**Query(linguaggio naturale)**

Le Query che verranno eseguite sulla base di dati sono le seguenti:

1. Estrarre i receptionist che non sono stati assegnati ad alcun turno di lavoro il giorno corrente
2. Estrarre i dati dei addetti alla sicurezza che percepiscono uno stipendio superiore ai 1000 euro mensili
3. Estrarre il numero di pacchetti offerta acquistati nel giorno corrente e i codici dei clienti che li hanno acquistati
4. Estrarre il numero dei partecipanti agli eventi nel giorno corrente
5. Registrare l’acquisto di un biglietto per un’attrazione
6. Registrare l’acquisto di un biglietto per un pacchetto offerta
7. Estrarre i dati degli animali che vanno nutriti dall’inserviente scelto
8. Registrare l’acquisto di un biglietto per un evento speciale
9. Estrarre il numero di clienti con età inferiore a quella scelta che sono entrati nel parco
10. Estrarre i dati dei lavoratori dipendenti che presentano dei benefit nel contratto
11. Assumere un nuovo dipendente ed eventualmente inserire dei bonus
12. Assumere un nuovo addestratore

**Query(SQL)**

Le Query il linguaggio SQL sono le seguenti:

1.

"create or replace view ReceptionistTurno as select \* from `addetto reception` r join `assegnazione tr` on CodiceD = CodiceDRec, `turno lavoro` where nTurno=NumT and DTurno=DataT;

select distinct CodiceD, Nome, Cognome, AnniServC from ReceptionistTurno

where CodiceD not in (select CodiceD from ReceptionistTurno where DataT = "+data+");"

2.

“select a. \* from `addetto sicurezza` a join `firma ds` on CodiceD = CodiceDSic, contratto

where IDContratto = ID and Stipendio>1000;”

3.

“create or replace view AcquistiPacchettiOfferta as select a.numEntrata, t.\*, r.idP from acquisto a, transazione t, `riferimento p` r where a.codiceT=t.codiceT and a.codiceT=r.codiceT;

select count(\*), numEntrata from AcquistiPacchettiOfferta a where a.data="+data+";”

4.

“create or replace view AcquistiEventi as select a.numEntrata, t.codiceT, e.DataEvento, e.NomeEvento from `riferimento e` e, transazione t, acquisto a

where a.codiceT=t.codiceT and a.codiceT=e.codiceT;

select count(\*) from AcquistiEventi where DataEvento="+data+";”

5.

“if(not exists(select numEntrata from cliente where Nome="+nome+" and Cognome="+cognome+" and età="+età+";))

then insert into cliente (nome, cognome, età) values("+nome+","+cognome+","+età+");

end if;

insert into transazione (importo, data) values(2,"+data+");

insert into acquisto (codiceT, numEntrata) values((select max(codiceT) from transazione),(select numEntrata from cliente where Nome="+nome+" and Cognome="+cognome+" and età="+età+”));

insert into `riferimento a` (codiceT, CodiceAttrazione) values((select max(codiceT) from transazione),"+codiceattrazione+");”

N.B. In questo caso si posono usare variabili per codiceT e numEntrata dovendo essere calcolate più volte

6.

“create or replace view acquistipacchettiinc as select a.\*, i.CodiceA from acquistipacchettiofferta a , inclusione i where i.numPac=a.idp;

if(not exists(select numEntrata from cliente where Nome="+nome+" and Cognome="+cognome+" and età="+età+"))

then insert into cliente (nome, cognome, età) values("+nome+","+cognome+","+età+");

end if;

if(not exists(select codcieT from acquistipacchettiinc where numEntrata="+numEntrata+" and idp="+idp+" and data="+data+"))

then insert into transazione (importo, data) values(4,"+data+");

insert into acquisto (codiceT, numEntrata) values((select max(codiceT) from transazione),(select numEntrata from cliente where Nome="+nome+" and Cognome="+cognome+" and età="+età+”));

insert into `riferimento p` (codiceT, CodiceAttrazione) values((select max(codiceT) from transazione),"+idp+");

end if;”

N.B. In questo caso si possono usare variabili per codiceT e numEntrata dovendo essere calcolate più volte

7.

“select a.\* from animale a, nutrimento , inserviente i where CodiceD=CodiceDI and a.nome = NomeAnimale and a.specie = SpecieAnimale and i.nome ="+nome+ " and i.cognome ="+cognome+";”

8.

