

ESERCIZI SQL

Esercizi GROUP BY

- 1) Trovare quante visite ogni paziente ha effettuato nel 2023

```
SELECT CF_PAZIENTE, COUNT(*)
```

```
FROM visite
```

```
WHERE YEAR(DATA)=2023
```

```
GROUP BY CF_PAZIENTE;
```

- 2) Trovare la temperatura max, il peso max e l'altezza max di ogni paziente → considero tabella visite

```
SELECT CF_PAZIENTE, MAX(TEMPERATURA), MAX(ALTEZZA), MAX(PESO)
```

```
FROM visite
```

```
GROUP BY CF_PAZIENTE;
```

- 3) Trovare quante visite sono state fatte nei vari anni a partire dal 2010

```
SELECT YEAR(DATA) AS ANNO, COUNT(*) AS CONTEGGIO_VISITE
```

```
FROM visite
```

```
WHERE YEAR(DATA)>2010
```

```
GROUP BY YEAR(DATA);
```

- 4) Trovare quanti giorni ogni paziente ha trascorso da ricoverato in ospedale

```
SELECT CODICE_FISC, SUM(DATEDIFF(DATA_USCITA, DATA_INGRESSO))
```

```
FROM ricoveri
```

```
GROUP BY(CODICE_FISC);
```

- 5) Trovare quanti sono stati i ricoveri nei vari anni e nei vari reparti

```
SELECT YEAR(DATA_INGRESSO), REPARTO, COUNT(*) AS NUMERO_RICOVERI
```

```
FROM ricoveri
```

```
GROUP BY YEAR(DATA_INGRESSO), REPARTO;
```

- 6) Trovare quanti sono stati tra il 2015 e il 2020 i pazienti dimessi con i vari DRG e in quanti anni (raggruppiamo anche per anno)

```
SELECT YEAR(DATA_INGRESSO), DRG COUNT(*) AS CONTEGGI
```

```
FROM ricoveri
```

```
WHERE YEAR(DATA_INGRESSO) BETWEEN 2015 AND 2020
```

```
GROUP BY YEAR(DATA_INGRESSO), DRG
```

Esercizi JOIN

- 1) Trovare per ogni paziente la data delle sue visite e il peso misurato e successivamente aggiungere la clausola che il peso debba essere maggiore di 50

```
SELECT g.codice_fisc, g.nome, g.cognome, v.data, v.peso
```

```
FROM generale g, visite v
```

```
WHERE g.codice_fisc=v.cf_paziente
```

```
AND peso>50
```

- 2) Trovare per i vari pazienti quando sono stati ricoverati e in che reparto; poi ordino per cognome, nome e data ingresso in ordine decrescente per uno stesso paziente

CF	NOME	COGNOME	DATA_RICOVERO	REPARTO
CF1	MARIO	ROSSI	1-1-2000	MEDICINA
CF1	MARIO	ROSSI	1-3-2010	ORTOPEDIA
CF2	LUIGI	VERDI	3-5-2020	PEDIATRIA

```
SELECT g.codice_fisc AS CF, g.nome, g.cognome, ric.data_ingresso, rep.nome
```

```
FROM generale g, ricoveri ric, reparti rep
```

```
WHERE g.codice_fisc=ric.codice_fisc
```

```
AND ric.reparto=rep.codice
```

```
ORDER BY g.cognome, g.nome, ric.data_ingresso desc
```

- 3) Trovare i farmaci prescritti dai diversi medici nell'ultimo anno e ordiniamo per cognome, nome e descrizione per vedere se ci sono medici che prescrivono troppo uno stesso farmaco→questa query serve per individuare i "super-prescrittori" (medici che prescrivono troppo perché hanno accordi con case farmaceutiche)

Tabelle che dobbiamo utilizzare per questa query:

PRESCRIZIONE	VISITE	GENERALE	FARMACI
--------------	--------	----------	---------

Risultato della query:

CF_MED	NOME	COGNOME	DATA_PRESC	FARMACO
--------	------	---------	------------	---------

```
SELECT g.codice_fisc, g.nome, g.cognome, v.data, f.descrizione
```

```
FROM generale g, visite v, prescrizione p, farmaci f
```

```
WHERE g.codice_fisc=v.cf_med
```

```
AND v.id_visita=p.id_visita
```

```
AND p.cod_farmaco=f.codice
```

```
AND year(v.data)=year(curdate())
```

```
ORDER BY g.cognome, g.nome, f.descrizione
```

Se adesso vogliamo contare quante prescrizioni un medico ha fatto: (aggiungo le descrizioni se voglio sapere anche quali farmaci)

```
SELECT g.codice_fisc, g.nome, g.cognome, f.descrizione  
count(*) AS numero_prescrizioni  
FROM generale g, visite v, prescrizione p, farmaci f  
WHERE g.codice_fisc=v.cf_med  
AND v.id_visita=p.id_visita  
AND p.cod_farmaco=f.codice  
AND year(v.data)=year(curdate())  
GROUP BY g.codice_fisc, g.cognome, g.nome, f.descrizione
```

SUBQUERY

- 1) Elencare le persone che sono state ricoverate almeno una volta

```
SELECT DISTINCT g.codice_fisc, g.cognome, g.nome  
FROM generale g, ricoveri r  
WHERE g.codice_fisc=r.codice_fisc
```

- 2) Elencare le persone che non hanno avuto nessun ricovero

```
SELECT g.codice_fisc, g.cognome, g.nome  
FROM generale g  
WHERE g.codice_fisc not in
```

(SELECT r.codice_fisc FROM ricoveri r) → subquery

Se tolgo la not ottengo lo stesso risultato della query precedente, ovvero le persone che sono state ricoverate almeno una volta

Quando lo stesso risultato può essere ottenuto da una join o da una subquery, USARE LA JOIN

