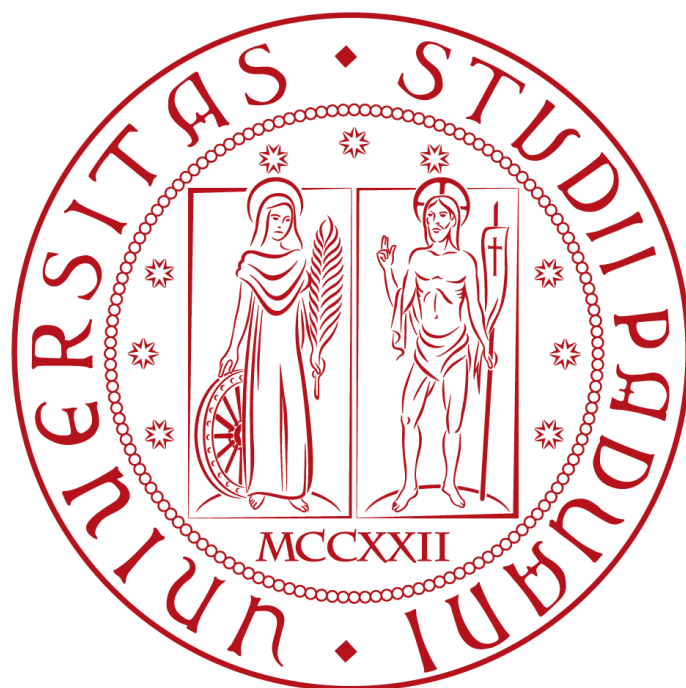


Università degli Studi di Padova



Progetto di Basi di Dati
Giovanni Damo
Matr. 610510

Anno Accademico 2013/2014

Indice

1. Abstract	5
2. Descrizione testuale dei requisiti e operazioni tipiche	5
3. Progettazione concettuale	7
3.1. Descrizione Testuale delle classi	7
3.1.1. Note preliminari:	7
3.1.2. Dispositivi	7
3.1.3. Assistenza	8
3.1.4. Servizi	9
3.1.5. Articoli	9
3.1.6. Categorie	10
3.1.7. Ordini	11
3.1.8. Dipendenti.....	12
3.1.9. Fornitori	12
3.1.10. Clienti.....	13
3.1.11. Amministratori	14
3.1.12. Gerarchia Persone	15
3.2. Descrizione testuale delle Associazioni	16
3.2.1. Clienti-Dispositivi: 1:N.....	16
3.2.2. Dispositivi-Categorie: N:1	16
3.2.3. Dispositivi-Assistenza: N:M.....	16
3.2.4. Assistenza-Servizi: N:M	16
3.2.5. Assistenza-Articoli: 1:N.....	17
3.2.6. Articoli-Categorie: N:1	17
3.2.7. Articoli-Ordini: N:1	17
3.2.8. Ordini-Fornitori(Persone): N:1	17
3.2.9. Ordini-Dipendenti: N:1	18
3.2.10. Dipendenti-Fornitori: N:1	18
3.2.11. Persone-Fornitori Clienti Amministratori:	18
3.3. Considerazioni sulle scelte progettuali	18
3.3.1. Trasformazione della gerarchia.....	18
3.4. Schema Concettuale in forma grafica	19

4. Progettazione logica	20
4.1. Descrizione testuale dello schema relazionale	20
4.1.1 Clienti.....	20
4.1.2 Amministratori.....	20
4.1.3 Fornitori	20
4.1.4 Persone	20
4.1.5 Dispositivi.....	20
4.1.6 Dipendenti.....	20
4.1.7 Servizi	21
4.1.8 Categorie	21
4.1.9 Ordini	21
4.1.10 Assistenza	21
4.1.11 Articoli	21
4.1.12 DispositiviInAssistenza.....	21
4.1.13 ServiziScelti	22
4.2. Schema logico in forma grafica.....	23
5. Implementazione dello schema logico	24
6. Query.....	30
6.1. Query 1:	30
6.2. Query 2.....	32
6.3. Query 3.....	33
6.4. Query 4.....	34
6.5. Query 5.....	35
6.6. Query 6.....	36
6.7. Query 7.....	38
6.8. Query 8.....	39
6.9. Query 9.....	40
7. Trigger-Funzioni-Procedure	43
7.1. Trigger 1) ControlloSconto	43
7.2. Trigger 2) ControlloInserimentoOrdine.....	44
7.3. Funzione 1) ControllaGaranziaArticoli.....	45

7.4. Funzione 2) ControllaGaranziaDispositivi	47
7.5. Procedura 1) AumentaPrezzo	48
7.6. Procedura 2) DiminuiscePrezzo	49
 8. Interfaccia web	 50
8.1. Composizione sito web.....	50
8.2. Layout Pagine.....	52
8.3. Sessioni PHP	52
8.4. Login e password sito web.....	53

1. Abstract

Questo progetto consiste nella realizzazione semplificata di un gestionale aziendale; ovvero un sistema in grado di tener traccia di clienti, fornitori, ordini, ed assistenze.

L'idea è nata da una mia passione, riparare/sistemare computer e altri dispositivi (smartphone, tablet etc) ad amici e conoscenti.

Con il tempo le persone che mi chiedono di risolvere i loro problemi con l'informatica sono aumentate, assieme anche alle tipologie di servizi da me offerti.

Per me è interessante sapere un cliente chi è, che dispositivi possiede, le loro caratteristiche, le assistenze fatte, che pezzi sono stati montati, gli eventuali problemi legati ai dispositivi e da che fornitore provengono.

L'uso di questa applicazione è quello di memorizzare e modificare efficacemente tutte queste informazioni, in modo tale da avere uno storico generale di tutte le attività sopra elencate.

2. Descrizione testuale dei requisiti e operazioni tipiche

Il progetto consiste nella realizzazione di una base di dati che modelli alcune classi coinvolte nella gestione di un'azienda nel settore dell'information technology; il personale amministratore interagisce con la base di dati tramite una interfaccia *web*, che è anche (in parte) destinata ai clienti.

- È presente una gerarchia chiamata *Persone*, che serve a racchiudere 3 categorie: i *Clienti*, gli *Amministratori* e i *Fornitori*.
- I *Clienti* sono coloro che posseggono dei *Dispositivi* ai quali verrà collegata un'eventuale *Assistenza*. La classe *Amministratori* serve solamente per la gestione della parte web che permette l'interazione con il database. I *Fornitori* possono essere persone singole (come un utente Ebay) oppure un'azienda (es. Crismatica s.a.s.) a cui vengono collegati degli *Ordini* da evadere.
- Dei *Clienti* interessa la loro anagrafica compresa anche di username e password per il login nel sito web e i *Dispositivi* a loro associati.
- Dei *Fornitori* interessa: la loro anagrafica, i loro eventuali *Dipendenti* e gli *Ordini* ad essi associati.
- Dei *Dipendenti* interessa: l'anagrafica e il *fornitore* a cui sono collegati. Ho creato questa classe per ottenere un maggiore dettaglio nella composizione di un ordine; è utile sapere con chi si ha parlato, per stabilire un rapporto di fiducia tra le parti o anche semplicemente nel caso di un problema si può contattare la persona direttamente collegata.
- Di un *ordine* interessano: il *Fornitore*, le tempistiche, il prezzo e gli *Articoli* in questione.
- Degli *Articoli* interessa: il *Fornitore* di provenienza, le caratteristiche dell'articolo comprese date di acquisto e durata garanzia, e la Categoria a cui appartiene.
- La classe *Categoria* collega sia i *Dispositivi* che gli *Articoli* e serve appunto a categorizzare la merce.
- Dei *Dispositivi* interessano: le caratteristiche tecniche, il *Cliente* possessore e le eventuali *Assistenze* effettuate.

- Delle *Assistenze* interessano: il/i *Dispositivo/i* associati, la richiesta/problema del *cliente*, la soluzione proposta, il prezzo con eventuale sconto, i *Servizi* utilizzati ed eventualmente degli *Articoli* inseriti.
- La classe *Servizi* serve a contenere i *Servizi* offerti dall'azienda da poter inserire in un'assistenza. Di essi interessano: la descrizione e il prezzo.

Dopo questa panoramica generale tutte le operazioni tipiche sono:

Fornitori

- Inserimento nuovo fornitore
- Modifica fornitore
- Cancellazione fornitore

Servizi

- Inserimento nuovo servizio
- Modifica servizio
- Cancellazione servizio

Clienti

- Inserimento nuovo cliente
- Modifica cliente
- Cancellazione cliente

Dispositivo

- Inserimento nuovo dispositivo
- Modifica dispositivo
- Cancellazione dispositivo

Amministratori

- Inserimento nuovo amministratore
- Modifica amministratore
- Cancellazione amministratore

Categorie

- Inserimento nuova categoria
- Modifica categoria
- Cancellazione categoria

Dipendenti

- Inserimento nuovo dipendente
- Modifica dipendente
- Cancellazione dipendente

Ordini

- Inserimento nuovo ordine
- Modifica ordine

Assistenza

- Inserimento nuova assistenza
- Modifica assistenza

Articoli

- Inserimento nuovo articolo
- Modifica articolo
- Cancellazione articolo

Nota:

Queste sono, come detto sopra, tutte le operazioni effettuabili nella base di dati, in questo progetto tuttavia ne sono state realizzate solo alcune.

3. Progettazione concettuale

3.1. Descrizione Testuale delle classi

3.1.1. Note preliminari:

- I campi con tipo DATE hanno il formato originario del tipo data, ovvero AAAA.MM.GG

3.1.2. Dispositivi

La classe *Dispositivi* descrive tutte le caratteristiche di un dispositivo hardware in possesso ad un cliente. Nella mia realtà questa classe serve a racchiudere tutte quelle informazioni dei rispettivi dispositivi delle varie persone in modo da sapere in anticipo (se è già stato inserito in precedenza) per esempio il sito dove scaricare i driver aggiornati, oppure che processore è installato, quanta ram possiede etc. Oltre a queste caratteristiche, quando un *cliente* richiede un'assistenza la richiede su uno o più *dispositivi* suo possesso; quindi posso avere uno storico di assistenze e modifiche di tutti i *dispositivi*.

Attributi

- **IdDispositivo - INTEGER NOT NULL - <<PK>>**
Indica l'id del dispositivo, è unico in tutta la base di dati, quindi ci saranno tanti IdDispositivo quanti sono i dispositivi. È stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto un nuovo dispositivo.
- **IdCliente - INTEGER NOT NULL - <<FK>>**
Indica l'IdCliente proprietario del dispositivo, ovviamente è stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto un nuovo dispositivo. Non può esistere un dispositivo senza un proprietario.
- **Nome VARCHAR(50)**
Indica il nome del dispositivo.
- **Descrizione BLOB**
Indica una descrizione dettagliata delle componenti hardware del dispositivo. È stato deciso di metterle tutte in un campo descrittivo perché si trattano dispositivi di varia natura e non tutti hanno le stesse caratteristiche (es. di una stampante non mi interessa il processore, mentre di un Pc sì).
- **SerialNumber VARCHAR(20)**
Indica il serial number del dispositivo, non è stata aggiunta la specifica NOT NULL, perché potrebbero esserci dispositivi con questo dato mancante (illeggibile, rimosso etc) così da non precludere l'inserimento del dispositivo.
- **Immagine BLOB,**
Campo predisposto per contenere l'immagine del dispositivo.

- **DataAcq DATE**
Indica la data di acquisto del dispositivo, utile per ricavarne la garanzia. Vedi funzione2.
- **Categoria VARCHAR(20) - <<FK>>**
Indica la categoria merceologica del dispositivo.
- **SitoDriver VARCHAR(100)**
Indica il link al sito dove poter andare a reperire dati e informazioni sul dispositivo.
- **Note BLOB**
Indica un campo di testo dove poter annotare dei particolari dettagli in merito al Dispositivo.

3.1.3. Assistenza

La classe *Assistenza* descrive cosa vado a fare a un determinato *dispositivo*. Come descriverò più dettagliatamente nella sezione della molteplicità, posso eseguire *un'assistenza* su più *dispositivi*, a patto che siano dello stesso *cliente*. Memorizza il problema lamentato dal cliente, la soluzione proposta, le date di richiesta dell'assistenza e la data di termine e il prezzo con eventualmente abbinato uno sconto.

Attributi

- **IdAssistenza - INTEGER NOT NULL - <<PK>>**
Indica univocamente l'assistenza nella base di dati.
- **Richiesta BLOB**
Indica il problema esposto dal cliente, è un campo BLOB perché potrebbero non bastare 255 caratteri per esprimere il concetto.
- **Soluzione BLOB**
Indica la risoluzione alla richiesta del problema sopra esposto, anche qui è un campo BLOB perché potrebbero non bastare 255 caratteri per esprimere il concetto.
- **DataRichiesta DATE**
Indica la data di richiesta dell'assistenza.
- **DataTermine DATE**
Indica la data di termine dell'assistenza.
- **PrezzoTot FLOAT**
Indica il prezzo impostato per il pagamento dell'assistenza. Questo campo è di libero inserimento, ovviamente bisogna confrontarsi con i servizi inseriti nell'assistenza per poter formulare un prezzo.

- **Sconto FLOAT**

Indica un eventuale sconto da applicare al prezzo totale. Non è espresso in percentuale ma in euro. Si veda nello specifico la funzione1 per il controllo dello sconto.

3.1.4. Servizi

La classe *Servizi* descrive tutti i *servizi* offerti in modo da poter avere una lista di cosa è in grado di offrire l'azienda, utile soprattutto per avere una sorta di "listino" prezzi di base per stabilire il prezzo totale di un'assistenza.

Attributi

- **IdServizio INTEGER NOT NULL - <<PK>>**

Indica univocamente nella base di dati il numero di servizio. È stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto un nuovo servizio.

