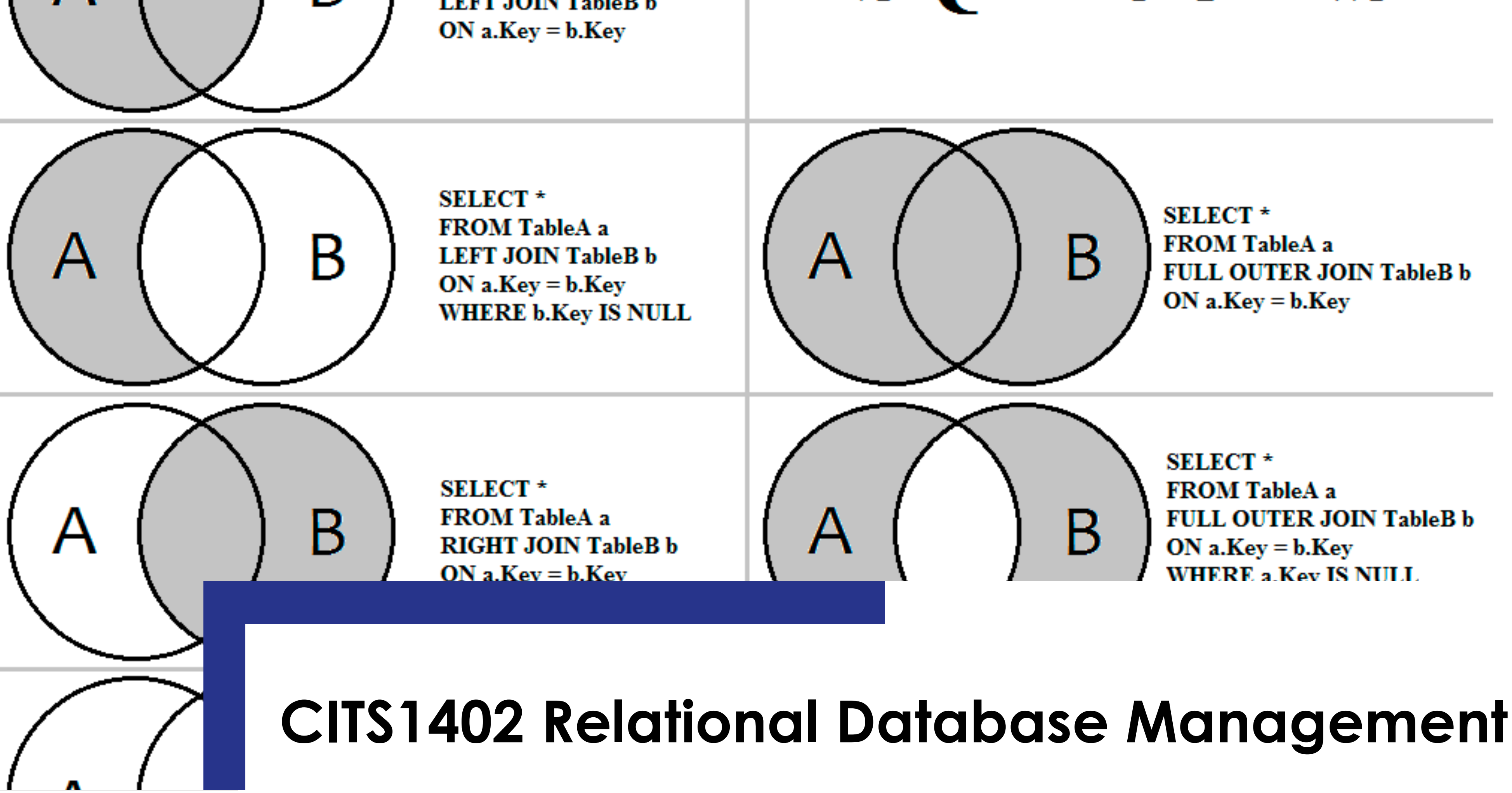


THE UNIVERSITY OF WESTERN AUSTRALIA

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

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).

