

GESTIONE FARMACIA

*ELABORATO PER IL
CORSO DI BASI DI DATI*

VITTORIO DAMIANO BRASINI matr. 0000883274

BEATRICE DI GREGORIO matr. 0001040859

FABRIZIO MARZO matr. 0000998072

INDICE

La documentazione tratta la progettazione e l'implementazione dell'elaborato sulla gestione della farmacia sviluppato da Vittorio Damiano Brasini, Beatrice di Gregorio, Fabrizio Marzo. La struttura è la seguente:

1. Introduzione
2. Analisi dei requisiti
 - 2.1. Indagine per la conoscenza del dominio applicativo
 - 2.2. Requisiti in linguaggio naturale
 - 2.3. Estrazione dei concetti fondamentali
3. Progettazione concettuale
 - 3.1. Sviluppo del concetto di oggetto di vendita
 - 3.2. Sviluppo del concetto di prodotto
 - 3.3. Sviluppo del concetto di disponibilità dei prodotti
 - 3.4. Sviluppo del concetto di vendita
 - 3.5. Sviluppo del concetto di rifornimento e reso
 - 3.6. Sviluppo del concetto di turnazione
 - 3.7. Anteprima dello schema generale
4. Progettazione logica
 - 4.1. Stima dei volumi dei dati
 - 4.2. Descrizione delle operazioni principali
 - 4.3. Stima frequenza delle operazioni principali
 - 4.4. Schemi di navigazione e tabelle degli accessi
 - 4.5. Raffinamento dello schema
 - 4.6. Analisi delle ridondanze
 - 4.7. Traduzione entità e associazioni in relazioni
 - 4.8. Schema relazionale finale
5. Progettazione fisica
 - 5.1. Traduzione delle operazioni in query SQL
6. Progettazione dell'applicazione
 - 6.1. Descrizione dell'interfaccia utente

1.INTRODUZIONE

Il progetto consiste nella realizzazione di un sistema database che funga da supporto alle farmacie locali. Il database, oltre all'immagazzinare informazioni riguardo ai prodotti in vendita e alle loro disponibilità, dovrà:

- gestire gli ordini per il rifornimento presso le case farmaceutiche e/o grossisti autorizzati;
- gestire le prenotazioni dei clienti in caso in cui i farmaci non siano disponibili;
- gestire la vendita dei prodotti che richiedono prescrizione medica e dei prodotti che non richiedono prescrizione medica;
- gestire i dipendenti e le loro turnazioni;
- gestire l'emissione degli scontrini e delle ricevute di vendita;
- gestire gli sconti (a termine) applicabili ai servizi o ai prodotti;
- gestire i movimenti dei prodotti da parte dei dipendenti.

2. ANALISI DEI REQUISITI

2.1. INDAGINE PER LA CONOSCENZA DEL DOMINIO

Per avere chiara un'idea generale dell'infrastruttura logica e funzionale che necessita una farmacia, ovvero tutto ciò che concerne i medicinali e la loro vendita, è stata condotta un'intervista presso la Farmacia Lanzoni di Cesena. Del resto, per riuscire a fornire una rappresentazione verosimile di tali dati su di un database, era quantomeno necessario e d'aiuto apprendere sufficientemente le convenzioni (anche di stampo legale) adottate in ambito farmaceutico per descrivere le caratteristiche dei farmaci e la loro identificazione, anche per quanto riguarda gli atti di vendita (quest'ultima nella necessità di fornire alla platea dei clienti una somministrazione di prodotti — potenzialmente nocivi — in maniera controllata e in linea col regolamento statale, che per alcuni casi prevede prezzi agevolati).

La farmacia intervistata è stata piuttosto cortese nel rispondere a tutti i dubbi e le domande nate anche durante la fase di colloquio.

Hanno inizialmente mostrato e spiegato il funzionamento del loro robot-scaffalista (non presente in tutte le farmacie) il quale organizza tutti i farmaci che non si trovano ancora in esposizione e tutti quelli nuovi appena arrivati col corriere e che necessitano quindi di essere registrati nel sistema. In sostanza, tutti i nuovi arrivi vengono dati “in pasto” al macchinario, che si occupa di scansionarli uno ad uno e posizionarli in maniera intelligente dentro al deposito). Da questo aspetto si può evincere sin da subito la distinzione tra due tipi di “giacenze”: una relativa alla merce esposta, ed una che si può tradurre sostanzialmente in “mercanzia di scorta” (tipicamente stipata in un deposito). In realtà si potrebbe ulteriormente suddividere la giacenza, avendo addirittura tre tipologie di scomparti, tenendo conto anche dei cassetti (che solitamente sono presenti dietro al bancone dell'operatore) i quali consentono una veloce fruizione di gran parte dei prodotti non esposti negli scaffali, ma senza dover necessariamente andarli a cercare in deposito. In realtà, per semplicità di gestione e probabilmente derivante dal fatto che diventa un po' scomodo (non essendo automatizzato) dover continuamente tener traccia dei movimenti interni tra le varie tipologie di scomparti, nella farmacia presa in esame è quindi stata considerata convenzionalmente, a livello informatico, un'unica tipologia di giacenza, senza curarsi in quale delle tre tipologie di scaffalature, materialmente, i farmaci si trovino. La farmacia intervistata non utilizza una funzionalità che invece il software gestionale da loro in uso permetteva (per comodità abbiamo scelto, in fase di Analisi, di ignorare anche noi

questo aspetto, e quindi di considerare una unica giacenza in cui immagazzinare i prodotti all'interno del nostro database).

La farmacia Lanzoni ci ha sinteticamente mostrato anche il funzionamento del software gestionale di loro utilizzo, denominato "Wingesfar"; l'intera amministrazione dell'attività commerciale è centralizzata in questo programma: sia per quanto riguarda l'ordine dei nuovi prodotti (attraverso i fornitori che li mettono concorrentemente a disposizione esponendone un prezzo), sia per quanto riguarda la convalida delle ricette (probabilmente attraverso API su sistema dell'SSN il quale ne detiene il funzionamento) ed ovviamente anche per: la gestione dei farmaci in giacenza, della registrazione delle vendite effettuate, della possibilità di inserire saldi a tempo, di estrarre dettagli aggiuntivi relativi un determinato farmaco, eccetera...

Anche il robot-scaffalista si interfaccia al gestionale, comunicando tutti i farmaci in ingresso ed in uscita nel deposito da lui gestito.

La farmacia ci ha successivamente descritto la gestione dei prodotti in scadenza: sebbene tutti i farmaci presentino un codice progressivo (identificato anche da un barcode) che quindi consentirebbe di identificarli uno-ad-uno registrandone per ciascuno la data di scadenza, è altresì importante tener conto che: 1. numerose farmacie non sono munite di macchinari automatici (come la Lanzoni qui presa in esame) che consentono una rapida scansione dei codici; 2. numerosi prodotti non farmaceutici non hanno un codice univoco e comunque non espongono un sistema di riconoscimento universale come invece accade per i farmaci registrati dall'AIFA.

Prendendo quindi in considerazione questi due fattori, ci è stato spiegato che la tracciabilità dei prodotti in scadenza è più grossolanamente attuabile attraverso la registrazione dei lotti di produzione; come è noto, farmaci dello stesso lotto riportano tutti la stessa scadenza. Per tale ragione, all'atto di ricezione di nuovi prodotti da parte del corriere, essi vengono registrati nel sistema tenendo anche conto del loro lotto di appartenenza in accoppiata alla comune data di scadenza.

Il farmacista intervistato ci ha infine chiarito che, sempre attraverso il gestionale da loro utilizzato, si deve tener traccia di eventuali farmaci che un cliente richiede, ma che non sono immediatamente reperibili nella giacenza. Capita spesso infatti che molti prodotti, sia per un erroneo o scarso approvvigionamento, sia per una limitata domanda di vendita che rende inutile detenere farmaci di quel tipo (rischiando di non riuscire a venderli in tempo sino alla

loro scadenza), non sono presenti in farmacia. Quando si verifica ciò con un cliente, viene generato un documento, simile ad una postilla, denominato “sospeso”. I sospesi tengono traccia del cliente e dei prodotti da lui richiesti; si accumulano fintantoché non viene effettuato un ordine generale che soddisfi tutte le quantità mancanti richieste. I tempi di consegna per i prodotti farmaceutici sono di norma molto brevi (si parla di 24-48 ore); già dal giorno successivo, non appena la merce arriva in negozio, il sospeso può essere considerato soddisfatto (riservando ovviamente le quantità di farmaci richiesti al cliente che li ha prenotati e che passerà a ritirarli).

Qui di seguito sono riportati invece aspetti tecnici, definiti per lo più per motivi legali dagli enti che regolarizzano la vendita di farmaci. Riguardano la categorizzazione e l'identificazione dei farmaci a livello nazionale (si consideri come un glossario inerente prettamente al dominio preso in esame). Le informazioni sono state parzialmente illustrate dai dipendenti della farmacia Lanzoni e integrati con materiale reperito da internet. Molte di queste informazioni torneranno utili a comporre con dati ‘reali’ il diagramma ER.

AIC – Autorizzazione all'Immissione in Commercio: provvedimento autorizzativo con il quale l'AIFA (l'Agenzia Italiana del Farmaco) certifica che un dato medicinale può essere commercializzato in Italia. Qualunque medicinale certificato da AIC deve aver subito, per legge, determinati test e deve garantire standard qualitativi durante la produzione. NB: se un medicinale contiene dei determinati principi attivi, oppure delle concentrazioni di principio attivo al di sopra di una certa soglia, può essere venduto solamente previo ottenimento dell'IAC da parte della casa farmaceutica.

Farmaco: medicinale provvisto di AIC.

NB: i farmaci che possono risultare particolarmente nocivi sono vendibili solo su ricetta medica.

Parafarmaco: medicinale o prodotto relativo alla cura/prevenzione di patologie, ma sprovvisto di AIC. Esistono diverse classi di parafarmaci, tra cui i “dispositivi medici”.

Principio attivo – Componente: principale molecola/e che assolve le funzioni per le quali un determinato prodotto è stato formulato.

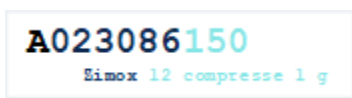
Numero progressivo: numero identificativo che rende univoca ciascuna confezione di uno specifico farmaco. Utilizzato per identificare i farmaci in cui è prevista, per legge,

un'esenzione fiscale da parte del cliente (all'atto di vendita si toglie il fustello dal farmaco e si appiccica in un raccoglitore, ciò per poterne richiedere un rimborso da parte dello Stato).

Grossista - Casa farmaceutica: fornitore dei prodotti vendibili in farmacia.

Farmacode 32: formato di codice a barre utilizzato in Italia per rappresentare il minsan (tipologia di identificativo adottato dal Ministero della Sanità italiano per registrati prodotti farmaceutici).

Esso è formato da 9 caratteri alfanumerici (esclusi i caratteri A, E, I e O) preceduti dalla lettera A



Minsan – Codice ministeriale: numero che il Ministero della Salute assegna ad ogni articolo (farmaco, dispositivo medico, presidio medico-chirurgico, parafarmaco, ecc...) nel momento in

cui viene registrato prima di essere immesso in commercio. Il codice minsan deve essere obbligatoriamente indicato su tutte le confezioni distribuite nel canale delle farmacie/parafarmacie italiano (può essere omesso per i prodotti che possiedono già un codice EAN/UPC e che non siano farmaci). Per i farmaci, minsan coincide con il numero di AIC ed è formato da quanto segue:

- i primi 6 caratteri identificano il farmaco (esempio: Zimox, Tachipirina, Gentilyn)
- gli altri 3 caratteri diversificano stessi farmaci con distinti metodi di sospensione e di dosi contenute (esempio: formulazione orale o venosa, numero di compresse o fiale)

2.2. REQUISITI IN LINGUAGGIO NATURALE

Il proprietario della Farmacia richiede che venga realizzato un sistema informatico di supporto alla conduzione della propria Farmacia. L'accesso e la manipolazione del database è limitato ai dipendenti della farmacia che possono accedere per visualizzare, inserire o aggiornare i dati.

Si vuole gestire in modo automatico i comuni prodotti che si possono trovare in farmacia, di qualunque tipologia essi siano (farmaci, parafarmaci suddivisi in classi di parafarmaci o prodotti non farmaceutici) e se ne deve amministrare la loro vendita che, in base alla tipologia di prodotto, può avvenire liberamente o necessita il possesso di ricetta medica da parte del cliente.

Nel caso di prodotti farmaceutici questi sono organizzati in base ai componenti in modo da poter riconoscere farmaci con principi attivi equivalenti.

Il proprietario vuole inoltre gestire la disponibilità dei prodotti all'interno della farmacia, ovvero tenere traccia delle quantità disponibili in magazzino e delle quantità in esposizione all'interno della farmacia, e i movimenti effettuati dai dipendenti. In base a queste devono essere gestiti gli ordini che vengono effettuati dai dipendenti ai fornitori (grossisti o case farmaceutiche) e i controlli sui prodotti ricevuti dopo gli ordini in cui si verificano le condizioni degli stessi. In caso di ricezione di prodotti non vendibili, il dipendente deve effettuarne il reso affinché siano smaltiti dai fornitori. Riguardo agli ordini si vogliono mantenere in memoria anche le relative fatture e le bolle consegnate dai fornitori.

Le disponibilità devono tenere conto anche della scadenza dei farmaci in quanto se superata i prodotti non sono più vendibili al pubblico, ma devono essere riconsegnati ai fornitori che si occuperanno dello smaltimento come accade per i prodotti difettati.

Per quanto riguarda la vendita, il proprietario ritiene necessario gestire gli scontrini e le relative voci scontrino nei quali si evidenzia la tipologia di oggetto di vendita che può essere, oltre ad un prodotto, anche un servizio (ad es. una prestazione sanitaria). Sulla vendita può essere applicato uno sconto in base alle promozioni attive.

Infine, si vogliono gestire i turni dei dipendenti della farmacia in modo tale che questi non si sovrappongano.

2.3. ESTRAZIONE DEI CONCETTI FONDAMENTALI

Si richiede la realizzazione di un database utile alla conduzione di una farmacia. Si devono gestire in modo automatico i prodotti, di qualunque tipologia essi siano (farmaci, parafarmaci suddivisi in classi di parafarmaci o prodotti non farmaceutici) e la loro vendita

che, in base alla tipologia di prodotto, può avvenire liberamente o necessita il possesso di ricetta medica da parte del cliente. Nel caso di prodotti farmaceutici, questi sono organizzabili in base ai componenti che li compongono, in modo da poter raggruppare farmaci con principi attivi equivalenti. Farmaci utilizzati per trattare stesse patologie presentano un comune 'codice d'equivalenza'. Essendo ciascun prodotto parte di un lotto, si vuole evitare la vendita di eventuali lotti considerati fallati dalla casa farmaceutica e quindi da considerare invendibili seppur materialmente ormai presenti in farmacia.

Si vuole, inoltre, gestire la disponibilità dei prodotti all'interno della farmacia, ovvero tenere traccia delle quantità disponibili. Se un cliente richiede un prodotto momentaneamente non disponibile, verrà generato un sospeso che consentirà di tener traccia degli specifici prodotti da lui richiesti e di poterne fare ordine di rifornimento. Devono quindi essere gestiti gli ordini, che vengono effettuati dai dipendenti ai fornitori (che siano grossisti o case farmaceutiche) e i relativi controlli sui prodotti ricevuti (con conseguente caricamento in giacenza) una volta ricevuta la bolla di consegna dal corriere. In caso di ricezione di prodotti non vendibili, il dipendente deve effettuarne il reso affinché siano smaltiti dai fornitori. Riguardo agli ordini si vogliono mantenere in memoria anche le relative fatture e le bolle consegnate dai fornitori e dai corrieri. Le disponibilità devono tenere conto anche della scadenza dei farmaci.

Per quanto riguarda la vendita, è necessario gestire gli scontrini e le relative voci scontrino nei quali si evidenzia la tipologia di oggetto di vendita che può essere, oltre ad un prodotto, anche un servizio o prestazione sanitaria. Sulla vendita può essere applicato uno sconto in base alle promozioni attive al momento dell'acquisto da parte del cliente.

Infine, si vogliono gestire i turni dei dipendenti della farmacia.

3. PROGETTAZIONE CONCETTUALE

Lo schema concettuale nella sua versione finale si avvarrà delle entità e delle associazioni elencate nella tabella sottostante e la sua visione integrale verrà mostrata alla fine del capitolo.

