

Relazione per il corso di Basi di Dati
A.A 2014/2015

Gestione di un'Agenzia Assicurativa

Candidato: Barco Simone

Matricola: 1052302

ABSTRACT

L'assicurazione ha lo scopo principale di trasformare il rischio in una spesa, infatti attraverso la stipula di un contratto (o polizza), l'assicurando quantifica il danno patrimoniale che esso avrebbe se l'evento garantito (rischio) si verificasse.

Le polizze assicurative possono essere sottoscritte come libera scelta tra individui e società di assicurazione (agenzie) e possono riguardare svariati campi (assicurazioni sul verificarsi di infortuni, di malattie, di incidenti, di eventi naturali, di danni a cose, ecc).

La polizza assicurativa mette in relazione l'agenzia assicurativa con tre tipi di persone:

- Contraente: persona fisica che stipula la polizza;
- Assicurato: persona fisica il cui interesse è protetto dalla polizza;
- Beneficiario: persona fisica destinataria della prestazione cui si obbliga l'assicuratore.

Per la base di dati che si desidera progettare si suppone che questi tre tipi di persone siano sempre la stessa (quindi contraente, assicurato e beneficiario sono sempre lo stesso cliente).

Una polizza oltre a specificare le persone coinvolte in questo contratto, descrive un massimale (indica il livello massimo che l'assicurato può essere risarcito per i danni provocati dall'evento che si assicura). Specifica anche la data di stipula della polizza e la data di scadenza, dopo la quale si può rinnovare o no.

Un'agenzia assicurativa mette a disposizione varie persone (dipendenti) che possono stipulare le polizze.

Si desidera progettare una base di dati per la gestione automatizzata di un'agenzia generale di assicurazioni, composta da un certo numero di agenzie gestite da un agente in cui lavorano diversi dipendenti. In particolare, si vogliono gestire le varie sedi staccate dell'agenzia generale e le polizze assicurative sulla Casa e RC auto.

Le operazioni principali sono: l'aggiunta di nuovi clienti ed dipendenti; la modifica di clienti e dipendenti già esistenti ed infine la creazione di nuove polizze.

Tale progetto prevede anche la realizzazione di un'interfaccia web attraverso la quale tutti gli utenti possono interagire con la base di dati.

DESCRIZIONE DEI REQUISITI

Gli utenti di tale base si suddividono in clienti e dipendenti, i quali possono essere agenti, subagenti e produttori. Di ogni utente è richiesto: codice fiscale, nome, cognome, e password per accedere all'interfaccia web.

Di ogni cliente interessa conoscere: data di nascita, città e provincia di residenza, data di conseguimento della patente e la classe di merito.

Per ogni dipendente è richiesta una provvigione (percentuale che ogni agente prende dalla stipula di una polizza); per il subagente è di interesse conoscere anche la sede del suo ufficio, mentre per il produttore è richiesta la durata del suo contratto presso l'agenzia di assicurazione. L'agente è il responsabile dell'agenzia.

Quando un cliente stipula una polizza assicurativa il dipendente si occupa di registrarla nel sistema, inserendo: massimale (indica il livello massimo che l'assicurato può essere risarcito per i danni provocati dall'evento che si assicura), data di stipula e scadenza della polizza e tipo di assicurazione

L'assicurazione presenta una durata, una descrizione, un prezzo annuo e può essere:

- RCAuto: per poter inserire questo tipo di assicurazione sono richiesti i seguenti dati: potenza e cilindrata;
- Casa: sono richiesti l'anno di costruzione della casa, anno di ristrutturazione (se presente), la copertura legale in caso di causa in tribunale e infine per cosa si assicura la casa (incendio, eventi naturali, ecc);

Un dipendente possiede un portafoglio in cui sono presenti tutti gli importi delle polizze che ha stipulato, calcolati in base alla provvigione del dipendente.

Inoltre un dipendente fa riferimento ad un'agenzia, che è gestita da un agente che possiede tutti i portafogli di subagenti e produttori che appartengono a questa agenzia.

Le operazioni che il sistema deve offrire sono l'inserimento e la modifica dei dati di un cliente, di un dipendente e l'inserimento e rinnovo di una polizza; l'inserimento automatico di una polizza nel portafoglio del dipendente che la stipula calcolando l'importo in base al prezzo annuo dell'assicurazione e la provvigione di tale dipendente.

GLOSSARIO

<u>Termine</u>	<u>Descrizione</u>	<u>Sinonimi</u>	<u>Collegamenti</u>
Utente	Persona che usufruisce della base di dati. Si suddivide in Cliente e Dipendente	Cliente, Dipendente	
Cliente	Utente che richiede la stipula di una Polizza Assicurativa	Contraente, Beneficiario, Assicurato	Polizza Assicurativa
Dipendente	Utente che lavora per l'Agenzia; può essere Agente, SubAgente o Produttore. Ogni dipendente possiede un portafoglio. Stipula Polizze Assicurative.	Agente, SubAgente, Produttore	Polizza Assicurativa, Portafoglio, Agenzia
Assicurazione	Descrive il tipo di evento che si vuole assicurare. Si suddivide in Casa o RCAuto	Casa, RCAuto	Polizza Assicurativa
Portafoglio	Descrive la lista in cui sono presenti gli importi delle polizze assicurative stipulate da un Dipendente		Polizza Assicurativa, Dipendente
Polizza Assicurativa	Descrive il contratto che viene stipulato tra Dipende e Cliente. Riguarda un tipo di Assicurazione	Polizza	Dipendente, Cliente, Assicurazione
Agenzia	Indica la sede dell'agenzia che fa capo a un determinato numero di Dipendenti		Dipendente

PROGETTAZIONE CONCETTUALE

Lista delle entità:

→ **Utente:** descrive gli utenti che possono accedere all'interfaccia web.

