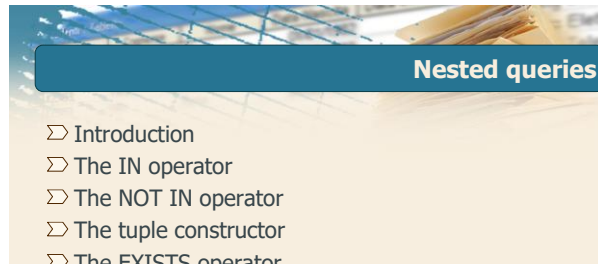


SQL language: basics

Nested queries




1

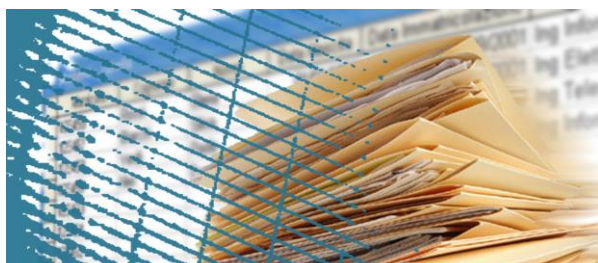


Nested queries

- ▷ Introduction
- ▷ The IN operator
- ▷ The NOT IN operator
- ▷ The tuple constructor
- ▷ The EXISTS operator
- ▷ The NOT EXISTS operator
- ▷ Correlation among queries
- ▷ The division operation
- ▷ Table functions




2

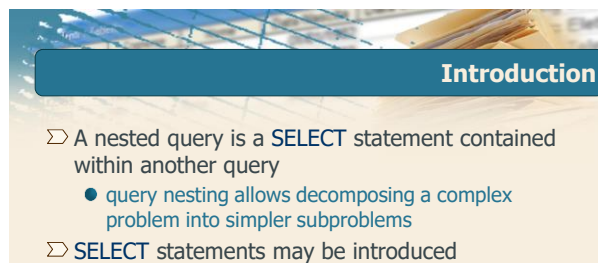


Nested queries

Introduction




3



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



4




Supplier and part DB (1/2)

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



5




Supplier and Product database

Foreign keys

PId	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

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

SId	PId	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



6

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



7

Nested queries (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');
```

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



8

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



9

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



10

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');
```



11

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
...
```



12

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');
```



13

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
...
```



14

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');
```



15

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';
```



16

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?



17

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



18

Nested queries

The IN operator

DBG

19

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

DBG

20

The IN operator (no.1)

⊃ Find the names of the suppliers that provide product P2

SP

SId	PId	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*

DBG

21

The IN operator (no.1)

⊃ Find the names of the suppliers that provide product P2

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

DBG

22

The IN operator (no.1)

⊃ Find the names of the suppliers that provide product P2

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

?

DBG

23

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

DBG

24

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



25

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



26

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');
```



27

Equivalent formulation (no.1)

- ▷ Find the names of the suppliers that provide product P2

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



28

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



29

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'));
```




30

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'));
```




31

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'));
```




32

FROM clause (no.2)

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

```
SELECT ...
FROM S, SP, P
...
```




33

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'));
```






34

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
```

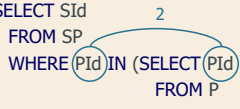




35

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'));
```

36

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

DBG
MG

37

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'));
```

DBG
MG

38

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';
```

DBG
MG

39

A complex example (no.3)

- ▷ Find the names of the suppliers that supply at least one product supplied by suppliers of red products

P	PId	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	

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

SP	SId	PId	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	

DBG
MG

40

A complex example (no.3)

- ▷ Find the names of the suppliers that supply at least one product supplied by suppliers of red products

DBG
MG

41

A complex example (no.3)

- ▷ Find the names of the suppliers that supply at least one product supplied by suppliers of red products

- ▷ The formulation with join is awkward
- it is easier to decompose the problem into subproblems through nested queries

DBG
MG

42


A complex example (no.3)

Find the names of the suppliers that supply at least one product supplied by suppliers of red products

```

SELECT PId
FROM P
WHERE Color='Red'
    
```

Codes of the red products



43


A complex example (no.3)

Find the names of the suppliers that supply at least one product supplied by suppliers of red products

```

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

Codes of the suppliers of red products



44


A complex example (no.3)

Find the names of the suppliers that supply at least one product supplied by suppliers of red products