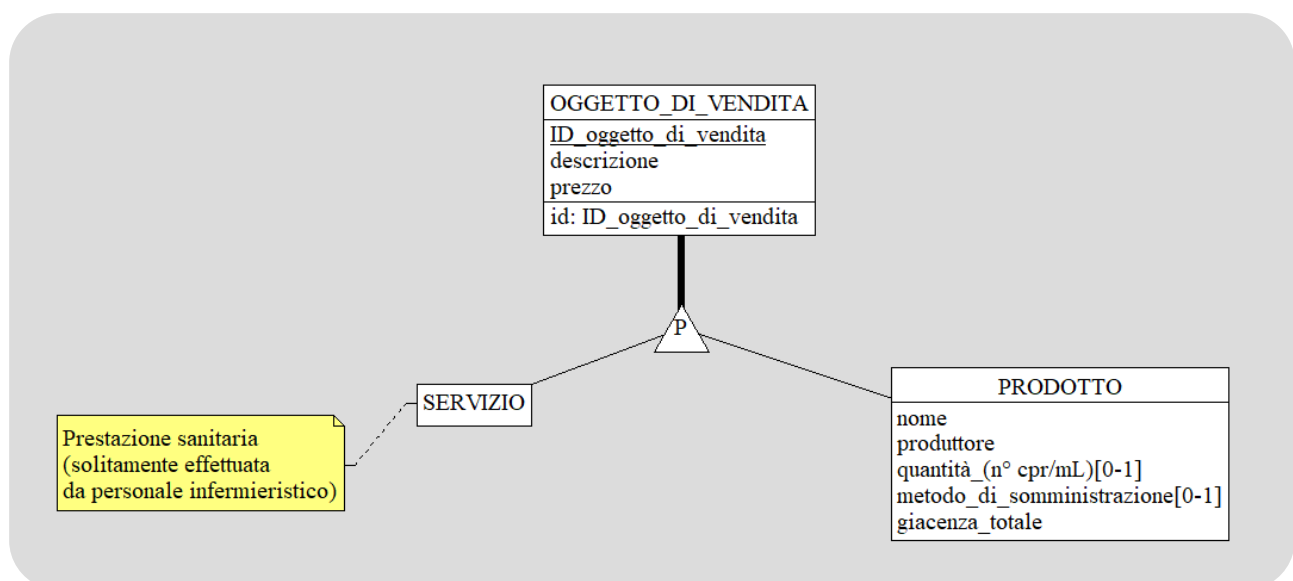
NOME	TIPO	DESCRIZIONE
PRODOTTO	E	Elemento della gerarchia oggetto di vendita
FARMACO	E	Tipologia di prodotto vendibile
PARAFARMACO	E	Tipologia di prodotto vendibile
NON_FARMACEUTICO	E	Tipologia di prodotto vendibile
CLASSIFICAZIONE	R	Lega il parafarmaco ad una classe
CLASSE_PARAFARMACO	E	Classificazione dei parafarmaci
COMPOSIZIONE	R	Lega il prodotto ai suoi componenti
COMPONENTE	E	Elementi contenuti in un prodotto
TRACCIAMENTO	R	Lega il prodotto alla propria scadenza
SCADENZA	E	Informazione sul prodotto
APPARTENENZA_LOTTO	R	Lega il prodotto al lotto di produzione
LOTTO	E	Informazione sulla produzione del prodotto
PRESCRIZIONE	R	Lega il prodotto alla ricetta medica
RICETTA_MEDICA	E	Informazione sulla vendita
PRENOTAZIONE	R	Lega il prodotto alla richiesta sospesa
SOSPESO	E	Informazione per la prenotazione
RICHIESTA	R	Lega la richiesta sospesa al cliente
APPARTENENZA	R	Lega la ricetta medica al cliente
CLIENTE	E	Individua l'acquirente
SERVIZIO	E	Elemento della gerarchia oggetto di vendita
OGGETTO_DI_VENDITA	E	Rappresenta l'oggetto vendibile
PROMOZIONE	R	Lega l'oggetto di vendita allo sconto
SCONTO	E	Rappresenta lo sconto applicabile sull'oggetto
RIFERIMENTO_SCONTRINO	R	Lega l'oggetto alla relativa voce scontrino
VOCE_SCONTRINO	E	Individua uno degli elementi dello scontrino
LISTA_SCONTRINO	R	Lega la voce scontrino al relativo scontrino
SCONTRINO	E	Elemento che certifica la vendita
SOTTOSCRIZIONE	R	Lega lo scontrino al dipendente che lo ha emesso
DIPENDENTE/OPERATORE	E	Dipendente della farmacia
ASSEGNAZIONE	R	Lega il dipendente al turno lavorativo
TURNO	E	Elemento che identifica il turno lavorativo
OPERAZIONE	R	Lega il dipendente al movimento di un prodotto
MOVIMENTO	E	Tiene traccia delle modifiche delle giacenze
MODIFICA	R	Lega il movimento al relativo prodotto
EFFETTUAZIONE	R	Lega il dipendente al reso
RESO	E	Indica il reso di alcuni prodotti
DESTINATARIO	R	Lega il reso al fornitore a cui viene fatto
FORNITORE	E	Individua un fornitore di prodotti
GROSSISTA	E	Tipologia di fornitore
CASA_FARMACEUTICA	E	Tipologia di fornitore
EFFETTUA	R	Lega l'ordine al fornitore a cui viene effettuato

ORDINE	E	Individua un ordine
CONSEGNA	R	Lega l'ordine alla relativa bolla
BOLLA	E	Bolla di consegna che può riferirsi a più ordini
PRODOTTO_CONTENUTO	R	Lega la bolla ai relativi prodotti contenuti
ACQUISTO	R	Lega l'ordine alla relativa fattura
FATTURA	E	Fattura di un ordine effettuato
RIFORNIMENTO	R	Lega l'ordine al dipendente che lo effettua
RIFORNIMENTO_ORDINE	R	Lega l'ordine ad un possibile reso
OFFERTA	R	Lega il fornitore ai prodotti da acquistare
RIFORNIMENTO_RESO	R	Lega i prodotti al reso
VOCE_ORDINE	E	Lega l'ordine alle voci relative l'ordine
LISTA_ORDINE	R	Lega la voce ordine al prodotto relativo
RIFERIMENTO_VOCE_ORDINE	R	Indica i componenti di un ordine
VOCE_BOLLA	E	Indica i componenti di una bolla
LISTA_BOLLA	R	Lega la bolla ai prodotti contenuti

Nei sotto capitoli verranno esaminati in modo più rigoroso le varie porzioni dello schema Entity-Relationship.

3.1. SVILUPPO DEL CONCETTO DI OGGETTO DI VENDITA

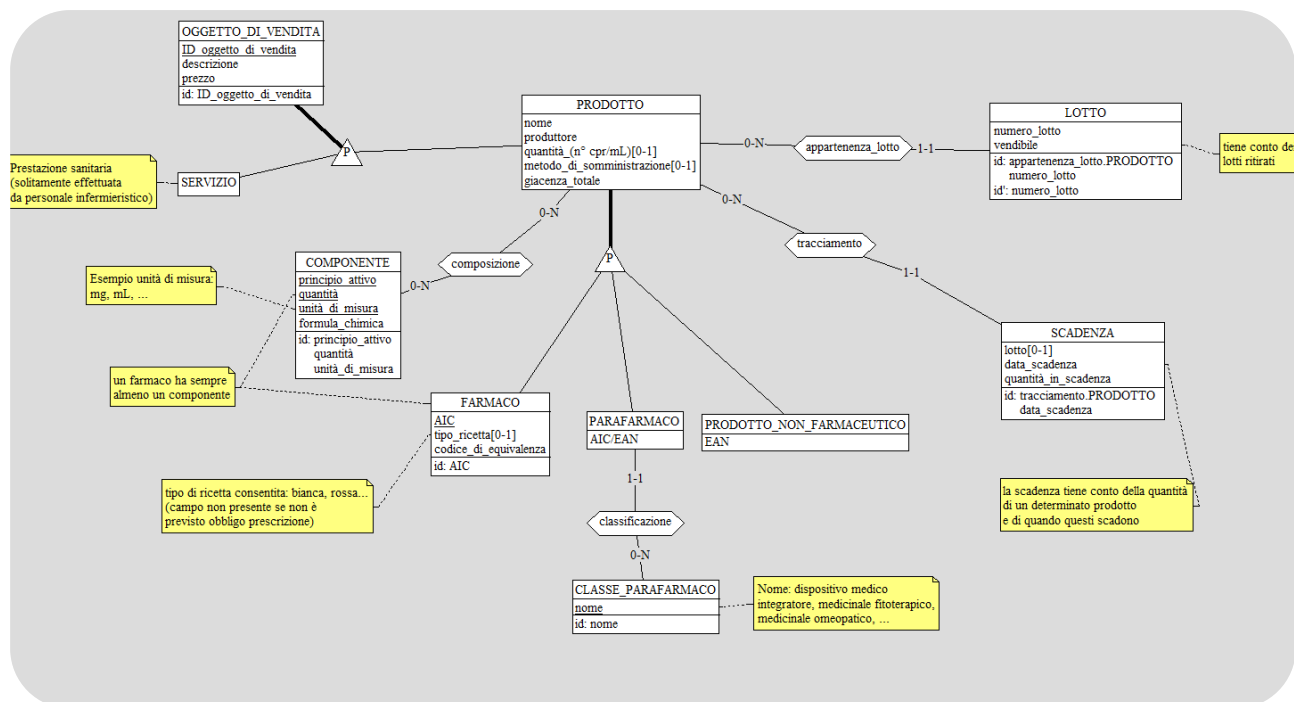
Per oggetto di vendita si intende un qualsiasi oggetto vendibile da una farmacia. Si tratta dell'entità che generalizza gli oggetti. Si ha infatti una gerarchia con copertura totale ed esclusiva che si compone di due entità figlie: SERVIZIO e PRODOTTO. Con servizio si intendono tutte le prestazioni sanitarie che possono essere effettuate all'interno di una farmacia come prestazioni di fisioterapisti ed infermieri o prestazioni analitiche di prima istanza (test autodiagnostici). Con prodotto, invece, si intendono gli articoli presenti nella tabella merceologica della farmacia che comprende, oltre ai medicinali, una serie di prodotti. Si considera quindi che la farmacia è autorizzata a vendere non solo medicinali.



3.2. SVILUPPO DEL CONCETTO DI PRODOTTO

Per prodotto, come specificato precedentemente, si intende ogni articolo presente nella tabella merceologica della farmacia. Distinguiamo all'interno dei prodotti tre sottocategorie: FARMACO, PARAFARMACO e PRODOTTO NON FARMACEUTICO. Anche in questo caso il prodotto corrisponde all'entità generale. Si ha, infatti, una gerarchia con copertura totale ed esclusiva che si compone delle tre entità indicate precedentemente. Viene modellata una gerarchia perché le tre entità presentano attributi comuni ai quali si aggiungono attributi specifici.

Al prodotto sono poi collegate le entità SCADENZA, COMPONENTE e LOTTO. L'entità scadenza ha lo scopo di mantenere in memoria la data di scadenza di un prodotto e la quantità di prodotti che presentano quella determinata scadenza. L'entità lotto serve a modellare la parte di realtà in cui un lotto di produzione può essere ritirato dalla vendita. Infine, l'entità componente prende in considerazione il principio attivo del prodotto. Si tiene in considerazione che un farmaco presenta sempre almeno un componente affinché possa ritenersi tale. Per quanto riguarda i parafarmaci e i prodotti non farmaceutici, il componente può essere considerato opzionale.

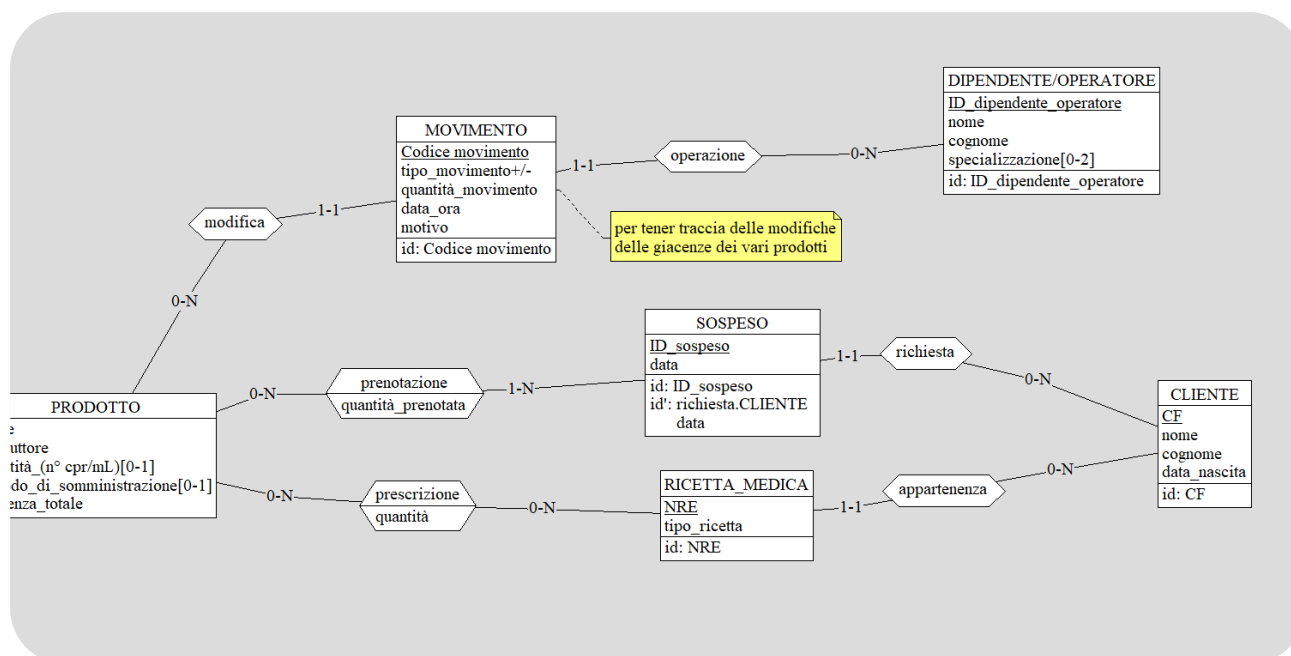


Il prodotto è l'entità centrale della base di dati e per questo presenta altre associazioni ad entità che verranno prese in esame nei prossimi sotto capitoli in quanto utili a modellare altri concetti della realtà.

3.3. SVILUPPO DEL CONCETTO DI DISPONIBILITÀ

Il concetto di disponibilità è stato sviluppando inserendo all'interno dell'entità **PRODOTTO** un attributo che ne indichi la quantità disponibile. Questa quantità può essere poi modificata mediante l'entità **MOVIMENTO** che è legata all'entità prodotto mediante un'associazione. Con l'entità movimento è possibile modificare la disponibilità indicando la quantità di prodotti da modificare, la data dell'operazione e una motivazione. Ogni movimento rimane tracciato associando ad ognuno di essi il **DIPENDENTE/OPERATORE** che lo compie.

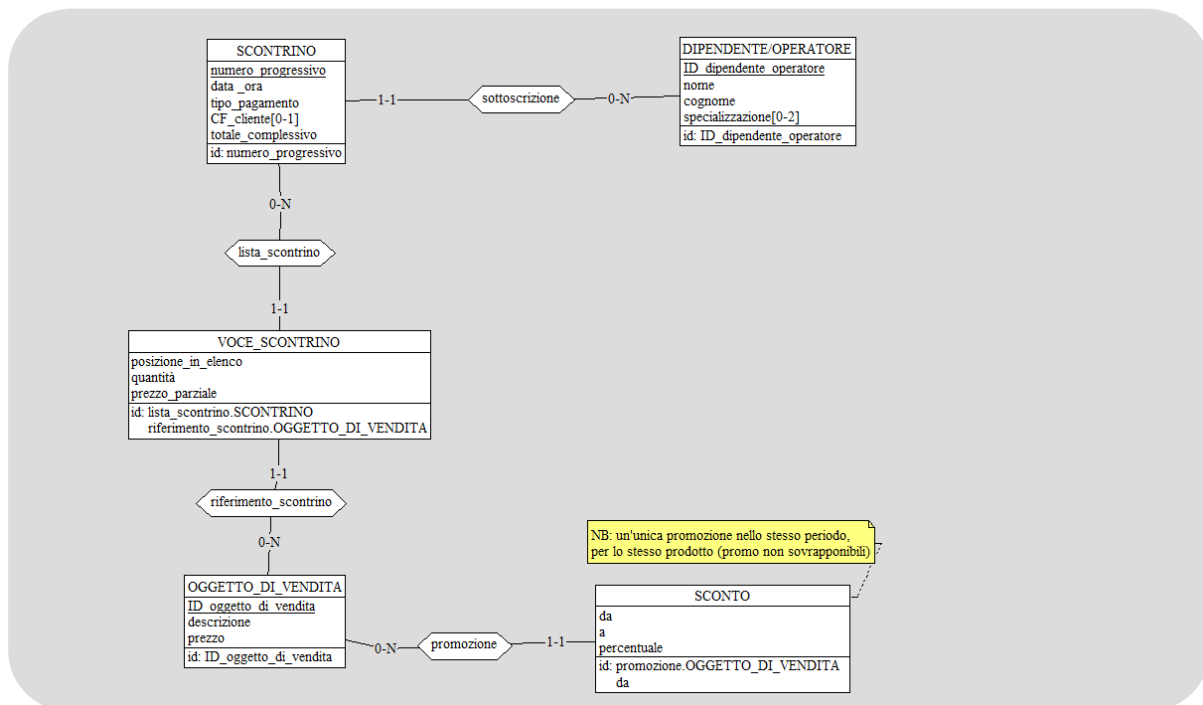
Al concetto di disponibilità viene poi aggiunta la possibilità di prenotare farmaci non disponibili. Le prenotazioni sono gestite mediante un'entità **SOSPESO** direttamente associata all'entità **PRODOTTO** e all'entità **CLIENTE**. In questo modo è possibile tenere traccia del prodotto prenotato e del relativo cliente che ha effettuato la prenotazione.



