



UNIVERSITA' DEGLI STUDI DI FERRARA

Facoltà di Ingegneria

Progetto: Database per assicurazione rc auto

Insegnamento : Basi di Dati

Studente:

Simona Finotti

Docente:

Ing. Denis Ferraretti

Anno Accademico 2012/2013

Indice :

Capitolo 1: Descrizione del “mini-mondo”	2
1.1 Descrizione entità.....	3
1.2 Descrizione relazioni.....	8
Capitolo 2: Costruzione modello Entity-Relationship	9
Capitolo 3: Trasformazione da modello ER a modello relazionale	10
Capitolo 4: Normalizzazione del modello relazionale	14
Capitolo 5: Codice SQL	19
Capitolo 6: Vincoli di check, asserzioni, trigger e viste	23
6.1 Vincoli di check.....	23
6.2 Asserzioni e trigger.....	24
6.3 Viste.....	25
Capitolo 7: Interrogazioni SQL	27
Capitolo 8: Interfaccia grafica	28

Capitolo 1:

Descrizione del “mini-mondo”

Lo scopo del progetto è lo sviluppo di una base di dati per gestire le informazioni relative alle polizze auto (e/o altri mezzi) di una società di assicurazione. In particolare, si dovranno memorizzare le informazioni sui clienti assicurati (codice fiscale, nome, cognome, genere, data di nascita, indirizzo, recapito telefonico, indirizzo e-mail, anno in cui si è conseguita la patente, classe di merito assicurativa).

Nel caso in cui l'assicurato sia una società invece di una persona fisica si vogliono mantenere informazioni sul nome della società, referente per la società, indirizzo, recapito telefonico, email, partita IVA. Inoltre, è utile preservare alcuni dati di carattere generale sulle tipologie di polizze offerte (nome, tipologia di polizza, massimale e costo). Ogni cliente può sottoscrivere una o più polizze. Per ogni polizza sottoscritta da un cliente si vuole memorizzare la data in cui è stata stipulata, la data di scadenza della rata ed il mezzo per cui è stata stipulata, oltre al tipo di frazionamento scelto per il pagamento (annuale o semestrale) ed al suo importo. Si vuole anche registrare l'agente assicurativo con cui è stata sottoscritta la polizza (matricola, nome, cognome, telefono). Nel caso di mezzo assicurato da una società è invece sempre necessario dichiarare alcune generalità sulle persone autorizzate a guidare i veicoli aziendali; si dovranno registrare codice fiscale, nome, cognome, indirizzo, recapito telefonico relativo ad esse. Ogni pratica contiene informazioni sulla tipologia di polizza stipulata (si suppone che esistano dei pacchetti assicurativi fra i quali potere scegliere). In particolare ogni cliente potrà scegliere un pacchetto assicurativo fra quelli offerti per la categoria del mezzo da assicurare. Ogni mezzo è identificato da un numero di targa (univoco), un tipo (auto, moto, veicolo commerciale), marca, modello, potenza (kW) e dall'anno di immatricolazione.

1.1 Descrizione delle entità:

Una volta analizzato accuratamente il “mini-mondo” per il quale vogliamo costruire il database, passiamo ora a identificare e descrivere le singole *entità* che rappresentano classi di oggetti (fatti, cose, persone, ...) che hanno proprietà comuni e le *relazioni*, ovvero i legami che intercorrono tra più entità. Tale processo ci permetterà poi di costruire un modello concettuale (modello E-R), utile ai fini della corretta progettazione della base di dati.

Entità MEZZO:

Non avendo a disposizione un database della motorizzazione civile, contenente tutte le informazioni delle auto immatricolate, ne simuleremo il funzionamento immagazzinando le sole informazioni delle auto assicurate presso la nostra agenzia.

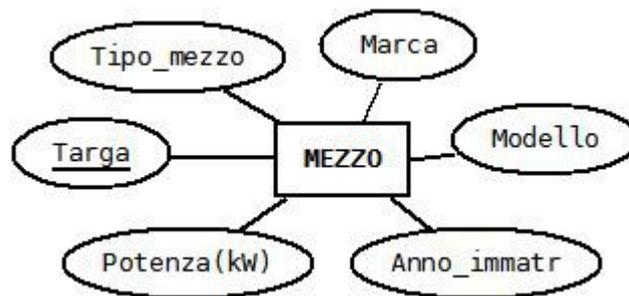


Figura 1.1: entità mezzo

Attributo chiave Targa: la targa identifica univocamente ciascun veicolo;

Attributo Tipo_mezzo: identifica il tipo di mezzo, può assumere i valori “auto”, “moto” o “commerciale”;

Attributo Marca: identifica la ditta costruttrice del veicolo (es: fiat, ford, ducati) ;

Attributo Modello: identifica il modello del veicolo (es: panda, fiesta, monster);

Attributo Potenza(KW): identifica la potenza del veicolo espressa in KW ;

Attributo Anno_immatr: identifica l'anno di immatricolazione del veicolo.

Entità ASSICURATORE:

Descrive i dati di interesse per gli Assicuratori che lavorano per la nostra agenzia.

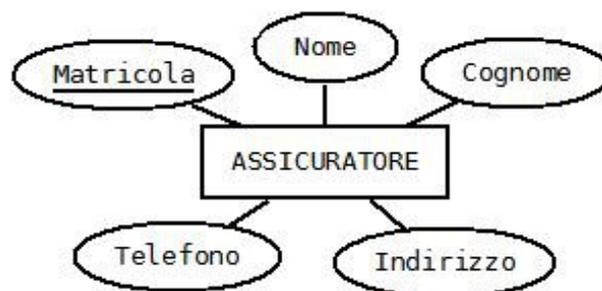


Figura 1.2: entità assicuratore

Attributo chiave Matricola: la matricola è un codice alfanumerico associato a ciascun assicuratore, che lo identifica univocamente per l'assicurazione di cui è dipendente;

- Attributo Cognome:* cognome dell'assicuratore;
- Attributo Nome:* nome dell'assicuratore;
- Attributo Indirizzo:* indirizzo di residenza dell'assicuratore ;
- Attributo Telefono:* recapito telefonico dell'assicuratore(formato da 10 cifre);

Entità CLIENTE_PERSONA:

Contiene i dati che caratterizzano le persone fisiche, che hanno contratto una polizza presso la nostra agenzia assicurativa.

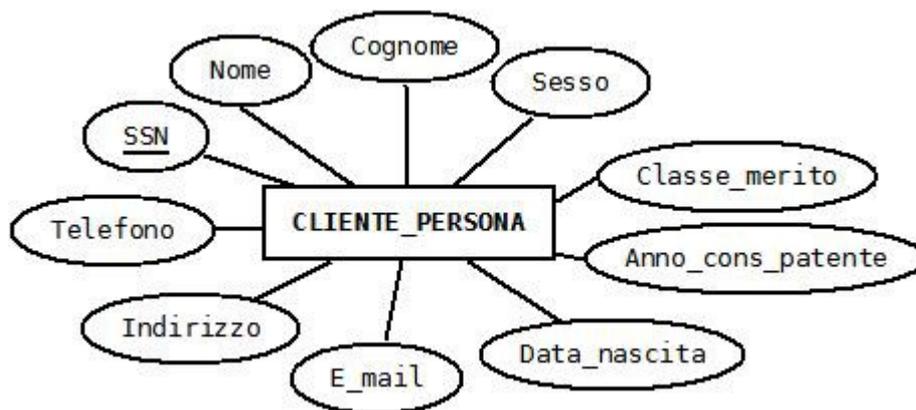


Figura 1.3: entità cliente_persona

Attributo chiave SSN: l'SSN è un codice che identifica univocamente ciascuna persona, in Italia utilizzeremo il codice fiscale ovvero un codice alfanumerico di 16 cifre che supporremo essere univoco;

- Attributo Nome:* nome del contraente della polizza;
- Attributo Cognome:* cognome del contraente della polizza ;
- Attributo Data_nascita:* data di nascita del contraente della polizza ;
- Attributo Anno_cons_patente:* anno di conseguimento della patente di guida;
- Attributo Classe_merito:* classe di merito a cui appartiene il contraente ;
- Attributo Indirizzo:* indirizzo presso cui risiede il contraente della polizza;
- Attributo Telefono:* recapito telefonico fornito del cliente (formato da 10 cifre);
- Attributo E_mail:* indirizzo di posta elettronica fornito del cliente.

Nota: La classe di merito è un punteggio numerico sulla base del quale viene individuato, secondo il meccanismo bonus malus, un premio da pagare per la polizza. I conducenti più bravi, quelli con il punteggio più basso, pagano meno. Per avere un ordine di grandezza, un assicurato con la migliore classe bonus malus può pagare addirittura la metà rispetto all'assicurato che, nelle medesime condizioni, parta dalla classe base (ovvero quella con cui ci si assicura per la prima volta). I punteggi delle classi di merito vanno da 1 (il migliore) a 18 (il peggiore). Il punteggio d'ingresso, quello che si ha quando non si ha una storia assicurativa alla spalle, al quale corrisponde il premio base, è 14. Le classi di merito vengono mantenute anche se si contrae una polizza con una società assicurativa diversa. Nel nostro caso non considereremo i sinistri commessi, per semplificare il problema in esame, tuttavia le possibili variazioni della classe di merito saranno registrabili modificando i dati del cliente e quindi sarà possibile calcolare correttamente il premio assicurativo da pagare in caso di variazioni.

Entità **CLIENTE_SOCIETA'**:

Contiene le caratteristiche delle società, che hanno contratto una polizza presso la nostra agenzia assicurativa.



Figura 1.4: entità cliente_società

Attributo chiave P.iva: la partita iva è un codice numerico di 11 cifre che identifica univocamente ciascuna società italiana;

Attributo Nome_società: Nome della società cliente dell'agenzia assicurativa;

Attributo Referente: Nome e Cognome di un possibile referente della società, potrebbe essere il titolare, l'amministratore delegato o presidente della società ;

Attributo Indirizzo: indirizzo della sede legale o amministrativa della società;

Attributo Telefono: recapito telefonico fornito dalla società (formato da 10 cifre);

Attributo E_mail: indirizzo di posta elettronica fornito dalla società.

