

SQL language: basics ➤ SQL Language >Language Instruction >Sample notation and database ➤ SELECT Statement ➤ Aggregate Functions ➤ Operator GROUP BY

0

## The SQL language

- A language for managing relational databases
  - Structured Query Language
- SQL provides commands to
  - define the schema of a relational database
  - read and write data
     define the schema of derived tables
  - · define user access privileges
- manage transactions
- The SQL language may be used in two ways
  - compiled
    - a host language encapsulates the SQL commands
    - SQL commands can be distinguished from the host language commands by means of appropriate syntactic mechanisms

1

The SQL language

• SQL is a set-level language

• SQL is a *declarative* language

 operators are applied to relations (tables) the result is always a relation (table)

• it describes what to do and not how to do it

• it has a higher level of abstraction compared to traditional programming

## **SQL** instructions The SQL language

The SQL language • Can be divided into • DML (Data Manipulation Language) • language for querying and updating the data • DDL (Data Definition Language) • language for defining the database structure

### Data Manipulation Language

- To query a database in order to extract data of interest • SELECT
- To modify a database instance
  - INSERT: insertion of new information into a table
  - UPDATE: update of the information in the database
  - DELETE: cancellazione di dati obsoleti

<u>⊅</u>β<sub>G</sub>

## Data Definition Language

- To define a database schema
  - creation, modification and deletion of tables: CREATE, ALTER, DROP TABLE
- To define derived tables
- creation, modification and deletion of tables whose content is obtained from other database tables: CREATE, ALTER, DROP VIEW
- To define complementary data structures for efficiently retrieving the data
  - creation and deletion of indices: CREATE, DROP INDEX
- To define user access privileges

Syntax of SQL commands

- · grant and revocation of privileges on resources: GRANT, REVOKE
- To define transactions
  - termination of a transaction: COMMIT, ROLLBACK

7

## Notation and example database

8

10

Notation
Ianguage keywords
upper case
variable terms
Grammar
angle brackets <>
to isolate a syntactic term
square brackets []
the enclosed term is optional
braces {}
the enclosed term may not appear or may be repeated an arbitrary number of items
vertical bar |

a term must be chosen among the options separated by the vertical bars

D 9

Example database: Supply-Product P1 Jumper P2 Jeans Red 40 48 London P3 Blouse Blue 48 Rome London 200 Skirt Blue 400 100 S2 P1 300 Smith London S2 S3 S4 P2 P2 400 S2 Jones S3 Blake 10 30 Paris 200 Paris S4 Clark S5 Adams Athens

• Supplier and part DB

• table P describes the available products

• primary key: Pld

• table S describes the suppliers

• primary key: Sld

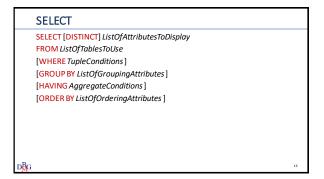
• table SP describes supplies, by relating each product to the suppliers that provide it

• primary key: (Sld, Pld)

• Pld: Foreign key (SP) REFERENCES Pld(P)

• Sid: Foreign key (SP) REFERENCES Sld(S)



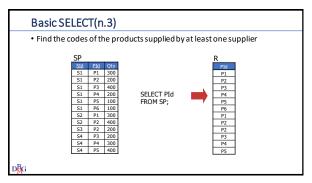


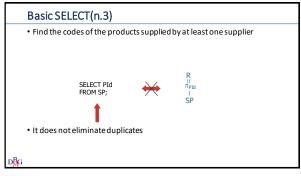
15

Basic SELECT(n.1)										
<ul> <li>Find the codes and the number of employees of the suppliers based in Paris</li> </ul>										
		SELECT SId, #Employees         R           FROM S         "Sid, #Empolyees           WHERE City='Paris';         "Copy-Paris'								
	S				Ś					
	SId	SName	#Employees	City	R					
	S1	Smith	20	London	SId #Employees					
	52	Jones	10	Paris	S2 10					
	53 S4	Blake	30	Paris	53 30					
	S4									
,	- 55	ricallis	30	Achella	<b>⊣</b>					
D <mark>B</mark> G					14					