- **Nome VARCHAR(50) NOT NULL**

Indica il nome (essenziale) del servizio offerto, questa informazione è fondamentale, quindi è stata inserita la specifica "NOT NULL".

- **Descrizione VARCHAR(250) NOT NULL**

Indica in modo verboso cosa consiste il servizio scelto, anche questa è un'informazione fondamentale, quindi è stata inserita la specifica "NOT NULL".

- **Prezzo FLOAT NOT NULL**

Indica il prezzo del servizio, è un'informazione è fondamentale, quindi è stata inserita la specifica "NOT NULL".

3.1.5. Articoli

La classe *articoli* descrive per ogni singolo *articolo* tutte quelle informazioni che ne derivano dall'*ordine* di acquisto, la descrizione del pezzo, i prezzi di vendita e di acquisto, la garanzia, l'utilizzo che se ne deve fare (se ad uso interno o uso vendita), l'eventuale *assistenza* associata e un campo note generico.

Vedendo questa classe più dall'alto, essa funge praticamente da magazzino, ovvero contiene al suo interno tutti gli *articoli* acquistati (che siano già venduti o meno, di uso interno o vendita).

Attributi

- **IdArticolo INTEGER NOT NULL - <<PK>>**

Indica univocamente l'articolo nella base di dati. Ogni articolo presente è diverso da un altro. Casi speciali: se ad esempio ordino 2 banchi di ram identiche, con le stesse caratteristiche, marca modello etc, avranno comunque due IdArticolo diversi.

- **IdOrdine INTEGER - <<FK>>**

Indica l'ordine a cui fa riferimento l'articolo. Più articolo possono far parte di uno stesso ordine.

- **SerialNumber VARCHAR(20) NOT NULL**

Indica il numero di serie dell'articolo. È stato scelto il tipo VARCHAR perché non è detto che il seriale sia composto da soli numeri. È stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto un nuovo articolo.

- **Descrizione VARCHAR(250)**

Indica una descrizione più approfondita dell'articolo, e comprende eventuali caratteristiche tecniche.

- **Categoria VARCHAR(20) - <<FK>>**

Indica la categoria merceologica dell'articolo, e fa riferimento alla classe Categoria come per i Dispositivi.

- **PrezzoAcq FLOAT**

Indica il prezzo di acquisto dell'articolo.

- **PrezzoVend FLOAT**

Indica il prezzo scelto per la vendita dell'articolo ai clienti.

- **Garanzia ENUM ('Fuori Garanzia', '1', '2', '3', '5', 'A vita')**

Indica il numero di anni di durata della Garanzia. Questo dato è di tipo enumerazione perché si presume che il range di valori possa essere da '0' ovvero senza garanzia a 'A vita' e il suo inizio è da intendersi a partire dalla data di acquisto dell'articolo. Per un maggiore dettaglio sull'argomento garanzia vedere il capitolo sulle funzioni.

- **Uso ENUM('Interno','Vendita')**

Indica l'utilizzo che se ne vuol fare dell'articolo, se per uso interno all'azienda (potrebbe essere uno strumento) o per uso vendita.

- **Venduto TINYINT(1)**

Indica lo stato attuale dell'articolo; se settato a '0' l'articolo NON è venduto, se settato a '1' invece sì.

- **IdAssistenza INTEGER - <<FK>>**

Indica il numero dell'assistenza a cui viene collegato l'articolo.

- **Note VARCHAR(250)**

È un campo note generico, usato per inserire alcuni dettagli che potrebbero essere utili nell'utilizzo dell'articolo o semplicemente potrebbero essere delle informazioni che non si riescono a memorizzare nei campi precedenti.

3.1.6. Categorie

La classe *Categorie* serve per aumentare il livello di dettaglio delle classi *Articoli* e *Dispositivi*, infatti viene collegata ad entrambe queste due classi. Descrive la tipologia di *articolo* o *dispositivo* in base alla *categoria* merceologica di esso.

Attributi

- **NomeCategoria VARCHAR(50) NOT NULL, - <<PK>>**
Indica il nome (sintetico) che identifica la categoria merceologica di un Articolo o di un Dispositivo. È stata inserita la specifica “NOT NULL” per obbligare ad compilare questo campo nel momento in cui viene aggiunto una nuova categoria.
- **Descrizione VARCHAR(250),**
Indica in maniera più verbosa le specifiche della categoria merceologica in questione.

3.1.7. Ordini

La classe *Ordini* descrive ogni singolo *ordine* effettuato a un *Fornitore*. In essa sono racchiuse le informazioni riguardanti il *Fornitore*, un eventuale *Referente*, le date di emissione dell’ordine e consegna della merce, il prezzo totale e un campo note generico.

Attributi

- **IdOrdine INTEGER NOT NULL - <<PK>>**
Indica il numero dell’ordine in modo univoco all’interno della base di dati. È stata inserita la specifica “NOT NULL” per obbligare ad compilare questo campo nel momento in cui viene aggiunto una nuovo ordine.
- **IdPersona INTEGER NOT NULL - <<FK>>**
Indica la persona quindi il fornitore a cui fa riferimento l’ordine. È stata inserita la specifica “NOT NULL” per obbligare ad compilare questo campo nel momento in cui viene aggiunto una nuovo ordine; infatti un ordine deve essere evaso da un Fornitore.
- **Referente VARCHAR(30) - <<FK>>**
Indica la persona di riferimento per questo ordine. Questo campo può essere omesso in quanto non sempre è presente un referente per ordine. In ogni caso riferisce la tabella Dipendenti che ovviamente fanno parte a un solo tipo di Fornitore.
- **DataEmissione DATE NOT NULL**
Indica la data di richiesta/emissione dell’ordine. È stata inserita la specifica “NOT NULL” per obbligare ad compilare questo campo nel momento in cui viene aggiunto una nuovo ordine, infatti se viene creato un nuovo ordine non può mancare la data.
- **DataConsegna DATE**
Indica la data di consegna della merce e quindi la terminazione dell’ordine. È stato scelto di non settare l’impostazione NOT NULL perché nel momento che si stipula un ordine, non si sa quando

la merce viene effettivamente consegnata e quindi si inserisce solo la data di consegna, poi quando l'ordine è concluso con l'arrivo della merce si aggiorna l'ordine inserendo anche la data di termine.

- **PrezzoTot**

Indica l'ammontare totale dell'ordine, i valori sono espressi in euro.

3.1.8. Dipendenti

La classe *Dipendenti* descrive tutte le persone alle dipendenze di un determinato *Fornitore*. In essa sono contenute le essenziali informazioni di anagrafica del *Dipendente*, la sua figura all'interno dell'azienda e i recapiti (telefono, ufficio ed email)

Attributi

- **IdFornitore INTEGER NOT NULL - <<FK>>**

Indica l'IdFornitore a cui questa persona è alle dipendenze. È stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto un nuovo Dipendente, infatti un dipendente deve avere assolutamente un datore di lavoro.

- **NomeCognome VARCHAR(40) NOT NULL - <<PK>>**

Indica il nome e cognome del dipendente. È stato scelto un campo unico per includere il nome e cognome per il motivo che non si sa sempre il nome o il cognome completi di un referente, spesso se ne inserisce uno solamente uno dei due (es rif Carlo o rif Rossi) inoltre questo campo identifica univocamente il dipendente nella base di dati. È stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto un nuovo Dipendente.

- **Ruolo VARCHAR(50)**

Identifica il ruolo del dipendente all'interno dell'azienda.

- **Email VARCHAR(30)**

Identifica l'email del dipendente.

- **Cellulare VARCHAR(12)**

Identifica il numero di cellulare del dipendente.

- **TelUfficio VARCHAR(12)**

Identifica il numero di telefono dell'ufficio del dipendente.

3.1.9. Fornitori

La classe *Fornitori* descrive l'anagrafica di ogni *Fornitore*, comprendendo i recapiti (telefono, fax, mail) il sito internet, la ragione sociale e la partita iva.

Anche se il *fornitore* è una persona singola (come ad esempio un utente ebay) viene comunque contrassegnato nel campo ragione sociale. Se fa parte di un sito on-line, quest'ultimo viene racchiuso tra parentesi quadre in maiuscolo in modo da facilitarne l'individuazione (esempio [EBAY] MemoryCooler).

Attributi

- **IdFornitore INTEGER NOT NULL - <<PK>> - <<FK>>**
Indica univocamente il fornitore all'interno della base di dati. È stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto una nuovo Fornitore.
- **RagioneSociale VARCHAR(50) NOT NULL**
Indica il nome del Fornitore che sia indipendentemente un'azienda o una persona singola, è stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto una nuovo Fornitore.
- **Piva VARCHAR(20)**
Indica la partita iva del Fornitore.
- **TelUfficio VARCHAR(14)**
Indica il numero di telefono del fornitore.
- **Fax VARCHAR(14)**
Indica il numero di fax del Fornitore.
- **Email VARCHAR(30)**
Indica l'email principale del Fornitore.
- **WebSite VARCHAR(50)**
Indica il sito internet del Fornitore.

3.1.10. Clienti

La classe *Clienti* descrive per ogni singolo *cliente* la sua anagrafica, completa di recapiti (cellulare, telcasa, email). I campi username e password servono per identificare il cliente nel sito web e servono a lui stesso per effettuare la login.

Attributi

- **IdCliente INTEGER NOT NULL - <<PK>> - <<FK>>**
Indica in maniera univoca il Cliente all'intero della base di dati. È stata inserita la specifica "NOT NULL" per obbligare ad compilare questo campo nel momento in cui viene aggiunto una nuovo Cliente.
- **Nome VARCHAR(20)**

Indica il nome del Cliente.

- **Cognome VARCHAR(20) NOT NULL**

Indica il cognome del Cliente. È stata inserita la specifica “NOT NULL” per obbligare ad compilare questo campo nel momento in cui viene aggiunto un nuovo Cliente.

- **DataDiNascita DATE**

Indica la data di nascita del Cliente.

- **Email VARCHAR(50)**

Indica l’email del Cliente.

- **TelCasa VARCHAR(14)**

Indica il numero di telefono di casa del Cliente.

- **Cellulare VARCHAR(14)**

Indica il numero di cellulare del Cliente.

- **Username VARCHAR(15)**

Indica l’username del cliente per l’autenticazione all’area riservata nel sito web. È stato scelto di non mettere il vincolo NOT NULL perché si presuppone che possa esistere un cliente che non ha un login nel sito, in quanto è stato inserito nella base di dati dall’amministratore.

- **Password VARCHAR(15)**

Indica la password corrispondente al login per l’autenticazione del cliente nel sito web.

3.1.11. Amministratori

La classe *Amministratori* serve solamente per il sito web, indica quali sono le credenziali delle persone che possono modificare e interagire completamente con l’intero database tramite appunto l’interfaccia web. Essa descrive il nome, cognome, l’username e la password. È stato scelto di non mettere il vincolo NOT NULL su tutti i campi perché si presuppone che siano tutti di obbligo inserimento per la creazione di un utente *Amministratore*.

Attributi

- **IdAmministratore INTEGER NOT NULL - <<PK>> - <<FK>>**

Indica l’IdAmministratore che riferito dalla tabella persona individua univocamente l’amministratore nella base di dati.

- **Nome VARCHAR(20) NOT NULL**

Indica il nome dell’Amministratore.

- **Cognome VARCHAR(20) NOT NULL**

Indica il cognome dell’Amministratore.

- **Username VARCHAR(15) NOT NULL**

Indica l'username dell' Amministratore per l'autenticazione all'area riservata nel sito web.

- **Password VARCHAR(15) NOT NULL**

Indica la password corrispondente al login per l'autenticazione dell' Amministratore nel sito web.

3.1.12. Gerarchia Persone

La classe *Persone* è una gerarchia e comprende le classi: *Clienti*, *Amministratori* e *Fornitori*.

È di tipologia partizione e le sottoclassi sono disgiunte. Un cliente non può essere un fornitore con lo stesso nome. Nella situazione in cui un cliente o un fornitore volessero essere anche (rispettivamente) un fornitore o cliente, verrebbero inseriti come se fossero persone “diverse” e quindi con un nome diverso. Ad esempio se Mario Rossi è un cliente, può diventare anche un fornitore, ma deve esser inserito nella tabella fornitori nel campo RagioneSociale Mario Rossi.

Attributi

- **IdPersona INTEGER NOT NULL**

Indica l'id di ogni persona presente nella base di dati, esso è univoco sia che siano Clienti, Fornitori o Amministratori. È stata inserita la specifica “NOT NULL” per obbligare ad compilare questo campo nel momento in cui viene aggiunto una nuova Persona.

- **Indirizzo VARCHAR(50) - <<PK>>**

Indica l'indirizzo della Persona.

- **Citta VARCHAR(20)**

Indica la città attuale della Persona..

- **Cap CHAR(5)**

Indica il cap della città della Persona.

3.2. Descrizione testuale delle Associazioni

3.2.1. Clienti-Dispositivi: 1:N

Molteplicità:	Un <i>Cliente</i> può possedere uno o più <i>dispositivi</i> . Un <i>Dispositivo</i> può essere posseduto da al massimo un <i>Cliente</i> .
Totalità:	Totale verso <i>Clienti</i> / Parziale verso <i>Dispositivi</i> Se esiste un <i>Dispositivo</i> deve essere di un <i>Cliente</i> Un <i>Cliente</i> può anche non avere <i>dispositivi</i> (è stato inserito preventivamente per un eventuale contatto futuro).