- *CodiceFiscale*: varchar(16)
- *Password*: varchar(10)
- *Nome*: varchar(30)
- *Cognome*: varchar(30)

E' stata definita la seguente generalizzazione totale ed esclusiva:

→ **Cliente:** descrivono i clienti che possono richiedere una polizza assicurativa.

- *DataNascita*: date
- *CittaResidenza*: varchar(30)
- *ProvinciaResidenza*: char(2)
- *DataPatente*: date
- *AnniPatente*: int(2)
- *ClassediMerito*: int(2)

→ **Dipendente:** descrivono i dipendenti che lavorano per l'agenzia generale, stipulano polizze, inseriscono nuovi clienti, rinnovano polizze già esistenti.

- *Provvigione*: int(5)

E' stata definita un'ulteriore generalizzazione totale ed esclusiva:

- ♦ **Agente:** descrivono i capi delle sedi staccate dell'agenzia generale, oltre alle funzioni di un Dipendente normale possono anche aggiungere nuovi subagenti o produttori alla sua sede staccata.
- ♦ **SubAgente:** descrivono dei Dipendenti che hanno un proprio ufficio.
 - *SedeUfficio*: varchar(30)
- ♦ **Produttore:** descrivono dei Dipendenti che hanno un contratto determinato.
 - *DurataContratto*: int(3)

→ **Polizza:** descrivono le Polizze che vengono stipulate tra Cliente e Dipendente.

- *Numero*: int(5)
- *Massimale*: int(7)
- *DataStipula*: date
- *DataScadenza*: date

→ **Assicurazione:** descrivono i tipi di assicurazione, possono essere RCAuto e Casa.

- *Codice*: int(4)
- *Durata*: int(2)
- *PrezzoAnnuo*: int(4)
- *Descrizione*: varchar(10)

E' stata definita una generalizzazione totale ed esclusiva:

→ **Casa**: describe la polizza sulla Casa

- *AnnoCostruzione*: int(4)
- *AnnoRistrutturazione*: int(4)
- *SpeseLegali*: int(5)
- *Tipo*: varchar(10)

→ **RCAuto**: describe la polizza RCAuto.

- *Potenza*: int(4)
- *Cilindrata*: int(5)

→ **Portafoglio**: describe il portafoglio del Dipendente con l'Importo della polizza che ha stipulato calcolato in base alla sua provvigione.

- *Importo*: int(4)

→ **Agenzia**: describe la sede staccata dell'agenzia generale.

- *Codice*: int(4)
- *CittaSede*: varchar(30)
- *ProvinciaSede*: char(2)

Lista delle relazioni:

→ **Dipendente – Agenzia: appartenenza**

- Un dipendente appartiene a un'agenzia;
- Ad un'agenzia appartiene almeno un dipendente;
- Molteplicità: uno a molti;

→ **Agente – Agenzia: responsabile**

- Un agente è responsabile di un'agenzia;
- Un'agenzia ha uno e un solo agente;
- Molteplicità: uno a uno;

→ **Dipendente – Portafoglio: possesso**

- Un dipendente possiede uno e un solo portafoglio;
- Un portafoglio è posseduto da uno e un solo dipendente
- Molteplicità: uno a uno;

→ **Dipendente – Polizza: stipulazione**

- Un dipendente stipula almeno una polizza;
- Una polizza è stipulata da uno e un solo dipendente;
- Molteplicità: uno a molti;

➔ **Polizza – Portafoglio: composizione**

- Una polizza compone uno e un solo portafoglio;
- Un portafoglio è composto da almeno una polizza;
- Molteplicità: uno a molti;

➔ **Cliente – Polizza: richiesta**

- Un cliente richiede almeno una polizza;
- Una polizza viene richiesta da almeno un cliente;
- Molteplicità: molti a molti;

➔ **Polizza – Assicurazione: riferimento**

- Un polizza fa riferimento a una e una sola assicurazione;
- Una assicurazione si riferisce ad almeno una polizza;
- Molteplicità: uno a molti;

Lista delle generalizzazioni:

