CITS1402 Relational Database Management Systems

Week 6—SQL Subqueries

Sem1/2024

Dr Mehwish Nasim/ CSSE

cits1402-sem1-pmc@uwa.edu.au

CITS1402 Week5

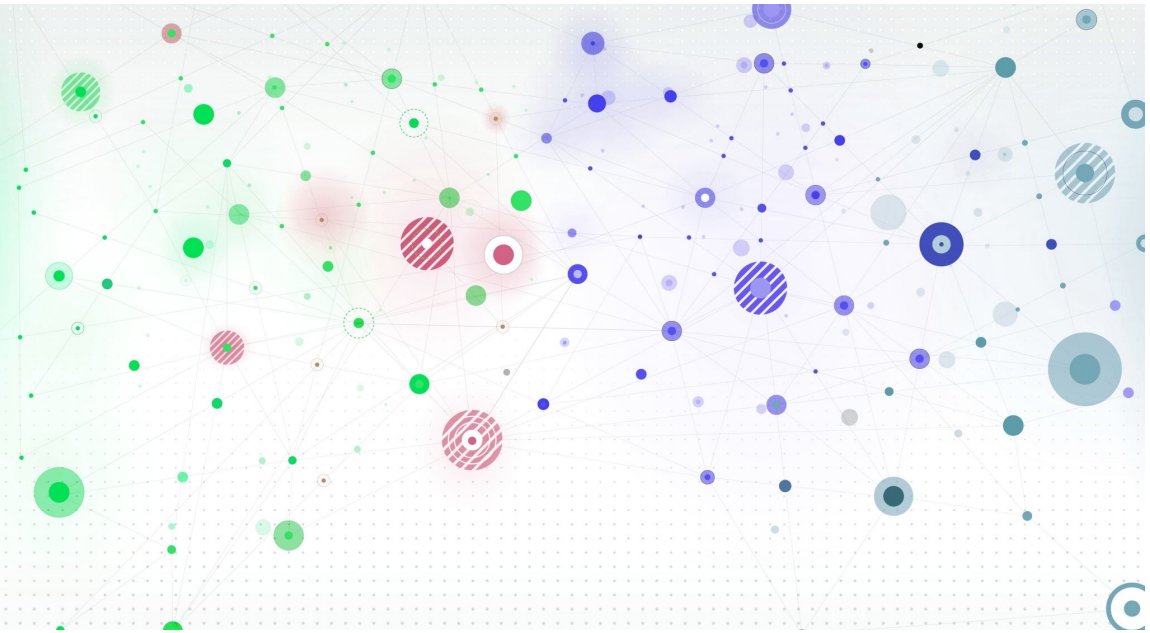
Contents

Chapter 6 - Objectives

THE UNIVERSITY OF WESTERN AUSTRALIA

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]
```

Subqueries

Some SQL statements can have a SELECT embedded within them.

A (inner)SELECT can be used in WHERE and HAVING clauses of an outer SELECT, where it is called a *subquery* or *nested query*.

Subselects may also appear in INSERT, UPDATE, and DELETE statements.

Example 6.19 Subquery with Equality

List staff who work in branch at '163 Main St'.

Before we start doing SQL statements…

Know your SCHEMA

DreamHome Database

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)

DreamHome Database

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 6.19 Subquery with Equality

List staff who work in branch at '163 Main St'.

Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)

Branch (branchNo, street, city, postcode)

Example 6.19 Subquery with Equality

List staff who work in branch at '163 Main St'.

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)

```
SELECT branchNo
FROM Branch
WHERE street = '163 Main St'
```

B003

Example 6.19 Subquery with Equality

List staff who work in branch at '163 Main St'.

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)

```
SELECT branchNo
FROM Branch
WHERE street = '163 Main St'
```

B003

staffNo	fName	lName	position
SG37	Ann	Beech	Assistant
SG14	David	Ford	Supervisor
SG5	Susan	Brand	Manager

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE branchNo = 'B003';
```

Example 6.19 Subquery with Equality

List staff who work in branch at ‘163 Main St’.