```

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

Codes of the products supplied by suppliers of red products



45


A complex example (no.3)

Find the names of the suppliers that supply at least one product supplied by suppliers of red products

```

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

Codes of the suppliers of the products supplied by suppliers of red products



46

Complete query (no.3)



```

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



47

Formulation with join (no.3)





48

Formulation with join (no.3)

```

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




49

FROM clause (no.3)

```

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




50

FROM clause (no.3)

```

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

Diagram annotations: SPA points to the innermost SP query, SPB to the middle SP query, and SPC to the outer SP query.




51

FROM clause (no.3)

```

SELECT ...
FROM S, SP AS SPA, SP AS SPB, SP AS SPC, P
...
    
```




52

Join conditions (no.3)

```

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

Diagram annotation: A circled '1' connects the SId in the FROM clause to the SId in the WHERE clause of the innermost SP query.




53

Join conditions (no.3)

```

SELECT ...
FROM S, SP AS SPA, SP AS SPB, SP AS SPC, P
WHERE S.SId=SPA.SId
...
    
```

Diagram annotation: A circled '1' is placed next to the join condition S.SId=SPA.SId.



54

Join conditions (no.3)

```

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

DBG

55

Join conditions (no.3)

```

SELECT ...
FROM S, SP AS SPA, SP AS SPB, SP AS SPC, P
WHERE S.SId=SPA.SId AND
      SPA.PId=SPB.PId
...
    
```

2

DBG

56

Join conditions (no.3)

```

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

DBG

57

Join conditions (no.3)

```

SELECT ...
FROM S, SP AS SPA, SP AS SPB, SP AS SPC, P
WHERE S.SId=SPA.SId AND
      SPA.PId=SPB.PId AND
      SPB.SId=SPC.SId
...
    
```

3

DBG

58

Join conditions (no.3)

```

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

DBG

59

Join conditions (no.3)

```

SELECT ...
FROM S, SP AS SPA, SP AS SPB, SP AS SPC, P
WHERE S.SId=SPA.SId AND
      SPA.PId=SPB.PId AND
      SPB.SId=SPC.SId AND
      SPC.PId=P.PId
...
    
```

4

DBG

60

Selection predicate (no.3)

```

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

```

DBG

61

Selection predicate (no.3)

```

SELECT ...
FROM S, SP AS SPA, SP AS SPB, SP AS SPC, P
WHERE S.SId=SPA.SId AND
      SPA.PId=SPB.PId AND
      SPB.SId=SPC.SId AND
      SPC.PId=P.PId AND
      Color='Red'

```

DBG

62

SELECT statement (no.3)

```

SELECT DISTINCT SName
FROM S, SP AS SPA, SP AS SPB, SP AS SPC, P
WHERE S.SId=SPA.SId AND
      SPA.PId=SPB.PId AND
      SPB.SId=SPC.SId AND
      SPC.PId=P.PId AND
      Color='Red';

```

DBG

63

Nested queries

The NOT IN operator

DBG

64

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';

```

DBG

65

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';


```

DBG

66

Wrong solution (no.1)

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

Sid	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Sid	PId	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

R

SName
Smith
Jones
Clark

67

Wrong solution (no.1)

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

Which query does it answer?

68

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

69

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

70

Concept of exclusion (no.1)

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

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

Codes of the suppliers that supply P2

71

Concept of exclusion (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');
```

Codes of the suppliers that supply P2

72


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



73

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



74

NOT IN and relational algebra (no.1)

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




75

NOT IN and relational algebra (no.1)

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

p: S.SId=SP.SId



76

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




77

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



78

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')
...
```



79

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;
```



80

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);
```



81

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)



82

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



83

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'))
```