➔ **Utente: Cliente/Dipendente**

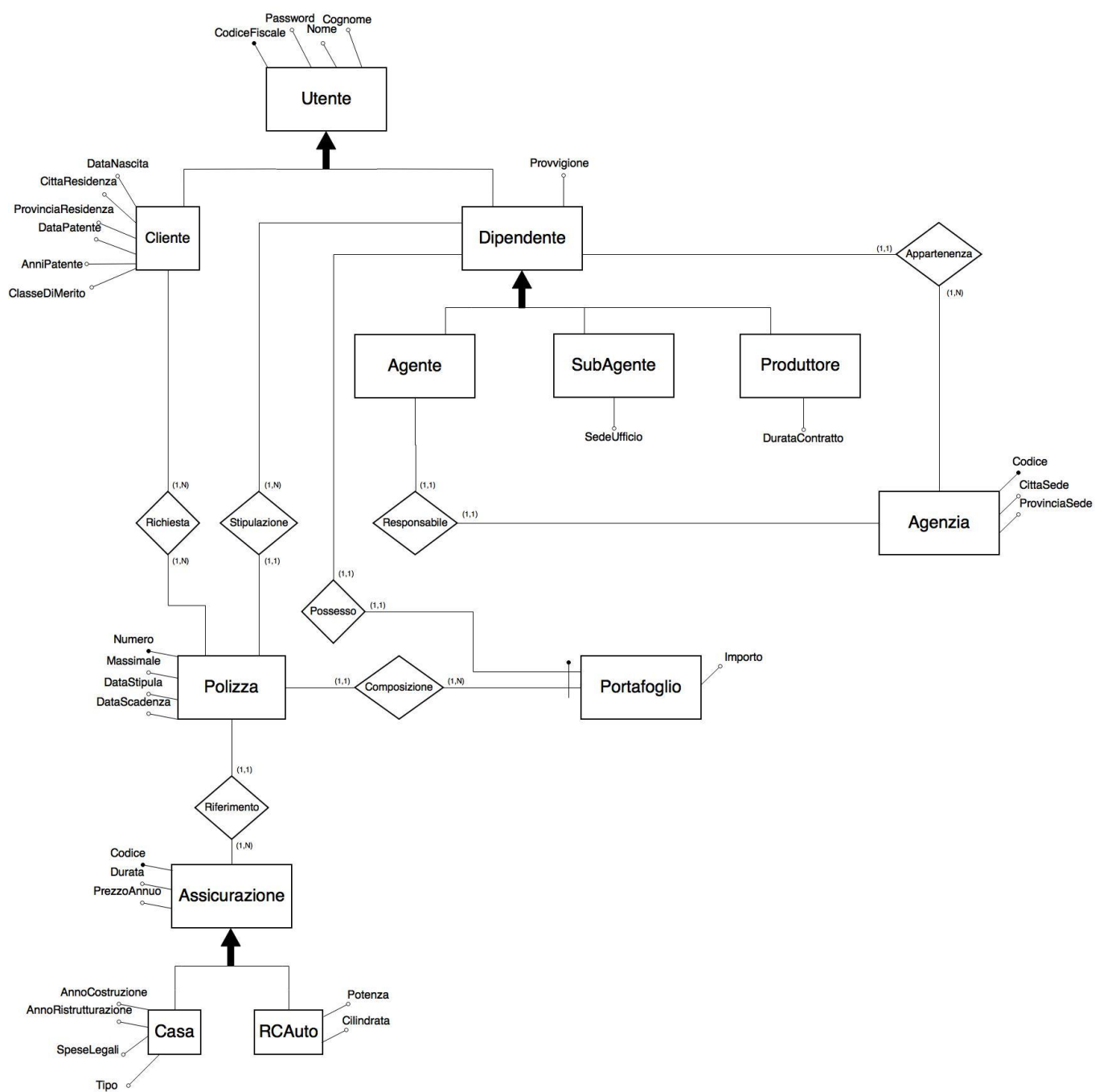
- Le entità figlie Cliente e Dipendente sono esclusive in quanto un Dipendente non può essere un Cliente dal momento che ha una provvigione.
- La generalizzazione è totale poiché ogni occorrenza dell'entità Utente è una occorrenza di una delle due entità figlie.

➔ **Dipendente: Agente/SubAgente/Produttore**

- Le entità figlie Agente, SubAgente e Produttore sono esclusive in quanto un SubAgente non può essere Produttore o Agente dal momento che possiede una SedeUfficio; un Produttore non può essere Agente o SubAgente in quando possiede una DurataContratto; un Agente non può essere SubAgente o Produttore visto che è responsabile di un'agenzia.
- La generalizzazione è totale poiché ogni occorrenza dell'entità Dipendente è occorrenza di una delle entità figlie.

➔ **Assicurazione: Casa/RCAuto**

- Le entità figlie Casa e RCAuto sono esclusive in quanto Casa non può essere RCAuto dal momento che possiede un AnnoCostruzione, AnnoRistrutturazione, SpeseLegali e Tipo.
- La generalizzazione è totale poiché ogni occorrenza dell'entità Assicurazione è occorrenza di una delle entità figlie.



PROGETTAZIONE LOGICA

Ristrutturazione dello schema concettuale:

Per semplificare la traduzione dello schema logico in schema relazionale è necessaria una ristrutturazione dello schema concettuale che permetterà un maggiore supporto dallo schema risultante anche in fatto di ottimizzazione attraverso costrutti supportati dallo schema stesso.

Si considerino per la base di dati le seguenti operazioni, la tavola dei volumi e delle operazioni.

Operazioni:

Operazione 1: inserimento nuovo produttore

Operazione 2: stampa dei portafogli

Operazione 3: inserimento nuovo tipo di assicurazione

Operazione 4: stampa tutte le polizze attive

Operazione 5: inserimento nuovo cliente

Operazione 6: inserimento nuova polizza

Tavola dei Volumi e Tavola delle Operazioni:

<u>Concetto</u>	<u>Tipo</u>	<u>Volume</u>
Utente	E	15
Cliente	E	4
Dipendente	E	11
Agente	E	2
SubAgente	E	5
Produttore	E	4
Polizza	E	6
Assicurazione	E	6
Casa	E	3
RCAuto	E	3
Portafoglio	E	6
Agenzia	E	2
Appartenenza	R	11
Responsabile	R	2
Possesso	R	11
Stipulazione	R	6
Composizione	R	6
Richiesta	R	6
Riferimento	R	6

<u>Operazione</u>	<u>Frequenza</u>
Operazione 1	3 volte al giorno
Operazione 2	4 volte al giorno
Operazione 3	3 volte al giorno
Operazione 4	5 volte al giorno
Operazione 5	4 volte al giorno
Operazione 6	5 volte al giorno

