SQL language: basics

The SELECT statement: basics

DBMG

The SELECT statement: basics

- Basic structure
- WHERE clause
- Result ordering
- Join

The SELECT statement: basics

Basic structure

The SELECT statement: example

- Find the codes and the number of employees of the suppliers based in Paris

Supplier and part DB

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

The SELECT statement: example

- Find the codes and the number of employees of the suppliers based in Paris

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

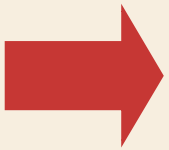
The SELECT statement: example

- Find the codes and the number of employees of the suppliers based in Paris

```
SELECT SId, #Employees
FROM   S
WHERE City='Paris';
```

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



R

SId	#Employees
S2	10
S3	30

Basic SELECT (no.1)

- Find the codes of all products in the database

```
SELECT PId
FROM P;
```

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

R

PId
P1
P2
P3
P4
P5
P6

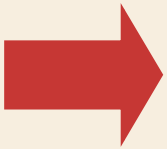
Basic SELECT (no.2)

- Find the codes of the products supplied by at least one supplier

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

```
SELECT PId
FROM SP;
```



It does not eliminate duplicates

R

PId
P1
P2
P3
P4
P5
P6
P1
P2
P2
P3
P4
P5

Elimination of duplicates

- DISTINCT keyword
  - elimination of duplicates

- Find the codes of the *distinct* products supplied by at least one supplier

Basic SELECT (no.2)

- Find the codes of the *distinct* products supplied by at least one supplier

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

```
SELECT DISTINCT PId
FROM SP;
```

R

PId
P1
P2
P3
P4
P5
P6

Selection of all information

- Find all information related to products

```
SELECT PId, PName, Color, Weight, Store
FROM P;
```

or

```
SELECT *
FROM P;
```

R

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

Selection with an expression (1/3)

- Find the codes of the products and the sizes expressed with the US standard

```
SELECT PId, Size-14
FROM P;
```

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

R

PId	
P1	26
P2	34
P3	34
P4	30
P5	26
P6	28

Selection with an expression (2/3)

- Definition of a new *temporary* column for the computed expression
  - the name of the temporary column may be defined by means of the AS keyword

Selection with an expression (3/3)

- Find the codes of the products and the sizes expressed with the US standard

```
SELECT PId, Size-14 AS USSize
FROM P;
```

R

PId	USSize
P1	26
P2	34
P3	34
P4	30
P5	26
P6	28

Structure of the SELECT statement

SELECT [DISTINCT] *ListOfAttributesToDisplay*
FROM *ListOfTablesToUse;*

The SELECT statement: basics

The WHERE clause

DBMG

The WHERE clause

- It allows expressing selection conditions applied to each tuple individually
- A Boolean expression of predicates
- Simple predicates
  - comparison expressions between attributes and constants
  - text search
  - NULL values

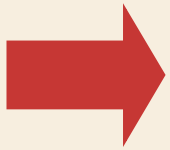
The WHERE clause (no.1)

- Find the codes of the suppliers based in Paris

```
SELECT SId
FROM S
WHERE City='Paris';
```

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



R

SId
S2
S3

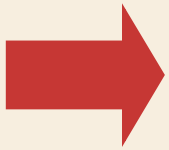
The WHERE clause (no.2)

- Find the codes and the number of employees of the suppliers that are not based in Paris

```
SELECT SId, #Employees
FROM S
WHERE City<>'Paris';
```

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



R

SId	#Employees
S1	20
S4	20
S5	30

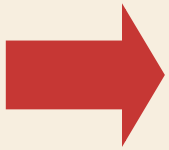
Boolean expressions (no.1)

- Find the codes of the suppliers based in Paris that have more than 20 employees

```
SELECT SId
FROM S
WHERE City='Paris' AND #Employees>20;
```

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



R

SId
S3

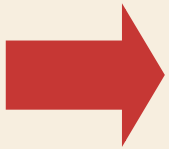
Boolean expressions (no.2)

- Find the codes and the number of employees of the suppliers based in Paris or London

```
SELECT SId, #Employees
FROM S
WHERE City='Paris' OR City='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



R

SId	#Employees
S1	2
S2	1
S3	3
S4	2

Boolean expressions (no.3)

- Find the codes and the number of employees of the suppliers based in Paris and in London
  - the query may not be satisfied
    - each supplier has only one city

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

Text search

- LIKE operator

  AttributeName LIKE *CharacterString*

  - the _ character represents a single arbitrary character (non-empty)
  - the % character represents an arbitrary sequence of characters (possibly empty)

Text search (no.1)

- Find the codes and the names of the products whose name begins with the letter B

