Database Programming

Nonequijoins

Objectives

In this lesson, you will learn to:

- Construct and execute a SELECT statement to access data from more than one table using a nonequijoin

Purpose

What happens if you want to retrieve data from a table that has no corresponding column in another table? For instance, your math percentage grade of 92 is stored in the GRADES column in one table; the letter grade is stored in the LETTER_GRADE column in another table. How can we join the number grade with the letter grade? When data is recorded using a range, retrieving it is the job of a nonequijoin.

Nonequijoin

00-15 Logged = 15 Paid
16-30 Logged = 30 Paid
31-45 Logged = 45 Paid
46-60 Logged = 60 Paid

Example:

A company pays its employees who earn an hourly wage in 15-minute increments. One table stores the hours and minutes recorded by a time clock and another table stores the pay range. If the minutes worked is between 0 and 15, the worker is paid for 15 minutes. If the minutes worked is between 16 and 30, the worker is paid for 30 minutes. If the minutes worked is between 31 and 45, the worker is paid for 45 minutes. If the minutes worked is between 46 and 60, the worker is paid for 1 hour

Nonequijoin (cont.)

To join your number grade (or %) in math with its corresponding letter grade, a nonequijoin is needed. Since there is no exact match between the two columns in each table, the equality operator = can't be used. Although comparison conditions such as < = and > = can be used, BETWEEN...AND is a more effective way to execute a nonequijoin.

Nonequijoin (cont.)

Results of this query on the next slide joins the D_EVENTS cost column with the D_PACKAGES low_range and high_range columns using BETWEEN...AND

```
SELECT d_packages.code, d_events.cost
FROM d_packages, d_events
WHERE d_events.cost BETWEEN d_packages.low_range AND d_packages.high_range
```

Nonequijoin (cont.)

D_EVENTS

ID	NAME	EVENT_DATE	DESCRIPTION	COST	VENUE_ID	PACKAGE_CODE	THEME_CODE	CLIENT_NUMBER
100	Peters Graduation	14-MAY-2004	Party for 200	8000	100	112	200	5922
100	Vigils Wedding	28-APR-2004	Black tie, Four Seasons	10000	220	200	200	6133

D_PACKAGES

CODE	LOW_RANGE	HIGH_RANGE
79	500	2500
87	2501	5000
112	5001	10000
200	10001	15000

```
SELECT d_packages.code, d_events.cost
FROM d_packages, d_events
WHERE d_events.cost BETWEEN d_packages.low_range AND d_packages.high_range
```

CODE	COST
112	8000
112	10000

Terminology

Key terms used in this lesson included:

- Nonequijoin
- BETWEEN…AND

Summary

In this lesson you have learned to:

- Construct and execute a SELECT statement to access data from more than one table using a nonequijoin