Analisi delle ridondanze:

Tavola degli accessi con attributo ridondante Operazione 6			
<u>Concetto</u>	<u>Costrutto</u>	<u>Accesso</u>	<u>Tipo</u>
Cliente	E	1	L
Dipendente	E	1	L
Assicurazione	E	1	L
Casa	E	1	L
RCAuto	E	1	L
Richiesta	R	1	S
Stipulazione	R	1	S
Riferimento	R	1	S
Polizza	E	1	S

Tavola degli accessi senza attributo ridondante Operazione 6			
<u>Concetto</u>	<u>Costrutto</u>	<u>Accesso</u>	<u>Tipo</u>
Cliente	E	1	L
Dipendente	E	1	L
Casa	E	1	L
RCAuto	E	1	L
Richiesta	R	1	S
Stipulazione	R	1	S
Riferimento	R	1	S
Polizza	E	1	S

- Viene eliminato l'attributo che identifica il tipo di Assicurazione in Polizza e viene diviso in due, seguendo la scelta successiva di eliminare la generalizzazione Assicurazione – RCAuto – Casa in quanto aumenterebbe il tempo di esecuzione, come si può notare dalle tavole degli accessi sopra si riduce il numero di accessi con una conseguente velocizzazione delle operazioni
 - Con ridondanza:
Si prevede un accesso in lettura per l'entità Cliente per ottenere i dati, uno in lettura per Dipendente per estrarre i dati, uno per Assicurazione per ottenere il tipo (e quindi due in lettura, uno per Casa e uno per RCAuto per ottenere il tipo corretto di Assicurazione). Successivamente a queste letture si prosegue con gli accessi in scrittura in Richiesta per inserire i dati del Cliente, per Stipulazione per i dati del Dipendente, per Riferimento per il tipo di assicurazione e infine in Polizza per inserire la nuova polizza. Considerando che l'operazione 6 viene svolta 5 volte al giorno si hanno quindi 25 accessi in lettura e 40 accessi in scrittura al giorno.
 - Senza ridondanza:
In questo caso eliminando l'entità Assicurazione, viene eliminato un accesso in lettura in quanto verranno letti solo le entità RCAuto e Casa e quindi l'operazione 6 richiederà 20 accessi in lettura e sempre 40 accessi in scrittura al giorno.

Eliminazione delle generalizzazioni:

- La generalizzazione Utente – Cliente – Dipendente viene sostituita spostando gli attributi dell'entità genitore nelle entità figlie. Viene considerata questa scelta dal momento che per le entità figlie ci sono degli attributi che le distinguono in modo preciso l'una dall'altra. In questo modo si riduce la possibilità di avere valori nulli nell'entità genitore.
- La generalizzazione Dipendente – Agente – SubAgente – Produttore viene trasformata con tre associazioni uno a uno che legano rispettivamente l'entità genitore Dipendente con le tre entità figlie. Viene considerata questa alternativa in quanto ci sono attributi (o associazioni nel caso di Agente) che distinguono l'una dall'altra, tuttavia l'entità genitore dovrà partecipare in ogni caso ad una occorrenza delle tre date dalle entità figlie.
- La generalizzazione Assicurazione – RCAuto – Casa viene sostituita spostando gli attributi dell'entità genitore nelle entità figlie. Viene considerata questa opzione in quanto per le entità figlie ci sono degli attributi che le distinguono in modo preciso l'una dall'altra. In questo modo si riduce la possibilità di avere valori nulli nell'entità genitore.

Schema Relazionale:

→ **Dipendente** (CodiceFiscale, Password, Nome, Cognome, Provvigione, Agenzia)

Vincolo di integrità:

- l'attributo Agenzia in Dipendente e la chiave primaria Codice in Agenzia;

La generalizzazione è stata modificata in una relazione con i figli:

→ **Agente** (CFDip)

Vincolo di integrità referenziale:

- l'attributo CFDip in Agente e la chiave primaria CodiceFiscale in Dipendente;

→ **Produttore** (CFDip, DurataContratto)

Vincolo di integrità:

- l'attributo CFDip in Produttore e la chiave primaria CodiceFiscale in Dipendente;

→ **SubAgente** (CFDip, SedeUfficio)

Vincolo di integrità:

- l'attributo CFDip in Subagente e la chiave primaria CodiceFiscale in Dipendente;

→ **Agenzia** (Codice, CittàSede, ProvinciaSede, AgenteGenerale)

Vincolo di integrità:

- l'attributo AgenteGenerale in Agenzia e CFDip in Agente;

→ **Casa** (Codice, Durata, PrezzoAnnuo, AnnoCostruzione, AnnoRistrutturazione, SpeseLegali, Tipo,)

➔ **RCAuto** (Codice, Durata, PrezzoAnnuo, Potenza, Cilindrata)

➔ **Cliente** (CodiceFiscale, Password, Nome, Cognome, DataNascita, CittaResidenza, ProvinciaResidenza, DataPatente, AnniPatente, ClasseDiMerito)

➔ **Polizza** (Numero, Massimale, DataStipula, DataScadenza, CFCliente, CodCasa, CodRC, CFDipendente)

