



UNIVERSITÀ DEGLI STUDI DI PERUGIA

Documento di progetto di
Basi Di Dati

Corso di Laurea in Ingegneria Informatica ed Elettronica – A.A. 2020-2021
DIPARTIMENTO DI INGEGNERIA

docente
Prof. Giuseppe Liotta

Azienda di Tabacco

Progettazione di una base di dati per la gestione logistica di ricezione, stoccaggio e spedizione per un'azienda che tratta tabacco grezzo.

Studenti

311020	Adriana Vella	adriana.vella@studenti.unipg.it
312879	Edoardo Marchetti	edoardo.marchetti1@studenti.unipg.it
311311	Mark Sorin Mardare	marksorin.mardare@studenti.unipg.it

Sommario

PROGETTAZIONE CONCETTUALE	2
Raccolta dei requisiti	2
Analisi delle specifiche	3
Glossario dei termini principali	7
Elenco delle operazioni	8
Schema Entità-Relationship	9
Vincoli non esprimibili	15
Dizionario dei dati (Entità)	16
Dizionario dei dati (Relationship)	18
PROGETTAZIONE LOGICA	20
Tavola dei volumi	20
Tavola delle operazioni	22
Analisi delle ridondanze	24
Eliminazione delle generalizzazioni	33
Accorpamento/partizionamento di relationship	35
Scelta degli identificatori primari	35
Ristrutturazione dello schema E-R	35
Modello Relazionale	37
Traduzione di Entità	37
Traduzione di Relationship	38
Normalizzazione	39
IMPLEMENTAZIONE DELLE OPERAZIONI	39
Creazione delle tabelle	39
Creazione delle query	47
ESECUZIONE DELLE OPERAZIONI PRINCIPALI	74

PROGETTAZIONE CONCETTUALE

Raccolta dei requisiti

Per raccolta dei requisiti si intende la completa individuazione dei problemi che la base di dati da realizzare deve risolvere e le caratteristiche che tale base di dati dovrà avere a seguito dell'operazione di interrogazione del committente del database.

Si vuole realizzare una base di dati per la gestione logistica di ricezione, stoccaggio e spedizione per un'azienda che tratta tabacco grezzo.

L'impresa ha tre **strutture** in cui il tabacco può essere conservato. Si vuole pertanto registrare la *località* in cui è situato il magazzino indicando la città e l'indirizzo, la *capacità massima* di casse che ciascun magazzino può ospitare e il *numero di colli* attualmente presenti.

Il tabacco giunge nei magazzini in **carichi** che possono appartenere o meno ad uno stesso lotto. Ogni carico viene identificato con un *valore numerico* e l'*id del lotto* a cui appartiene (anch'esso numerico). Inoltre al momento della ricezione vengono salvati anche il *mezzo* con cui il carico è stato portato e la *data*. Nel caso in cui il carico sia arrivato via terra sarà necessario memorizzare il **camion**, identificato dalla targa, con cui è stato effettuato il trasporto; mentre se il trasferimento è avvenuto via mare si vuole conoscere la **nave**, identificata da una targa navale, con cui è arrivato ciascun **container** in cui viene stipato il carico, e la città di arrivo. Per entrambe le modalità si registra anche la data di partenza.

Per ciascun **lotto** è richiesto conoscere la *varietà*, la *forma*, l'*origine* e la *provenienza* del tabacco. Inoltre si vuole sapere lo *status doganale* del lotto, come i carichi sono stati trasportati e in quale *deposito* è conservato.

Il tabacco può essere di tre varietà: flue-cured-virginia (FCV), light-air-cured (LAC) o dark-air-cured (DAC). Le forme che il prodotto può assumere sono strips, loose leaves (TTL) e by product (BP). La forma BP si suddivide ulteriormente in stem, stem2, scraps1, scraps2 e fines. Per origine del lotto si intende la nazionalità del coltivatore che ha coltivato il tabacco con cui è stato composto, mentre per provenienza si indica la nazionalità del venditore del lotto. E' possibile che venditore e coltivatore corrispondano. Lo status del lotto indica la posizione doganale e può assumere quattro valori diversi: FC, libera circolazione in Europa; EXEU, sigla che indica l'origine extra europea del tabacco; FP, libera pratica, che indica origine da stati extra UE i quali hanno accordi commerciali con l'Europa in funzione dei quali possono beneficiare dell'azzeramento di dazi; NAZ, indica un prodotto nazionale. Il trasporto del lotto può invece avvenire via mare o via terra. I carichi di un lotto sono trasportati tutti nello stesso modo.

Ogni lotto è costituito da **casse** omogenee, ovvero casse contenenti tabacco di stessa varietà, forma, origine, provenienza, tipo di imballaggio e peso. Le casse (o colli) sono contraddistinte da un codice barcode univoco. Nella base di dati si dovrà tener traccia di tutti e tre i fattori: *barcode*, *peso* e *tipo di imballaggio*. Inoltre si registra anche il numero del lotto a cui appartiene e il carico con cui è arrivata.

Quando un carico giunge in magazzino inizia un iter per verificare la qualità del prodotto. E' dunque necessario tener traccia del processo di esaminazione.

Nello stesso giorno in cui arriva un carico viene effettuato un controllo ingresso nel quale si osserva se è presente un'infestazione. Nel caso in cui ci siano delle irregolarità il carico viene immediatamente isolato per circa 48/72 ore e in base alla gravità dell'infestazione trovata, terminata la quarantena, viene trattato o meno prima di passare all'ispezione qualità. Il carico che non presenta nessuna

infestazione viene direttamente passato all'ispezione qualità la quale consiste in un controllo visivo su alcune casse del carico prese a campione. Successivamente 20 casse per lotto vengono portate in **laboratorio** per delle analisi dalle quali si redige un referto. Il **referto** indica i valori di NIC, RS, NO3 e NH3 riscontrati nella cassa in valore percentuale. Per ciascun carico si vogliono conoscere le *date* in cui ogni step di verifica viene effettuato, per la quarantena si vuole conoscere anche la data di fine isolamento in quanto il trattamento può iniziare in un altro giorno. Per le casse del carico invece si vuole sapere se sono state selezionate per l'ispezione, se sono state analizzate e, eventualmente, il referto associato con i relativi valori.

Lo stoccaggio dei lotti inizia quando giunge in magazzino un ordine da un cliente. Della **società richiedente** si vuole conoscere il *nome*, la *partita IVA* e la *città* della sede. Per l'**ordine** invece registriamo un *id*, il *numero di colli* totali e il *numero di batch* con cui deve essere composto. Nel caso in cui i lotti da cui si prendono le casse siano di status FP o EXEU devono essere sdoganati e all'ordine si allega una sezione **custom clearance** nella quale sono indicati i lotti che devono cambiare status. Dopo la registrazione si passa al pre-kitting durante il quale l'ordine viene composto in **batch** seguendo una **ricetta**. Ogni batch è un gruppo di casse che vengono caricate sul camion insieme, mentre la ricetta indica da quali lotti devono essere prese le casse. Di ogni batch si deve conoscere *quali casse* lo compongono, a *quale ordine* appartiene, il *numero progressivo nell'ordine* e secondo *quale ricetta* è stato assemblato. Per una ricetta invece si registra un *valore identificativo*, il *lotto* da cui si prendono le casse e quanti colli servono per un batch di un determinato lotto. Le due "metà" saranno indicate nella base di dati con lo stesso barcode della cassa originale e individuate come *partizione A* e *B*.

Al termine dell'iter appena descritto l'ordine viene spedito. Della spedizione registriamo la *data*, l'*ordine* inviato e il *mezzo* usato per spedire l'ordine.

Analisi delle specifiche

L'analisi delle specifiche consiste nel chiarimento e nell'organizzazione delle informazioni raccolte durante l'intervista. Si procede a decomporre il documento in gruppi di frasi omogenee, relative agli stessi concetti.

Frase di carattere generale:

Si vuole realizzare una base di dati per la gestione logistica di ricezione, stoccaggio e spedizione per un'azienda che tratta tabacco grezzo.

Frase relative ai depositi:

L'impresa ha tre strutture in cui il tabacco può essere conservato. Ognuna di esse viene identificata da un codice identificativo univoco. Si vuole pertanto registrare la località in cui è situato il magazzino, riportando città e indirizzo, la capacità massima di casse che ciascun magazzino può ospitare e il numero di colli attualmente presenti.

Frase relative al lotto:

Ciascun lotto è dotato di un valore numerico identificativo univoco. Ogni lotto che arriva in magazzino è costituito da casse omogenee contenenti tabacco della stessa varietà, forma, origine e

provenienza. Ad ogni lotto viene attribuito uno status doganale che può assumere diversi valori in base al paese di origine e di provenienza, europeo o extraeuropeo. Poiché il trasferimento del lotto può avvenire via terra o via mare, si deve indicare la modalità di arrivo del carico. E' possibile che un lotto arrivi in deposito diviso in carichi diversi e in più giorni, pertanto bisogna indicare il numero di carichi per lotto da cui è composto e la quantità totale delle casse che comprende. Per origine si intende la nazionalità produttore che ha coltivato il prodotto, mentre il venditore è l'azienda da cui si è comprato il lotto e la sua nazionalità definisce la provenienza del lotto. A volte coltivatore e venditore possono coincidere.

Frase relative al carico:

Ogni carico viene identificato tramite un valore numerico e l'id del lotto di appartenenza. Si vuole registrare con quale mezzo arriva in magazzino: nel caso in cui sia arrivato via terra si dovrà tener traccia della targa del camion, mentre nel caso in cui il trasporto è avvenuto via mare si deve conoscere il container su cui è conservato e la nave. Per ciascun carico si vogliono conoscere le date in cui ogni step di verifica dell'iter di controllo qualità viene effettuato e nel caso di quarantena la data di fine isolamento. Se è stato necessario un ulteriore trattamento in seguito all'isolamento, deve essere indicata la data in cui è stato effettuato.

Frase relative alle casse:

Ogni cassa viene identificata tramite un barcode univoco e appartiene ad un solo lotto. Le caratteristiche dei colli sono: il peso, il quale viene espresso in kg e indica la quantità di tabacco contenuto, e il tipo di imballaggio, che assume valori C48 o C96. Quando una cassa è di tipo C96 significa che ha subito un'operazione di split nella fase di stivaggio. Si vuole registrare dunque barcode, lotto di appartenenza, carico con cui è arrivata, peso e tipo di imballaggio.

Inoltre si vuole conoscere se la cassa è stata selezionata per l'ispezione qualità (se sì il giorno in cui è stata controllata) e se è stata analizzata (e eventuale referto).

In fase di stoccaggio è necessario tener traccia del batch in cui la cassa viene inserita in modo tale da sapere con quale ordine è stata eventualmente spedita.

Frase relative allo split:

E' possibile che un batch richieda lo split di una cassa, ovvero il frazionamento del contenuto in due casse, ognuna delle quali avrà lo stesso barcode, ereditando gli attributi della cassa originaria, e in aggiunta un campo in cui si differenziano le due metà della cassa originaria.

Frase relative alle società terze:

Con società terze si indicano le aziende che comprendono venditori e coltivatori. Per ognuna di queste società si registra partita IVA, ragione sociale, nazionalità, città sede, indirizzo. Coltivatore e venditore possono anche corrispondere.

Frase relative al venditore:

Solitamente il tabacco viene acquistato da un'azienda che compra il tabacco da più coltivatori lo trasforma. Non è escluso però che questa fase di lavorazione venga effettuata dalla stessa

impresa coltivatrice del prodotto. La nazionalità è necessaria per determinare lo status del lotto.

Frase relative al coltivatore:

Il coltivatore può essere una singola azienda oppure un consorzio di aziende.

Frase relative al mezzo di trasporto:

Tutti i carichi di uno stesso lotto vengono trasportati in una delle due modalità: via mare o via terra. Stessa cosa accade in fase di spedizione, ovvero le casse che compongono un singolo ordine vengono inviate tramite camion o tramite nave.