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]
```

Multi-Table Queries

DreamHome Database

Client (clientNo, fName, lName, telNo, prefType, maxRent, email)

Viewing (clientNo, propertyNo, viewDate, comment)

List names of all clients who have viewed a property

```
SELECT clientNo, fName, lName
from Client
where clientNo IN (select clientNo from Viewing)
```

DreamHome Database

Client (<u>clientNo</u>, fName, lName, telNo, prefType, maxRent, email)

Viewing (<u>clientNo</u>, <u>propertyNo</u>, viewDate, comment)

List names of all clients who have viewed a property along with any comment supplied

SELECT clientNo, fName, lName, propertyNo, comment
from Client ????
where ????

Multi-Table Queries

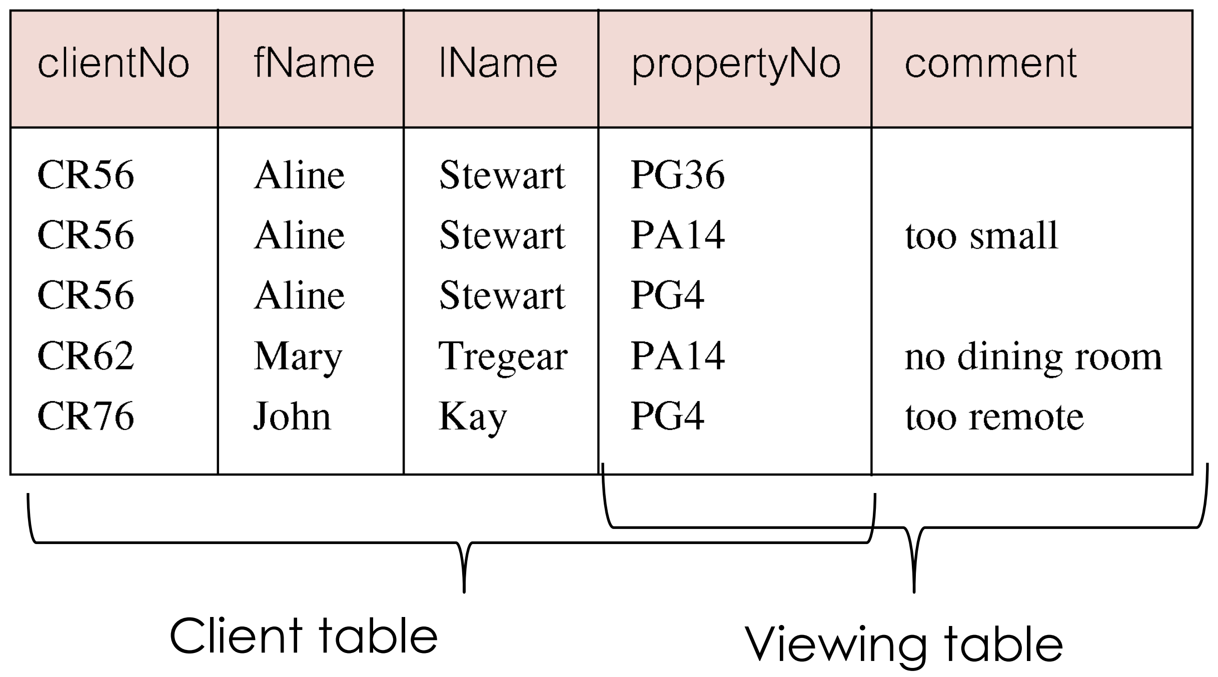

Client table
Viewing table

Table 5.24 Result table for Example 5.24.

clientNo	fName	lName	propertyNo	comment
CR56	Aline	Stewart	PG36	
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR62	Mary	Tregear	PA14	no dining room
CR76	John	Kay	PG4	too remote

Multi-table Query

Multi-Table Queries

Can use subqueries provided result columns come from same table.

If result columns come from more than one table must use a join.

To perform join, include more than one table in FROM clause.

Use comma as separator and typically include WHERE clause to specify join column(s).

Multi-Table Queries

Also possible to use an alias for a table named in FROM clause.

Alias is separated from table name with a space.

Alias can be used to qualify column names when there is ambiguity.

DreamHome Database

Client (clientNo, fName, lName, telNo, prefType, maxRent, email)
Viewing (clientNo, propertyNo, viewDate, comment)

List names of all clients who have viewed a property along with any comment supplied

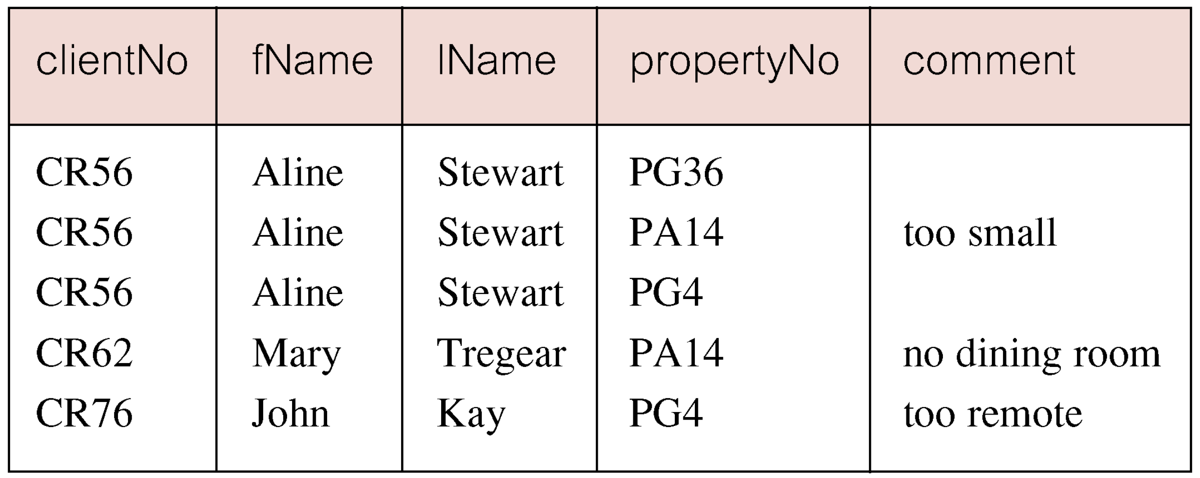
SELECT c.clientNo, fName, lName, propertyNo, comment
from Client c, Viewing v
where ????

Example 6.24 Simple Join

Only those rows from both tables that have identical values in the clientNo columns (c.clientNo = v.clientNo) are included in result.

Equivalent to equi-join in relational algebra.

Table 5.24 Result table for Example 5.24.



clientNo	fName	lName	propertyNo	comment
CR56	Aline	Stewart	PG36	
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR62	Mary	Tregear	PA14	no dining room
CR76	John	Kay	PG4	too remote

DreamHome Database

Client	(clientNo, fName, lName, telNo, prefType, maxRent, email)
Viewing	(clientNo, propertyNo, viewDate, comment)

List names of all clients who have viewed a property along with any comment supplied