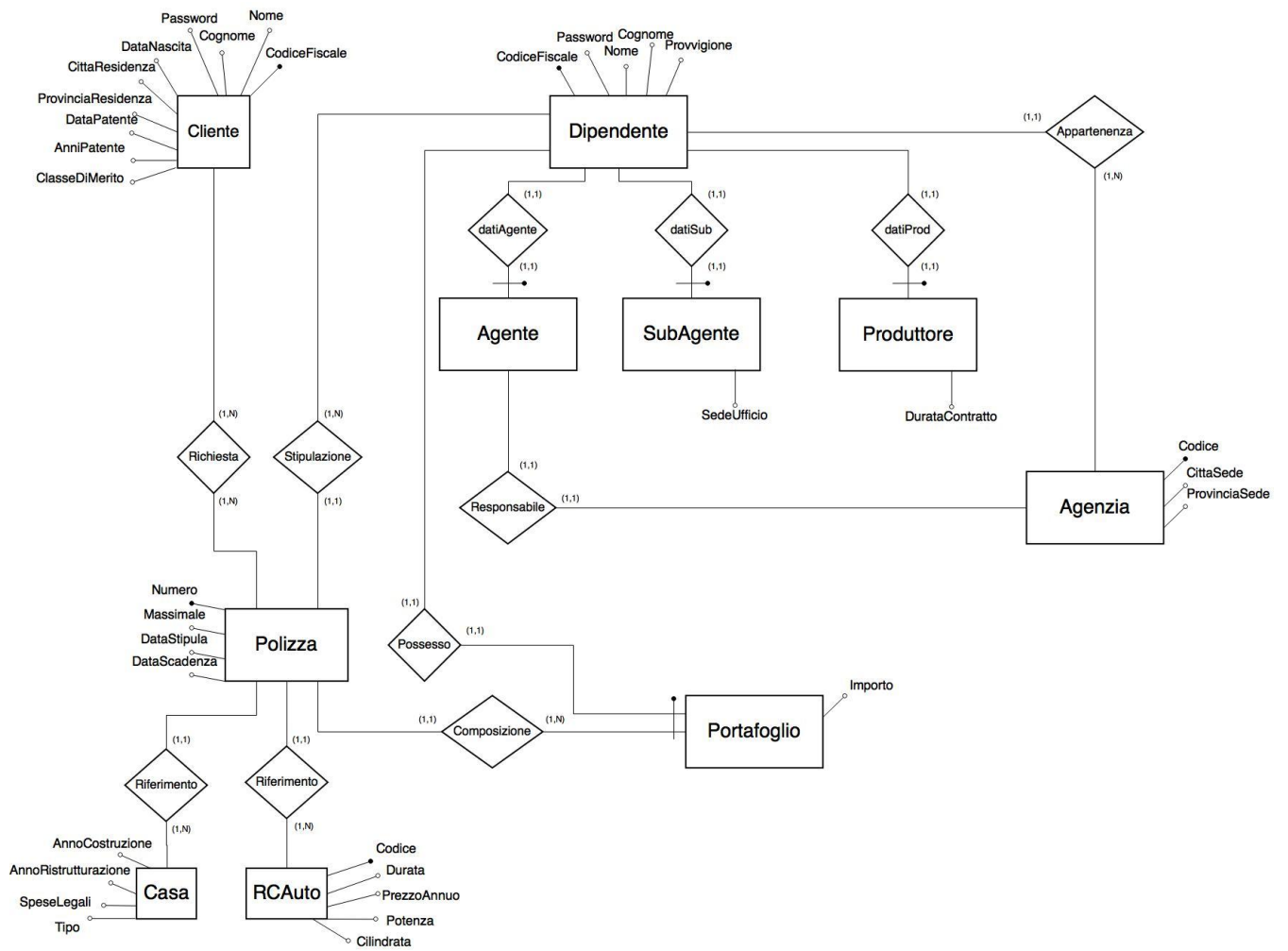
Vincoli di integrità:

- l'attributo CFCliente in Polizza e la chiave primaria CodiceFiscale in Cliente;
- l'attributo CodCasa in Polizza e la chiave primaria Codice in Casa;
- l'attributo CodRC in Polizza e la chiave primaria Codice in RCAuto;
- l'attributo CFDipendente in Polizza e la chiave primaria CodiceFiscale in Dipendente;

➔ **Portafoglio** (NPolizza, CFDip, Importo)

Vincoli di integrità:

- l'attributo NPolizza in Portafoglio e la chiave primaria Numero in Polizza;
- l'attributo CFDip in Portafoglio e la chiave primaria CodiceFiscale in Dipendente



Query, Funzioni e Trigger

Query:

- ➔ Query 1: visualizza il portafoglio dell'Agente con CodiceFiscale DGILMN70A01A479Z. Per semplificare questa query è stata creata una vista.

Codice Vista:

```
CREATE VIEW PortafoglioAgenzie AS
SELECT Npolizza, CFDip, C.Nome AS Nome, C.Cognome AS
      Cognome, Importo, D.Agenzia AS Agenzia
FROM ((Portafoglio JOIN Polizza ON Npolizza=Numero)
      JOIN Cliente C ON CFCliente=C.CodiceFiscale)
      JOIN Dipendente D ON CFDip=D.CodiceFiscale
```

Codice Query:

```
SELECT NPolizza, CFDip, Nome, Cognome, Importo
FROM (Portafoglio JOIN Polizza ON Npolizza=Numero)
      JOIN Cliente ON CFCliente=CodiceFiscale
WHERE CFDip='DGILMN70A01A479Z'
UNION
SELECT Npolizza, CFDip, Nome, Cognome, Importo
FROM PortafoglioAgenzie
WHERE Agenzia=(
      SELECT Codice
      FROM Agenzia
      WHERE AgenteGenerale='DGILMN70A01A479Z'
)
```

Risultato Query:

NPolizza	CFDip	Nome	Cognome	Importo
4	DGILMN70A01A479Z	Simone	Barco	800
5	DGILMN70A01A479Z	Paolo	Rossi	300
1	MNGVNI77E10C111D	Simone	Barco	210

➔ Query 2: trovare tutte le polizze sulla Casa con Città che comincia per C e PrezzoAnnuo maggiore di 300.