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE branchNo =
        ‘B003’
```

Example 6.19 Subquery with Equality

List staff who work in branch at '163 Main St'.

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE branchNo =
        (SELECT branchNo
         FROM Branch
         WHERE street = '163 Main St');
```

subquery returns a
single value
for equality

Example 6.19 Subquery with Equality

Inner SELECT finds branch number for branch at '163 Main St' ('B003').

Outer SELECT then becomes:

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE branchNo = 'B003';
```

Outer SELECT then retrieves details of all staff who work at this branch.

Example 6.20 Subquery with Aggregate

List all staff whose salary is greater than the average salary, and show by how much.

staffNo	fName	lName	position	salDiff
SL21	John	White	Manager	13000.00
SG14	David	Ford	Supervisor	1000.00
SG5	Susan	Brand	Manager	7000.00

Example 6.20 Subquery with Aggregate

Cannot write 'WHERE salary > AVG(salary)'

```
SELECT staffNo, fName, lName, position,
                salary – AVG(salary) AS salDiff
FROM Staff
WHERE salary > AVG(salary);
```

Example 6.20 Subquery with Aggregate

Cannot write 'WHERE salary > AVG(salary)'

Instead, use subquery to find average salary (17000), and then use outer SELECT to find those staff with salary greater than this:

```
SELECT avg(salary)
FROM Staff;
```

1700

```
SELECT staffNo, fName, lName, position,
        salary – 17000 As salDiff
FROM Staff
WHERE salary > 17000;
```

Example 6.20 Subquery with Aggregate

List all staff whose salary is greater than the average salary, and show by how much.

```
SELECT staffNo, fName, lName, position,
  salary – (SELECT AVG(salary) FROM Staff) As SalDiff
FROM Staff
WHERE salary > (SELECT AVG(salary)
                FROM Staff);
```

Subquery Rules

Subquery Rules

Subquery must always be enclosed in parenthesis

```
SELECT staffNo,
FROM Staff
WHERE branchNo =
      SELECT branchNo
       FROM Branch
       WHERE street = '163 Main St';
```

no parenthesis ()

```
SELECT staffNo,
FROM Staff
WHERE branchNo =
      (SELECT branchNo
       FROM Branch
       WHERE street = '163 Main St');
```

Subquery Rules

ORDER BY clause may not be used in a subquery although it may be used in outermost SELECT.

```
SELECT staffNo,
FROM Staff
WHERE branchNo =
        (SELECT branchNo
         FROM Branch
         WHERE street = '163 Main St'
         ORDER BY branchNo);
```

INVALID

```
SELECT staffNo,
FROM Staff
WHERE branchNo =
        (SELECT branchNo
         FROM Branch
         WHERE street = '163 Main St');
ORDER BY staffNo;
```

Subquery Rules

Subquery SELECT list must consist of a single column name or expression except for subqueries that use EXISTS.

```
SELECT staffNo,
FROM Staff
WHERE branchNo =
      (SELECT branchNo, street
       FROM Branch
       WHERE street = '163 Main St');
```

INVALID: two columns

Subquery Rules

By default, column names in a subquery refer to table in FROM clause of that subquery.

Can refer to a outer table in FROM using an *alias*.

- **only reference outer queries**
- **correlated subquery (more later)**

```
SELECT staffNo,
FROM Staff
WHERE branchNo =
      (SELECT branchNo
       FROM Branch
       WHERE street = '163 Main St');
```

Subquery Rules – ISO Standard

When subquery is an operand in a comparison, subquery must appear on right-hand side.

```
SELECT propertyNo, rent
FROM PropertyForRent
WHERE rent > (SELECT avg(rent) FROM PropertyForRent);
```

a subquery cannot be compared to a subquery!