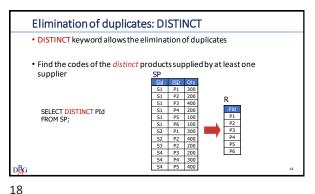
Basic SELECT(n.2) • Find the codes of all products in the database SELECT PId FROM P; P1 Jumper Red

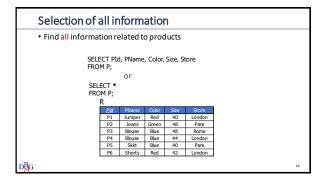
14

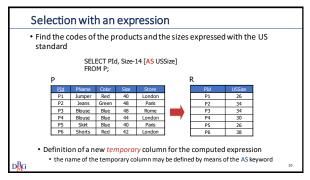


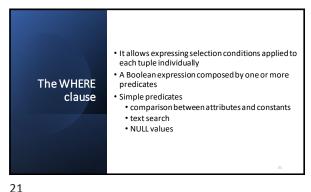


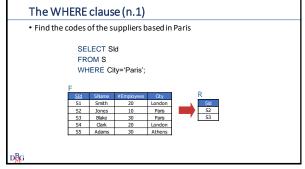
17 16

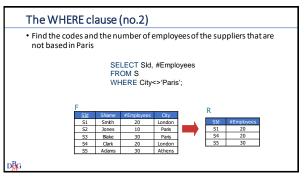


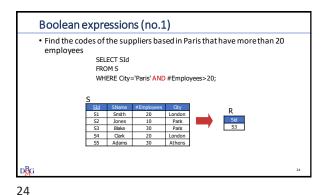


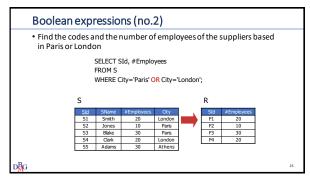


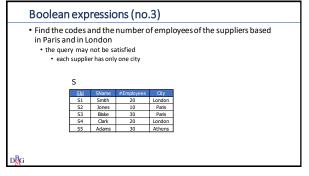


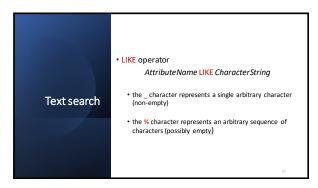












26 27

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%';										
	Р				-					
	PId P1	PName Jumper	Color	Size 40	Store London	R				
	P2	Jeans	Green	48	Paris	PId PName				
	P3	Blouse	Blue	48	Rome	P3 Blouse				
	P4	Blouse	Blue	44	London	P4 Blouse				
	P5	Skirt	Blue	40	Paris					
	P6	Shorts	Red	42	London					
D₿G						28				

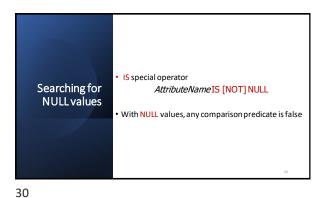
Text search (no.2)

• The Address attribute contains the string 'London'
Address LIKE '%London%'

• The supplier identification number is 3 and
• it is preceded by a single unknown character
• it is exactly 2 characters long
SId LIKE '\_3'

• The Store attribute does not have an 'e' in the second position
Store NOT LIKE '\_e%'

28 29



Managing NULL values

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

SELECT PId, PName
FROM P
WHERE Size>44;

P

Pi Junper Red Jole See Store
Pi Junper Red Jole London
Pi Junper Red Jole London
Pi Junper Red His Paris
Pi

31

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

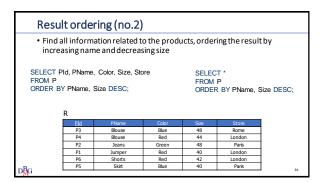
Pid	PName	Red	40	London		
P2	Jeans	Green	48	Paris		
P3	Bouse	Bue	48	Rome		
P4	Bouse	Bue	44	London		
P5	Skirt	Bue	NULL	Paris		
P6	Shorts	Red	42	London		
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red			
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red		
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	
P8	Red	Red	Red	Red	Red	Red
P8	Red					
P8	Red					
P8	Red					
P8	Red					