Entità **ALTRO_CONDUCENTE:**

Contiene i dati dei conducenti assicurati da una società nostra cliente. Ciascuna società può infatti avere più dipendenti che hanno accesso ai veicoli aziendali, ed è bene conservare le loro generalità ai fini assicurativi. Dato che non potranno esistere altri conducenti assicurati se la società di appartenenza non è registrata all'interno della nostra base di dati, questa entità sarà di tipo debole.



Figura 1.5: entità altro_conducente

Attributo chiave SSN: l'SSN è un codice che identifica univocamente ciascuna persona, in Italia utilizzeremo il codice fiscale ovvero un codice alfanumerico di 16 cifre che suppremo essere univoco;

Attributo Cognome: cognome del conducente;

Attributo Nome: nome del conducente;

Attributo Telefono: recapito telefonico fornito del conducente (formato da 10 cifre);
Attributo Indirizzo: indirizzo presso cui risiede il conducente;
Attributo Società_appartenenza: attributo che identificherà la società di appartenenza del conducente, tale attributo coinciderà quindi la partiva iva della società .

Entità CLIENTE:

Dal momento che, l'entità CLIENTE_SOCIETA' e l'entità CLIENTE_PERSONA hanno degli attributi comuni, si può pensare di utilizzare una specializzazione che comprenda entrambe le tipologie di clienti e sarà così costruita:

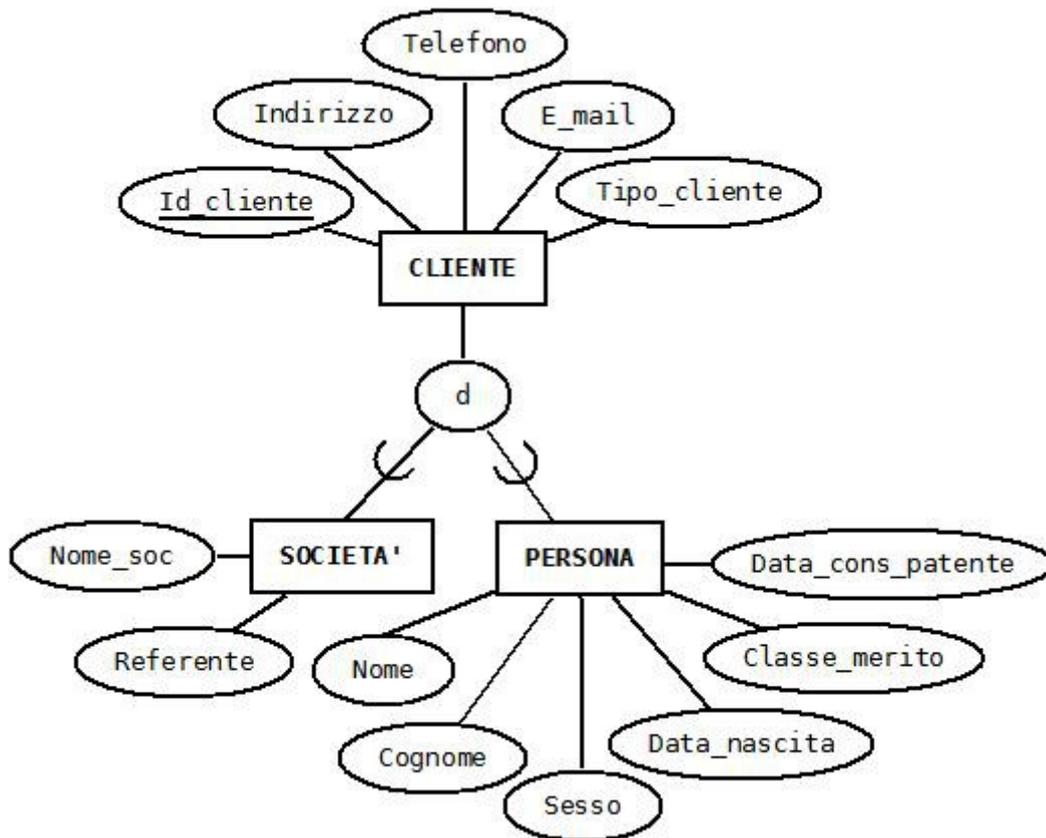


Figura 1.6: entità cliente con specializzazione

Attributo chiave Id_cliente: per la società sarà la partita iva, mentre per le persone fisiche utilizzeremo il codice fiscale.

Attributo Tipo_cliente: discrimina il tipo del cliente, potrà assumere solo il valore “persona” oppure “società”;

Attributo Indirizzo: indirizzo della sede legale o amministrativa della società, oppure del domicilio della persona fisica;

Attributo Telefono: recapito telefonico fornito dalla società o dalla persona fisica (formato da 10 cifre);

Attributo E_mail: indirizzo di posta elettronica fornito dalla società o della persona fisica;

Gli altri attributi assumono ancora il significato descritto rispettivamente nell'entità CLIENTE_PERSONA e CLIENTE_SOCIETA'.

Entità CLAUSOLA_CONTRATTUALE:

Contiene i dati relativi ai pacchetti assicurativi offerti dall'assicurazione, tra i quali i clienti potranno scegliere. Ciascun cliente potrà scegliere una sola clausola contrattuale, che identificherà il pacchetto assicurativo scelto. Potrebbero esserci delle restrizioni nella scelta dei pacchetti, ad esempio una polizza per “Guidatori esperti” può essere stipulata solo da persone fisiche che abbiano conseguito la patente da almeno 10 anni.



Figura 1.7: entità clausola_contrattuale

Attributo chiave Nome: nome del pacchetto assicurativo;

Attributo Tipologia_polizza: specifica se la polizza, è stata formulata per assicurare auto, moto oppure veicoli commerciali. E' un attributo utile nella progettazione del db, sia per i vantaggi che offrirebbe per poter riadattare facilmente il codice in caso di emissioni di nuovi pacchetti assicurativi, sia per mantenere il db in uno stato consistente (vincoli di integrità referenziale);

Attributo Massimale: calcolato in Euro. Per massimale si intende il livello massimo che può essere risarcito per i danni provocati a terze persone. Il massimale minimo che le compagnie assicurative dovevano garantire di risarcire, era equivalente a 750mila euro, per quanto stabilito dalla legge sulla R.C. auto del 1969, cifra oggi riadeguata a 5 milioni di euro;

Attributo Importo_base: calcolato in Euro. L'importo rappresenta il costo annuale della polizza, calcolato in corrispondenza della classe di merito 14. Le società assicurate non avendo una classe di merito associata, pagheranno un importo ricavato dal costo base riadattato solo in base alla potenza del veicolo .

Entità CONTRATTO:

Contiene tutte le informazioni relative alle pratiche contrattuali stipulate dai clienti.



Figura 1.8: entità contratto

Attributo chiave Num.pratica: numero progressivo che identifica il contratto stipulato con l'agenzia di assicurazione;

Attributo Clausola_contrattuale_scelta: scelta del pacchetto assicurativo effettuata dal cliente;

Attributo Contraente: identificatore del cliente che ha sottoscritto la polizza;

Attributo Venditore: identificatore dell'assicuratore che ha venduto la polizza;

Attributo Mezzo: identificatore del mezzo per cui è stato stipulato il contratto di assicurazione;

Attributo Frazionamento: discriminerà il tipo di frazionamento scelto per il pagamento, potrà assumere i valori “Semestrale” o “Annuale”;

Attributo Data_sottoscrizione: data sottoscrizione del contratto;

Attributo Data_scadenza_rata: data di scadenza della rata, dipende dalla data di sottoscrizione e dal frazionamento scelto;

Attributo Importo_da_pagare: importo da pagare calcolato per il pacchetto assicurativo scelto (clausola contrattuale);

1.2 Descrizione delle associazioni:

Per ogni entità partecipante ad una relazione viene specificata la cardinalità, cioè una coppia di numeri naturali che specifica il numero minimo e massimo di istanze di relazione a cui una istanza dell'entità può partecipare. E' possibile indicare con la costante N, un numero generico maggiore di uno, quando la cardinalità non è nota con precisione.

appartiene a cardinalità 1:N : è la relazione che intercorre fra l'entità SOCIETA' e l'entità ALTRO_CONDUCENTE. N altri conducenti assicurati possono appartenere ad 1 società, e ad 1 società possono appartenere N dipendenti. Dato che gli altri eventuali conducenti, esistono solo se esiste un' istanza dell'entità società che li identifica, ALTRO_CONDUCENTE sarà un tipo di entità debole e SOCIETA' sarà l'entità proprietaria; quindi la relazione che le lega si dirà **identificante**.

stipula cardinalità 1:N : è la relazione che lega l'entità CLIENTE e l'entità CONTRATTO infatti 1 cliente può stipulare N polizze assicurative (quindi più contratti) presso la nostra agenzia di assicurazione, mentre un CONTRATTO potrà essere associato ad un solo CLIENTE.

assicura cardinalità 1:1 : è la relazione che lega l'entità MEZZO e l'entità CONTRATTO infatti mediante un contratto assicurativo è possibile assicurare un solo mezzo, ed un solo mezzo viene assicurato da solo contratto.

contiene cardinalità 1:1 : è la relazione che lega l'entità CLAUSOLA_CONTRATTUALE e l'entità CONTRATTO, un contratto conterrà una clausola contrattuale che rappresenta uno dei pacchetti offerti dall'assicurazione.

Per prendere visione delle relazioni, fare riferimento alla fig. 2.1.