```
SELECT c.clientNo, fName, lName, propertyNo, comment
from Client c, Viewing v
where c.clientNo = v.clientNo;
```

Alternative JOIN Constructs

SQL provides alternative ways to specify joins:

```
FROM Client c JOIN Viewing v ON c.clientNo = v.clientNo
FROM Client JOIN Viewing USING clientNo
FROM Client NATURAL JOIN Viewing
```

In each case, FROM replaces original FROM and WHERE.

However, first produces table with two identical clientNo columns.

JOIN USING is more concise and convenient when joining tables with columns of the same name, and it automatically removes duplicate columns. JOIN ON offers more flexibility by allowing you to specify any join condition, and it retains all columns from both tables in the result set.

Example 6.25 Sorting a join

For each branch, list numbers and names of staff who manage properties, and properties they manage.

```
        SELECT s.branchNo, s.staffNo, fName, lName,
               propertyNo
FROM Staff s, PropertyForRent p
WHERE s.staffNo = p.staffNo
ORDER BY s.branchNo, s.staffNo, propertyNo;
```

Example 6.25 Sorting a join

branchNo	staffNo	fName	lName	propertyNo
B003	SG14	David	Ford	PG16
B003	SG37	Ann	Beech	PG21
B003	SG37	Ann	Beech	PG36
B005	SL41	Julie	Lee	PL94
B007	SA9	Mary	Howe	PA14

Example 6.26 Three Table Join

For each branch, list staff who manage properties, including city in which branch is located and properties they manage.

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)

Example 6.26 Three Table Join

For each branch, list staff who manage properties, including city in which branch is located and properties they manage.