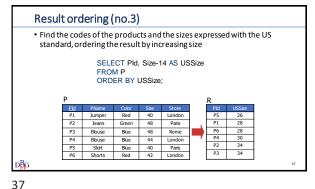
32 33

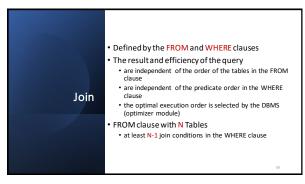
• ORDER BY clause
ORDER BY AttributeName [ASC | DESC]
{, AttributeName [ASC | DESC]}
• the default ordering is ascending
• if DESC is not specified
• the ordering attributes must appear in the SELECT clause
• even implicitly (as in SELECT \*)

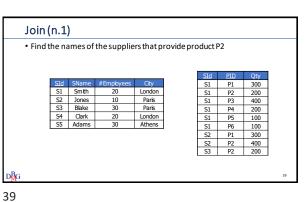
P P Red House Blue 49 London
P5 Skrt Blue 49 Pars
P6 Shorts Red 42 London

34 35







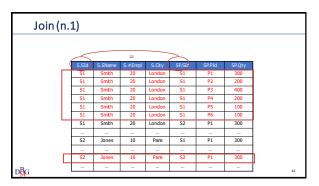


38

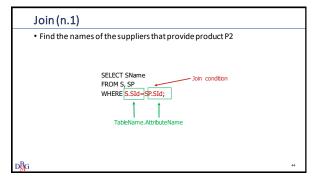
Cartesian product								
Find the names of the suppliers that provide product P2								
SELECT SName								
FROM S, SP;								
D <mark>∛</mark> G								

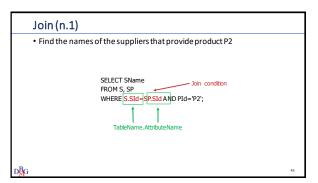
Cartesi	an pı	roduc	t					
	S.SId	S.SName	S.#Empl	S.City	SP.SId	SP.PId	SP.Qty	İ
	S1	Smith	20	London	S1	P1	300	
	S1	Smith	20	London	S1	P2	200	
	S1	Smith	20	London	S1	P3	400	
	S1	Smith	20	London	S1	P4	200	
	S1	Smith	20	London	S1	P5	100	
	S1	Smith	20	London	S1	P6	100	
	S1	Smith	20	London	52	P1	300	
	52	Jones	10	Paris	S1	P1	300	
	52	Jones	10	Paris	S2	P1	300	
₿G								4

40 41



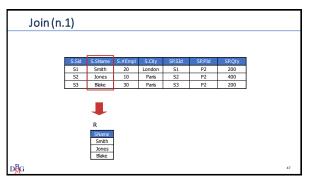
Join (	n.1)							
	_		=		~			
	S.SId	S.SName	S.#Empl	S.City	SP.SId	SP.PId	SP.Qty	
	SI	Smith	20	London	51	P1	300	
	S1	Smith	20	London	S1	P2	200	
	S1	Smith	20	London	S1	P3	400	
	S1	Smith	20	London	S1	P4	200	
	S1	Smith	20	London	S1	P5	100	
	S1	Smith	20	London	S1	P6	100	
	S2	Jones	10	Paris	52	P1	300	
	S2	Jones	10	Paris	52	P2	400	
	S3	Blake	30	Paris	S3	P2	200	
	S4	Clark	20	London	S4	P3	200	
	S4	Clark	20	London	S4	P4	300	
	54	Clark	20	London	54	P5	400	
BG								ı



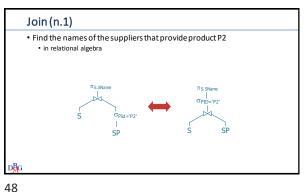


44 45

Join	(n.1)								
	_				_	SP.PId='P	2'		
	S.SId	S.SName	S.#Empl	S.City	SP.SId)	SP.PId	SP.Qty		
	SI	Smith	20	London	51	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	51	P5	100		
	S1	Smith	20	London	S1	P6	100		
	S2	Jones	10	Paris	52	P1	300		
[	S2	Jones	10	Paris	52	P2	400	1	
[	S3	Blake	30	Paris	53	P2	200		
	S4	Clark	20	London	54	P3	200		
	54	Clark	20	London	54	P4	300		
	S4	Clark	20	London	54	P5	400		
o <mark>B</mark> G								l	4