3.2.2. Dispositivi-Categorie: N:1

Molteplicità:	A una <i>Categoria</i> possono appartenere uno o più <i>Dispositivi</i> . Un <i>Dispositivo</i> può appartenere a una e una sola <i>Categoria</i> .
Totalità:	Totale verso <i>Categorie</i> / Parziale verso <i>Dispositivi</i> Una <i>Categoria</i> può anche non avere <i>dispositivi</i> (esempio <i>Categoria Stampanti</i> , ma non sono ancora state inserite come dispositivo in nessun <i>cliente</i>). Se esiste un <i>dispositivo</i> deve appartenere a una <i>Categoria</i> .

3.2.3. Dispositivi-Assistenza: N:M

Molteplicità:	Un <i>Dispositivo</i> può andare in <i>assistenza</i> più volte. Un <i>Assistenza</i> può avere più <i>dispositivi</i> nello stesso momento.
Totalità:	Totale verso <i>Dispositivi</i> / Parziale verso <i>Assistenza</i> Un <i>Assistenza</i> deve avere almeno un dispositivo. Un <i>Dispositivo</i> può anche non esser mai andato in <i>assistenza</i> (es. è appena stato inserito ed è nuovo).
Attributi:	IdDispositivo: indica il/i <i>Dispositivi</i> in <i>Assistenza</i> IdAssistenza: indica l'identificatore dell' <i>Assistenza</i> .

3.2.4. Assistenza-Servizi: N:M

Molteplicità:	In un' <i>assistenza</i> possono essere inseriti più <i>Servizi</i> . Un <i>servizio</i> può comparire in più di un' <i>Assistenza</i> .
Totalità:	Totale verso <i>Servizi</i> / Parziale verso <i>Assistenza</i> Un' <i>assistenza</i> deve corrispondere almeno un <i>Servizio</i> . Un <i>servizio</i> può non essere associato ad un' <i>Assistenza</i> .
Attributi:	IdAssistenza: indica l'identificatore dell' <i>Assistenza</i> . IdServizio: indica l'identificatore del/i <i>servizio/i</i> inserito.

3.2.5. Assistenza-Articoli: 1:N

Molteplicità:	In un' <i>assistenza</i> può comprendere più <i>Articoli</i> da magazzino. Un <i>Articolo</i> può comparire in una sola <i>Assistenza</i> .
Totalità:	Parziale verso <i>Assistenza</i> / parziale verso <i>Articoli</i> Un <i>Articolo</i> può non essere inserito in nessuna <i>Assistenza</i> (es. <i>Articoli</i> ad uso Interno). Un' <i>Assistenza</i> può non comprendere alcun <i>Articolo</i> .

3.2.6. Articoli-Categorie: N:1

Molteplicità:	A una <i>Categoria</i> possono appartenere uno o più <i>Articoli</i> . Un <i>Articolo</i> può appartenere a una e una sola <i>Categoria</i> .
Totalità:	Totale verso <i>Categorie</i> / Parziale verso <i>Articoli</i> Un <i>Articolo</i> deve appartenere a una <i>Categoria</i> Una <i>Categoria</i> può non avere <i>Articoli</i> (momentaneamente).

3.2.7. Articoli-Ordini: N:1

Molteplicità:	In un <i>Ordine</i> possono essere richiesti più <i>Articoli</i> . Un <i>Articolo</i> può provenire da uno e uno solo <i>Ordine</i> .
Totalità:	Totalità verso <i>Articoli</i> / Totalità verso <i>Ordini</i> Un <i>Articolo</i> deve sempre provenire da un <i>Ordine</i> . Un <i>Ordine</i> deve esser composto da almeno un <i>Articolo</i> .

Nota: Può esistere il caso dell'acquisto/ritiro di un Articolo usato da un Cliente o da altra fonte. In questo caso un Cliente dovrebbe diventare un Fornitore, ma non si vuole percorrere questa strada. Per ovviare a questo problema, è stato inserito un IdOrdine "fantasma" con valore '999' che sta a indicare che quell'articolo proviene da un'altra fonte e molto probabilmente è un Articolo usato. Per fugare ogni dubbio sulla provenienza del prodotto, si utilizza il campo "Note" per descriverne i dettagli.

3.2.8. Ordini-Fornitori(Persone): N:1

Molteplicità:	Un <i>Fornitore</i> può evadere più <i>Ordini</i> . Un <i>ordine</i> può essere evaso da un solo <i>Fornitore</i> .
Totalità:	Totalità verso <i>Persone</i> / Parzialità verso <i>Ordini</i> Un <i>Ordine</i> deve essere evaso da un <i>Fornitore</i> esistente. Un <i>Fornitore</i> può anche non aver evaso ancora un <i>ordine</i> (es. è appena stato inserito come <i>Fornitore</i> futuro).

3.2.9. Ordini-Dipendenti: N:1

Molteplicità:	Un <i>Dipendente</i> (Referente) può seguire più <i>Ordini</i> Un <i>Ordine</i> può essere seguito da al massimo un <i>Dipendente</i> (Referente).
Totalità:	Parziale verso <i>Dipendenti</i> / Parziale verso <i>Ordini</i> In un <i>Ordine</i> può non essere inserito alcun <i>Dipendente</i> (Referente). Un <i>Dipendente</i> (Referente) può non aver seguito ancora alcun <i>Ordine</i> .

3.2.10. Dipendenti-Fornitori: N:1

Molteplicità:	Un <i>Fornitore</i> può avere più <i>Dipendenti</i> . Un <i>Dipendente</i> può essere alla dipendenza di un solo Titolare alla volta.
Totalità:	Totale verso <i>Fornitori</i> / Parziale verso <i>Dipendenti</i> Un <i>Dipendente</i> deve essere alle dipendenze di un <i>Fornitore</i> . Un <i>Fornitore</i> può anche non avere <i>Dipendenti</i> (es. <i>Fornitore</i> come persona singola che rappresenta l'intera azienda).

3.2.11. Persone-Fornitori|Clienti|Amministratori:

La gerarchia prevede una superclasse *Persone* e come sottoclassi *Clienti*, *Amministratori* e *Fornitori*.

Disgiunzione:	Non possono esistere <i>Clienti</i> che siano <i>Fornitori</i> e viceversa. Se una <i>persona</i> è un <i>Amministratore</i> non può essere ne <i>Cliente</i> ne <i>Fornitore</i> .
Copertura:	Non può esistere un <i>Cliente</i> al di fuori delle tre sottoclassi, come anche un <i>Fornitore</i> e anche un <i>Amministratore</i> .

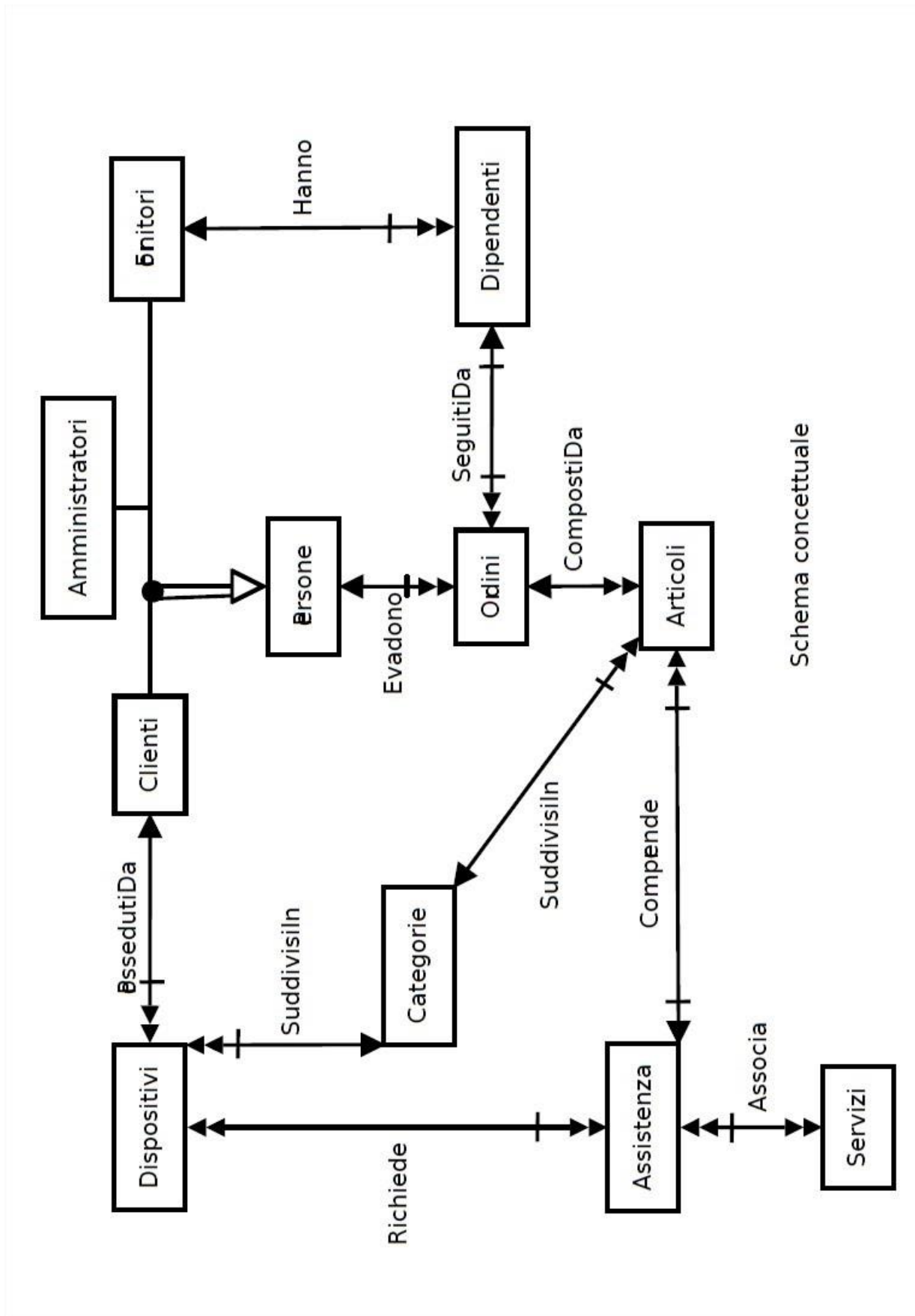
3.3. Considerazioni sulle scelte progettuali

3.3.1. Trasformazione della gerarchia

La Gerarchia esistente nel modello concettuale avente *Persone* come super classe e *Clienti*, *Amministratori*, *Fornitori* come sottoclassi è stata resa nel modello relazionale tramite partizionamento verticale visto che la classe radice ha delle associazioni proprie e che le sottoclassi hanno numerosi attributi autonomi e associazioni proprie. Si utilizza questa soluzione perché si

vogliono distinguere le tipologie di *Persona* (*Cliente*, *Amministratore*, *Fornitore*) volendo conoscere informazioni diverse a seconda del tipo di *Persona*.

3.4. Schema Concettuale in forma grafica



4. Progettazione logica

4.1. Descrizione testuale dello schema relazionale

4.1.1. Clienti (IdCliente*:int, Nome:string, Cognome:string, DataDiNascita: date, Email:string, TelCasa:string, Cellulare:string, Username:string, Password:string)

Chiave Primaria PK:	IdCliente
Chiave Esterna FK:	IdCliente (Persone)

4.1.2. Amministratori (IdAmministratore*:int, Nome:string, Cognome:string, Username:string, Password:string)

Chiave Primaria PK:	IdAmministratore
Chiave Esterna FK:	IdAmministratore (Persone)

4.1.3. Fornitori (IdFornitore*:int, RagioneSociale:string, Piva:string, TelUfficio:string, Fax:string, Email:string, WebSite:string)

Chiave Primaria PK:	IdFornitore
Chiave Esterna FK:	IdFornitore (Persone)

4.1.4. Persone (IdPersona:int, Indirizzo:string, Citta:string, Cap:string)

Chiave Primaria PK:	IdPersona
---------------------	-----------

4.1.5. Dispositivi (IdDispositivo:int, IdCliente*:int, Nome:string, Descrizione:string, SerialNumber:string, Immagine:blob, DataAcq:data, Categoria*:string, SitoDriver:string, Note:string)

Chiave Primaria PK:	IdDispositivo
Chiave Esterna FK:	IdCliente (Clienti) Categoria (Categorie)

4.1.6. Dipendenti (IdFornitore*:int, NomeCognome:string, Ruolo:string, Email:string, Cellulare:string, TelUfficio:string)

Chiave Primaria PK:	NomeCognome
Chiave Esterna FK:	IdFornitore (Fornitori)

4.1.7. Servizi (IdServizio:int, Nome:string, Descrizione:string, Prezzo:int)

Chiave Primaria PK:	IdServizio
----------------------------	------------

4.1.8. Categorie (NomeCategoria:string, Descrizione:string)

Chiave Primaria PK:	NomeCategoria
----------------------------	---------------

4.1.9. Ordini (IdOrdine:int, IdPersona*:int, Referente*:string, DataEmissione:date, DataConsegna:date, PrezzoTot:float, Note:blob)

Chiave Primaria PK:	IdOrdine
Chiave Esterna FK:	IdPersona (Fornitori) Referente (Dipendenti)

4.1.10. Assistenza (IdAssistenza:int, Richiesta:string, Soluzione:string, DataRichiesta:date, DataTermine:date, PrezzoTot:float, Sconto:float)

Chiave Primaria PK:	IdAssistenza
----------------------------	--------------

4.1.11. Articoli (IdArticolo:int, IdOrdine*:int, SerialNumber:string, Descrizione:string, Categoria*:string, PrezzoAcq:float, PrezzoVend:float, Garanzia: ENUM('Fuori Garanzia','1', '2', '3', '5', 'A vita'), Uso: ENUM('Interno','Vendita'), Venduto:bool, IdAssistenza*:int, Note:string)

Chiave Primaria PK:	IdArticolo
Chiave Esterna FK:	Categoria (Categorie) IdAssistenza (Assistenza) IdOrdine (Ordini)