Codice Query:

```
SELECT Numero, Massimale, DataStipula, DataScadenza,
       Durata, PrezzoAnnuo, AnnoCostruzione,
       AnnoRistrutturazione, SpeseLegali, Tipo,
       CittaResidenza
FROM (Polizza JOIN Casa ON CodCasa=Codice)
     JOIN Cliente ON CFCliente=CodiceFiscale
WHERE CittaResidenza LIKE 'C%' AND PrezzoAnnuo>300
```

Num.	Massimale	Data Stipula	Data Scadenza	Durata	Prezzo Annuo	Anno Costruz.	Anno Ristrutturaz.	Spese Legali	Tipo	Citta Residenza
4	10000	2015-08-19	2016-08-19	180	800	1980	2010	7000	Furto	Camposampiero
8	9000	2015-04-20	2016-04-20	180	500	1993	2009	8000	Furto	Castelfranco V.to

➔ Query 3: visualizzare il totale delle entrate delle polizze RCAuto per ogni Dipendente.

Codice Query:

```
SELECT D.Nome, D.Cognome, SUM(Importo) AS TotaleEntrate
FROM (PortafoglioAgenzie JOIN Polizza ON Npolizza=Numero)
     JOIN Dipendente D ON CFDip=D.CodiceFiscale
WHERE CodRC IN (
                SELECT Codice
                FROM RCAuto
                )
GROUP BY D.Nome
```

Nome	Cognome	TotaleEntrate
Limone	Diego	600
Meneghini	Ivano	385

➔ Query 4: visualizza le polizze RCAuto in scadenza negli anni 2015 al 2017.

Codice Query:

```
SELECT Numero, DataStipula, DataScadenza, C.Nome,
        C.Cognome, C.AnniPatente,
C.ClasseDiMerito, RC.Potenza, RC.Cilindrata
FROM (Polizza JOIN Cliente C ON
        CFCliente=CodiceFiscale) JOIN RCAuto RC
        ON CodRC= RC.Codice
WHERE YEAR(DataScadenza) >=2015 AND
        YEAR(DataScadenza)<= 2017
```

Num.	DataStipula	DataScadenza	Nome	Cognome	AnniPatente	Classe	Potenza	Cilindrata
1	2015-08-22	2016-08-22	Simone	Barco	4	12	55	1400
5	2015-08-20	2016-08-20	Paolo	Rossi	25	14	100	2000
10	2014-09-12	2015-09-12	Pasticcia	Ciccio	15	14	70	1600

➔ Query 5: visualizzare il Nome e Cognome del Dipendente con TotaleEntrate massimo. Per semplificare questa query è stata creata una vista.

Codice Vista:

```
CREATE VIEW TotEntrate AS
SELECT DISTINCT Nome, Cognome, SUM(Importo)AS
TotaleEntrate
from (Portafoglio JOIN Dipendente ON CFDip=CodiceFiscale)
GROUP BY Nome
ORDER BY TotaleEntrate desc
```

Codice Query:

```
SELECT Nome, Cognome, MAX(TotaleEntrate) AS TotaleMassimo
FROM TotEntrate
```

Nome	Cognome	TotaleMassimo
Limone	Diego	1100

- Query 6: visualizzare Nome, Cognome, AnniPatente e Classe di merito del Cliente che ha il Nome che comincia con 'S', Cognome che comincia con 'B' e Classe di merito maggiore o uguale a 5.

Codice Query:

```
SELECT Nome, Cognome, AnniPatente, ClasseDiMerito
FROM Cliente
WHERE Nome LIKE 'S%' AND Cognome LIKE 'B%' AND
      ClasseDiMerito >= 5
```

Nome	Cognome	AnniPatente	ClasseDiMerito
Simone	Barco	4	12

- Funzione 1: procedura per inserire nella tabella Portafoglio una Polizza RCAuto *Npol* con l'Importo calcolato in base alla Provvigione del Dipendente *dip*.

Codice Funzione:

```
DROP PROCEDURE IF EXISTS provaRC;
DELIMITER //
CREATE PROCEDURE provaRC (INOUT Npol INT(5), INOUT dip
VARCHAR(16))
BEGIN
    DECLARE ImportoTemp INT(5);
    DECLARE ImportoFin INT(5);
    DECLARE ProvTemp INT(3);

    SELECT PrezzoAnnuo INTO ImportoTemp
    FROM Polizza JOIN RCAuto ON CodRC=Codice
    WHERE Numero= Npol;

    SELECT Provvigione INTO ProvTemp
    FROM Dipendente
    WHERE CodiceFiscale=dip;

    SET ImportoFin=ImportoTemp*(ProvTemp/100);

    INSERT INTO Portafoglio (NPolizza, CFDip, Importo)
    VALUES (Npol, dip, ImportoFin);
END //
```

➔ Funzione 2: funzione per calcolare gli anni di patente di un Cliente.

Codice Funzione:

```
CREATE FUNCTION AnniPatente(CFClie VARCHAR(16)) RETURNS
int(11)
BEGIN
    DECLARE Anni INT;
    SELECT (YEAR(CURRENT_DATE)-YEAR(DataPatente)) INTO
Anni
    FROM Cliente
    WHERE CodiceFiscale= CFClie;
    RETURN Anni;
END
```

➔ Funzione 3: procedura per inserire nella tabella Portafoglio una Polizza Casa *Npol* con l'Importo calcolato in base alla Provvigione del Dipendente *dip*.