NOT EXISTS = è un operatore logico che restituisce VERO se la SELECT successiva restituisce l'insieme vuoto, mentre restituisce FALSO se la SELECT successiva restituisce almeno una riga (EXISTS fa il contrario)

```
SELECT g.codice_fisc, g.nome, g.cognome  
FROM generale g  
WHERE NOT EXISTS (SELECT * FROM ricoveri r WHERE r.codice_fisc=g.codice_fisc)
```

- 1) Trovare le femmine che non hanno mai subito interventi chirurgici

```
SELECT g.codice_fisc, g.nome, g.cognome  
FROM generale g  
WHERE NOT EXISTS (SELECT * FROM interventi WHERE i.codice_fisc=g.codice_fisc)
```

AND upper(g.sesso)="F"

2) Trovare i farmaci che non sono mai stati prescritti

```
SELECT f.codice, f.descrizione
```

```
FROM farmaci f
```

```
WHERE f.codice NOT IN (SELECT p.cod_farmaco FROM prescrizione p)
```

3) Trovare i farmaci che non sono mai stati prescritti nell'ultimo anno

```
SELECT f.codice, f.descrizione
```

```
FROM farmaci f
```

```
WHERE f.codice NOT IN (SELECT p.cod_farmaco FROM prescrizione p, visite v WHERE  
p.id_visita=v.id_visita AND DATEDIFF(v.data, curdate())<365)
```

Facciamo lo stesso es con la NOT EXISTS

```
SELECT f.codice, f.descrizione
```

```
FROM farmaci f WHERE NOT EXISTS (SELECT * FROM prescrizione p, visite v
```

```
WHERE f.codice=p.cod_farmaco AND p.id_visita=v.id_visita AND DATEDIFF(v.data, CURDATE())<365)
```

Quando la SUBQUERY utilizza nelle sue condizioni degli attributi che sono anche nella query principale si dice che la subquery è "correlata" alla query principale

4) Trovare i pazienti per i quali non è mai stato possibile calcolare il BMI

CF_PZ	PESO	ALTEZZA
CF1	-	160
CF1	50	-
CF1	-	-
CF1	55	162
CF2	80	-
CF2	-	152

I pazienti per cui è stato possibile sono quelli che hanno almeno una riga con entrambe le misure

I pazienti per cui non è mai stato possibile sono quelli che non hanno alcuna riga con entrambe le misure

```
SELECT g.codice_fisc, g.nome, g.cognome
```

```
FROM generale g
```

```
WHERE g.codice_fisc NOT IN (SELECT v.cf_paziente FROM visite v
```

```
WHERE v.altezza IS NOT NULL AND v.peso IS NOT NULL)
```

```
AND g.codice_fisc=v.cf_paziente
```

Se non mettiamo visite in JOIN nella query principale, la query ci darebbe anche tutte le persone che sono nella generale ma non sono mai state visitate

5) Trovare la persona più pesante

CF	PESO
CF1	80
CF1	82
CF2	50
CF3	82

```
SELECT DISTINCT V.CF_PAZIENTE, G.NOME, G.COGNOME
FROM VISITE V, GENERALE G
WHERE V.PESO IN (SELECT MAX(PESO) FROM VISITE V)
AND V.CF_PAZIENTE=G.CODICE_FISC
```

6) Elencare sulla stessa riga i pazienti visitati e il medico che li ha visitati (voglio ricevere una tabella con CF_PZ, NOME_PZ, COGNOME_PZ, DATA_VIS, CF_MED, NOME_MED, COGNOME_MED)

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME, V.DATA, V.CF_MED, GM.NOME, GM.COGNOME
FROM generale G, visite V, generale GM
WHERE (G.CODICE_FISC=V.CF_PAZIENTE) AND (GM.CODICE_FISC=V.CF_MED)
```

7) Trovare i pazienti che sono stati ricoverati a distanza di pochi giorni dalla precedente dimissione (≤ 3 gg).

```
SELECT DISTINCT G.CODICE_FISC,G.NOME,G.COGNOME
FROM ricoveri R, ricoveri R1, generale G
WHERE R.CODICE_FISC=G.CODICE_FISC AND R1.CODICE_FISC=G.CODICE_FISC
AND datediff(R1.DATA_USCITA,R.DATA_USCITA)BETWEEN 1 AND 3 → per escludere i ricovedi dello stesso
giorno, andrebbe bene anche imporre che gli ID_VISITE siano diversi
```

INDICE DI RISCHIO (IR) - ($0 < IR < 7$)

- Maschio +1
- Vecchio (>65) +1
- Iiperteso (sistolica >150 oppure diastolica >90) +1,5
- Obeso (BMI >30) +2,5

1) Calcolare IR per tutte le visite effettuate (! Gli operatori di confronto restituiscono 1 se VERO, 0 se FALSO).

```
SELECT G.CODICE_FISC,G.NOME,G.COGNOME,V.DATA, (lower(G.SESSO)='m')+
(datediff(V.DATA,G.NASCITA)/365.25)>65)+
1.5*(V.P_SISTOLICA>140 OR V.P_DIASTOLICA>90)+
2.5*(V.PESO/((V.ALTEZZA*V.ALTEZZA)>90) AS IR
FROM generale G, visite V
```

WHERE G.CODICE_FISC=V.CF_PAZIENTE

- 2) Trovare i pazienti che hanno compiuto gli anni da ricoverati.

Funzioni che agiscono su stringhe

- 1) Scrivere nome e cognome dei pazienti con solo l'iniziale maiuscola

```
SELECT CONCAT(UPPER(LEFT(NOME,1)),LOWER(RIGHT(NOME,LENGTH(NOME)-1)))  
AS NOME,  
CONCAT(UPPER(LEFT(COGNOME,1)),LOWER(RIGHT(COGNOME,LENGTH(COGNOME)-1)))  
AS COGNOME  
FROM generale
```

- 2) Trovare le persone che hanno trascorso il Natale in ospedale