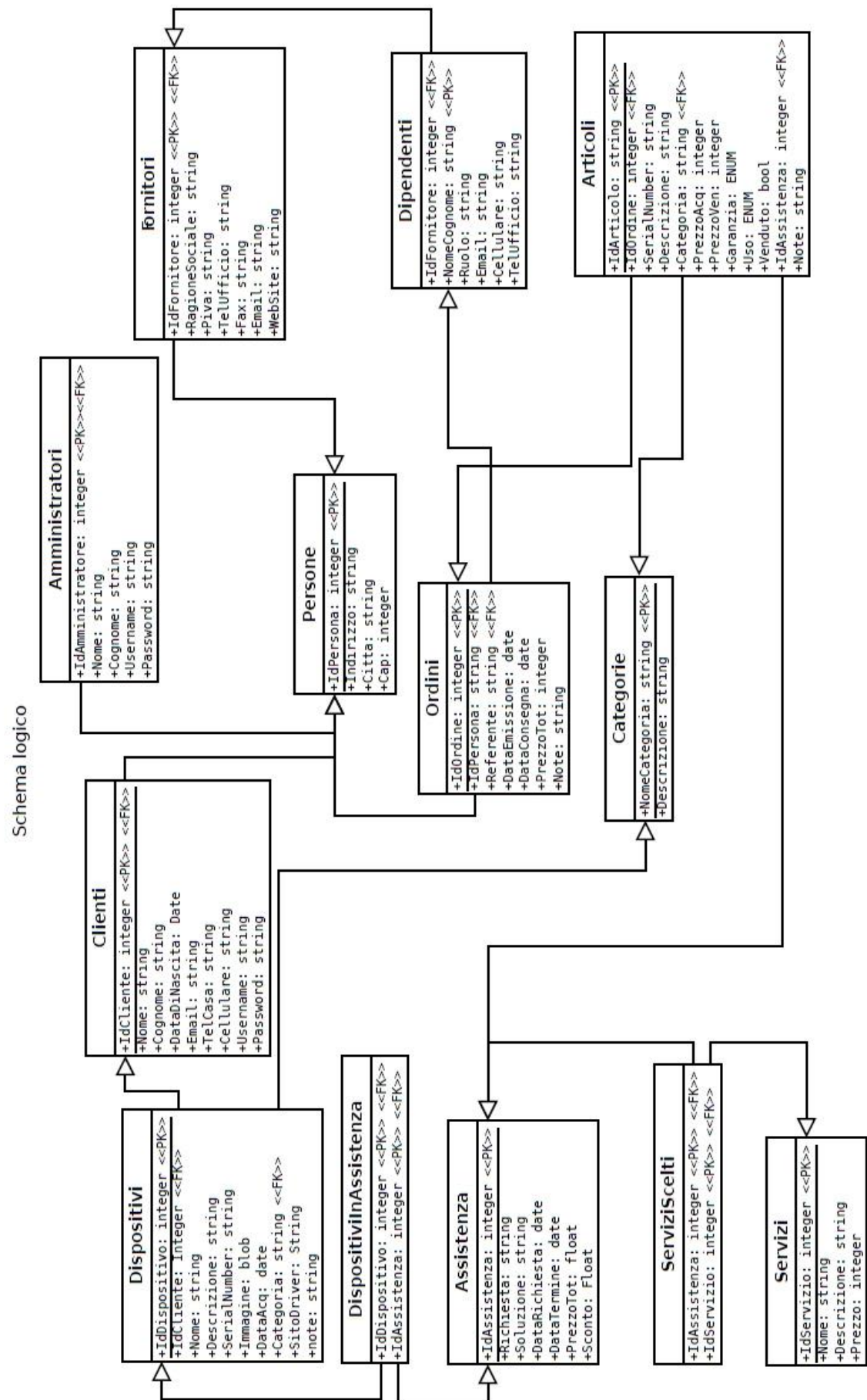
4.1.12. DispositiviInAssistenza (IdAssistenza*:int, IdDispositivo*:int)

Chiave Primaria PK:	IdDispositivo IdAssistenza
Chiave Esterna FK:	IdDispositivo (Dispositivi) IdAssistenza (Assistenza)

4.1.13. ServiziScelti (IdAssistenza*:int, IdServizio*:int)

Chiave Primaria PK:	IdAssistenza IdServizio
Chiave Esterna FK:	IdAssistenza (Assistenza) IdServizio (Servizi)

4.2. Schema logico in forma grafica



5. Implementazione dello schema logico

Il listato MySql seguente implementa lo schema descritto in precedenza per la sola creazione delle tabelle.

Tra il codice per la creazione di ogni tabella c'è un commento che indica quale azione si sta per compiere.

È stato inserito anche il codice che riguarda la cancellazione delle tabelle se già esistenti.

```
/* Elimino le tabelle-trigger-funzioni-procedure se esistono già'
*/
DROP TRIGGER IF EXISTS ControlloSconto;
DROP TRIGGER IF EXISTS ControlloInserimentoOrdine;
DROP FUNCTION IF EXISTS ControllaGaranziaArticoli;
DROP FUNCTION IF EXISTS ControllaGaranziaDispositivi;
DROP PROCEDURE IF EXISTS AumentaPrezzo;
DROP PROCEDURE IF EXISTS DiminuisciPrezzo;
DROP TABLE IF EXISTS Errori;
DROP TABLE IF EXISTS Articoli;
DROP TABLE IF EXISTS Persone;
DROP TABLE IF EXISTS Clienti;
DROP TABLE IF EXISTS Amministratori;
DROP TABLE IF EXISTS Fornitori;
DROP TABLE IF EXISTS Dipendenti;
DROP TABLE IF EXISTS Categorie;
DROP TABLE IF EXISTS Dispositivi;
DROP TABLE IF EXISTS DispositiviInAssistenza;
DROP TABLE IF EXISTS Assistenza;
DROP TABLE IF EXISTS Servizi;
DROP TABLE IF EXISTS ServiziScelti;
DROP TABLE IF EXISTS Ordini;

/*CREAZIONE DELLE TABELLE*/

/* Creo la tabella delle Persone */
CREATE TABLE Persone(
IdPersona INTEGER NOT NULL AUTO_INCREMENT,
Indirizzo VARCHAR(50),
Citta VARCHAR(20),
Cap CHAR(5),
PRIMARY KEY (IdPersona)
)
ENGINE=InnoDB;

/* Creo la tabella dei Clienti */
CREATE TABLE Clienti (
IdCliente INTEGER NOT NULL, -- AUTO_INCREMENT,
Nome VARCHAR(20),
```



```

Cognome VARCHAR(20) NOT NULL,
DataDiNascita DATE,
Email VARCHAR(50),
TelCasa VARCHAR(14),
Cellulare VARCHAR(14),
Username VARCHAR(15),
Password VARCHAR(15),
PRIMARY KEY (IdCliente),
FOREIGN KEY (IdCliente) REFERENCES Persone(IdPersona) ON UPDATE
CASCADE ON DELETE CASCADE
)
ENGINE=InnoDB;

/* Creo la tabella dei Amministratori */
CREATE TABLE Amministratori (
IdAmministratore INTEGER NOT NULL, -- AUTO_INCREMENT,
Nome VARCHAR(20) NOT NULL,
Cognome VARCHAR(20) NOT NULL,
Username VARCHAR(15) NOT NULL,
Password VARCHAR(15) NOT NULL,
PRIMARY KEY (IdAmministratore),
FOREIGN KEY (IdAmministratore) REFERENCES Persone(IdPersona) ON
DELETE CASCADE ON UPDATE CASCADE
)
ENGINE=InnoDB;

/* Creo la tabella dei Fornitori */
CREATE TABLE Fornitori(
IdFornitore INTEGER NOT NULL, -- AUTO_INCREMENT,
RagioneSociale VARCHAR(50) NOT NULL,
Piva VARCHAR(20),
TelUfficio VARCHAR(14),
Fax VARCHAR(14),
Email VARCHAR(30),
WebSite VARCHAR(50),
PRIMARY KEY (IdFornitore),
FOREIGN KEY (IdFornitore) REFERENCES Persone(IdPersona) ON UPDATE
CASCADE ON DELETE CASCADE
)
ENGINE=InnoDB;

/* Creo la tabella dei Dipendenti */
CREATE TABLE Dipendenti(
IdFornitore INTEGER NOT NULL,
NomeCognome VARCHAR(40) NOT NULL,
Ruolo VARCHAR(50),
Email VARCHAR(30),
Cellulare VARCHAR(12),
TelUfficio VARCHAR(12),
PRIMARY KEY (NomeCognome),

```

```

FOREIGN KEY (IdFornitore) REFERENCES Fornitori(IdFornitore) ON
DELETE CASCADE ON UPDATE CASCADE
)
ENGINE=InnoDB;

/* Creo la tabella delle Categorie */
CREATE TABLE Categorie (
NomeCategoria VARCHAR(50) NOT NULL,
Descrizione VARCHAR(250),
PRIMARY KEY (NomeCategoria)
)
ENGINE=InnoDB;

/* Creo la tabella dei Dispositivi dei clienti */
CREATE TABLE Dispositivi (
IdDispositivo INTEGER NOT NULL,
IdCliente INTEGER NOT NULL,
Nome VARCHAR(50),
Descrizione BLOB,
SerialNumber VARCHAR(20),
Immagine BLOB,
DataAcq DATE,
Categoria VARCHAR(20),
SitoDriver VARCHAR(100),
Note BLOB,
PRIMARY KEY (IdDispositivo),
FOREIGN KEY (Categoria) REFERENCES Categorie(NomeCategoria),
FOREIGN KEY (IdCliente) REFERENCES Clienti(IdCliente) ON UPDATE
CASCADE ON DELETE CASCADE
)
ENGINE=InnoDB;

/* Creo la tabella dei DispositiviInAssistenza */
CREATE TABLE DispositiviInAssistenza (
IdAssistenza INTEGER NOT NULL,
IdDispositivo INTEGER NOT NULL,
PRIMARY KEY (IdDispositivo, IdAssistenza),
FOREIGN KEY (IdDispositivo) REFERENCES Dispositivi(IdDispositivo)
ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (IdAssistenza) REFERENCES Assistenza(IdAssistenza)
)
ENGINE=InnoDB;

/* Creo la tabella Assistenza */
CREATE TABLE Assistenza (
IdAssistenza INTEGER NOT NULL AUTO_INCREMENT,
Richiesta BLOB,
Soluzione BLOB,
DataRichiesta DATE,
DataTermine DATE,
PrezzoTot FLOAT,
Sconto FLOAT,

```

```

PRIMARY KEY (IdAssistenza)
)
ENGINE=InnoDB;

/* Creo la tabella dei Servizi */
CREATE TABLE Servizi (
  IdServizio INTEGER NOT NULL,
  Nome VARCHAR(50) NOT NULL,
  Descrizione VARCHAR(250) NOT NULL,
  Prezzo FLOAT NOT NULL,
  PRIMARY KEY (IdServizio)
)
ENGINE=InnoDB;

/* Creo la tabella ServiziScelti */
CREATE TABLE ServiziScelti (
  IdAssistenza INTEGER NOT NULL,
  IdServizio INTEGER NOT NULL,
  PRIMARY KEY (IdAssistenza,IdServizio),
  FOREIGN KEY (IdAssistenza) REFERENCES Assistenza(IdAssistenza),
  FOREIGN KEY (IdServizio) REFERENCES Servizi(IdServizio)
)
ENGINE=InnoDB;

/* Creo la tabella degli Ordini */
CREATE TABLE Ordini (
  IdOrdine INTEGER NOT NULL,
  IdPersona INTEGER NOT NULL,
  Referente VARCHAR(30),
  DataEmissione DATE NOT NULL,
  DataConsegna DATE,
  PrezzoTot FLOAT,
  Note BLOB,
  PRIMARY KEY (IdOrdine),
  FOREIGN KEY (IdPersona) REFERENCES Fornitori(IdFornitore) ON
  UPDATE CASCADE ON DELETE CASCADE,
  FOREIGN KEY (Referente) REFERENCES Dipendenti(NomeCognome) ON
  UPDATE CASCADE ON DELETE CASCADE
)
ENGINE=InnoDB;

/* Creo la tabella degli Articoli */
CREATE TABLE Articoli (
  IdArticolo INTEGER NOT NULL,
  IdOrdine INTEGER,
  SerialNumber VARCHAR(20) NOT NULL,
  Descrizione VARCHAR(250),
  Categoria VARCHAR(20),
  PrezzoAcq FLOAT,
  PrezzoVend FLOAT,
  Garanzia ENUM('Fuori Garanzia','1', '2', '3', '5', 'A vita'),
  Uso ENUM('Interno','Vendita'),

