Advanced SQL in Oracle and SQL Server

PARTITION BY/RIGHT OUTER JOIN

Scott L. Hecht
http://www.sheepsqueezers.com
@sheepsqueezers

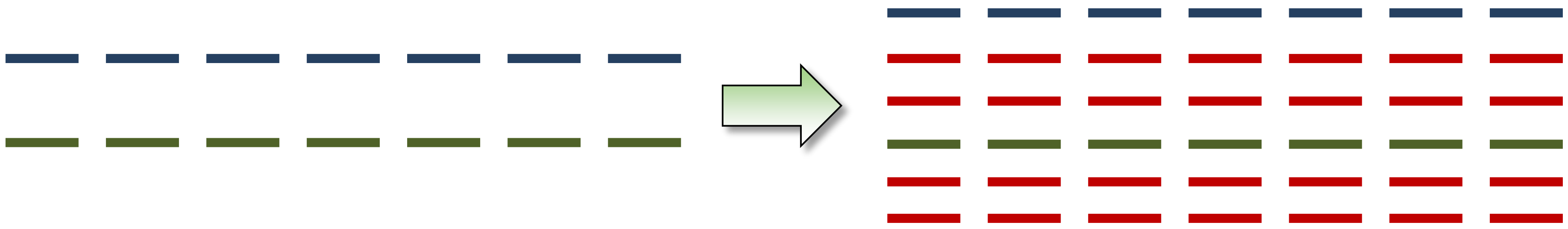
pluralsight
hardcore developer training

Module Contents

- **PARTITION BY/RIGHT OUTER JOIN**
  - Introduction
  - Data used in Module
  - The Desired Results
  - Why INNER/LEFT/RIGHT/FULL JOINs Won't Work
  - Tried and True Method: Cartesian Product/UNION ALL
  - Syntax for PARTITION BY/RIGHT OUTER JOIN
  - Example
  - Explanation
  - Summary

Introduction

- **Why Learn PARTITION BY/RIGHT OUTER JOIN?**
  - Easily fill in missing data
  - Specify partition(s) for fine-grained fills
  - Additional column(s) of interest will be filled with NULLs



Data Used in Module

- **Table**
  - MYRXDATA
- **Columns**
  - DATEKEY – monthly date column
  - NDC_KEY – character string containing drug code number
  - COPAY_AMT – dollar amount of copay for drug
- **Data**

```
DATEKEY   NDC_KEY     COPAY_AMT
01-JAN-05 11111111111 5.12
01-FEB-05 22222222222 10.24
```

Data Used in Module

- **Table**
  - DATEDIMENSION
- **Columns**
  - DATEKEY – monthly date column
- **Data**

DATEKEY
01-JAN-05
01-FEB-05
01-MAR-05

The Desired Results

- **What do we want to see at the end of this module?**
  - All three months appear
  - Both NDC_KEYs appear for each month
  - COPAY_AMT is NULL where there is not a value in MYRXDATA

```
DATEKEY   NDC_KEY     COPAY_AMT
01-JAN-05 11111111111 5.12
01-FEB-05 11111111111
01-MAR-05 11111111111
01-JAN-05 22222222222
01-FEB-05 22222222222 10.24
01-MAR-05 22222222222
```

INNER JOIN

- **What does an INNER JOIN Produce?**

```
SELECT B.DATEKEY,A.NDC_KEY,A.COPAY_AMT
 FROM MYRXDATA A INNER JOIN DATEDIMENSION B
 ON A.DATEKEY=B.DATEKEY
```

```
DATEKEY   NDC_KEY     COPAY_AMT
01-JAN-05 11111111111 5.12
01-FEB-05 22222222222 10.24
```

LEFT JOIN

- **What does a LEFT JOIN Produce?**

```
SELECT B.DATEKEY,A.NDC_KEY,A.COPAY_AMT
 FROM MYRXDATA A LEFT JOIN DATEDIMENSION B
 ON A.DATEKEY=B.DATEKEY
```

```
DATEKEY   NDC_KEY     COPAY_AMT
01-JAN-05 11111111111 5.12
01-FEB-05 22222222222 10.24
```