Capitolo 2:

Costruzione del modello Entity-Relationship :

Il modello E-R costituisce una rappresentazione concettuale dei dati, e ne fornisce un'astrazione di alto livello. In base alle considerazioni fatte in merito a entità e relazioni che descrivono il “mini-mondo”, il modello E-R della base di dati assicurazione responsabilità civile è il seguente:

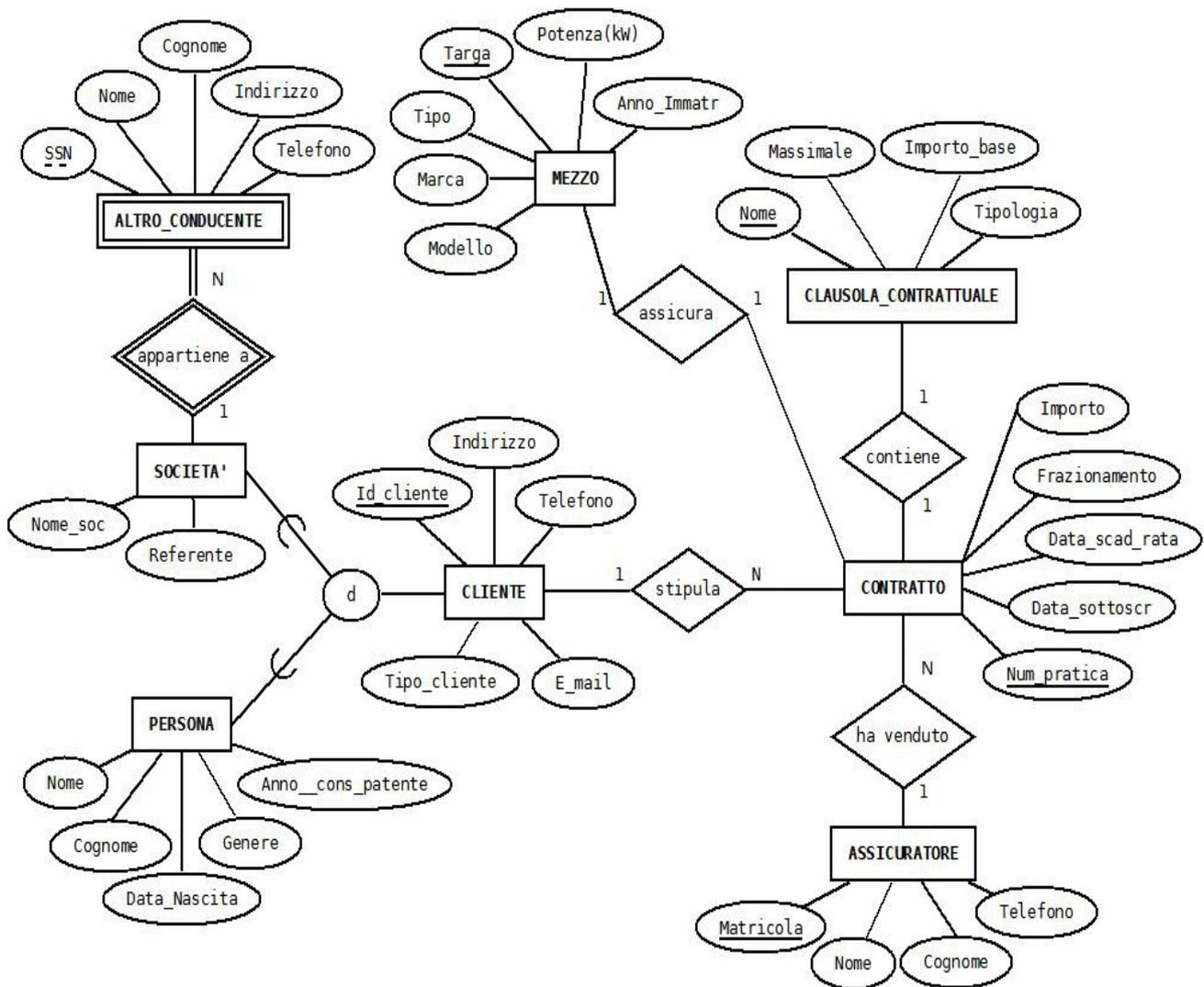


Figura 2.1: Modello E-R completo

Capitolo 3:

Trasformazione da modello ER a modello relazionale

Il modello relazionale è un modello logico di rappresentazione dei dati, per passare dal modello E-R al modello relazionale occorre seguire il seguente algoritmo:

Passo 1: traduzione di tipi di entità

Passo 2: traduzione di tipi di entità debole

Passo 3: traduzione di tipi di associazioni binarie 1:1

Passo 4: traduzione di tipi di associazioni binarie 1:N

Passo 5: traduzione di tipi di associazioni binarie M:N

Passo 6: traduzione di attributi multivalore

Passo 7: traduzione di tipi di associazione N-arie

Passo 8: traduzione della specializzazione

Applichiamo quindi, il suddetto algoritmo alla base di dati assicurazione r.c.auto.

Nota: in verde sono rappresentati gli attributi chiave delle entità,
in giallo sono rappresentate le chiavi esterne,
in azzurro sono rappresentate le chiavi deboli;

Passo 1: Traduzione dei tipi di entità

CLAUSOLA_CONTRATTUALE

NOME_CLAUSOLA	TIPOLOGIA	MASSIMALE_euro	IMPORTO_BASE_euro
---------------	-----------	----------------	-------------------

CONTRATTO

NUM_PRATICA	DATA_SOTTOSCRIZIONE	FRAZIONAMENTO	DATA_SCAD_RATA	IMPORTO_euro
-------------	---------------------	---------------	----------------	--------------

MEZZO

TARGA	MARCA	MODELLO	POTENZA_kw	ANNO_IMMATRICOLAZIONE
-------	-------	---------	------------	-----------------------

ASSICURATORE

MATRICOLA	COGNOME	NOME	INDIRIZZO	TELEFONO
-----------	---------	------	-----------	----------

CLIENTE

ID_CLIENTE	INDIRIZZO	TELEFONO	E_MAIL
------------	-----------	----------	--------

Figura 3.1: modello relazionale, traduzione tipi di entità

Passo 2: Traduzione dei tipi di entità debole

Per i tipi di entità debole occorre riportare come chiave esterna la chiave dell'entità "proprietaria" CLIENTE_ID_CLIENTE, oltre alla chiave debole SSN

ALTRO_CONDUCENTE

C_ID_CLIENTE	SSN	COGNOME	NOME	INDIRIZZO	TELEFONO
--------------	-----	---------	------	-----------	----------

Figura 3.2: modello relazionale, traduzione tipi di entità debole

Passo 3: traduzione di tipi di associazioni binarie 1:1

Sia l'entità CLAUSOLA_CONTRATTUALE che l'entità MEZZO hanno una relazione con CONTRATTO, in caso di associazioni 1:1 si può scegliere in quale delle entità riportare la chiave esterna, tuttavia appare più conveniente aggiungere le chiavi esterne a CONTRATTO.

CLAUSOLA_CONTRATTUALE

NOME_CLAUSOLA	TIPOLOGIA	MASSIMALE_euro	IMPORTO_BASE_euro
---------------	-----------	----------------	-------------------

CONTRATTO

NUM_PRATICA	DATA_SOTTOSCRIZIONE	FRAZIONAMENTO	DATA_SCAD_RATA	IMPORTO_euro	CL_NOME_CL	M_TARGA
-------------	---------------------	---------------	----------------	--------------	------------	---------

MEZZO

TARGA	MARCA	MODELLO	POTENZA_kw	ANNO_IMMATRICOLAZIONE
-------	-------	---------	------------	-----------------------

ASSICURATORE

MATRICOLA	COGNOME	NOME	INDIRIZZO	TELEFONO
-----------	---------	------	-----------	----------

CLIENTE

ID_CLIENTE	INDIRIZZO	TELEFONO	E_MAIL
------------	-----------	----------	--------

ALTRO_CONDUCENTE

C_ID_CLIENTE	SSN	COGNOME	NOME	INDIRIZZO	TELEFONO
--------------	-----	---------	------	-----------	----------

Figura 3.3: modello relazionale completo fino al passo 3, traduzione relazioni 1:1

Passo 4: traduzione di tipi di associazioni binarie 1:N

L'entità CLIENTE e l'entità ASSICURATORE hanno una relazione con CONTRATTO; in caso di associazione 1:N si riporta la chiave esterna su CONTRATTO.