3.4. SVILUPPO DEL CONCETTO DI VENDITA

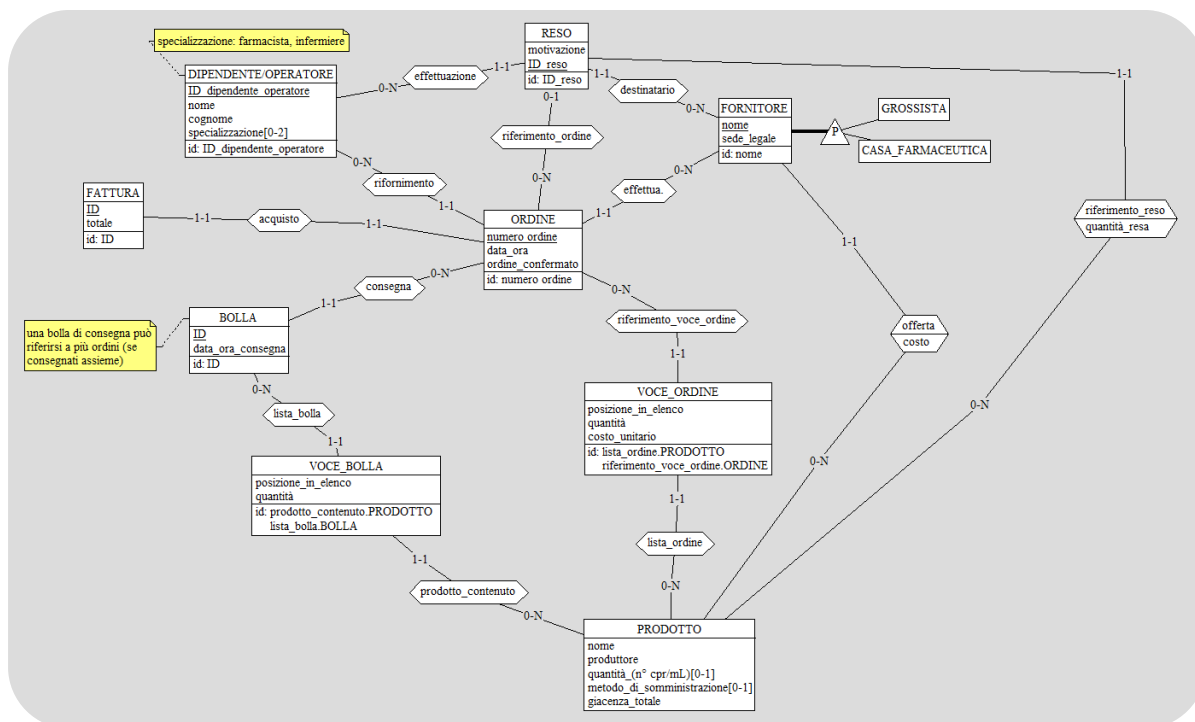
Il concetto di vendita si basa sull'entità **OGGETTO_DI_VENDITA** sul quale può essere applicato uno **SCONTO**, ovvero una promozione per l'oggetto. La promozione è una promozione unica che ha come attributi una data di inizio e una data di fine ed è relativa al prodotto in quanto identificata da un identificatore esterno. Nello schema E/R non è possibile modellare la realtà in modo tale che non ci siano sovrapposizioni. Si tratta però di un vincolo che, seppur non rappresentato nello schema, verrà preso in considerazione nelle fasi successive dello sviluppo del sistema.

Ogni oggetto di vendita viene associato ad una VOCE_SCONTRINO che va poi a comporre lo SCONTRINO finale prodotto come ricevuta della vendita. Ogni scontrino è inoltre associato al DIPENDENTE/OPERATORE che lo ha emesso.



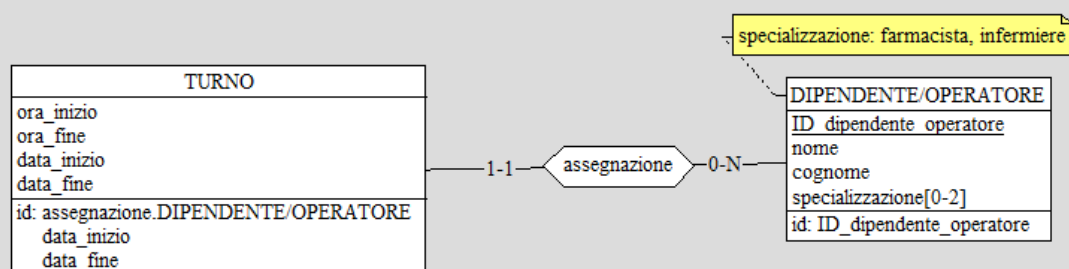
3.5. SVILUPPO DEL CONCETTO DI RIFORNIMENTO E RESO

Il concetto di rifornimento si basa sul concetto di ORDINE che viene effettuato da un DIPENDENTE/OPERATORE nel caso in cui la giacenza di un farmaco è in esaurimento o se è necessario ordinare un determinato farmaco su richiesta di un cliente. L'entità ordine è l'entità centrale di questo concetto al quale viene associata una FATTURA e una BOLLA. Ogni ordine viene effettuato ad un FORNITORE che può essere direttamente una CASA FARMACEUTICA o un GROSSISTA. Il concetto di reso viene inserito per i farmaci che presentano un difetto di produzione o che devono essere smaltiti in quanto scaduti.



3.6. SVILUPPO DEL CONCETTO DI TURNAZIONE

Il concetto di turnazione si basa sull'entità **TURNO** composta dagli attributi di inizio e di fine turno. Questa è associata al relativo **DIPENDENTE/OPERATORE**. Lo schema Entity-Relationship non ci permette di gestire la sovrapposizione dei turni. Si tratta però di un vincolo che verrà preso in considerazione in fase di progettazione logica in quanto un dipendente non può avere turni lavorativi che si sovrappongono. Verranno poi gestite anche le condizioni dei turni lavorativi in quanto un dipendente non può superare un numero definito di ore di lavoro giornaliero.



4. PROGETTAZIONE LOGICA

La traduzione dello schema concettuale in uno schema logico ha come obiettivo quello di rappresentare i concetti e i requisiti analizzati nello schema concettuale in modo più efficiente. L'efficienza è legata alle prestazioni del sistema che nella progettazione concettuale non sono state prese in considerazione. Lo schema relazionale che verrà prodotto in questa fase deve essere il più fedele possibile allo schema concettuale iniziale in quanto i due schemi sono equivalenti dal punto di vista della loro capacità informativa.

4.1. STIMA DEI VOLUMI DEI DATI

La tabella dei volumi riporta ogni concetto presente nello schema Entity-Relationship associato al loro volume, ovvero ad una stima del numero di istanze per ogni elemento. La nostra analisi si basa su una stima dei volumi annuale.

NOME	TIPO	VOLUME	NOTE
PRODOTTO	E	6500	Consideriamo un numero ristretto di prodotti disponibili
FARMACO	E	5000	Farmaci differenti
PARAFARMACO	E	1000	Parafarmaci differenti
NON_FARMACEUTICO	E	500	Non parafarmaci differenti
CLASSIFICAZIONE	R	1000	Come i prodotti parafarmaci
CLASSE_PARAFARMACO	E	50	Consideriamo un numero ristretto di classi
COMPOSIZIONE	R	13000	Un prodotto si compone in media di due principi attivi
COMPONENTE	E	10000	Consideriamo un numero ristretto di componenti
TRACCIAMENTO	R	6500	Ogni prodotto deve essere tracciato
SCADENZA	E	365	Ogni giorno è una possibile scadenza
APPARTENENZA_LOTTO	R	6500	Ogni prodotto appartiene ad un lotto
LOTTO	E	6500	Ogni prodotto appartiene ad un lotto differente
PRESCRIZIONE	R	30000	In media consideriamo 2 farmaci per ricetta
RICETTA_MEDICA	E	30000	Si stima che l'85% degli scontrini necessitano di ricetta medica
PRENOTAZIONE	R	3500	Come i sospesi
SOSPESO	E	3500	si stimano 10 sospesi al giorno
RICHIESTA	R	3500	Si stimano 10 richieste al giorno

APPARTENENZA	R	30000	Si stimano 5 ricette mediche l'anno per cliente
CLIENTE	E	6000	Si stimano tra i 15 e i 20 clienti al giorno
SERVIZIO	E	50	Servizi disponibili
OGGETTO_DI_VENDITA	E	6550	Prodotti+servizi
PROMOZIONE	R	50	Come gli sconti previsti
SCONTO	E	50	1 a settimana
RIFERIMENTO_SCONTRINO	R	375000	
VOCE_SCONTRINO	E	875000	In media 2,5 voci in ogni scontrino
LISTA_SCONTRINO	R	875000	Come voce scontrino
LISTA_ORDINE	R	5000	Ordini per prodotti di un ordine
SCONTRINO	E	350000	In media 100 scontrini al giorno
SOTTOSCRIZIONE	R	350000	Come gli scontrini
DIPENDENTE/OPERATORE	E	5	Consideriamo un numero ristretto di dipendenti
ASSEGNAZIONE	R	1750	Ogni giorno si assegna un turno a dipendente
TURNO	E	1750	Si stima che i turni variano una volta alla settimana
OPERAZIONE	R	350	Una al giorno
MOVIMENTO	E	350	Un movimento al giorno per modificare le giacenze
MODIFICA	R	350	Uguale ai movimenti
EFFETTUA	R	1000	Come numero di ordini
RESO	E	50	Si stima 1 reso a settimana
DESTINATARIO	R	50	Ogni reso ha un destinatario
FORNITORE	E	20	Numero limitato di fornitori
GROSSISTA	E	10	Compongono i fornitori
CASA_FARMACEUTICA	E	10	Compongono i fornitori
EFFETTUAZIONE	R	50	Come i resi
ORDINE	E	1000	Si stimano 2-3 ordini al giorno
RIFERIMENTO_VOCE_ORDINE	R	5000	Si stimano 5 elementi per ordine
CONSEGNA	R	1500	Ogni bolla è una consegna
BOLLA	E	1500	Mediamente un ordine arriva con 1,5 bolle
PRODOTTO_CONTENUTO	R	3000	Si stima che per ogni bolla ci siano in media 2 prodotti
ACQUISTO	R	1000	Come gli ordini
FATTURA	E	1000	Ogni ordine ha una fattura
RIFORNIMENTO	R	1000	Come gli ordini
RIFERIMENTO_ORDINE	R	50	Come i resi
OFFERTA	R	30000	Su ogni prodotto ci sono offerte in media di 5 fornitori
RIFORNIMENTO_RESO	R	50	Ogni reso si riferisce ad un prodotto

VOCE_BOLLA	E	3000	In media una bolla si compone di due elementi
LISTA_BOLLA	R	1500	Come il numero di bolle

4.2. DESCRIZIONE DELLE OPERAZIONI PRINCIPALI

Elenco delle operazioni principali:

1. Vendita di uno o più prodotti ad un cliente: consiste nella vendita di uno o più oggetti di vendita che possono essere servizi o prodotti con la relativa creazione dello scontrino.
2. Effettuazione di ordini da parte di un dipendente/operatore: consiste nell'ordine di prodotti ai relativi fornitori.
3. Effettuazione di un reso relativo ad un ordine per danneggiamento: consiste nel reso di prodotti legate a problematiche di produzione. Si riferisce solo ai prodotti e non ai servizi.
4. Effettuazione di un reso per smaltimento: consiste nello smaltimento di prodotti scaduti che devono essere riconsegnati ai fornitori per lo smaltimento.
5. Ricerca dettagliata di un prodotto: consiste nella ricerca di un prodotto nel database per verificare la disponibilità e i componenti di cui si compone. Se si tratta di un parafarmaco, viene indicata anche la sua classe di appartenenza.
6. Visualizzazione delle scadenze imminenti (batch): visualizza i prodotti che presentano una scadenza ravvicinata decisa dall'operatore.
7. Visualizzazione statistiche di vendite giornaliere: visualizza il numero di scontrini battuti in un giorno e l'incasso giornaliero.
8. Aggiunta di un cliente: consiste nel verificare che il cliente non sia già presente nel database e inserirlo al suo interno.
9. Inserimento di una bolla: consiste nell'inserire una bolla a seguito della consegna di un ordine. Una bolla non è relativa ad un ordine completo, ma alla sua consegna che può essere anche parziale.
10. Visualizzazione di ordini, bolle e fatture di un determinato giorno: rende possibile una ricerca nello storico selezionando un giorno di riferimento.
11. Visualizzazione del dipendente con vendite maggiori: permette di visualizzare il dipendente che ha effettuato maggiori vendite in un anno producendo un incasso maggiore.
12. Aggiunta di un oggetto di vendita: modifica i prodotti disponibili.
13. Visualizzazione dei fornitori che offrono un prodotto: permette di confrontare per ciascun prodotto i fornitori che presentano il prodotto da ordinare.

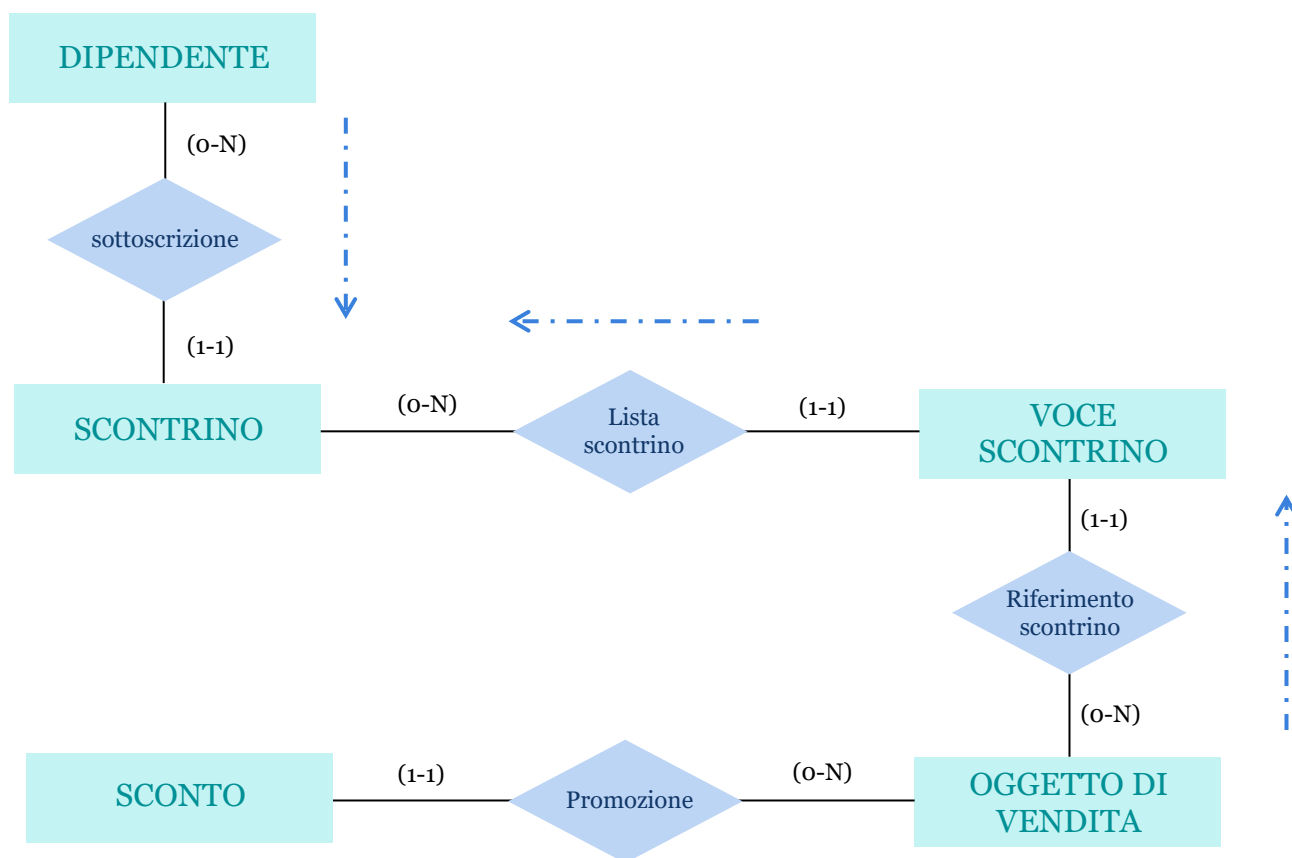
14. Notificare l'imminente esaurimento di un prodotto: la notifica dell'esaurimento di un prodotto avviene sotto una determinata soglia prestabilita.
15. Visualizzazione mese con massimo guadagno: permette di visualizzare il mese in cui si la farmacia ha avuto massimo guadagno durante l'anno.
16. Visualizzazione dei farmaci equivalenti di un prodotto: si verifica attraverso il codice di equivalenza.
17. Inserimento di una fattura: serve a confermare l'ordine dopo che è stato effettuato.

