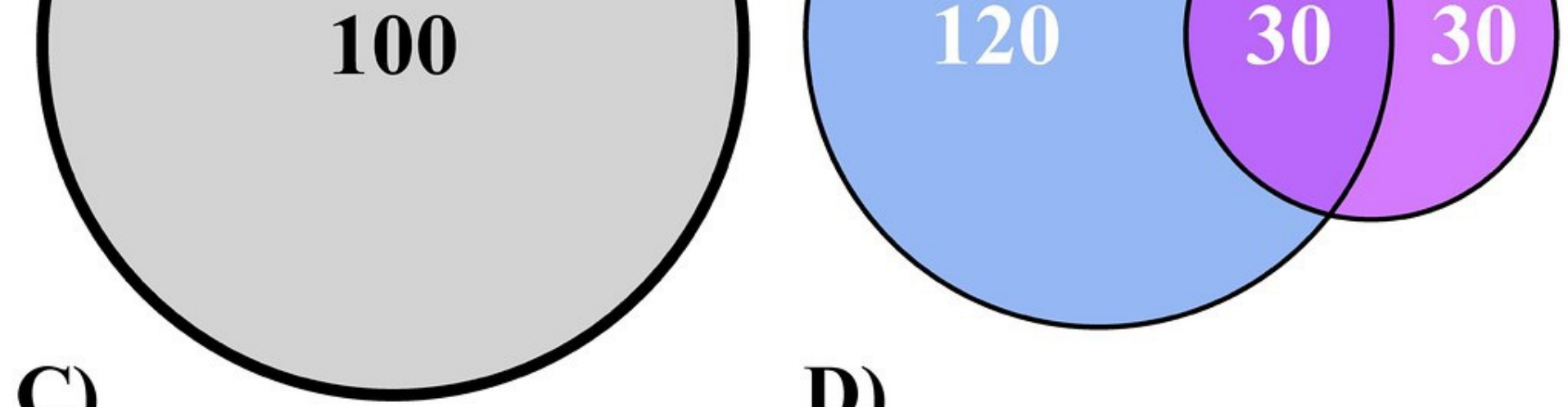
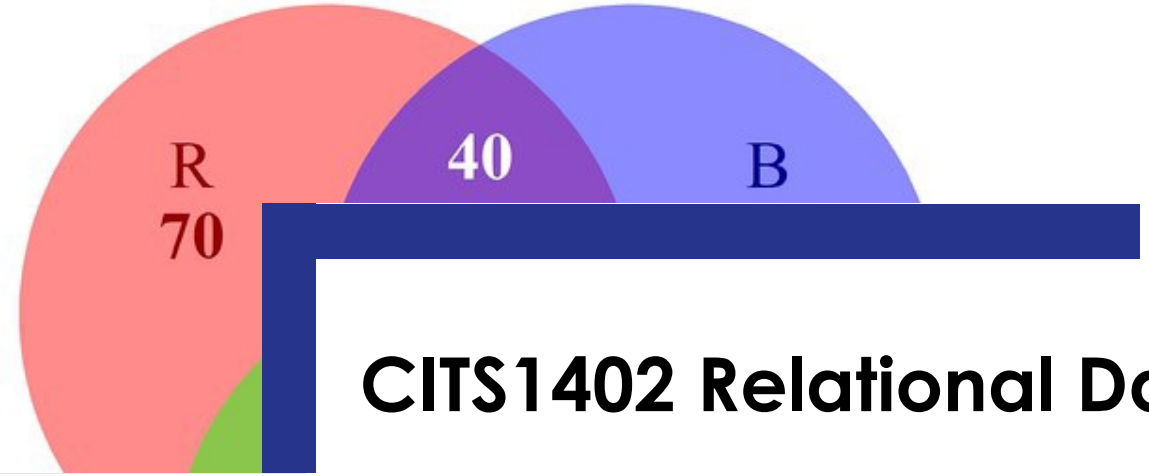



C)

D)





THE UNIVERSITY OF WESTERN AUSTRALIA

# CITS1402 Relational Database Management Systems

## Week 9—SQL JOINS

Sem1/2024

Dr Mehwish Nasim/ CSSE
cits1402-sem1-pmc@uwa.edu.au

CITS1402 Week9

THE UNIVERSITY OF WESTERN AUSTRALIA

# Contents

**SET operations in SQL**

*Courtesy: TWL*

# Chapter 6 - Objectives

**How to retrieve data from database using SELECT and:**

- **Use compound WHERE conditions.**
- **Use aggregate functions.**
- **Sort query results using ORDER BY.**
- **Group data using GROUP BY and HAVING.**
- **Use subqueries.**
- **Join tables together.**
- **Perform set operations (UNION, INTERSECT, EXCEPT).**