```
SELECT propertyNo, rent
FROM PropertyForRent
WHERE (SELECT avg(rent) FROM PropertyForRent) < rent;
```

invalid

Subquery Rules

A subquery may not be used as an operand in an expression.

```
SELECT propertyNo, rent
FROM PropertyForRent
WHERE rent > (SELECT max(rent) FROM PropertyForRent)/2;
```

Subquery Rules

A subquery may not be used as an operand in an expression.

```
SELECT propertyNo, rent
FROM PropertyForRent
WHERE rent > (SELECT max(rent) FROM PropertyForRent)/2;
```

```
SELECT propertyNo, rent
FROM PropertyForRent
WHERE rent*2 > (SELECT max(rent) FROM PropertyForRent);
```

Subquery Rules

When subquery is an operand in a comparison, subquery must appear on right-hand side.

A subquery may not be used as an operand in an expression.

These two rules may not always apply!

Subqueries for multi-row results

- Previous subqueries return one row results
  - Use with =, <, >, etc.
- Subqueries that return multi-row results must use (ISO standard)
  - IN
  - ALL
  - ANY/SOME

Example 6.21 Nested subquery: use of IN

List properties handled by staff at ‘163 Main St’.

…wait a minute!

Before we start doing SQL statements…

Know thy SCHEMA

Example 6.21 Nested subquery: use of IN

List properties handled by staff at '163 Main St'.

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 6.21 Nested subquery: use of IN

List properties handled by staff at '163 Main St'.

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 6.21 Nested subquery: use of IN

List properties handled by staff at '163 Main St'.

```
SELECT propertyNo, street, city, postcode, type, rooms, rent
FROM PropertyForRent
WHERE staffNo IN
  (SELECT staffNo
   FROM Staff
   WHERE branchNo =
                (SELECT branchNo
                 FROM Branch
                 WHERE street = '163 Main St'));
```

{'SG37', 'SG14', 'SG5'}

'B003'

Example 6.21 Nested subquery: use of IN

propertyNo	street	city	postcode	type	rooms	rent
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375
PG21	18 Dale Rd	Glasgow	G12	House	5	600

ANY and ALL

ANY and ALL may be used with subqueries that produce
  a single column of numbers
  and possible multiple rows

ALL: condition will only be true if it is satisfied by *all* values produced by subquery.

ANY: condition will be true if it is satisfied by *any* values produced by subquery.

If subquery is empty:
  ALL returns true, ANY returns false.

SOME may be used in place of ANY.

ANY and ALL

ANY and ALL NOT support by SQLite!!!

Can replace with correct use of

- **>, <, <=, >=**
- **and**
- **max/min aggregate functions**

Example 6.22 Use of ANY/SOME

Find staff whose salary is larger than salary of at least one member of staff at branch B003.

Example 6.22 Use of ANY/SOME

Find staff whose salary is larger than salary of at least one member of staff at branch B003.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary > SOME
                (SELECT salary
                 FROM Staff
                 WHERE branchNo = 'B003');
```

{12000, 18000, 24000}

Example 6.22 Use of ANY/SOME

Inner query produces set {12000, 18000, 24000} and outer query selects those staff whose salaries are greater than any of the values in this set.

staffNo	fName	lName	position	salary
SL21	John	White	Manager	30000.00
SG14	David	Ford	Supervisor	18000.00
SG5	Susan	Brand	Manager	24000.00

Example 6.22 Use of ANY/SOME

Find staff whose salary is larger than salary of at least one member of staff at branch B003.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary > SOME
                (SELECT salary
                 FROM Staff
                 WHERE branchNo = 'B003');
```

{12000, 18000, 24000}

Example 6.22 Use of ANY/SOME - SQLite

Find staff whose salary is larger than salary of at least one member of staff at branch B003.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary >
        (SELECT min(salary)
         FROM Staff
         WHERE branchNo = 'B003');
```

{12000}

Example 6.23 Use of ALL

Find staff whose salary is larger than salary of every member of staff at branch B003.