```
SELECT CODICE_FISC  
FROM ricoveri  
WHERE YEAR(DATA_USCITA+6)>YEAR(DATA_INGRESSO+6)
```

Create view

```
CREATE VIEW ELENCO_PERSONE AS  
SELECT CODICE_FISC,NOME,COGNOME  
FROM generale
```

- 3) Cercare i pazienti che svolgono o hanno svolto la professione di medico

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME  
FROM esercita_prof E, generale G
```

Esercizi sulle percentuali

```
WHERE E.CODICE_FISC=G.CODICE_FISC AND LEFT(E.CODICE_PROF,5)='2.4.1'
```

- 1) Trovare la percentuale dei dipendenti attuali che lavorano in ciascun reparto

Creo prima una view in cui calcolo il totale dei dipendenti:

```
CREATE VIEW Totale_di_dipendenti AS  
SELECT COUNT(*) FROM dipendenti  
WHERE data_dimissione IS NULL
```

Utilizzare gli alias con le
create view

Creo prima una view con il numero di dipendenti per reparto:

```
CREATE VIEW Num_dip_per_reparto AS  
SELECT reparto, COUNT(*) AS num_dip FROM dipendenti  
WHERE DATA_DIMISSIONE IS NULL  
GROUP BY reparto
```

Ora calcolo la percentuale

```
SELECT 100*(numdip/num_tot_dip) AS PERCENTUALE, reparto AS REPARTO
FROM TOTDIP, Num_dip_per_reparto AS ND, reparti AS REP
WHERE REP.CODICE=ND.REPARTO
```

- 2) Costo ricovero per singolo ricovero e per tutti... preso un paziente a piacere CF1 controllare se i costi dei suoi ricoveri sono stati coperti o meno dai relativi DRG.

Vista per farmaci, per prestazione, per costo reparto, query che somma tutti i costi, query per vedere se DRG>costi

```
CREATE VIEW COSTOREPARTO ASb
```

```
SELECT SUM(DATEDIFF(R.DATA_USCITA,R.DATA_INGRESSO)*REP.COSTO_GIORNO) AS costorep
FROM ricoveri R, reparti REP
WHERE R.REPARTO=REP.CODICE AND R.CODICE_FISC='CF1'
```

```
SELECT sum(P.TARIFFA) AS COSTO PRESTAZIONI
from ricoveri R, visite V, richiestaprestazioni RP, prestazioni P
where R.CODICE FISC="CF1" and
R.CODICE FISC-V.CF PAZIENTE and V.DATA between R.DATA INGRESSO and R.DATA USCITA
AND RP.ID_VISITA=V.ID_VISITA AND RP.CodPrest=P.CODICE
```

```
SELECT SUM(F.prezzo_unita_dose S.DoseSomministrata) AS COSTO_FARMACI FROM ricoveri R,
somministrazione S, farmaci F, visite V, prescrizione P WHERE R.CODICE FISC="CF1" AND S.DataOra
BETWEEN R.DATA_INGRESSO AND R.DATA_USCITA AND S.ID PRESC=P.ID PRESC AND P.ID VISITA=V.ID
VISITA AND V.CF_PAZIENTE=R.CODICE_FISC AND P.COD_FARMACO=F.codice
```

```
SELECT coalesce(COSTO_RICOVERI,0)+ coalesce(COSTO PRESTAZIONI,0)+ coalesce(COSTO FARMACI,0) AS
COSTO TOTALE_CF1 FROM COSTORICOVERI, COSTOPRESTAZIONI, COSTOFARMACI
```

COALESCE: restituisce il primo elemento non NULL, se sono tutti NULL restituisce NULL

```
SELECT sum(TARIFFA_O) as RIMBORSO_DRG from ricoveri R, drg D where R.CODICE_FISC="CF1" and
R.DRG=D.CODICE
```

```
SELECT (R.RIMBORSO_DRG-C.costo_totale_cf1) as GUADAGNO FROM Costototale CF1 C, RIMBORSIDRG R
```

Funzione per convertire stringa in valore numerico

- 1) SELECT CODICE, CONVERT(CODICE, DOUBLE) AS VAL1, CAST(CODICE AS DOUBLE) AS VAL2
FROM farmaci

2) SELECT FROM generale

where left(CODICE_FISC,2)="CF"

order by CONVERT(RIGHT(CODICE FISC, length(CODICE FISC)-2),DOUBLE)

3) Trovare quante prescrizioni di farmaci sono state effettuate nel reparto di pediatria sopra i 20 euro

SELECT COUNT(*) as NUM_FARMACI_COSTOSI

FROM RICOVERI R, VISITE V, FARMACI F, PRESCRIZIONE P, REPARTI REP

WHERE F.CODICE=P.COD_FARMACO AND

R.CODICE FISC=V.CF PAZIENTE AND

V.DATA BETWEEN R.DATA_INGRESSO AND R.DATA_USCITA

AND P.ID_VISITA=V.ID_VISITA AND

R.REPARTO=REP.CODICE

AND UPPER(REP.NOME)="PEDIATRIA"

AND F.PREZZO>20

- 1) Trovare la percentuale di pazienti che hanno/hanno avuto un tumore rispetto al totale dei pazienti che hanno/hanno esercitato un certa professione