# SELECT Statement

```
SELECT [DISTINCT | ALL]
        {* | [columnExpression [AS newName]] [,...] }
FROM            TableName [alias] [, ...]
[WHERE          condition]
[GROUP BY       columnList]
[HAVING         condition]
[ORDER BY       columnList]
```

# Union, Intersect, and Difference (Except)

**Can use normal set operations of Union, Intersection, and Difference to combine results of two or more queries into a single result table.**

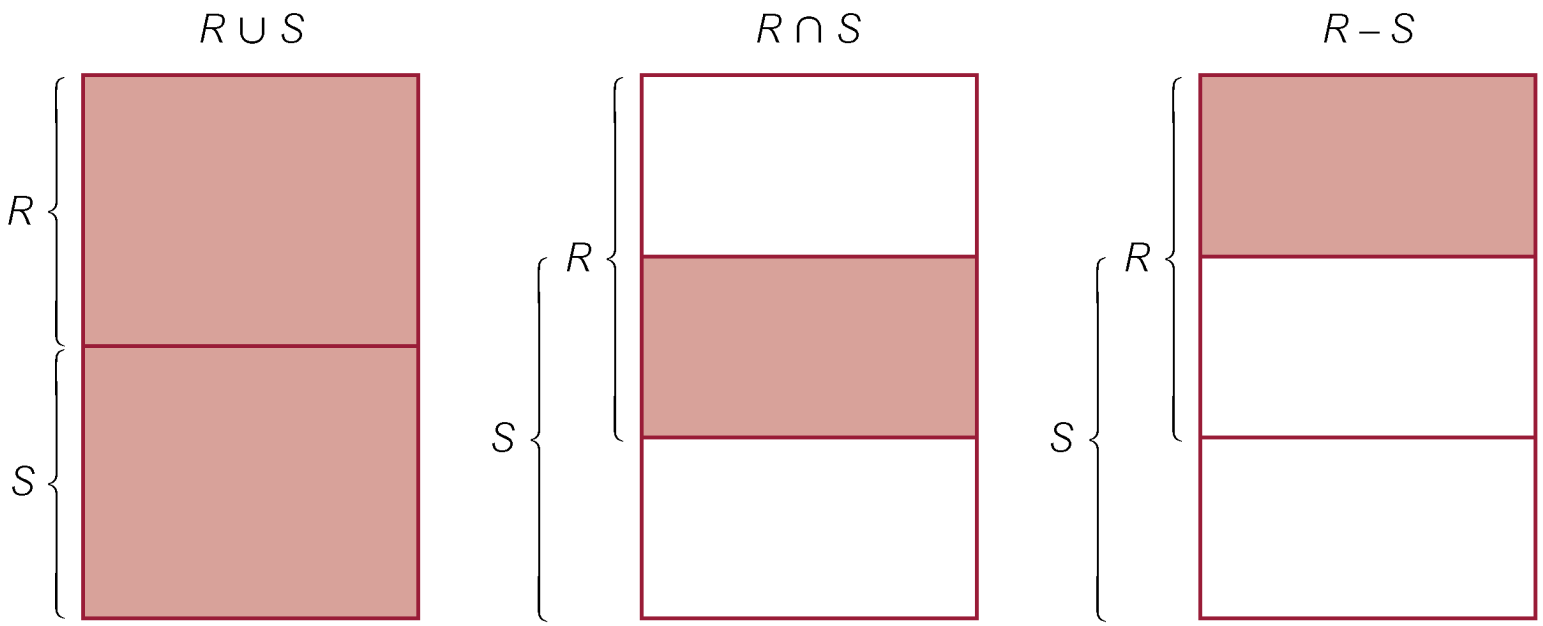
**Union of two tables, A and B, is table containing all rows in either A or B or both.**

**Intersection is table containing all rows common to both A and B.**

**Difference is table containing all rows in A but not in B.**

**Two tables must be *union compatible*.**

# Union, Intersect, and Difference (Except)




(a) Union (b) Intersection (c) Difference

# Union, Intersect, and Difference (Except)

**Format of set operator clause in each case is:**

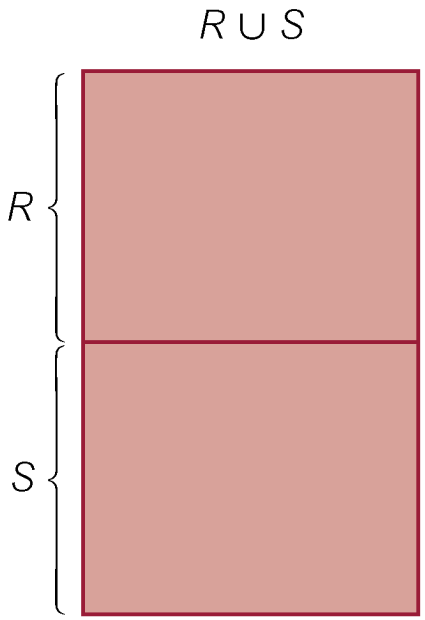
***op* [ALL] [CORRESPONDING [BY {column1 [, ...]}]]**