```

```

Venduto TINYINT(1),
IdAssistenza INTEGER,
Note VARCHAR(250),
PRIMARY KEY (IdArticolo),
FOREIGN KEY (Categoria) REFERENCES Categorie(NomeCategoria) ON
UPDATE CASCADE,
FOREIGN KEY (IdAssistenza) REFERENCES Assistenza(IdAssistenza) ON
UPDATE CASCADE,
FOREIGN KEY (IdOrdine) REFERENCES Ordini(IdOrdine) ON UPDATE
CASCADE ON DELETE CASCADE
)
ENGINE=InnoDB;

/* POPOLAMENTO TABELLE */
LOAD DATA LOCAL INFILE 'Clienti.txt' INTO TABLE Clienti FIELDS
TERMINATED BY '#'
(IdCliente, Nome, Cognome, DataDiNascita, Email, TelCasa, Cellulare, Use
rname, Password);
LOAD DATA LOCAL INFILE 'Amministratori.txt' INTO TABLE
Amministratori FIELDS TERMINATED BY '#'
(IdAmministratore, Nome, Cognome, Username, Password);
LOAD DATA LOCAL INFILE 'Fornitori.txt' INTO TABLE Fornitori
FIELDS TERMINATED BY '#'
(IdFornitore, RagioneSociale, Piva, TelUfficio, Fax, Email, WebSite);
LOAD DATA LOCAL INFILE 'Persone.txt' INTO TABLE Persone FIELDS
TERMINATED BY '#' (IdPersona, Indirizzo, Citta, Cap);
LOAD DATA LOCAL INFILE 'Dispositivi.txt' INTO TABLE Dispositivi
FIELDS TERMINATED BY '#'
(IdDispositivo, IdCliente, Nome, Descrizione, SerialNumber, Immagine, D
ataAcq, Categoria, SitoDriver, Note);
LOAD DATA LOCAL INFILE 'Dipendenti.txt' INTO TABLE Dipendenti
FIELDS TERMINATED BY '#'
(IdFornitore, NomeCognome, Ruolo, Email, Cellulare, TelUfficio);
LOAD DATA LOCAL INFILE 'Servizi.txt' INTO TABLE Servizi FIELDS
TERMINATED BY '#' (IdServizio, Nome, Descrizione, Prezzo);
LOAD DATA LOCAL INFILE 'Categorie.txt' INTO TABLE Categorie
FIELDS TERMINATED BY '#' (NomeCategoria, Descrizione);
LOAD DATA LOCAL INFILE 'Ordini.txt' INTO TABLE Ordini FIELDS
TERMINATED BY '#'
(IdOrdine, IdPersona, Referente, DataEmissione, DataConsegna, PrezzoTo
t, Note);
LOAD DATA LOCAL INFILE 'Assistenza.txt' INTO TABLE Assistenza
FIELDS TERMINATED BY '#'
(IdAssistenza, Richiesta, Soluzione, DataRichiesta, DataTermine, Prezz
oTot, Sconto);
LOAD DATA LOCAL INFILE 'Articoli.txt' INTO TABLE Articoli FIELDS
TERMINATED BY '#'
(IdArticolo, IdOrdine, SerialNumber, Descrizione, Categoria, PrezzoAcq
, PrezzoVend, Garanzia, Uso, Venduto, IdAssistenza, Note);
LOAD DATA LOCAL INFILE 'DispositiviInAssistenza.txt' INTO TABLE
DispositiviInAssistenza FIELDS TERMINATED BY '#'
(IdAssistenza, IdDispositivo);

```

```
LOAD DATA LOCAL INFILE 'ServiziScelti.txt' INTO TABLE
ServiziScelti FIELDS TERMINATED BY '#' (IdAssistenza,IdServizio);
```

Il listato MySql seguente implementa il popolamento delle tabelle sopra create tramite l'inserimento di dati da file .txt. Il contenuto dei vari campi è delimitato dal carattere '#'.

NOTA: Ricordo che per poter usare il comando 'LOAD DATA ...' occorre aver avviato il client con il flag '--local-infile=1'.

```
/* POPOLAMENTO TABELLE */
LOAD DATA LOCAL INFILE 'Clienti.txt' INTO TABLE Clienti FIELDS
TERMINATED BY '#'
(IdCliente, Nome, Cognome, DataDiNascita, Email, TelCasa, Cellulare, Use
rname, Password);
LOAD DATA LOCAL INFILE 'Amministratori.txt' INTO TABLE
Amministratori FIELDS TERMINATED BY '#'
(IdAmministratore, Nome, Cognome, Username, Password);
LOAD DATA LOCAL INFILE 'Fornitori.txt' INTO TABLE Fornitori
FIELDS TERMINATED BY '#'
(IdFornitore, RagioneSociale, Piva, TelUfficio, Fax, Email, WebSite);
LOAD DATA LOCAL INFILE 'Persone.txt' INTO TABLE Persone FIELDS
TERMINATED BY '#' (IdPersona, Indirizzo, Citta, Cap);
LOAD DATA LOCAL INFILE 'Dispositivi.txt' INTO TABLE Dispositivi
FIELDS TERMINATED BY '#'
(IdDispositivo, IdCliente, Nome, Descrizione, SerialNumber, Immagine, D
ataAcq, Categoria, SitoDriver);
LOAD DATA LOCAL INFILE 'Dipendenti.txt' INTO TABLE Dipendenti
FIELDS TERMINATED BY '#'
(IdFornitore, NomeCognome, Ruolo, Email, Cellulare, TelUfficio);
LOAD DATA LOCAL INFILE 'Servizi.txt' INTO TABLE Servizi FIELDS
TERMINATED BY '#' (IdServizio, Nome, Descrizione, Prezzo);
LOAD DATA LOCAL INFILE 'Categorie.txt' INTO TABLE Categorie
FIELDS TERMINATED BY '#' (NomeCategoria, Descrizione);
LOAD DATA LOCAL INFILE 'Ordini.txt' INTO TABLE Ordini FIELDS
TERMINATED BY '#'
(IdOrdine, IdPersona, Referente, DataEmissione, DataConsegna, PrezzoTo
t, Note);
LOAD DATA LOCAL INFILE 'Assistenza.txt' INTO TABLE Assistenza
FIELDS TERMINATED BY '#'
(IdAssistenza, Richiesta, Soluzione, DataRichiesta, DataTermine, Prezz
oTot, Sconto);
LOAD DATA LOCAL INFILE 'Articoli.txt' INTO TABLE Articoli FIELDS
TERMINATED BY '#'
(IdArticolo, IdOrdine, SerialNumber, Descrizione, Categoria, PrezzoAcq
, PrezzoVend, Garanzia, Uso, Venduto, IdAssistenza, Note);
LOAD DATA LOCAL INFILE 'DispositiviInAssistenza.txt' INTO TABLE
DispositiviInAssistenza FIELDS TERMINATED BY '#'
(IdAssistenza, IdDispositivo);
LOAD DATA LOCAL INFILE 'ServiziScelti.txt' INTO TABLE
ServiziScelti FIELDS TERMINATED BY '#' (IdAssistenza, IdServizio);
```

6. Query

Elenco titoli query:

Query 1: mostrare, per ogni cliente, il/i dispositivo/i in suo possesso.

Query 2: mostrare per ogni fornitore quanti articoli ho comprato e in totale quanto ho ordinato in euro.

Query 3: mostrare la data dell'ultima assistenza effettuata ai clienti.

Query 4: mostrare per ogni dispositivo di ogni cliente il numero di assistenze effettuate su di esso.

Query 5: mostrare per ogni anno: il numero di assistenze fatte, il totale dei soldi incassati dalle assistenze(fatturato), l'ammontare totale degli sconti e fatturato senza sconti (andamento "finanziario" dell'attività).

Query 6: ottenere una lista delle assistenze fatte a tutti i clienti con annessa descrizione dell'intervento eseguito, il prezzo e lo sconto.

Query 7: mostrare il materiale "in vendita" in magazzino ma non ancora venduto.

Query 8: mostrare il materiale ad uso "INTERNO" presente nel magazzino.

Query 9: supporto per l'utilizzo delle funzioni sul controllo della garanzia; stampa IdArticolo, descrizione, IdDispositivo, la Descrizione.

6.1. Query 1:

La sua funzione è quella di mostrare, per ogni cliente, il/i dispositivo/i in suo possesso. Stamperà il Nome e Cognome del cliente, il Nome del/i dispositivo/i, la Categoria, e la Descrizione delle sue caratteristiche. La lista risultante è ordinata alfabeticamente per cognome.

È utile per vedere in un colpo solo tutti i clienti che dispositivi hanno e la loro descrizione.

NOME | COGNOME | NOME_DISPOSITIVO | CATEGORIA | DESCRIZIONE

Listato query1:

```
SELECT c.Nome, c.Cognome, d.nome AS Nome_Dispositivo,  
d.Categoria, d.Descrizione  
FROM Dispositivi d JOIN Clienti c ON d.IdCliente = c.IdCliente  
ORDER BY c.Cognome  
;
```

Risultato query1:

```
source query1.sql;
```

Nome	Cognome	Nome_Dispositivo	Categoria	Descrizione
Nico	Baldo	Iphone 4S	Smartphone Apple	Iphone 4S
Stefano	Biasi	PC portatile Asus A6T	Pc Portatile	intel i7 500GB HD, 6GB RAM, sc video 2 GB Ati
Stefano	Biasi	Iphone 5C	Smartphone Apple	Iphone 5C
Stefano	Biasi	Ipad 4	Tablet Apple	Ipad 4 WIFI 16GB
Alberto	Buoro	Acer Aspire cappe	Pc Fisso	QuadCore 2.0Ghz, HD 600GB, RAM 6GB, sc vido Nvidia 512BM, WIN8.1 PRO x64
Alberto	Buoro	Iphone 4	Smartphone Apple	Iphone 4
Angela	Covre	HP Pavillion DV6000	Pc Portatile	Dual core centrino 2.5Ghz HD 320GB RAM 4GB WIN7 PRO x64
Alessandra	Dalla Libera	HP Pavillion dmlz	Pc Portatile	Intel i3, Intel E7900 2.2 GHz, HD 500 GB - RAM: 4 GB
Piero	Dalla Libera	PC Fisso ASUS AS90	Pc Fisso	Intel i5 2.8Ghz 6GB Ram, 1 Tera HD, usb3, sc video integrata
Piero	Dalla Libera	PC Fisso ASUS AS90	Pc Fisso	Intel i5 2.8Ghz 6GB Ram, 1 Tera HD, usb3, sc video integrata
Piero	Dalla Libera	PC Fisso ASUS AS90	Pc Fisso	Intel i5 2.8Ghz 6GB Ram, 1 Tera HD, usb3, sc video integrata
Luigi	Finotto	Iphone 4S	Smartphone Apple	Iphone 4S
Luigi	Finotto	Ipad 3	Tablet Apple	Ipad 3 WIFI+3G 16GB
Adriano	Ghiriti	Iphone 4	Smartphone Apple	Iphone 4
Matteo	Grando	PC fisso Assemblato	Pc Fisso	dual core 3.0Ghz, HD 500GB, sc video ATI X1600, RAM 2GB, WIN 7 PRO x64
Simone	Milk	Alienware laptop x2x	Pc Portatile	Dual Core 1.90Ghz, SSD 120 GB, RAM 2GB, sc video ATI HD4500, WIN7 PRO x64
Iseo	Modolo	HP Compaq 500B	Pc Fisso	Intel Core 2 Duo, Intel Pentium E5500 2.8 MHz HD 320 GB - RAM: 4 GB

Nicolo'	Moretto	Dell Vostro 1520	Pc Portatile	Intel
Core 2 Duo E6500 3.2 MHz, HD 320 GB, RAM: 4 GB				
Nicolo'	Moretto	Iphone 5	Smartphone Apple	
Iphone 5				
Martina	Moretto	Sony Vaio S13	Pc Portatile	Intel
i3, Intel E8000 2.1 GHz, HD 640 GB - RAM: 4 GB				
Nicolo'	Moretto	PC Fisso Assemblato	Pc Fisso	Intel
Pentium 4 E2500, 3.2 GHz, HD 320 GB, RAM 2 GB, SCvideo ATI HD9500 XL				
Giovanni	Paneghel	Acer 1720	Pc Portatile	dual
core centrino 1.8GHz, HD 250GB, RAM 2GB WIN XP x86				
Giovanni	Paneghel	Iphone 4S	Smartphone Apple	
Iphone 4S				
Enrico	Peruch	SONY VAIO ES2	Pc Portatile	intel
dual Core 2.4Ghz 500GB HD, 4GB RAM, sc video 1 GB Ati				
Enrico	Peruch	Iphone 4	Smartphone Apple	
Iphone 4 32GB				
Enrico	Peruch	Ipad 2	Tablet Apple	Ipad
2 WIFI 32GB				
Federico	Tessari	Iphone 4	Smartphone Apple	
Iphone 4				
+-----+-----+-----+-----+-----+				
-----+				
27 rows in set (0.00 sec)				

6.2. Query 2

La sua funzione è quella di mostrare per ogni fornitore quanti articoli gli ho comprato e in totale quanto gli ho ordinato in euro.

RAGIONE_SOCIALE|TOTALE_SPESE_FORNITORE|NUMERO_ORDINI_TOT|NUMERO_ARTICOLI_ORDINATI

Listato query2:

```
SELECT f.RagioneSociale, SUM(a.PrezzoAcq) AS
TOTALE_SPESE_FORNITORE, COUNT(o.IdOrdine=a.IdOrdine) AS
NUMERO_ORDINI_TOT, COUNT(a.IdArticolo) AS
NUMERO_ARTICOLI_ORDINATI
FROM Fornitori f JOIN Persone p JOIN Ordini o JOIN Articoli a ON
(f.IdFornitore = p.IdPersona) AND (p.IdPersona = o.IdPersona) AND
(o.IdOrdine = a.IdOrdine)
GROUP BY f.RagioneSociale
;
```


Risultato query2:

```
source query2.sql;
```

RagioneSociale	TOTALE_SPESE_FORNITORE	NUMERO_ORDINI_TOT	NUMERO_ARTICOLI_ORDINATI
Crismatica Network S.r.l.	1019	17	17
IC Intracom Italia S.p.A.	105	6	6
[EBAY]360Handy	161	7	7
[EBAY]MemoryCooler	30	2	2

4 rows in set (0.01 sec)

6.3. Query 3

La sua funzione è quella di mostrare la data dell'ultima assistenza effettuata ai clienti, così da controllare in base alla data odierna se un determinato cliente è da troppo tempo che non viene contattato.

Ho fatto uso di una view, chiamata ListaAssistenzaDecr che serve per ottenere una lista dei clienti con le annesse date di assistenza ordinate dalla più recente alla più lontana. La SELECT successiva serve a prendere un record per ogni cliente presente nella view e questo sarà il primo, con la ultima data assistenza.