4.3. STIMA FREQUENZA DELLE OPERAZIONI PRINCIPALI

N.	NOME OPERAZIONE	FREQUENZA
1	Vendita di prodotti (creazione scontrino)	50 al giorno
2	Effettuazione di un ordine	15 alla settimana
3	Effettuazione di un reso su ordine per danneggiamento	1 alla settimana
4	Effettuazione di un reso per smaltimento	1 alla settimana
5	Ricerca dettagliata di un prodotto	100 al giorno
6	Visualizzazione delle scadenze imminenti	1 al giorno
7	Visualizzazione statistiche di vendite giornalieri	1 al giorno
8	Aggiunta di un cliente	15 al giorno
9	Inserimento di una bolla	4 al giorno
10	Visualizzazione di ordini, bolle e fatture di un giorno	5 alla settimana
11	Visualizza dipendente con entrate maggiori	1 alla settimana
12	Aggiunta di un oggetto di vendita	1 al mese
13	Visualizzazione dei fornitori che offrono un prodotto	50 alla settimana
14	Notificare l'imminente esaurimento di un prodotto (batch)	1 al giorno
15	Visualizzazione mese con massimo guadagno	1 all'anno
16	Visualizzazione dei farmaci equivalenti	50 al giorno
17	Inserimento di una fattura	15 alla settimana

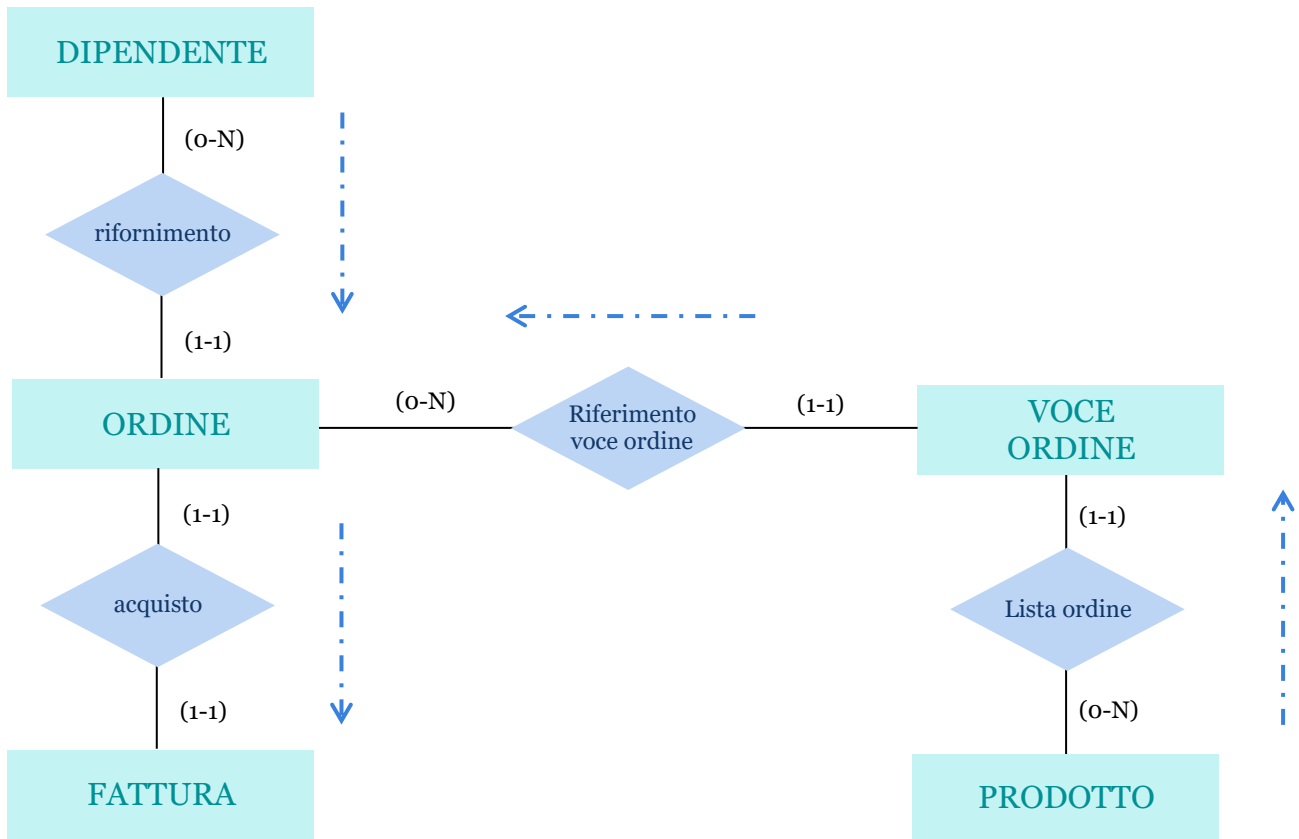
4.4. SCHEMI DI NAVIGAZIONE E TABELLE DEGLI ACCESSI

Op. 1: Vendita di prodotti (creazione di uno scontrino)



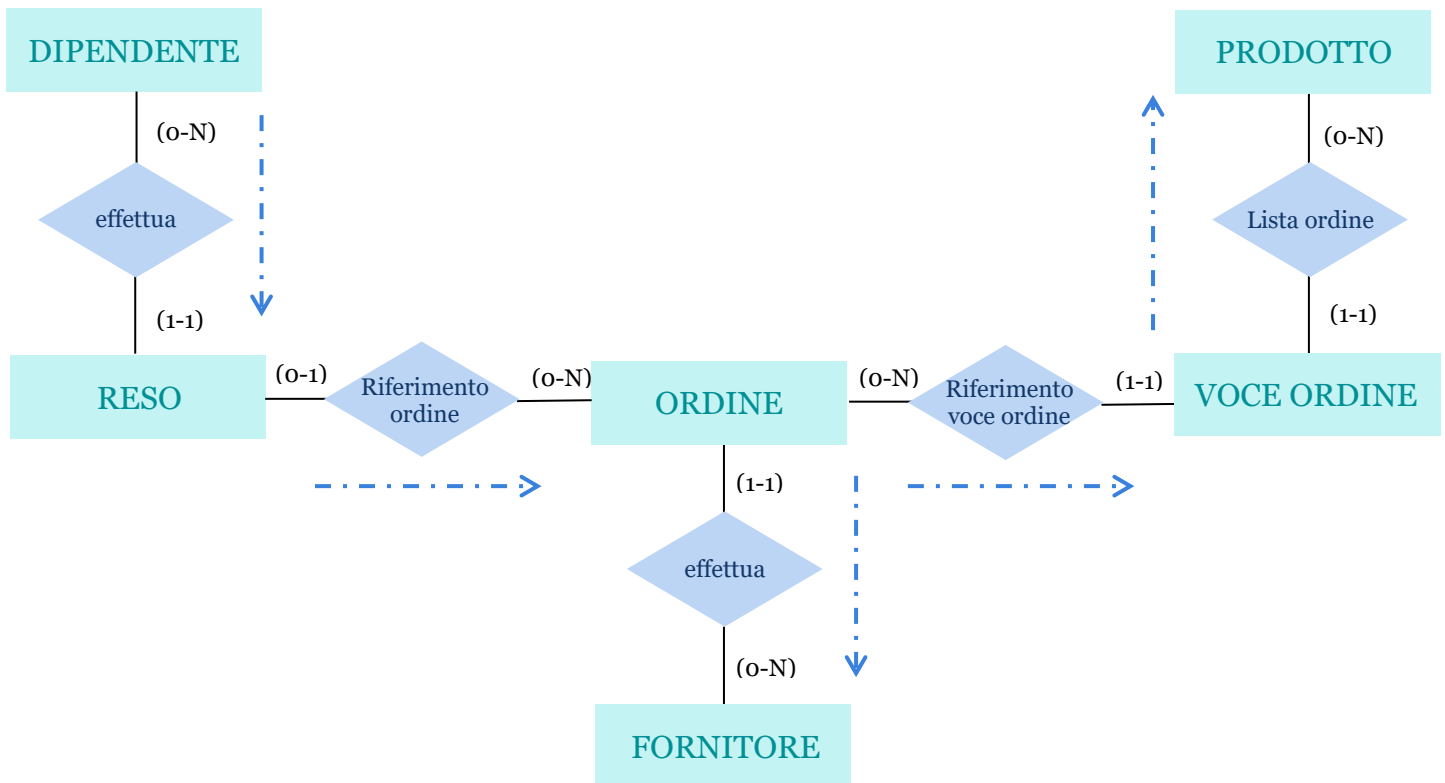
Concetto	Costrutto	Accessi	Tipo di accesso
Dipendente	E	1	L
Scontrino	E	1	S
Lista scontrino	R	2.5	S
Oggetto di vendita	E	2.5	L
Oggetto di vendita	E	2.5	S
Voce scontrino	E	2.5	S
Sconto	E	1	L
TOTALE: $4.5L + 6S = (4.5+12) * 50 = 825$ al giorno			

Op. 2: Effettuazione di un ordine



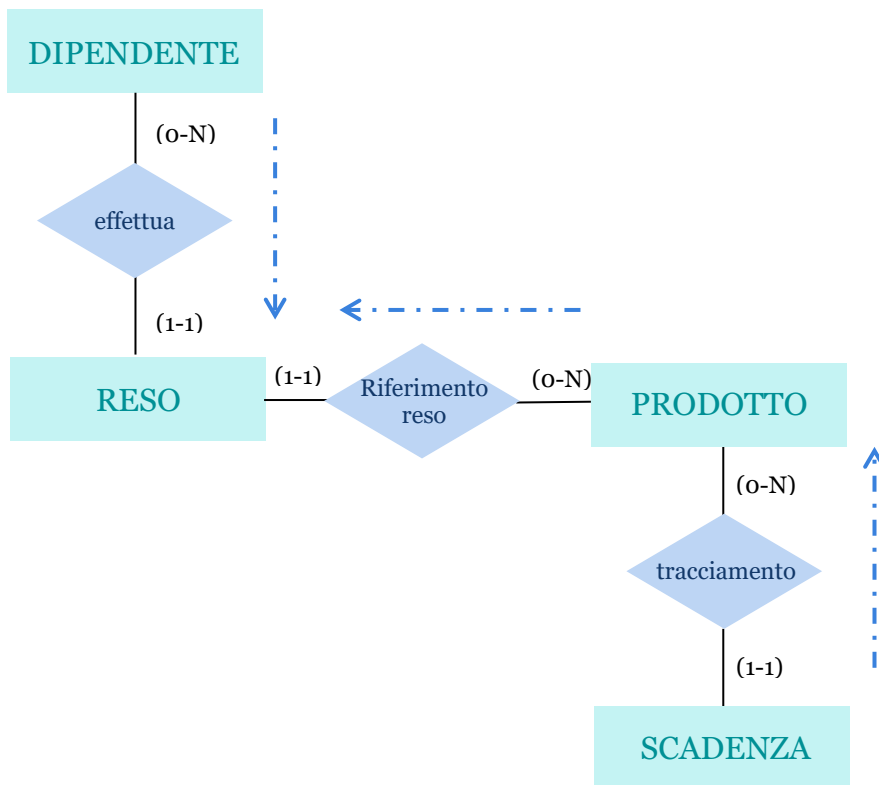
Concetto	Costrutto	Accessi	Tipo di accesso
Dipendente	E	1	L
Ordine	E	1	S
Riferimento voce ordine	R	5	S
Prodotto	E	5	L
Voce ordine	E	5	S
Fattura	E	1	S
TOTALE: $6L + 12S = (6+24) * 15 = 450$ a settimana			

Op. 3: Effettuazione di un reso su ordine per danneggiamento



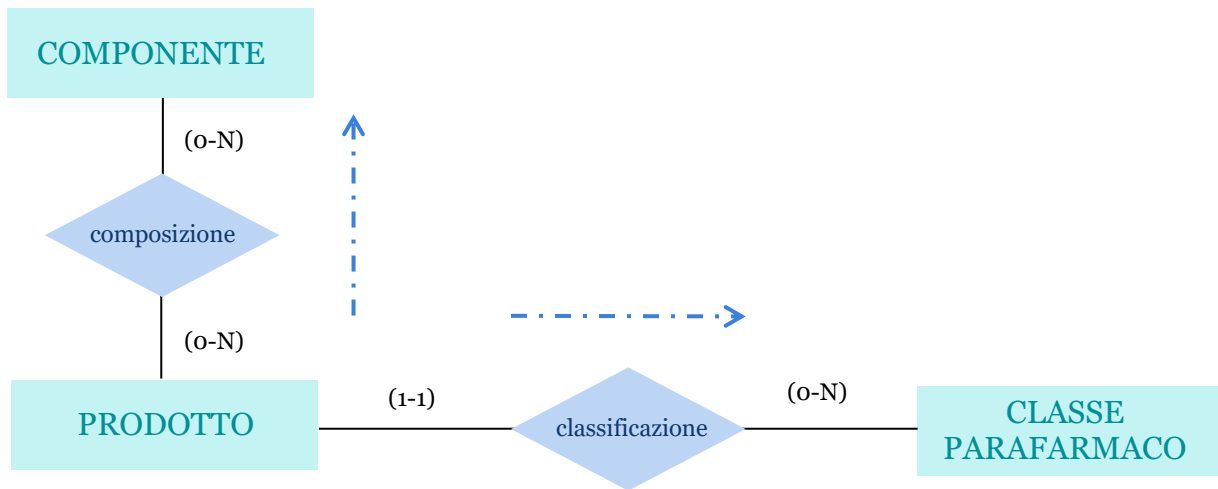
Concetto	Costrutto	Accessi	Tipo di accesso
Dipendente	E	1	L
Ordine	E	1	L
Voce ordine	E	1	L
Reso	E	1	S
Fornitore	E	1	L
Prodotto	E	1	S
Prodotto	E	1	L
TOTALE: 5L + 2S = (5+4) =9 a settimana			

Op. 4: Effettuazione di un reso per smaltimento



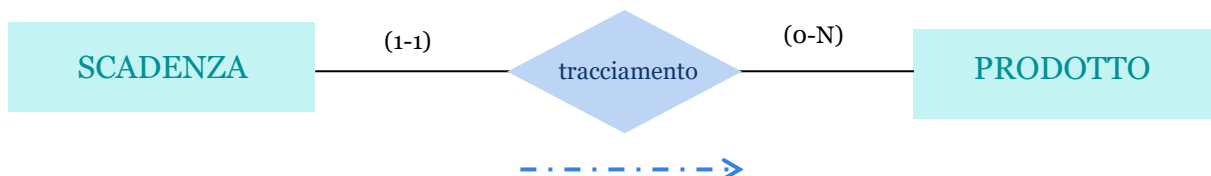
Concetto	Costrutto	Accessi	Tipo di accesso
Dipendente	E	1	L
Prodotto	E	1	L
Prodotto	E	1	S
Scadenza	E	1	L
Fornitore	E	1	L
Reso	E	1	S
TOTALE: 4L + 1S = (4+2) =6 a settimana			

Op. 5: Ricerca dettagliata di un prodotto



Concetto	Costrutto	Accessi	Tipo di accesso
Prodotto	E	1	L
Componente	E	1	L
Classe parafarmaco	E	1	L
TOTALE: $3L = 3 \cdot 100 = 300$ al giorno			

Op. 6: Visualizzazione delle scadenze imminenti (batch)



Concetto	Costrutto	Accessi	Tipo di accesso
Scadenza	E	1	L
Prodotto	E	1	L
TOTALE: $2L = 2$ al giorno			

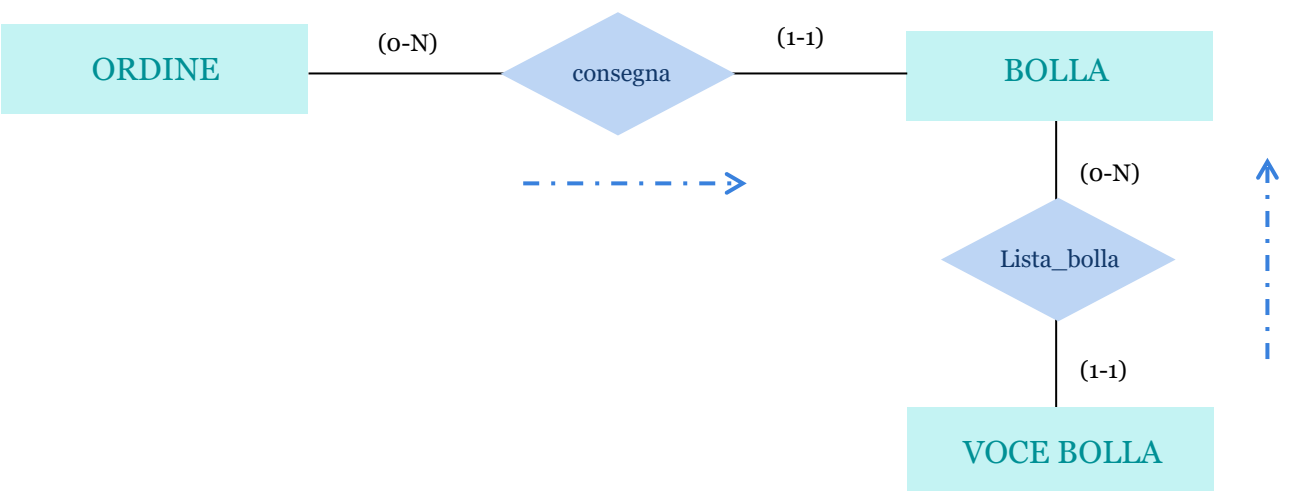
Op. 7: Visualizzazione statistiche di vendita giornaliera

