

SQL

- Funzioni
 - Riga
 - Aggregate
 - group by – having
- Subquery
- Progetto di riferimento
 - <https://github.com/egalli64/mpjp> mySql (*modulo 3*)

Funzioni su riga singola

- Operano su e ritornano una singola riga
 - Caratteri e stringhe
 - Numeri
 - Date
 - Espressioni regolari
 - Conversione: **CAST()**
 - `select cast(12345.67 as char), cast('2019-05-01' as date);`

Alcune funzioni su stringhe

- **ASCII()**: codice ASCII di un carattere, **CONVERT()** + **CHAR()**: da codice ASCII a carattere
`select ascii('A') as A, convert(char(90) using utf8) as '90';`
- **CONCAT()**: concatenazione di stringhe
`select concat(first_name, ' ', last_name) from employees;`
- **UPPER()**: tutto maiuscolo, **LOWER()**: tutto minuscolo
`select upper('upper') up, lower('LOWER') low;`
- **POSITION()**, **LOCATE()**: sub, target [, start] → [1..n], 0 not found
`select position('ba' in 'crab') as "not found", position('ra' in 'crab') as pos;`
`select locate('ab', 'crab abba rabid cab', 13) as pos;`
- **LENGTH()**: per string e numeri, convertiti implicitamente in stringhe
`select length('name'), length(42000);`

Alcune funzioni su stringhe /2

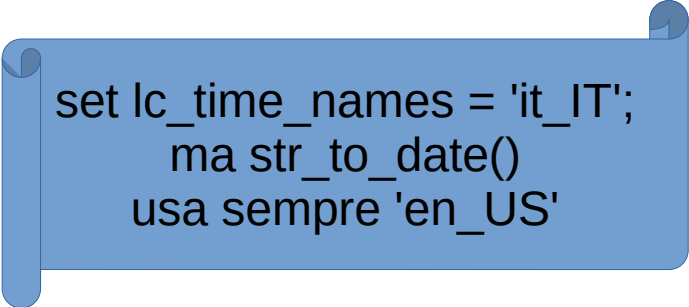
- **LPAD()**, **RPAD()**: padding. Stringa → dimensione, con eventuale pad specificato
`select lpad('tom', 30, '.') tom, rpad('tim', 30, '_- _') tim;`
- **LTRIM()**, **RTRIM()**, **TRIM()**: rimozione di caratteri dall'input
`select ltrim(' Hi!') "left", concat('[', rtrim('Hi! '), ']') "right", concat('[', trim(' Hi! '), ']') "both";`
`select trim(leading 'xy' from 'xy!xy') "left", trim(trailing 'xy' from 'xy!xy') "right", trim(both 'xy' from 'xy!xy') "both";`
- **RIGHT()**: estrae da una stringa n caratteri a destra
`select right('discardedXYZ', 3);`
- **REPLACE()**: sostituzione di substring, **SUBSTR()**: estrazione di substring
`select replace('Begin here', 'Begin', 'End'), substr('ABCDEFGFG', 3, 4);`

Alcune funzioni numeriche

- **ABS()**: valore assoluto
- **CEIL()**: 'soffitto', **FLOOR()**: 'pavimento'
- **MOD()**: modulo, resto di divisione intera
- **POWER()**: potenza; **EXP()**: e^x ; **SQRT()**: radice 2; **LN()**, **LOG()**: logaritmi
- **ROUND()**, **TRUNCATE()**: arrotonda/tronca a decimali (-) o potenze di 10 (-)
- **SIGN()**: -1, 0, 1 per numeri negativi, zero, positivi
- **PI()**: pi greco
- **SIN()**, **COS()**, **TAN()**,...: funzioni trigonometriche

Alcune funzioni su date

- `CURDATE()`, `NOW()`: data, data e time corrente
- `DAYNAME()`, `MONTHNAME()`: nome del giorno o del mese
- `DATE_FORMAT()`, `STR_TO_DATE()`: conversione tra data e stringa
- `DATE_ADD`(date, INTERVAL expr unit), `DATE_SUB()`: data +/- intervallo
 `date_add(curdate(), interval 1 day)`
- `EXTRACT`(unit FROM date): estrae parte della data(-time)
 `select extract(year from now());`
- `DATEDIFF()`: giorni di distanza tra due date(-time)
- `LAST_DAY`(date): ultimo giorno del mese