Lavoratori “artigiani e operai dell’industria estrattiva...”: ce ne sono 30 nel database. Fra questi, quanti hanno avuto nella vita un tumore? 5. La percentuale è $100 \cdot 5/30$. Vogliamo questa percentuale per tutte le professioni.

```
CREAT VIEW TOTALEPROF AS
SELECT CODICE_PROF, COUNT(*) S TOTALE_PROF
FROM ESERCITA_PROF
GROUP BY CODICE_PROF
CREAT VIEW CONTUMOREPROF AS
SELECT EP.CODICE_PROF, COUNT(*) AS CONTUMORE_PROF
FROM ESERCITA_PROF EP, ANAMNESI A
WHERE EP.CODICE_FISC=A.CODICE_FISC
AND
CAST(LEFT(A.MALATTIA,3)) BETWEEN “140” AND “239”
GROUP BY EP.CODICE_PROF
SELECT T.DESCRIZIONE, 100*C.CONTUMORE_PROF/T.TOTALE_PROF AS
PERC_CON_TUMORE
FROM TOTALEPROF T, CONTUMOREPROF C, PROFESSIONI P
WHERE C.CODICE_PROF=T.CODICE_PROF AND P.CODICE=T.CODICE_PROF
```

- 2) Trovare i dipendenti che negli ultimi due anni non hanno mai lavorato né a Natale né a capodanno (la notte di San Silvestro)

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME FROM dipendenti D, generale G
WHERE D.CODICE_FISC=G.CODICE_FISC AND D.CODICE_FISC NOT IN
(SELECT CF_MED FROM visite
WHERE YEAR(DATA)> YEAR(CURDATE())-2 AND MONTH(DATA)=12 AND DAY(DATA) IN (25,31)) AND
D.CODICE_FISC NOT IN
(SELECT CF_MEDICO FROM ricoveri
WHERE YEAR(DATA_INGRESSO)>YEAR(CURDATE())-2 AND MONTH(DATA_INGRESSO)=12 AND
DAY(DATA_INGRESSO) IN (25,31)) AND
(SELECT CF_INFERMIERE FROM somministrazione WHERE YEAR(DATAORA) > YEAR(CURDATE())-2 AND
MONTH(DATAORA)=12 AND DAY(DATAORA) IN (25,31)) AND
(SELECT CF_CHIRURGO FROM interventi WHERE YEAR(DATA_INTERVENTO)>YEAR(CURDATE())-2 AND
MONTH (DATA_INTERVENTO)=12 AND DAY(DATA_INTERVENTO) IN (25,31))
```

- 3) Trovare l’ultima volta che ogni dipendente ha lavorato a Natale o Capodanno

Dovrebbe uscire così:

CF	NOME	COGNOME	DATA
CF2	ANDREA	FRANCA	31/12/2019
CF5	PAOLO	MARIA	25/12/202
CF78	LUIGI	ROSSI	31/12/2018

```
SELECT S.CF_INFERMIERE AS CF, MAX(S.DATAORA) AS DATAORA
FROM SOMMINISTRAZIONI S
WHERE MONTH(S.DATAORA)=12 AND DAYS(S.DATAORA) IN (25,31)
GROUP BY S.CF_INFERMIERE
UNION
SELECT R.CF_MEDICO, MAX(DATA_INGRESSO)
FROM RICOVERI R
WHERE MONTH(S.DATA_INGRESSO)=12 AND DAY(R.DATA_INGRESSO) IN (25,31)
GROUP BY R.CF_MEDICO //idem per visite e interventi chirurgici
```

- 4) Trovare i pazienti che sono sempre stati ricoverati nello stesso reparto (ovviamente per chi ha fatto più di un ricovero)

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME, REP.NOME AS REP, COUNT(*) AS NUM_RIC
FROM generale G, ricoveri R, reparti REP
WHERE G.CODICE_FISC=R.CODICE_FISC AND REP.CODICE=R.REPARTO
GROUP BY G.CODICE_FISC, G.NOME, G.COGNOME
HAVING COUNT(*)>1 AND MIN(R.REPARTO)=MAX(R.REPARTO)
```

- 5) Trovare i pazienti che sono stati ricoverati in almeno due reparti diversi

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME, REP.NOME AS REP, COUNT(*) AS NUM_RIC
FROM generale G, ricoveri R, reparti REP
WHERE G.CODICE_FISC=R.CODICE_FISC AND REP.CODICE=R.REPARTO
GROUP BY G.CODICE_FISC, G.NOME, G.COGNOME
```

HAVING COUNT(*)>1 AND MIN(R.REPARTO)<MAX(R.REPARTO)

- 6) Trovare i pazienti che sono stati ricoverati nel corso di un anno sempre nello stesso reparto

```
SELECT
    G.CODICE_FISC,
    G.NOME,
    G.COGNOME,
    YEAR(R.DATA_INGRESSO),
    MIN(REP.NOME) AS REPARTO,
    COUNT(*) AS NUM_RIC
FROM
    generale G,
    ricoveri R,
    reparti REP
WHERE
    G.CODICE_FISC = R.CODICE_FISC
    AND REP.CODICE = R.REPARTO
GROUP BY G.CODICE_FISC , G.NOME , G.COGNOME , YEAR(R.DATA_INGRESSO)
HAVING COUNT(*) > 1
AND MIN(R.REPARTO) = MAX(R.REPARTO)
```

- 7) Trovare i pazienti che sono stati ricoverati 3 volte in 3 diversi reparti nel corso di un anno