Concetto	Costrutto	Accessi	Tipo di accesso
Scontrino	E	1	L
TOTALE: 1L = 1 al giorno			

Op. 8: Aggiunta di un cliente

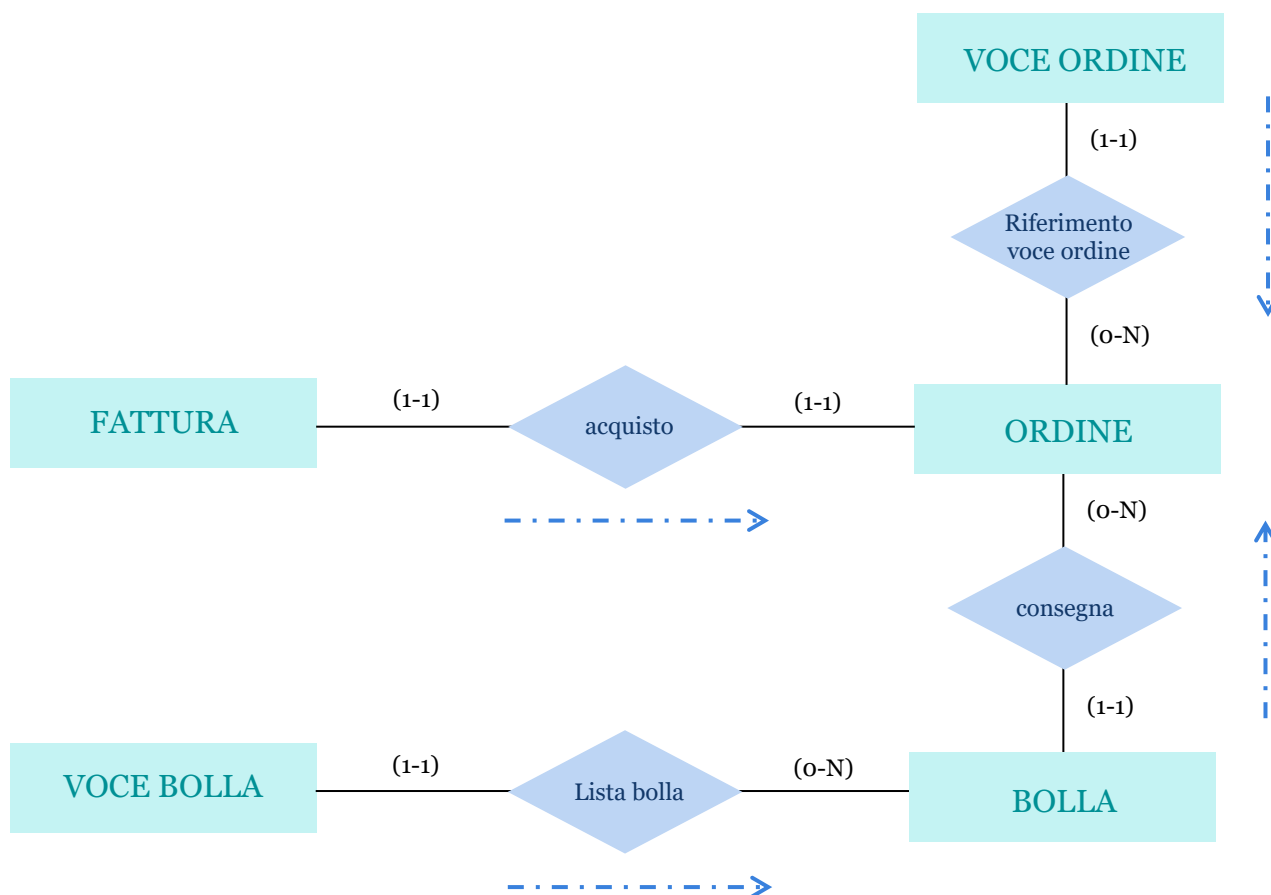
Concetto	Costrutto	Accessi	Tipo di accesso
Cliente	E	1	L
Cliente	E	1	S
TOTALE: 3L = 3*15=75 al giorno			

Op. 9: Inserimento di una bolla



Concetto	Costrutto	Accessi	Tipo di accesso
Bolla	E	1	L
Ordine	E	1	L
Bolla	E	1	S
Voce bolla	E	2	S
Lista bolla	R	3000 (voce bolla) / 1500 (bolla)	S
TOTALE: 2L + 4S = (2+8) *4=40 al giorno			

Op. 10: Visualizzazione di ordini, bolle e fatture di un giorno



Concetto	Costrutto	Accessi	Tipo di accesso
Ordine	E	1	L
Riferimento voce ordine	R	5000 (voce ordine) / 1000 (ordine)	L
Bolla	E	1	L
Consegna	R	1500 (consegna)/ 1000(ordine)	L
Voce bolla	E	2	L
Lista bolla	R	3000 (voce bolla) / 1500 (bolla)	L
Fattura	E	1	L
Acquisto	R	1000 (acquisto)/ 1000(ordine)	L
TOTALE: 14.5 L *5= 72.5 a settimana			

Op. 11: Visualizzazione dipendente con entrate maggiori



Concetto	Costrutto	Accessi	Tipo di accesso
Dipendente	E	1	L
Scontrino	E	6730 (scontrini annuali/52 settimane)	L
TOTALE: 6731L a settimana			

Op. 12: Aggiunta di un oggetto di vendita

Concetto	Costrutto	Accessi	Tipo di accesso
Oggetto di vendita	E	1	L
Oggetto di vendita	E	1	S
Prodotto	E	1	S
servizio	E	1	S
TOTALE: 1L + 3S = (1+6) = 7 al mese			

Op. 13: Visualizzazione dei fornitori che offrono un prodotto



Concetto	Costrutto	Accessi	Tipo di accesso
Prodotto	E	1	L
Fornitore	E	1	L
Offerta	R	5	L
TOTALE: 7L = 7*50= 350 a settimana			

Op. 14: Notificare l'imminente esaurimento di un prodotto (batch)

Concetto	Costrutto	Accessi	Tipo di accesso
Prodotto	E	1	L
TOTALE: 1L = 1 al giorno			

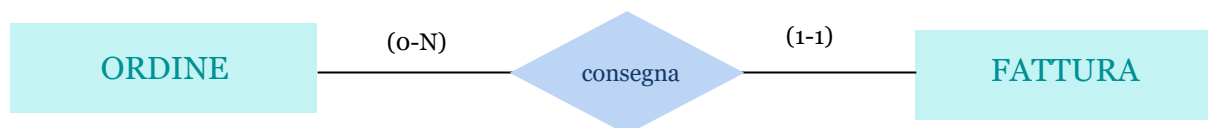
Op. 15: Visualizzazione mese con massimo guadagno

Concetto	Costrutto	Accessi	Tipo di accesso
Scontrino	E	1	L
TOTALE: 1L = 1 al mese			

Op. 16: Visualizzazione dei farmaci equivalenti di un prodotto

Concetto	Costrutto	Accessi	Tipo di accesso
Prodotto	E	1	L
Composizione	R	2	L
TOTALE: 3L = 3*50 = 150 al giorno			

Op. 17: Inserimento di una fattura



Concetto	Costrutto	Accessi	Tipo di accesso
Fattura	E	1	S
Ordine	E	1	L
Ordine	E	1	S
TOTALE: 1L + 2S = (1+4) *15 = 75 a settimana			

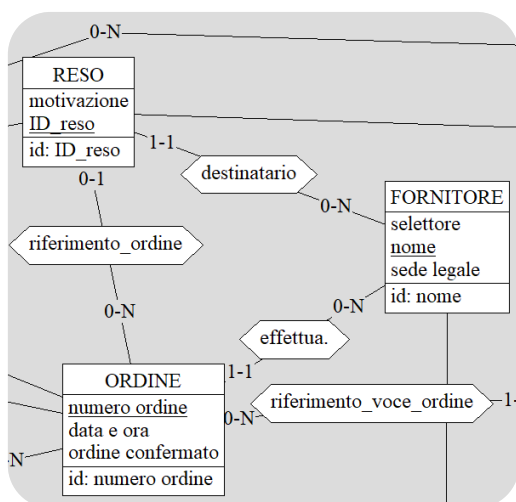
4.5. RAFFINAMENTO DELLO SCHEMA

Il processo di raffinamento dello schema consiste nell'eliminazione delle strutture che non possono essere rappresentate attraverso un modello relazionale come gerarchie e attributi compositi.

Nel nostro caso non sono presenti attributi compositi, per cui sarà sufficiente eliminare le gerarchie per ottenere uno schema raffinato. Sono presenti tre gerarchie:

- FORNITORE come entità padre e GROSSISTA e CASA_FARMACEUTICA come entità figlie;
- PRODOTTO come entità padre e FARMACO, PARAFARMACO e PRODOTTO_NON_FARMACEUTICO come entità figlie;
- OGGETTO_DI_VENDITA come entità padre e PRODOTTO e SERVIZIO come entità figlie.

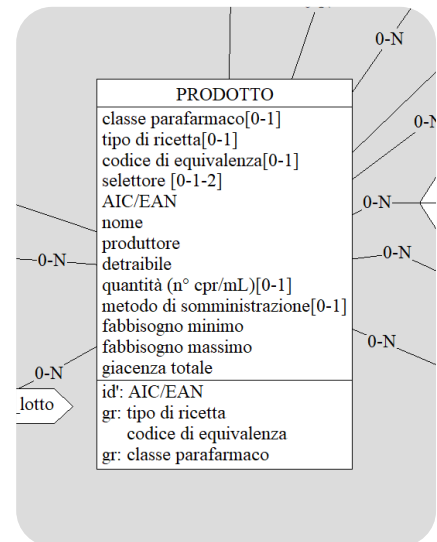
Nel caso della gerarchia su fornitore si può applicare un collasso verso l'alto inserendo un attributo all'entità padre di tipo selettore che ci permette di identificare il tipo di fornitore che si inserisce. L'attributo selettore è un attributo di tipo binario che assume valore zero se il fornitore è un grossista e valore uno se il fornitore è una casa farmaceutica.



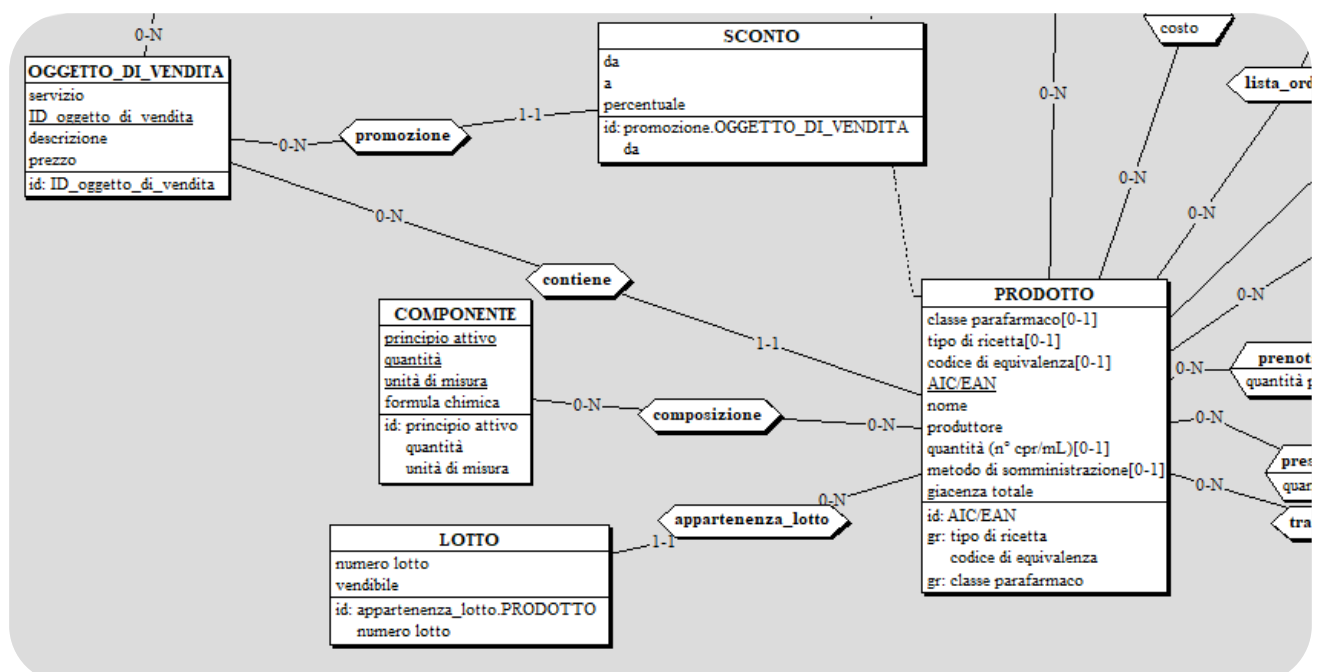
Nel caso della gerarchia su prodotto si può applicare un collasso verso l'alto inserendo un attributo all'entità padre di tipo selettore che ci permette di identificare il tipo di prodotto che si inserisce. L'attributo selettore assumere tre valori differenti: valore zero se è un farmaco, valore uno se è un parafarmaco, valore due se è un prodotto non farmaceutico. Nell'applicare il collasso verso l'alto occorre inserire gli attributi delle sotto entità sull'entità principale. Dato che tutte le sotto entità presentano un codice di tipo AIC/EAN, questo può essere inserito come attributo obbligatorio nell'entità padre. Inoltre, dato che si tratta di codici univoci, può essere considerato come identificatore dell'entità prodotto. Per quanto riguarda gli attributi delle entità figlie che non sono comuni tra loro, occorre procedere nel

seguente modo: l'entità farmaco presenta gli attributi tipo ricetta e codice di equivalenza che vengono inseriti come attributi opzionali dell'entità padre e raggruppati in un gruppo. Questi attributi saranno obbligatori solo nel caso in cui l'attributo selettore indica che il prodotto inserito è un farmaco, ovvero assume valore zero.

Infine, l'entità figlia parafarmaco presenta un'associazione con l'entità classe parafarmaco. Per raffinare lo schema è possibile sostituire questa associazione mediante l'aggiunta di un attributo opzionale sull'entità padre che indichi la classe di appartenenza del parafarmaco. Si tratta di un attributo che sarà obbligatorio solo nel caso in cui il selettore indica che il prodotto inserito è un parafarmaco, ovvero se assume valore uno.



Nel caso della gerarchia su oggetto di vendita la situazione è più complessa perché si ha come sotto entità prodotto che rappresenta l'entità principale del progetto. In questo caso si utilizza una soluzione ibrida tra collasso verso l'alto e sostituzione con associazioni. L'entità oggetto di vendita presenta due entità figlie: servizio e prodotto. Per l'entità servizio è sufficiente utilizzare il collasso verso l'alto e inserire in oggetto di vendita un attributo di tipo booleano chiamato servizio che indica se l'oggetto di vendita inserito è un servizio (valore 1) o un prodotto (valore 0). Per l'entità prodotto è invece necessario sostituire la gerarchia mediante un'associazione.