Frase relative al camion:

Nel caso in cui il trasferimento sia avvenuto via terra si vuole conoscere la targa del camion, quando e da dove questo è partito.

Frase relative alla nave:

Se il trasporto viene effettuato mediante una nave, si vuole conoscere la targa navale, il porto di partenza, il porto di arrivo e la data della partenza.

Frase relative al container:

In un container si può conservare un solo carico. In caso di trasferimento marittimo è necessario conoscere il valore che identifica univocamente il container, con quale nave viene trasportato e il carico contenuto.

Frase relative al referto:

Il referto è un resoconto delle analisi effettuate su una cassa campione e indica, in percentuale, i valori delle molecole di NIC, RS, NO₃ e NH₃. Inoltre in ogni referto viene indicata anche la data di rilascio, su quale cassa è stata fatta l'analisi e il laboratorio che lo ha redatto.

Frase relative al laboratorio:

Le analisi vengono effettuate da laboratori terzi, pertanto si vuole conoscere di ogni laboratorio la città e l'indirizzo in cui è collocato, il nome del laboratorio e il numero telefonico.

Frase relative all'ordine di spedizione:

Al magazzino viene inviato un ordine di spedizione in cui viene indicato il numero di casse da spedire e in quanti batch devono essere raggruppate. Si vuole conoscere il codice identificativo dell'ordine, il cliente, il numero di colli totali necessari per comporlo e il numero di batch. Inoltre è possibile che venga allegato un custom clearance.

Fraasi relative al custom clearance:

All'interno del custom clearance vengono indicati quali lotti devono essere sdoganati, ovvero devono passare dallo status FP o EXEU a quello di libera circolazione, FC. Per ogni custom clearance si vuole indicare a quale ordine appartiene e i lotti da sdoganare. Infine è richiesto anche tener traccia dello status attuale.

Fraasi relative alla ricetta:

L'ordine viene composto nella fase di pre-kitting durante la quale viene definita una ricetta, ovvero una formula di come l'ordine deve essere suddiviso, a cui viene associato un indice identificativo univoco. Per ogni ricetta si registra un valore numero identificativo, i lotti da cui prendere i colli, il numero di casse per singolo lotto necessarie.

Fraasi relative al batch:

Un batch è un sottoinsieme di casse che compongono un ordine assemblato seguendo una determinata ricetta. Per ogni batch si registra il numero progressivo nell'ordine, l'ordine di riferimento e la ricetta con cui è assemblato.

Fraasi relative alla spedizione:

Si vuole conoscere la data di spedizione dell'ordine, di cui si riporta anche il codice identificativo, la targa del mezzo con cui è stato spedito a destinazione e il cliente dell'ordine.

Fraasi relative al cliente:

Della società richiedente si vuole conoscere il nome, partita iva e la città della sede.

Glossario dei termini principali

Il glossario dei termini che viene presentato ha la finalità di agevolare la comprensione e la decomposizione delle specifiche nonché l'individuazione dei concetti di maggior interesse.

TERMINI	DEFINIZIONI	SINONIMI	TERMINI COLLEGATI
Deposito	Edificio dove viene sistemato ciascun carico che arriva in azienda o che non viene utilizzato	Struttura, magazzino	Lotto
Società Terza	Azienda che coltiva e/o rivende il prodotto		Coltivatore, Venditore
Carico	Parte di un lotto che arriva in uno stesso container tramite mezzo di trasporto		Lotto, Cassa, Container, Camion
Lotto	Insieme di casse omogenee		Venditore, Coltivatore, Deposito, Carico, Custom Clearance
Cassa	Unità minima di quantità di tabacco	Collo	Cassa Splittata, Carico, Referto, Batch
Mezzo di trasporto	Modalità con cui il carico arriva a destinazione, ovvero attraverso un camion o una nave		Camion, Nave, Ordine
Referto	Relazione scritta emessa da un laboratorio a seguito delle analisi e che riporta i risultati dell'idoneità del tabacco appartenente alla cassa analizzata	Risultato	Laboratorio, Cassa
Ordine di spedizione	Numero di casse di ogni lotto da cui si vuole prendere il tabacco; viene richiesto da un cliente che fornisce una specifica composizione dell'ordine	Ordine	Batch, Mezzo di trasporto, Custom Clearance, Cliente
Cliente	Società che richiede un ordine all'azienda	Società richiedente	Ordine
Custom clearance	Allegato che certifica il numero di casse di un lotto da sdoganare poiché aventi status FP o EXEU	Allegato	Lotto, Ordine

Tab. 1: Glossario dei termini principali

Elenco delle operazioni

L'elenco delle operazioni principali per la base di dati completa la fase preliminare della progettazione concettuale. L'informazione e la frequenza con cui sono svolte tali operazioni diventano determinanti in fase di progettazione logica in termini di ottimizzazione dell'informazione da rappresentare.

- OP1:** Aggiunta di un lotto, definendo il venditore, il coltivatore e il deposito in cui viene collocato (25 volte al mese);
- OP2:** Aggiunta di una cassa (12500 volte al mese, circa 500 casse a lotto);
- OP3:** Aggiunta di un carico definendo il mezzo di arrivo (125 volte al mese, circa 5 carichi per lotto);
- OP4:** Registrazione di un nuovo venditore (3 volte all'anno);
- OP5:** Registrazione di un nuovo coltivatore (8 volte all'anno);
- OP6:** Inserimento di una nave (7 volte al mese);
- OP7:** Aggiunta di un container, indicando la nave con cui è stato trasferito (80 volte al mese);
- OP8:** Inserimento di un camion (5 volte al mese);
- OP9:** Registrazione di un nuovo cliente (3 volte all'anno);
- OP10:** Registrazione di un nuovo ordine e eventuale custom clearance (20 volte al mese);
- OP11:** Registrazione di una spedizione di un ordine (20 volte al mese);
- OP12:** Registrazione di una nuova ricetta (10 volte al mese);
- OP13:** Registrazione di un custom clearance (8 volte al mese);
- OP14:** Modifica del tipo di una cassa (400 volte al mese);
- OP15:** Modifica di status di un lotto (10 volte al mese);
- OP16:** Cancellazione di un cliente (1 volta all'anno);
- OP17:** Cancellazione di un venditore (1 volta all'anno);
- OP18:** Cancellazione di un coltivatore (3 volte all'anno);
- OP19:** Cancellazione di una ricetta (5 al mese);
- OP20:** Stampare la quantità di casse, raggruppate per lotti, presenti nei magazzini (10 volte al mese);
- OP21:** Stampare la quantità in kg e in numero di casse di tabacco arrivato in magazzino in un intervallo di tempo raggruppato per lotto (4 volte al mese);
- OP22:** Stampare il numero di casse di tabacco spedito, in un determinato intervallo, raggruppato per lotto (4 volte al mese);
- OP23:** Stampare il numero di casse di tabacco, raggruppato per lotto, che ha cambiato di status, stampando anche lo status di partenza (3 volte al mese);
- OP24:** Stampare la quantità in kg e in numero di casse di tabacco, raggruppato per lotto, andato in trattamento in un intervallo di tempo (1 volta al mese);
- OP25:** Stampare la quantità in kg e in numero di casse della merce, raggruppata per lotto, ispezionate in un intervallo di tempo (1 volta al mese);
- OP26:** Stampare il numero di analisi fatte in un anno con cadenza mensile (1 volta al mese);
- OP27:** Stampare il numero di split effettuati in un anno con cadenza mensile (1 volta al mese);
- OP28:** (100 al mese) Stampare un report dell'intero iter della cassa nel magazzino comprensivo di:
- Lotto di appartenenza (comprensivo di status e trasporto);
 - Data di arrivo (eventuale di fine quarantena e di trattamento);
 - Nazione di Origine e provenienza del tabacco;
 - La data di ispezione, se ispezionata;
 - L'eventuale referto;
 - Ordine con cui è stata spedita (con data e mezzo di spedizione).

Schema Entità-Relazione

La progettazione concettuale di una base di dati consiste nella costruzione di uno schema Entità-Relazione in grado di descrivere le specifiche raccolte durante la fase preliminare. I passi successivi mirano inoltre a raffinare la rappresentazione dei concetti primari.

La strategia che verrà illustrata nel seguito è una strategia ibrida la quale prevede la creazione di uno schema scheletro, sulla base dei concetti principali precedentemente individuati, e la raffinazione di ogni componente secondo la strategia top-down, aggiungendo inoltre concetti non ancora presentati secondo la strategia inside-out.

Il modello concettuale viene costruito partendo dai due concetti cardine Lotto e Carico e si procede a macchia d'olio aggiungendo allo schema tutti gli altri concetti richiesti.

Lo schema scheletro di primo livello è quindi composto dalle due entità **LOTTO** e **CARICO** legate dalla relazione *Divisione*, le cui occorrenze rappresentano i carichi in cui ciascun lotto viene suddiviso al fine di facilitarne il trasporto.



Figura 1 : Schema scheletro (primo livello)

Poiché questo schema non permette di rappresentare correttamente tutte le possibili relazioni delle due entità, si passa alla fase di raffinazione dei concetti.

Lo schema di secondo livello specifica le principali relazioni di ciascuna entità cardine, procedendo ad analizzare la prima fase di ricevimento del prodotto. Si analizza l'entità **LOTTO** la quale viene legata all'entità **DEPOSITO** tramite la relationship *Collocazione* per indicare l'edificio in cui sono sistemati i lotti quando arrivano in magazzino e quando non vengono utilizzati momentaneamente per alcun ordine. Viene inserita nello schema anche la generalizzazione totale e sovrapposta che vede come entità padre **SOCIETÀ TERZE** e come figlie i concetti **VENDITORE**, **COLTIVATORE**. Quest'ultimi sono legati rispettivamente tramite le relationship *Provenienza* e *Origine* all'entità **LOTTO**.

Per descrivere l'entità **CARICO** si analizzano due aspetti: la composizione del carico e il mezzo di trasporto con cui esso viene portato in magazzino. Il primo elemento viene espresso mediante la relazione *Composizione* che associa ogni occorrenza dell'entità **CASSA** un solo carico di appartenenza, sebbene quest'ultimo sia formato da una o più casse nel suo complesso. Il secondo elemento, invece, viene rappresentato dall'entità **MEZZO DI TRASPORTO**, la quale è una generalizzazione totale ed esclusiva delle tipologie ammesse per questo tipo di trasporto, e pertanto le entità figlie sono **NAVE** e **CAMION**. In particolare l'entità **CAMION** viene collegate tramite la relationship *Trasferimento#2* all'entità **CARICO**. Inoltre è bene distinguere che, sebbene ogni camion possa trasportare un carico alla volta, la nave può trasportare più carichi e pertanto l'entità **CONTAINER**, in cui viene stivato il carico, si lega all'entità **NAVE** attraverso l'associazione *Stivaggio* e all'entità **CARICO** tramite la relationship *Trasferimento#1*.

Lo schema che si ottiene questo punto è il seguente.