```
SELECT
    G.CODICE_FISC, G.NOME, G.COGNOME
FROM
    generale G,
    ricoveri R,
    reparti REP
WHERE
    G.CODICE_FISC = R.CODICE_FISC
    AND REP.CODICE = R.REPARTO
GROUP BY G.CODICE_FISC, YEAR(DATA_INGRESSO)
[HAVING COUNT(*) = 3]
```

AND AVG(R.REPARTO) != (MIN(R.REPARTO) + MIN(R.REPARTO) + MAX(R.REPARTO)) / 3

AND AVG(R.REPARTO) != (MIN(R.REPARTO) + MAX(R.REPARTO) + MAX(R.REPARTO)) / 3]

HAVING COUNT(DISTINCT(*))=3

- 8) Si vogliono recuperare le generalità di un paziente sapendo dolo che era maschio, nato prima del 1950, era stato ricoverato una sola volta da un medico che di cognome faceva rossi e aveva festeggiato il compleanno durante quel ricovero

SELECT

G.CODICE_FISC, G.NOME, G.COGNOME

FROM

generale G,

ricoveri R

WHERE

R.CODICE_FISC = G.CODICE_FISC

AND YEAR(G.NASCITA) < 1950

AND LOWER(G.SESSO) = 'm'

AND TRUNCATE((DATEDIFF(R.DATA_INGRESSO, NASCITA) / 365.25),0) <
TRUNCATE((DATEDIFF(DATA_USCITA, NASCITA) / 365.25),0)

AND R.CF_MEDICO IN (SELECT

G2.CODICE_FISC

FROM

generale G2

WHERE

LOWER(G2.COGNOME) = 'rossi')

GROUP BY G.CODICE_FISC , G.NOME , G.COGNOME

HAVING COUNT(*) = 1

- 9) Si vogliono elencare i medici ordinandoli secondo il carico di lavoro (numero di ricoveri + visite effettuate)

CREATE VIEW CARICOLAVORO AS

SELECT R.CF_MEDICO AS CF, COUNT(*) AS NUMERO

FROM ricoveri R

GROUP BY R.CF_MEDICO

UNION

SELECT V.CF_MED, COUNT(*)

FROM visite V

GROUP BY V.CF_MED;

SELECT G.CODICE_FISC, G.NOME, G.COGNOME, SUM(NUMERO) AS CARICO

FROM CARICOLAVORO C, generale G

WHERE C.CF=G.CODICE_FISC

GROUP BY G.CODICE_FISC, G.COGNOME, G.NOME ORDER BY 4

- 10) Trovare anno per anno a partire da 10 anni fa il numero di ricoveri effettuati dai vari reparti separando fra maschi e femmine e indicando la degenza media

ANNO	NOME REPARTO	SESSO	DEGENZA MEDIA	NUMERO RICOVERI
2013	Pediatria	M	20	305
2013	Pediatria	F	15	400
2013	Medicina	M	4	1000
2013	Medicina	F	7	1005

SELECT YEAR(R.DATA_INGRESSO) AS ANNO, REP.NOME, UPPER(G.SESSO) AS SESSO,

AVG(DATEDIFF(R.DATA_USCITA,R.DATA_INGRESSO)) AS DEGENZA_MEDIA,

COUNT(*) AS RICOVERI

FROM generale G, ricoveri R, reparti REP

WHERE R.CODICE_FISC=G.CODICE_FISC AND DATEDIFF(CURDATE(),R.DATA_INGRESSO)/365.25<10

AND REP.CODICE=R.REPARTO

GROUP BY YEAR(R.DATA_INGRESSO), REP.NOME, G.SESSO

ORDER BY YEAR(R.DATA_INGRESSO)

- 11) Trovare il dipendente che ha avuto più periodi lavorativi presso l'ospedale, si calcoli anche la durata media dei suoi periodi di lavoro già giunti al termine (ce ne possono essere di più con il valore massimo)

DIPENDENTI	MEDIA PERIODI LAVORATIVI
CF1	10 gg
CF4	13 gg

CREATE VIEW PERIODI_A AS

SELECT CODICE_FISC,COUNT(*) AS NUMERO_PERIODI,

AVG(DATEDIFF(DATA_DIMISSIONE,DATA_ASSUNZIONE)) AS DURATA_MEDIA

FROM dipendenti

GROUP BY CODICE_FISC;

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME, DURATA_MEDIA  
FROM generale G, PERIODI_A P  
WHERE G.CODICE_FISC=P.CODICE_FISC AND P.NUMERO_PERIODI=(SELECT MAX(P1.NUMERO_PERIODI)  
FROM PERIODI_A P1)
```

- 1) Trovare i pazienti che sono ingrassati troppo durante le feste di Natale

Consideriamo solo pazienti che sono stati dentro almeno una settimana tra il periodo del 24 dicembre e il 6 gennaio. Ipotesi: i pazienti sono tutti visitati almeno una volta il giorno del ricovero e il giorno della dimissione.

Def 1 di ingrassato) BMI iniziale normale(≤ 25) e BMI finale >25

Def 2 di ingrassato) aumento di almeno 5 kg

→ *Def 3 di ingrassato) aumento giornaliero di almeno 300 g*

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME, (V2.PESO-V1.PESO) AS AUMENTO_DI_PESO
FROM ricoveri R, generale G, visite V1, visite V2
WHERE R.CODICE_FISC=G.CODICE_FISC AND V1.CF_PAZIENTE=G.CODICE_FISC
AND V1.CF_PAZIENTE=V2.CF_PAZIENTE AND DATE(V1.DATA)=DATE(R.DATA_INGRESSO)
AND DATE(V2.DATA)=DATE(R.DATA_USCITA)
AND DATEDIFF(R.DATA_USCITA,R.DATA_INGRESSO)>6 AND (V2.PESO-
V1.PESO)/DATEDIFF(R.DATA_USCITA,R.DATA_INGRESSO)>0.3
AND R.DATA_INGRESSO>'2023-12-24' AND R.DATA_USCITA<'2024-01-06'
```

Ho utilizzato la DATE() per prendere soltanto la data, in certi casi in data infatti è presente sia data che ora, TIME() fa lo stesso ma trattenendo l'ora.

- 2) Trovare quanti letti liberi ci sono nei vari reparti oggi (nella tabella reparti, per ogni reparto è riportato il numero di letti)