4.6. ANALISI DELLE RIDONANZE

Prendendo in considerazione l'attributo totale complessivo nella tabella SCONTRINO, si può osservare che il totale complessivo utile alla prima operazione si può ottenere in due modi distinti:

- Sommando i prezzi unitari delle VOCI SCONTRINO;

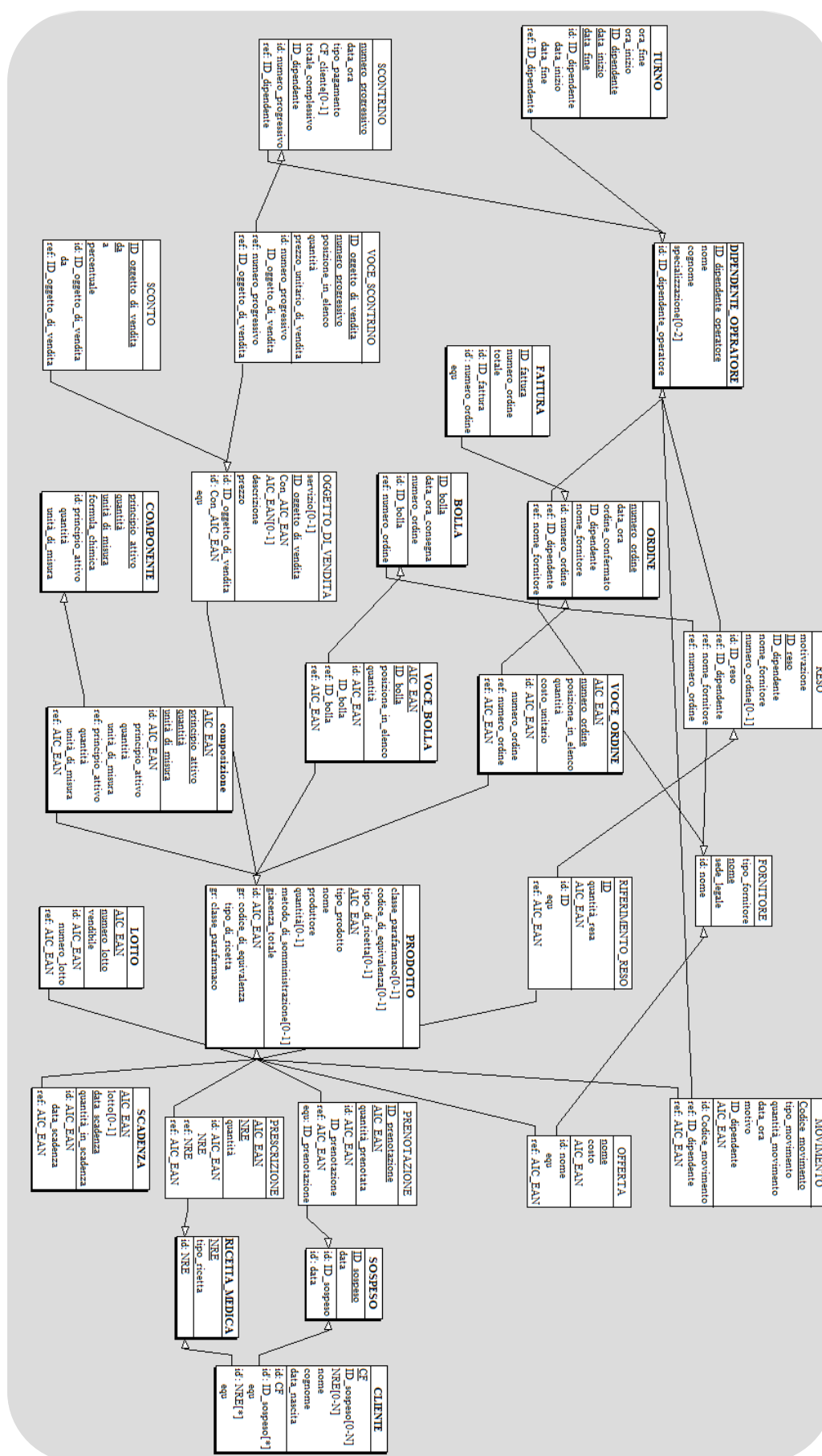
Concetto	Costrutto	Accessi	Tipo di accesso
Dipendente	E	1	L
Scontrino	E	1	S
Lista scontrino	R	2.5	S
Oggetto di vendita	E	2.5	L
Oggetto di vendita	E	2.5	S
Voce scontrino	E	2.5	S
Sconto	E	1	L
TOTALE: $4.5L + 6S = (4.5+12) * 50 = 825$ al giorno			

- Leggendo il totale complessivo nello SCONTRINO (risulta più conveniente).

Concetto	Costrutto	Accessi	Tipo di accesso
Dipendente	E	1	L
Scontrino	E	1	L
TOTALE: $2L = (2) * 50 = 100$ al giorno			

4.7. TRADUZIONE ENTITÀ E ASSOCIAZIONI IN RELAZIONI

TURNO (data_inizio, data_fine, ID_dipendente, ora_inizio, ora_fine)
DIPENDENTE/OPERATORE (ID_dipendente_operatore, nome, cognome, specializzazione)
SCONTRINO (numero_progressivo, data_ora, tipo_pagamento, CF_cliente*, ID_dipendente*, totale_complessivo)
VOCE_SCONTRINO (ID_oggetto_di_vendita*, numero_progressivo*, posizione_in_elenco, quantità, prezzo_parziale)
OGGETTO DI VENDITA (id_oggetto_di_vendita, AIC_EAN, servizio, descrizione, prezzo)
SCONTO (Da, a, percentuale, ID_oggetto_di_vendita*)
ORDINE (numero_ordine, data_ora, ordine_confermato, ID_dipendente*, nome_fornitore*)
FATTURA (ID_fattura, numero_ordine*, totale)
BOLLA (ID_bolla, data_ora_consegna, numero_ordine*)
VOCE_BOLLA (AIC_EAN*, id_bolla*, posizione_in_elenco, quantità)
RESO (ID_reso, ID_dipendente*, nome_fornitore*, numero_ordine*, motivazione)
FORNITORE (nome, sede_legale, tipo_fornitore)
VOCE_ORDINE (AIC_EAN*, numero_ordine*, posizione_in_elenco, costo_unitario, quantità)
PRODOTTO (AIC/EAN, nome, produttore, quantità, metodo_di_somministrazione, giacenza_totale, codice_di_equivalenza, tipo_di_ricetta, tipo_prodotto, classe_parafarmaco)
COMPONENTE (principio_attivo, quantità, unità_di_misura, principio_attivo)
COMPOSIZIONE (AIC-EAN*, principio_attivo, quantità, unità_di_misura)
LOTTO (numero_lotto, AIC_EAN*, vendibile)
MOVIMENTO (codice_movimento, tipo_movimento, quantità_movimento, data_ora, motivo, AIC_EAN*, ID_dipendente*)
SCADENZA (AIC_EAN*, lotto*, data_di_scadenza, quantità_in_scadenza)
RICETTA_MEDICA (NRE, tipo_ricetta)
PRESCRIZIONE (AIC_EAN*, NRE*, quantità)
CLIENTE (CF, ID_sospeso*, NRE*, nome, cognome, data_nascita)
SOSPESO (ID_sospeso, data)
PRENOTAZIONE (ID_prenotazione, AIC_EAN, quantità_prenotata)
RIFERIMENTO_RESO (ID, AIC_EAN*, quantità_resa)
OFFERTA (nome_fornitore*, AIC_EAN*, costo)



5. PROGETTAZIONE FISICA

5.1. TRADUZIONE DELLE OPERAZIONI IN QUERY SQL

Per la traduzione delle operazioni in query SQL abbiamo utilizzato alcune funzionalità di MySQL tra cui la possibilità di creare funzioni da richiamare e la possibilità di creare delle procedure che permettono di chiedere in input dei parametri ed eseguire le operazioni su di essi per poi restituire un parametro in output.

Dato che in molte operazioni ci veniva richiesto di ricavare il dipendente/operatore dalla tabella dei turni dei dipendenti, abbiamo dichiarato una funzione *CurrentOperator* da richiamare nelle varie query. La funzione è implementata nel seguente modo:

```
1 • CREATE DEFINER='root'@'localhost' FUNCTION `CurrentOperator`() RETURNS int
2     DETERMINISTIC
3     BEGIN
4         DECLARE CurrentOperator int ;
5         SELECT ID_dipendente_operatore INTO CurrentOperator
6         FROM ( dipendente_operatore D JOIN turno T ON (D.ID_dipendente_operatore = T.ID_dipendente))
7         WHERE T.data_inizio= current_date()
8         AND current_time() between convert(T.ora_inizio, time)
9         AND convert(T.ora_fine, time);
10
11     RETURN CurrentOperator;
12     END
```

OP 1 – Vendita di prodotti (creazione scontrino)

La store procedure *AggiuntaScontrino* crea un nuovo scontrino che presenta un numero progressivo calcolato automaticamente dal sistema, inserendo la data corrente e l'operatore di turno della farmacia. Vengono poi chiesti in input il tipo di pagamento e il codice fiscale del cliente che può essere opzionale. Lo scontrino viene creato impostando il totale complessivo pari a zero.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `AggiuntaScontrino`( in Pagamento char(100), CodiceFiscale char(16) )
2     BEGIN
3
4         -- creazione scontrino --
5         DECLARE CountNumScontrini int;
6         SET CountNumScontrini = (SELECT COUNT(*)
7         FROM scontrino);
8         INSERT INTO scontrino ( numero_progressivo, data_ora, tipo_pagamento, CF_cliente, totale_complessivo, ID_dipendente)
9         VALUES (CountNumScontrini+1, current_date() ,Pagamento , CodiceFiscale ,0.00,(CurrentOperator()));
10    END
```

Successivamente viene creata una voce scontrino per ogni oggetto di vendita da inserire nello scontrino. Prima di inserire la voce scontrino viene effettuato un controllo sulla giacenza in modo che non sia possibile vendere una quantità maggiore di prodotto rispetto a quella che si ha disponibile. Questo controllo viene effettuato mediante la store procedure *CheckGiacenza* che restituisce come output un valore booleano.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CheckGiacenza` ( in QuantitàOggettoInput int, ID_Oggetto bigint , OUT result BOOL)
2 BEGIN
3     DECLARE Quantity int;
4     DECLARE giacenza_totale int;
5
6     SELECT P.giacenza_totale into Quantity
7     FROM prodotto as P, oggetto_di_vendita as O
8     WHERE P.AIC_EAN = O.AIC_EAN
9     AND O.ID_oggetto_di_vendita = ID_Oggetto;
10
11     SET giacenza_totale = Quantity - QuantitàOggettoInput;
12
13     IF giacenza_totale < 0 THEN
14         SET RESULT = FALSE;
15     ELSE
16         SET RESULT = TRUE;
17     END IF;
18 END
```

Una volta effettuato il controllo, viene creata la voce scontrino mediante la store procedure *CreazioneVoceScontrino*.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CreazioneVoceScontrino` ( in ID_Oggetto bigint, QuantitàOggetto int)
2 BEGIN
3     DECLARE NumProg int;
4     DECLARE Posizione int;
5     SET NumProg = (SELECT COUNT(*) FROM scontrino);
6     SET Posizione = (SELECT COUNT(*) FROM voce_scontrino WHERE numero_progressivo = NumProg);
7     -- creazione voce scontrino --
8     INSERT INTO voce_scontrino (ID_oggetto_di_vendita, numero_progressivo, posizione_in_elenco, quantità, prezzo_parziale)
9     VALUES ( ID_Oggetto, NumProg , Posizione+1, QuantitàOggetto,
10     (SELECT O.prezzo*QuantitàOggetto
11     FROM oggetto_di_vendita AS O
12     WHERE ID_Oggetto= O.ID_oggetto_di_vendita));
13     -- update del totale complessivo nello scontrino --
14     UPDATE scontrino SET totale_complessivo = totale_complessivo +
15     (SELECT O.prezzo*QuantitàOggetto
16     FROM oggetto_di_vendita AS O
17     WHERE ID_Oggetto= O.ID_oggetto_di_vendita) WHERE numero_progressivo= NumProg;
18     -- update della giacenza nella tabella prodotto --
19     UPDATE prodotto SET giacenza_totale = giacenza_totale - QuantitàOggetto WHERE AIC_EAN =
20     (SELECT AIC_EAN
21     FROM oggetto_di_vendita
22     WHERE ID_oggetto_di_vendita = ID_Oggetto);
23 END
```

All'interno della store procedure, oltre alla creazione della voce scontrino, viene effettuato l'update del costo complessivo del relativo scontrino e viene effettuato l'update della giacenza dell'oggetto venduto.

OP 2 – Effettuazione di un ordine

La store procedure *EffettuazioneOrdine* permette la creazione di un nuovo ordine che presenta un numero ordine progressivo, la data corrente, il dipendente di turno e un valore booleano ordine confermato di default a zero (non confermato). Il nome del fornitore viene preso in input e inserito nell'istanza del nuovo ordine.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `EffettuazioneOrdine`( in Fornitore char(100))
2 BEGIN
3     DECLARE NumOr int;
4     SET NumOr = ( SELECT COUNT(*) FROM ordine as O );
5     INSERT INTO ordine (numero_ordine, data_ora, ordine_confermato, ID_dipendente, nome_fornitore)
6     VALUES( NumOr+1, curdate(), '0', (CurrentOperator()), Fornitore);
7 END
```

Successivamente viene creata una voce ordine per ogni oggetto di vendita da inserire nell'ordine. Prima di inserire la voce ordine viene effettuato un controllo sul catalogo in modo da verificare che il prodotto inserito sia presente all'interno del catalogo della farmacia. In caso negativo, occorre inserire il prodotto all'interno del catalogo mediante l'operazione 12 (aggiunta di un oggetto di vendita). La store procedure restituisce un valore booleano che rappresenta la presenza o meno nel catalogo dell'oggetto di vendita.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CheckCatalogo`( in aic_ean char(15), out result bool)
2 BEGIN
3     DECLARE c int;
4     SELECT COUNT(*) into c
5     FROM offerta O
6     WHERE O.AIC_EAN = aic_ean;
7
8     IF c = 0 THEN
9         SET result = FALSE;
10    ELSE
11        SET result = TRUE;
12    END IF;
13 END
```

Una volta effettuato il controllo, è possibile inserire la voce ordine mediante la store procedure *CreazioneVoceOrdine*.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CreazioneVoceOrdine` ( in AIC_EAN char(15), quantità int)
2 BEGIN
3     DECLARE NumOr int;
4     DECLARE Posizione int;
5     SET NumOr = ( SELECT COUNT(*) FROM ordine as O );
6     SET Posizione = (SELECT COUNT(*) FROM voce_ordine WHERE numero_ordine = NumOr);
7
8     INSERT INTO voce_ordine (AIC_EAN, numero_ordine, posizione_in_elenco, quantità, costo_unitario)
9     VALUES(
10     (SELECT AIC_EAN
11     FROM (offerta OFE JOIN prodotto P ON (OFE.AIC_EAN = P.AIC_EAN))
12     WHERE OFE.nome = P.prodotto
13     AND OFE.AIC_EAN = AIC_EAN)
14     , NumOr, Posizione+1, quantità,
15     (SELECT costo
16     FROM offerta as O
17     WHERE O.AIC_EAN= AIC_EAN)
18     );
19 END

```

OP 3 – Effettuazione di un reso su ordine per danneggiamento

Il reso per danneggiamento viene effettuato nel momento in cui un ordine arriva danneggiato e i prodotti non risultano conformi alla vendita. Prima di effettuare il reso occorre verificare che la quantità da rendere sia uguale o minore da quella consegnata con la bolla specificata. La verifica viene effettuata mediante la store procedure *CheckQuantitàDanneggiata* che restituisce un valore booleano.

```

CREATE DEFINER='root'@'localhost' PROCEDURE `CheckQuantitàDanneggiata` ( in QuantitàDaRendere int, IDBolla bigint, NumeroLotto bigint, out result bool)
BEGIN
    DECLARE quantity int; -- variabile per il check --
    DECLARE AIC_EAN_ByLotto char(15);
    SET AIC_EAN_ByLotto = ( SELECT AIC_EAN
        FROM lotto
        WHERE numero_lotto = NumeroLotto);
    SET quantity = (SELECT quantità
        FROM voce_bolla
        WHERE AIC_EAN = AIC_EAN_ByLotto
        AND ID_bolla = IDBolla);
    IF QuantitàDaRendere > quantity THEN
        SET result = false;
    ELSE
        SET result = true;
    END IF;
END

```

Una volta effettuato il controllo, è possibile procedere con il reso con la store procedure *ResoNonVendibile*.


```

1 ● CREATE DEFINER='root'@'localhost' PROCEDURE `ResoNonVendibile`(in ID_reso int, NumeroLotto bigint,
2                                     ID_Bolla bigint, QuantitàDaRendere int )
3 BEGIN
4     DECLARE AIC_EAN_ByLotto char(15);
5     SET AIC_EAN_ByLotto = ( SELECT AIC_EAN FROM lotto WHERE numero_lotto = NumeroLotto);
6     -- inserisco il reso del prodotto --
7     INSERT INTO reso( motivazione, ID_reso, ID_dipendente, nome_fornitore, numero_ordine)
8     VALUES('Danneggiato', -- Motivazione --
9             ID_reso, -- ID_reso --
10             (CurrentOperator()), -- ID _ dipendente --
11             (SELECT nome_fornitore -- Nome_Fornitore --
12             FROM ((ordine O JOIN bolla B ON (O.numero_ordine = B.numero_ordine) ) JOIN voce_bolla VB
13                  ON (B.ID_bolla = VB.ID_bolla) JOIN lotto LO ON(VB.AIC_EAN = LO.AIC_EAN) )
14             WHERE numero_lotto = NumeroLotto),
15             (SELECT O.numero_ordine -- Numero_ordine --
16             FROM ((ordine O JOIN bolla B ON (O.numero_ordine = B.numero_ordine) ) JOIN voce_bolla VB
17                  ON (B.ID_bolla = VB.ID_bolla) JOIN lotto LO ON(VB.AIC_EAN = LO.AIC_EAN) )
18             WHERE numero_lotto = NumeroLotto));
19
20     INSERT INTO riferimento_reso(ID, quantità_resa, AIC_EAN)
21     VALUES (ID_reso,QuantitàDaRendere,AIC_EAN_ByLotto);
22
23     UPDATE prodotto
24     SET giacenza_totale= giacenza_totale - QuantitàDaRendere
25     WHERE AIC_EAN = AIC_EAN_ByLotto;
26 END

```

OP 4 – Effettuazione di un reso per smaltimento

Il reso per smaltimento viene effettuato nel momento in cui sono presenti all'interno della farmacia prodotti scaduti. Prima di effettuare il reso occorre verificare che la quantità da rendere sia uguale o minore a quella che risulta scaduta. La verifica viene effettuata mediante la store procedure CheckQuantitàScaduta che restituisce un valore booleano.

```

1 ● CREATE DEFINER='root'@'localhost' PROCEDURE `CheckQuantitàScaduta`( in QuantitàDaRendere int, NumeroLotto bigint,
2                                     IDReso int, out result boolean)
3 BEGIN
4     DECLARE quantity int; -- variabile per il check --
5     DECLARE AIC_EAN_ByLotto char(15);
6     SET AIC_EAN_ByLotto = ( SELECT AIC_EAN
7                             FROM lotto
8                             WHERE numero_lotto = NumeroLotto);
9     SET quantity = (SELECT quantità_in_scadenza
10                    FROM scadenza
11                    WHERE AIC_EAN = AIC_EAN_ByLotto
12                      AND lotto = NumeroLotto);
13     IF QuantitàDaRendere > quantity THEN
14         SET result = false;
15     ELSE
16         SET result = true;
17     END IF;
18 END

```

Una volta effettuato il controllo, è possibile procedere con il reso mediante la store procedure *ResoSmaltimento*.