```
SELECT b.branchNo, b.city, s.staffNo, fName, lName,
        propertyNo
FROM Branch b, Staff s, PropertyForRent p
WHERE b.branchNo = s.branchNo AND
        s.staffNo = p.staffNo
ORDER BY b.branchNo, s.staffNo, propertyNo;
```

Example 6.26 Three Table Join

branchNo	city	staffNo	fName	lName	propertyNo
B003	Glasgow	SG14	David	Ford	PG16
B003	Glasgow	SG37	Ann	Beech	PG21
B003	Glasgow	SG37	Ann	Beech	PG36
B005	London	SL41	Julie	Lee	PL94
B007	Aberdeen	SA9	Mary	Howe	PA14

Alternative formulation for FROM and WHERE:

```
FROM (Branch b JOIN Staff s USING branchNo) AS bs
    JOIN PropertyForRent p USING staffNo
```

Example 6.27 Multiple Grouping Columns

Find number of properties handled by each staff member.

```
        SELECT s.branchNo, s.staffNo, COUNT(*) AS myCount
FROM Staff s, PropertyForRent p
WHERE s.staffNo = p.staffNo
GROUP BY s.branchNo, s.staffNo
ORDER BY s.branchNo, s.staffNo;
```

Example 6.27 Multiple Grouping Columns

branchNo	staffNo	myCount
B003	SG14	1
B003	SG37	2
B005	SL41	1
B007	SA9	1

Computing a Join

Procedure for generating results of a join are:

1. Form Cartesian product of the tables named in FROM clause.

2. If there is a WHERE clause, apply the search condition to each row of the product table, retaining those rows that satisfy the condition.

3. For each remaining row, determine value of each item in SELECT list to produce a single row in result table.

Computing a Join

4. If DISTINCT has been specified, eliminate any duplicate rows from the result table.

6. If there is an ORDER BY clause, sort result table as required.

SQL provides special format of SELECT for Cartesian product:

```
SELECT     [DISTINCT | ALL]   {* | columnList}
FROM Table1 CROSS JOIN Table2
```

Outer Joins

If one row of a joined table is unmatched, row is omitted from result table.

Outer join operations retain rows that do not satisfy the join condition.

Consider following tables:

Branch1

branchNo	bCity
B003	Glasgow
B004	Bristol
B002	London

PropertyForRent1

propertyNo	pCity
PA14	Aberdeen
PL94	London
PG4	Glasgow

Outer Joins

The (inner) join of these two tables:

```
SELECT b.*, p.*
FROM Branch1 b, PropertyForRent1 p
WHERE b.bCity = p.pCity;
```

branchNo	bCity	propertyNo	pCity
B003	Glasgow	PG4	Glasgow
B002	London	PL94	London

Outer Joins

Result table has two rows where cities are same.

There are no rows corresponding to branches in Bristol and Aberdeen.

To include unmatched rows in result table, use an Outer join.

Example 6.28 Left Outer Join

List branches and properties that are in same city along with any unmatched branches.

```
            SELECT b.*, p.*
  FROM Branch1 b LEFT JOIN
                  PropertyForRent1 p ON b.bCity = p.pCity;
```

Example 6.28 Left Outer Join

Includes those rows of first (left) table unmatched with rows from second (right) table.

Columns from second table are filled with NULLs.

branchNo	bCity	propertyNo	pCity
B003	Glasgow	PG4	Glasgow
B004	Bristol	NULL	NULL
B002	London	PL94	London

Not in SQLite

Example 6.29 Right Outer Join

List branches and properties in same city and any unmatched properties.

```
   SELECT b.*, p.*
FROM Branch1 b RIGHT JOIN
            PropertyForRent1 p ON b.City = p.City;
```

Not in SQLite

Example 6.29 Right Outer Join

Right Outer join includes those rows of second (right) table that are unmatched with rows from first (left) table.

Columns from first table are filled with NULLs.

branchNo	bCity	propertyNo	pCity
NULL	NULL	PA14	Aberdeen
B003	Glasgow	PG4	Glasgow
B002	London	PL94	London

Not in SQLite

Example 6.30 Full Outer Join

List branches and properties in same city and any unmatched branches or properties.

```
    SELECT b.*, p.*
FROM Branch1 b FULL JOIN
        PropertyForRent1 p ON b.bCity = p.pCity;
```

Not in SQLite

Example 6.30 Full Outer Join

Includes rows that are unmatched in both tables.

Unmatched columns are filled with NULLs.

branchNo	bCity	propertyNo	pCity
NULL	NULL	PA14	Aberdeen
B003	Glasgow	PG4	Glasgow
B004	Bristol	NULL	NULL
B002	London	PL94	London

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).