## **5.2 *Definizione delle interrogazioni per la visualizzazione dei dati***

**Query [1]:**

*Fornire l’elenco dei pazienti che hanno richiesto, per due volte consecutive, l’assegnazione temporanea ad uno stesso medico.*

select P.Codice\_Fiscale, P.Nome, P.Cognome

from Afferenza\_Medica\_Temporanea as A, Afferenza\_Medica\_Temporanea as B, Persona as P

where A.Persona=B.Persona and A.Medico=B.Medico and A.Data\_Inizio<>B.Data\_Inizio and A.Data\_Fine<>B.Data\_Fine and A.Persona=P.Codice\_Fiscale and B.Persona=P.Codice\_Fiscale

and not exists (select \*

from Afferenza\_Medica\_Temporanea as C

where C.Persona=A.Persona and C.Persona=B.Persona and C.Medico<>A.Medico and C.Medico<>B.Medico and

((C.Data\_Inizio>=A.Data\_Fine and C.Data\_Fine<=B.Data\_Inizio) or

(C.Data\_Inizio>=B.Data\_Fine and C.Data\_Fine<=A.Data\_Inizio)))

**Query[2]:**

*Fornire, per ogni ospedale di una data ASP, nome, cognome e specializzazione del medico che ha effettuato il maggior numero di interventi.*

select P.Nome, P.Cognome, NM.Numero\_Interventi, S.Specializzazione

from Numero\_Interventi\_Massimi as NM, Persona as P,Specializzazione\_Medico as S

where NM.Medico=P.Codice\_Fiscale and NM.Medico=S.Medico\_Specialista

**Query[3]:**

*Trovare i pazienti che hanno effettuato almeno un monitoraggio per la patologia “diabete”, verificando i valori di pressione e glicemia, ma non hanno mai ottenuto valori accettabili (cioè all’interno del range di riferimento).*

select pr.Codice\_Fiscale,pr.Nome,pr.Cognome

from Patologia\_Effettiva as pe,Patologia as p,Persona as pr

where p.Codice=pe.Patologia and p.Nome like "Diabete%"

and pr.Codice\_Fiscale=pe.Paziente and not exists

(select \*

from Monitoraggio as m, Dato\_Monitorabile as dm

where m.Patologia=pe.ID\_Patologia\_Effettiva and

m.ID\_Indicatore=dm.ID and (dm.Nome="Glicemia" or dm.Nome="Pressione") and m.Valore>=dm.Valore\_Minimo and m.Valore<=dm.Valore\_Massimo)

**Query[4]:**

*Trovare l’elenco dei pazienti che hanno sostenuto esami del sangue solo nell’ambito di un ricovero in ospedale o in cliniche private.*

select P.Codice\_Fiscale, P.Nome, P.Cognome

from Persona as P

where P.Codice\_Fiscale IN (select ES.Paziente

from Esame\_Sostenuto as ES, Esame as E

where ES.Esame=E.Codice and E.Categoria="Sangue" and

NOT EXISTS (select \*

from Ricovero as Ric

where Ric.Paziente=ES.Paziente and

(ES.Data<Ric.Data\_Inizio\_Ricovero or ES.Data>Data\_Fine\_Ricovero)));

**Query[5]:**

*Trovare l’elenco dei medici per cui esiste almeno un paziente al quale egli ha prescritto, mediante una ricetta rossa, farmaci prescrivibili per ognuna delle patologie avute.*

Select AMDB.Medico

From Afferenza\_Medica as AMDB

Where AMDB.Paziente NOT in ( Select PE.Paziente

From Patologia\_Effettiva as PE

Where NOT exists

(Select \*

From Prescrizione\_Paziente as PP, Ricetta as R, Farmaco as F

Where PP.Patologia=PE.ID\_Patologia\_Effettiva AND

((PP.Ricetta=R.Codice AND R.Tipo<>"Rossa") OR

(PP.Farmaco=F.Codice\_Ean AND F.Prescrivibile<>"1"))));

**Stored Function [6]:**

*Scrivere una stored function che, dato un ospedale e una specializzazione, fornisca il numero di ricoveri durante i quali c’è stato uno spostamento di letto ma non di stanza.*

drop function if exists Ricoveri;

