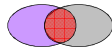
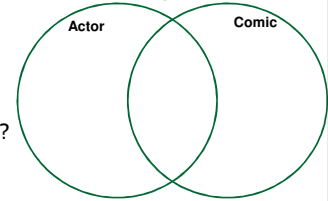


Chapter 9 Set Operators



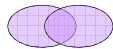
Warm Up

- Solve each of the following expressions
 - $6-2=$ _____
 - $2-6=$ _____
 - $6+2=$ _____
 - $2+6=$ _____
- Place the following persons in the Venn Diagram
 - Denzel Washington
 - Billy Crystal
 - Steven Colbert
 - Marina Franklin
 - Annete Bening
- What would each produce?
 - comic INTERSECT actor
 - actor MINUS comic
 - comic MINUS actor
 - comic UNION actor



2

UNION Operator



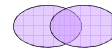
- Combines the results of two separate queries into a single result set with duplicate rows removed
- Each query must return the same number of columns and the columns must be of compatible data type

```
SELECT ln, fn, phone
  FROM writer
UNION
SELECT last_name, first_name, phone
  FROM student;
```

- The rows of the result set will automatically be sorted
 - why?

3

UNION Operator



- Can include a column of **literals** to identify which rows are from each SELECT statement

```
SELECT ln, phone, 'Writer' AS Type
  FROM writer
UNION
SELECT last_name, phone, 'Student'
  FROM student
UNION
SELECT last_name, phone, 'Instructor'
  FROM instructor;
```

Query Result x

SQL | Fetched 100 rows in 0.03 seconds

| | LN | PHONE | TYPE |
|----|-------------|----------------|------------|
| 43 | Caro | 203-555-5555 | Student |
| 44 | Channonkool | 718-555-5555 | Student |
| 45 | Chataan | 718-555-5555 | Student |
| 46 | Cheevens | 203-555-5555 | Student |
| 47 | Chirichella | 718-555-5555 | Student |
| 48 | Chow | 2125551212 | Instructor |
| 49 | Citron | 201-555-5555 | Student |
| 50 | Cohen | (910) 338-1875 | Writer |
| 51 | Cook | 201-555-5555 | Student |
| 52 | Cowie | 201-555-5555 | Student |
| 53 | Cox | (210) 783-5415 | Writer |

4

Column Aliases and Set Operations

- Column names/aliases from the **topmost** SELECT statement are used for the result set

| LN | FN | PHONE |
|----|-------|-------------------------|
| 1 | Abdou | Jerry 718-555-5555 |
| 2 | Abdou | Mardig 718-555-5555 |
| 3 | Abid | Suzanne M. 718-555-5555 |

| Last | First | PHONE |
|---------|------------|--------------|
| 1 Abdou | Jerry | 718-555-5555 |
| 2 Abdou | Mardig | 718-555-5555 |
| 3 Abid | Suzanne M. | 718-555-5555 |

ORDER BY and Set Operations

- If used, ORDER BY must appear in the **last** query
 - can use a **number** to refer to the **position** of the sort column
 - you must use a number if column names are different in the queries

| PHONE | LN | TYPE |
|----------------|----------|---------|
| 1 718-555-5555 | Abdou | Student |
| 2 718-555-5555 | Abid | Student |
| 3 201-555-5555 | Affinito | Student |

UNION ALL Operator

- A UNION query in which **duplicate rows** are retained
- Does not automatically sort rows

| | |
|--|--|
| <pre>SELECT zip FROM student UNION SELECT zip FROM instructor;</pre> <p>149 rows</p> | <pre>SELECT zip FROM student UNION ALL SELECT zip FROM instructor;</pre> <p>278 rows</p> |
|--|--|

- How is the UNION ALL query's output different?

-
-

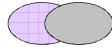
MINUS Operator



- Identifies rows in one set that don't exist in another
- Oracle eliminates duplicate rows from each set before determining the difference
- Each of the queries must return the same number of columns and the columns must be of compatible data type
- Which instructors haven't been assigned to a section?

```
SELECT instructor_id FROM instructor
MINUS
SELECT instructor_id FROM section;
```

MINUS Operator

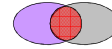


- Be careful which SELECT appears above the MINUS operator and which appears below it

```
SELECT instructor_id FROM section
MINUS
SELECT instructor_id FROM instructor;
```

9

INTERSECT Operator

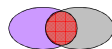


- Identifies rows that are **common** to both sets
- Oracle eliminates duplicate rows from each set before determining the intersection
- Each query must return the same number of columns and the columns must be of compatible data type
- Doesn't matter which SELECT appears above the INTERSECT operator and which appears below it
- Which instructors have been assigned to a section?

```
SELECT instructor_id FROM section
INTERSECT
SELECT instructor_id FROM instructor;
```

10

INTERSECT Operator



- Similar to an equijoin
 - an equijoin produces a result set consisting of rows in one table that have a **matching row** in another table

```
SELECT DISTINCT i.instructor_id
FROM instructor i, section s
WHERE i.instructor_id = s.instructor_id;
```

- an equijoin can display **additional fields from either table**

```
SELECT i.instructor_id, last_name, section_id
FROM instructor i, section s
WHERE i.instructor_id = s.instructor_id
ORDER BY i.instructor_id;
```

11

Set Operations Practice

- Preliminaries
 - Which salutations are present in the instructor table?
 - Which salutations are present in the student table? Note the '.'s
 - Which salutations (after strip off the '.') are present in the student table?
- Show a non-duplicated list of all salutations, whether from the instructor or student table.
- Which salutations are common to both the instructor and student tables?
- Which salutations are present in the instructor table but not in the student table?
- Which salutations are present in the student table but not in the instructor table?

12