Example 5.6
Compound Comparison Search Condition

List addresses of all branch offices in London or Glasgow.

```
SELECT *
    FROM Branch
    WHERE city = 'London' OR city = 'Glasgow';
```

Table 5.6 Result table for Example 5.6.

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B003	163 Main St	Glasgow	G11 9QX
B002	56 Clover Dr	London	NW10 6EU

Example 5.7 Range Search Condition

List all staff with a salary between 20,000 and 30,000.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary BETWEEN 20000 AND 30000;
```

- BETWEEN test includes the endpoints of range.

Example 5.7 Range Search Condition

Table 5.7 Result table for Example 5.7.

staffNo	fName	lName	position	salary
SL21	John	White	Manager	30000.00
SG5	Susan	Brand	Manager	24000.00

Example 5.7 Range Search Condition

- Also a negated version **NOT BETWEEN.**
- BETWEEN does not add much to SQL's expressive power. Could also write:

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary>=20000 AND salary <= 30000;
```

- Useful, though, for a range of values.

Example 5.8 Set Membership

List all managers and supervisors.

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE position IN ('Manager', 'Supervisor');
```

Table 5.8 Result table for Example 5.8.

staffNo	fName	lName	position
SL21	John	White	Manager
SG14	David	Ford	Supervisor
SG5	Susan	Brand	Manager

Example 5.8 Set Membership

- There is a negated version (NOT IN).
- IN does not add much to SQL's expressive power.
- Could have expressed this as:

```
SELECT staffNo, fName, lName, position
FROM Staff
WHERE position='Manager' OR
                position='Supervisor';
```

- IN is more efficient when set contains many values.

Example 5.9 Pattern Matching

Find all owners with the string 'Glasgow' in their address.

```
SELECT clientNo, fName, lName, address, telNo
FROM PrivateOwner
WHERE address LIKE '%Glasgow%';
```

Table 5.9 Result table for Example 5.9.

ownerNo	fName	lName	address	telNo
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025

Example 5.9 Pattern Matching

- SQL has two special pattern matching symbols:
  - **%**: sequence of zero or more characters;
  - _ (underscore): any single character.
- LIKE '%Glasgow%' means a sequence of characters of any length containing '*Glasgow*'.

Example 5.10 NULL Search Condition

List details of all viewings on property PG4 where a comment has not been supplied.

- There are 2 viewings for property PG4, one with and one without a comment.
- Have to test for null explicitly using special keyword **IS NULL**:

```
SELECT clientNo, viewDate
    FROM Viewing
    WHERE propertyNo = 'PG4' AND
                comment IS NULL;
```

Example 5.10 NULL Search Condition

Table 5.10 Result table for Example 5.10.

clientNo	viewDate
CR56	26-May-01

- Negated version (IS NOT NULL) can test for non-null values.

Example 5.11 Single Column Ordering

List salaries for all staff, arranged in descending order of salary.

```
SELECT staffNo, fName, lName, salary
    FROM Staff
    ORDER BY salary DESC;
```

Example 5.11 Single Column Ordering

Table 5.11 Result table for Example 5.11.

staffNo	fName	lName	salary
SL21	John	White	30000.00
SG5	Susan	Brand	24000.00
SG14	David	Ford	18000.00
SG37	Ann	Beech	12000.00
SA9	Mary	Howe	9000.00
SL41	Julie	Lee	9000.00

Example 5.12 Multiple Column Ordering

Produce abbreviated list of properties in order of property type.

```
SELECT propertyNo, type, rooms, rent
    FROM PropertyForRent
    ORDER BY type;
```

Example 5.12 Multiple Column Ordering

Table 5.12(a) Result table for Example 5.12 with one sort key.

propertyNo	type	rooms	rent
PL94	Flat	4	400
PG4	Flat	3	350
PG36	Flat	3	375
PG16	Flat	4	450
PA14	House	6	650
PG21	House	5	600

Example 5.12 Multiple Column Ordering

- Four flats in this list - as no minor sort key specified, system arranges these rows in any order it chooses.
- To arrange in order of rent, specify minor order:

```
SELECT propertyNo, type, rooms, rent
    FROM PropertyForRent
    ORDER BY type, rent DESC;
```

Example 5.12 Multiple Column Ordering

Table 5.12(b) Result table for Example 5.12 with two sort keys.

propertyNo	type	rooms	rent
PG16	Flat	4	450
PL94	Flat	4	400
PG36	Flat	3	375
PG4	Flat	3	350
PA14	House	6	650
PG21	House	5	600

SELECT Statement - Aggregates

ISO standard defines five aggregate functions:

COUNT returns number of values in specified column.

SUM returns sum of values in specified column.

AVG returns average of values in specified column.

MIN returns smallest value in specified column.

MAX returns largest value in specified column.

SELECT Statement - Aggregates

Each operates on a single column of a table and returns a single value.

- COUNT, MIN, and MAX apply to numeric and non-numeric fields, but **SUM and AVG may be used on numeric fields only**.
- Apart from COUNT(*), each function eliminates nulls first and operates only on remaining non-null values.

SELECT Statement - Aggregates

COUNT(*) counts all rows of a table, regardless of whether nulls or duplicate values occur.

- Can use DISTINCT before column name to eliminate duplicates.
- DISTINCT has no effect with MIN/MAX, but may have with SUM/AVG.

SELECT Statement - Aggregates

Aggregate functions can be used only in SELECT list and in HAVING clause.

- If SELECT list includes an aggregate function and there is no GROUP BY clause, SELECT list cannot reference a column out with an aggregate function. For example, the following is illegal:

```
SELECT staffNo, COUNT(salary)
FROM Staff;
```

Example 5.13 Use of COUNT(*)

How many properties cost more than £350 per month to rent?

```
SELECT COUNT(*) AS count
    FROM PropertyForRent
    WHERE rent > 350;
```

Table 5.13 Result table for Example 5.13.

count
5

Example 5.14 Use of COUNT(DISTINCT)

How many different properties viewed in May '01?

```
SELECT COUNT(DISTINCT propertyNo) AS count
FROM Viewing
WHERE viewDate BETWEEN '1-May-01'
      AND '31-May-01';
```

Table 5.14 Result table for Example 5.14.

count
2

Example 5.15 Use of COUNT and SUM

Find number of Managers and sum of their salaries.

```
SELECT COUNT(staffNo) AS count,
       SUM(salary) AS sum
FROM Staff
WHERE position = 'Manager';
```

Table 5.15 Result table for Example 5.15.

count	sum
2	54000.00