NOME | COGNOME | DATATERMINE

Listato query3:

```
DROP VIEW IF EXISTS ListaAssistenzeDecr;

CREATE VIEW ListaAssistenzeDecr (Nome, Cognome, DataTermine) AS
SELECT c.Nome, c.Cognome, a.DataTermine AS DATA_ULTIMA_ASSISTENZA
FROM Clienti c JOIN Dispositivi d JOIN DispositiviInAssistenza da
JOIN Assistenza a ON (c.IdCliente = d.IdCliente) AND
(d.IdDispositivo = da.IdDispositivo) AND (da.IdAssistenza =
a.IdAssistenza)
ORDER BY a.DataTermine DESC
;

SELECT *
FROM ListaAssistenzeDecr
GROUP BY Nome, Cognome
```

```
;
```

Risultato query3:

```
source query3.sql;
Query OK, 0 rows affected (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
+-----+-----+-----+
| Nome   | Cognome   | DataTermine |
+-----+-----+-----+
| Piero  | Dalla Libera | 2014-05-20 |
| Alessandra | Dalla Libera | 2014-01-30 |
| Angela | Covre       | 2013-12-10 |
| Iseo   | Modolo      | 2013-05-15 |
| Nicolo' | Moretto     | 2013-04-25 |
| Stefano | Biasi       | 2013-04-15 |
| Simone | Milk        | 2012-06-10 |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

6.4. Query 4

La funzione di questa query è quella di mostrare per ogni dispositivo di ogni cliente il numero di assistenze effettuate su di esso.

L'utilizzo di queste informazioni può avere vari scopi, mi è sembrato interessante ottenerle.

NOME | COGNOME | NOME_DISPOSITIVO | NUMERO_ASSISTENZE_EFFETTUATE

Listato query4:

```
SELECT c.Nome, c.Cognome, d.Nome AS NOME_DISPOSITIVO,
COUNT(a.DataTermine) AS NUMERO_ASSISTENZE_EFFETTUATE
FROM Clienti c JOIN Dispositivi d JOIN DispositiviInAssistenza da
JOIN Assistenza a
                        ON (c.IdCliente =
d.IdCliente) AND (d.IdDispositivo = da.IdDispositivo) AND
(da.IdAssistenza = a.IdAssistenza)
GROUP BY d.SerialNumber
;
```

Risultato query4:

```
source query4.sql;
```

```
+-----+-----+-----+
-----+
```

Nome	Cognome	NOME_DISPOSITIVO	NUMERO_ASSISTENZE_EFFETTUATE
Angela	Covre	HP Pavilion DV6000	1
Stefano	Biasi	PC portatile Asus A6T	1
Nicolo'	Moretto	PC Fisso Assemblato	2
Simone	Milk	Alienware laptop x2x	1
Piero	Dalla Libera	PC Fisso ASUS AS90	1
Piero	Dalla Libera	PC Fisso ASUS AS90	1
Piero	Dalla Libera	PC Fisso ASUS AS90	1
Iseo	Modolo	HP Compaq 500B	2
Nicolo'	Moretto	Dell Vostro 1520	1
Alessandra	Dalla Libera	HP Pavilion dmlz	1

10 rows in set (0.00 sec)

6.5. Query 5

Stampa per ogni anno: il numero di assistenze fatte, il totale dei soldi incassati dalle assistenze(fatturato), l'ammontare totale degli sconti e fatturato senza sconti.

Questa tabella riassuntiva sta a indicare l'andamento "finanziario" dell'attività.

ANNO | TOT_ASSISTENZE | FATTURATO | SCONTO | FATT_SENZA_SCONTI

Listato query5:

```

DROP VIEW IF EXISTS PrezziAcqArticoli;

CREATE VIEW PrezziAcqArticoli (PrezzoAcq, DataTermine) AS

SELECT SUM(art.PrezzoAcq), year(ass.DataTermine) AS anno
FROM Articoli art JOIN Assistenza ass ON (art.IdAssistenza =
ass.IdAssistenza)
GROUP BY anno
ORDER BY anno ASC
;

SET @miavar = 0;
SELECT DISTINCT year(a.DataTermine) AS ANNO,
COUNT(a.IdAssistenza) AS TOT_ASSISTENZE, SUM(a.PrezzoTot) AS
FATTURATO,

```

```

SUM(a.Sconto) AS SCONTO, @miavar := SUM(a.PrezzoTot) -
SUM(a.Sconto) AS FATT_SENZA_SCONTI
FROM Assistenza a, PrezziAcqArticoli p
GROUP By ANNO
;

```

Risultato query5:

```

source query5.sql;
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

```

ANNO	TOT_ASSISTENZE	FATTURATO	SCONTO	FATT_SENZA_SCONTI
2011	2	40	20	20
2012	4	440	120	320
2013	10	180	60	120
2014	4	340	0	340

4 rows in set (0.00 sec)

6.6. Query 6

La funzione di questa query è quella di ottenere una lista delle assistenze fatte a tutti i clienti con annessa descrizione dell'intervento eseguito, il prezzo e lo sconto.

NOME	COGNOME	NOME_DISPOSITIVO	DATATERMINE	BREVE_DESCRIZIONE_INTERVENTO	TOT€	SCONTO
------	---------	------------------	-------------	------------------------------	------	--------

Listato query6:

```

SELECT DISTINCT c.Nome, c.Cognome, d.Nome, a.DataTermine,
a.Soluzione AS Breve_descrizione_intervento, a.PrezzoTot AS TOT€,
a.Sconto
FROM Clienti c JOIN Dispositivi d JOIN DispositiviInAssistenza di
JOIN Assistenza a JOIN ServiziScelti sc JOIN Servizi s
ON (c.IdCliente = d.IdCliente) AND (d.IdDispositivo =
di.IdDispositivo) AND (di.IdAssistenza = a.IdAssistenza) AND
(a.IdAssistenza = sc.IdAssistenza) AND (sc.IdServizio =
s.IdServizio)
;

```

Risultato query6:

```
source query6.sql;
```

```
+-----+-----+-----+-----+-----+
-----
-----
-----
-----+-----+-----+
| Nome      | Cognome   | Nome      | DataTermine |
Breve_descrizione_intervento
| TOT€     | Sconto   |
+-----+-----+-----+-----+-----+
-----
-----
-----
-----+-----+-----+
| Simone    | Milk      | Alienware laptop x2x | 2012-06-10 |
Formattazione e installazione win7 pro x64 + sw base problemi però con
l'installazione del driver della scheda ethernet, xk in teoria non funziona
l'hardware e quindi non rileva il dispositivo
| 40 | 10 |
| Angela    | Covre     | HP Pavilion DV6000    | 2013-12-10 |
Formattazione e installazione win7 pro x64 + sw base salvataggio completamente
riuscito di dati personali.
| 20 | 0 |
| Iseo      | Modolo    | HP Compaq 500B        | 2013-05-15 | Pc
formattato, installato win7 x32 pro sp1 + aggiornamenti + sw base,
installazione nuova sc rete ethernet, controllo funzionamento chiavetta
vodafone (che funzionava)
| 30 | 0 |
| Iseo      | Modolo    | HP Compaq 500B        | 2013-08-10 |
Sistemazione tramite re-inserimento crack (win7) + controllo pc generico
| 0 | 0 |
| Stefano   | Biasi     | PC portatile Asus A6T | 2013-04-15 | SSD
Samsung 840 comprato da lui, installato, formattato pc, installato win 8.1 pro
N + sw base
| 20 | 20 |
| Piero     | Dalla Libera | PC Fisso ASUS AS90    | 2014-05-20 |
installazione 3 pc asus comprati alla crismatica, configurazione rete
interna+internet tramite nuovo switch asus, installazione monitor 22" Asus,
configurazione 3 pc win7 pro + update, installazione office 2013 small
business+kaspersky antivirus 2014 | 150 | 0 |
| Alessandra | Dalla Libera | HP Pavilion dmlz      | 2014-01-30 | Pc
formattato, installato win 7 pro + SW base, salvataggio e copia dati sensibili
| 20 | 0 |
| Nicolò    | Moretto    | Dell Vostro 1520      | 2013-04-25 | pc
formattato, installato win 7 pro x64, SW BASE + kaspersky is 2013
| 20 | 10 |
| Nicolò    | Moretto    | PC Fisso Assemblato   | 2012-05-30 | pc
formattato e installato win xp pro x32 installata SW base + comprata e
installata sc video ATI HD 5300
| 180 | 50 |
| Nicolò    | Moretto    | PC Fisso Assemblato   | 2011-05-30 | pc
formattato e installato win xp pro x32 installata SW base+salvataggio dati
| 20 | 10 |
+-----+-----+-----+-----+-----+
-----
-----
-----
-----+-----+-----+
10 rows in set (0.00 sec)
```

6.7. Query 7

La funzione di questa query è quella di mostrare il materiale "in vendita" in magazzino ma non ancora venduto.

È interessante sapere questa informazione perché se un cliente vuole del materiale usato, posso sapere se è disponibile o no, o semplicemente prima di fare un ordine a un fornitore controllare se è disponibile qualcosa in magazzino.

DESCRIZIONE_ARTICOLO	PREZZO_ACQUISTO	PREZZO_VENDITA	DATA_ACQUISTO	GARANZIA
----------------------	-----------------	----------------	---------------	----------

Listato query7:

```
SELECT a.Descrizione AS DESCRIZIONE_ARTICOLO, a.PrezzoAcq AS  
PERZZO_ACQUISTO, a.PrezzoVend AS PREZZO_VENDITA, o.DataConsegna  
AS DATA_ACQUISTO, a.Garanzia  
FROM Articoli a JOIN Ordini o ON (a.IdOrdine = o.IdOrdine)  
WHERE a.Uso = "VENDITA" AND a.Venduto = 0  
;
```

Risultato query7:

```
source query7.sql;  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| DESCRIZIONE_ARTICOLO | PERZZO_ACQUISTO |  
PREZZO_VENDITA | DATA_ACQUISTO | Garanzia |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| Schermo Per iphone 4 (bassa qualitÃ ) | 35.5 |  
40 | 2014-05-17 | 1 |  
| Schermo Per iphone 4 (bassa qualitÃ ) | 35.5 |  
40 | 2014-05-17 | 1 |  
| Batteria per iphone 4 | 15 |  
20 | 2014-05-17 | 1 |  
| Batteria per iphone 4 | 15 |  
20 | 2014-05-17 | 1 |  
| Schermo Per iphone 4S (bassa qualitÃ ) | 37.5 |  
42 | 2014-02-19 | 1 |  
| Schermo Per iphone 4S (bassa qualitÃ ) | 37.5 |  
42 | 2014-02-19 | 1 |  
| Antivirus Kasperky Internet Security 2014 Licenza 1 pc | 40 |  
45 | 2013-05-15 | |  
| Antivirus Kasperky Internet Security 2014 Licenza 1 pc | 40 |  
45 | 2013-05-15 | |  
| Cavo ethernet 1.5mt cat 6 | 2 |  
4 | 2014-01-29 | 2 |  
| Cavo ethernet 1.5mt cat 6 | 2 |  
4 | 2014-01-29 | 2 |  
| Antivirus Kasperky Internet Security 2014 Licenza 3 pc | 50 |  
55 | 2014-03-12 | Fuori Garanzia |  
| 1 mouse logitech wireless | 15 |  
20 | 2013-06-15 | 2 |
```

	1	supporto per iphone da auto		5	
12		2013-02-15		2	
	1	alimentatore 500Watt Macroware		35	
45		2012-02-21		2	
	Tasto home per iphone 4		5		
5		2012-04-17		1	
	Tasto home per iphone 4		5		
5		2012-04-17		1	
	Tasto home per iphone 4		5		
5		2012-04-17		1	
+-----+-----+-----+-----+-----+					
-----+-----+-----+-----+-----+					
17 rows in set (0.00 sec)					

6.8. Query 8

La funzione di questa query è quella di mostrare il materiale ad uso "INTERNO" presente nel magazzino.

È utile sapere questa informazione prima di comprare ulteriore materiale.

DESCRIZIONE_ARTICOLO | PERZZO_ACQUISTO | DATA_ACQUISTO | GARANZIA

Listato query8:

```
SELECT a.Descrizione AS DESCRIZIONE_ARTICOLO, a.PrezzoAcq AS
PERZZO_ACQUISTO, o.DataConsegna AS DATA_ACQUISTO, a.Garanzia
FROM Articoli a JOIN Ordini o ON (a.IdOrdine = o.IdOrdine)
WHERE a.Uso = "INTERNO"
;
```

Risultato query8:

```
source query8.sql;
+-----+-----+-----+-----+
----+
| DESCRIZIONE_ARTICOLO          | PERZZO_ACQUISTO | DATA_ACQUISTO |
Garanzia |
+-----+-----+-----+-----+
----+
| Router Asus DN300 radar edition | 85 | 2014-01-29 | 2 |
|
| Box Esterno per HD 3.5" USB 3   | 20 | 2014-01-29 | 2 |
|
| 1 cover ipad 2 tucano           | 30 | 2013-02-15 | 1 |
|
| 1 gopro hero 2 outdoor edition  | 350 | 2013-05-02 | 2 |
|
| 1 kit scuba per gopro hero       | 60 | 2013-05-02 | 2 |
|
| 1 banchi ram DRR3 800Ghz        | 30 | 2012-06-03 | 2 |
|
```

1	banchi ram DRR3 800Ghz		30		2012-06-03		2
1	banchi ram DRR3 800Ghz		30		2012-06-03		2
1	banchi ram DRR3 800Ghz		30		2012-06-03		2
1	20DVD doublelayer verbatim 42X		25		2012-02-21		1
1	alimentatori 40Watt Coolermaster		25		2011-06-30		2
1	alimentatori 40Watt Coolermaster		25		2011-06-30		2
+-----+-----+-----+-----+							
----+							
12 rows in set (0.01 sec)							

Listato query9:

