

**Basi di Dati**  
**Prova di SQL del 23-03-2023**  
**soluzioni**

COGNOME	NOME	WORKSTATION

**Salvare la(le) query sulla workstation, con suffisso .sql (il compito non verrà valutato se il file non è in formato .sql)**

Facendo riferimento allo schema del database PROVA contenente le tabelle DIPART (dipartimenti), DIPENDENTI, FORNITORI, PARTI, PRODOTTI, FORNITURE scrivere la query SQL corrispondenti alle seguenti interrogazioni:

1) Ideare una query che verifichi se il vincolo di integrità referenziale fra SPED\_DETTAGLI e FORNITURE è effettivamente rispettato. La query deve elencare le tuple che non rispettano il vincolo; se tutte le tuple rispettano il vincolo, la query deve mettere in output la stringa "Vincolo di integrità rispettato". Sull'istanza PROVA del server questa query deve confermare che il vincolo è rispettato.

```
select Fps.*, "Vincolo violato"
from
(select fcod, prcod, pcod
from forniture
group by fcod, prcod, pcod) as FPF right join
(select fcod, prcod, count(*) as NCS
from sped_dettagli
group by fcod, prcod) as FPS
on FPF.fcod=FPS.fcod and FPF.prcod=FPS.prcod
where FPF.fcod is null
```

2) Si considerino le città presenti nello schema del database in tutte le cinque table DIPART, FORNITORI, PRODOTTI, PARTI, SPEDIZIONI. Produrre un tabella che elenchi le città, il numero di volte in cui ciascuna di esse è presente in totale, e la percentuale di presenza in ciascuna delle table. Un esempio fittizio di risultato è elencato qui sotto. La tabella deve essere ordinata per valori decrescenti del TOTALE.

Ex: ( dati di fantasia)

CITTA	TOTALE	NFOR%	NPAR%	NPRO%	NSPED%
'ROMA'	100	70%	15%	10%	5%
'MANTOVA'	90	85%	5%	10%	0%
'FIRENZE'	80	35%	35%	15%	15%
'MILANO'	45	0%	80%	10%	10%
'TORINO'	45	33%	33%	33%	1%
'ANCONA'	35	40%	60%	0%	0%
'CATANIA'	5	55%	35%	10%	0%

Commento: il blocco A costruisce tutte le CITTA senza duplicazioni

Ciascuno dei blocchi B, C, D, E, F aggrega le città di una table, con relativo conteggio

La query fa il LEFT JOIN fra A e ciascuno dei blocchi

La target list elenca tutte le città (da A) e fa i conteggi (TOTALE e percentuali)

```
select A.CITTA ,
coalesce(NP,0)+coalesce(NF,0)+coalesce(NPR,0)+coalesce(NDEST,0)+coalesce(NPART,0) as TOTALE,
coalesce(NP,0)/(coalesce(NP,0)+coalesce(NF,0)+coalesce(NPR,0)+coalesce(NDEST,0)+coalesce(NPART,0))*100 as NPAR,
coalesce(NF,0)/(coalesce(NP,0)+coalesce(NF,0)+coalesce(NPR,0)+coalesce(NDEST,0)+coalesce(NPART,0))*100 as NFOR,
coalesce(NPR,0)/(coalesce(NP,0)+coalesce(NF,0)+coalesce(NPR,0)+coalesce(NDEST,0)+coalesce(NPART,0))*100 as NPRO,
(coalesce(NDEST,0)+coalesce(NPART,0))/(coalesce(NP,0)+coalesce(NF,0)+coalesce(NPR,0)+coalesce(NDEST,0)+coalesce(NPART,0))*100 as NSPE
from
```

```
( select citta
from fornitori
where citta is not null
union
select citta
from parti
where citta is not null
union
select citta
from prodotti
where citta is not null
union
select cittadest as citta
from spedizioni
where cittadest is not null
union
select cittapart as citta
from spedizioni
where cittapart is not null) A
```

left join

```
(select citta, count(*) as NP
from parti
where citta is not null
group by citta) C
on A.citta=C.CITTA
```

left join

```
(select citta, count(*) as NF
from fornitori
where citta is not null
group by citta) B
on A.citta=B.CITTA
```

left join

```
(select citta, count(*) as NPR
from prodotti
where citta is not null
group by citta) D
on A.citta=D.CITTA
```

left join

```
(select cittadest as citta, count(*) as NDEST
from spedizioni
where cittadest is not null
group by cittadest) E
on A.citta=E.CITTA
```

left join

```
(select cittapart as citta, count(*) as Npart
from spedizioni
where cittapart is not null
group by cittapart) F
on A.citta=F.CITTA
```

order by TOTALE desc