

SQL

Autor: Antonio Castañares

- Relational Database: is a collection of data stored in table
- Tables are composed by: rows and columns

PERSONS					
PERSON	NAME	BDATE	GENDER	COUNTRY	JOB
1	Einstein	1879-03-20	Male	Germany	S
2	Dickens	1812-07-22	Male	England	W
3	Dickinson	1830-07-15	Female	USA	W

- Each table is identify by a name
- Each column is identify by its name
- Each row is identify by its content

• SQL = structured query language, developed by IBM. It's consist of six types of statements

SQL STATEMENTS BY CATEGORY

CATEGORY	STATEMENT	PURPOSE
Query	SELECT	Display rows of one or more tables
Maintenance	INSERT	Add rows to a table
	UPDATE	Change rows in a table
	DELETE	Remove rows from a table
Definition	CREATE	Add tables, indices, views
	DROP	Remove tables, indices, views

QUERY STMT
SELECT columns
FROM table
WHERE comparisons

Maintenance Stmt
DELETE
FROM table
WHERE comparisons

DEFINITION STMT
CREATE
INDEX index
ON table (columns)

- SQL clause: Define conditions, normally one per line
- Key words: Reserve words
- Parameters: Variables of one SQL clause

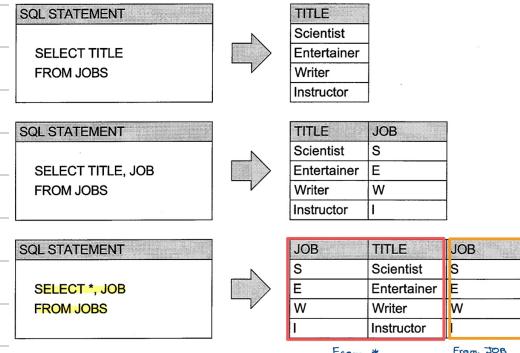


• Select all columns and rows **SELECT *** FROM Table

Means all columns

• Select specific columns **SELECT Column FROM Table** →

Column names are separated by commas



• Select specific rows We use comparisons using WHERE Column-name operator Value

VALUE SUMMARY

CATEGORY	DESCRIPTION	EXAMPLES
NUMERIC	positive values	3, +12
	negative values	-7, -1024000
	decimal values	3.141519, -.96
NON-NUMERIC	single words	'Chamberlin', 'SELECT'
	multiple words	'We love SQL', 'The LORD is good to me'
	single quotes	'10 O'Clock', 'I don't know'
DATE	'yyyy-mm-dd' format	'1996-01-01', '1996-12-31'

COMPARISON OPERATORS

OPERATOR	MEANING	EXAMPLE
=	Equal to	NAME = 'EINSTEIN'
<>	Not equal to	BDATE <> '1944-05-02'
<	Less than	POP < 100000
<=	Less than or equal to	NAME <= 'O'Grady'
>	Greater than	AREA > 999
>=	Greater than or equal to	BDATE >= '1962-06-19'

SQL STATEMENT		
SELECT NAME, BDATE FROM PERSONS WHERE BDATE < '1300-01-01'		

NAME	BDATE
Dante	1265-03-21
Shikabu	1000-09-03
Augustino	0354-04-30
Magnus	1193-12-05
Paul	0013-06-01

String

Summary of the last page

SYNTAX SUMMARY

CLAUSE	PARAMETERS	EXAMPLE
SELECT	*	*
	col list	LANGUAGE, COUNTRY
FROM	table	JOBs
WHERE	col oper value	AREA > 3000000

- Sorting rows (We must use ORDER BY (it must be the last clause))

Sorts the result by the second column ("area") (ascending)

SQL STATEMENT	
SELECT COUNTRY, AREA FROM COUNTRIES ORDER BY 2	

COUNTRY	AREA
Monaco	1
Vatican City	1
Nauru	8
Tuvalu	9
San Marino	24

Sort the results by "area" in descending order

SQL STATEMENT	
SELECT COUNTRY, AREA FROM COUNTRIES ORDER BY AREA DESC	

COUNTRY	AREA
Russia	6592800
Canada	3849674
China	3696100
USA	3679192
Brazil	3286500

Sort the column language alphabetically (ascending by default) and within each language group, sorts pop in descending order

SQL STATEMENT	
SELECT LANGUAGE, POP FROM COUNTRIES ORDER BY LANGUAGE, POP DESC	

LANGUAGE	POP
Afrikaans	1651545
Albanian	3413904
Amharic	55979018
Arabic	65359623
Arabic	30120420
Arabic	29168848
Arabic	28539321

Alphabetically (ascending)

Within each language, sorts pop (descending)