84

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'));
```




85

Alternative (correct?) (no.3)

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

Codes of the suppliers that supply at least one non-red product

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




86

Alternative (correct?) (no.3)

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

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




87

Wrong alternative (no.3)

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

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




88

Wrong alternative (no.3)

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

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

Codes of the suppliers of non-red products



89


Wrong alternative (no.3)

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

PId	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

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

SId	PId	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



90

Wrong alternative (no.3)

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

PId	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

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

Wrong SQL:

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

The set of elements to be excluded is incorrect

91

Wrong alternative (no.3)

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

Wrong SQL:

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

The set of elements to be excluded is incorrect

92

Nested queries

The tuple constructor

93

The tuple constructor

- It allows defining a temporary structure for a tuple
 - the attributes belonging to the tuple must be listed within ()

(AttributeName₁, AttributeName₂, ...)

- It enhances the expressive power of the IN and NOT IN operators

94

Example (no.1)

TRIP (TId, StartingPlace, Destination, DepartureTime, ArrivalTime)

Find the pairs of starting places and destinations for which none of the trips lasts more than 2 hours

95

Example (no.1)

TRIP (TId, StartingPlace, Destination, DepartureTime, ArrivalTime)

Find the pairs of starting places and destinations for which none of the trips lasts more than 2 hours

SQL:

```
SELECT StartingPlace, Destination
FROM TRIP
WHERE (StartingPlace, Destination) NOT IN
(SELECT StartingPlace, Destination
FROM TRIP
WHERE ArrivalTime-DepartureTime>2);
```

Tuple constructor

96

Nested queries

The EXISTS operator

97

The EXISTS operator (no.1)

▷ Find the names of the suppliers of product P2

↓

Find the names of the suppliers *for which there exists a product supply for P2*

98

Correlation condition (no.1)

▷ Find the names of the suppliers of product P2

```

SELECT SName
FROM S
WHERE EXISTS (SELECT *
              FROM SP
              WHERE PId='P2'
              AND SP.SId=S.SId);
    
```

Correlation condition

99

How EXISTS works (no.1)

▷ Find the names of the suppliers of product P2

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

SId	PId	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

```

SELECT *
FROM SP
WHERE PId='P2'
AND SP.SId='S1';
    
```

Value of SId in the current line of table S

100

How EXISTS works (no.1)

▷ Find the names of the suppliers of product P2

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

SId	PId	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

▷ The predicate including EXISTS is true for S1 since there exists a supply for P2 by S1

- S1 belongs to the result of the query

101

How EXISTS works (no.1)

▷ Find the names of the suppliers of product P2

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

SId	PId	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

▷ The predicate including EXISTS is false for S4 since there exists no supply for P2 by S4

- S4 is not included in the result of the query

102

Result of the query (no.1)

- ▷ Find the names of the suppliers of product P2

R
SName
Smith
Jones
Blake



103

Predicates with EXISTS

- ▷ The predicate including EXISTS is
- true if the inner query returns at least one tuple
 - false if the inner query returns the empty set



104

Predicates with EXISTS

- ▷ The predicate including EXISTS is
- true if the inner query returns at least one tuple
 - false if the inner query returns the empty set
- ▷ In the query inside EXISTS, the SELECT clause is mandatory but irrelevant, since its attributes are never displayed
- ▷ The correlation condition binds the execution of the inner query to the values of attributes of the current tuple in the outer query



105

Scope of attributes

- ▷ A nested query may reference attributes defined within outer queries
- ▷ A query may not reference attributes defined
- within a nested query at an inner level
 - within a different query at the same level



106



Nested queries

The NOT EXISTS operator



107

The NOT EXISTS operator (no.1)

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



Find the names of the suppliers for which there does not exist a product supply for P2



108


The NOT EXISTS operator (no.1)

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

```

SELECT SName
FROM S
WHERE NOT EXISTS (SELECT *
                  FROM SP
                  WHERE PId='P2'
                  AND SP.SId=S.SId );
    