```
SELECT PId, PName
FROM P
WHERE PName LIKE 'B%';
```

P

PId	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	**B**louse	Blue	48	Rome
P4	**B**louse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London

R

PId	PName
P3	Blouse
P4	Blouse

Text search (no.2)

- The Address attribute contains the string ‘London’

```
Address LIKE '%London%'
```

Text search (no.3)

- The supplier identification number is 3 and
  - it is preceded by a single unknown character
  - it is exactly 2 characters long

```
SId LIKE '_3'
```

Text search (no.4)

- The Store attribute does not have an ‘e’ in the second position

```
Store NOT LIKE  '_e%'
```

Managing NULL values (no.1)

- Find the codes and the names of products with a size greater than 44

```
SELECT PId, PName
FROM P
WHERE Size>44;
```

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	NULL	Paris
P6	Shorts	Red	42	London

R

PId	PName
P2	Jeans
P3	Blouse

The NULL value

- The tuples with a NULL size are not selected
  - the predicate Size>44 evaluates to false
- With NULL values, any comparison predicate is false

Searching for NULL values

- IS special operator

AttributeName IS [NOT] NULL

Searching for NULL values (no.1)

- Find the codes and the names of the products whose size is unknown

```
SELECT PId, PName
FROM P
WHERE Size IS NULL;
```

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	NULL	Paris
P6	Shorts	Red	42	London

R

PId	PName
P5	Skirt

Searching for NULL values (no.2)

- Find the codes and the names of products with a size greater than 44, or that may have a size greater than 44

```
SELECT PId, PName
FROM P
WHERE Size>44 OR Size IS NULL;
```

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	NULL	Paris
P6	Shorts	Red	42	London

R

PId	PName
P2	Jeans
P3	Blouse
P5	Skirt

Structure of the SELECT statement (2)

SELECT [DISTINCT] *ListOfAttributesToDisplay*
FROM *ListOfTablesToUse*
[WHERE *TupleConditions*];

The SELECT statement: basics

Result ordering

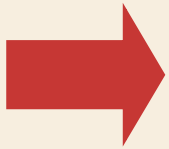
Result ordering (no.1)

- Find the codes of the products and their sizes, ordering the result by decreasing size

```
SELECT PId, Size
FROM P
ORDER BY Size DESC;
```

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



R

PId	Size
P2	48
P3	48
P4	44
P6	42
P1	40
P5	40

Ordering

- ORDER BY clause

  ORDER BY *AttributeName* [ASC | DESC]
  {, *AttributeName* [ASC | DESC]}

  - the default ordering is ascending
    - if DESC is not specified
  - the ordering attributes must appear in the SELECT clause
    - even implicitly (as in SELECT *)

Result ordering (no.2)

- Find all information related to the products, ordering the result by increasing name order and decreasing size

```
SELECT PId, PName, Color, Size, Store
FROM P
ORDER BY PName, Size DESC;
```

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

Result ordering (no.2)

- Find all information related to the products, ordering the result by increasing name order and decreasing size

```
SELECT PId, PName, Color, Size, Store
FROM P
ORDER BY PName, Size DESC;
```

R

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

Result ordering (no.2)

- Find all information related to the products, ordering the result by increasing name order and decreasing size

```
SELECT *
FROM P
ORDER BY PName, Size DESC;
```

R

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

Result ordering (no.3)

- Find the codes of the products and the sizes expressed with the US standard, ordering the result by increasing size

```
SELECT PId, Size-14 AS USSize
FROM P
ORDER BY USSize;
```

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

R

PId	USSize
P5	26
P1	26
P6	28
P4	30
P2	34
P3	34

Structure of the SELECT statement

SELECT [DISTINCT] *ListOfAttributesToDisplay*
FROM *ListOfTablesToUse*
[WHERE *TupleConditions*]
[ORDER BY *ListOfOrderingAttributes*];

The SELECT statement: basics

Join

Join (no.1)

- Find the names of the suppliers that provide product P2

Supplier and part DB

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

Supplier and part DB

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

Cartesian product

- Find the names of the suppliers that provide product P2