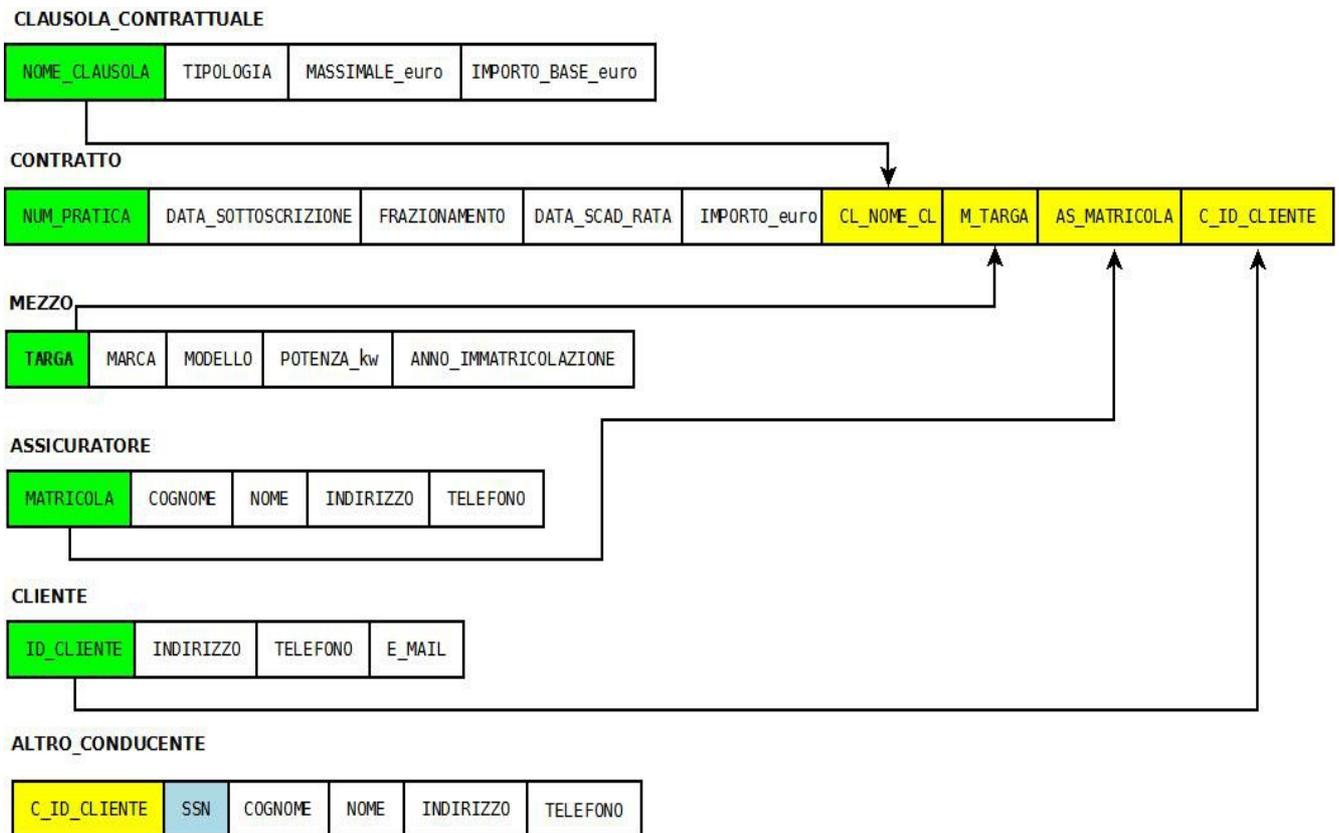


Figura 3.4: modello relazionale completo fino al passo 4, traduzione relazioni 1:N

Dato che non sono presenti associazioni di tipo N:M, attributi multivalore e tipi di associazioni n-arie, si passa direttamente al passo 8 dell'algoritmo.

Passo 8: traduzione della specializzazione



Figura 3.5: modello relazionale, traduzione della specializzazione per l'entità cliente

Schema relazionale completo:

In figura 3.6, viene riportato il modello prodotto dopo aver applicato tutti i passaggi dell'algoritmo di traduzione.

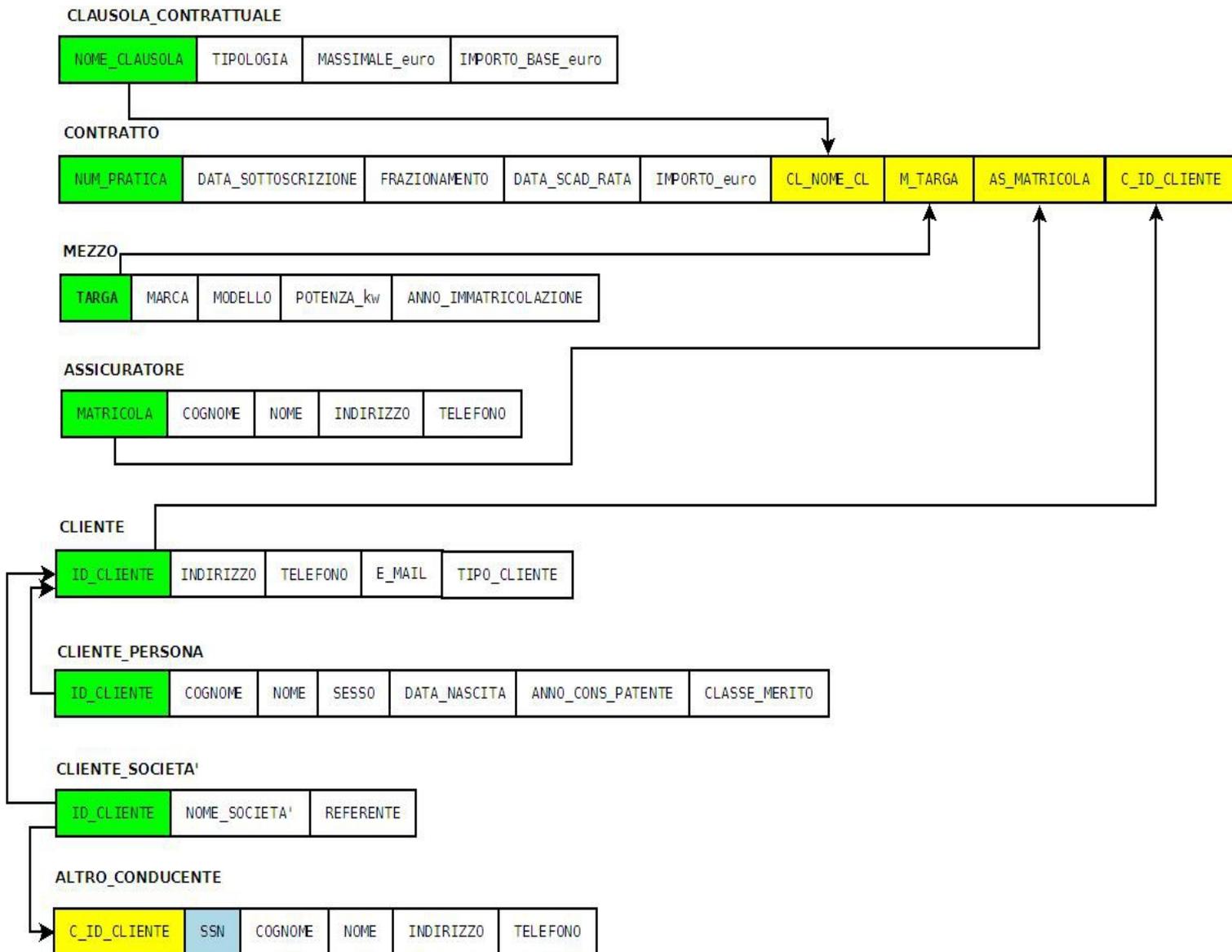


Figura 3.6: modello relazionale completo

Capitolo 4:

Normalizzazione del modello relazionale

Richiami teorici:

La normalizzazione è un procedimento volto all'eliminazione della ridondanza e del rischio di incoerenza del database. Esistono vari livelli di normalizzazione (forme normali) che certificano la qualità dello schema del database. Questo processo si fonda su un semplice criterio: se una relazione presenta più concetti tra loro indipendenti, la si decompone in relazioni più semplici.

Prima forma normale (1NF):

Richiede che il dominio di un attributo comprenda solo valori atomici e che il valore di qualsiasi attributo sia un valore singolo del dominio.

La prima forma normale è già parte integrante della definizione formale di relazione nel modello relazionale, quindi il modello relazionale visto al punto precedente è già 1NF.

Seconda forma normale (2NF):

Uno schema relazionale R è in 2NF se ogni attributo non-primario A di R dipende funzionalmente in modo completo dalla chiave primaria di R .

Nell'ambito della seconda forma normale entrano in gioco le dipendenze funzionali complete e parziali:

Una dipendenza funzionale $X \rightarrow Y$ si dice completa se la rimozione di qualsiasi attributo A da X comporta che la dipendenza funzionale non sussista più, cioè per ogni attributo $A \rightarrow X$, $(X - \{A\})$ NON determina funzionalmente Y .

Una dipendenza funzionale $X \rightarrow Y$ si dice parziale se si possono rimuovere da X certi attributi $A \rightarrow X$ e la dipendenza continua a sussistere, cioè per qualche $A \rightarrow X$, $(X - \{A\}) \rightarrow Y$.

Per il raggiungimento della 2NF occorre decomporre le relazioni nelle quali gli attributi non-primari sono associati solo alla parte della chiave primaria da cui dipendono funzionalmente in modo completo.

Terza forma normale (3NF):

Uno schema di relazione R è in 3NF se, quando sussiste in R una dipendenza funzionale non-banale $X \twoheadrightarrow A$, o X è una superchiave di R oppure A è un attributo primario di R .

Definizione alternativa:

Uno schema di relazione R è in 3NF se ogni attributo non-primario di R è funzionalmente dipendente in modo completo da ogni chiave di R e contemporaneamente non è dipendente in modo transitivo da alcuna chiave di R .

Nell'ambito della terza forma normale entrano in gioco le dipendenze funzionali transitive:

Una dipendenza funzionale $X \rightarrow Y$ si dice transitiva se esiste un insieme di attributi Z , che non è né una chiave candidata né un sottoinsieme di una chiave di R , per cui valgono contemporaneamente $X \rightarrow Z$ e $Z \rightarrow Y$.

Raggiungimento della 1NF per la base di dati assicurazione responsabilità civile:

Nel modello relazionale, se correttamente tradotto è insito anche il raggiungimento della prima forma normale.

Raggiungimento della 2NF per la base di dati assicurazione responsabilità civile:

Le considerazioni teoriche fatte precedentemente, ci portano a considerare le relazioni che possono essere ritenute problematiche. Nel nel caso in esame, sono solamente l'entità ALTRO_CONDUCENTE e l'entità CONTRATTO, dato che hanno come chiave candidata la chiave primaria e le chiavi chiavi esterne. Occorre quindi analizzare le dipendenze funzionali parziali e complete delle suddette entità:

Si può osservare che ALTRO_CONDUCENTE ha come chiavi l'SSN del conducente e C_ID_CLIENTE, che rappresenta la chiave esterna dell'entità CLIENTE.