46 47



Join (n.1)  $\bullet \ \ \text{Find the names of the suppliers that provide product P2} \\$ · in relational algebra SELECT SName SELECT SName FROM S, SP FROM S,SP WHERE S.SId=SP.SId WHERE PId='P2' AND S.SId=SP.SId; AND PId='P2'; • The result and efficiency are independent • from the order of the predicates in the WHERE clause • from the order of the tables in the FROM clause

## **SQL** Declarability

- $\bullet$  In relational algebra (procedural language) we define the order in which the operators are applied
- In SQL (declarative language) the best order is chosen by the optimizer independently
  - from the order of the conditions in the WHERE clause
  - from the order of the tables in the FROM clause

50

### Join (n.2)

• Find the name of suppliers who provide at least one red product

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

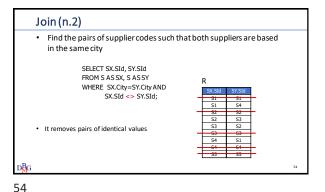
- FROM Clause with N Tables
  - at least N-1 join conditions in the WHERE clause

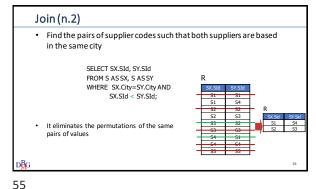
51

49

Join (n.2) • Find the pairs of supplier codes such that both suppliers are based in the same city SELECT SX.SId, SY.SId FROM S AS SX, S AS SY WHERE SX.City=SY.City; S AS SX Smith London Smith London

Join (n.2) Find the pairs of supplier codes such that both suppliers are based in the same city SELECT SX.SId, SY.SId FROM S AS SX, S AS SY WHERE SX.City=SY.City; The result includes pairs of identical values permutations of the same pairs of values





, · ·

# Different types of join may be specified outer join It allows differentiating between join conditions and tuple selection conditions SELECT [DISTINCT] Attributes FROM Table JoinType JOIN Table ON JoinCondition [WHERE TupleConditions]; JonType = < INNER [FULL | LEFT | RIGHT] OUTER >

Join: alternative syntax

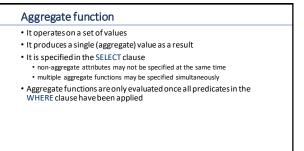
Find the names of the suppliers that supply at least one red product

SELECT SName
FROM P INNER JOIN SP ON PPId=SP.PId
INNER JOIN S ON S.SId=SP.SId
WHERE P.Color='Red';

56 57

OUTERjoin				
<ul> <li>Find the codes and the names of the sup the codes of the products they provide, a suppliers that are not supplying any products.</li> </ul>	also ir			th
	S.Sid	S.SName	SP.Sid	
	S1	Smith	P1	
	S1	Smith	P2	
	S1	Smith	P3	
SELECT S.SId, SName, PId	S1	Smith	P4	
FROM S LEFT OUTER JOIN SP ON	S1	Smith	P5	
FROM S LEFT OUTER JOIN SP ON	S1	Smith	P6	
S.SId=SP.SId;	S2	Jones	P1	
, i	S2	Jones	P2	
	S3	Blake	P2	
	S4	Clark	P3	
	S4	Clark	P4	
	S4	Clark	P5	
D <mark>B</mark> G	55	Adams	NULL	58





Aggregate functions

COUNT: count of elements in a given attribute

SUM: sum of values for a given attribute

AVG: average of values for a given attribute

MAX: maximum value of a given attribute

MIN: minimum value of a given attribute

61

63

60

COUNT

C

62

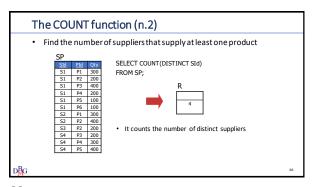
The COUNT function (n.2)