```
SELECT SName
FROM S, SP ;
```

Cartesian product

S.SId	S.SName	S.#Empl	S.City	SP.SId	SP.PId	SP.Qty
S1	Smith	20	London	S1	P1	300
S1	Smith	20	London	S1	P2	200
S1	Smith	20	London	S1	P3	400
S1	Smith	20	London	S1	P4	200
S1	Smith	20	London	S1	P5	100
S1	Smith	20	London	S1	P6	100
S1	Smith	20	London	S2	P1	300
…	…	…	…	…	…	…
S2	Jones	10	Paris	S1	P1	300
…	…	…	…	…	…	…
S2	Jones	10	Paris	S2	P1	300
…	…	…	…	…	…	…

Join (no.1)

=

S.SId	S.SName	S.#Empl	S.City	SP.SId	SP.PId	SP.Qty
S1	Smith	20	London	S1	P1	300
S1	Smith	20	London	S1	P2	200
S1	Smith	20	London	S1	P3	400
S1	Smith	20	London	S1	P4	200
S1	Smith	20	London	S1	P5	100
S1	Smith	20	London	S1	P6	100
S1	Smith	20	London	S2	P1	300
…	…	…	…	…	…	…
S2	Jones	10	Paris	S1	P1	300
…	…	…	…	…	…	…
S2	Jones	10	Paris	S2	P1	300
…	…	…	…	…	…	…

Join (no.1)

- Find the names of the suppliers that provide product P2

```
SELECT SName
FROM S, SP
WHERE  S.SId = SP.SId
```

TableName.AttributeName

Join (no.1)

- Find the names of the suppliers that provide product P2

```
SELECT SName
FROM S, SP
WHERE   S.SId = SP.SId
```

Join condition

Join (no.1)

=

S.SId	S.SName	S.#Empl	S.City	SP.SId	SP.PId	SP.Qty
S1	Smith	20	London	S1	P1	300
S1	Smith	20	London	S1	P2	200
S1	Smith	20	London	S1	P3	400
S1	Smith	20	London	S1	P4	200
S1	Smith	20	London	S1	P5	100
S1	Smith	20	London	S1	P6	100
S1	Smith	20	London	S2	P1	300
…	…	…	…	…	…	…
S2	Jones	10	Paris	S1	P1	300
…	…	…	…	…	…	…
S2	Jones	10	Paris	S2	P1	300
…	…	…	…	…	…	…

Join (no.1)

S.SId	S.SName	S.#Empl	S.City	SP.SId	SP.PId	SP.Qty
S1	Smith	20	London	S1	P1	300
S1	Smith	20	London	S1	P2	200
S1	Smith	20	London	S1	P3	400
S1	Smith	20	London	S1	P4	200
S1	Smith	20	London	S1	P5	100
S1	Smith	20	London	S1	P6	100
S2	Jones	10	Paris	S2	P1	300
S2	Jones	10	Paris	S2	P2	400
S3	Blake	30	Paris	S3	P2	200
S4	Clark	20	London	S4	P3	200
S4	Clark	20	London	S4	P4	300
S4	Clark	20	London	S4	P5	400

Join (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';
```

Join (no.1)

SP.PId='P2'

S.SId	S.SName	S.#Empl	S.City	SP.SId	SP.PId	SP.Qty
S1	Smith	20	London	S1	P1	300
S1	Smith	20	London	S1	P2	200
S1	Smith	20	London	S1	P3	400
S1	Smith	20	London	S1	P4	200
S1	Smith	20	London	S1	P5	100
S1	Smith	20	London	S1	P6	100
S2	Jones	10	Paris	S2	P1	300
S2	Jones	10	Paris	S2	P2	400
S3	Blake	30	Paris	S3	P2	200
S4	Clark	20	London	S4	P3	200
S4	Clark	20	London	S4	P4	300
S4	Clark	20	London	S4	P5	400

Join (no.1)

S.SId	S.SName	S.#Empl	S.City	SP.SId	SP.PId	SP.Qty
S1	Smith	20	London	S1	P2	200
S2	Jones	10	Paris	S2	P2	400
S3	Blake	30	Paris	S3	P2	200

Join (no.1)

- Find the names of the suppliers that provide product P2

R

SName
Smith
Jones
Blake

Join (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';
```

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

- The result and the efficiency are independent of the order of predicates in the WHERE clause

Join (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';
```

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

- The result and the efficiency are independent of the order of tables in the FROM clause

Join (no.1)

- Declarativity of the SQL language
  - in relational algebra we define the order in which operators are applied
  - in SQL the best order is chosen by the optimizer independently of
    - the order of conditions in the WHERE clause
    - the order of tables in the FROM clause

Join (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 P.PId=SP.PId
           AND Color='Red';
```

- FROM clause with N tables
  - at least N-1 join conditions in the WHERE clause

Structure of the SELECT statement

SELECT [DISTINCT] *ListOfAttributesToDisplay*
FROM *ListOfTablesToUse*
[WHERE *TupleConditions*]
[GROUP BY *ListOfGroupingAttributes*]
[HAVING *AggregateConditions*]
[ORDER BY *ListOfOrderingAttributes*];