

Chapter 10 Complex Joins

Outer Joins
Self Joins

Inner Join (Chapter 7)

- Rows from one table are included in the result set only if there is a **matching row** in the other table
 - rows won't be shown if there is no matching row in the other table
 - the **missing rows** phenomenon
- Join Conditions
 - indicate the common field(s) used to join rows across the tables

SQL92

```
SELECT title, descr, length
FROM article a, type t
WHERE a.type = t.type
ORDER BY descr;
```

But...which
DESCRs **don't**
you see?

SQL99

```
SELECT title, descr, length
FROM article a INNER JOIN type t
ON a.type = t.type
ORDER BY descr;
```

2

Outer Join Concepts

- Every row from one table is included in the result set even if no matching row exists in the other table
 - rows that have a matching row in the other table are shown, as usual
 - when a row does not have a matching row in the other table, Oracle generates a Null-filled row to match it to
- Driving Table** (nib)
 - the tables whose full set of records is selected
 - when every row from Table A is to be included in the result set even if no matching rows exist in Table B, Table A is the driving table
- 2 ways to perform an Outer Join in Oracle (9i or later)
 - use ANSI99 **OUTER JOIN** in FROM clause
 - portable to non-Oracle databases
 - use the Oracle outer join operator (**+**) on the **non-driving table**
 - Oracle specific, the only available outer join method prior to Oracle 9i

3

ANSI Outer Joins

SQL99

- Show article titles and their type descriptions, including types that don't have a matching article

SQL99

```
SELECT title, descr, length
FROM type t LEFT OUTER JOIN article a
ON t.type = a.type
ORDER BY descr;
```

- Which **rows** have NULLs?
- Which **columns** have NULLs?

4

ANSI Outer Joins

SQL99

- Which is the driving table in each?

```
SELECT title, descr, length
FROM type t LEFT OUTER JOIN article a
ON t.type = a.type
ORDER BY descr;
```

```
SELECT title, descr, length
FROM article a RIGHT OUTER JOIN type t
ON a.type = t.type
ORDER BY descr;
```

```
SELECT title, descr, length
FROM article a LEFT OUTER JOIN type t
ON a.type = t.type
ORDER BY descr;
```

5

ANSI Outer Joins

SQL99

- How can you show only those types that lack a matching article?

```
SELECT title, descr, length
FROM type t LEFT OUTER JOIN article a
ON t.type = a.type
WHERE articlenum IS NULL
ORDER BY descr;
```

- Add a WHERE clause that tests for IS NULL on the non-driving table's primary key

6

Using Where Conditions and Outer Joins Together

- An Outer Join is used to select every row from the driving table, whether or not a matching row exists in the other table
 - when there are no naturally-occurring matching rows in the other table, Oracle generates a Null-filled row to match it to
- Using WHERE clauses and Outer Joins together can give unexpected results
 - beware of the timing of these two operations! (pg410)
 - the OUTER JOIN is performed first, then the WHERE condition is evaluated and applied to the result set

7

Using Where Conditions and ANSI Outer Joins Together (pg 411)

Show all BUS articles

```
SELECT title, type, ln, fn, freelancer
FROM writer w RIGHT OUTER JOIN article a
ON w.writerid = a.writerid
WHERE type = 'BUS';
```

6 rows

Show all BUS articles written by a freelancer

```
SELECT title, type, ln, fn, freelancer
FROM writer w RIGHT OUTER JOIN article a
ON w.writerid = a.writerid
WHERE type = 'BUS'
AND freelancer = 'Y';
```

4 rows

Show all BUS articles written by a freelancer

```
SELECT title, type, ln, fn, freelancer
FROM writer w RIGHT OUTER JOIN article a
ON (w.writerid = a.writerid AND
freelancer = 'Y' )
WHERE type = 'BUS';
```

6 rows

- When an OUTER JOIN has additional conditions that involve column(s) from the non-driving table, including those conditions in the ON clause causes the condition to be evaluated as the outer join is being performed, not afterward

8

Oracle Outer Join Operator (+)

- Indicates that Oracle should generate Null rows when there is no matching row in the other table
 - is used in conjunction with the **non-driving** table
- Show article titles and their descriptions, including types that don't have any matching articles

```
SELECT title, descr, length
FROM type t, article a
WHERE a.type(+) = t.type
ORDER BY descr;
```

- Which is the **driving** table?
- Which **rows** have NULLS?
- Which **columns** have NULLS?

9

Oracle Outer Join Operator (+) Examples

- What would each of the following produce?

```
SELECT title, ln, fn
FROM writer w, article a;
```

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

```
SELECT title, ln, fn
FROM writer w, article a
WHERE w.writerid(+) = a.writerid;
```

```
SELECT title, ln, fn
FROM writer w, article a
WHERE w.writerid = a.writerid(+);
```

10

Using Where Conditions and Oracle Outer Join Operator(+) Together (pg 410)

```
SELECT title, type, ln, fn , freelancer
FROM writer w, article a
WHERE w.writerid(+) = a.writerid
AND type = 'BUS';
```

6 rows

```
SELECT title, type, ln, fn , freelancer
FROM writer w, article a
WHERE w.writerid(+) = a.writerid
AND type = 'BUS'
AND freelancer = 'Y';
```

4 rows

```
SELECT title, type, ln, fn, freelancer
FROM writer w, article a
WHERE w.writerid(+) = a.writerid
AND type = 'BUS'
AND w.freelancer(+) = 'Y';
```

6 rows

- When an OUTER JOIN has additional conditions that involve column(s) from the **non-driving** table, including an outer join operator on the column(s) from the **non-driving** table causes the condition to be evaluated as the outer join is being performed

