



SQL language: basics

Nested queries



Nested queries

Introduction

Introduction

- A nested query is a **SELECT** statement contained within another query
 - query nesting allows decomposing a complex problem into simpler subproblems
- **SELECT** statements may be introduced
 - within a predicate in the **WHERE** clause
 - within a predicate in the **HAVING** clause
 - in the **FROM** clause

Supplier and part DB (1/2)

- P (PId, PName, Color, Size, Store)
- S (SId, SName, #Employees, City)
- SP (SId, PId, Qty)

Supplier and part DB (1/2)

P

<u>PId</u>	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London

SP

<u>SId</u>	<u>PId</u>	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P3	200
S4	P4	300
S4	P5	400

S

<u>SId</u>	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Nested queries (no.1)

- Find the codes of the suppliers that are based in the same city as S1

- By using a formulation with nested queries, the problem may be decomposed into two subproblems
 - city of supplier S1
 - codes of the suppliers based in the same city

Nested queries (no.1)

➤ Find the codes of the suppliers that are based in the same city as S1

Codes of the suppliers based in the same city as S1

```
SELECT SId
FROM S
WHERE City = (SELECT City
              FROM S
              WHERE SId='S1');
```

City of the supplier of S1

➤ The '=' operator may be used only if it is known in advance that the inner SELECT statement always returns a single value

Equivalent formulation (no.1)

- Find the codes of the suppliers that are based in the same city as S1
- An equivalent formulation may be defined using a join operation

Equivalent formulation

- The equivalent formulation with join is characterized by
- a FROM clause including the tables referenced by the FROM clauses of each SELECT statement
 - appropriate join conditions in the WHERE clause
 - possible selection predicates added in the WHERE clause

FROM clause (no.1)

- Find the codes of the suppliers that are based in the same city as S1

```
SELECT SId
FROM (S) ← SX
WHERE City = (SELECT City
              FROM (S) ← SY
              WHERE SId='S1');
```

FROM clause (no.1)

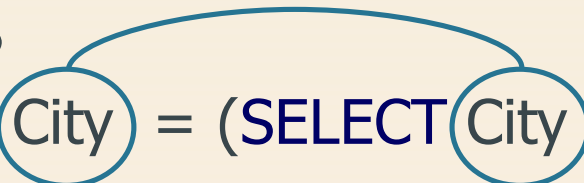
- Find the codes of the suppliers that are based in the same city as S1

```
SELECT ...  
FROM S AS SX, S AS SY  
...
```

Join condition (no.1)

- Find the codes of the suppliers that are based in the same city as S1

```
SELECT SId
FROM S
WHERE City = (SELECT City
              FROM S
              WHERE SId='S1');
```

A blue curved line connects the 'City' column of the outer query to the 'City' column of the subquery, indicating the join condition.

Join condition (no.1)

- Find the codes of the suppliers that are based in the same city as S1

```
SELECT ...  
FROM S AS SX, S AS SY  
WHERE SX.City=SY.City  
...
```

Selection predicate (no.1)

- Find the codes of the suppliers that are based in the same city as S1

```
SELECT SId
FROM S
WHERE City = (SELECT City
              FROM S
              WHERE SId='S1');
```

SELECT clause (no.1)

- Find the codes of the suppliers that are based in the same city as S1

```
SELECT SY.SId  
FROM S AS SX, S AS SY  
WHERE SX.City=SY.City AND  
SX.SId='S1';
```

Equivalent formulation (no.2)

- Find the codes of the suppliers whose number of employees is smaller than the maximum number of employees

```
SELECT SId
FROM S
WHERE #Employees < (SELECT MAX(#Employees)
                     FROM S);
```

- Is it possible to define an equivalent formulation with join?

Equivalent formulation (no.2)

- Find the codes of the suppliers whose number of employees is smaller than the maximum number of employees

```
SELECT SId
FROM S
WHERE #Employees < (SELECT MAX(#Employees)
                     FROM S);
```

- An equivalent formulation with join is not possible



Nested queries

The IN operator

The IN operator (no.1)

- Find the names of the suppliers that provide product P2

- Decomposition of the problem into two subproblems
 - codes of the suppliers of product P2
 - names of the suppliers with such codes

The IN operator (no.1)

➤ Find the names of the suppliers that provide product P2

SP

<u>SId</u>	<u>PId</u>	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P3	200
S4	P4	300
S4	P5	400



SId
S1
S2
S3

```
(SELECT SId  
FROM SP  
WHERE PId='P2')
```

*Codes
of the
suppliers
of P2*

The IN operator (no.1)