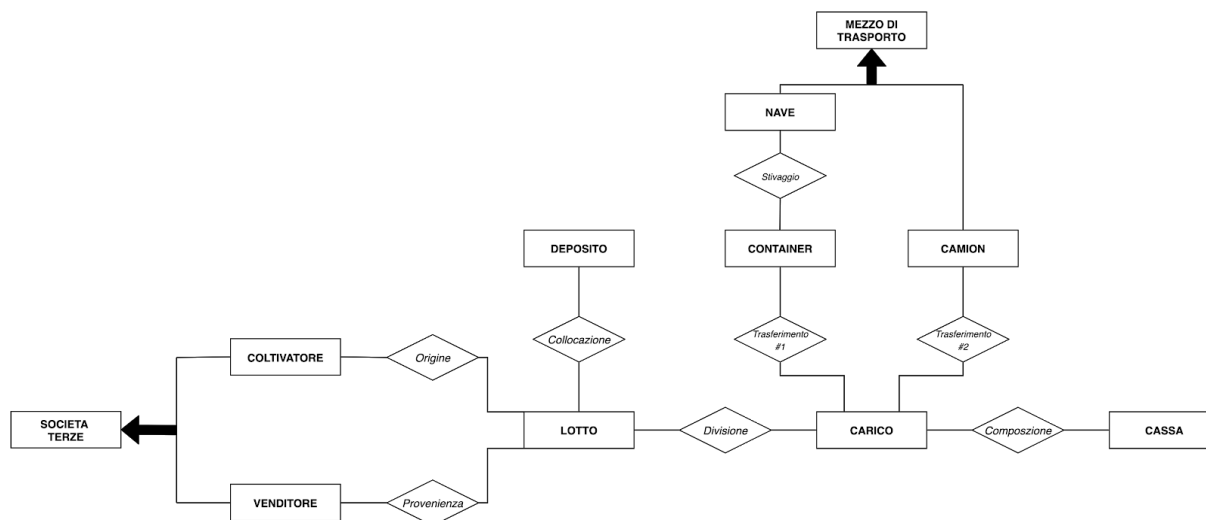


Figura 2 : Prima raffinazione dello schema (secondo livello)

Per raffinare ulteriormente lo schema, si introducono nuovi concetti relativi all'iter per la verifica della qualità del prodotto, suddividendo gli step delle casse e del carico. Per esplicitare tale distinzione fatta nel documento, si individuano i macro concetti che riguardano ciascuna entità già presente nello schema di secondo livello.

Il processo di esaminazione prevede innanzitutto un controllo sul **CARICO**, il quale viene sottoposto a un controllo ingresso in base al quale potrebbe essere messo in isolamento e successivamente anche eventualmente trattato. Questi concetti vengono esplicitati con degli attributi composti che rappresentano le date delle tre situazioni, espresse in forma di giorno, mese ed anno, e alcune delle quali sono sovrapposte: *Data di controllo*, sovrapposta alla data di inizio eventuale quarantena; *Data di fine quarantena*; *Data del trattamento*.

La fase del controllo di qualità avviene su alcune delle occorrenze di cassa prese a campione. Tale step viene così rappresentato dall'attributo composto *Data ispezione*, espressa in giorno, mese, anno, e dall'associazione *Analisi* che collega l'entità **CASSA** alla nuova entità **REFERTO**, ovvero l'allegato, uno per ogni cassa esaminata, che riporta le analisi effettuate. Le specifiche pertanto richiedono che una nuova entità **LABORATORIO** sia legata a sua volta all'entità **REFERTO** attraverso la relationship *Esame*, in modo tale da rappresentare il laboratorio che ha effettuato le analisi molecolari sul tabacco di ciascuna cassa campione.

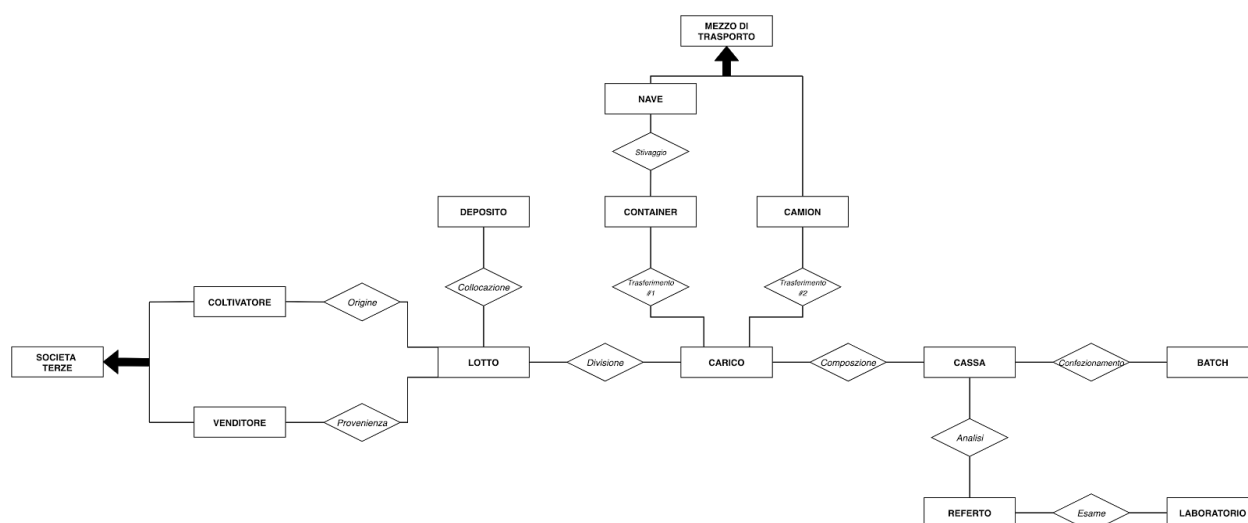


Figura 3 : Evoluzione dello schema (terzo livello)

Lo schema fin qui realizzato non rappresenta ancora le informazioni relative alla fase finale dell'iter. Il concetto principale di questa fase è la nuova entità **ORDINE**, la quale richiede numerosi collegamenti per assemblare alcuni elementi dello schema di terzo livello. Infatti il testo richiede che ogni ordine sia composto da uno o più batch, seguendo una ricetta indicata; pertanto si crea un'entità **BATCH** la quale viene connessa all'entità **ORDINE** tramite la relationship *Per* e all'entità **RICETTA** tramite *Formazione*. Inoltre è necessario specificare la composizione di ciascuna ricetta in base al lotto di interesse da cui si va a prelevare un certo numero di casse: per semplicità si utilizza l'entità **DETTAGLIO** legata alla ricetta tramite la relazione *Specifiche*. Poiché ogni cassa appartiene a un singolo batch tramite l'associazione *Confezionamento* si legano l'entità **CASSA** e l'entità **BATCH**, per rappresentare tutte le casse che compongono un determinato batch. Si deve mostrare tuttavia la possibilità, espressa nei requisiti, che una cassa possa subire il processo di split e per fare ciò si introduce una generalizzazione parziale ed esclusiva di una sola entità figlia **CASSA SPLITTATA**, specializzazione di **CASSA**. Inoltre gli ordini ricevuti dall'azienda vengono emessi da uno o più clienti, i quali vengono rappresentati dall'entità **CLIENTE** connessa a **ORDINE** per mezzo della correlazione *Richiesta*. Inoltre si deve mostrare l'opzionalità di un allegato dell'ordine nel caso in cui un lotto abbia uno status che necessita di sdoganamento. Si crea così un ciclo collegando l'entità **ORDINE** mediante la relazione *Allegato* a una nuova entità **CUSTOM CLEARANCE** che successivamente si collega attraverso l'associazione *Sdoganamento* all'entità cardine **LOTTO**.

Le ultime informazioni richieste per la base di dati sono quelle riguardanti la logistica dei trasporti, che si occupa della gestione della rete di distribuzione della merce, secondo gli accordi intercorsi fra l'azienda e il cliente. Poiché sono già presenti nello schema le entità necessarie a rappresentare le occorrenze, è sufficiente mettere in relazione le entità **MEZZO DI TRASPORTO** e **ORDINE** attraverso l'associazione *Spedizione* per descrivere la conclusione dell'iter.

Di seguito viene rappresentato lo schema con queste ultime aggiunte.

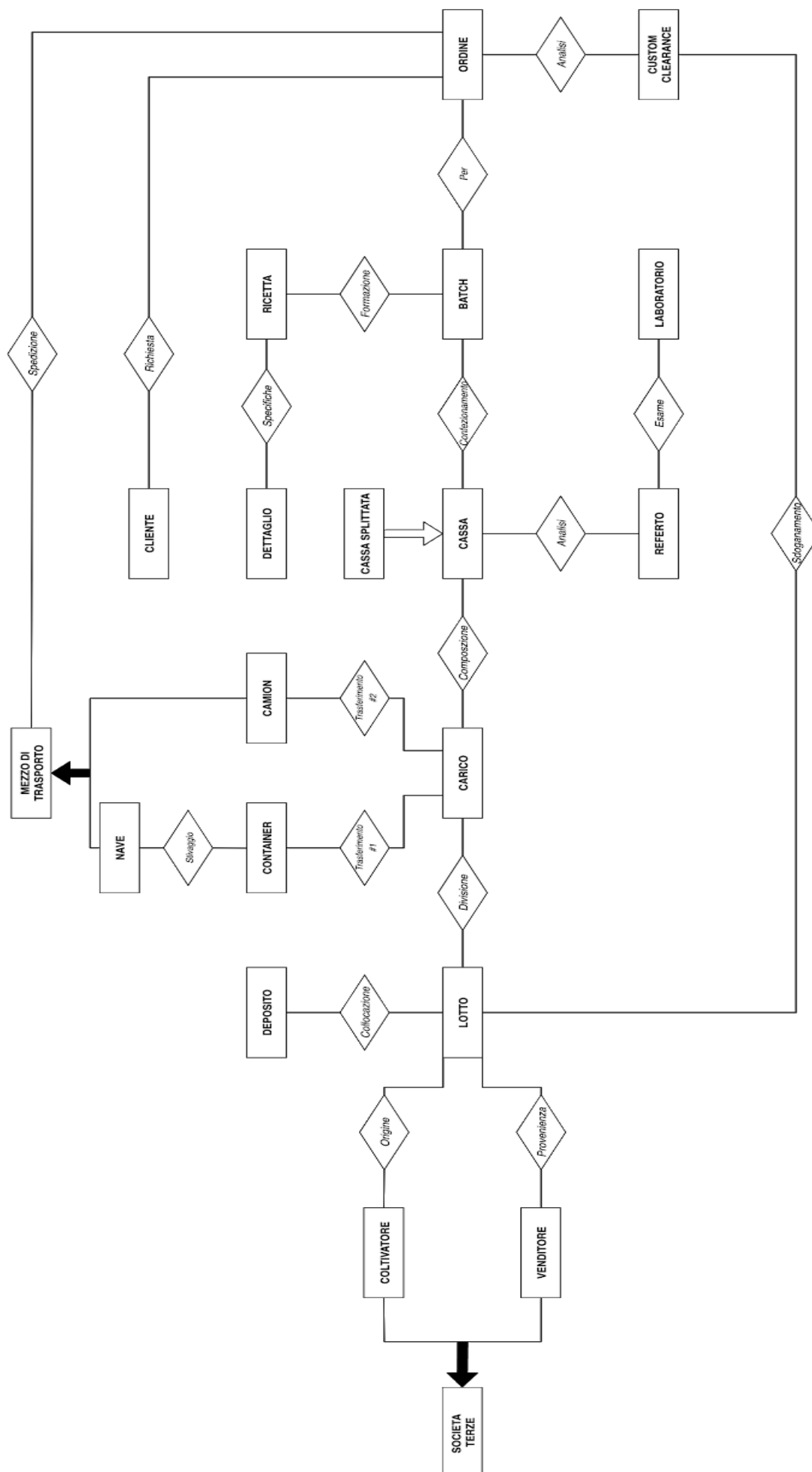


Figura 4 : Evoluzione dello schema (quarto livello)

Per terminare la fase di progettazione concettuale è necessario completare lo schema di Figura 4 con alcune informazioni qualitative e quantitative, ovvero con gli attributi, le cardinalità delle relazioni e gli identificatori delle entità.

La maggior parte di queste informazioni sono ricavate ancora una volta dall'analisi delle specifiche. Per rendere più veritiera la descrizione della realtà rappresentata dallo schema, viene aggiunto l'attributo ex novo *Data Spedizione* per l'entità CASSA. Questo attributo viene dedotto in maniera implicita dall'intervista.

Lo schema E-R completo viene mostrato in figura 5.

Vincoli non esprimibili

A volte risulta impossibile esprimere nello schema concettuale tutte le informazioni presenti nelle specifiche dei dati. Le informazioni che non possono essere rappresentate con i costrutti del modello E – R vengono allora elencate a parte e catalogate come vincoli non esprimibili. Di queste informazioni si deve comunque tenere conto nelle fasi successive.