```
SELECT RE.CODICE, RE.NUMEROLETTI-COUNT(*) AS POSTILIBERI
FROM RICOVERI RI, REPARTI RE
WHERE RI.REPARTO=RE.CODICE AND RI.DATA_USCITA IS NULL
GROUP BY RE.NOME, RE.NUMEROLETTI
```

!Se un reparto è vuoto, non ha pazienti attualmente ricoverati, la query non lo andrebbe a considerare.

UNION

```
SELECT RE.NOME, RE.NUMEROLETTI
FROM REPARTI RE
WHERE RE.CODICE NOT IN (SELECT R.REPARTO FROM RICOVERI R WHERE R.DATA_USCITA IS NULL)
```

- 3) Trovare i pazienti che pur avendo avuto la febbre alta non hanno ricevuto farmaci nel giro di un giorno (es. nella visita del 3-3-2020 il sig. Rossi ha 39 di febbre e quindi mi aspetto che riceva qualcosa entro 24h(considero qualsiasi farmaco))

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME
FROM visite V, generale G
WHERE V.CF_PAZIENTE=G.CODICE_FISC AND V.TEMPERATURA>38.5 AND
```

```
NOT EXISTS (SELECT * FROM somministrazione S, prescrizione P WHERE  
TIMEDIFF(S.DATAORA,V.DATA)<=24 AND S.ID_PRESC=P.ID_PRESC AND P.ID_VISITA=V.ID_VISITA)
```

- 4) Confrontare fra maschi e femmine le età a cui sono stati eseguiti interventi chirurgici “importanti”
(importanti: periodo di ricovero di almeno 20 gg)

Importante scrivere che la data dell'intervento sia compresa tra la data del ricovero e della dimissione.

```
CREATE VIEW CONFRONTO_B AS
```

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME, G.SESSO,  
FLOOR(DATEDIFF(I.data_intervento,G.NASCITA)/365.25) AS ETA
```

```
FROM ricoveri R, generale G, interventi I
```

```
WHERE G.CODICE_FISC=R.CODICE_FISC AND DATEDIFF(R.DATA_USCITA,R.DATA_INGRESSO)>19 AND  
I.codice_fisc=G.CODICE_FISC AND I.data_intervento IS BETWEEN(R.DATA_USCITA,R.DATA_INGRESSO)
```

```
;
```

```
SELECT SESSO,COUNT(*) AS CONTEGGIO, AVG(ETA) AS ETA_MEDIA, STDDEV(ETA) AS  
DEVIAZIONE_STANDARD
```

```
FROM CONFRONTO_B
```

```
WHERE UPPER(SESSO)='M'
```

```
UNION
```

```
SELECT SESSO,COUNT(*) AS CONTEGGIO, AVG(ETA) AS ETA_MEDIA, STDDEV(ETA) AS  
DEVIAZIONE_STANDARD
```

```
FROM CONFRONTO_B
```

```
WHERE UPPER(SESSO)='F'
```

```
GROUP BY SESSO
```

Oppure

```
SELECT G.SESSO, TRUNCATE(AVG(DATEDIFF(I.DATA_INTERVENTO,G.NASCITA)/365.25),1) AS ETAMEDIA,  
TRUNCATE(STDDEV(DATEDIFF(I.data_intervento,G.NASCITA)/365.25),1) AS DEVIAZIONE,COUNT(*) AS  
CONTEGGIO
```

```
FROM generale G, interventi I, ricoveri R
```

```
WHERE G.CODICE_FISC=R.CODICE_FISC AND G.CODICE_FISC=I.codice_fisc
```

```
AND I.data_intervento BETWEEN R.DATA_INGRESSO AND R.DATA_USCITA
```

```
AND DATEDIFF(R.DATA_USCITA,R.DATA_INGRESSO)>=20
```

```
GROUP BY G.SESSO
```

- 5) Si trovino i pazienti che durante le visite ambulatoriali sono sempre stati visitati dallo stesso medico (indicare cf, nome, cognome del medico e il suo reparto attuale [data dimissione nulla])

[ambulatoriale: eseguito al di fuori di un ricovero]

```
SELECT G1.CODICE_FISC AS CF_PZ, G1.NOME AS NOME_PZ, G1.COGNOME AS COGNOME_PZ,
MAX(V.CF_MED), MAX(G2.NOME) AS NOME_MED,
MAX(G2.COGNOME) AS COGNOME_MED, MAX(R.NOME) AS REPARTO_ATTUALE
FROM generale G1, generale G2, visite V, dipendenti D, reparti R
WHERE G1.CODICE_FISC=V.CF_PAZIENTE AND G2.CODICE_FISC=V.CF_MED AND V.CF_MED=D.CODICE_FISC
AND R.CODICE=D.REPARTO
AND D.DATA_DIMISSIONE IS NULL AND NOT EXISTS (SELECT * FROM ricoveri RI WHERE
V.CF_PAZIENTE=RI.CODICE_FISC AND V.DATA BETWEEN RI.DATA_INGRESSO AND RI.DATA_USCITA)
GROUP BY G1.CODICE_FISC, G1.NOME, G1.COGNOME
HAVING MAX(V.CF_MED)=MIN(V.CF_MED)
```

- 6) Considerando il prezzo della dose unitaria si dica, per ogni principio attivo/combinazione di principi attivi, qual è la ditta che lo produce al prezzo più basso

```
SELECT
F.descrizione AS NOME_FARMACO,
MIN(D.NomeDitta) AS NOME_DITTA,
F.principiattivi AS PRINCIPI_ATTIVI,
MIN(F.prezzo_unita_dose) AS PREZZO_DOSE
FROM
dittafarm D,
farmaci F
WHERE
D.IDditta = F.IdDitta
AND F.principiattivi IS NOT NULL
AND F.prezzo_unita_dose = (SELECT MIN(F2.prezzo_unita_dose)
FROM
farmaci F2
WHERE
F2.principiattivi = F.principiattivi)
GROUP BY F.principiattivi
```

- 7) Nell'ambito di un controllo di qualità dei dati, si elenchino i farmaci per i quali non è possibile risalire al responsabile della ditta che li produce

(i motivi possono essere: non c'è il dato del responsabile, il suo telefono, la sua mail o la ditta)