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `ResoSmaltimento`( in IDReso int, LottoScaduto bigint)
2 BEGIN
3
4 DECLARE QuantitàDaRendereScaduta bigint;
5 DECLARE AIC_EAN_ByScadenza char(15);
6 SET QuantitàDaRendereScaduta = (SELECT S.quantità_in_scadenza -- Prendo la quantità in scadenza da rendere --
7 FROM scadenza as S, lotto as L
8 WHERE current_date() > S.data_scadenza
9 AND L.numero_lotto = LottoScaduto
10 AND L.AIC_EAN = S.AIC_EAN);
11 SET AIC_EAN_ByScadenza = (SELECT S.AIC_EAN
12 FROM scadenza as S
13 WHERE S.lotto = LottoScaduto);
14 -- inserisco il 'reso' del prodotto --
15 INSERT INTO reso( motivazione, ID_reso, ID_dipendente, nome_fornitore, numero_ordine)
16 -- Scadenza --
17 VALUES ('smaltimento', -- Motivazione --
18 IDReso, -- Id--
19 CurrentOperator(), -- Id_Dipendente --
20 (SELECT P.prodotto -- Nome_Fornitore --
21 FROM (prodotto P JOIN scadenza S ON (P.AIC_EAN = S.AIC_EAN))
22 WHERE current_date() > S.data_scadenza
23 AND S.lotto = LottoScaduto),
24 null); -- Numero Ordine a null, non serve per le scadenze --
25
26 -- Inserimento di una riga nella tabella 'riferimento_reso' --
27 INSERT INTO riferimento_reso (ID, quantità_resa, AIC_EAN)
28 VALUES (IDReso, -- Id_Reso --
29 QuantitàDaRendereScaduta, -- Quantità da Rendere perchè Scaduto--
30 (SELECT S.AIC_EAN
31 FROM scadenza as S, lotto as L
32 WHERE L.numero_lotto = LottoScaduto
33 AND L.AIC_EAN = S.AIC_EAN));
34
35 -- Metto il 'Lotto' a non vendibile (è scaduto)
36 UPDATE lotto SET vendibile = 0
37 WHERE numero_lotto = LottoScaduto
38 AND AIC_EAN = AIC_EAN_ByScadenza;
39
40 -- Cambio la Giacenza in 'Prodotto' per la Scadenza--
41 UPDATE prodotto SET giacenza_totale= giacenza_totale - QuantitàDaRendereScaduta
42 WHERE AIC_EAN = AIC_EAN_ByScadenza;
43 END
```

OP 5 – Ricerca dettagliata di un prodotto

La ricerca dettagliata di un prodotto prevede l'inserimento del nome all'interno del database e produce la lista dei farmaci aventi quel nome affiancati dalle informazioni relative ad essi.

Presenta un primo controllo effettuato mediante la store procedure *CheckProdottoPresente* che restituisce un valore booleano pari a 1 se il nome inserito corrisponde ad un prodotto presente nel database, o altrimenti.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CheckProdottoPresente`( in NomeProdotto char(100) , out result boolean )
2 BEGIN
3     DECLARE presenza int;
4     SET presenza = (SELECT COUNT(*) FROM prodotto WHERE nome = NomeProdotto);
5     -- se il count restituisce un valore > 0, il prodotto è presente nella lista --
6     IF presenza = 0 THEN
7         SET RESULT = FALSE; -- Non è presente nella Lista 0 --
8     ELSE
9         SET RESULT = TRUE; -- E' presente nella lista 1 --
10    END IF;
11    END
```

Se il controllo va a buon fine, allora viene utilizzata la store procedure *RicercaDettagliataProdotto* per restituire le informazioni del prodotto selezionato.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `RicercaDettagliataProdotto`( in NomeProdotto char(100) )
2 BEGIN
3     SELECT *
4     FROM prodotto
5     WHERE nome = NomeProdotto;
6    END
```

OP 6 – Visualizzazione delle scadenze imminenti

La visualizzazione delle scadenze imminenti prevede l'inserimento di una valore che indica i giorni mancanti alla scadenza e seleziona tutti i prodotti presenti in magazzino la cui scadenza avviene prima dei giorni inseriti.

Utilizza una store procedure di tipo *CheckScadenzeImminenti* che verifica che i giorni mancanti alla scadenza siano maggiori di 0.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CheckScadenzeImminenti`( in GiorniMancantiScadenza int,
2                                     out result boolean)
3 BEGIN
4     -- verificare che i giorni mancanti alla scadenza siano maggiori di 0 --
5     IF GiorniMancantiScadenza < 0 THEN
6         SET result= FALSE;
7     ELSE
8         SET result = TRUE;
9     END IF;
10    END
```

Successivamente utilizza una store procedure *ScadenzeImminenti* che stampa tutti i prodotti con scadenza minore rispetto a quella inserita, inclusi i prodotti già scaduti.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `ScadenzeImminenti`( in GiorniMancantiScadenza int)
2 BEGIN
3     SELECT *
4     FROM scadenza as S
5     WHERE datediff(data_scadenza, current_date())<= GiorniMancantiScadenza
6     OR data_scadenza <= current_date();
7 END

```

OP 7 – Visualizzazione statistiche di vendita giornalieri

La store procedure *CheckDataInserita* serve a verificare che la data inserita sia una data passata e restituisce un valore booleano.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CheckDataInserita`( in Data_AAAA_MM_GG date, out result boolean )
2 BEGIN
3     -- il giorno inserito deve essere un giorno passato --
4     IF Data_AAAA_MM_GG > current_date() THEN
5         SET result = FALSE;
6     ELSE
7         SET result = TRUE;
8     END IF;
9 END

```

La store procedure *StatisticheGiornaliere* dato un giorno, un mese e un anno, restituisce il guadagno di quel giorno e il numero di scontrini battuti.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `StatisticheGiornaliere`( in Data_AAAA_MM_GG date)
2 BEGIN
3     DECLARE Guadagno decimal(65,2);
4     DECLARE CountScontrini int;
5     DECLARE NumProgIniziale int;
6     DECLARE NumScontrini int;
7     SET Guadagno = 0;
8     SET CountScontrini = (SELECT COUNT(*)
9         FROM scontrino
10        WHERE data_ora = Data_AAAA_MM_GG);
11     SET NumScontrini = CountScontrini;
12     SET NumProgIniziale = (SELECT numero_progressivo
13        FROM scontrino
14        WHERE data_ora = Data_AAAA_MM_GG
15        LIMIT 1);
16     -- assumiamo che gli scontrini di un giorno abbiano numero progressivo successivo --
17     WHILE CountScontrini>0 DO
18         SET Guadagno = Guadagno +
19         (SELECT totale_complessivo
20        FROM scontrino
21        WHERE data_ora = Data_AAAA_MM_GG
22        AND numero_progressivo = NumProgIniziale);
23         SET CountScontrini = CountScontrini -1;
24         SET NumProgIniziale = NumProgIniziale+1;
25     END WHILE;
26     SELECT Guadagno, NumScontrini;
27 END

```

OP 8 – Aggiunta di un cliente

Utilizza una store procedure *CheckAggiuntaCliente* per verificare che il cliente non sia già presente nel database e per verificare che la data di nascita sia minore di quella corrente.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CheckAggiuntaCliente` ( in CodiceFiscale char(16),
2                                     DataDiNascita date, out result boolean)
3 BEGIN
4     -- il cliente non deve essere già presente nel db--
5     -- la data di nascita deve essere minore della data corrente --
6     DECLARE presente int;
7     SET presente = (SELECT COUNT(*) FROM cliente WHERE CF = CodiceFiscale);
8
9     IF (presente > 0) OR (current_date() < DataDiNascita) THEN
10         SET result = FALSE; -- cliente già presente --
11     ELSE
12         SET result = TRUE; -- cliente non presente --
13     END IF;
14
15 END
```

Utilizza poi una store procedure *AggiuntaCliente* che prende in input le informazioni del cliente e lo registra all'interno del database.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `AggiuntaCliente` ( in CodiceFiscale char(16),
2                                     NomeCliente char(100), CognomeCliente char(100), DataDiNascita date )
3 BEGIN
4     INSERT INTO cliente (CF, nome, cognome, data_nascita)
5     VALUES (CodiceFiscale, NomeCliente, CognomeCliente, DataDiNascita);
6 END
```

OP 9 – Inserimento di una bolla

L'inserimento di una bolla avviene mediante la store procedure *InserimentoBolla* che prende in input il numero dell'ordine di cui si sta inserendo la bolla e il numero di voci bolla che devono essere inseriti.

Contiene al suo interno un check: l'ordine relativo alla bolla da inserire deve essere confermato per poter essere consegnato dal fornitore.

Nel momento in cui viene inserita una bolla, il prodotto viene consegnato alla farmacia per cui è necessario inserire i nuovi lotti e le relative scadenze e modificare la giacenza in magazzino dei prodotti.

```

1 ● CREATE DEFINER='root'@'localhost' PROCEDURE `InserimentoBolla`( in NumeroOrdine int, NumVociBollaToAdd int)
2 BEGIN
3     DECLARE CountNumLotti int;
4     DECLARE CountVociOrderByOrder int; -- il numero di voci bolle non può essere maggiore delle voci ordini --
5     DECLARE posizione int;
6     DECLARE CountNumBolle int;
7     DECLARE CheckConfermaByOrder int;
8     DECLARE CountVociBollaByOrder int;
9     -- Conto il Numero di Lotti che ci sono nella Tabella lotto --
10    SET CountNumLotti = (SELECT COUNT(*)
11                          FROM lotto);
12    -- Conto il Numero di VociBolla su quel determinato Ordine, inizialmente a 0 ancora voci bolla--
13    -- non creati, in questo caso --
14    -- CountVociBollaByOrder serve per inserire il numero di bolla in modo progressivo --
15    SET CountVociBollaByOrder = ( SELECT COUNT(*)
16                                  FROM ordine O, bolla BO, voce_bolla VB
17                                  WHERE O.numero_ordine = NumeroOrdine
18                                  AND BO.numero_ordine = NumeroOrdine
19                                  AND BO.ID_bolla = VB.ID_bolla );
20    -- Conto lo voci ordine da un determinato ordine --
21    SET CountVociOrderByOrder = ( SELECT COUNT(*)
22                                  FROM voce_ordine
23                                  WHERE numero_ordine = NumeroOrdine);
24    SET posizione = 1;
25    -- Conto il numero di bolle dentro la tabella Bolla --
26    SET CountNumBolle = ( SELECT COUNT(*)
27                          FROM bolla as BO );
28    -- Controllo se l'ordine nel quale voglio fare la bolla sia Confermato (1) oppure NO (0) --
29    -- CheckConfermaByOrder serve per verificare che l'ordine sia stato Confermato --
30    SET CheckConfermaByOrder = (SELECT ordine_confermato
31                                FROM ordine
32                                WHERE numero_ordine = NumeroOrdine);
33
34    -- Solo se l'ordine è stato confermato posso creare la bolla su quel ordine --
35    IF CheckConfermaByOrder = 1 THEN
36        -- Inserisco la Nuova Bolla --
37        -- L'ID_bolla va in modo progressivo-> 1,2,3,4...,Data odierna, NumOrdineInserito --
38        INSERT INTO bolla (ID_bolla, data_ora_consegna, numero_ordine)
39        VALUES( CountNumBolle+1, current_date(), NumeroOrdine);
40    IF (CountVociOrderByOrder - CountVociBollaByOrder) >= NumVociBollaToAdd THEN
41        WHILE NumVociBollaToAdd > 0 DO -- Inserisco ogni voce_Bolla --
42            INSERT INTO voce_bolla (AIC_EAN, ID_bolla, posizione_in_elenco, quantità)
43            VALUES(
44                (SELECT AIC_EAN
45                 FROM voce_ordine
46                 WHERE numero_ordine = NumeroOrdine
47                 AND posizione_in_elenco = (CountVociBollaByOrder+posizione))
48                ,CountNumBolle+1, posizione,
49                (SELECT quantità
45                 FROM voce_ordine
46                 WHERE numero_ordine = NumeroOrdine
47                 AND posizione_in_elenco = (CountVociBollaByOrder+posizione))
48            );

```



```

54
55      -- Diminuisco il numero di voci da inserire
56      SET NumVociBollaToAdd = NumVociBollaToAdd-1;
57
58      -- Inserimento lotto --
59      SET CountNumLotti = CountNumLotti + 1;
60      INSERT INTO lotto (AIC_EAN, numero_lotto, vendibile)
61      VALUES ((SELECT AIC_EAN
62                FROM voce_ordine
63                WHERE numero_ordine = NumeroOrdine
64                AND posizione_in_elenco = (CountVociBollaByOrder+posizione) ), CountNumLotti, 1 );
65
66      -- inserimento scadenza --
67      INSERT INTO scadenza (AIC_EAN, lotto, data_scadenza, quantità_in_scadenza)
68      VALUES ((SELECT AIC_EAN
69                FROM voce_ordine
70                WHERE numero_ordine = NumeroOrdine
71                AND posizione_in_elenco = (CountVociBollaByOrder+posizione) ), CountNumLotti,
72            (date_add( current_date(), interval 1 year)),
73            (SELECT quantità
74              FROM voce_ordine
75              WHERE numero_ordine = NumeroOrdine
76              AND posizione_in_elenco = (CountVociBollaByOrder+posizione)));
77
78      -- modifica giacenza --
79      UPDATE prodotto SET giacenza_totale= giacenza_totale + (SELECT quantità
80                FROM voce_ordine
81                WHERE numero_ordine = NumeroOrdine
82                AND posizione_in_elenco = (CountVociBollaByOrder+posizione))
83      WHERE AIC_EAN = (SELECT AIC_EAN
84                FROM voce_ordine
85                WHERE numero_ordine = NumeroOrdine
86                AND posizione_in_elenco = (CountVociBollaByOrder+posizione));
87
88      SET posizione = posizione + 1;
89  END WHILE;
90 END IF;
91 END IF;
92 END

```

OP 10 – Visualizzazione di ordini, bolle e fatture di un giorno

La visualizzazione degli ordini, delle bolle e delle fatture relative ad un giorno prevede l'utilizzo di una store procedure che controlla la data inserita in input. Si tratta del *CheckDataInserita* utilizzata anche in altre operazioni.

Successivamente viene chiamata la store procedure *VisualizzazioneOrdiniBolleFatture* che unisce le varie tabelle e stampa le informazioni relative alla data inserita.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `VisualizzazioneOrdiniBolleFatture`( in DataVisualizzazione char(100))
2 BEGIN
3     -- tabelle unite --
4     SELECT O.numero_ordine, O.data_ora, O.ordine_confermato, O.ID_dipendente, O.nome_fornitore,
5         F.ID_fattura, F.totale, B.ID_bolla, B.data_ora_consegna
6     FROM fattura as F, ordine as O, bolla as B
7     WHERE O.data_ora = convert(DataVisualizzazione, date)
8     AND F.numero_ordine = O.numero_ordine
9     AND B.numero_ordine = O.numero_ordine;
10
11 END

```

OP 11 – Visualizzazione dipendente con entrate maggiori

Prevede una procedura per il controllo dei dati in input. La procedura *CheckEntrateMaggiori* verifica che l'anno inserito sia minore o uguale a quello corrente e che il valore del mese sia compreso tra 1 e 12.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CheckEntrateMaggiori`( in mese_MM int, anno_AAAA int, out result boolean)
2 BEGIN
3     -- il mese inserito deve essere un valore compreso tra 1 e 12 --
4     -- l'anno inserito deve essere minore o uguale a quello corrente perchè si tratta di uno storico --
5     IF (anno_AAAA > (YEAR(Current_date()))) OR (mese_MM <= 0) OR (mese_MM >12 )) THEN
6         SET result =FALSE;
7     ELSE
8         SET result =TRUE;
9     END IF;
10 END

```

Utilizza successivamente la procedura *DipendenteEntrateMaggiori* che determina il dipendente che ha venduto di più nel mese indicato restituendo l'ID del dipendente.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `DipendenteEntrateMaggiori`( in mese_MM int, in anno_AAAA int)
2 BEGIN
3     -- dipendente con entrata maggiore --
4     DECLARE Dipendente int;
5     DECLARE Guadagnoprec decimal(65,2);
6     DECLARE Guadagno decimal (65,2);
7     DECLARE IdDipendente int;
8     SET IdDipendente = (SELECT COUNT(*)
9         FROM dipendente_operatore);
10    SET guadagno = 0;
11    SET Guadagnoprec = 0;
12    SET Dipendente = 0;
13    WHILE IdDipendente > 0 DO
14        SET Guadagnoprec =( SELECT SUM(totale_complessivo) FROM scontrino
15            WHERE ID_dipendente = IdDipendente
16            AND MONTH(data_ora) = mese_MM
17            AND YEAR(data_ora) = anno_AAAA);
18        IF Guadagnoprec > guadagno THEN
19            SET Guadagno = Guadagnoprec;
20            SET Dipendente = IdDipendente;
21        END IF;
22        SET IdDipendente = IdDipendente -1;
23    END WHILE;
24    SELECT Dipendente, Guadagno;
25 END