- ★ Tutti i carichi di un lotto devono essere immagazzinati in un unico deposito;
- ★ Un carico può contenere al massimo 100 casse;
- ★ Da ogni lotto devono essere prelevate 20 casse campione per l'analisi in laboratorio del tabacco in esse contenuto;
- ★ Lo status di un lotto dipende dalla nazionalità del venditore;
- ★ Un lotto può essere sdoganato se ha status FP o EXEU e una volta sdoganato assume lo status di FC;
- ★ Un batch viene assemblato seguendo un'unica ricetta;
- ★ Una cassa può cambiare di tipo solo se è di tipo C48;
- ★ Una cassa C48 pesa 200 kg, mentre una C96 pesa 100 kg;
- ★ Una cassa splittata ha cardinalità massima di 2 per ogni cassa (cassa originaria C48 splittata in cassa C96 parte A e in cassa C96 parte B);
- ★ La data di controllo di un carico corrisponde alla sua data di arrivo in magazzino;
- ★ Un magazzino contiene fino a 80.000 lotti.

Dizionario dei dati (Entità)

Entità	Identificatore	Attributi	Descrizione
SOCIETÀ TERZE	Partita IVA	Telefono, Nazionalità, Ragione Sociale, Città Sede, Indirizzo	Società che possono essere coltivatrici e/o venditrici di tabacco
COLTIVATORE	Partita IVA	Telefono, Nazionalità, Ragione Sociale, Città Sede, Indirizzo	Coltivatori di tabacco
VENDITORE	Partita IVA	Telefono, Nazionalità, Ragione Sociale, Città Sede, Indirizzo	Società venditrice di tabacco
DEPOSITO	ID_Deposito	Capacità MAX, Città, Indirizzo	Capannone dove sono conservati i lotti
CONTAINER	ID_Container		Container con cui è trasportato un lotto
MEZZO DI TRASPORTO	Targa		Mezzo attraverso il quale viene trasferito un carico
NAVE	Targa		Mezzo di trasporto via mare
CAMION	Targa		Mezzo di trasporto via terra
LOTTO	ID	Varietà, Forma, Status, Trasporto, Origine, Provenienza	Insieme di carichi contenenti casse omogenee
CARICO	Numero Progressivo, Lotto	Data Trattamento, Data Fine Quarantena, Data di Controllo	Insieme di casse arrivate con lo stesso mezzo di trasporto
CASSA	Barcode	Peso, Tipo Imballaggio, Data Spedizione, Data Ispezione	Unità minima di contenimento del tabacco
CASSA SPLITTATA	Partizionamento, Data_Split		Cassa divisa a metà
REFERTO	Codice_Referto	Data Rilascio, %NIC, %RS, %NO3, %NH3	Risultato delle analisi
LABORATORIO	Nome, Indirizzo	Telefono, Sede Città	Luogo dove viene analizzato il tabacco
BATCH	Numero_Progressivo, Ordine		Sottoinsieme di casse appartenenti ad un ordine

RICETTA	ID_Ricetta		Ricetta di composizione
DETTAGLIO	Numero_Lotto, Ricetta	Casse_per_Lotto	Specifiche di composizione di ogni ricetta
ORDINE	Codice	Casse totali, Num batch richiesti	Lotti dai quali si devono prendere le casse
CUSTOM CLEARANCE	ID, Codice		Numero di casse da dover sdoganare
CLIENTE	Partita IVA	Indirizzo, Sede Città, Nome, Telefono	Società richiedente

Tabella 2: Dizionario dei dati (entità)

Dizionario dei dati (Relazioni)

Relazioni	Entità	Attributi	Descrizione
PROVENIENZA	Venditore, Lotto		Indica presso quali venditori sono stati acquistati i lotti
COLLOCAZIONE	Deposito, Lotto		Sistemazione di ciascun lotto in un deposito
DIVISIONE	Lotto, Carico		Suddivisione di un lotto arrivato in più carichi
TRASFERIMENTO #1	Carico, Container	Data Partenza, Porto arrivo	Container con cui ciascun carico è trasportato
TRASFERIMENTO #2	Carico, Camion	Data Partenza	Camion con cui ciascun lotto è stato trasportato
STIVAGGIO	Container, Nave		Nave che trasporta il container che contiene un carico
COMPOSIZIONE	Carico, Cassa		Casse che compongono ciascun carico
CONFEZIONAMENTO	Cassa, Batch		Casse utilizzate per un determinato batch
ANALISI	Cassa, Referto		Referto molecolare associato a ogni cassa analizzata a campione
ESAME	Referto, Laboratorio		Referto emesso da un certo laboratorio che ha effettuato l'analisi molecolare sul tabacco
PER	Batch, Ordine		Ordine che richiede un certo batch per la sua composizione
FORMAZIONE	Batch, Ricetta		Formula con cui viene assemblato un singolo batch
SPECIFICHE	Ricetta, Dettaglio		Specifica i dettagli di ciascuna ricetta
SPEDIZIONE	Ordine, Mezzo di Trasporto	Data	Ordine che viene spedito attraverso un mezzo di trasporto a destinazione
RICHIESTA	Ordine, Cliente		Cliente che richiede un certo ordine dall'azienda
ALLEGATO	Ordine, Custom Clearance		Documento di Custom Clearance allegato a ciascun

			ordine che richiede un lotto sdoganato
SDOGANAMENTO	Ordine, Lotto	Status precedente	Operazione di sdoganamento di un lotto cui viene associato un Custom Clearance

Tabella 3: Dizionario dei dati (relazioni)

PROGETTAZIONE LOGICA

In questa fase, al fine di creare uno schema logico consono per una traduzione diretta in schema relazionale, è necessario riorganizzare e modificare lo schema E-R ottenuto durante la progettazione concettuale.

Tavola dei volumi

I valori del volume vengono stimati su un arco temporale di 80 mesi, unità di tempo minima per definire il volume numerico di ogni entità e relazione del database dell'azienda committente.

Concetto	Tipo	Volume
LOTTO	E	2.000
CASSA	E	1.000.000
CARICO	E	10.000
SOCIETÀ TERZA	E	120
VENDITORE	E	20
COLTIVATORE	E	100
CONSORZIO	E	25
AZIENDA	E	75
MEZZO DI TRASPORTO	E	9.400
NAVE	E	1.500
CONTAINER	E	8.000
CAMION	E	7900
REFERTI	E	40.000
LABORATORIO	E	10
ORDINE	E	1.600
BATCH	E	8.000

RICETTA	E	400
DETTAGLIO	E	2000
CUSTOM CLEARANCE	E	300
CLIENTE	E	50
DEPOSITO	E	3
COLLOCAZIONE	R	2.000
COMPOSIZIONE	R	1.000.000
DIVISIONE	R	10.000
PROVENIENZA	R	2.000
STIVAGGIO	R	8.000
TRASFERIMENTO	R	10.000
ANALISI	R	40.000
ESAME	R	40.000
CONFEZIONAMENTO	R	1.000.000
PER	R	8.000
FORMAZIONE	R	8.000
SPECIFICHE	R	4000
ALLEGATO	R	300
SDOGANAMENTO	R	1.000
RICHIESTA	R	1.600
SPEDIZIONE	R	6.400
TRASFERIMENTO #1	R	2.000
TRASFERIMENTO #2	R	8.000

Tabella 4: Tavola dei volumi

Tavola delle operazioni

L'ipotesi alla base della definizione del tipo di ogni operazione è la seguente: un'operazione per essere considerata interattiva deve essere svolta almeno una volta a settimana, ovvero 4 volte al mese.

Operazione	Tipo	Frequenza
OP1: Aggiunta di un lotto	I	25 volte al mese
OP2: Aggiunta di una cassa	I	2500 volte al mese (circa 500 casse a lotto)
OP3: Aggiunta di una carico	I	125 volte al mese (circa 5 carichi per lotto)
OP4: Registrazione di un nuovo venditore	B	3 volte all'anno
OP5: Registrazione di un nuovo coltivatore	B	8 volte all'anno
OP6: Inserimento di una nave	I	7 volte al mese
OP7: Aggiunta di un container	I	80 volte al mese
OP8: Inserimento di un camion	I	5 volte al mese
OP9: Registrazione di un nuovo cliente	B	3 volte all'anno
OP10: Registrazione di un nuovo ordine	I	20 volte al mese
OP11: Registrazione di una spedizione di un ordine	I	20 volte al mese
OP12: Registrazione di una nuovo ricetta	I	10 volte al mese
OP13: Registrazione di un custom clearance	I	8 volte al mese
OP14: Modifica del tipo di cassa	I	400 volte all'anno
OP15: Modifica dello status di un lotto	I	10 volte al mese
OP16: Cancellazione di un cliente	B	1 volte all'anno
OP17: Cancellazione di un venditore	B	1 volte all'anno
OP18: Cancellazione di un coltivatore	B	3 volte all'anno

OP19: Cancellazione di una ricetta	I	5 volte al mese
OP20: Stampare la quantità di casse, raggruppate per lotti, presenti nei magazzini	B	10 volte al mese
OP21: Stampare la quantità di tabacco arrivato in un arco temporale	I	4 volte al mese
OP22: Stampare la quantità il numero di casse spedite	I	4 volte al mese
OP23: Stampare il numero di casse dei lotti che hanno cambiato status	B	3 volte al mese
OP24: Stampare la quantità di tabacco e il numero di casse in trattamento	B	1 volta al mese
OP25: Stampare la quantità di tabacco e il numero di casse in ispezione qualità	B	1 volta al mese
OP26: Stampare il numero di analisi effettuate in un anno con cadenza mensile	B	1 volta al mese
OP27: Stampare il numero di split effettuati in un anno con cadenza mensile	B	1 volta al mese
OP28: Stampare un report dell'intero iter di una cassa	I	100 volte al mese

Tabella 5: Tavola delle operazioni

Analisi delle ridondanze

Una ridondanza, in uno schema concettuale, consiste nella presenza di un dato che può essere derivato da altri dati mediante una serie di operazioni. L'analisi delle ridondanze, basata sui costi in termini di spazio e tempo, è necessaria per comprendere se una data ridondanza comporta nel complesso un vantaggio od uno svantaggio.

Osservazioni su attributi ridondanti

Gli unici attributi che possono risultare essere ridondanti sono *Origine* e *Provenienza* nell'entità **LOTTO** in quanto l'unica operazione che li coinvolge è l'OP28 la quale ha una frequenza di 100 volte al mese e richiede degli accessi in lettura. Di conseguenza se ipotizziamo una dimensione di 4 byte per ogni attributo si ha un'occupazione di memoria di $4 * 1.000 = 4.000$ byte in più per ottenere un vantaggio di soli 400 accessi in lettura per ricavare sia l'origine che la provenienza.