ALTRO_CONDUCENTE_1NF

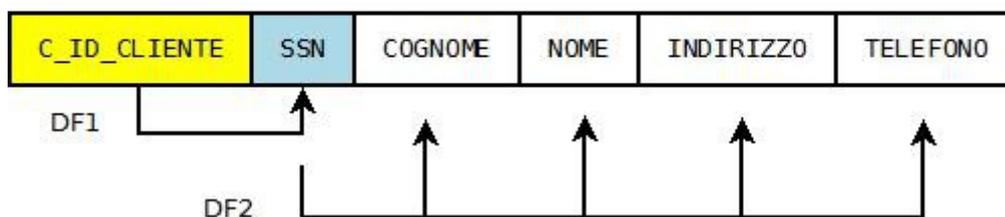
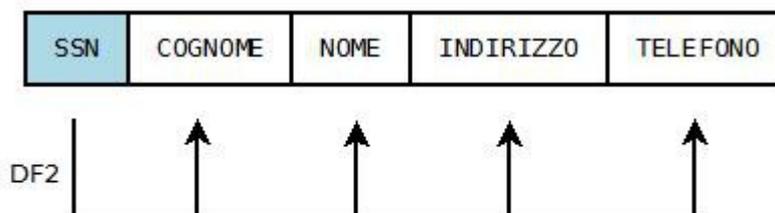


Figura 4.1: entità altro_conducente, dipendenze funzionali parziali e complete

Si nota che gli attributi non primi dipendono solo da una parte della chiave, cioè solo da SSN, quindi per arrivare in 2NF occorre decomporre nel modo seguente:

ALTRO_CONDUCENTE_2NF



CLIENTE_ALTRO_CONDUCENTE_2NF



Figura 4.2: scomposizione dell'entità altro_conducente per il raggiungimento della 2NF

Si può invece notare che CONTRATTO, contiene le sole informazioni essenziali alla registrazione di una nuova pratica contrattuale, infatti l'attributo chiave NUM_PRATICA

rappresenta la numerazione interna utilizzata dall'agenzia di assicurazione gestire più comodamente i contratti; gli unici dati registrati non dipendono solamente da parte della chiave, ma dipendono da ciascuno degli attributi che compongono la chiave candidata, infatti un nuovo contratto sarà stipulato per un mezzo, da un cliente che sceglierà una clausola contrattuale da cui senza dubbio dipenderà l'importo da pagare, quindi considerando le sole dipendenze funzionali complete o parziali non risulta necessaria la decomposizione della relazione per il raggiungimento della seconda forma normale.

CONTRATTO



Figura 4.3:entità contratto, dipendenze funzionali complete

Schema relazionale completo in seconda forma normale (2NF):

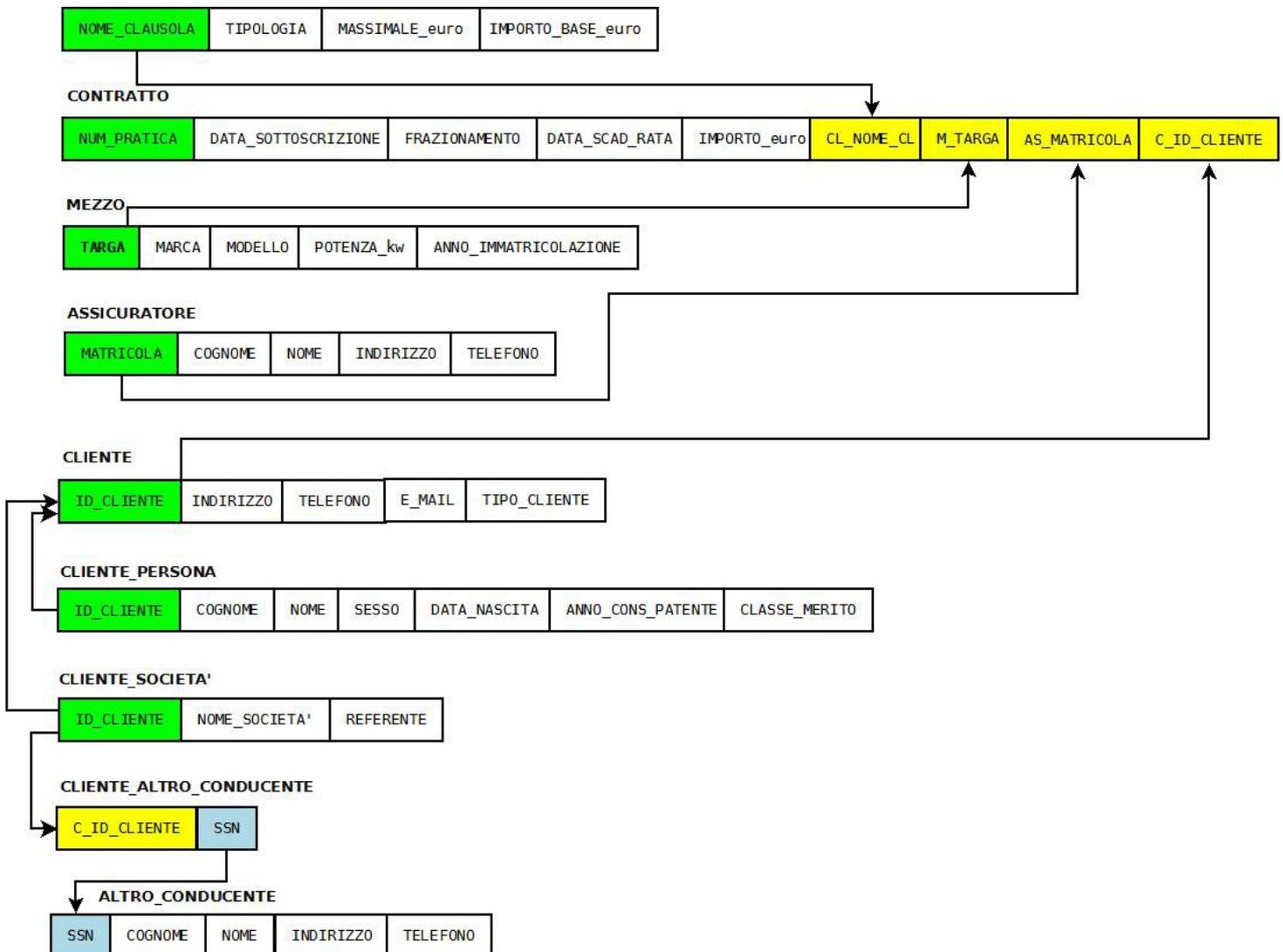


Figura 4.4:schema relazionale completo normalizzato in 2NF

Raggiungimento della 3NF per la base di dati assicurazione responsabilità civile:

A questo punto bisogna valutare le eventuali dipendenze funzionali transitive nelle relazioni. L'unica relazione che potrebbe essere decomposta è CONTRATTO, infatti analizzando gli attributi non primi si può notare che dalla data di sottoscrizione e dal frazionamento scelto dipenderà anche la data di scadenza della rata.

CONTRATTO

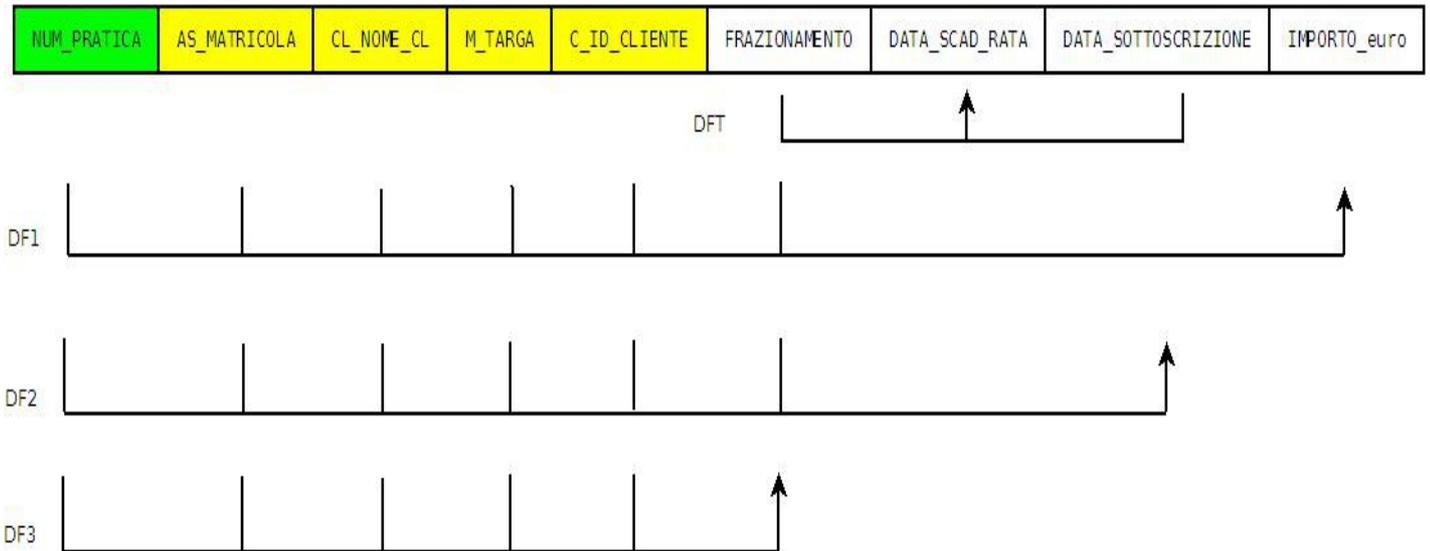


Figura 4.5: entità contratto, dipendenze funzionali complete e transitive

CONTRATTO



Figura 4.6: entità contratto, raggiungimento della terza forma normale

In questo caso particolare, in cui la data di scadenza è ricavabile svolgendo un semplice calcolo matematico a partire dai valori assunti dai campi DATA_SOTTOSCRIZIONE e FRAZIONAMENTO, possiamo pensare di eliminare l'attributo non-primario DATA_SCAD_RATA dall'entità CONTRATTO, senza decomporre ulteriormente la relazione. L'assenza di questo attributo, non limiterà la realizzazione del database e dell'interfaccia grafica in php, visto che MySql supporta il tipo di dato DATE, ed esistono innumerevoli funzioni sql e php che permettono di manipolare agevolmente le date.