RIGHT JOIN

- **What does a RIGHT JOIN Produce?**

```
SELECT B.DATEKEY,A.NDC_KEY,A.COPAY_AMT
 FROM MYRXDATA A RIGHT JOIN DATEDIMENSION B
 ON A.DATEKEY=B.DATEKEY
```

```
DATEKEY    NDC_KEY     COPAY_AMT
01-JAN-05 11111111111 5.12
01-FEB-05 22222222222 10.24
01-MAR-05
```

FULL JOIN

- **What does a FULL JOIN Produce?**

```
SELECT B.DATEKEY,A.NDC_KEY,A.COPAY_AMT
 FROM MYRXDATA A FULL JOIN DATEDIMENSION B
 ON A.DATEKEY=B.DATEKEY
```

```
DATEKEY    NDC_KEY     COPAY_AMT
01-JAN-05 11111111111 5.12
01-FEB-05 22222222222 10.24
01-MAR-05
```

Tried and True Method

- **Tried and True Method**
  - Step 1: Create Cartesian Product of NDC_KEY and DATEKEY
    - Must add an additional NULL column for COPAY_AMT
  - Step 2: UNION ALL the results of Step 1 with original data
  - Step 3: Summarize the combined data

```
SELECT C.DATEKEY,C.NDC_KEY,SUM(C.COPAY_AMT) AS TOT
 FROM (
       SELECT A.DATEKEY,B.NDC_KEY,NULL AS COPAY_AMT
        FROM (SELECT DISTINCT DATEKEY
               FROM DATEDIMENSION) A,(SELECT DISTINCT NDC_KEY
                                       FROM MYRXDATA) B
       UNION ALL
       SELECT DATEKEY,NDC_KEY,COPAY_AMT
        FROM MYRXDATA
      ) C
 GROUP BY C.DATEKEY,C.NDC_KEY
```

Tried and True Method

- **Tried and True Method (*continued*)**
  - Step 1: Create Cartesian Product of NDC_KEY and DATEKEY
    - Must add an additional NULL column for COPAY_AMT
  - Step 2: UNION ALL the results of Step 1 with original data
  - Step 3: Summarize the combined data

DATEKEY	NDC_KEY	COPAY_AMT
01-JAN-05	11111111111	5.12
01-FEB-05	11111111111	
01-MAR-05	11111111111	
01-JAN-05	22222222222	
01-FEB-05	22222222222	10.24
01-MAR-05	22222222222	

Syntax

- PARTITION BY/RIGHT OUTER JOIN Syntax

```
SELECT …
  FROM fact F
    PARTITION BY (F.colF,…)
    RIGHT OUTER JOIN dimension D
     ON F.colD = D.colD
```

- where
  - *fact* represents the data missing rows of data
  - *dimension* represents the data you want filled in
  - *colF* is the column that will be partitioned
  - *colD* is the column that will be filled in
  - Availability:
    - Oracle: 10g/R1
    - SQL Server: N/A

Example

- Task: Fill in the missing dates for each NDC in MYRXDATA.
- Note: Use PARTITION BY/RIGHT OUTER JOIN Syntax

```
SELECT B.DATEKEY,A.NDC_KEY,A.COPAY_AMT
 FROM MYRXDATA A
  PARTITION BY (A.NDC_KEY)
  RIGHT OUTER JOIN DATEDIMENSION B
   ON A.DATEKEY=B.DATEKEY
```

```
DATEKEY   NDC_KEY     COPAY_AMT
01-JAN-05 11111111111 5.12
01-FEB-05 11111111111
01-MAR-05 11111111111
01-JAN-05 22222222222
01-FEB-05 22222222222 10.24
01-MAR-05 22222222222
```

Explanation

- **Why does this work?**

DATEKEY	NDC_KEY	COPAY_AMT
01-JAN-05	11111111111	5.12
01-FEB-05	22222222222	10.24

DATEKEY
01-JAN-05
01-FEB-05
01-MAR-05

DATEKEY	NDC_KEY	COPAY_AMT
01-JAN-05	**11111111111**	**5.12**
01-FEB-05	11111111111	
01-MAR-05	11111111111	
01-FEB-05	**22222222222**	**10.24**
01-JAN-05	22222222222	
01-MAR-05	22222222222	

Summary

- Allows you to fill in missing data
- PARTITION BY allows fine-grained filling in
- RIGHT OUTER JOIN ensures all missing data filled in