```
SELECT IdArticolo, Descrizione
FROM Articoli
;

SELECT IdDispositivo, Descrizione
FROM Dispositivi
;
```

6.9. Query 9

Questa query è stata creata come supporto per l'utilizzo delle funzioni sul controllo della garanzia; infatti serve a stampare l'IdArticolo, la descrizione, l'IdDispositivo, la Descrizione.

Per poter controllare la garanzia di un articolo presente in magazzino o di un dispositivo di un cliente tramite la funzione bisogna inserire il corretto id.

IDARTICOLO | DESCRIZIONE

IDDISPOSITIVO | DESCRIZIONE

Risultato query9:

```
source query10.sql;
+-----+-----+-----+-----+
| IdArticolo | Descrizione |
+-----+-----+-----+-----+
| 1 | Schermo Per iphone 4 (bassa qualità ) |
| 2 | Schermo Per iphone 4 (bassa qualità ) |
| 3 | Batteria per iphone 4 |
| 4 | Batteria per iphone 4 |
| 5 | Schermo Per iphone 4S (bassa qualità ) |
| 6 | Schermo Per iphone 4S (bassa qualità ) |
| 7 | Antivirus Kasperky Internet Security 2014 Licenza 1 pc |
```


	8	Antivirus Kasperky Internet Security 2014 Licenza 1 pc
	9	Router Asus DN300 radar edition
	10	Box Esterno per HD 3.5" USB 3
	11	Cavo ethernet 1.5mt cat 6
	12	Cavo ethernet 1.5mt cat 6
	13	Antivirus Kasperky Internet Security 2014 Licenza 3 pc
	14	1 scheda rete realtech
	15	1 mouse logitech wireless
	16	1 supporto per iphone da auto
	17	1 cover ipad 2 tucano
	18	1 gopro hero 2 outdoor edition
	19	1 kit scuba per gopro hero
	20	1 banchi ram DRR3 800Ghz
	21	1 banchi ram DRR3 800Ghz
	22	1 banchi ram DRR3 800Ghz
	23	1 banchi ram DRR3 800Ghz
	24	1 scheda video Ati Radeon HD9500 XL
	25	1 alimentatore 500Watt Macroware
	26	1 20DVD doublelayer verbatim 42X
	27	1 alimentatori 40Watt Coolermaster
	28	1 alimentatori 40Watt Coolermaster
	29	Tasto home per iphone 4
	30	Tasto home per iphone 4
	31	Tasto home per iphone 4
	32	Antivirus Kasperky Internet Security 2013 Licenza 1 pc
	33	Pc fisso Anna
	34	Pc fisso Anna Internet
	35	Pc fisso studio 1
	36	Pc fisso studio 2
	37	Pc fisso studio 3
-----+		
37 rows in set (0.00 sec)		
-----+		
-----+		
	IdDispositivo	Descrizione
-----+		
-----+		
7 PRO x64	1	dual core 3.0GHz, HD 500GB, sc video ATI X1600, RAM 2GB, WIN
	2	dual core centrino 1.8GHz, HD 250GB, RAM 2GB WIN XP x86
	3	Dual core centrino 2.5GHz HD 320GB RAM 4GB WIN7 PRO x64
	4	Iphone 4
	5	Iphone 4S
	6	Iphone 4
WIN7 PRO x64	7	Dual Core 1.90Ghz, SSD 120 GB, RAM 2GB, sc video ATI HD4500,
	8	QuadCore 2.0Ghz, HD 600GB, RAM 6GB, sc vido Nvidia 512BM,
WIN8.1 PRO x64	9	Iphone 4
	10	Iphone 4S
	11	Iphone 4S

```

|      12 | Ipad 3 WIFI+3G 16GB
|
|      13 | intel i7 500GB HD, 6GB RAM, sc video 2 GB Ati
|
|      14 | Iphone 5C
|
|      15 | Ipad 4 WIFI 16GB
|
|      16 | intel dual Core 2.4Ghz 500GB HD, 4GB RAM, sc video 1 GB Ati
|
|      17 | Iphone 4 32GB
|
|      18 | Ipad 2 WIFI 32GB
|
|      19 | Intel Core 2 Duo, Intel Pentium E5500 2.8 MHz HD 320 GB -
RAM: 4 GB |
|      20 | Intel Core 2 Duo E6500 3.2 MHz, HD 320 GB, RAM: 4 GB
|
|      21 | Iphone 5
|
|      22 | Intel i3, Intel E8000 2.1 GHz, HD 640 GB - RAM: 4 GB
|
|      23 | Intel i3, Intel E7900 2.2 GHz, HD 500 GB - RAM: 4 GB
|
|      24 | Intel Pentium 4 E2500, 3.2 GHz, HD 320 GB, RAM 2 GB, SCvideo
ATI HD9500 XL |
|      25 | Intel i5 2.8Ghz 6GB Ram, 1 Tera HD, usb3, sc video integrata
|
|      26 | Intel i5 2.8Ghz 6GB Ram, 1 Tera HD, usb3, sc video integrata
|
|      27 | Intel i5 2.8Ghz 6GB Ram, 1 Tera HD, usb3, sc video integrata
|
+-----+-----+
-----+
27 rows in set (0.01 sec)

```

7. Trigger-Funzioni-Procedure

Nota: è necessario cambiare i limitatori con 'DELIMITER //' prima di creare i trigger, le funzioni o le procedure. Quando sono state create si chiude il limitatore con DELIMITER;

7.1. Trigger 1) ControlloSconto

La funzione di questo trigger è di controllare che non si inserisca uno sconto errato in un'*assistenza*.

Gli scenari errati sono due: o si inserisce uno sconto negativo o si inserisce uno sconto superiore al costo totale dell'*assistenza*.

Questi due casi limite vengono trattati nel seguente modo. Nel primo caso non potendo inserire uno sconto negativo, lo si corregge mettendo il valore dello sconto uguale a zero. Nel secondo invece si mette uno sconto pari al prezzo totale dell'*assistenza*, rendendo di fatto gratuito l'intervento.

```
/*Creo il trigger1 ControlloSconto*/
CREATE TRIGGER ControlloSconto
BEFORE INSERT ON Assistenza
FOR EACH ROW
BEGIN
    IF new.Sconto < 0 THEN
        SET new.Sconto = 0;
    ELSEIF new.Sconto > new.PrezzoTot THEN
        SET new.Sconto = new.PrezzoTot;
    END IF;
END; //
```

Stampiamo la lista delle assistenze (semplificata) presenti nel database con il comando:

```
SELECT IdAssistenza, DataTermine, PrezzoTot, Sconto FROM Assistenza;
+-----+-----+-----+-----+
| IdAssistenza | DataTermine | PrezzoTot | Sconto |
+-----+-----+-----+-----+
|          1 | 2012-06-10 |         40 |      10 |
|          2 | 2013-12-10 |         20 |       0 |
|          3 | 2013-05-15 |         30 |       0 |
|          4 | 2013-08-10 |          0 |       0 |
|          5 | 2013-04-15 |         20 |      20 |
|          6 | 2014-05-20 |        150 |       0 |
|          7 | 2014-01-30 |         20 |       0 |
|          8 | 2013-04-25 |         20 |      10 |
|          9 | 2012-05-30 |        180 |      50 |
|         10 | 2011-05-30 |         20 |      10 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Sono presenti 10 *Assistenze*, ora con il comando seguente inseriamo un'*assistenza* e volutamente mettiamo il valore dello sconto a 40 a fronte di un costo totale dell'*assistenza* di 20.

```
INSERT INTO Assistenza
(IdAssistenza,Richiesta,Soluzione,DataRichiesta,DataTermine,PrezzoTot,Sconto) VALUES (11,'Assistenza PROVA', 'Soluzione Prova', '2013.4.5', '2013.4.15', 20, 40);
Query OK, 1 row affected (0.00 sec)
```

L'inserimento va a buon fine, come testimonia la stampa delle *assistenze*, e vediamo come il trigger abbia corretto il valore dello sconto errato (40) con il massimo sconto possibile ovvero il prezzo totale dell'*assistenza* (20).

```
SELECT IdAssistenza, DataTermine, PrezzoTot, Sconto FROM Assistenza;
```

IdAssistenza	DataTermine	PrezzoTot	Sconto
1	2012-06-10	40	10
2	2013-12-10	20	0
3	2013-05-15	30	0
4	2013-08-10	0	0
5	2013-04-15	20	20
6	2014-05-20	150	0
7	2014-01-30	20	0
8	2013-04-25	20	10
9	2012-05-30	180	50
10	2011-05-30	20	10
11	2013-04-15	20	20

```
11 rows in set (0.00 sec)
```

7.2. Trigger 2) ControlloInserimentoOrdine

Questo trigger impedisce l'inserimento di un *ordine* d'acquisto.

Si presuppone che un ordine si possa fare a una persona, ma non lo si può richiedere per esempio a un *cliente*. Questo trigger controlla se l'Id inserito corrisponde a un Id presente nella tabella *Fornitori* e se sì, consente l'inserimento dell'*ordine*, altrimenti lo blocca.

```
/*Creo il trigger2 ControlloInserimentoOrdine e la relativa
tabella Errori*/
CREATE TABLE Errori(
                                Id CHAR(3) PRIMARY KEY)
ENGINE=InnoDB;

CREATE TRIGGER ControlloInserimentoOrdine
BEFORE INSERT ON Ordini
FOR EACH ROW
BEGIN
IF new.IdPersona NOT IN (SELECT IdFornitore FROM
Fornitori WHERE IdFornitore=new.IdPersona)
THEN
```

```

VALUES (NULL);

INSERT INTO Errori

END IF;

END; //
```

Stampiamo l'elenco dei *fornitori* attualmente presenti nel database con il seguente comando:

```

SELECT IdFornitore, RagioneSociale FROM Fornitori;
+-----+-----+
| IdFornitore | RagioneSociale |
+-----+-----+
|          1 | Crismatica Network S.r.l. |
|          2 | IC Intracom Italia S.p.A. |
|          3 | [EBAY]MemoryCooler |
|          4 | [EBAY]360Handy |
|          5 | Marco Polo Shop Conegliano |
+-----+-----+
5 rows in set (0.00 sec)
```

I *fornitori* presenti del database hanno l'Id che va dal 1 al 5, con in seguente inserimento cerchiamo di inserire un *ordine* al *fornitore* con Id=100, che non esiste e il trigger correttamente blocca l'inserimento.

```

INSERT INTO Ordini
(IdOrdine, IdPersona, Referente, DataEmissione, DataConsegna, PrezzoTot, Note)
VALUES (18, 100, 'Gianni', '2013.1.2', '2014.1.3', 35, '1 Licenza Antivirus
Kaspersky IS 2013');

ERROR 1048 (23000): Column 'Id' cannot be null
```

7.3. Funzione 1) ControllaGaranziaArticoli

Ho creato questa funzione perché è utile sapere se un *articolo* presente in magazzino è ancora in garanzia o meno.

La funzione mette in relazione: la data odierna, la data di acquisto del prodotto e la durata della garanzia. Siccome quest'ultimo parametro può variare tra “Fuori garanzia, 0,1,2,3,5 anni o A vita” con il passare del tempo, la garanzia può scadere.

```

/*Creo la funzione ControllaGaranziaArticoli*/
CREATE FUNCTION ControllaGaranziaArticoli (IdArticolo INT)
RETURNS VARCHAR(100)
BEGIN
    DECLARE AnniGaranzia VARCHAR(10);
    DECLARE GiorniGaranzia INT;
    DECLARE DataAcquisto DATE;
```

```

DECLARE x INT;

SELECT a.Garanzia INTO AnniGaranzia
FROM Articoli a
WHERE a.IdArticolo = IdArticolo;

IF (AnniGaranzia = 'Fuori Garanzia') THEN
SET GiorniGaranzia = 0;
END IF;
IF (AnniGaranzia = 1) THEN
SET GiorniGaranzia = 365;
END IF;
IF (AnniGaranzia = 2) THEN
SET GiorniGaranzia = 730;
END IF;
IF (AnniGaranzia = 3) THEN
SET GiorniGaranzia = 1095;
END IF;
IF (AnniGaranzia = 5) THEN
SET GiorniGaranzia = 1825;
END IF;
IF (AnniGaranzia = 'A vita') THEN
SET GiorniGaranzia = 50000;
END IF;

SELECT o.DataConsegna INTO DataAcquisto
FROM Ordini o JOIN Articoli a ON (o.IdOrdine =
a.IdOrdine)
WHERE a.IdArticolo = IdArticolo
GROUP BY a.IdArticolo;

SELECT DATEDIFF(current_date(),DataAcquisto) INTO x;

IF (x>GiorniGaranzia) THEN
RETURN 'Questo articolo è fuori dalla garanzia';
ELSE
RETURN 'Questo articolo è ancora in garanzia';
END IF;

END; //

```

Con la seguente select selezioniamo un *prodotto* con la data di acquisto e la durata della garanzia che staranno a indicare (in base alla data odierna) che è ancora in garanzia.

```

SELECT a.IdArticolo, a.Descrizione, o.DataConsegna, a.Garanzia FROM Articoli a
JOIN Ordini o ON (a.IdOrdine=o.IdOrdine) WHERE a.IdArticolo = 18;

```

IdArticolo	Descrizione	DataConsegna	Garanzia
18	1 gopro hero 2 outdoor edition	2013-05-02	2

```
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Chiamiamo la funzione passandogli l'IdArticolo=18 con il comando:

```
SELECT ControllaGaranziaArticoli (18);
+-----+
| ControllaGaranziaArticoli (18) |
+-----+
| Questo articolo è ancora in garanzia |
+-----+
1 row in set (0.01 sec)
```

Con la seguente select, invece, selezioniamo un *prodotto* che in base alla data di acquisto e alla durata della garanzia lo faranno risultare fuori dalla garanzia.