A blue rectangular sticky note with rounded corners and a small tab on the right side. It contains text about locale settings for date functions.

```
set lc_time_names = 'it_IT';  
ma str_to_date()  
usa sempre 'en_US'
```

Espressioni regolari

- **REGEXP_LIKE()** versione estesa di LIKE
 - Es: cognomi che iniziano per A o E:
select last_name
from employees
where regexp_like(last_name, '^[ae].*');

Altre funzioni

- **VERSION()**
 - Versione di MySQL in esecuzione
- **USER()**
 - Utente connesso
- **SCHEMA()**
 - Lo schema corrente

Esercizi

- Employees
 - Qual è il salario corrente, quale sarebbe con un incremento dell'8.5%, qual è il delta come valore assoluto
 - Quanti giorni sono passati dall'assunzione a oggi
 - Quant'è la commissione di ognuno o 'no value'

Funzioni aggregate

- Ignorano i NULL
- Uso di DISTINCT per filtrare duplicati
- **AVG()**: media
- **COUNT()**: numero di righe
- **MAX()**: valore massimo
- **MIN()**: minimo
- **SUM()**: somma
- **STDDEV()**: deviazione standard
- **VARIANCE()**: varianza

Raggruppamento via GROUP BY

- Divide il risultato della select in gruppi
- È possibile applicare funzioni aggregate sui gruppi
select department_id, truncate(avg(salary), 0)
from employees
group by department_id
order by 1;

GROUP BY – HAVING

- HAVING filtra i risultati di GROUP BY
- È possibile filtrare prima le righe della SELECT con WHERE, e poi il risultato della GROUP BY con HAVING

```
select manager_id, round(avg(salary))
```

```
from employees
```

```
where salary < 8000
```

```
group by manager_id
```

```
having avg(salary) > 6000
```

```
order by 2 desc;
```

Subquery

- In WHERE:

```
select first_name, last_name from employees
where employee_id = (select manager_id from employees where last_name = 'Chen');
```

- In FROM (inline view):

```
select max(e.salary)
from (select employee_id, salary from employees where employee_id between 112 and 115) e;
```

- In HAVING:

```
select department_id, round(avg(salary)) from employees group by department_id
having avg(salary) < (select max(x.sal) from
(select avg(salary) sal from employees group by department_id) x)
order by 2 desc;
```

JOIN con subquery

- Subquery genera una tabella temporanea → join
select region_name, c.country_count
from regions natural join (
select region_id, count(*) country_count
from countries
group by region_id) c;

subquery multirighe in WHERE

- Uso dell'operatore IN

es: nome di EMPLOYEES che sono manager

```
select first_name, last_name from employees
```

```
where employee_id in (
```

```
    select distinct manager_id
```

```
    from employees where manager_id is not null)
```

```
order by 2;
```

Esercizi

- Employees
 - Salary: maggiore, minore, somma, media
 - Come sopra, ma per ogni job_id
 - Quanti dipendenti per ogni job_id
 - Quanti sono gli IT_PROG
 - Quanti sono i manager
 - Nome dei dipendenti che non sono manager
 - Qual è la differenza tra il salario maggiore e il minore
 - Come sopra, ma per ogni job_id, non considerando dove non c'è differenza
 - Qual è il salario minimo con i dipendenti raggruppati per manager, non considerare chi non ha manager, né i gruppi con salario minimo inferiore a 6.000€

Esercizi /2

- Indirizzi completi, tra locations e countries
- Employees
 - Nome di tutti i dipendenti e nome del loro department
 - Come sopra, ma solo per chi è basato a Toronto
 - Chi è stato assunto dopo David Lee
 - Chi è stato assunto prima del proprio manager
 - Chi ha lo stesso manager di Lisa Ozer
 - Chi lavora in un department in cui c'è almeno un employee con una 'u' nel cognome
 - Chi lavora nel department Shipping
 - Chi ha come manager Steven King