```
SELECT DISTINCT F.descrizione AS NOMEFARMACO
```

```
FROM farmaci F, dittafarm D
```

```
WHERE F.IDditta=D.IDditta AND (D.telefono IS NULL OR D.Responsabile IS NULL OR D.email IS NULL) OR  
F.IdDitta IS NULL
```

```
UNION SELECT F.descrizionefarmaco
```

```
FROM farmaci F
```

```
WHERE F.IdDitta IS NULL
```

- 8) Si elenchino i pazienti ricoverati per i quali, dal momento del ricovero, non è stata richiesta alcuna prestazione, si aggiunga al risultato il tempo trascorso del ricovero

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME, DATEDIFF(CURDATE(),R.DATA_INGRESSO) AS GIORNI
```

```
FROM generale G, ricoveri R
```

```
WHERE G.CODICE_FISC=R.CODICE_FISC AND R.DATA_USCITA IS NULL AND
```

```
NOT EXISTS (SELECT *
```

```
FROM visite V, richiestaprestazioni RIC
```

```
WHERE V.ID_VISITA=RIC.ID_VISITA AND R.CODICE_FISC=V.CF_PAZIENTE AND V.DATA BETWEEN  
R.DATA_INGRESSO AND R.DATA_USCITA)
```

- 1) Trovare i dipendenti che non sono mai stati licenziati

```
SELECT G.CODICE_FISC, G.NOME, G.COGNOME
FROM generale G, dipendenti DIP
WHERE G.CODICE_FISC=DIP.CODICE_FISC AND G.CODICE_FISC NOT IN (SELECT D.CODICE_FISC
FROM dipendenti D WHERE D.LICENZIATO=1 AND D.DATA_DIMISSIONE IS NOT NULL)
```

- 2) Si trovino i pazienti che all'interno dello stesso ricovero sono sempre stati visitati dallo stesso medico

```
SELECT R.CODICE_FISC,R.DATA_INGRESSO, G.NOME AS NOMEPAZ, G.COGNOME AS COGNOME PAZ,
GM.NOME AS NOMEMED, GM.COGNOME AS COGNOMEMED, COUNT(*) AS NUMVISITE
FROM generale G, ricoveri R, visite V, generale GM
WHERE R.CODICE_FISC=V.CF_PAZIENTE AND
GM.CODICE_FISC=V.CF_MED AND R.CODICE_FISC=G.CODICE_FISC AND
V.DATA BETWEEN R.DATA_INGRESSO AND R.DATA_USCITA
GROUP BY R.CODICE_FISC, R.DATA_INGRESSO
HAVING MIN(V.CF_MED)=MAX(V.CF_MED)
```

! nella select devo usare gli stessi attributi della group by... se no, se voglio inserire anche il nome del medico inserisco nella from un'altra tabella generale ma con un altro alias.

Quando c'è il SEMPRE usare HAVING MIN(...)=MAX(...)

- 3) Confrontare la spesa farmaceutica del 2018 e del 2019 vedendo il costo delle prescrizioni

```
SELECT YEAR(V.DATA),
AVG(F.prezzo_unita_dose*P.dose_giorno*(DATEDIFF(P.DATA_INIZIO,P.DATA_FINE))),
STDDEV(F.prezzo_unita_dose*P.dose_giorno*(DATEDIFF(P.DATA_INIZIO,P.DATA_FINE))),
COUNT(*) AS NUMERO_PRESCRIZIONI
FROM prescrizione P, visite V, farmaci F
WHERE P.ID_VISITA=V.ID_VISITA AND YEAR(V.DATA) BETWEEN 2018 AND 2019
# un'altra possibilità: YEAR(V.DATA) IN (2018,2019) oppure (YEAR(V.DATA)=2018 OR YEAR(V.DATA)=2019)
TRIM(P.COD_FARMACO)=TRIM(F.CODICE)
YEAR(V.DATA) BETWEEN 2018 AND 2019
GROUP BY YEAR(V.DATA)
```

- 4) Si dica qual è la malattia più frequente come causa principale di ricovero

```
CREATE VIEW DIAGNOSI AS
SELECT COUNT(*) AS numeroricoveri, I.DESCRIZIONE
FROM ricoveri R, icd9 I
WHERE R.DIAGNOSI_RICOVERO=I.CODICEHMJ
```

GROUP BY I.DESCRIZIONE ORDER BY numericoveri DESC LIMIT 1

SELECT D.* FROM diagnosi D

WHERE D.numericoveri=(SELECT MAX(D1.numericoveri) FROM diagnosi D1)

5) Si scriva una query per confrontare le età dei pazienti operati dai vari chirurghi

SELECT I.CF_CHIRURGO, GC.NOME, GC.COGNOME,

AVG(DATEDIFF(I.DATA_INTERVENTO,GP.NASCITA)/365.25),

STDDEV(DATEDIFF(I.DATA_INTERVENTO,GP.NASCITA)/365.25),

COUNT(*)