```

Correlation condition



109


How NOT EXISTS works (no.1)

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

S

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

SELECT * FROM SP WHERE PId='P2' AND SP.SId='S1' *value of SId in the current line of table S*



110

How NOT EXISTS works (no.1)

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


S

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

SELECT * FROM SP WHERE PId='P2' AND SP.SId='S1'

SP

SId	PId	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



111

How NOT EXISTS works (no.1)

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

S


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

SP

SId	PId	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

The predicate including NOT EXISTS is false for S1 since there exists a supply for P2 by F1

- S1 *does not* belong to the result of the query



112

How NOT EXISTS works (no.1)


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

S

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

SP

SId	PId	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




113

How NOT EXISTS works (no.1)

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

S

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



114

How NOT EXISTS works (no.1)

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

S

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

SP

SId	PId	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

DBG

115

How NOT EXISTS works (no.1)

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

S

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

SP

SId	PId	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

The predicate including NOT EXISTS is true of S4 since there exists no supply for P2 by S4

- S4 belongs to the result of the query

DBG

116

How NOT EXISTS works (no.1)

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

S

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

SP

SId	PId	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

DBG

117

Result of the query (no.1)

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

R

SName
Clark
Adams

DBG

118

Predicates with NOT EXISTS

- The predicate including NOT EXISTS is
 - true if the inner query returns the empty set
 - false if the inner query return at least one tuple
- The correlation condition binds the execution of the inner query to the values of attributes of the current tuple in the outer query

DBG

119

Nested queries

Correlation among queries

DBG

120

Correlation among queries

- ▷ It may be required to bind the computation of a nested query to the value(s) of one or more attributes in an outer query
 - the binding is expressed by one or more correlation conditions



121

Correlation condition

- ▷ A correlation condition
 - must be specified in the WHERE clause of the nested query that requires it
 - is a predicate that binds some attributes of tables appearing in the nested query's FROM clause to attributes of tables appearing in the FROM clause of outer queries
- ▷ Correlation conditions may not be expressed
 - within queries at the same nesting level
 - with references to attributes of a table appearing in the FROM clause of a nested query



122

Correlation among queries (no.1)

- ▷ For each product, find the code of the supplier that provides the highest quantity



123

Correlation among queries (no.1)

- ▷ For each product, find the code of the supplier that provides the highest quantity

```
SELECT PId, SId
FROM SP AS SPX
WHERE Qty = (...
```

) } *Maximum quantity for the current product*



124

Correlation among queries (no.1)

- ▷ For each product, find the code of the supplier that provides the highest quantity

```
SELECT PId, SId
FROM SP AS SPX
WHERE Qty = (SELECT MAX(Qty)
             FROM SP AS SPY
             ... ) } Maximum quantity
```



125

Correlation among queries (no.1)

- ▷ For each product, find the code of the supplier that provides the highest quantity

```
SELECT PId, SId
FROM SP AS SPX
WHERE Qty = (SELECT MAX(Qty)
             FROM SP AS SPY
             WHERE SPY.PId=SPX.PId); } Maximum quantity for the current product
```



126

Correlation among queries (no.1)

- ▷ For each product, find the code of the supplier that provides the highest quantity

```
SELECT PId, SId
FROM SP AS SPX
WHERE Qty = (SELECT MAX(Qty)
             FROM SP AS SPY
             WHERE SPY.PId=SPX.PId);
```

Correlation condition



127

Correlation among queries (no.1) – Alternative solution

- ▷ For each product, find the code of the supplier that provides the highest quantity

```
SELECT PId, SId
FROM SP AS SPX
WHERE (PID, QTY) IN (SELECT PID, Max(Qty)
                   FROM SP
                   GROUP BY PID)
```



128

Correlation among queries (no.1) – Wrong solution

- ▷ For each product, find the code of the supplier that provides the highest quantity

```
SELECT SId
FROM SP AS SPX
WHERE QTY = (SELECT Max(Qty)
            FROM SP
            GROUP BY PID)
```



129

Correlation among queries (no.1) – Wrong solution

- ▷ For each product, find the code of the supplier that provides the highest quantity

```
SELECT SId
FROM SP AS SPX
WHERE QTY = (SELECT Max(Qty)
            FROM SP)
```



130

Correlation among queries (no.1) – Wrong solution

- ▷ For each product, find the code of the supplier that provides the highest quantity

```
SELECT SId
FROM SP AS SPX
WHERE QTY = (SELECT Max(Qty)
            FROM SP SPY
            WHERE SPY.PID=SPX.PID
            GROUP BY PID)
```



131

Correlation among queries (no.2)

TRIP (TId, StartingPlace, Destination, DepartureTime, ArrivalTime)

- ▷ Find the codes of the trips whose duration is lower than the average duration of the trips on the same route (i.e., same starting place and destination)

```
SELECT TId
FROM TRIP AS TA
WHERE ArrivalTime-DepartureTime < (...
```

Average duration of trips on the current route



132

Correlation among queries (no.2)


TRIP (Tid, StartingPlace, Destination, DepartureTime, ArrivalTime)

- Find the codes of the trips whose duration is lower than the average duration of the trips on the same route (i.e., same starting place and destination)

