

SQL subqueries

- Intro
- Inner & outer query
- Key word IN vs =
- How does the DBMS evaluate?
- Name clashes
- EXISTS and NOT EXISTS

Intro

- Subqueries can be used as an alternative to joins
- A subquery is a SELECT inside another SELECT

```
SELECT room_no, types, price
FROM room
WHERE hotel_no IN
  (SELECT hotel_no
   FROM hotel
   WHERE address LIKE '%Roskilde%');
```

Inner & outer query

```
SELECT room_no, types, price
FROM room
WHERE hotel_no IN
  (SELECT hotel_no
   FROM hotel
   WHERE address LIKE '%Roskilde%');
```

- The result from the *inner query* (subquery) is used as input to the *outer query*

Key word IN vs =

```
SELECT room_no, types, price
FROM room
WHERE hotel_no IN
  (SELECT hotel_no
   FROM hotel
   WHERE address LIKE '%Roskilde%');
```

- Use the key word **IN**, when there is more than one result from the subquery - otherwise use **=**

How does the DBMS evaluate?

```
SELECT room_no, types, price
FROM room
WHERE hotel_no IN
  (SELECT hotel_no
   FROM hotel
   WHERE address LIKE '%Roskilde%');
```

1. Inner query is evaluated: e.g. hotel_no = 6, 7
2. Result is used in outer query, so final SQL sentence is evaluated as:

```
SELECT room_no, types, price
FROM room
WHERE hotel_no IN (6, 7);
```

Name clashes, I

- Fields in different tables can have the same name – name clash
- You can qualify the name by pre-fixing it with the table name
 - Room.hotel_no
 - Hotel.hotel_no
- Aliases can also be used

Name clashes, II

```
SELECT Room.Room_No, Room.Types,  
       Room.Price, Hotel.Name  
FROM Room, Hotel  
WHERE Hotel.Hotel_No IN  
       (SELECT Hotel.Hotel_No  
        FROM Hotel  
        WHERE Hotel.Address LIKE  
        '%Roskilde%');
```

EXISTS and NOT EXISTS

- The keywords EXISTS and NOT EXISTS can only be used with subqueries, not joins
- EXISTS is true, if there exists at least one row in the result table returned by the subquery
- EXISTS is false, if the subquery returns an empty result table
- NOT EXISTS is the opposite of EXISTS

Example EXISTS

Example 5.31 Query using EXISTS

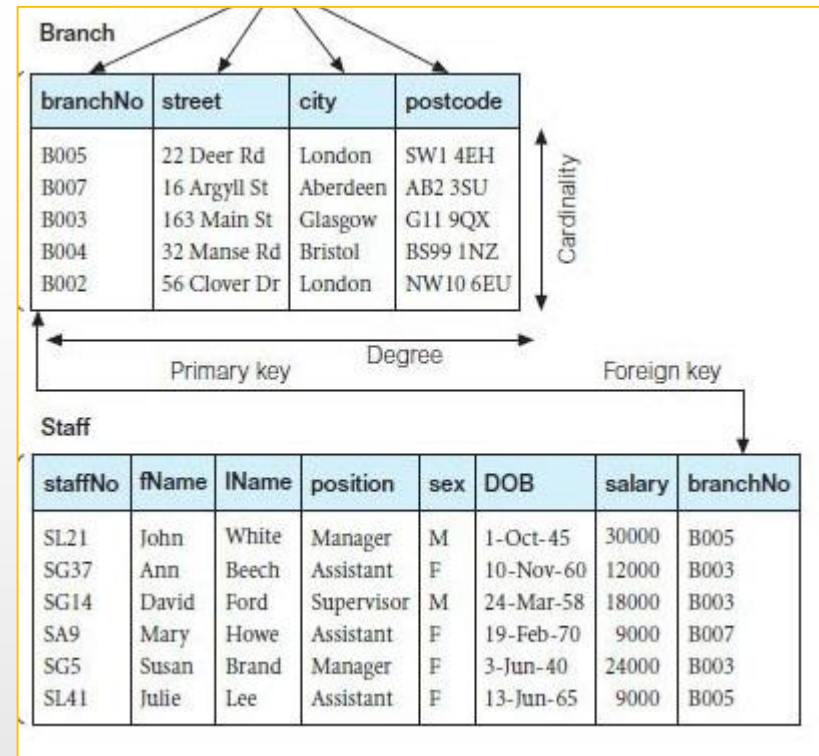
Find all staff who work in a London branch office.

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

This query could be rephrased as 'Find all staff such that there exists a Branch row containing his/her branch number, branchNo, and the branch city equal to London'. The test for inclusion is the existence of such a row. If it exists, the subquery evaluates to true. The result table is shown in Table 5.31.

Table 5.31 Result table for Example 5.31.

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



We could also have written this query using the join construct:

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

Exercise – subqueries

- With the data in place for the ***HotelDB database*** then run the below queries:
 - How many bookings are there at Scandic hotel today? Today could be set to a specific date, if you don't want to use a date function.