Codice Funzione:

```
DROP PROCEDURE IF EXISTS provaCasa;
DELIMITER //
CREATE PROCEDURE provaCasa (INOUT Npol INT(5), INOUT dip
VARCHAR(16))
BEGIN
    DECLARE ImportoTemp INT(5);
    DECLARE ImportoFin INT(5);
    DECLARE ProvTemp INT(5);

    SELECT PrezzoAnnuo INTO ImportoTemp
    FROM Polizza JOIN Casa ON CodCasa=Codice
    WHERE Numero= Npol;

    SELECT Provvigione INTO ProvTemp
    FROM Dipendente
    WHERE CodiceFiscale=dip;

    SET ImportoFin=ImportoTemp*(ProvTemp/100);

    INSERT INTO Portafoglio (NPolizza, CFDip, Importo)
    VALUES (Npol, dip, ImportoFin);
END //
```

- ➔ Funzione 4: procedura per il rinnovo delle polizze. Aggiorna la DataStipula alla data odierna e la DataScadenza alla data odierna più un anno. Inoltre se la Polizza è RCAuto diminuisce di uno la Classe di merito del Cliente fino ad arrivare ad 1 (classe massima).

Codice Funzione:

```
DROP PROCEDURE IF EXISTS rinnovo
DELIMITER //
CREATE PROCEDURE rinnovo (IN Npol INT(5))
BEGIN
    DECLARE NDataS DATE;
    DECLARE NDataSc DATE;
    DECLARE Num INT(5);
    DECLARE cli VARCHAR(16);
    DECLARE anni INT(2);

    UPDATE Polizza SET DataStipula= CURDATE(),
    DataScadenza= DATE_ADD(CURDATE(),INTERVAL 1 YEAR)
    WHERE Numero= Npol;

    SELECT CodRC INTO Num
    FROM Polizza
    WHERE Numero=Npol;

    IF Num IS NOT NULL THEN
        SELECT CFCLiente INTO cli
        FROM Polizza
        WHERE Numero=Npol;

        SELECT AnniPatente INTO anni
        FROM Cliente
        WHERE CodiceFiscale=cli;

        IF anni > 1 THEN
            UPDATE Cliente
            SET ClasseDiMerito= ClasseDiMerito-1
            WHERE CodiceFiscale=cli;
        END IF;
    END IF;
END //
```

- ➔ Funzione 5: procedura per inserire la Provvigione a 30 per il Produttore con CodiceFiscale *dip*.

Codice Funzione:

```
DROP PROCEDURE IF EXISTS provaProd;
DELIMITER //
CREATE PROCEDURE provaProd (IN dip VARCHAR(16))
BEGIN
    DECLARE provvi INT(3);

    SET provvi=30;

    UPDATE Dipendente
    SET Provvigione=provvi
    WHERE CodiceFiscale= dip;
END //
```

- ➔ Funzione 6: procedura per inserire la Provvigione a 70 per il SubAgente con CodiceFiscale *dip*.

Codice Funzione:

```
DROP PROCEDURE IF EXISTS provaSub;
DELIMITER //
CREATE PROCEDURE provaSub (IN dip VARCHAR(16))
BEGIN
    DECLARE provvi INT(3);

    SET provvi=70;

    UPDATE Dipendente
    SET Provvigione=provvi
    WHERE CodiceFiscale= dip;
END //
```

- ➔ Trigger 1: questo trigger permette di inserire una Polizza solo se uno dei due campi CodRC e CodCasa ha un valore.

Codice Trigger:

```
DROP TRIGGER IF EXISTS provaPolizza;
DELIMITER |
CREATE TRIGGER provaPolizza
BEFORE INSERT ON Polizza
FOR EACH ROW
BEGIN
    IF (!(NEW.CodCasa IS NOT NULL XOR NEW.CodRC IS NOT
        NULL)) THEN
        INSERT INTO Polizza
        VALUES (NEW.Numero, NEW.Massimale,
            NEW.DataStipula, NEW.DataScadenza,
            NEW.CFCliente,NEW.CodCasa,
NEW.CodRC, NEW.CFDipendente);
    END IF;
END |
DELIMITER ;
```

- ➔ Trigger 2: trigger che si attiva dopo aver inserito un nuovo SubAgente. Richiama la funzione *provaSub* che aggiorna la Provvigione del nuovo SubAgente a 70.

Codice Trigger:

```
DROP TRIGGER IF EXISTS triggerSub;
DELIMITER |
CREATE TRIGGER triggerSub
AFTER INSERT ON SubAgente
FOR EACH ROW
BEGIN
    SET @dip= NEW.CFdip;

    CALL provaSub(@dip);
END |
DELIMITER ;
```

- ➔ Trigger 3: questo trigger permette, dopo aver fatto l'inserimento di una nuova Polizza, l'aggiunta di una riga nella tabella Portafoglio con l'Importo calcolato tramite la funzione *provaRC* o *provaCasa*, controllando quale dei due campi (CodRC o CodCasa) abbia un valore.