```

OP 12 – Aggiunta di un oggetto di vendita

Si distinguono due casi:

- Aggiunta servizio mediante la procedura *AggiuntaServizio*

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `AggiuntaServizio`( in Descrizione char(200), Prezzo decimal(65,2) )
2 BEGIN
3     DECLARE IDOgg int;
4     SET IDOgg = ( SELECT COUNT(*) FROM oggetto_di_vendita as O );
5
6     INSERT INTO oggetto_di_vendita (servizio, ID_oggetto_di_vendita, AIC_EAN, descrizione, prezzo)
7     VALUES (1, IDOgg+1, null, Descrizione, Prezzo);
8
9 END
```

- Aggiunta prodotto mediante la procedura *AggiuntaProdotto*

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `AggiuntaProdotto`( in Descrizione char(200), Prezzo decimal(65,2),
2     aic_ean char(15), ClasseParaFarmaco char(200), CodiceEquivalenza char(100), TipoRicetta char(100),
3     TipoProdotto char(100), Nome char(100), Produttore char(100), MetodoSomministrazione char(100),
4     Quantità char(100))
5 BEGIN
6     DECLARE IDOgg int;
7     SET IDOgg = ( SELECT COUNT(*) FROM oggetto_di_vendita as O );
8
9     INSERT INTO oggetto_di_vendita (servizio, ID_oggetto_di_vendita, AIC_EAN, descrizione, prezzo)
10    VALUES (0, IDOgg+1, aic_ean, Descrizione, Prezzo);
11
12    INSERT INTO prodotto(classe_parafarmaco, codice_di_equivalenza, tipo_di_ricetta, AIC_EAN, tipo_prodotto,
13        nome, produttore, quantità, metodo_di_somministrazione, giacenza_totale)
14    VALUES (ClasseParaFarmaco, CodiceEquivalenza, TipoRicetta, aic_ean, TipoProdotto, Nome, Produttore,
15        Quantità, MetodoSomministrazione, 0);
16 END
```

OP 13 – Visualizzazione dei fornitori che offrono un prodotto

La store procedure *VisualizzazioneFornitori* prende in input il nome di un prodotto e restituisce la lista dei fornitori che offrono quel prodotto.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `VisualizzazioneFornitori`( in NomeProdotto char(100))
2 BEGIN
3     SELECT F.*
4     FROM prodotto P, fornitore F
5     WHERE P.nome = NomeProdotto
6     AND P.produttore = F.nome;
7 END
```

OP 14 – Notificare l'imminente esaurimento di un prodotto

La procedura *EsaurimentoImminente* serve a notificare quando la giacenza di un determinato prodotto è minore o uguale a 2 unità.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `EsaurimentoImminente`()
2 BEGIN
3     SELECT *
4     FROM prodotto
5     WHERE giacenza_totale <= 2;
6 END

```

OP 15 – Visualizzare mese con massimo guadagno

La store procedure *MeseMassimoGuadagno* prende in input un anno e confronta i guadagni di ogni mese determinando il mese con incasso maggiore.

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `MeseMassimoGuadagno`( in anno int)
2 BEGIN
3     DECLARE mese int;
4     DECLARE Count int;
5     DECLARE guadagno decimal(65,2);
6     DECLARE guadagno_mensile decimal(65,2);
7     SET Count = 12;
8     SET mese = 0;
9     SET guadagno = 0.0; -- variabile utile al confronto --
10    WHILE Count > 0 DO
11        SET guadagno_mensile = 0.0;
12        SET guadagno_mensile = guadagno_mensile + (SELECT SUM(totale_complessivo) FROM scontrino
13        WHERE MONTH(data_ora) = Count
14        AND YEAR(data_ora) = anno);
15        IF guadagno_mensile > guadagno THEN
16            SET guadagno = guadagno_mensile;
17            SET mese = Count;
18        END IF;
19        SET Count = Count -1;
20    END WHILE;
21    SELECT
22    CASE
23        WHEN mese=1 THEN 'gennaio'
24        WHEN mese=2 THEN 'febbraio'
25        WHEN mese=3 THEN 'marzo'
26        WHEN mese=4 THEN 'aprile'
27        WHEN mese=5 THEN 'maggio'
28        WHEN mese=6 THEN 'giugno'
29        WHEN mese=7 THEN 'luglio'
30        WHEN mese=8 THEN 'agosto'
31        WHEN mese=9 THEN 'settembre'
32        WHEN mese=10 THEN 'ottobre'
33        WHEN mese=11 THEN 'novembre'
34        WHEN mese=12 THEN 'dicembre'
35    END AS mese;
36 END

```


OP 16 – Visualizzazione dei farmaci equivalenti

La procedura *FarmaciEquivalenti* prende in input un codice di equivalenza e restituisce tutti i farmaci che presentano lo stesso codice di equivalenza.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `FarmaciEquivalenti`(in CodiceEquivalenza char(100))
2 BEGIN
3     SELECT *
4     FROM prodotto
5     WHERE codice_di_equivalenza = CodiceEquivalenza;
6 END
```

OP 17 – Inserimento di un fattura

Presenta un *CheckFattura* che controlla che il numero di ordine inserito sia esistente.

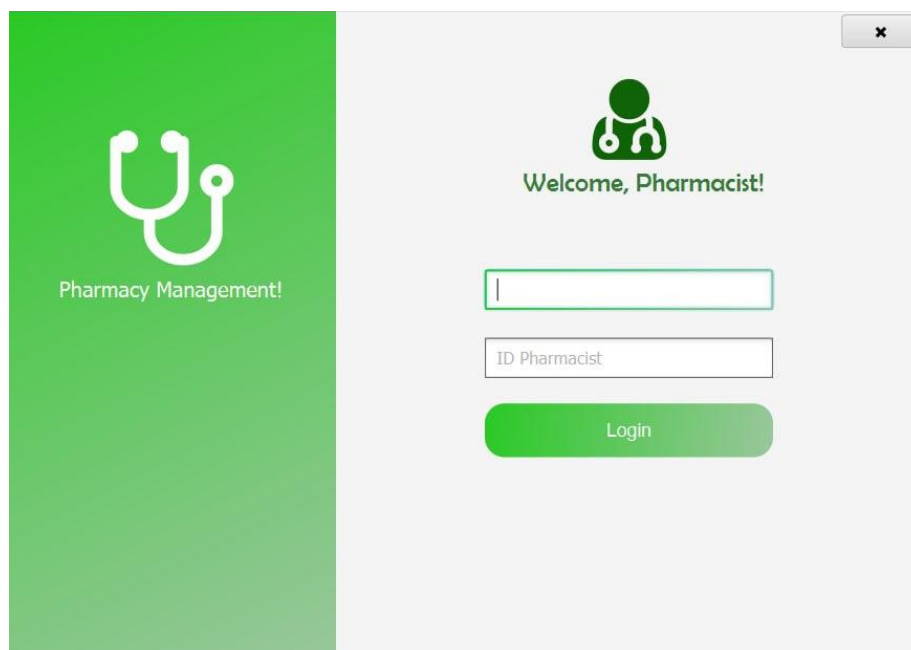
```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `CheckFattura`( in NumeroOrdine int, out result boolean)
2 BEGIN
3     -- devo verificare che sia stato inserito un numero di ordine esistente --
4     DECLARE VarCheck int;
5     SET VarCheck = (SELECT COUNT(*) FROM ordine WHERE numero_ordine = NumeroOrdine);
6     IF VarCheck <= 0 THEN
7         SET result = FALSE;
8     ELSE
9         SET result = TRUE;
10    END IF;
11 END
```

Presenta poi una procedura *InserimentoFattura* che inserisce la fattura relativa all'ordine calcolandone il totale e modifica lo stato dell'ordine in confermato.

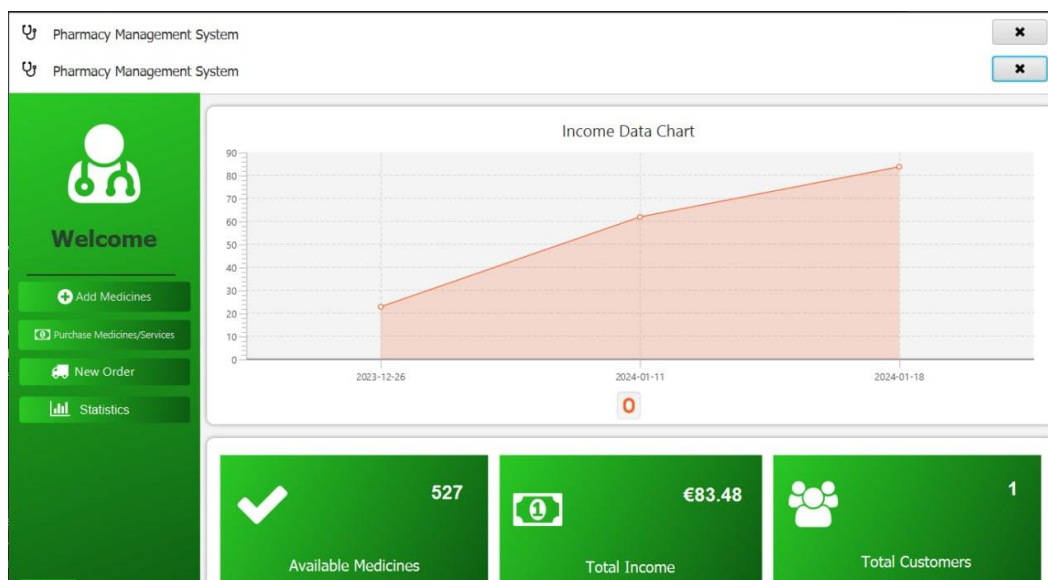
```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `InserimentoFattura`( in NumeroOrdine bigint)
2 BEGIN
3     DECLARE NumFa int;
4     DECLARE NumVO int;
5     SET NumFa = (SELECT COUNT(*) FROM fattura as F);
6     SET NumVO = ( SELECT COUNT(*) FROM voce_ordine as VO WHERE VO.numero_ordine= NumeroOrdine);
7     -- Inserire la fattura e calcolare il totale --
8     INSERT INTO fattura (ID_fattura, numero_ordine, totale)
9     VALUES ( NumFa+1, NumeroOrdine, 0.00); -- Primo inserito --
10
11     -- 2 --
12     WHILE NumVO > 0 DO
13
14         UPDATE fattura SET totale = totale +
15         (SELECT VO.costo_unitario*VO.quantità
16         FROM voce_ordine as VO
17         WHERE VO.numero_ordine = NumeroOrdine
18         AND NumVO = VO.posizione_in_elenco) WHERE ID_fattura = NumFa+1;
19
20         SET NumVO = NumVO -1;
21     END WHILE;
22     -- Ordine Confermato --
23     UPDATE ordine SET ordine_confermato = 1
24     WHERE numero_ordine = NumeroOrdine;
25
26 END
```

6. PROGETTAZIONE DELL'APPLICAZIONE

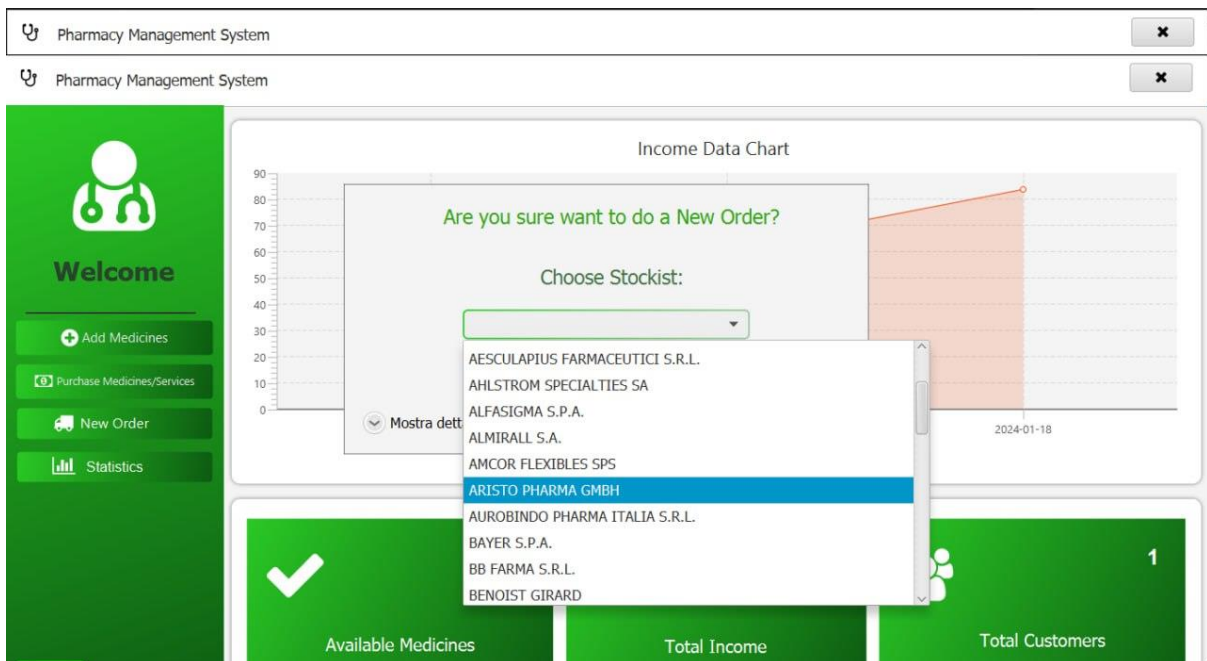
6.1. DESCRIZIONE DELL'INTERFACCIA UTENTE



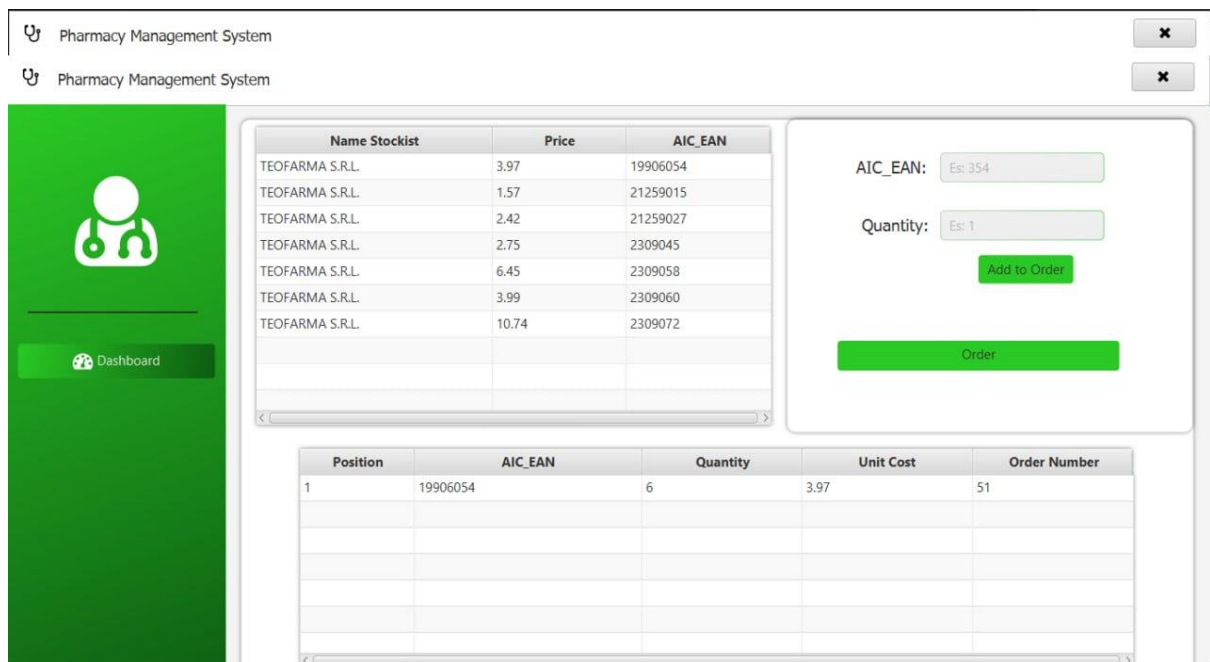
All'apertura del software, viene visualizzata l'interfaccia di accesso che consente di accedere all'area personale di uno specifico operatore



Effettuato l'accesso, viene mostrata la dashboard relativa all'operatore, contenente le sue statistiche di vendita giornaliere



La voce *New Order* consente di selezionare un fornitore da cui effettuare il nuovo ordine



Una volta selezionato il fornitore, verrà mostrata l'offerta di farmaci da lui messa a disposizione. Da qui è quindi possibile selezionare i farmaci da ordinare

 Pharmacist Statistics!

Dipendente con Entrate Maggiori per Mese di quest'anno

Mese:

Mese con Massimo Guadagno per Anno:

Anno:

Statistiche Giornaliere:

Giorno:

E' possibile visualizzare statistiche più approfondite selezionando la voce *Statistics* della dashboard.