**If CORRESPONDING BY specified, set operation performed on the named column(s).**

**If CORRESPONDING specified but not BY clause, operation performed on common columns.**

**If ALL specified, result can include duplicate rows.**

# Union, Intersect, and Difference (Except)




(a) Union

# Example 6.32  Use of UNION

**List all cities where there is either a branch office or  a property.**

# …wait a minute!

**Before we start doing SQL statements…**

Know thy SCHEMA

# DreamHome Database

List all cities where there is either a branch office or a property.

| | |
|---|---|
| Branch | (branchNo, street, city, postcode) |
| Staff | (staffNo, fName, lName, position, sex, DOB, salary, branchNo) |
| PropertyForRent | (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo) |
| Client | (clientNo, fName, lName, telNo, prefType, maxRent, email) |
| PrivateOwner | (ownerNo, fName, lName, address, telNo, email, password) |
| Viewing | (clientNo, propertyNo, viewDate, comment) |
| Registration | (clientNo, branchNo, staffNo, dateJoined) |

# Example 6.32 Use of UNION

**List all cities where there is either a branch office or a property.**

```
(SELECT city
FROM Branch
WHERE city IS NOT NULL)
```

# Example 6.32 Use of UNION

**List all cities where there is either a branch office or a property.**

```
(SELECT city
FROM Branch
WHERE city IS NOT NULL)

(SELECT city
FROM PropertyForRent
WHERE city IS NOT NULL);
```

# Example 6.32 Use of UNION

List all cities where there is either a branch office or a property.

```
(SELECT city
FROM Branch
WHERE city IS NOT NULL)
UNION
(SELECT city
FROM PropertyForRent
WHERE city IS NOT NULL);
```

# Example 6.32 Use of UNION

**Produces result tables from both queries and merges both tables together.**

| city |
| --- |
| London |
| Glasgow |
| Aberdeen |
| Bristol |

Not in SQLite

# Example 6.32 Use of UNION

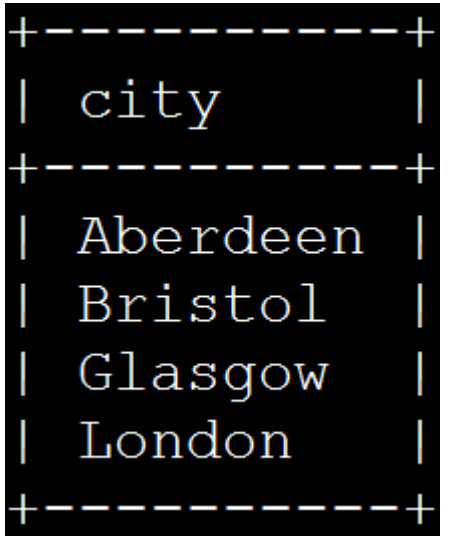
**Or**

```
(SELECT *
        FROM Branch
        WHERE city IS NOT NULL)
        UNION CORRESPONDING BY city
        (SELECT *
        FROM PropertyForRent
        WHERE city IS NOT NULL);
```

# Example 6.32 Use of UNION – ORDER BY

**List all cities where there is either a branch office or a property.**

```
(SELECT city
FROM Branch
WHERE city IS NOT NULL)
UNION
(SELECT city
FROM PropertyForRent
WHERE city IS NOT NULL)
ORDER BY city;
```



| city |
| --- |
| Aberdeen |
| Bristol |
| Glasgow |
| London |

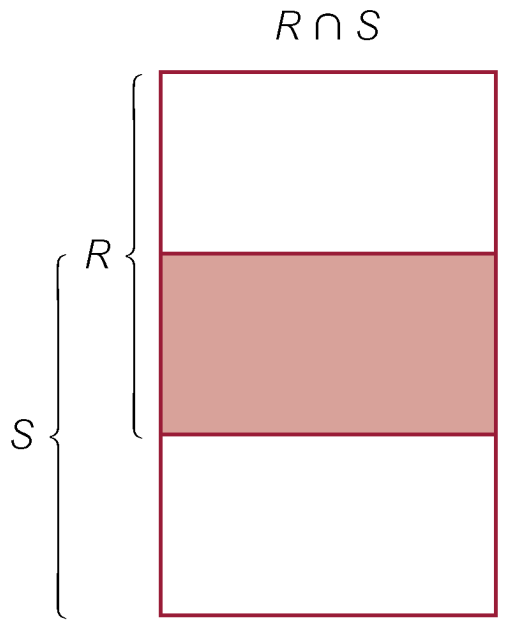
# Example 6.32 Use of UNION – Identifying Parts