Codice Trigger:

```
DROP TRIGGER IF EXISTS provaPortafoglio;
DELIMITER |
CREATE TRIGGER provaPortafoglio
AFTER INSERT ON Polizza
FOR EACH ROW
BEGIN
    DECLARE CFdip VARCHAR(16);
    DECLARE RC INT(4);
    DECLARE Casa INT(4);

    SELECT CFDipendente INTO CFdip
    FROM Polizza
    WHERE Numero=NEW.Numero;

    SELECT CodRC INTO RC
    FROM Polizza
    WHERE Numero=NEW.Numero;

    SELECT CodCasa INTO Casa
    FROM Polizza
    WHERE Numero=NEW.Numero;

    IF RC IS NULL THEN
        SET @pol= NEW.Numero;
        SET @dip= CFdip;
        CALL provaCasa(@pol, @dip);
    ELSEIF RC IS NOT NULL THEN
        SET @pol= NEW.Numero;
        SET @dip= CFdip;
        CALL provaRC(@pol, @dip);
    END IF;
END |
DELIMITER ;
```

- ➔ Trigger 4: trigger che si attiva dopo aver inserito un nuovo Produttore. Richiama la funzione *provaProd* che aggiorna la Provvigione del nuovo Produttore a 30.

Codice Trigger:

```
DROP TRIGGER IF EXISTS triggerProd;
DELIMITER |
CREATE TRIGGER triggerProd
AFTER INSERT ON Produttore
FOR EACH ROW
BEGIN
    SET @dip= NEW.CFdip;

    CALL provaProd(@dip);
END |
DELIMITER ;
```

Interfaccia Web

L'interfaccia web si presenta contenuta all'interno di un sito di presentazione dell'agenzia assicurativa.

La possibilità di accedere alle varie funzionalità della base di dati si ottiene passando l'autenticazione della pagina *login.php*.

L'autenticazione è realizzata tramite la memorizzazione in variabili di sessione del codice fiscale e del tipo di login (cliente, agente, subagente o produttore) dopo aver controllato nella base di dati se la password utilizzata è corretta per l'utente che vuole accedere alla pagina.

Ogni pagina controlla il tipo di login e in base a questo visualizzerà le varie informazioni, altrimenti restituirà una pagina di errore.

Autenticazione come Cliente

Un utente autenticato come Cliente, potrà accedere alla pagina *gest_cliente.php*, nella quale verranno visualizzati i dati anagrafici del cliente e la possibilità di modificare la residenza attraverso la pagina

mod_cli.php. Infine sarà possibile visualizzare le polizze attive del cliente autenticato attraverso la pagina *polizzeCliente.php*

Autenticazione come Agente

Un utente autenticato come Agente, potrà accedere alla pagina *gest_dip.php*, nella quale verranno visualizzati i dati del Dipendente e la possibilità di aggiungere una nuova Polizza attraverso la pagina *new_pol.php*, aggiungere un nuovo SubAgente e un nuovo Produttore alla propria agenzia utilizzando, rispettivamente, le pagine *new_sub.php* e *new_prod.php*. E' possibile inoltre aggiungere un nuovo Cliente attraverso l'utilizzo della pagina *new_cli.php*, visualizzare il Portafoglio attraverso la pagina *port.php* e il rinnovo polizze utilizzando la pagina *rinn.php*.

Autenticazione come SubAgente

Un utente autenticato come SubAgente, potrà accedere alla pagina *gest_dip.php*, nella quale verranno visualizzati i dati del Dipendente e la possibilità di aggiungere una nuova Polizza attraverso la pagina *new_pol.php*, modificare la Sede del suo ufficio utilizzando la pagina *mod.php*.

E' possibile inoltre aggiungere un nuovo Cliente attraverso l'utilizzo della pagina *new_cli.php*, visualizzare il Portafoglio attraverso la pagina *port.php* e il rinnovo polizze utilizzando la pagina *rinn.php*.

Autenticazione come Produttore

Un utente autenticato come Produttore, potrà accedere alla pagina *gest_dip.php*, nella quale verranno visualizzati i dati del Dipendente e la possibilità di aggiungere una nuova Polizza attraverso la pagina *new_pol.php*, aggiungere un nuovo Cliente utilizzando la pagina *new_cli.php* e visualizzare il Portafoglio attraverso l'utilizzo della pagina *port.php*. Inoltre è possibile rinnovare le polizze attraverso la pagina *rinn.php*.

Interfaccia comune

Tutte le pagine presentano un menù di navigazione principale (a sinistra), chiamato da *library.php* che permette di orientarsi tra la homepage, la pagina di login/logout e la pagina di gestione dipendente se si è Agente, SubAgente o Produttore, se si è Cliente invece si potrà navigare nella homepage, nella pagina di login/logout e nella pagina di gestione cliente. Nelle pagine di gestione è presente un secondo menù di navigazione (orizzontale) che permette di orientarsi tra le funzionalità.

I passaggi dei parametri tra le pagine vengono realizzati tramite variabili di sessione per l'autenticazione, che sblocca/blocca le varie funzionalità, mentre per gli ulteriori parametri vengono utilizzati sia il metodo POST che GET.

Nel caso di passaggio tramite form è stato preferito il metodo POST, mentre nelle pagine richiamate da semplici link è stato utilizzato il metodo GET.

Per semplicità indico quattro utenti con cui è possibile testare l'interfaccia web:

Cliente 1:

- Codice Fiscale: BRCSMN93C23C111I
- Password: 00

Cliente 2:

- Codice Fiscale: RSSPLA70A01C111A
- Password: 0

Agente:

- Codice Fiscale: DGILMN70A01A479Z
- Password: 12

SubAgente:

- Codice Fiscale: MNGVNI77E10C111D
- Password: 11

Produttore:

- Codice Fiscale: MRLMHL90D15A346R
- Password: 123