Schema relazionale completo in terza forma normale (3NF):

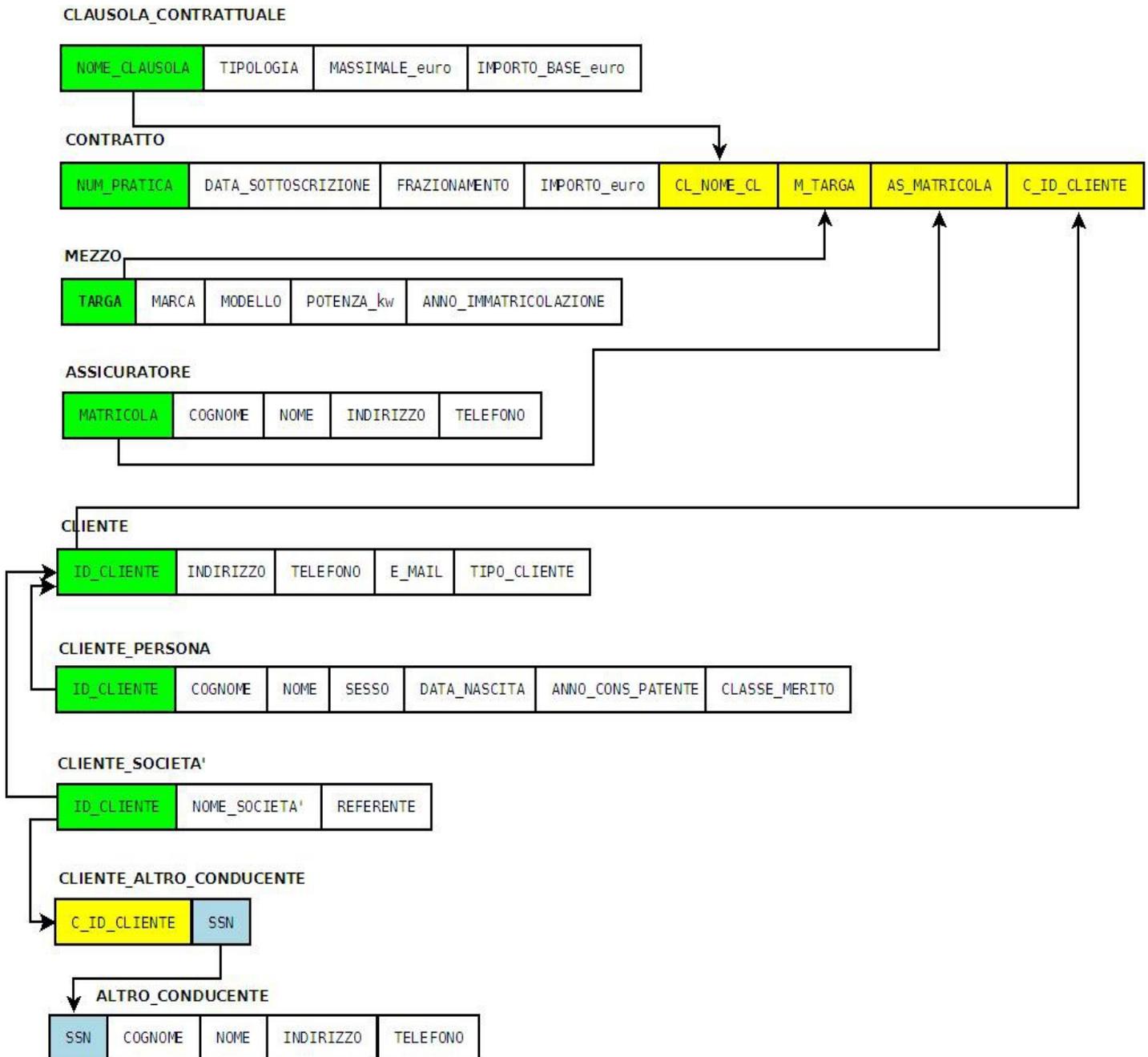


Figura 4.7: schema relazionale completo, normalizzato in 3NF

Per la realizzazione fisica del database, ho comunque deciso di utilizzare il modello relazionale in 1NF, dato che le modifiche da apportare sarebbero minime; in questo caso, anche in prima forma normale non si verificherebbe un'effettiva ridondanza di dati o al contrario l'eventuale presenza di campi nulli.

Capitolo 5:

Codice SQL

In seguito viene riportato il codice sql esportato da phpMyAdmin. Comprende i costrutti sql che permettono di creare le tabelle assicuratore, mezzo, cliente, società, altro_conducente, persona, clausola_contrattuale, contratto ed i vincoli di integrità referenziale. Per l'implementazione del modello E-R si può utilizzare MySql Workbench.

```
-- phpMyAdmin SQL Dump
-- version 3.5.1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generato il: Set 07, 2013 alle 17:10
-- Versione del server: 5.5.24-log
-- Versione PHP: 5.4.3

--
-- Database: `rca_3`
--
-----

--
-- Struttura della tabella `altro_conducente`
--

CREATE TABLE IF NOT EXISTS `altro_conducente` (
  `SSN` varchar(16) NOT NULL,
  `NOME` varchar(20) DEFAULT NULL,
  `COGNOME` varchar(20) DEFAULT NULL,
  `INDIRIZZO` varchar(45) DEFAULT NULL,
  `TELEFONO` varchar(10) DEFAULT NULL,
  `CLIENTE_Id_CLIENTE` varchar(20) NOT NULL,
  PRIMARY KEY (`SSN`),
  UNIQUE KEY `SSN` (`SSN`),
  KEY `fk_ALTRO_CONDUCENTE_CLIENTE1_idx` (`CLIENTE_Id_CLIENTE`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--
-- Struttura della tabella `assicuratore`
--

CREATE TABLE IF NOT EXISTS `assicuratore` (
  `MATRICOLA` char(9) NOT NULL,
  `NOME_AS` varchar(20) DEFAULT NULL,
  `COGNOME_AS` varchar(20) DEFAULT NULL,
```

```

`INDIRIZZO` varchar(45) DEFAULT NULL,
`TELEFONO` varchar(10) DEFAULT NULL,
PRIMARY KEY (`MATRICOLA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Struttura della tabella `clausola_contrattuale`
--

```

```

CREATE TABLE IF NOT EXISTS `clausola_contrattuale` (
  `NOME_CLAUSOLA` varchar(20) NOT NULL,
  `TIPOLOGIA_POLIZZA` varchar(20) DEFAULT NULL,
  `MASSIMALE_euro` int(10) DEFAULT NULL,
  `IMPORTO_BASE_euro` int(10) DEFAULT NULL,
  PRIMARY KEY (`NOME_CLAUSOLA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Struttura della tabella `cliente`
--

```

```

CREATE TABLE IF NOT EXISTS `cliente` (
  `ID_CLIENTE` varchar(16) NOT NULL,
  `INDIRIZZO` varchar(45) DEFAULT NULL,
  `TELEFONO` varchar(10) DEFAULT NULL,
  `E_MAIL` varchar(45) DEFAULT NULL,
  `TIPO_CLIENTE` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`ID_CLIENTE`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Struttura della tabella `contratto`
--

```

```

CREATE TABLE IF NOT EXISTS `contratto` (
  `NUM_PRATICA` int(11) NOT NULL AUTO_INCREMENT,
  `DATA_SOTTOSC` date DEFAULT NULL,
  `DATA_SCAD_RATA` date DEFAULT NULL,
  `FRAZIONAMENTO` varchar(15) DEFAULT NULL,
  `IMPORTO` varchar(5) DEFAULT NULL,
  `CLIENTE_ID_CLIENTE` varchar(16) NOT NULL,
  `MEZZO_TARGA` varchar(7) NOT NULL,
  `ASSICURATORE_MATRICOLA` varchar(15) NOT NULL,
  `CLAUSOLA_CONTRATTUALE_NOME_CLAUSOLA` varchar(20) NOT NULL,
  PRIMARY KEY (`NUM_PRATICA`),
  UNIQUE KEY `MEZZO_TARGA` (`MEZZO_TARGA`),
  KEY `fk_CONTRATTO_CLIENTE1_idx` (`CLIENTE_ID_CLIENTE`),

```

```

KEY `fk_CONTRATTO_MEZZO1_idx` (`MEZZO_TARGA`),
KEY `fk_CONTRATTO_ASSICURATORE1_idx` (`ASSICURATORE_MATRICOLA`),
KEY `fk_CONTRATTO_CLAUSOLA_CONTRATTUALE1_idx`
(`CLAUSOLA_CONTRATTUALE_NOME_CLAUSOLA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=14 ;

```

```

--
-- Struttura della tabella `mezzo`
--

```

```

CREATE TABLE IF NOT EXISTS `mezzo` (
  `TARGA` varchar(7) NOT NULL,
  `TIPO` varchar(15) DEFAULT NULL,
  `MARCA` varchar(15) DEFAULT NULL,
  `MODELLO` varchar(15) DEFAULT NULL,
  `POTENZA_KW` varchar(3) DEFAULT '0',
  `ANNO_IMMATR` year(4) DEFAULT '1999',
  PRIMARY KEY (`TARGA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Struttura della tabella `persona`
--

```

```

CREATE TABLE IF NOT EXISTS `persona` (
  `NOME_C` varchar(20) DEFAULT NULL,
  `COGNOME_C` varchar(20) DEFAULT NULL,
  `SESSO` varchar(1) DEFAULT NULL,
  `ANNO_CONS_PATENTE` year(4) DEFAULT NULL,
  `DATA_NASCITA` date DEFAULT NULL,
  `CLASSE_MERITO` varchar(2) DEFAULT '14',
  `CLIENTE_ID_CLIENTE` varchar(16) NOT NULL,
  PRIMARY KEY (`CLIENTE_ID_CLIENTE`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Struttura della tabella `societa`
--