Num_Casse_Iniziale

Dall'osservazione dello schema ER e da una prima analisi delle tabelle degli accessi per le operazioni definite precedentemente, si nota che potrebbe essere utile inserire un attributo ridondante *Num_Casse_Iniziale* nell'entità **CARICO** o nell'entità **LOTTO** per indicare il numero di casse appartenenti ad un carico e/o ad un lotto al momento della consegna e il loro peso totale. Le operazioni che richiedono di conoscere i valori indicati dagli attributi appena descritti sono:

- **OP21:** Stampare la quantità in kg e in numero di casse di tabacco arrivato in magazzino in un intervallo di tempo raggruppato per lotto (4 volte al mese);
- **OP23:** Stampare la quantità in kg e in numero di casse di tabacco, raggruppato per lotto, che ha cambiato di status, stampando anche lo status di partenza(3 volte al mese);

Gli schemi ER risultanti sarebbero i seguenti:

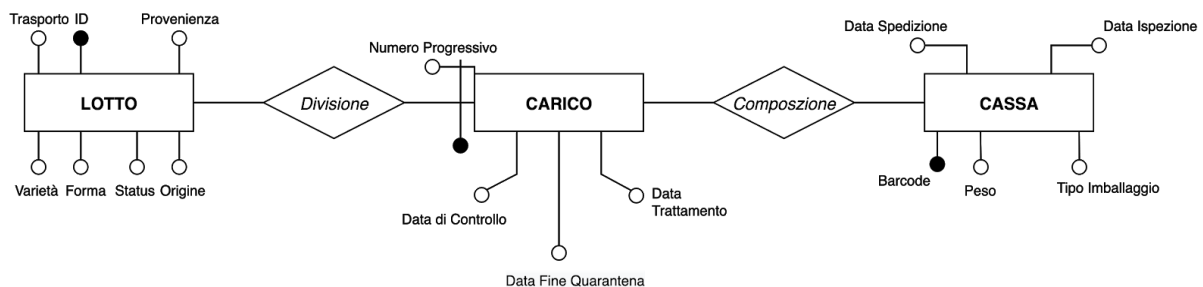


Figura 11.a : Caso senza ridondanze

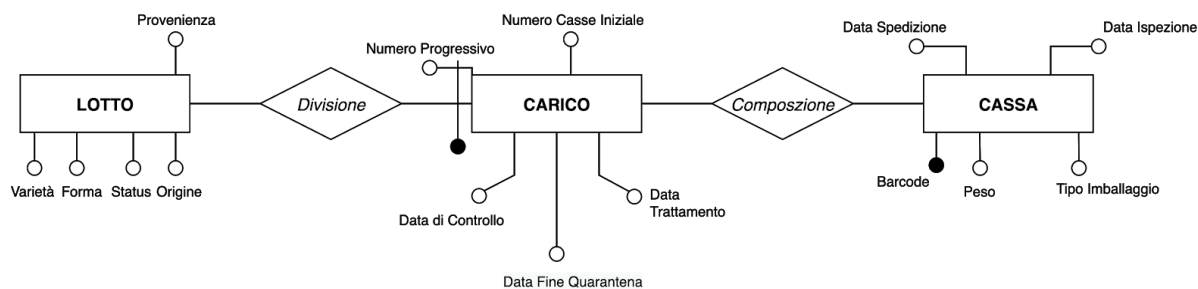


Figura 11.b : Ridondanze su CARICO

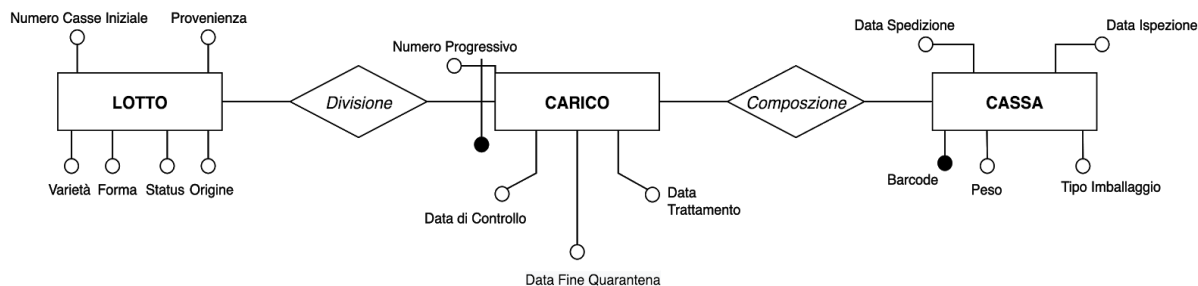


Figura 11.c : Ridondanze su LOTTO

Per lo studio delle ridondanze si riportano le Tavole degli Accessi di ogni singola operazione caso per caso. Per l'operazione 21 si considera un arco temporale medio di un mese. Nell'operazione 23 si assume invece che i lotti con status EXEU o FC siano circa il 50% del totale.

1) Caso senza ridondanze

Se non viene aggiunto nessun attributo si hanno le seguenti tavole

OP21

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Carico	E	125	L
Composizione	R	6.250	L
Cassa	E	6.250	L

Frequenza = 10 volte al mese

Numero di accessi totale per OP21 = 12.625 Letture

Costo_in_accessi (OP21) = 12.625 * 10 = 126.250 L

OP23

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Lotto	E	1.000	L
Sdoganamento	R	1.000	L
Divisione	R	5.000	L
Carico	E	5.000	L
Composizione	R	250.000	L

Cassa	E	250.000	L
-------	---	---------	---

Frequenza = 4 volte al mese

Numero di accessi totale per OP23 = 262.000 Letture

Costo_in_accessi (OP23) = 262.000 * 4 = 1.048.000 Letture

Costo_in_accessi_totale_caso1 = 126.250 L + 1.048.000 L = 1.174.250 Letture

2) Caso con Num_Casse_Iniziale aggiunto a CARICO

Quando si aggiungono i due attributi ridondanti su carico è necessario considerare anche l'operazione di inserimento di una cassa indicata con OP2 nell'elenco delle operazioni.

Il costo di una scrittura viene di seguito valutato come due accessi in lettura.

OP2

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cassa	E	1	S
Composizione	R	1	S
Carico	E	1	L
Carico	E	1	S

Frequenza = 12.500 al mese

Numero di accessi totale per OP2 = 7 Letture

Costo_in_accessi(OP2) = 12.500 * 7 = 87.500 Letture

OP21

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Carico	E	125	L

Frequenza = 10 volte al mese

Numero di accessi totale per OP21 = 125 Letture

Costo_in_accessi (OP21) = 10 * 125 L = 1.250 Letture

OP23

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Lotto	E	1000	L
Sdoganamento	R	1000	L
Divisione	R	5000	L
Carico	E	5000	L

Frequenza = 4 volte al mese

Numero di accessi totale per OP23 = 12.000 Letture

Costo_in_accessi (OP23) = 4 * 12.000 L = 48.000 Letture

Costo_in_accessi_totale_caso2 = 48.000 L + 1.250 L + 87.500 L = 136.750 Letture

3) Caso con Num_Casse_Iniziale aggiunto a LOTTO

Anche in questo caso è da considerare l'operazione numero 2 in quanto comporta una scrittura sull'occorrenza dell'entità LOTTO a cui appartiene.

OP2

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cassa	E	1	S
Composizione	R	1	S
Carico	E	1	L
Divisione	E	1	L
Lotto	E	1	L
Lotto	E	1	S

Frequenza = 12.500 al mese

Numero di accessi totale per OP2 = 10 Letture

Costo_in_accessi (OP2) = 12.500 * 10 = 125.000 Letture

OP21

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Carico	E	125	L
Composizione	R	6.250	L
Cassa	E	6.250	L

Frequenza = 10 volte al mese

Numero di accessi totale per OP21 = 12.625 Letture

Costo_in_accessi (OP21) = 12.625 * 10 = 126.250 L

OP23

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Lotto	E	1000	L
Sdoganamento	R	1000	L

Frequenza = 3 volte al mese

Numero di accessi totale per OP21 = 2.000 Letture

Costo_in_accessi (OP21) = 2.000 * 3 = 6.000 L

Costo_in_accessi_totale_caso3 = 125.000 L + 126.250 L + 6.000 L = 257.250 Letture

CONFRONTO DEI TRE CASI

Per confrontare correttamente i tre casi si calcola il costo in accessi dell'operazione 2 in assenza di ridondanza

OP2 senza ridondanze

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cassa	E	1	S
Composizione	R	1	S

Frequenza = 12.500 al mese

Numero di accessi totale per OP2 = 4 Letture

Costo_in_accessi (OP2) = $12.500 * 4 = 50.000$ Letture

Costo_in_accessi_totale_caso1 = $1.174.250 \text{ L} + 50.000 \text{ L} = 1.224.250$ Letture

Dunque se consideriamo solamente il numero degli accessi, la scelta più vantaggiosa è quella di aggiungere gli attributi ridondanti sull'entità CARICO (caso 2) in quanto si hanno i seguenti risultati:

Costo_in_accessi_totale_caso1 = $1.174.250 \text{ L} + 50.000 \text{ L} = 1.224.250$ Letture

Costo_in_accessi_totale_caso2 = $48.000 \text{ L} + 1.250 \text{ L} + 87.500 \text{ L} = 136.750$ Letture

Costo_in_accessi_totale_caso3 = $125.000 \text{ L} + 126.250 \text{ L} + 6.000 \text{ L} = 257.250$ Letture

Per quanto riguarda lo spazio occupato in memoria si devono fare alcune considerazioni:

- Un carico può contenere al più 100 casse (Vincoli non esprimibili) => 2 byte
- Un lotto può contenere al più 1000 casse (Vincoli non esprimibili) => 2 byte
- Inizialmente ogni cassa è di tipo C48 e dunque contiene 200 kg. Perciò un carico inizialmente può pesare al più 20.000 kg (=> 3 byte), mentre un lotto può raggiungere 200.000kg (=> 3 byte).

In conclusione si hanno i seguenti valori in termini di costi in numero di accessi e in memoria utilizzata:

Caso_1:

Costo_in_accessi_totale = $1.174.250 \text{ L} + 50.000 \text{ L} = 1.224.250$ Letture

Costo_in_memoria = stretto necessario

Caso_2:

Costo_in_accessi_totale = $48.000 \text{ L} + 1.250 \text{ L} + 87.500 \text{ L} = 136.750$ Letture

Costo_in_memoria = $5 \text{ byte} * 10.000 = 50.000$ byte

Caso_3:

Costo_in_accessi_totale = $125.000 \text{ L} + 126.250 \text{ L} + 6.000 \text{ L} = 257.250$ Letture

Costo_in_memoria = $5 \text{ byte} * 2.000 = 10.000$ byte

Pertanto si preferisce utilizzare la ridondanza su CARICO poiché i costi tra il caso 2 e il caso 3 sono simili, ma la frequenza di OP21 è maggiore rispetto a quella di OP23.

Num_casse_disponibili

Viene ora studiato l'inserimento di un attributo ridondante *Num_casse_disponibili* sull'entità **LOTTO** oppure su **MAGAZZINO** per ridurre il numero di accessi richiesti da **OP20**, ovvero “*Stampare la quantità di casse, raggruppate per lotti, presenti nei magazzini (10 volte al mese)*”. Nelle valutazioni viene fatta l'assunzione che tutti e tre i magazzini siano pieni e quindi che ci siano $80.000 * 3 = 240.000$ casse, le quali possono corrispondono in media a $240.000 / 500 = 480$ lotti (160 in ogni magazzino). Inoltre si fa la supposizione che ciascun lotto sia composto da 5 carichi da 100 colli ciascuno.

Le operazioni interessate, oltre a OP20, sono quelle che portano ad aggiungere o togliere una cassa dal magazzino, ovvero :

- **OP2**: Aggiunta di una cassa (12500 volte al mese, circa 500 casse a lotto);
- **OP11**: Registrazione di una spedizione di un ordine (20 volte al mese);

Gli schemi seguenti rappresentano i casi studiati:

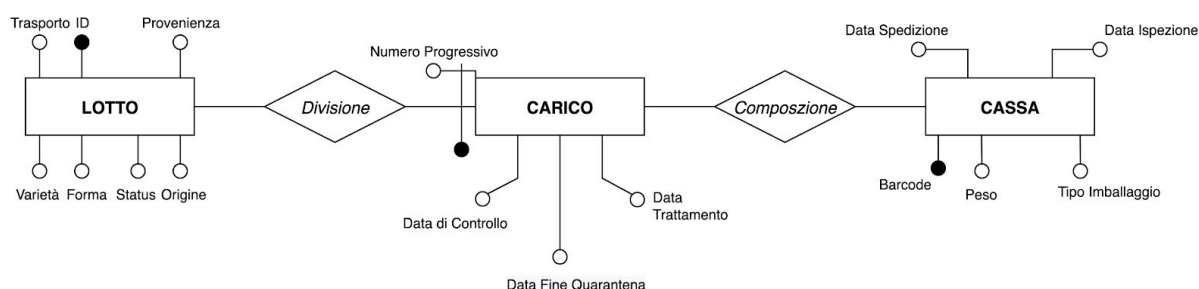


Figura 12.a : Senza ridondanze

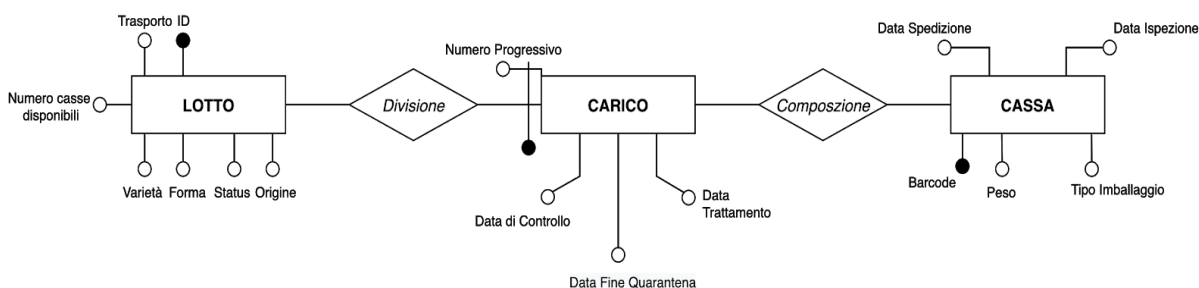


Figura 12.b : Ridondanze su LOTTO

Si analizza prima il caso con la ridondanza. Si ricorda che una scrittura (S) è valutata come due letture (L).