**List all cities where there is either a branch office or a property.**

```
(SELECT city, 'branch'
FROM BranchWHERE city IS NOT NULL)
UNION
(SELECT city, 'property'
FROM PropertyForRent
WHERE city IS NOT NULL)
ORDER BY city;
```

| city | branch |
|---|---|
| Aberdeen | branch |
| Aberdeen | property |
| Bristol | branch |
| Glasgow | branch |
| Glasgow | property |
| London | branch |
| London | property |

# Union, Intersect, and Difference (Except)




(b) Intersection

# Example 6.33 Use of INTERSECT

**List all cities where there is both a branch office and a property.**

```
(SELECT city FROM Branch)

(SELECT city FROM PropertyForRent);
```

# Example 6.33 Use of INTERSECT

**List all cities where there is both a branch office and a property.**

```
    (SELECT city FROM Branch)
INTERSECT
(SELECT city FROM PropertyForRent);
```

Not in SQLite

# Example 6.33 Use of INTERSECT

**Or**

```
(SELECT * FROM Branch)
INTERSECT CORRESPONDING BY city
(SELECT * FROM PropertyForRent);
```

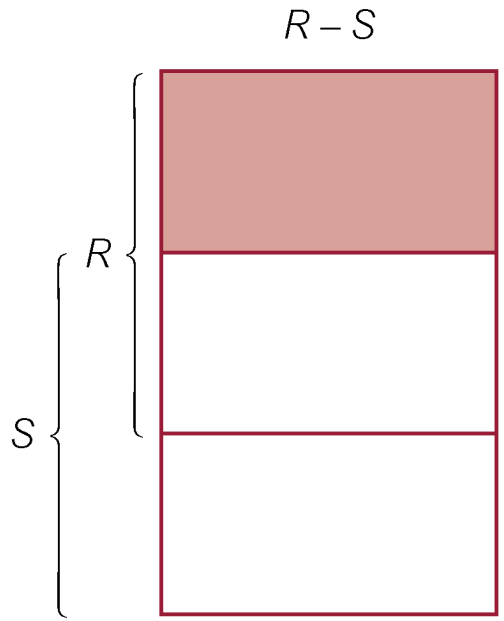
| city |
| --- |
| Aberdeen |
| Glasgow |
| London |

# Example 6.33 Use of INTERSECT

**Rewrite this query without INTERSECT operator:**

```
SELECT DISTINCT city
FROM Branch b
    WHERE EXISTS
            (SELECT *
   FROM PropertyForRent p
            WHERE p.city = b.city);
```

# Union, Intersect, and Difference (Except)




(c) Difference

# Example 6.34 Use of EXCEPT (Set Difference)

List of all cities where there is a branch office but no properties.

```
(SELECT city FROM Branch)

  (SELECT city FROM PropertyForRent);
```

Or

```
  (SELECT * FROM Branch)

  (SELECT * FROM PropertyForRent);
```

| city |
| --- |
| Bristol |

# Example 6.34 Use of EXCEPT (Set Difference)

**List of all cities where there is a branch office but no properties.**

```
(SELECT city FROM Branch)
EXCEPT
(SELECT city FROM PropertyForRent);
```

Can also be MINUS

Or

```
(SELECT * FROM Branch)
EXCEPT CORRESPONDING BY city
(SELECT * FROM PropertyForRent);
```

| city |
| --- |
| Bristol |

# Example 6.34 Use of EXCEPT

Could rewrite this query without EXCEPT:

```
SELECT DISTINCT city
FROM Branch
WHERE city NOT IN
        (SELECT city FROM PropertyForRent);
```

Or

```
SELECT DISTINCT city
FROM Branch b
WHERE NOT EXISTS
        (SELECT * FROM PropertyForRent p
         WHERE p.city = b.city);
```

# Chapter 6 - Objectives

**How to retrieve data from database using SELECT and:**

- **Use compound WHERE conditions.**
- **Use aggregate functions.**
- **Sort query results using ORDER BY.**
- **Group data using GROUP BY and HAVING.**
- **Use subqueries.**
- **Join tables together.**
- **Perform set operations (UNION, INTERSECT, EXCEPT).**