FROM interventi I, generale GP, generale GC

WHERE I.CODICE_FISC=GP.CODICE_FISC AND I.CF_CHIRURGO=GC.CODICE_FISC

GROUP BY I.CF_CHIRURGO, GC.NOME, GC.COGNOME

- 1) Scrivere una query che eseguita giornalmente elenchi i reparti in cui la media delle temperature misurate ai pazienti ancora ricoverati (considerando le visite delle ultime 24 ore) è superiore a 38 gradi. Si riporti il nome del reparto e la temperatura media e si ordini il risultato per temperatura media decrescente

```
SELECT REP.NOME AS NOME_REPARTO, AVG(V.TEMPERATURA) AS MEDIA_TEMPERATURA
FROM ricoveri R, visite V, reparti REP
WHERE R.CODICE_FISC=V.CF_PAZIENTE AND TIMEDIFF(CURDATE(),V.DATA)<24 AND R.DATA_USCITA IS
NULL
AND R.REPARTO=REP.CODICE AND V.DATA BETWEEN R.DATA_INGRESSO AND R.DATA_USCITA
GROUP BY REP.NOME
HAVING AVG(V.TEMPERATURA)>38
ORDER BY AVG(V.TEMPERATURA) DESC
```

- 2) Scrivere una query che trovi per ogni reparto il rapporto tra il numero di dipendenti ed il numero di posti letto in modo da capire se ci sono troppi o troppi pochi dipendenti per reparto

```
SELECT REP.NOME AS NOME_REPARTO, COUNT(*)/REP.NumeroLetti AS RAPPORTO_DIPENDENTI_LETTI
FROM reparti REP, dipendenti D
WHERE D.REPARTO=REP.CODICE AND D.DATA_DIMISSIONE IS NULL
GROUP BY REP.NOME, REP.NumeroLetti
ORDER BY COUNT(*)/REP.NumeroLetti ASC
```

Nella group by bisogna mettere tutte le colonne della select che stanno fuori funzioni di aggregazione, in questo caso anche REP.NumeroLetti che è dividendo di una funzione di aggregazione.

- 3) Trovare se ci sono reparti con un numero sproporzionato di uomini e donne

```
SELECT R.NOME, UPPER(G.SESSO) AS SESSO,COUNT(*) AS NUM_DIPENDENTI
FROM generale G, dipendenti D, reparti R
WHERE G.CODICE_FISC=D.CODICE_FISC AND D.REPARTO=R.CODICE AND D.DATA_DIMISSIONE IS NULL
GROUP BY G.NOME, G.SESSO
```

- 4) Trovare i chirurghi che dopo un intervento, una volta che i pazienti sono stati dimessi, sono stati ricoverati di nuovo entro 4gg

```
SELECT DISTINCT G.CODICE_FISC, G.NOME, G.COGNOME
FROM generale G, interventi I, ricoveri R1, ricoveri R2
WHERE I.codice_fisc=R1.CODICE_FISC AND I.codice_fisc=R2.CODICE_FISC AND G.CODICE_FISC=I.Cf_chirurgo
AND I.data_intervento
BETWEEN R1.DATA_INGRESSO AND R1.DATA_USCITA
AND DATEDIFF(R2.DATA_INGRESSO,R1.DATA_USCITA)<=4 AND R2.DATA_INGRESSO>R1.DATA_USCITA
```

- 5) Si trovino i medici che non hanno mai misurato la temperatura durante le visite

Quando c'è il MAI bisogna utilizzare una subquery

```
SELECT DISTINCT G.CODICE_FISC, G.NOME, G.COGNOME
FROM visite V, generale G
WHERE V.CF_MED=G.CODICE_FISC AND V.CF_MED NOT IN (SELECT V2.CF_MED FROM visite V2
WHERE V2.TEMPERATURA IS NOT NULL)
```

6) Si trovino i medici che non hanno mai misurato la temperature e il peso

```
SELECT DISTINCT G.CODICE_FISC, G.NOME, G.COGNOME
FROM visite V, generale G
WHERE V.CF_MED=G.CODICE_FISC AND V.CF_MED NOT IN (SELECT V2.CF_MED FROM visite V2
WHERE V2.TEMPERATURA IS NOT NULL OR V2.TEMPERATURA IS NOT NULL)
```

Non va bene la AND se no c'è se ci fosse un medico che una volta non ha mai misurato entrambe e altre volte lo ha fatto uscirebbe comunque)

7) Trovare i pazienti che hanno fatto solo visite ambulatoriali (= visite al di fuori del ricovero).

```
SELECT G.CODICE_FISC,G.NOME,G.COGNOME
FROM visite V, generale G
WHERE G.CODICE_FISC=V.CF_PAZIENTE AND V.CF_PAZIENTE NOT IN
(SELECT V1.CF_PAZIENTE
FROM visite V1, ricoveri R
WHERE V1.CF_PAZIENTE=R.CODICE_FISC AND (R.DATA_USCITA IS NOT NULL
AND V1.DATA BETWEEN R.DATA_INGRESSO AND R.DATA_USCITA) OR R.DATA_USCITA IS NULL AND V1.DATA
BETWEEN R.DATA_INGRESSO AND CURDATE())
```

oppure

Ipotesi: chi ha fatto solo visite ambulatoriali vuol dire che non è mai stato ricoverato:

```
SELECT DISTINCT G.CODICE_FISC, G.NOME,G.COGNOME
FROM generale G, visite V
WHERE G.CODICE_FISC=V.CF_PAZIENTE AND V.CF_PAZIENTE NOT IN
(SELECT R.CODICE_FISC FROM ricoveri R)
```

8) Capire se c'è una correlazione fra il ruolo dei dipendenti e la durata dei loro periodi lavorativi

```
SELECT P.descrizione AS PROFESSIONE, COUNT(*) AS CONTEGGIO,
AVG(DATEDIFF(D.DATA_DIMISSIONE,D.DATA_ASSUNZIONE) AS DURATA_MEDIA
STDDEV(DATEDIFF(D.DATA_DIMISSIONE,D.DATA_ASSUNZIONE)) AS DEVIAZIONE_STANDARD
FROM professioni P, esercita_prof EP, dipendenti D
```

WHERE D.CODICE_FISC=EP.CODICE_FISC AND P.codice=EP.CODICE_PROF AND D.DATA_DIMISSIONE IS NOT NULL

GROUP BY P.descrizione