1) Caso con Num_casse_disponibili aggiunto a LOTTO

OP2

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cassa	E	1	S
Composizione	R	1	S
Carico	E	1	L

Divisione	E	1	L
Lotto	E	1	L
Lotto	E	1	S

Frequenza = 12.500 al mese

Numero di accessi totale per OP2 = 9 Letture

Costo_in_accessi (OP2) = 12.500 * 9 L = 112.500 Letture

OP11

I valori riportati nella tabella seguente sono ottenuti considerando che un ordine coinvolga mediamente casse di tre lotti diversi.

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Spedizione	R	1	S
Ordine	E	1	L
Per	R	5	L
Batch	E	5	L
Confezionamento	R	125	L
Cassa	E	125	S
Composizione	R	15	L
Carico	E	15	L
Divisione	R	3	L
Lotto	E	3	S

Frequenza = 20 al mese

Numero di accessi totale per OP2 = 427 Letture

Costo_in_accessi (OP11) = 20 * 427 L = 8.540 Letture

OP20

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Magazzino	E	3	L
Collocazione	R	480	L
Lotto	E	480	L

Frequenza = 20 al mese

Numero di accessi totale per OP2 = 963 Letture

Costo_in_accessi (OP20) = 20 * 963 L = 19.260 Letture

2) Caso senza ridondanze

OP2

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cassa	E	1	S
Composizione	R	1	S

Frequenza = 12.500 al mese

Numero di accessi totale per OP2 = 4 Letture

Costo_in_accessi (OP2) = 12.500 * 4 = 50.000 Letture

OP11

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Spedizione	R	1	S
Ordine	E	1	L
Per	R	5	L
Batch	E	5	L
Confezionamento	R	125	L
Cassa	E	125	S

Frequenza = 20 al mese

Numero di accessi totale per OP2 = 388 Letture

Costo_in_accessi (OP11) = 20 * 388 L = 7760 Letture

OP20

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Magazzino	E	3	L
Collocazione	R	480	L
Lotto	E	480	L
Divisione	R	2400	L
Carico	E	2400	L
Composizione	R	240000	L
Cassa	E	240000	L

Frequenza = 20 volte al mese

Numero di accessi totale per OP2 = 485.763 Letture

Costo_in_accessi (OP20) = 20 * 485.763 = 9.715.260 Letture

CONFRONTO DEI CASI

Consideriamo che per rappresentare il *Num_casse_disponibili* nel caso 1 (aggiunto a LOTTO) siano necessari 2 byte, mentre se si prende il caso 2 (aggiunto a MAGAZZINO) sono necessari 3 byte.

Caso_1:

Costo_in_accessi_totale = 112.500 L + 8.540 L + 19.260 L = 140.300 Letture

Costo_in_memoria = 2 byte * 2.000 = 4.000 byte

Caso_2:

Costo_in_accessi_totale = 50.000 L + 7.760 L + 9.715.260 L = 9.773.020 Letture

Costo_in_memoria = stretto necessari

Risulta evidente come c'è un notevole vantaggio ad aggiungere un attributo ridondante su lotto che ne indichi il numero di casse disponibili.

Eliminazione delle generalizzazioni

Le generalizzazioni non sono direttamente rappresentabili nei tradizionali sistemi per la gestione delle basi di dati. Per questo motivo, in questa fase, risulta spesso necessario trasformare questo costruito tramite l'utilizzo esclusivo di implementazioni naturali, ovvero di entità e relazioni.

Lo schema finale ottenuto (Figura 5) presenta due macro generalizzazioni: una riguardante l'entità SOCIETÀ TERZE e una l'entità MEZZO DI TRASPORTO. Si parte inizialmente con l'analizzare la tipologia di generalizzazione per individuare il metodo adatto per eliminare tali generalizzazioni.

Iniziamo l'analisi della generalizzazione dell'entità SOCIETÀ TERZE. La generalizzazione è totale e sovrapposta, in quanto una società terza può rappresentare solo un coltivatore, solo un venditore oppure un coltivatore che è anche venditore.

La strategia che si utilizza per questa ristrutturazione è l'accorpamento del genitore della generalizzazione nelle figlie, che è possibile solo nel caso di generalizzazione totale. Si procede pertanto con l'eliminazione dell'entità padre e, per la proprietà dell'ereditarietà, i suoi attributi e il suo identificatore vengono aggiunti alle entità figlie COLTIVATORE e VENDITORE. Rimangono invariate le relazioni *Origine* e *Provenienza* legate alle rispettive entità figlie.

Questa alternativa è stata scelta poiché nelle operazioni è di interesse distinguere solo tra le sue entità figlie, anche perché l'entità padre non viene utilizzata nelle operazioni. Inoltre in questo caso si ha un risparmio di memoria poiché, in linea di principio, gli attributi non assumono mai valori nulli, e anche una riduzione del numero di accessi poiché non si visita l'entità padre.

La figura seguente mostra lo schema prima e dopo l'analisi.

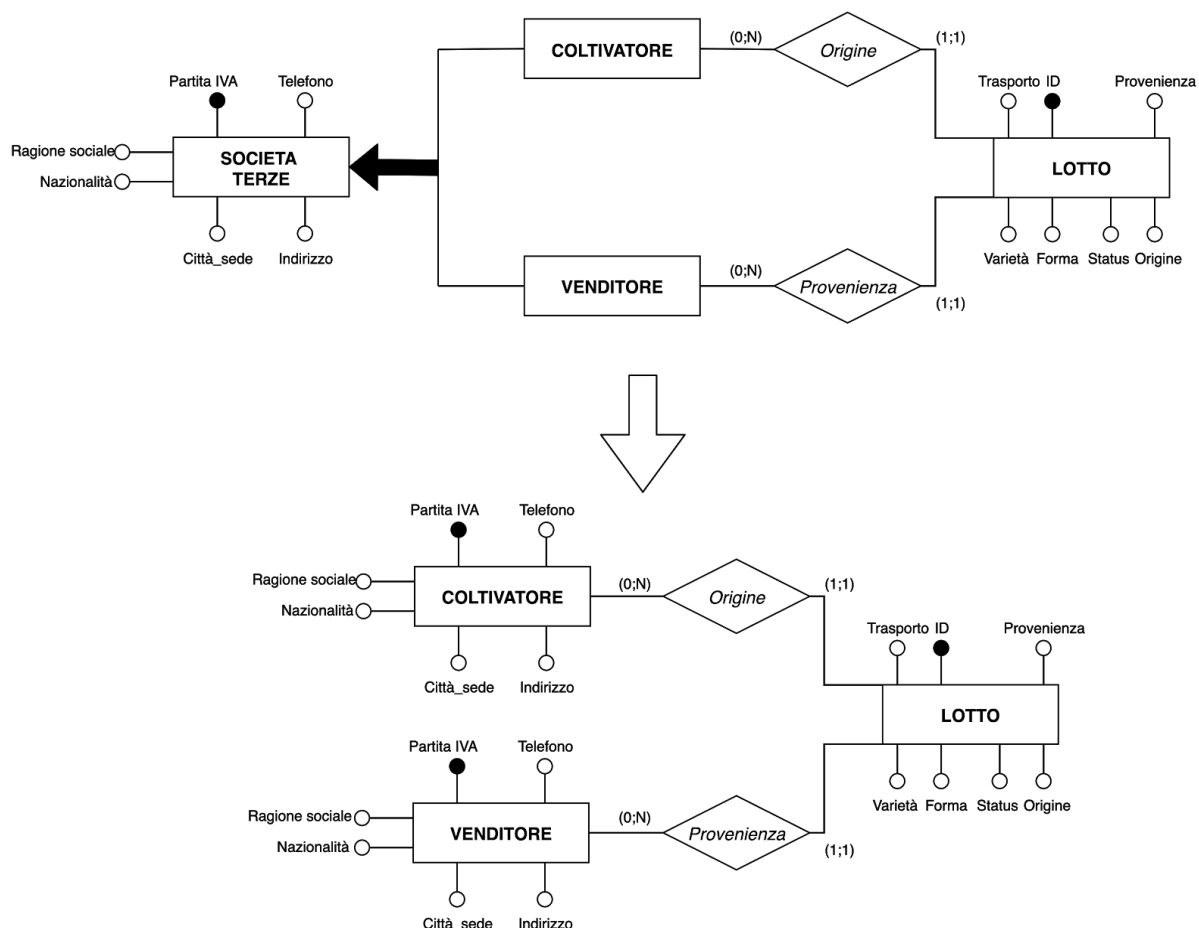


Figura 13 : Analisi e schema ristrutturato della generalizzazione SOCIETÀ TERZE

Si passa successivamente all'analisi della generalizzazione dell'entità **MEZZO DI TRASPORTO**. Questa generalizzazione è totale ed esclusiva, in quanto un mezzo di trasporto può essere un camion che parte dall'azienda venditrice e arriva direttamente a destinazione, oppure una nave che trasporta il carico in un container, trascurando le specifiche.

La strategia che si utilizza per questa ristrutturazione è l'accorpamento delle entità figlie della generalizzazione nel genitore. Le entità **CAMION** e **NAVE** vengono eliminate e le loro proprietà, che in questo caso riguardano solo la partecipazione ad associazioni, vengono aggiunte all'entità genitore **MEZZO DI TRASPORTO**. A tale entità viene aggiunto un ulteriore attributo *Tipo* che serve a distinguere il tipo di una occorrenza di **MEZZO DI TRASPORTO**, che può essere appunto un camion o una nave. Inoltre le associazioni originarie delle due entità figlie vengono rinominate per esprimere meglio il concetto di trasporto.

Questa alternativa è stata scelta poiché le operazioni non fanno molta distinzione tra le occorrenze di **MEZZO DI TRASPORTO**, **CAMION** e **NAVE**. Inoltre in questo caso non abbiamo molto spreco di memoria a causa della presenza di valori nulli perché le entità figlie non hanno attributi propri. Anche il numero degli accessi è minore poiché le occorrenze e gli attributi non sono distribuiti tra le varie entità. Lo schema finale che si ottiene per questa generalizzazione è il seguente.