```
SELECT a.IdArticolo, a.Descrizione, o.DataConsegna, a.Garanzia FROM Articoli a
JOIN Ordini o ON (a.IdOrdine=o.IdOrdine) WHERE a.IdArticolo = 23;
+-----+-----+-----+-----+
| IdArticolo | Descrizione | DataConsegna | Garanzia |
+-----+-----+-----+-----+
| 23 | 1 banchi ram DDR3 800Ghz | 2012-06-03 | 2 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Chiamiamo la funzione passandogli l'IdArticolo=23 con il comando:

```
SELECT ControllaGaranziaArticoli (23);
+-----+
| ControllaGaranziaArticoli (23) |
+-----+
| Questo articolo è fuori dalla garanzia |
+-----+
1 row in set (0.01 sec)
```

7.4. Funzione 2) ControllaGaranziaDispositivi

Questa ha la stessa funzionalità della precedente, solamente che viene applicata ai *dispositivi* dei clienti. Gli viene passato sempre l'IdDispositivo e in base alla data odierna ritorna se la garanzia è scaduta o meno.

Si è assunto per default che un *dispositivo* (pc/cellulare/tablet che sia) abbia per normativa europea 2 anni di garanzia.

```
/*Creo la funzione ControllaGaranziaDispositivi*/
CREATE FUNCTION ControllaGaranziaDispositivi (IdDispositivo INT)
```

```

RETURNS VARCHAR(50)
BEGIN
    DECLARE GiorniGaranzia INT DEFAULT 730;
    DECLARE DataAcquisto DATE;
    DECLARE x INT;

    SELECT d.DataAcq INTO DataAcquisto
    FROM Dispositivi d
    WHERE d.IdDispositivo = IdDispositivo
    GROUP BY d.IdDispositivo;

    SELECT DATEDIFF(current_date(),DataAcquisto) INTO x;

    IF (x>GiorniGaranzia) THEN
        RETURN 'Questo dispositivo è fuori dalla garanzia';
    ELSE
        RETURN 'Questo dispositivo è ancora in garanzia';
    END IF;
END; //
```

7.5. Procedura 1) AumentaPrezzo

Le due seguenti procedure servono rispettivamente ad aumentare o diminuire (in valore percentuale) il prezzo di tutti i *servizi* offerti.

```

/*Creo la procedura AumentaPrezzo*/
CREATE PROCEDURE AumentaPrezzo (IN num DOUBLE)
BEGIN
    UPDATE Servizi
    SET Prezzo = Prezzo + (Prezzo/100*num);
END; //
```

Stampiamo la tabella Servizi PRIMA della chiamata della procedura

```
SELECT IdServizio, Nome, Prezzo FROM Servizi;
```

IdServizio	Nome	Prezzo
1	Formattazione	5
2	Formattazione + Salvataggio	10
3	Formattazione + Salvataggio + SW Base	15
4	Formattazione + Salvataggio + SW Base + SW Extra	30
5	Salvataggio Dati	5
6	Installazione SW Base + Pulizia PC	5
7	Installazione SW Extra + Pulizia PC	10
8	Installazione Hardware	0
9	Manutenzione Cellulare	10
10	Manutenzione Cellulare	20
11	Preventivo	0
12	Controllo Generale	0
13	Prima Configurazione PC	5


```
13 rows in set (0.01 sec)
```

Come esempio ho voluto aumentare il prezzo di tutti i *servizi* del 100%, chiamo la procedura tramite il comando:

```
CALL AumentaPrezzo(100);
```

Ora controlliamo la tabella *Servizi* con il campo Prezzo correttamente modificato

```
SELECT IdServizio, Nome, Prezzo FROM Servizi;
```

IdServizio	Nome	Prezzo
1	Formattazione	10
2	Formattazione + Salvataggio	20
3	Formattazione + Salvataggio + SW Base	30
4	Formattazione + Salvataggio + SW Base + SW Extra	60
5	Salvataggio Dati	10
6	Installazione SW Base + Pulizia PC	10
7	Installazione SW Extra + Pulizia PC	20
8	Installazione Hardware	0
9	Manutenzione Cellulare	20
10	Manutenzione Cellulare	40
11	Preventivo	0
12	Controllo Generale	0
13	Prima Configurazione PC	10

13 rows in set (0.00 sec)

7.6. Procedura 2) DiminuiscePrezzo

L'esempio è analogo al precedente, solo con la diminuzione del prezzo.

```
/*Creo la procedura DiminuiscePrezzo*/  
CREATE PROCEDURE DiminuiscePrezzo (IN num DOUBLE)  
BEGIN  
    UPDATE Servizi  
    SET Prezzo = Prezzo - (Prezzo/100*num);  
END; //
```

8. Interfaccia web

8.1. Composizione sito web

Il sito web realizzato prevede le seguenti pagine principali che costituiscono il menù orizzontale:

- Home
- Assistenza
- Servizi
- Registrazione
- Il mio account
- Down load
- Dove siamo

La pagina *Home* rappresenta la homepage del sito e offre una panoramica generale dei servizi principali che propone l'azienda.

La pagina *Assistenza* contiene un form per l'invio tramite mail di una richiesta di assistenza. Il visitatore espone un suo problema in merito alla sua realtà informatica. Non è necessario esser registrati al sito per poter compilare questo form.

La pagina *Servizi* contiene un elenco testuale con annessa descrizione di tutti i servizi che effettua l'azienda, pagina utile al visitatore per capire se quest'ultima è in grado di soddisfare a priori le proprie esigenze.

La pagina *Registrazione* contiene un form di registrazione per ogni nuovo visitatore. Questa pagina va ad aggiungere tutte le informazioni inserite nella base di dati nella classi Clienti e Persone. Con quest'operazione di registrazione il cliente sceglie anche il proprio username e password per poter accedere ad un'area riservata nel sito (visibile appunto solo ai clienti registrati) chiamata *Download*.

La pagina *Il mio account* permette al visitatore precedentemente registrato di loggarsi al sito e accedere alla propria area riservata. Come detto in precedenza questo sito è suddiviso nelle due parti Cliente e Amministratore, ma la parte più ricca e funzionale è quella Amministratore. Il visitatore registrato come Cliente, una volta loggato, è abilitato alla consultazione della pagina Download (specificata successivamente) cosa che un visitatore non registrato non può visionare. Il visitatore loggato come Amministratore invece ha a disposizione una serie di operazioni che vanno ad interagire direttamente con il database.

Queste operazioni sono:

- Inserimento di un nuovo cliente
- Inserimento di un nuovo fornitore
- Inserimento di un nuovo amministratore
- Eliminazione di uno specifico cliente
- Eliminazione di uno specifico fornitore
- Inserimento di un'assistenza

Le prime cinque operazioni sono degli inserimenti o delle cancellazioni di clienti/fornitori/amministratori.

L'ultima operazione invece è piuttosto elaborata, infatti richiede l'inserimento di dati in più tabelle (nello specifico le tabelle riguardano: Dispositivi, DispositiviInAssistenza, Assistenza, ServiziScelti, Servizi).

L'inserimento di un'assistenza è stato suddiviso in più step:

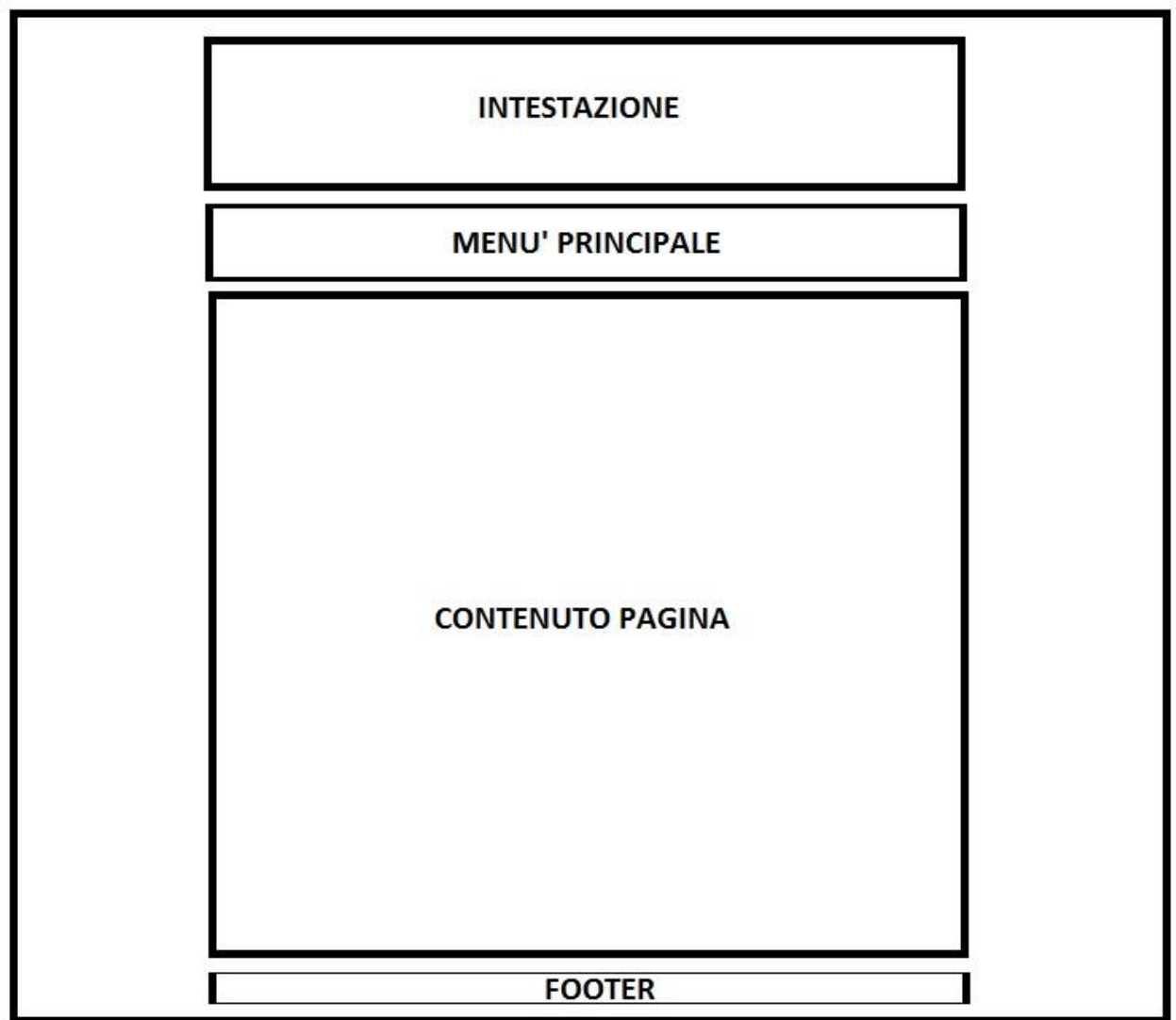
1. La visualizzazione di tutti i clienti presenti nella base di dati, con la possibilità di sceglierne uno (per ogni singola assistenza) tramite un radio button.
2. Dopo la scelta del cliente, vengono mostrati tutti i dispositivi in suo possesso e la possibilità di scegliere a quale di essi verrà applicata l'assistenza. Avendo predisposto la caratteristica che in una singola assistenza possono essere inseriti più dispositivi (dello stesso cliente), questa scelta è stata implementata tramite delle check box.
3. Dopo la scelta del cliente e dei dispositivi a cui effettuare l'assistenza, vengono mostrati tramite un form i campi per descrivere l'assistenza: il problema riscontrato, la soluzione proposta, le date di richiesta e termine dell'intervento, il prezzo lo sconto e un campo note generico.
4. Abbinato al precedente form, viene mostrato l'elenco dei servizi selezionabili tramite check box utilizzati nell'assistenza.
5. Conclusione con un messaggio di conferma del buon esito dell'operazione.

Nota: non è stato implementato (anche se effettivamente possibile nel database MySql) l'inserimento di alcuni Articoli utilizzati nell'assistenza. Per descrivere l'utilizzo/installazione di uno speciale articolo nell'assistenza lo si descrive nel campo Note della medesima.

La pagina *Download* è una pagina il cui contenuto è mostrato solo agli utenti registrati e loggati al sito web. Contiene un elenco di programmi (free-ware) linkabili al sito del produttore per il download. Questa pagina è stata creata per offrire a un visitatore più esperto una serie di programmi validi per migliorare l'efficienza del suo computer prima di richiede un'assistenza.

La pagina *Dove siamo* contiene i contatti dell'azienda e la mappa tratta da Google maps.

8.2. Layout Pagine



8.3. Sessioni PHP

Per mantenere le informazioni sullo stato della permanenza di un utente loggato sul sito, è stata usata una variabile di sessione `$_SESSION` diversa a seconda dell'utente. In particolare vengono utilizzate due variabili di sessione diverse:

- `$_SESSION['cliente']`
- `$_SESSION['amministratore']`

Ogni singolo script PHP delle aree riservate, prima di interrogare la base di dati, andrà a controllare se questa variabile di sessione è settata, ed il tipo di utente loggato grazie a due funzioni PHP

- `authenticateCl()`
- `authenticateAmm()`

che verificano se le corrispondenti variabili di sessioni sono settate o meno.

Quando l'utente fa il logout dal sito, la variabile viene distrutta. Viene fatto anche l'unset delle variabili associate alla sessione.

```
$sname = session name();
session destroy();
```

8.4. Login e password sito web

Per poter testare il corretto funzionamento del sito e del database, elenco delle credenziali d'accesso sia per un cliente che per un amministratore.

TIPO UTENTE	USERNAME	PASSWORD
Cliente	mang	123456
Amministratore	admin	admin