➤ Find the names of the suppliers that provide product P2

```
SELECT SName
FROM S
WHERE SId IN (SELECT SId
              FROM SP
              WHERE PId='P2');
```

Set membership

Codes of the suppliers of P2

The IN operator

- It expresses the concept of membership to a set of values
 - *AttributeName* IN (*NestedQuery*)
- It allows writing a query by
 - decomposing the problem into subproblems
 - following a “bottom-up” procedure

Equivalent formulation

- The equivalent formulation with join is characterized by
- a FROM clause including the tables referenced by the FROM clauses of each SELECT statement
 - appropriate join conditions in the WHERE clause
 - possible selection predicates added in the WHERE clause

Equivalent formulation

- Find the names of the suppliers that provide product P2

```
SELECT SName
FROM S, SP
WHERE S.SId=SP.SId
      AND PId='P2';
```

```
SELECT SName
FROM S
WHERE SId IN (SELECT SId
              FROM SP
              WHERE PId='P2');
```

The IN operator (no.2)

- Find the names of the suppliers that supply at least one red product
- Decomposition of the problem into subproblems
 - codes of the red products
 - codes of the suppliers of such products
 - names of the suppliers with such codes

The IN operator (no.2)

- Find the names of the suppliers that supply at least one red product

```
SELECT SName
FROM S
WHERE SId IN (SELECT SId
              FROM SP
              WHERE PId IN (SELECT PId
                           FROM P
                           WHERE Color='Red'));
```

Equivalent formulation (no.2)

- Find the names of the suppliers that supply at least one red product

```
SELECT SName
FROM S
WHERE SId IN (SELECT SId
              FROM SP
              WHERE PId IN (SELECT PId
                            FROM P
                            WHERE Color='Red'));
```

FROM clause (no.2)

- Find the names of the suppliers that supply at least one red product

```
SELECT SName
```

```
FROM (S)
```

```
WHERE SId IN (SELECT SId
```

```
FROM (SP)
```

```
WHERE PId IN (SELECT PId
```

```
FROM (P)
```

```
WHERE Color='Red'));
```

FROM clause (no.2)

- Find the names of the suppliers that supply at least one red product

```
SELECT ...
```

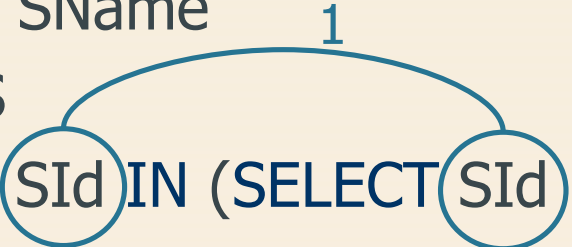
```
FROM S, SP, P
```

```
...
```

Join conditions (no.2)

- Find the names of the suppliers that supply at least one red product

```
SELECT SName
FROM S
WHERE (SId) IN (SELECT (SId)
                FROM SP
                WHERE PId IN (SELECT PId
                              FROM P
                              WHERE Color='Red'));
```



Join conditions (no.2)

- Find the names of the suppliers that supply at least one red product

```
SELECT ...
```

```
FROM S, SP, P
```

```
WHERE S.SId=SP.SId
```

1

Join conditions (no.2)

- Find the names of the suppliers that supply at least one red product

```
SELECT SName
```

```
FROM S
```

```
WHERE SId IN (SELECT SId
```

```
FROM SP
```

```
WHERE PId IN (SELECT PId
```

```
FROM P
```

```
WHERE Color='Red'));
```

2

Join conditions (no.2)

➤ Find the names of the suppliers that supply at least one red product

```
SELECT ...  
FROM S, SP, P  
WHERE S.SId=SP.SId AND  
      SP.CodP=S.CodP  
...
```

2

Selection predicate (no.2)

- Find the names of the suppliers that supply at least one red product

```
SELECT SName
FROM S
WHERE SId IN (SELECT SId
              FROM SP
              WHERE PId IN (SELECT PId
                            FROM P
                            WHERE Color='Red')));
```

SELECT clause (no.2)

➤ Find the names of the suppliers that supply at least one red product

```
SELECT DISTINCT SName  
FROM S, SP, P  
WHERE S.SId=SP.SId AND  
       SP.PId=P.PId AND  
       Color='Red';
```




Nested queries

The NOT IN operator

Concept of exclusion (no.1)