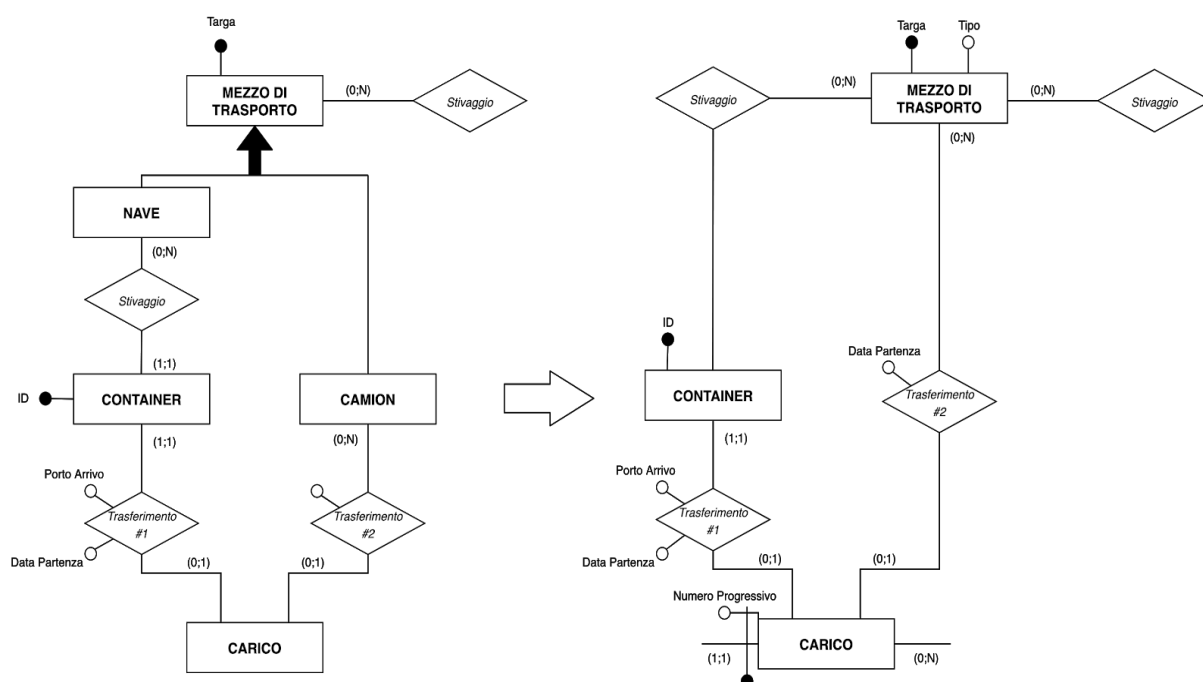


Figura 14 : Analisi e schema ristrutturato della generalizzazione MEZZO DI TRASPORTO

Infine si analizza la generalizzazione dell'entità **CASSA**. Questa generalizzazione è parziale ed esclusiva e coinvolge una sola entità figlia, **CASSA SPLITTATA**, che rappresenta una eventuale operazione a cui è sottoposta una occorrenza di **CASSA**.

La strategia che si utilizza per questa semplice ristrutturazione è la sostituzione della generalizzazione con una associazione che lega con una relazione uno a uno l'entità genitore **CASSA** con l'entità figlia **CASSA SPLITTATA**. In questo caso non ci sono trasferimenti di attributi o associazioni e l'entità **CASSA SPLITTATA** viene identificata esternamente dall'entità **CASSA**. Inoltre ogni occorrenza dell'entità **CASSA** può partecipare a una occorrenza di *Split*.

Questa alternativa, conveniente quando la generalizzazione non è totale, viene scelta poiché le operazioni si riferiscono solo a occorrenze di **CASSA** oppure di **CASSA SPLITTATA** e dunque fanno distinzioni tra le due entità. Si ha inoltre un risparmio di memoria per l'assenza di valori nulli, ma c'è un incremento degli accessi.

La seguente figura rappresenta la ristrutturazione effettuata a partire dalla porzione di schema originale.

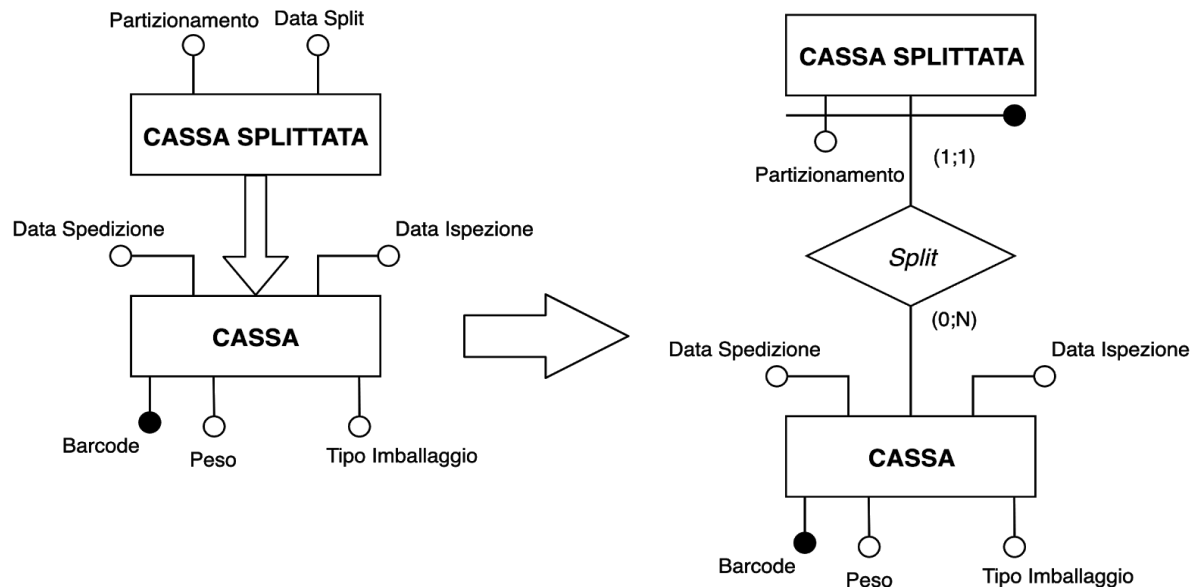


Figura 15 : Analisi e schema ristrutturato della generalizzazione CASSA

Accorpamento/partizionamento di relationship

L'accorpamento (operazione inversa del partizionamento) è un'operazione effettuata solitamente su associazioni con cardinalità di tipo uno a uno che porta ad accorpare due diverse entità. Il partizionamento orizzontale (operazione inversa dell'accorpamento) è un'operazione di decomposizione di relationship tra due entità in due (o più) relationship tra le medesime entità.

Lo schema non presenta possibili relationship da accorpare o partizionare.

Scelta degli identificatori primari

La scelta degli identificatori primari è essenziale per tradurre lo schema E-R nel modello relazionale, poiché in tale modello le chiavi vengono utilizzate per stabilire legami tra dati in relazioni diverse.

Come visibile dallo schema concettuale completo, la totalità delle entità possiede un solo identificatore, interno o esterno, che sarà assunto come identificatore primario.

Ristrutturazione dello schema E-R

Nella pagina seguente viene riportato lo schema E-R ristrutturato a seguito delle considerazioni appena esposte.

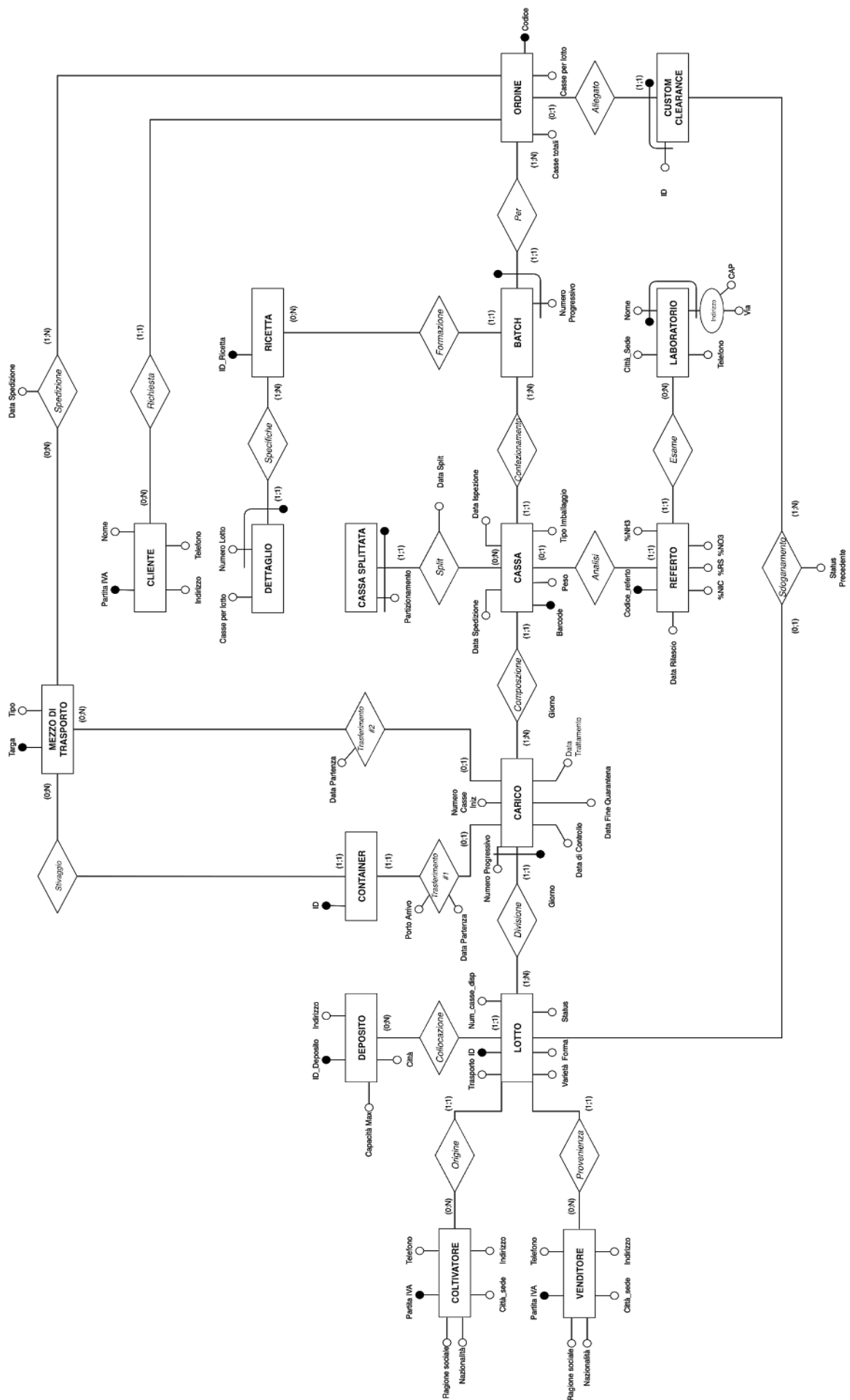


Figura 16 : Schema finale ristrutturato

Modello Relazionale

A partire dallo schema E-R ristrutturato si costruisce uno schema logico equivalente in grado di rappresentare le medesime informazioni. Tale schema, chiamato modello relazione, costituisce il risultato della progettazione logica.

Traduzione di Entità:

COLTIVATORE (Partita IVA, Telefono, Ragione Sociale, Nazionalità, Città_Sede, Indirizzo)

VENDITORE (Partita IVA, Telefono, Ragione Sociale, Nazionalità, Città_Sede, Indirizzo)

LOTTO (ID, Trasporto, Varietà, Forma, Status, Coltivatore, Venditore, Deposito, Num_casse_disp)

Vincoli di integrità referenziale:

- tra l'attributo *Coltivatore* in Lotto e la relazione *Coltivatore*;
- tra l'attributo *Venditore* in Lotto e la relazione *Venditore*;
- tra l'attributo *Deposito* in Lotto e la relazione *Deposito*.

DEPOSITO (ID_Deposito, Indirizzo, Capacità_Max, Città)

CARICO (Numero_Progressivo, Lotto, Data_di_controllo, Data_fine_Quarantena, Data_Trattamento, Num_casse_iniz)

Vincolo di integrità referenziale tra l'attributo *Lotto* in Carico e la relazione *Lotto*.

MEZZO DI TRASPORTO (Targa, Tipo)

CONTAINER (ID, Porto_Arrivo, Data_Partenza, Nave, Carico, Lotto)

Vincoli di integrità referenziale:

- Vincolo di integrità referenziale tra l'attributo *Nave* in Container e la relazione *Mezzo Di Trasporto*;
- Vincolo di integrità referenziale tra gli attributi *Lotto* e *Carico* in Container e la relazione *Carico*.