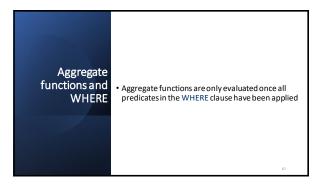
• Find the number of suppliers that supply at least one product

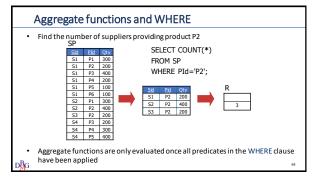
SP

SI
SI
PI
300
SI
PP
300
SI
PP
400
SI
SI
PP
400
SI
PP
300
SI
PP
400
SI
PP

64 65







SUM, MAX, MIN and AVG

• they allow an attribute or an expression as argument

SUM and AVG

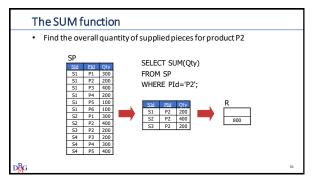
• they only allow numeric type or time interval attributes

• MAX and MIN

• they require an expression that can be ordered

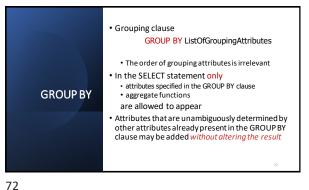
• may also be applied to character strings and time instants

68 69



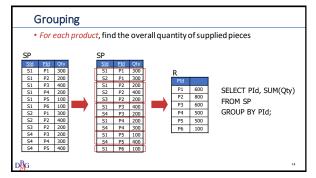


70 71



Grouping • For each product, find the overall quantity of supplied pieces S1 P1 300 S1 P2 200 S1 P3 400 S1 P4 200 S1 P5 100 S2 P1 300 S2 P2 400 S3 P2 200 S4 P3 200 S4 P4 300 S4 P4 300 S1 P3 400 P5 500 S4 P5 400

73



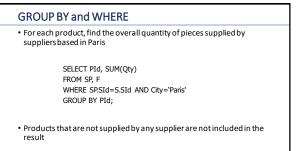
**GROUP BY and WHERE** • For each product, find the overall quantity of pieces supplied by suppliers based in Paris S3 P2 200

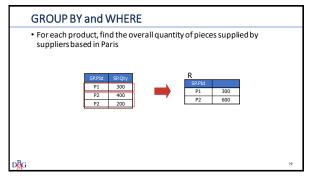
75 74

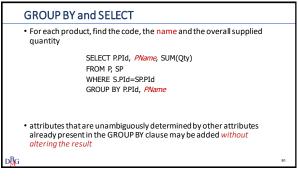
GROUP BY and WHERE	
<ul> <li>For each product, find the overall quantity of pieces supplied by suppliers based in Paris</li> </ul>	
SELECT FROM SP, S WHERE SP.SId=S.SId AND City='Paris'	
₽₿G	76

**GROUP BY and WHERE** • For each product, find the overall quantity of pieces supplied by suppliers based in Paris S1 Smith London 300 200 400 S1 Smith London S1 200 S1 Smith 20 London S1 100 *Blake* Clark Paris London *53* S4 *30* 20 *P2* S4 Clark 20 London S4 P4 P5 300 Clark

76 77







• You cannot use the WHERE clause to define selection conditions on groups

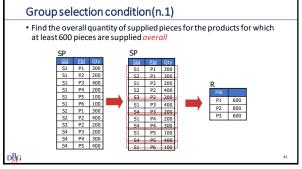
 • Selection condition on groups expressed in HAVING clause:

 HAVING Group Conditions

 • It is possible to specify conditions only on aggregated functions

81

80



Group selection condition (n.1)

 Find the overall quantity of supplied pieces for the products for which at least 600 pieces are supplied overall

 SELECT PId, SUM(Qty)
 FROM SP
 GROUP BY PId
 HAVING SUM(Qty)>=600;

 The HAVING clause allows the specification of conditions on the aggregate functions

82 83