```

```

CREATE TABLE IF NOT EXISTS `societa` (
  `NOME_SOC` varchar(30) DEFAULT NULL,
  `REFERENTE` varchar(45) DEFAULT NULL,
  `CLIENTE_ID_CLIENTE` varchar(16) NOT NULL,
  PRIMARY KEY (`CLIENTE_ID_CLIENTE`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Limiti per le tabelle scaricate
--

--
-- Limiti per la tabella `altro_conducente`
--
ALTER TABLE `altro_conducente`
  ADD CONSTRAINT `altro_conducente_ibfk_2` FOREIGN KEY
  (`CLIENTE_Id_CLIENTE`) REFERENCES `cliente` (`ID_CLIENTE`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Limiti per la tabella `contratto`
--
ALTER TABLE `contratto`
  ADD CONSTRAINT `contratto_ibfk_6` FOREIGN KEY (`CLIENTE_ID_CLIENTE`)
  REFERENCES `cliente` (`ID_CLIENTE`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `contratto_ibfk_7` FOREIGN KEY (`MEZZO_TARGA`)
  REFERENCES `mezzo` (`TARGA`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `contratto_ibfk_8` FOREIGN KEY
  (`ASSICURATORE_MATRICOLA`) REFERENCES `assicuratore` (`MATRICOLA`) ON
  DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `contratto_ibfk_9` FOREIGN KEY
  (`CLAUSOLA_CONTRATTUALE_NOME_CLAUSOLA`) REFERENCES
  `clausola_contrattuale` (`NOME_CLAUSOLA`) ON DELETE CASCADE ON UPDATE
  CASCADE;

--
-- Limiti per la tabella `persona`
--
ALTER TABLE `persona`
  ADD CONSTRAINT `persona_ibfk_1` FOREIGN KEY (`CLIENTE_ID_CLIENTE`)
  REFERENCES `cliente` (`ID_CLIENTE`) ON DELETE CASCADE ON UPDATE CASCADE;

```

Capitolo 6:

Vincoli di check, asserzioni e trigger, viste

6.1: Vincoli di check

I vincoli di check permettono di limitare i valori assunti dai campi di una singola tabella o da più tabelle. Nel caso della base di dati assicurazione r.c, ci sarebbero innumerevoli vincoli da imporre, ad esempio si potrebbe fare in modo di escludere l'inserimento di un cliente persona fisica che sicuramente non ha più la patente di guida (perché troppo anziano) , impostando un vincolo sull'anno del conseguimento della patente, lo stesso vale anche per la classe di merito che potrà assumere solo valori numerici compresi tra 1 e 18.

Dichiarazione dei vincoli di check:

```
CREATE TABLE
NomeTabella (  NomeAttributo Dominio [check(Condizione)]
               NomeAttributo Dominio [check(Condizione)]
               ...
               )
```

La condizione contenuta nel vincolo di check, va specificata al momento della creazione della tabella e deve essere sempre verificata affinché la base di dati sia in uno stato consistente; sarà il DBMS ad occuparsi del controllo prima dell'inserimento o della modifica dei dati, eventualmente abortendo l'operazione (rollback parziale) in caso di violazione anche di un solo vincolo fra quelli dichiarati.

Esempio:

```
--
-- Struttura della tabella `persona`
--
CREATE TABLE IF NOT EXISTS `persona` (
  `NOME_C` varchar(20) DEFAULT NULL,
  `COGNOME_C` varchar(20) DEFAULT NULL,
  `SESSO` varchar(1) DEFAULT 'M' CHECK (`SESSO` = 'M' OR `SESSO` = 'F'),
  `ANNO_CONS_PATENTE` year(4) DEFAULT NULL CHECK (`ANNO_CONS_PATENTE` >=
    '1933'),
  `DATA_NASCITA` date DEFAULT NULL,
  `CLASSE_MERITO` varchar(2) DEFAULT '14' CHECK (`CLASSE_MERITO` >= '1' AND
    `CLASSE_MERITO` <= '18' ),
  `CLIENTE_ID_CLIENTE` varchar(16) NOT NULL,
  PRIMARY KEY (`CLIENTE_ID_CLIENTE`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Purtroppo non è possibile inserire vincoli di check utilizzando phpMyAdmin, tuttavia questa parte di controlli è comunque fattibile implementando appositi costrutti php a livello di interfaccia.

6.2: Asserzioni e trigger

Le asserzioni rappresentano vincoli non associati a nessun attributo o tabella in particolare, ma appartengono direttamente allo schema della base di dati, anche in questo caso il DBMS si occuperà di mantenere il db in uno stato consistente, assicurando che le condizioni specificate nel vincolo di check vengano sempre rispettate.

Le asserzioni si dichiarano in questo modo:

```
CREATE ASSERTION NomeAsserzione
check(Condizione)
```

Nel caso dell'assicurazione rc auto, si può pensare di utilizzare l'attributo "tipologia_polizza" appartenente all'entità CLAUSOLA_CONTRATTUALE combinato all'attributo "tipo" relativo all'entità MEZZO, per fare in modo che non possa mai esistere all'interno del database, un collegamento tra un tipo di veicolo ed una polizza non emessa per quella categoria. Sarebbe anche utile, mettere dei vincoli check sui singoli attributi al momento della creazione della tabella MEZZO e CLAUSOLA_CONTRATTUALE.

Esempio:

```
-----
--
-- Struttura della tabella `clausola_contrattuale`
--
CREATE TABLE IF NOT EXISTS `clausola_contrattuale` (
  `NOME_CLAUSOLA` varchar(20) NOT NULL,
  `TIPOLOGIA_POLIZZA` varchar(20) DEFAULT 'polizza auto' CHECK
  (`TIPOLOGIA_POLIZZA`= 'polizza auto' OR `TIPOLOGIA_POLIZZA`= 'polizza
  moto') OR `TIPOLOGIA_POLIZZA`= 'polizza commerciale'),
  `MASSIMALE_euro` int(10) DEFAULT NULL,
  `IMPORTO_BASE_euro` int(10) DEFAULT NULL,
  PRIMARY KEY (`NOME_CLAUSOLA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-----
--
-- Struttura della tabella `mezzo`
--
CREATE TABLE IF NOT EXISTS `mezzo` (
  `TARGA` varchar(7) NOT NULL,
  `TIPO` varchar(15) DEFAULT 'auto' CHECK(`TIPO`='auto' OR
  `TIPO`='moto' OR `TIPO`='commerciale'),
  `MARCA` varchar(15) DEFAULT NULL,
  `MODELLO` varchar(15) DEFAULT NULL,
  `POTENZA_KW` varchar(3) DEFAULT '0',
  `ANNO_IMMATR` year(4) DEFAULT '1999',
  PRIMARY KEY (`TARGA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-----
```

```

CREATE ASSERTION VINCOLO_CLAUSOLA_MEZZO_auto
CHECK (NOT EXISTS (SELECT *
                    FROM CLAUSOLA_CONTRATTUALE AS CL,MEZZO AS M,CONTRATTO AS
                    CON
                    WHERE
                    CL.NOME_CLAUSOLA=CON.CLAUSOLA_CONTRATTUALE_NOME_CLAUSOLA
                    AND
                    M.TARGA=CON.MEZZO_TARGA AND
                    CL.TIPOLOGIA_POLIZZA='polizza auto' AND M.TIPO<>'auto'
                    ))

CREATE ASSERTION VINCOLO_CLAUSOLA_MEZZO_moto
CHECK (NOT EXISTS (SELECT *
                    FROM CLAUSOLA_CONTRATTUALE AS CL,MEZZO AS M,CONTRATTO AS
                    CON
                    WHERE
                    CL.NOME_CLAUSOLA=CON.CLAUSOLA_CONTRATTUALE_NOME_CLAUSOLA
                    AND
                    M.TARGA=CON.MEZZO_TARGA AND
                    CL.TIPOLOGIA_POLIZZA='polizza moto' AND M.TIPO<>'moto'
                    ))

CREATE ASSERTION VINCOLO_CLAUSOLA_MEZZO_commerciale
CHECK (NOT EXISTS (SELECT *
                    FROM CLAUSOLA_CONTRATTUALE AS CL,MEZZO AS M,CONTRATTO AS
                    CON
                    WHERE
                    CL.NOME_CLAUSOLA=CON.CLAUSOLA_CONTRATTUALE_NOME_CLAUSOLA
                    AND
                    M.TARGA=CON.MEZZO_TARGA AND
                    CL.TIPOLOGIA_POLIZZA='polizza commerciale' AND
                    M.TIPO<>'commerciale'
                    ))

```

I trigger specificano invece le azioni da eseguire nel caso si verifichi la violazione di alcuni vincoli. Nell'ambito di questo progetto, non è necessario intraprendere azioni particolari, basta solo evitare di inserire dati che comprometterebbero la consistenza del database; a tal fine è sufficiente l'utilizzo di check ed asserzioni. Nel caso in cui, il programma scelto per l'implementazione fisica del progetto non supporti l'inserimento di tali vincoli, è comunque possibile inserire dei controlli nell'interfaccia in php.

6.3: Viste

Una vista è una singola tabella virtuale, perché non esiste necessariamente in forma fisica, che deriva da altre tabelle o da altre viste. Per la base di dati potrebbe essere utile creare delle viste per le tabelle che vengono utilizzate frequentemente in JOIN.

Vediamo alcuni esempi a pagina seguente:

Tabella CLIENTE in join con la tabella PERSONA:

```
CREATE VIEW CLIENTE_PERSONA
AS SELECT
ID_CLIENTE, COGNOME_C, NOME_C, DATA_NASCITA, TELEFONO, INDIRIZZO, E_MAIL, ANNO_C
ONS_PATENTE, CLASSE_MERITO
FROM CLIENTE, PERSONA
WHERE ID_CLIENTE=CLIENTE_ID_CLIENTE
```

Tabella CLIENTE in join con la tabella PERSONA:

```
CREATE VIEW CLIENTE_SOCIETA
AS SELECT ID_CLIENTE, NOME_SOC, REFERENTE, TELEFONO, INDIRIZZO, E_MAIL
FROM CLIENTE, SOCIETA
WHERE ID_CLIENTE=CLIENTE_ID_CLIENTE
```

Tabella SOCIETA' in join con la tabella ALTRO_CONDUCENTE:

```
CREATE VIEW SOCIETA_CONDUCENTI
AS SELECT
S.CLIENTE_ID_CLIENTE, NOME_SOC, REFERENTE, SSN, COGNOME, NOME, TELEFONO, INDIRIZ
ZO
FROM SOCIETA AS S, ALTRI_CONDUCENTI AS A
WHERE S.CLIENTE_ID_CLIENTE=A.CLIENTE_ID_CLIENTE
```

Tabella che contiene solo i contratti in scadenza:

