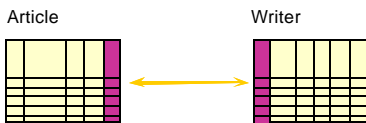


Chapter 6 Equijoins

Review

- Why do we store data for articles and writers in two separate tables?
- How can we link an article to its writer?



2

Joins in SQL92

- State the join criteria in the WHERE clause
- Prefix the column name with the table name when the same column name appears in more than one table

SQL92

```
SQL> SELECT ln, fn, title, type
FROM writer, article
WHERE writer.writerid = article.writerid;
```

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column1 = table2.column2;
```

3



Inner Join (aka Equijoin)

- A join in which records from *two* tables are combined and added to the result set only when there are *matching values* in the common fields

SQL92

```
SQL> SELECT ln, fn, title, type
       FROM writer, article
       WHERE writer.writerid = article.writerid;
```

- If a record in one table doesn't have a matching record in the other table, it *won't* be selected
 - Outer joins show *all* of a table's records, even if there are no matching record in the other table (chapter 9)

4



Table Ambiguities

- Why do we get error message here?

```
SQL> SELECT writerid, ln, fn, title, type
       FROM writer, article
       WHERE writerid = writerid;
```

- Remove ambiguity by also citing the table name

```
SQL> SELECT writer.writerid, ln, fn,
           title, type
       FROM writer, article
       WHERE writer.writerid = article.writerid;
```

5



Practice Time

- Display title, length, and description for any article longer than 2000 words.

6



Table Alias

- Makes it easier to refer to a table
- Required when need to join a table to itself
 - Recursive relationship (chapter 9)

```
SQL> SELECT writer.writerid, ln, fn, title, type
FROM writer, article
WHERE writer.writerid = article.writerid;
```

```
SQL> SELECT w.writerid, ln, fn, title, type
FROM writer w, article a
WHERE w.writerid = a.writerid;
```

7



Nulls and Joins

- NULL is not equal to any other value, including another NULL
- Records having NULL in the join criteria field won't appear in the result set

8



Inner Joins: ANSI SQL99 Syntax new9

- Alternate syntax to produce an inner join (equijoin)
- Produces same result set as ANSI SQL92

SQL92

```
SQL> SELECT ln, fn, title, type
FROM writer, article
WHERE writer.writerid = article.writerid;
```

- INNER JOIN... ON syntax
 - ON clause cites fully qualified primary key/foreign key references

SQL99

```
SQL> SELECT ln, fn, title, type
FROM writer INNER JOIN article
ON writer.writerid = article.writerid;
```

9

Inner Joins: ANSI SQL99 Syntax new9

SQL92

```
SQL> SELECT ln, fn, title, type
FROM writer, article
WHERE writer.writerid = article.writerid;
```

- INNER JOIN... USING Syntax
 - when primary key and foreign key have same name

SQL99

```
SQL> SELECT ln, fn, title, type
FROM writer INNER JOIN article
USING (writerid);
```

10

Cartesian Product

- A result set that joins every record in one table to every record in another table

SQL92

```
SQL> SELECT ln, fn, title, type
FROM writer, article;
```

- Occurs when a join expression is omitted
- To avoid a Cartesian product, always include a valid join condition
 - in either the FROM clause or the WHERE clause

11

Joining Three or More Tables

Table 1

Table 2



Table 3

12

Joining Three or More Tables

- Use additional join expressions in the WHERE clause
- Each join expression always involves *two* tables and common field(s)

SQL92

```
SQL> SELECT title, descr, ln
FROM writer w, article a, type t
WHERE w.writerid = a.writerid
AND a.type = t.type;
```

13

Joining Three or More Tables

- Use additional INNER JOIN... ON expressions
- Each INNER JOIN expression always involves *two* tables and common field(s)

SQL99

```
SQL> SELECT title, descr, ln
FROM writer w
INNER JOIN article a
ON (w.writerid = a.writerid)
INNER JOIN type t
ON (a.type = t.type)
ORDER BY descr;
```

14

Practice Time

- Using the [SQL92](#) join style, show each instructor's full (concatenated) name, course description, and the start date/time and location for each section they're teaching. Only show sections that meet in the M building.

- Modify the previous statement to use a [SQL99](#) join.

15



Multicolumn Join

- Composite primary key
 - A primary consisting of a **combination** of two (or more) fields
 - ENROLLMENT table has a **composite primary key** of STUDENT_ID and SECTION_ID together
 - ENROLLMENT is a **junction table** that breaks the Many-to-Many relationship between STUDENT and SECTION into two separate One-to-Many relationships
- Composite foreign key
 - A foreign key consisting of a **combination** of two (or more) fields
 - GRADE table has STUDENT_ID and SECTION_ID as a **composite foreign key** so each GRADE record can be matched to an ENROLLMENT record

16



Multicolumn Join

- Show student 283's last name, courses and grades

SQL92

```
SQL> SELECT stu.last_name, cou.description,
           gr.numeric_grade, gr.grade_type_code
   FROM student stu, enrollment enr,
        section sec, course cou, grade gr
  WHERE stu.student_id = enr.student_id AND
        enr.student_id = gr.student_id AND
        enr.section_id = gr.section_id AND
        enr.section_id = sec.section_id AND
        sec.course_no = cou.course_no AND
        stu.student_id = 283;
```

17



Multicolumn Join

- Show student 283's last name, courses, grades

SQL99

```
SQL> SELECT stu.last_name, cou.description,
           gr.numeric_grade, gr.grade_type_code
   FROM student stu
      INNER JOIN enrollment enr
            ON (stu.student_id = enr.student_id)
      INNER JOIN grade gr
            ON (enr.section_id = gr.section_id)
            AND (enr.student_id = gr.student_id)
      INNER JOIN section sec
            ON (enr.section_id = sec.section_id)
      INNER JOIN course cou
            ON (sec.course_no = cou.course_no)
  WHERE stu.student_id = 283;
```

18



Practice Time

- Provide a SQL statement that shows each instructor's last name, the section id and start date/time for sections they're teaching and, for each student enrolled in the instructor's sections, show the student's last name and numeric grade. Show only grades that are at least 80.

19