11

Referential Integrity

- Requires each **foreign key** in a row in the Many table to match the primary key of a row in the One table
 - eg: can't have an Article with an unmatched WRITERID
- When you define a relationship between two tables, referential integrity prevents unmatched rows on the Many side
 - Oracle won't permit an unmatched foreign key
 - use a Foreign Key **constraint** to enforce referential integrity (chapter 12)

```
INSERT INTO article VALUES(articlenum_seq.NEXTVAL, 'New Article', 'BUS', Null, Null, 'A000');
Error starting at line 9 in command:
INSERT INTO article VALUES(articlenum_seq.NEXTVAL, 'New Article', 'BUS', Null, Null, 'A000')
Error report:
SQL Error: ORA-02291: integrity constraint (TTROLLEN.ARTICLE_WRITERID_FK) violated - parent key not found
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"
*Cause: A foreign key value has no matching primary key value.
*Action: Delete the foreign key or add a matching primary key.
```

- Does referential integrity prevent unmatched rows on the One side?
 - eg: can you have a Writer without any matching Articles?

12

Create Unmatched Rows to Demo Full Outer Join

- The SQL statements below (Chapter 11 & 12) cause unmatched rows in both the WRITER and ARTICLE tables

```
ALTER TABLE article
  DISABLE CONSTRAINT article_writerid_fk;

INSERT INTO article VALUES(articlenum_seq.NEXTVAL, 'New
Article', 'BUS', Null, Null, 'A000');

UPDATE ARTICLE
  SET writerid = 'X999'
  WHERE writerid = 'J525';

INSERT INTO writer
VALUES ('X123', 'New', 'Writer', Null, Null, Null, Null, Null);

COMMIT;
```

13

Full Outer Join

- Shows every row in each table, including:
 - rows in table A that match rows in table B
 - rows in table A that don't match a row in table B
 - rows in table B that don't match a row in table A

```
SELECT title, ln, fn
  FROM writer w FULL OUTER JOIN article a
    ON w.writerid = a.writerid
 ORDER BY title;
```

14

Full Outer Join Using Oracle Outer Join Operator (+)

- Oracle doesn't support two (+) in a single SELECT statement

```
SELECT title, ln, fn
  FROM writer w, article a
 WHERE w.writerid(+) = a.writerid(+);
```

ORA-01468: a predicate may reference only one outer-joined table
01468. 00000 - "a predicate may reference only one outer-joined table"

- Use a UNION operator and an outer join operator (+) to drive from each table

```
SELECT title, ln, fn
  FROM writer w, article a
 WHERE w.writerid(+) = a.writerid
 UNION
 SELECT title, ln, fn
  FROM writer w, article a
 WHERE w.writerid = a.writerid(+);
```

All Rows Fetched: 31 in 0.562 seconds

	TITLE	LN	FN
1	\$100 Billion AOL Time Warner Merger Approved Johnson	Len	
2	25% Tax Cut Bill Approved	Hall	Val
3	AT&T Antitrust Settlement	Kim	Cho
4	Advertising Over the Past 25 Years	Ngo	Thu
5	Alternative Energy Sources	Seeger	Wai

Fix-Up Unmatched Rows

- Repair changes that violated referential integrity

```
DELETE
  FROM article
 WHERE title = 'New Article';

UPDATE article
  SET writerid = 'J525'
  WHERE writerid = 'X999';

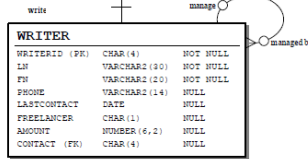
DELETE
  FROM writer
  WHERE writerid = 'X123';

ALTER TABLE article
  ENABLE CONSTRAINT article_writerid_fk;
```

16

Self-Join Concepts

- A join in which a column in a table references another column in the **same table**
 - a.k.a. recursive relationship
 - when a FK matches a PK of a row in the same table



WRITERID (FK)	LN	FN	PHONE	LASTCONTACT	FREELANCER	AMOUNT	CONTACT (FK)
CHAR(4)	VARCHAR2(20)	VARCHAR2(20)	VARCHAR2(14)	DATE	CHAR(1)	NUMBER(6,2)	CHAR(4)
NOT NULL	NOT NULL	NOT NULL					

```
SELECT writerid, ln, fn, phone, contact
FROM writer;
```

WRITERID	LN	FN	PHONE	CONTACT
1	L350	Lawton Pat	(null)	(null)
2	W432	Waldeck Kristine	(917) 361-8181	(null)
3	C200	Cox Kelly	(210) 783-5415	W432
4	C500	Cohen Julia Rice	(910) 338-1875	L350
5	E235	Epstein Diane	(610) 349-9689	W432
6	H400	Hall Valerie	(710) 918-7767	L350

Self-Joins

- A join in which a column in a table references another column in the **same table**
- The FROM clause cites the same table twice but uses a table **alias** to control the context each table is used
- Show each writer and their contact person/boss

```
SELECT w.ln || ', ' || w.fn as "Writer",
       w.freelancer,
       contact.ln || ', ' || contact.fn as Boss
FROM writer w INNER JOIN writer contact
ON w.contact = contact.writerid
ORDER BY Boss, w.ln, w.fn;
```

Statement Output x Query... x

SQL All Rows Fetched: 14 in 0.032 seconds

Writer	FREELANCER	BOSS
1 Cohen, Julia Rice	N	Lawton,Pat
2 Hall, Valerie	N	Lawton,Pat
3 Kim, Chong	N	Lawton,Pat
4 Leavitt, Morris	N	Lawton,Pat
5 Martinez, Maria	N	Lawton,Pat
6 Martinez, Santos	N	Lawton,Pat
7 Ngo, Thuy	N	Lawton,Pat
8 Yamamura, Shinjiro	N	Lawton,Pat
9 Cox, Kelly	Y	Waldeck,Kristin