Summary of sorting rows

SYNTAX SUMMARY

CLAUSE	PARAMETERS	EXAMPLE
SELECT	*	*
	col list	LANGUAGE, COUNTRY
FROM	table	JOBs
WHERE	col oper value	AREA > 3000000
ORDER BY	col [DESC] list	LANGUAGE DESC, COUNTRY
	pos [DESC] list	1 DESC, 3

- Eliminating a duplicate row: We must use DISTINCT (is always written after SELECT)

SYNTAX SUMMARY

CLAUSE	PARAMETERS	EXAMPLE
SELECT	*	*
	[DISTINCT]	LANGUAGE, COUNTRY
FROM	table	JOBs
WHERE	col oper value	AREA > 3000000
ORDER BY	col [DESC] list	LANGUAGE DESC, COUNTRY
	pos [DESC] list	1 DESC, 3

SQL STATEMENT

SQL STATEMENT	
SELECT DISTINCT RELIGION FROM RELIGIONS WHERE PERCENT = 100 ORDER BY RELIGION	

RELIGION
Catholic
Eastern Or...
Lutheran
Muslim
Sunni Mus...

Advanced operators:

There are seven advanced operators

Like: Find values that match a pattern

And: Combine two or more comparisons (all comparisons must be true)

Between...and...: Compare each column value with a range of values

Or: Combine two or more comparisons (either or both must be true)

In: Compare the column value with a list (uses instead of or)

Is null: Looks rows with NULL values

Precedence and negation:

Not: negate or invert the result

() Parenthesis indicates precedence

SYNTAX SUMMARY

CLAUSE	PARAMETERS	EXAMPLE
SELECT	*	*
[DISTINCT]	col list	LANGUAGE, COUNTRY
FROM	table	JOBS
WHERE	col oper value	AREA > 3000000
[NOT]	col LIKE pattern	NAME LIKE '%ST__N%'
	cmpr AND cmpr	NAME LIKE '%ST__N%' AND JOB = 'S'
	col BETWEEN i AND j	LITERACY BETWEEN 55 AND 60
	cmpr OR cmpr	NAME = 'LUTHER' OR NAME = 'CALVIN'
	col IN (value list)	NAME IN ('POE', 'HUGO', 'DAHL')
	col IS NULL	POP IS NULL
(compound cmpr)	(JOB = 'S' OR JOB = 'W')	
ORDER BY	col [DESC] list	LANGUAGE DESC, COUNTRY
	pos [DESC] list	1 DESC, 3

Like: Match with a pattern

/ zero or more character
— represents one character

SQL STATEMENT

```
SELECT NAME, COUNTRY
FROM PERSONS
WHERE NAME LIKE '%Z%'
```

NAME	COUNTRY
Zola	France
Zimbalist	USA
Zwingli	Sweden

SQL STATEMENT

```
SELECT NAME, COUNTRY
FROM PERSONS
WHERE NAME LIKE 'EINST__N';
```

NAME	COUNTRY
Einstein	Germany

SQL STATEMENT

```
SELECT NAME, COUNTRY
FROM PERSONS
WHERE NAME LIKE '%ST__N%'
```

NAME	COUNTRY
Einstein	Germany
Springsteen	USA
Steinbeck	USA
Silverstein	USA

SQL STATEMENT

```
SELECT NAME, BDATE
FROM PERSONS
WHERE NAME = 'LUTHER'
OR NAME = 'CALVIN'
```

NAME	BDATE
Luther	1483-06-15
Calvin	1509-06-04

SQL STATEMENT

```
SELECT COUNTRY, LANGUAGE
FROM COUNTRIES
WHERE COUNTRY = 'GHANA'
OR COUNTRY = 'USA'
OR COUNTRY = 'FIJI'
```

COUNTRY	LANGUAGE
Fiji	English
USA	English
Ghana	English

Or: Alternative to or

• Introduce a list
• Elements are separated by commas

SQL STATEMENT

```
SELECT NAME, BDATE
FROM PERSONS
WHERE NAME = 'POE'
OR NAME = 'HUGO'
OR NAME = 'DAHL'
```

NAME	BDATE
Hugo	1802-08-05
Dahl	1916-09-01
Poe	1809-04-09

SQL STATEMENT

```
SELECT NAME, BDATE
FROM PERSONS
WHERE NAME IN ('POE', 'HUGO', 'DAHL')
```

NAME	BDATE
Hugo	1802-08-05
Dahl	1916-09-01
Poe	1809-04-09

And: Combine Comparisons

• Both must be true

SQL STATEMENT

```
SELECT NAME, BDATE
FROM PERSONS
WHERE NAME LIKE 'A%'  
AND BDATE >= '1900-01-01'
```

NAME	BDATE
Anne	1950-06-15
Albert II	1930-06-08
Achbe	1930-11-16
Archer	1947-06-25
Azimov	1920-08-22
Andrews	1935-10-01

Between...and...