```

SELECT Tid
FROM TRIP AS TA
WHERE ArrivalTime-DepartureTime <
  (SELECT AVG(ArrivalTime-DepartureTime)
   FROM TRIP AS TB
   ... )
    
```

Average duration of trips



133

Correlation among queries (no.2)


TRIP (Tid, StartingPlace, Destination, DepartureTime, ArrivalTime)

- Find the codes of the trips whose duration is lower than the average duration of the trips on the same route (i.e., same starting place and destination)

```

SELECT Tid
FROM TRIP AS TA
WHERE ArrivalTime-DepartureTime <
  (SELECT AVG(ArrivalTime-DepartureTime)
   FROM TRIP AS TB
   WHERE TB.StartingPlace=TA.StartingPlace
   AND TB.Destination=TA.Destination);
    
```


Correlation conditions



134

Nested queries


The division operation



135

The division operation (no.1)


- Find the codes of the suppliers that supply *all* products
- In relational algebra we must use the division operator



136

The division operation (no.1)

- Find the codes of the suppliers that supply *all* products
- In relational algebra we must use the division operator

$$\begin{array}{c}
 R \\
 \parallel \\
 \pi_{SId, PId} \quad / \quad \pi_{PId} \\
 \text{SP} \qquad \qquad \text{P}
 \end{array}$$



137

Division in SQL (no.1)

- Find the codes of the suppliers that supply *all* products
- Remark
 - all products that may be supplied are listed in table P

↓

- a supplier is supplying all products if he is supplying a number of distinct products equal to the cardinality of P




138

Division in SQL (no.1)

Find the codes of the suppliers that supply *all* products

```
SELECT COUNT(*)
FROM P
```




139

Division in SQL (no.1)

Find the codes of the suppliers that supply *all* products

```
SELECT COUNT(*)
FROM P
```

Total number of products




140

Division in SQL (no.1)

Find the codes of the suppliers that supply *all* products

```
SELECT SId
FROM SP
GROUP BY SId
... (SELECT COUNT(*)
FROM P)
```




141

Division in SQL (no.1)

Find the codes of the suppliers that supply *all* products

```
SP(SID, PID, Qty)

SELECT SId
FROM SP
GROUP BY SId
HAVING COUNT(*)=(SELECT COUNT(*)
FROM P);
```




142

Division in SQL (no.1) – Different SP TABLE

Find the codes of the suppliers that supply *all* products

```
SP(SID, PID, Date, Qty)

SELECT SId
FROM SP
GROUP BY SId
HAVING COUNT(DISTINCT PID)=(SELECT COUNT(*)
FROM P);
```



143


Division in SQL: procedure (no.2)

Find the codes of the suppliers that supply at least *all* of the products supplied by supplier S2

We must count

- the number of products supplied by S2
- the number of products supplied both by an arbitrary supplier and by S2
- (Alternative formulation)
- Among the products supplied by S2, count the number of products supplied by an arbitrary supplier

The two counts must be equal



144

Division in SQL (no.2)

- ▷ Find the codes of the suppliers that supply at least *all* of the products supplied by supplier S2

```
SELECT COUNT(*)
FROM SP
WHERE SId='S2'
```



145

Division in SQL (no.2)

- ▷ Find the codes of the suppliers that supply at least *all* of the products supplied by supplier S2

```
SELECT COUNT(*)
FROM SP
WHERE SId='S2'
```

Number of products supplied by S2



146

Division in SQL (no.2)

- ▷ Find the codes of the suppliers that supply at least *all* of the products supplied by supplier S2

```
SELECT SId
FROM SP
WHERE PId IN (SELECT PId
              FROM SP
              WHERE SId='S2')
GROUP BY SId
... (SELECT COUNT(*)
      FROM SP
      WHERE SId='S2')
```



147

Division in SQL (no.2)

- ▷ Find the codes of the suppliers that supply at least *all* of the products supplied by supplier S2

```
SELECT SId
FROM SP
WHERE PId IN (SELECT PId
              FROM SP
              WHERE SId='S2')
GROUP BY SId
HAVING COUNT(*)=(SELECT COUNT(*)
                  FROM SP
                  WHERE SId='S2');
```



148