“create or replace view AcquistiEventi as select a.numEntrata, t.codiceT, e.DataEvento, e.NomeEvento from `riferimento e` e, transazione t, acquisto a

where a.codiceT=t.codiceT and a.codiceT=e.codiceT;create or replace view AcquistiEventi as select a.numEntrata, t.codiceT, e.DataEvento, e.NomeEvento from `riferimento e` e, transazione t, acquisto a

where a.codiceT=t.codiceT and a.codiceT=e.codiceT;

if(not exists(select numEntrata from cliente where Nome="+nome+" and Cognome="+cognome+" and età="+età+"))

then insert into cliente (nome, cognome, età) values("+nome+","+cognome+","+età+");

end if;

if(not exists(select codiceT from acquistieventi where numEntrata="+numEntrata+" and NomeEvento="+nomeevento+" and DataEvento="+dataevento+"))

then insert into transazione (importo, data) values(5,"+data+");

insert into acquisto (codiceT, numEntrata) values((select max(codiceT) from transazione),(select numEntrata from cliente where Nome="+nome+" and Cognome="+cognome+" and età="+età+”))

insert into `riferimento e` (codiceT, NomeEvento, DataEvento)

values((select max(codiceT) from transazione),"+nomeevento+","+dataevento+")

end if;”

N.B. In questo caso si posono usare variabili per codiceT e numEntrata dovendo essere calcolate più volte

9.

“select count(\*) from cliente where età<="+età+";”

10.

“create or replace view Lavoratori as select \* from `addetto sicurezza` union all select \* from `addetto reception` union all select \* from inserviente;

create or replace view ContrattiConBenefit as select i.\* from inclusioneb i, contratto c where c.ID=i.IDContratto;

create or replace view Firme as select \* from `firma di` union all select \* from `firma ds` union all select \* from `firma dr`;

select distinct l.\* from Lavoratori l, ContrattiConBenefit c, Firme f where l.CodiceD=f.CodiceDI and f.IDContratto=c.IDContratto;”

11.

“if(not exists(select CodiceD from “+tipod” where Nome="+nome+" and Cognome="+cognome+" and età="+età+"))

then insert into +”tipod”+ (nome, cognome, età) values("+nome+","+cognome+","+età+");

else

delete from "+tipoF+" where "+column+"=(select CodiceD from “+tipod” where Nome="+nome+" and Cognome="+cognome+" and età="+età+");

end if;

insert into contratto (Stipendio, AnniC, Selettore) values("+stipendio+","+anniC+",'dipendente');

insert into "+tipoF+"(IDContratto, "+column+") values ((select ID from contratto where ID in (select max(ID) from contratto)),(select CodiceD from “+tipod” where Nome="+nome+" and Cognome="+cognome+" and età="+età+"));

insert into stipulazione(CFDirettore,IDContratto) values((select CF from direttore),(select ID from contratto where ID in (select max(ID) from contratto))”

N.B.Si possono usare variabili

11a.

“create or replace view contrattiassegnati as select distinct l.\*, c.ID from lavoratori l join contratto c join `firma di` i join `firma ds` s join `firma dr` r on (l.CodiceD=i.CodiceDI and i.IDContratto=c.ID) or (l.CodiceD=r.CodiceDRec and r.IDContratto=c.ID) or (l.CodiceD=s.CodiceDSic and s.IDContratto=c.ID);

insert into inclusioneb(IDContratto,nOrdine) values((select ID from contrattiassegnati ca where ca.CodiceD="+codiceD+";),"+benefit+");”

N.B.Si possono usare variabili

12.

“if(not exists(select CF from addestratore where CF="+CFA+";))

then insert into addestratore (nome, cognome,CF) values("+nome+","+cognome+","+CFA+");

else delete from `firma a` where CFAddestratore="+CFA+";

end if;

insert into contratto (Stipendio, Selettore, Giorni) values("+stipendio+",'aOre',"+giorni+");

insert into `firma a`(IDContratto, CFAddestratore) values ((select ID from contratto where ID in (select max(ID) from contratto)),"+CFA+");

insert into stipulazione(CFDirettore,IDContratto) values((select ID from contratto where ID in (select max(ID) from contratto)),(select CF from direttore))”

N.B. Si Possono usare variabili

12a.

“insert into inclusioneb(IDContratto,nOrdine) values((select c.ID from contratto c, `firma a` f where f.CFAddestratore="+CF+" and f.IDContratto=c.ID),"+benefit+");”