Contents

Per ciascuna categoria

- Codice
- Descrizione
- Costo complessivo status=3

Costo complessivo?

```
select c.cin as codice, c.description as descr, sum(p.cost) as sum_body_once
from (((order o natural join orderstatus os)
join product p on o.product = p.pin)
join category c on c.cin = p.category)
```

```
where os.status = 3;
group by c.cin, c.description

select i.nome as nome
from impiegato i
where i.stipendio > 40;

select os_prod.non_lo_so as ma_che_cazzo
from (select os.status as status, o.product as product
from (orderstatus os join order o on os.oin=o.oin)) as os_prod
```

1 Schemi tabelle

- impiegato : matricola, nome, dipartimento, stipendio
- dipartimento : codice, nome, direttore
- progetto : codice, nome, budget, scadenza, responsabile
- pp : impiegato progetto

1.1 Quarta query

1.1.1 sottotabella responsabili

```
(impiegato i join progetto p on (p.responsabile = i.matricola)) as responsabile
```

1.1.2 query queryosa

```
select ip.matricola
from
(progetto p join impiegato i on (i.matricola = p.responsabile)) as responsabile,
(impiegato i_primo join pp on (pp.impiegato = i_primo.matricola)) as ip
where
(ip.progetto = responsabile.codice) and
(ip.stipendio > responsabile.stipendio);
```

1.2 Quinta query

facciamo il prodotto cartesiano di progetti con progetti, ma solo su progetti diversi tra loro con lo stesso budget

```
select p1.codice, p1.responsabile
from progetto p1, progetto p2;
```

Lode al Vignoz e ai join impliciti cartesiani

```
select p1.codice, p1.responsabile
from progetto p1, progetto p2
where
(p1.codice <> p2.codice) and
(p1.budget = p2.budget);
```

1.3 Sesta query

1.3.1 Così il Vignoz

1.3.2 È contento

```
select alleluja.matricola
from
(((impiegato i join pp on (i.matricola = pp.impiegato)) as ip
join progetto on (ip.progetto = progetto.codice)) as ip_man
join dipartimento d on (ip_man.dipartimento = d.codice)) as alleluja
where alleluja.responsabile = alleluja.direttore;
```

1.4 Settima query

non l'ho fatta

1.5 Ottava query

matricola di ciascun responsabile e il numero di progetti che dirige

```
from progetto p
group by responsabile -- vogliamo vedere per responsabile
```

che selezioniamo?

```
select responsabile, count(codice)
from progetto p
group by responsabile;
```

non possono esserci valori nulli su quell'attributo, inoltre tutti i valori devono essere diversi in quanto `codice` è una chiave

quindi non c'è differenza con

```
select responsabile, count(*)
from progetto p
group by responsabile;
```

(il voto del compitino massimo è 28)

1.6 Undicesima

salto temporale e cardinale

Per ciascun dipartiemnto, il nomde, il numero dei suoi affiliati, ed il numero dei suoi affiliati che sono responsabili di progetto

il numero degli affiliati del dipartimento, esplicitata solo nella tabella impiegato

e il numero di questa che risultano essere responsabili di progetto, relazione esplicitata solo nella clausola progetto

quindi il from sarà

```
from dipartiemnto d, impiegato i, progetto p
```

adesso mettaimo le condizioni del join

```
from dipartiemnto d, impiegato i, progetto p
where
```

1.7 Ok torna alla 9

nome, cognome di ciascun impiegato e il numero di progetto a cui partecipa

```
from impiegato i, pp
where pp.impiegato = i.matricola
```

ora

```
select i.nome, i.cognome, count(progetto)
from impiegato i, pp
where pp.impiegato = i.matricola
```

siccome ho chiesto nome e cog devo farci group by

```
select i.nome, i.cognome, count(progetto)
from impiegato i, pp
where pp.impiegato = i.matricola
group by pp.progetto, i.nome, i.cognome;
```

gli attributi della target list devono essere unici all'interno di ogni gruppo, quindi o ci faccio il group by, o ci faccio a fanculo.

la roba nella target list

1.7.1 nota

```
select i.nome, i.cognome, count(progetto)
from impiegato i, pp
where pp.impiegato = i.matricola
group by i.matricola;
```

funziona lo stesso, ma gli attributi nella target list non sono nel group by e la cosa fa schifo, capisco che gli attributi devono essere unici e quella è una chiave, quindi unica, ma la cosa fa schifo.