DELIMITER //

CREATE FUNCTION Ricoveri (Partita\_Iva VARCHAR(11), Specializzazione VARCHAR (20)) returns INT

BEGIN

DECLARE Numero INT;

select count(\*) into NUMERO

from Specializzazione\_Struttura as S,Ricovero as R

where S.Ospedale=Partita\_Iva and S.Nome=Specializzazione and

R.Ospedale=Partita\_Iva

and R.ID\_Ricovero IN

(select A.Ricovero

from Assegnazione\_Letto as A, Posto\_Letto as P

where A.Posto\_Letto=P.ID\_Posto\_Letto and

A.Ricovero IN

(select AL.Ricovero

from Assegnazione\_Letto as AL, Posto\_Letto as PS, Reparto as Re, Stanza as ST

where PS.ID\_Posto\_Letto<>P.ID\_Posto\_Letto and

P.ID\_Stanza=PS.ID\_Stanza and ST.ID\_Stanza=P.ID\_Stanza and

ST.ID\_Stanza=PS.ID\_Stanza and ST.ID\_Reparto=Re.ID\_Reparto and RE.ID\_Nome=Specializzazione));

return Numero;

END;

//

DELIMITER ;

**Stored Function[7]:**

*Scrivere una stored function che restituisca, per un dato medico, il numero medio mensile di ricette prescritte in un dato anno.*

CREATE FUNCTION `Ricette\_Medie`(Codice\_Fiscale VARCHAR (16), Anno INT) RETURNS int(11)

BEGIN

DECLARE Somma\_Ricette INT Default 0;

select count(\*) into Somma\_Ricette

from Ricetta as R, Visita\_Medica as V

where V.Codice\_Fiscale\_Medico=Codice\_Fiscale and year(V.Data)=Anno

and V.ID\_Visita=R.ID\_Visita;

return Somma\_Ricette/12;

END

**Trigger[8]:**

*Creare un trigger che, allo scadere dell’assegnazione temporanea di un paziente a un medico, prepari una email di avviso da inviare al medico e al paziente e la inserisca in una coda di invio e-mail (Si supponga che esista un’applicazione esterna che si occupa dell’attività).*

delimiter //

create trigger Invio\_Comunicazione

after update on Afferenza\_Medica

for each row

begin

declare eP VARCHAR(45);

declare eM VARCHAR(45);

declare d date;

set d=new.Data\_a;

if(new.Data\_a is not null)

then

select p.email,p2.email into eP,eM

from Persona as p,Persona as p2

where new.Paziente=p.Codice\_Fiscale and p2.Codice\_Fiscale=new.Medico;

insert into Comunicazione(DA,A,Testo,Stato,Data\_Invio)

values ("SistemaSanitarioNazionale@gov.it",eM,"Si comunica che è scaduta l'assegnazione temporanea.....","Da\_Invire",d);

insert into Comunicazione(DA,A,Testo,Stato,Data\_Invio)

values ("SistemaSanitarioNazionale@gov.it",eP,"Si comunica che è scaduta l'assegnazione temporanea.....","Da\_Invire",d);

end if;

end;//

delimiter ;

**Processo Schedulato [9]:**

*Scrivere un processo schedulato ogni 31 dicembre, che generi un report annuale che restituisce, per ogni ASL, per ogni mese e per ogni medico, l’elenco dei farmaci prescritti con relativo costo.*

delimiter //

create event Report\_Annuale

on schedule every 1 year

starts '2015-12-31'

do

begin

select v.Numero\_Progressivo,monthname(r.Data\_Emissione),

v.Codice\_Fiscale\_Medico,f.Codice\_Ean,f.Costo INTO OUTFILE '/home/marcuz/Scrivania/Report.txt'

from Visita\_Medica as v,Ricetta as r,Prescrizione\_Paziente as p,Farmaco as f

where v.ID\_Visita=r.ID\_Visita and p.Farmaco=f.Codice\_Ean and p.Ricetta=r.Codice

group by v.Numero\_Progressivo,month(r.Data\_Emissione),v.Codice\_Fiscale\_Medico,

f.Codice\_Ean,f.Costo;

end //

delimiter ;