

SQL

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

Funzioni su riga singola

- Operano su e ritornano una singola riga
 - Caratteri e stringhe
 - Numeri
 - Date
 - Espressioni regolari
 - Conversione

Alcune funzioni su stringhe

- **ASCII()**: codice ASCII di un carattere, **CHR()**: da codice ASCII a carattere
 - select ascii('A') as A, chr(90) as "90" from dual;
- **CONCAT()**: concatenazione di stringhe, cfr. operatore ||
 - select concat(first_name, last_name) from coders;
- **INITCAP()**: iniziali maiuscole, **UPPER()**: tutto maiuscolo, **LOWER()**: tutto minuscolo
 - select initcap('a new thing') as initcap, lower('NEW') low, upper('old') up from dual;
- **INSTR()**: x, target, start, occurrence → [1..n], 0 not found
 - select instr('crab', 'ba') as "not found", instr('crab abba rabid cab', 'ab', 4, 3) as pos from dual;
- **LENGTH()**: conversione implicita a stringa
 - select length('name'), length(42000) from dual;

Alcune funzioni su stringhe / null

- **LPAD()**, **RPAD()**: padding. Stringa → dimensione, con eventuale pad
 - `select lpad('tom', 30, '.') tom, rpad('tim', 30, '_- _-') tim from dual;`
- **LTRIM()**, **RTRIM()**, **TRIM()**: rimozione di caratteri dall'input
 - `select ltrim(' Hi!'), rtrim('Hi!abab', 'ab'), trim('0' from '00Hi!000') from dual;`
- **REPLACE()**: sostituzione di substring, **SUBSTR()**: estrazione di substring
 - `select replace('Begin here', 'Begin', 'End'), substr('ABCDEF G',3,4) from dual;`
- **NVL()**: null to value, se null → secondo parametro
 - `select employee_id, nvl(commission_pct, 0) from employees;`
- **NVL2()**: se non è null → secondo parametro, altrimenti il terzo
 - `select employee_id, nvl2(commission_pct, 'value', 'no value') from employees;`

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()**, **TRUNC()**: arrotonda/tronca a decimali (+) o potenze di 10 (-)
- **SIGN()**: -1, 0, 1 per numeri negativi, zero, positivi
- **SIN()**, **COS()**, **TAN()**,...: funzioni trigonometriche

Alcune funzioni su date

- **ADD_MONTHS()**: aggiunge mesi alla data
- **MONTHS_BETWEEN()**: mesi tra le due date
- **NEXT_DAY()**: giorno della settimana successivo al corrente
- **LAST_DAY()**: ultimo giorno del mese
- **ROUND()**, **TRUNC()**: arrotonda/tronca il giorno

```
select add_months(sysdate, 3), months_between(sysdate, '01-FEB-2020'),  
       last_day(sysdate), next_day(sysdate, 'lun'),  
       round(sysdate, 'year'), round(sysdate, 'month'),  
       trunc(sysdate, 'year'), trunc(sysdate, 'month')  
from dual;
```



dipendono dal locale corrente

Espressioni regolari

- **REGEXP_LIKE()** versione estesa di LIKE
 - Inizio stringa: ^
 - Fine stringa: \$
 - Un carattere un insieme: [...]
 - Un carattere qualunque: .
 - Zero, uno, o più elementi precedentemente definiti: *
 - ...
- Es: cognomi che iniziano per A o E:
`select last_name from employees where regexp_like(last_name, '^[AE].*');`

Alcune funzioni di conversione

- **TO_CHAR()**, **TO_NUMBER()**, **TO_DATE()**
 - convertono (formattando) nel tipo compatibile specificato
- **CAST()**
 - forma generica, il tipo è specificato via “**AS**”

```
select to_char(12345.67), to_char(12345.67, '99,999.999'),  
       to_char(12345.67, '9.9'), to_char(2020, 'RN'),  
       to_number('970,13') * 2, to_date('2020-06-05', 'YYYY-MM-DD'),  
       cast('05-APR-20' as date) + 2, cast(12345.678 as number(10,2))  
from dual;
```


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
- **MEDIAN()**: mediana

Raggruppamento via GROUP BY

- Divide il risultato della select in gruppi
- È possibile applicare funzioni aggregate sui gruppi
select department_id, round(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(salary)
from (select employee_id, salary from employees where employee_id between 112 and 115);
```

- In HAVING:

```
select department_id, round(avg(salary)) from employees group by department_id
having avg(salary) < (select max(sal) from
(select avg(salary) sal from employees group by department_id))
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