1.8 Decima

matricola, nome e cognome degli impiegati che sono resposabili di almeno due progetti

serve la tabella impiegati visto che il collegamento matricola, nome, cognome solo li

serve la tabella progetto

serve un count

```
select nome, count(progetto)
from impiegato i, progetto p
where responsabile = matricola
```

ora vogliamo avere gruppi con lo stesso impiegato, quindi group by avrà la matricola (e nome e cognome)

```
select i.matricola, i.nome, i.cognome
from impiegato i, progetto p
where responsabile = matricola
group by i.matricola, i.nome, i.cognome
having count(p.codice) >= 2;
```

Gino Rampollo

è sbagliato mettere nella `where` cazzi di operatori aggregati, nella clausola `on` siamo nella condizione di un hwere

nella `having` sono solo cazzi di operatori aggregati, non puoi metterci cazzi di attributi singoli

1.9 Ritorno alla 11

Per ciascun dipartimento

- Nome
- Numero dei suoi affiliati che sono responsabili di progetto

il legame tra impiegato e dipartimento affiliato è eplicitato nella relazione progetto il nome del dipartimento è esplicitato nella tabella progetto i responsabili di progetto sono esplicitatai nella tabella progetto

quindi

```
from dipratimento d,, impiegato i, progetto p
```

ora famo

```
from dipratimento d,, impiegato i, progetto p
where i.dipartimento = d.codice and i.matricola = p.responsabile
```

aggiungiamo la select, e la group by necessaria

```
select d.codice, d.nome
from dipratimento d,, impiegato i, progetto p
where i.dipartimento = d.codice and i.matricola = p.responsabile
group by d.codice, d.nome
```

in prima battuta si direbbe

```
select d.codice, d.nome, count(matricola)
from dipratimento d,, impiegato i, progetto p
where i.dipartimento = d.codice and i.matricola = p.responsabile
group by d.codice, d.nome
```

fa differenza

- `count(*)`
- `count(matricola)`
- `count distinct (matricola)`

?

Matricola è una chiave primaria, non può avere valori nulli, quindi

- `count(*)`
- `count(matricola)`

sono identici

per sapere se `count(distinct matricola)` può avere risultati diversi. possono essrci due righe con lo stesso valore di matricola qui?

con la costituzione della tabella che ho fatto nella from e where, possono esserci più valori con la stessa riga di matricola?

sì, quando uno stesso responsabile è responsabile di più progetto, stesso valore in impiegato, stesso valore di matricola, diverso valore di progetto per cui quello stesso impiegato è responsabile.

in questo caso che diff. c'è tra `count` e `count distinct`?

se faccio `count` sto contando i progetti se faccio `count distinct` sto contando le matricole

se scrivo count sto contando il numero di progetti se scrivo count distinct sto contanto il numero di impiegati che sono responsabili di progetto, se uno è responsabile di 3 progetti non conta 3, conta la presona, non i progetti

la `select` chiede il numero di affiliati che sono responsabili di progetto, non chiede il numero di progetti i cui responsabili sono affiliati a quel dipartimento, quindi qui si fa `count distinct`

```
select d.codice, d.nome, count(distinct matricola)
from dipartimento d, impiegato i, progetto p
where i.dipartimento = d.codice and i.matricola = p.responsabile
group by d.codice, d.nome;
```

1.10 la 12

selezionare per ciascun dipartiemnto il nome del dipartimento ed il numero dei suoi affiliati che hanno cognome casini

cognome casini non è una cosa aggregata, non è un count o company, quindi nel where

```
select d.codice, d.nome, count(matricola)
from dipartimento d, impiegato i
where i.dipartimento = d.codice and i.cognome = 'Casini'
group by d.codice, d.nome;
```

in questa tabella possono esserci righe divese con lo stesso valore di matricola, con questa join che ha fatti in realtà per ogni riga di impiegato, se vi

ricordata che il join che segue l'integrità referenziale ha la cardinalità della tabella interna, in questo caso l'interna è l'impiegato, quindi qui che scriva

- `count(*)`
- `count(matricola)`
- `count(distinct matricola)`

  non fa differenza
  usare `like` non faceva differenza

```
select d.codice, d.nome, count(matricola)
from dipartimento d, impiegato i
where i.dipartimento = d.codice and i.cognome like 'Casini'
group by d.codice, d.nome;
```

questo join è un join non completo, se io avessi voluto che nel join fossero presenti tutti i dipartimenti, eventualmente con il conteggio 0 sui dipartimenti che non hanno nessun casino, che avrei dovuto fare?