Example 6.23 Use of ALL

Find staff whose salary is larger than salary of every member of staff at branch B003.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary > ALL
                (SELECT salary
                 FROM Staff
                 WHERE branchNo = 'B003');
```

Example 6.23 Use of ALL

staffNo	fName	lName	position	salary
SL21	John	White	Manager	30000.00

Example 6.23 Use of ALL

Find staff whose salary is larger than salary of every member of staff at branch B003.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary > ALL
                (SELECT salary
                 FROM Staff
                 WHERE branchNo = 'B003');
```

Example 6.23 Use of ALL - SQLite

Find staff whose salary is larger than salary of every member of staff at branch B003.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary >
        (SELECT max(salary)
         FROM Staff
         WHERE branchNo = 'B003');
```

The Correlated Subquery

- An UNcorrelated subquery
- Subquery is independent of outer query

```
SELECT staffNo, fName, lName, position, salary
        FROM Staff
        WHERE salary > ALL
                        (SELECT salary
                         FROM Staff
                         WHERE branchNo = 'B003');
```

The Correlated Subquery

- Correlated subquery:
  - inner query refers to columns in outer query
  - inner query needs to be re-evaluated for each row returned from the outer query

The Correlated Subquery

Find the staffNo, lname and fname of the staff member with the minimum salary for each branch.

Uncorrelated attempt – for each…GROUP BY

```
SELECT branchNo, MIN(salary)
FROM Staff
GROUP BY branchNo;
```

Not in aggregate function

```
SELECT branchNo, staffNo, fName, lName, MIN(salary)
FROM Staff
GROUP BY branchNo;
```

The Correlated Subquery

Find the staffNo, lname and fname of the staff member with the minimum salary for each branch.

ch…GROUP BY

branchNo	staffNo	fName	lName	min(salary)
B003	SG37	Ann	Beech	12000
B005	SL21	John	White	9000
B007	SA9	Mary	Howe	9000

```
SELECT branchNo, staffNo, fName, lName, MIN(salary)
FROM Staff
GROUP BY branchNo;
```

The Correlated Subquery

Find the staffNo
member with
branch.

branchNo	staffNo	fname	lname	salary
B005	SL21	John	White	30000
B003	SG37	Ann	Beech	12000
B003	SG14	David	Ford	18000
B007	SA9	Mary	Howe	9000
B003	SG5	Susan	Brand	24000
B005	SL41	Julie	Lee	9000

ch...GROUP BY

branchNo	staffNo	fName	lName	min(salary)
B003	SG37	Ann	Beech	12000
B005	SL21	John	White	9000
B007	SA9	Mary	Howe	9000

SELECT branchNo, staffNo, fName, lName, MIN(salary)
FROM Staff
GROUP BY branchNo;

The Correlated Subquery

Find the staffNo member with branch.

branchNo	staffNo	fname	lname	salary
B005	SL21	John	White	30000
B003	SG37	Ann	Beech	12000
B003	SG14	David	Ford	18000
B007	SA9	Mary	Howe	9000
B003	SG5	Susan	Brand	24000
B005	SL41	Julie	Lee	9000

ch...GROUP BY

branchNo	staffNo	fName	lName	min(salary)
B003	SG37	Ann	Beech	12000
B005	SL21	John	White	9000
B007	SA9	Mary	Howe	9000

SELECT branchNo, staffNo, fName, lName, MIN(salary)
FROM Staff
GROUP BY branchNo;

Not in aggregate function

The Correlated Subquery

Find the staffNo, lname and fname of the staff member with the minimum salary for each branch.

Correlated attempt – find the minimum by checking if each row of Staff is less then all others at their branch

SELECT branchNo, staffNo, fName, lName, salary
FROM Staff
WHERE salary <= ALL *the others at the branch*

The Correlated Subquery

Find the staffNo, lname and fname of the staff member with the minimum salary for each branch.

Correlated attempt – find the minimum by checking if each row of Staff is less then all others at their branch