```
CREATE VIEW CONTRATTO_IN_SCADENZA
AS SELECT * FROM CONTRATTO WHERE DATA_SCAD_RATA BETWEEN '$twoweeksafter'
AND '$threeweeksafter'
```

le due variabili presentate tra apici, sono state definite in php ed entrambe rappresentano delle date.

Capitolo 7:

Interrogazioni SQL

In seguito vengono riportare alcune interrogazioni sql eseguite sulla base di dati assicurazione responsabilità civile:

1) selezionare tutti i contratti in scadenza(ovvero tutti i contratti che scadono in un intervallo di tempo tra 2 e 3 settimane a partire dalla data corrente):

```
SELECT *
FROM CONTRATTO
WHERE DATA_SCAD_RATA BETWEEN '$twoweeksafter' AND '$threeweeksafter'
```

2) selezionare tutti i contatti dei clienti (persone fisiche) che hanno dei contratti in scadenza, si fa riferimento alla viste CLIENTE_PERSONA e CONTRATTO_IN_SCADENZA implementate a pagina 26:

```
SELECT
ID_CLIENTE, COGNOME_C, NOME_C, INDIRIZZO, TELEFONO, E_MAIL, TARGA, MARCA, MODELLO
FROM CLIENTE_PERSONA, CONTRATTO_IN_SCADENZA, MEZZO
WHERE
ID_CLIENTE=CLIENTE_ID_CLIENTE AND TARGA=MEZZO_TARGA
```

3) selezionare tutti i contatti dei clienti (società) che hanno dei contratti in scadenza, si fa riferimento alla viste CLIENTE_SOCIETA e CONTRATTO_IN_SCADENZA implementate a pagina 26:

```
SELECT
ID_CLIENTE, NOME_SOC, REFERENTE, INDIRIZZO, TELEFONO, E_MAIL, TARGA, MARCA, MODELLO
FROM CLIENTE_SOCIETA, CONTRATTO_IN_SCADENZA, MEZZO
WHERE
ID_CLIENTE=CLIENTE_ID_CLIENTE AND TARGA=MEZZO_TARGA
```

Capitolo 8:

Interfaccia grafica

L'interfaccia grafica del database è stata realizzata in html e php. Per ogni entità sono state realizzate delle home pages dalle quali è possibile compiere ricerche, inserimenti, modifiche per l'entità corrente; le cancellazioni invece sono consentite solamente per mezzi, clienti, assicuratori che non sono ancora stati registrati in alcun contratto. L'accesso alla home page degli altri conducenti, legati ad una società assicurata, è consentito solo al momento dell'inserimento o della modifica dei dati relativi alla società.

L'interfaccia grafica php è stata pensata per uso interno amministrativo, tuttavia con semplici modifiche è possibile renderla fruibile anche per preventivi on-line; non è prevista la manipolazione delle tipologie di polizza emesse dell'agenzia assicurativa, dato che l'emissione di una nuova polizza è un evento poco frequente ed oltretutto sarebbero necessari maggiori privilegi per accedervi (occorrerebbe essere amministratori di sistema).

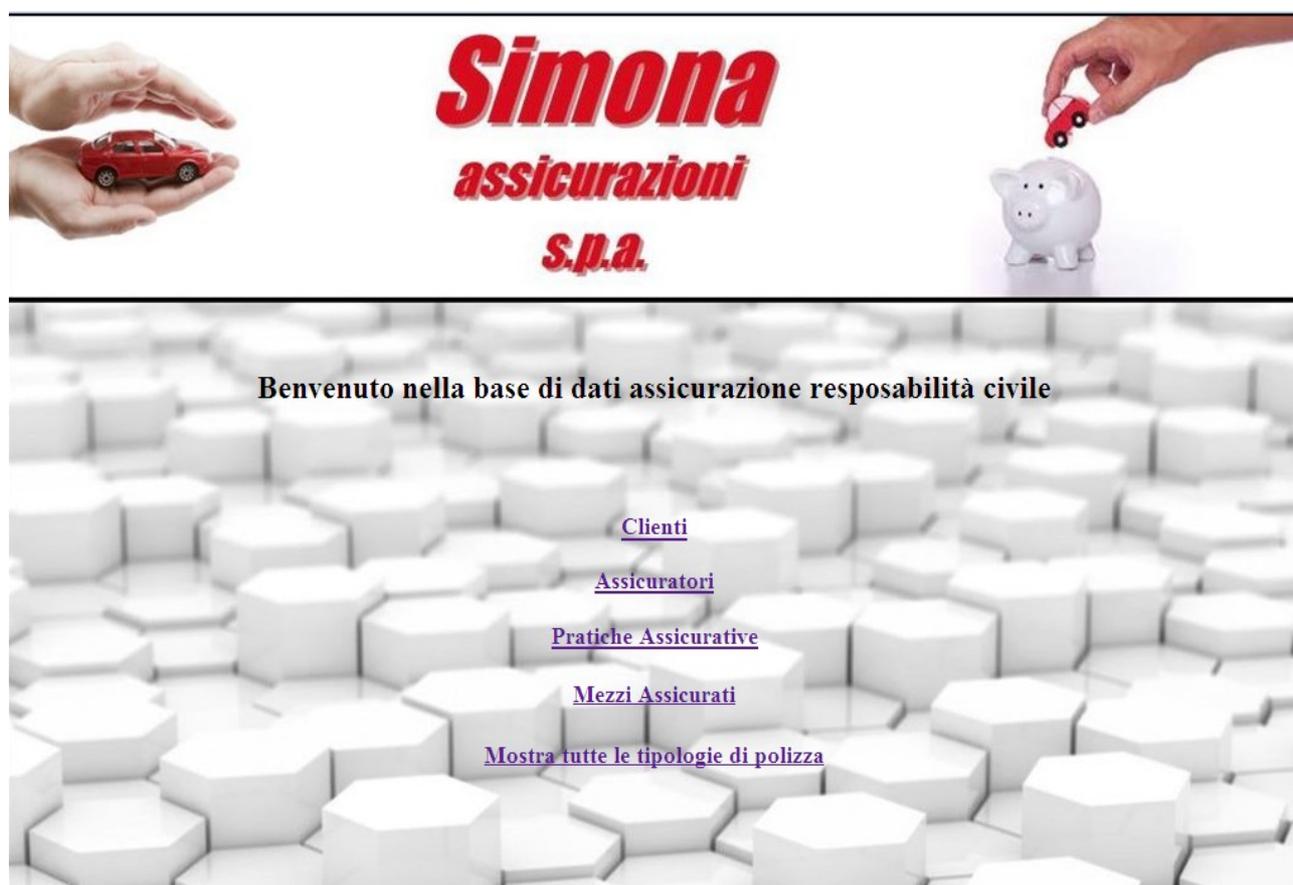


Figura 8.1: home page principale



Veicoli Assicurati



Ricerca Veicolo

Targa :

[Mostra Tutti I Mezzi Assicurati](#)



Figura 8.2: home page dei mezzi assicurati



Assicuratori



Ricerca Assicuratore

Matricola :

[Mostra Tutti Gli Assicuratori](#)



Figura 8.3: home page degli assicuratori



Clienti



Ricerca Cliente

Cod.Fiscale o Piva:

[Mostra Tutti I Clienti Assicurati](#)



Figura 8.4: home page dei clienti



Conducenti Assicurati



Ricerca conducente relativo alla società 'Edilcasa srl'(P.iva:'00000000004')

[Inserisci Nuovo Conducente](#)

[Mostra tutti i conducenti associati](#)

Figura 8.5: home page degli altri conducenti relativi ad una delle società assicurate



Polizze Disponibili



Tutte le tipologie di contratto disponibili

Nome Clausola	Tipologia polizza	Massimale (euro)	Importo base (euro)
casco	polizza auto	5500000	1500
furto e incendio	polizza auto	5000000	1200
guidatori esperti	polizza auto	5000000	900
rc base e cristalli	polizza auto	5000000	1100
respons. civile	polizza auto	5000000	1000



Figura 8.6: pagina che mostra tutti i pacchetti assicurativi offerti



Pratiche Assicurative



Inserire contraente, assicuratore, veicolo

Matricola Assicuratore

Cod.fiscale o p.iva

Targa Mezzo

[Mostra Tutte Le Pratiche](#)

[Mostra Solo le pratiche in scadenza](#)



Figura 8.7: home page per la gestione dei contratti



Pratiche Assicurative



Tutte le pratiche in scadenza tra il '2013-09-22' ed il '2013-09-29'

Num_pratica	Assicuratore	Cliente	Targa	Data sottoscrizione	Data scadenza	Nome Clausola	Frazionamento	Importo
9	L004545	00000000007	AB124CD	2013-03-24	2013-09-24	respons. civile	Semestrale	475
10	L004545	00000000007	AB124CE	2013-03-23	2013-09-23	rc base e cristalli	Semestrale	525
14	A005256	BNCSLV90B50A043Y	CG329MI	2012-09-25	2013-09-25	casco	Annuale	1060

[Elenca i clienti da avvisare](#)



Figura 8.8: select dei contratti in scadenza



Pratiche Assicurative



Tutte le pratiche in scadenza tra il '2013-09-22' ed il '2013-09-29'

Codice Fiscale	Cognome	Nome	Telefono	Indirizzo	E-mail	Targa	Marca	Modello
BNCSLV90B50A043Y	Bianchi	Silvio	0534556677	via torino 16, milano , italia	big_lebowski@gmail.it	CG329MI	Lancia	Delta

P. iva	Nome società	Referente	Telefono	Indirizzo	E-mail	Targa	Marca	Modello
00000000007	Consorzio Delta Po	Mauro Maestri	0423454561	via fiume 50, porto viro(ro),italia	con_delta@gmail.com	AB124CD	fiat	panda
00000000007	Consorzio Delta Po	Mauro Maestri	0423454561	via fiume 50, porto viro(ro),italia	con_delta@gmail.com	AB124CE	fiat	panda



Figura 8.9: reperimento dei contatti dei clienti da avvisare