• Compare in a range
• Range include both

SQL STATEMENT

```
SELECT COUNTRY, LITERACY
FROM COUNTRIES
WHERE LITERACY >= 55  
AND LITERACY <= 60
ORDER BY LITERACY
```

COUNTRY	LITERACY
Equatorial...	55
Guatemala	55
Congo	57
Algeria	57
Ghana	60
Iraq	60
Palau	60

Is null: Return rows with nulls

• Nulls are missing values
• Nulls are not blanks
• Nulls are not zeros

SQL STATEMENT

```
SELECT COUNTRY, POP
FROM COUNTRIES
WHERE POP IS NULL
```

COUNTRY	POP
Monaco	-
Andorra	-
Guatemala	-
Congo	-
Algeria	-
Ghana	-
Iraq	-
Palau	-

Precedence and negation:

• Parenthesis indicates precedence
• Not returns the inverse result

SQL STATEMENT

```
SELECT JOB, NAME
FROM PERSONS
WHERE COUNTRY = 'ITALY'  
AND (JOB = 'S' OR JOB = 'W')
ORDER BY JOB, NAME
```

JOB	NAME
S	Avogadro
S	Fermi
S	Galilei
W	Boccaccio
W	Dante
W	Petrarca

SQL STATEMENT

```
SELECT JOB, NAME
FROM PERSONS
WHERE COUNTRY = 'ITALY'  
AND NOT (JOB = 'S' OR JOB = 'W')
ORDER BY JOB, NAME
```

JOB	NAME
E	Fabio
M	Epiphani
M	Ptolemy
T	Augustino

Arithmetic expressions: Composed by

Operands They can be

Numeric values

Column names Their must be numeric or valid dates

Only addition and subtraction are valid with dates

ARITHMETIC OPERATORS		
OPERATOR	MEANING	EXAMPLE
+	Add	2 + 2
-	Subtract	BDATE - 365
*	Multiply	POP * 1.25
/	Divide	PERCENT / 100
()	Precedence	2 + (4 / 2)

These expressions products a new column as result

Arithmetic expressions can be located after SELECT, WHERE y ORDER BY

SQL STATEMENT		
SELECT COUNTRY, POP / AREA FROM COUNTRIES WHERE LANGUAGE = 'GERMAN' ORDER BY COUNTRY		

COUNTRY	POP/AREA
Austria	246.66
Germany	590.15
Liechtenst...	494.41
Switzerland	444.47

SQL STATEMENT		
SELECT COUNTRY, POP / AREA FROM COUNTRIES WHERE POP / AREA > 2000		

COUNTRY	POP/AREA
Maldives	2272.26
Malta	3029.58
Singapore	11702.29
Bahrain	2148.97
Bangladesh	2235.70

SQL STATEMENT		
SELECT COUNTRY, GNP * 1.1 FROM COUNTRIES WHERE LANGUAGE = 'GERMAN' ORDER BY COUNTRY		

COUNTRY	GNP*1.1
Austria	147840
Germany	1464100
Liechtenst...	693
Switzerland	164010

SQL STATEMENT		
SELECT COUNTRY, POP / AREA FROM COUNTRIES WHERE POP / AREA > 2000 ORDER BY POP / AREA DESC		

COUNTRY	POP/AREA
Singapore	11702.29
Malta	3029.58
Maldives	2272.26
Bangladesh	2235.70
Bahrain	2148.97

SYNTAX SUMMARY

CLAUSE	PARAMETERS	EXAMPLE
SELECT [DISTINCT]	*	*
	col list	LANGUAGE, COUNTRY
	expr list	POP / AREA
FROM	table	JOBES
WHERE	comparisons	AREA > 3000000
	expr value	POP / AREA > 300
ORDER BY	cal [DESC] list	LANGUAGE DESC, COUNTRY
	pos [DESC] list	1 DESC, 3
	expr [DESC] list	POP / AREA DESC

Column aliases: We can give a name to a column or arithmetic expression

SQL STATEMENT		
SELECT COUNTRY, POP / AREA FROM COUNTRIES WHERE POP / AREA > 2000 ORDER BY POP / AREA DESC		

COUNTRY	POP/AREA
Singapore	11702.29
Malta	3029.58
Maldives	2272.26
Bangladesh	2235.70
Bahrain	2148.97

In this case, density rename to the new column produced by the expression POP/Area

- Aliases can be in SELECT, WHERE or GROUP BY
- Normally they are after a column or arithmetic expression