1.10.1 join esterno

devo usare quindi la sintassi in cui il join è esplicitato nella clausola from
voglio che diparimento partecipi del tutto

```
from dipartimento d left join impiegato i
on i.dipartimento = d.codice and cognome = 'Casini'
```

la completa sarà

```
select d.codice, d.nome, count(matricola)
from dipartimento d left join impiegato i
on i.dipartimento = d.codice and cognome = 'Casini'
group by d.codice, d.nome;
```

```
select d.codice, d.nome, count(distinct matricola)
from dipartimento d left join impiegato i
on i.dipartimento = d.codice and cognome like 'Casini'
group by d.codice, d.nome;
```

non ci sto capendo molto, non mi sono svegliato e non mi sveglierò fino alla fine dei tempi intempati

1.10.2 una cosa a cui fare attenzione

```
select d.codice, d.nome, count(distinct matricola)
from dipartimento d left join impiegato i
on i.dipartimento = d.codice
where cognome like 'Casini'
group by d.codice, d.nome;
```

il risultato non è lo stesso, abbimo una perdita di righe

le condizioni del join possono essere esplicitate o nella clausola on, o nella clausola where

ma facendo così eseguo le due selezioni in due passi diversi, in questa query così com'è prima il dbms fa il join esterno, poi fa il where, nel join esterno compaiono anche i dipartimenti non incasinati, ora la clausola where trova il congome nullo per coloro che non hanno casini ed ecco che ti toglie le righe senza casini.

se volgio che effettivamente il join esterno mantenga effettivamente anche le righe nulle allora questa condizione del cognome questa condizione devo imporla dove compare il join eserno.

qui count matricola viene 0 e non null visto che il count conta i valori non nulli, se tutto nullo hai uno 0, alleluja

1.11 Numero ignoto

matricola, nome, cognome degli impiegati che non partecipano a nessun progetto

manetendo nel risultato anche gli impiegati che non partecipano a nessun progetto

```
from impiegato i join pp on i.matricola = pp.impiegato
```

un impiegato che non partecipa ad alcun progetto non compare in questo join, per metterci tutti gli impiegati, eventualmente con valori nulli in `impiegato.progetto`

quindi join esterno sinistro

```
from impiegato i left join pp on i.matricola = pp.impiegato
```

```
select *
from impiegato i left join pp on i.matricola = pp.impiegato;
```

con questa query puoi vedere che Giuliano casini è uno sfaticato

```
from impiegato i right join pp on i.matricola = pp.impiegato;
```

non avrebbe un senso, visto che pp dipende già da impiegato, dovrei mettere tutte le coppie di impiegato progetto, anche quelle senza impiegato, ma l'integrità referenziale dice che stigrancazzi.

```
select i.matricola, i.nome, i.cognome
from impiegato i left join pp on i.matricola = pp.impiegato
group by i.matricola, i.nome, i.cognome
having count(progetto)=0;
```

con questa query puoi vedere in modo ancora più chiaro che Giuliano Casini è uno sfaticato del cazzo

ha un conto di progetti non nulli pari a 0 ha solo progetti nulli non ha progetti non ha motivo di esistere è debole deve morire

si poteva fare anche con `progetto is null`

sql vuole mantenere una corrsipondenza con come esprimeresti la cosa in un linguaggio naturale, per design del linguaggio.

```
select i.matricola, i.nome, i.cognome
from impiegato i left join pp on i.matricola = pp.impiegato
where pp.progetto is null;
```

con questa query continuiamo a vedere che Giuliano Casini continua a essere uno sfaticato di merda

```
select i.matricola, i.nome, i.cognome
from impiegato i left join pp on i.matricola = pp.impiegato
group by i.matricola, i.nome, i.cognome
having count(progetto)=0;
```

va bene lo stesso dato che è nulla non viene contata, dato che non viene contata il conteggio è pari a 0.

2 Ultima query, arrivo dei nidi

```
select codice, d.nome, count(matricola) as numero
from impiegato i right join dipartimento on dipartimento=codice
group by codice, d.nome
having max(count(*))
```

eh, voleeevi!

```
select codice, d.nome, count(matricola) as numero
from impiegato i right join dipartimento on dipartimento=codice
group by codice, d.nome
order by numero desc
limit 1
```

ti da quello in cima, ma *voleeevi, eh?*, se ci sono più dipartimenti con quel massimo non funziona più

2.1 Nidi!

2.2 Nidi!

2.3 Nidi!

2.4 Nidi!

2.5 Nidi!

la `select` ritorna una tabella puoi mettere quella tabella in un from gloria!, gloria! gloria!, gloria!