```
SELECT branchNo, staffNo, fName, lName, salary
FROM Staff s1
WHERE salary <= ALL (SELECT salary
                     FROM Staff s2
                     WHERE s2.branchNo = s1.branchNo);
```

The Correlated Subquery

```
SELECT branchNo, staffNo, fName, lName, salary
FROM Staff s1
WHERE salary <= ALL (…)
```

branchNo	staffNo	fname	lname	salary
B005	SL21	John	White	30000
B003	SG37	Ann	Beech	12000
B003	SG14	David	Ford	18000
B007	SA9	Mary	Howe	9000
B003	SG5	Susan	Brand	24000
B005	SL41	Julie	Lee	9000

```
SELECT salary
FROM Staff s2
WHERE s2.branchNo =
'B005'
```

salary
30000
9000

```
SELECT branchNo, staffNo, fName, lName, salary
FROM Staff s1
WHERE 30000 <= ALL (30000,9000)
```

```
SELECT branchNo, staffNo, fName, lName, salary
FROM Staff s1
WHERE false
```

The Correlated Subquery

re-evaluate with next branchNo

SELECT branchNo, staffNo, fName, lName, salary
FROM Staff s1
WHERE salary <= ALL (…)

branchNo	staffNo	fname	lname	salary
B005	SL21	John	White	30000
B003	SG37	Ann	Beech	12000
B003	SG14	David	Ford	18000
B007	SA9	Mary	Howe	9000
B003	SG5	Susan	Brand	24000
B005	SL41	Julie	Lee	9000

SELECT salary
FROM Staff s2
WHERE s2.branchNo = 'B003'

salary
12000
18000
24000

SELECT branchNo, staffNo, fName, lName, salary
FROM Staff s1
WHERE 12000 <= ALL (12000,18000,24000)

SELECT branchNo, staffNo, fName, lName, salary
FROM Staff s1
WHERE true

The Correlated Subquery

Find the staffNo, lname and fname of the staff member with the minimum salary for each branch.

branchNo	staffNo	fName	lName	salary
B003	SG37	Ann	Beech	12000
B007	SA9	Mary	Howe	9000
B005	SL41	Julie	Lee	9000

Correlated attempt – find the minimum by checking if each row of Staff is less then all others

```
SELECT branchNo, staffNo, fName, lName, salary
FROM Staff s1
WHERE salary <= ALL (SELECT salary
                     FROM Staff s2
                     WHERE s2.branchNo = s1.branchNo);
```

EXISTS and NOT EXISTS

EXISTS and NOT EXISTS are for use only with subqueries

Can be used with subqueries that have many columns

Produce a simple true/false result.

True if and only if there exists at least one row in result table returned by subquery.

False if subquery returns an empty result table.

NOT EXISTS is the opposite of EXISTS.

EXISTS and NOT EXISTS

As (NOT) EXISTS check only for existence or non-existence of rows in subquery result table, subquery can contain any number of columns.

Common for subqueries following (NOT) EXISTS to be of form:

(SELECT * ...)

Example 6.31 Query using EXISTS

Find all staff who work in a London branch.

```
SELECT staffNo, fName, lName, position
FROM Staff s
WHERE EXISTS
        (SELECT *
         FROM Branch b
         WHERE s.branchNo = b.branchNo AND
                city = 'London');
```

Example 6.31 Query using EXISTS

staffNo	fName	lName	position
SL21	John	White	Manager
SL41	Julie	Lee	Assistant

Example 6.31 Query using EXISTS

Note, search condition s.branchNo = b.branchNo is necessary to consider correct branch record for each member of staff.

If omitted, would get all staff records listed out because subquery:

```
SELECT * FROM Branch WHERE city='London'
```

would always be true and query would be:

```
SELECT staffNo, fName, lName, position FROM Staff
WHERE true;
```

Can also achieve similar result using IN

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).**