SQL STATEMENT		
SELECT COUNTRY, POP / AREA AS DENSITY FROM COUNTRIES WHERE DENSITY > 2000 ORDER BY DENSITY DESC		

COUNTRY	DENSITY
Singapore	11702.29
Malta	3029.58
Maldives	2272.26
Bangladesh	2235.70
Bahrain	2148.97

- Statistical functions: Accept a parameter and return a summary value

Parameters
Expression
Column name

- Parameter must be enclosed in parenthesis

SYNTAX SUMMARY		
CLAUSE	PARAMETERS	EXAMPLE
SELECT	*	*
[DISTINCT]	col [AS alias] list expr [AS alias] list func [AS alias] list	LANGUAGE AS LANG, COUNTRY POP / AREA AS DENSITY MIN (POP) AS LOWEST
FROM	table	JOB
WHERE	comparisons	AREA > 3000000
ORDER BY	col [DESC] list pos [DESC] list expr [DESC] list	LANGUAGE DESC, COUNTRY 1 DESC, 3 POP / AREA DESC

Any type of value

Only numeric !!

FUNCTION	MEANING	EXAMPLE
COUNT ()	Count all rows	COUNT (*)
	Count non-null rows	COUNT (JOB)
	Count unique rows	COUNT (DISTINCT JOB)
SUM ()	Total value	SUM (POP)
MIN ()	Smallest value	MIN (POP)
MAX ()	Largest value	MAX (POP)
AVG ()	Average value	AVG (POP / AREA)

- Functions are defined in SELECT clause

- All results are in one row
- Different functions, different columns but in one row
- However, functions can be used to other clauses
- Having is similar to where, however having is executed after grouping

SQL STATEMENT

```
SELECT MIN ( POP ) AS LOWEST,
       MAX ( POP ) AS HIGHEST
  FROM COUNTRIES
 WHERE LANGUAGE = 'ENGLISH'
```

LOWEST	HIGHEST
16661	263814032

SQL STATEMENT

```
SELECT JOB, COUNT ( * ) AS TOTAL
  FROM PERSONS
 WHERE GENDER = 'MALE'
 GROUP BY JOB
 HAVING TOTAL > 30
 ORDER BY TOTAL DESC
```

JOB	TOTAL
E	113
W	99
M	36

SQL STATEMENT

```
SELECT JOB, COUNT ( * )
  FROM PERSONS
 WHERE GENDER = 'MALE'
 GROUP BY JOB
 ORDER BY COUNT ( * ) DESC
```

JOB	COUNT(*)
E	113
W	99
M	36
B	28
S	28
R	14
T	13

SQL STATEMENT

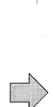
```
SELECT JOB, COUNT ( * ) AS TOTAL
  FROM PERSONS
 WHERE GENDER = 'MALE'
 GROUP BY JOB
 ORDER BY TOTAL DESC
```

JOB	TOTAL
E	113
W	99
M	36
B	28
S	28
R	14
T	13

- Grouping: Functions displays they results in one row, however if we want to get the total by category, we must use **Group By**

SQL STATEMENT

```
SELECT JOB,
       COUNT ( * ) AS TOTAL
  FROM PERSONS
 WHERE GENDER = 'MALE'
 GROUP BY JOB
 ORDER BY JOB
```



JOB	TOTAL
B	28
E	113
M	36
R	14
S	28
T	13
W	99

SQL STATEMENT

```
SELECT JOB, COUNTRY,
       COUNT ( * ) AS TOTAL
  FROM PERSONS
 WHERE GENDER = 'MALE'
 GROUP BY JOB, COUNTRY
 ORDER BY JOB, COUNTRY
```



JOB	COUNTRY	TOTAL
B	England	1
B	France	1
B	Germany	2
B	USA	24
E	Austria	1
E	Belgium	1
E	Canada	5
E	England	14

CLAUSE	PARAMETERS	EXAMPLE
SELECT	*	*
[DISTINCT]	col [AS alias] list expr [AS alias] list func [AS alias] list	LANGUAGE AS LANG, COUNTRY POP / AREA AS DENSITY MIN (POP) AS LOWEST
FROM	table	JOB
WHERE	comparisons	AREA > 3000000
GROUP BY	col list	JOB, COUNTRY
HAVING	comparisons with funcs	COUNT (*) > 30
ORDER BY	col [DESC] list pos [DESC] list expr [DESC] list func [DESC] list	LANGUAGE DESC, COUNTRY 1 DESC, 3 POP / AREA DESC COUNT (*) DESC

28

Display base on these categories

- Joints: Combine rows from two or more tables based on a related column, often a key like "id". They use column names with format tablename.columnname

Type	Image	Description	Example
Inner join		Returns rows where there is a match in both tables	<pre>SELECT name, department FROM employees INNER JOIN Departments ON Employees.department_id = Departments.department_id</pre>
Left join			