- Find the names of the suppliers that *do not* supply product P2
- is it possible to express the query with a join operation?

```
SELECT SName
FROM S, SP
WHERE S.SId=SP.SId
      AND PId<>'P2';
```

Wrong solution (no.1)

- Find the names of the suppliers that *do not* supply product P2
- the query may not be expressed by means of a join

```
SELECT SName  
FROM S, SP  
WHERE S.SId=SP.SId  
AND PId<>'P2';
```


Wrong solution (no.1)

➤ Find the names of the suppliers that *do not* supply product P2

S

<u>SId</u>	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

R



SName
Smith
Jones
Clark

SP

<u>SId</u>	<u>PId</u>	Qty
<i>S1</i>	<i>P1</i>	<i>300</i>
S1	P2	200
<i>S1</i>	<i>P3</i>	<i>400</i>
<i>S1</i>	<i>P4</i>	<i>200</i>
<i>S1</i>	<i>P5</i>	<i>100</i>
<i>S1</i>	<i>P6</i>	<i>100</i>
<i>S2</i>	<i>P1</i>	<i>300</i>
S2	P2	400
S3	P2	200
<i>S4</i>	<i>P3</i>	<i>200</i>
<i>S4</i>	<i>P4</i>	<i>300</i>
<i>S4</i>	<i>P5</i>	<i>400</i>

Wrong solution (no.1)

```
SELECT SName  
FROM S, SP  
WHERE S.SId=SP.SId  
      AND PId<>'P2';
```

➤ Which query does it answer?

Wrong solution (no.1)

```
SELECT SName  
FROM S, SP  
WHERE S.SId=SP.SId  
      AND PId<>'P2';
```



Find the names of the suppliers that supply
at least one product other than P2

Concept of exclusion (no.1)

- Find the names of the suppliers that *do not* supply product P2
- We need to exclude from the result
 - the suppliers that supply product P2

The NOT IN operator (no.1)

- Find the names of the suppliers that *do not* supply product P2

```
SELECT SName  
FROM S  
WHERE SId NOT IN (SELECT SId  
FROM SP  
WHERE PId='P2');
```

does not belong to

*Codes of the suppliers
that supply P2*

The NOT IN operator

- It expresses the concept of exclusion from a set of values
 - *AttributeName* NOT IN (*NestedQuery*)
- It requires the identification of an appropriate *set to be excluded*
 - defined by the nested query

The NOT IN operator (no.2)

- Find the names of the suppliers that *only* supply product P2



Find the names of the suppliers of P2 that have never supplied products other than P2

- Set to be excluded
 - suppliers of products other than P2

The NOT IN operator (no.2)

- Find the names of the suppliers that only supply product P2

```
SELECT SId  
FROM SP  
WHERE PId <> 'P2'
```

*Codes of the suppliers
that supply
at least one product
other than P2*

The NOT IN operator (no.2)

- Find the names of the suppliers that only supply product P2

```
SELECT SName
FROM S
WHERE SId NOT IN (SELECT SId
                  FROM SP
                  WHERE PId<>'P2')
```

...

The NOT IN operator (no.2)

- Find the names of the suppliers that only supply product P2

```
SELECT SName
FROM S, SP
WHERE S.SId NOT IN (SELECT SId
                    FROM SP
                    WHERE PId<>'P2')
AND S.SId=SP.SId;
```

Alternative solution (no.2)

- Find the names of the suppliers that only supply product P2

```
SELECT SName
FROM S
WHERE S.SId NOT IN (SELECT SId
                    FROM SP
                    WHERE PId<>'P2')
AND S.SId IN (SELECT SId
              FROM SP);
```

The NOT IN operator (no.3)

➤ Find the names of the suppliers that *do not* supply any red products

➤ P (PId, PName, Color, Size, Store)

➤ S (SId, SName, #Employees, City)

➤ SP (SId, PId, Qty)

The NOT IN operator (no.3)

- Find the names of the suppliers that *do not* supply any red products
- Set to be excluded?
 - suppliers of red products, identified by their codes

The NOT IN operator (no.3)

- Find the names of the suppliers that *do not* supply any red products

*Codes of the suppliers
of red products*

```
(SELECT SId
FROM SP
WHERE PId IN (SELECT PId
FROM P
WHERE Color='Red'))
```

The NOT IN operator (no.3)

- Find the names of the suppliers that *do not* supply any red products

```
SELECT SName
FROM S
WHERE SId NOT IN (SELECT SId
                  FROM SP
                  WHERE PId IN (SELECT PId
                               FROM P
                               WHERE Color='Red'));
```