは出したくない

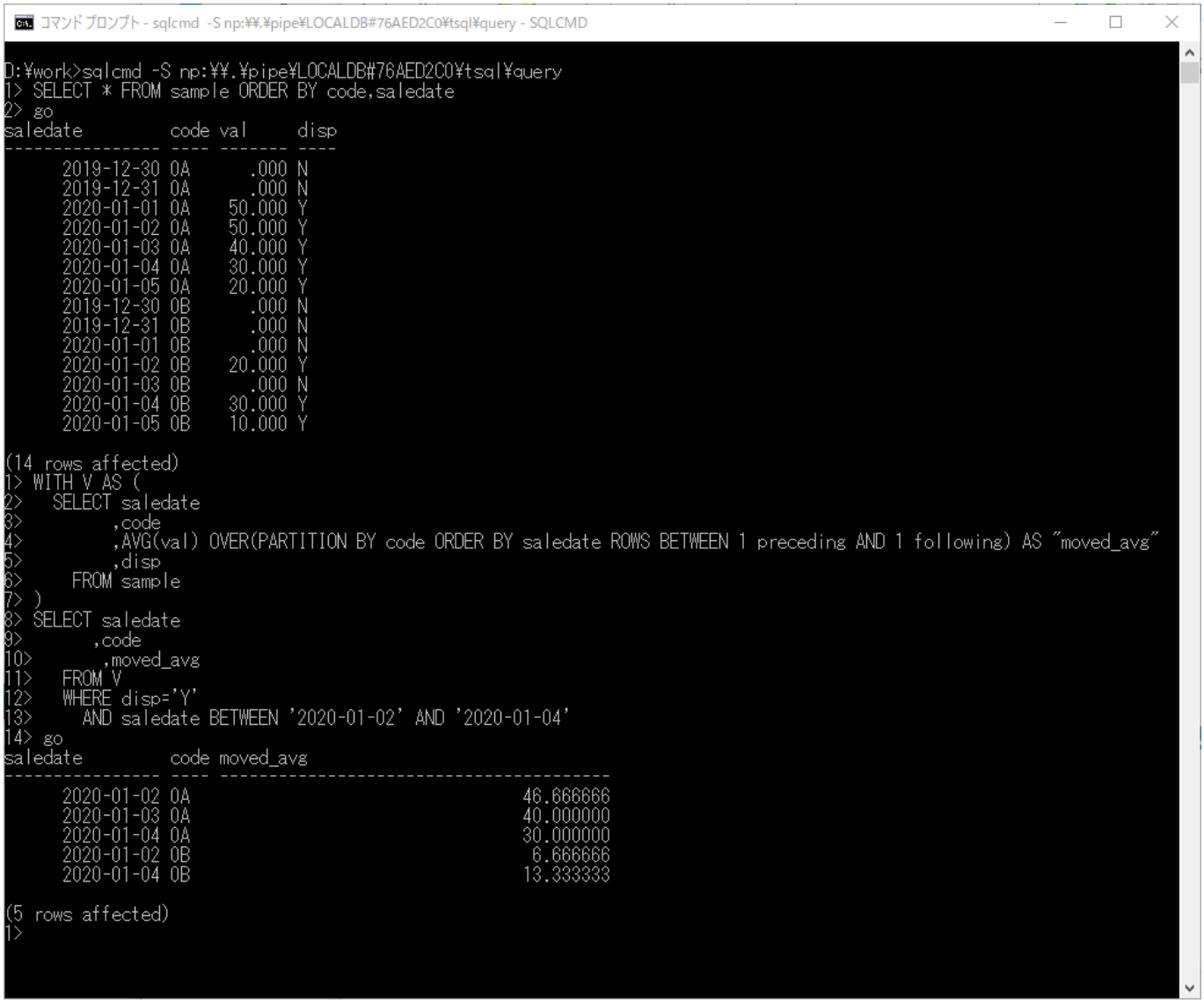
ダミーレコードを出したくないならテーブルを拡張して対応すればいいと思う。
この例だと計算結果の行セットVを作り、ここからダミー行を除いている。

```
DROP TABLE sample
GO
CREATE TABLE sample (
  saledate   DATE
 ,code       VARCHAR(2)
 ,val        NUMERIC(5,3)
 ,disp       VARCHAR(1)
);
GO
INSERT INTO sample VALUES('2020-01-01','0A', 50, 'Y');
INSERT INTO sample VALUES('2020-01-02','0A', 50, 'Y');
INSERT INTO sample VALUES('2020-01-04','0A', 30, 'Y');
INSERT INTO sample VALUES('2020-01-03','0A', 40, 'Y');
INSERT INTO sample VALUES('2020-01-05','0A', 20, 'Y');
INSERT INTO sample VALUES('2020-01-02','0B', 20, 'Y');
INSERT INTO sample VALUES('2020-01-04','0B', 30, 'Y');
INSERT INTO sample VALUES('2020-01-05','0B', 10, 'Y');
INSERT INTO sample VALUES('2019-12-31','0A',  0, 'N');
INSERT INTO sample VALUES('2019-12-30','0A',  0, 'N');
INSERT INTO sample VALUES('2019-12-31','0B',  0, 'N');
INSERT INTO sample VALUES('2019-12-30','0B',  0, 'N');
INSERT INTO sample VALUES('2020-01-01','0B',  0, 'N');
INSERT INTO sample VALUES('2020-01-03','0B',  0, 'N');
GO
WITH V AS (
  SELECT saledate
        ,code
        ,AVG(val) OVER(PARTITION BY code ORDER BY saledate ROWS BETWEEN 1
preceding AND 1 following) AS "moved_avg"
        ,disp
    FROM sample
)
SELECT saledate
      ,code
      ,moved_avg
  FROM V
  WHERE disp='Y'
GO
```



もし先頭日と最終日を除きたければ行セットVからその日付を除けばいい。

```
WITH V AS (
  SELECT saledate
        ,code
        ,AVG(val) OVER(PARTITION BY code ORDER BY saledate ROWS BETWEEN 1
preceding AND 1 following) AS "moved_avg"
        ,disp
    FROM sample
)
SELECT saledate
      ,code
      ,moved_avg
  FROM V
  WHERE disp='Y'
    AND saledate BETWEEN '2020-01-02' AND '2020-01-04'
GO
```



database, SQL, 不足レコード補填, SQLServer, 技術資料

From:
https://wiki.hgotoh.jp/ - **努力したWiki**

Permanent link:
https://wiki.hgotoh.jp/documents/database/sql-0011

Last update: **2023/04/14 02:32**