

【サブクエリを極める（その2）】

相関副問合せの書き換え

以下の「相関副問合せを使用したSELECT文」は前回の投稿の「主問合せと副問合せが関連するもの」の再掲ですが、この程度の単純なSELECT文であれば、ややこしい相関副問合せを使用することなく、以下の「一般的な副問合せを使用したSELECT文」又「内部結合を使用したSELECT文」で十分です。

表：社員

社員番号	部門	社員名
11111	1000	A
22222	2000	B
33333	3000	C
44444	3000	D
55555	4000	E

表：プロジェクト

プロジェクト番号	社員番号
P001	11111
P001	22222
P002	33333
P002	44444
P003	55555

相関副問合せを使用したSELECT文

```
SELECT プロジェクト番号、社員番号 FROM プロジェクト  
WHERE EXISTS (SELECT * FROM 社員 WHERE 部門 <= '2000' AND 社員番号 = プロジェクト.社員番号)
```

一般的な副問合せを使用したSELECT文

```
SELECT プロジェクト番号、社員番号 FROM プロジェクト WHERE 社員番号 IN (SELECT 社員番号 FROM 社員  
WHERE 部門 <= '2000')
```

内部結合を使用したSELECT文

```
SELECT プロジェクト番号、社員.社員番号 FROM プロジェクト INNER JOIN 社員 USING 社員番号  
WHERE 部門 <= '2000')
```

※SELECT句の属性の「社員.社員番号」は共通属性である「プロジェクト.社員番号」の重複表示を回避するための措置
※USING句の代わりに、「ON プロジェクト.社員番号=社員.社員番号」でもOK

上記3つのSELECT文の実行結果は、右のとおり、同じである。

プロジェクト番号	社員番号
P001	11111
P001	22222

WITH句

前回の投稿において、副問合せは「**SELECT文のWHERE句の条件の中で使用することが代表的です**」と記載しましたが、今回は副問い合わせがWHERE句ではなく、以下のとおり、FROM句にある場合について、説明します。

```
SELECT 名前 FROM (SELECT ID、名前 FROM 利用者表 WHERE 年齢 >= 20) AS 一時表
WHERE 名前 LIKE '%木%'
```

まず、上記のSELECT文を読み解くと「サブクエリでは、利用者表から20歳以上の人のIDと名前を抽出した**一時表自体をメモリ（記憶装置ではなく揮発性のメモリ）に保存**し、メインクエリでは、一時表から名前に「木」が含まれる人を抽出して、その名前を表示する」となります。ここで、サブクエリの内容は、それほど複雑ではありませんが、もっと複雑になった場合は、一時表をVIEWとして定義して、「FROM 一時表」とすることも可能です。

しかし、本当に一時的にしか一時表を使用しないのであれば、もっと簡単に一時表を別途保存することができます。それがWITH句です。以下に上記SELECT文をWITH句を使用して書き換えたものを示します。

```
WITH 一時表 AS (SELECT ID、名前 FROM 利用者表 WHERE 年齢 >= 20)
SELECT 名前 FROM 一時表 WHERE 名前 LIKE '%木%'
※上記の赤字と緑字は1つの文ではなく、独立している。
```

サブクエリが複雑ではないため、WITH句の恩恵は分かりにくいですが、2行目のSELECT文はスッキリしました。先ほどサブクエリでは「**一時表自体をメモリに保存**」と記載しましたが、サブクエリを含むSQL文が実行された後は、メモリから消えてしまいます。しかし、WITH句で指定した一時表自体は、メモリに保存されることで、同一セッションであれば、シャットダウンしない限り、再度WITH句を書くことなく一時表を読み込むことができます。

ただし、WITH句の連発によるメモリを使い過ぎは、パフォーマンスの低下に繋がるため、使用頻度を考慮して、サブクエリかWITH句かを選択する必要があります。