CASSA (Barcode, Peso, Tipo_Imballaggio, Data_Ispezione, Data_Spedizione, Carico, Lotto, Batch, Ordine)

Vincoli di integrità referenziale:

- tra gli attributi *Lotto* e *Carico* in Cassa e la relazione *Carico*;
- tra gli attributi *Batch* e *Ordine* in Cassa e la relazione *Batch*.

CASSA SPLITTATA (Cassa_Partizionamento, Data_Split)

Vincolo di integrità referenziale tra l'attributo *Cassa* in Cassa splittata e la relazione *Cassa*.

REFERTO (Codice_referto, %NH3, %NIC, %RS, %NO3, Data_rilascio, Cassa, Laboratorio, Via_lab, Cap_lab)

Vincoli di integrità referenziale:

- tra l'attributo *Cassa* in Referto e la relazione *Cassa*;
- tra gli attributi *Laboratorio*, *Via_lab* e *Cap_lab* in Referto e la relazione *Laboratorio*.

LABORATORIO(Nome, Via, Cap, Telefono, Città sede)

BATCH (Numero_Progressivo, Ordine, Ricetta)

Vincoli di integrità referenziale:

- tra l'attributo *Ordine* in Batch e la relazione *Ordine*;
- tra l'attributo *Ricetta* in Batch e la relazione *Ricetta*.

RICETTA(ID_Ricetta)

DETTAGLIO (Numero_Lotto, Ricetta, Casse_per_Lotto)

Vincolo di integrità referenziale tra l'attributo *Ricetta* in Dettaglio e la relazione *Ricetta*.

ORDINE (Codice, Casse_totali, Num_Batch_richiesti, Cliente)

Vincolo di integrità referenziale tra l'attributo *Cliente* in Ordine e la relazione *Cliente*.

CLIENTE (Partita_IVA, Nome, Indirizzo, Telefono)

CUSTOM CLEARANCE (ID, Ordine)

Vincolo di integrità referenziale tra l'attributo *Ordine* in Custom Clearance e la relazione *Ordine*.

Traduzione di Relazioni:

TRASFERIMENTO#2 (Numero_Progressivo, Lotto, Targa_Camion, Data_Partenza)

Vincoli di integrità referenziale:

- tra gli attributi *Numero_progressivo* e *Lotto* in Trasferimento#2 e la relazione *Carico*;
- tra l'attributo *Targa_Camion* in Trasferimento#2 e la relazione *Mezzo di Trasporto*.

SDOGANAMENTO (Lotto, Custom Clearance, Ordine, Status Precedente)

Vincoli di integrità referenziale:

- tra gli attributi *Custom Clearance* e *Ordine* in Sdoganamento e la relazione *Custom Clearance*;
- tra l'attributo *Lotto* in Sdoganamento e la relazione *Lotto*.

SPEDIZIONE (Ordine, Mezzo di trasporto, Data_Spedizione)

Vincoli di integrità referenziale:

- tra l'attributo *Ordine* in Spedizione e la relazione *Ordine*.
- tra l'attributo *Mezzo Di Trasporto* in Spedizione e la relazione *Mezzo di Trasporto*.

Normalizzazione

Per valutare la qualità della base dati si procede alla sua normalizzazione, verificando che le dipendenze funzionali includano come implicanti solo termini chiave delle relazioni (Forma Normale di Boyce e Codd).

Osservando le relazioni create in fase di traduzione si può notare come tale proprietà sia soddisfatta.

IMPLEMENTAZIONE DELLE OPERAZIONI

Creazione delle tabelle

```
CREATE TABLE public.batch
(
    numero_progressivo integer NOT NULL,
    ordine integer NOT NULL,
    ricetta integer,
    CONSTRAINT batch_pkey PRIMARY KEY (numero_progressivo,
ordine),
    CONSTRAINT batch_ordine FOREIGN KEY (ordine)
        REFERENCES public.ordine (codice) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE SET DEFAULT
        DEFERRABLE
        NOT VALID
)
```

```
CREATE TABLE public.carico
(
    numero_progressivo integer NOT NULL,
    lotto integer NOT NULL,
    data_di_controllo date,
    data_fine_quarantena date,
    data_trattamento date,
    num_casse_iniz integer,
    CONSTRAINT carico_pkey PRIMARY KEY (numero_progressivo,
lotto),
    CONSTRAINT carico_lotto FOREIGN KEY (lotto)
        REFERENCES public.lotto (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT date CHECK (data_di_controllo <
data_fine_quarantena AND data_fine_quarantena <
data_trattamento) NOT VALID,
    CONSTRAINT max_casse CHECK (num_casse_iniz <= 100) NOT
VALID )
```



```

CREATE TABLE public.cassa
(
    barcode character(15) COLLATE pg_catalog."default" NOT
NULL,
    peso integer,
    tipo_imballaggio character(4) COLLATE
pg_catalog."default",
    data_ispezione date,
    data_spedizione date,
    carico integer,
    lotto integer,
    batch integer,
    ordine integer,
    CONSTRAINT cassa_pkey PRIMARY KEY (barcode),
    CONSTRAINT cassa_batch_ordine FOREIGN KEY (batch, ordine)
        REFERENCES public.batch (numero_progressivo, ordine)
        MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE SET DEFAULT
        DEFERRABLE
        NOT VALID,
    CONSTRAINT cassa_carico FOREIGN KEY (carico, lotto)
        REFERENCES public.carico (numero_progressivo,lotto)
        MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE SET DEFAULT
        DEFERRABLE
        NOT VALID,
    CONSTRAINT date CHECK (data_ispezione < data_spedizione)
    NOT VALID,
    CONSTRAINT peso CHECK (peso <= 200) NOT VALID
)

```

```

CREATE TABLE public.cassa_split
(
    cassa character(15) COLLATE pg_catalog."default" NOT NULL,
    partizionamento integer NOT NULL,
    data_split date,
    CONSTRAINT cassa_split_pkey PRIMARY KEY (cassa,
partizionamento),
    CONSTRAINT cassa_split_cassa_fkey FOREIGN KEY (cassa)
        REFERENCES public.cassa (barcode) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

```

CREATE TABLE public.cliente

```
(
    partita_iva character varying COLLATE pg_catalog."default"
NOT NULL,
    nome character(50) COLLATE pg_catalog."default",
    indirizzo character(50) COLLATE pg_catalog."default",
    telefono character varying COLLATE pg_catalog."default",
    CONSTRAINT cliente_pkey PRIMARY KEY (partita_iva)
)
```

CREATE TABLE public.coltivatore

```
(
    partita_iva character varying(30) COLLATE
pg_catalog."default" NOT NULL,
    ragione_sociale character varying COLLATE
pg_catalog."default",
    nazionalita character varying(50) COLLATE
pg_catalog."default",
    citta_sede character varying(50) COLLATE
pg_catalog."default",
    indirizzo character varying(50) COLLATE
pg_catalog."default",
    telefono character(15) COLLATE pg_catalog."default",
    CONSTRAINT coltivatore_pkey PRIMARY KEY (partita_iva)
)
```

CREATE TABLE public.container

```
(
    id character(10) COLLATE pg_catalog."default" NOT NULL,
    porto_arrivo character varying(20) COLLATE
pg_catalog."default",
    data_partenza date,
    nave character(15) COLLATE pg_catalog."default",
    numero_progressivo_carico integer,
    lotto integer,
    CONSTRAINT container_pkey PRIMARY KEY (id),
    CONSTRAINT container_numero_progressivo_lotto_fkey FOREIGN
KEY (lotto, numero_progressivo_carico)
REFERENCES public.carico (lotto, numero_progressivo)
MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
)
```

```

CREATE TABLE public.custom_clearance
(
    id integer NOT NULL,
    ordine integer NOT NULL,
    CONSTRAINT custom_clearance_pkey PRIMARY KEY (id, ordine),
    CONSTRAINT custom_clearance_ordine_fkey FOREIGN KEY
(ordine)
        REFERENCES public.ordine (codice) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

```

```

CREATE TABLE public.deposito
(
    id_deposito integer NOT NULL,
    capacita_max integer,
    citta character varying(50) COLLATE pg_catalog."default",
    indirizzo character varying(50) COLLATE
pg_catalog."default",
    CONSTRAINT deposito_pkey PRIMARY KEY (id_deposito),
    CONSTRAINT "Max_capacity" CHECK (capacita_max <= 80000)
NOT VALID
)

```

```

CREATE TABLE public.dettaglio_ricetta
(
    id_ricetta integer NOT NULL,
    numero_lotto integer NOT NULL,
    casse_per_lotto integer,
    CONSTRAINT ricetta_pkey PRIMARY KEY (id_ricetta,
numero_lotto),
    CONSTRAINT dettaglio_ricetta FOREIGN KEY (id_ricetta)
        REFERENCES public.ricetta (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        DEFERRABLE
        NOT VALID,
    CONSTRAINT ricetta_casse_per_lotto_check CHECK
(casse_per_lotto <= 1000)
)

```

```

CREATE TABLE public.laboratorio
(
    nome character varying COLLATE pg_catalog."default" NOT
NULL,
    via character varying COLLATE pg_catalog."default" NOT
NULL,
    cap integer NOT NULL,
    citta_sede character varying COLLATE pg_catalog."default",
    telefono character varying COLLATE pg_catalog."default",
    CONSTRAINT laboratorio_pkey PRIMARY KEY (nome, via, cap)
)

```

```

CREATE TABLE public.lotto
(
    id integer NOT NULL,
    trasporto character(5) COLLATE pg_catalog."default",
    varieta character(3) COLLATE pg_catalog."default",
    forma character(15) COLLATE pg_catalog."default",
    coltivatore character varying(30) COLLATE
pg_catalog."default",
    venditore character varying(30) COLLATE
pg_catalog."default",
    status character(5) COLLATE pg_catalog."default",
    deposito integer,
    num_casse_disp integer,
    CONSTRAINT lotto_pkey PRIMARY KEY (id),
    CONSTRAINT lotto_coltivatore FOREIGN KEY (coltivatore)
REFERENCES public.coltivatore (partita_iva) MATCH
SIMPLE
ON UPDATE SET DEFAULT
ON DELETE SET DEFAULT
DEFERRABLE
NOT VALID,
    CONSTRAINT lotto_deposito FOREIGN KEY (deposito)
REFERENCES public.deposito (id_deposito) MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE SET DEFAULT
DEFERRABLE
NOT VALID,
    CONSTRAINT lotto_venditore FOREIGN KEY (venditore)
REFERENCES public.venditore (partita_iva) MATCH SIMPLE
ON UPDATE SET DEFAULT
ON DELETE SET DEFAULT
DEFERRABLE
NOT VALID,
    CONSTRAINT lotto_forma_check CHECK (forma =
'strips'::bpchar OR forma = 'loose leaves'::bpchar OR forma =
'stem'::bpchar OR forma = 'stem2'::bpchar OR forma =
'scraps1'::bpchar OR forma = 'scraps2'::bpchar OR forma =
'fines'::bpchar),
    CONSTRAINT via CHECK (trasporto = 'mare'::bpchar OR
trasporto = 'terra'::bpchar) NOT VALID )

```

```

CREATE TABLE public.mezzo_di_trasporto
(
    targa character(15) COLLATE pg_catalog."default" NOT NULL,
    tipo character(6) COLLATE pg_catalog."default",
    CONSTRAINT mezzo_di_trasporto_pkey PRIMARY KEY (targa),
    CONSTRAINT mezzo_check CHECK (tipo = 'camion'::bpchar OR
tipo = 'nave'::bpchar) NOT VALID
)

```

```

CREATE TABLE public.ordine
(
    codice integer NOT NULL,
    casse_totali integer,
    num_batch_richiesti integer,
    cliente character(30) COLLATE pg_catalog."default",
    CONSTRAINT ordine_pkey PRIMARY KEY (codice),
    CONSTRAINT ordine_cliente_fkey FOREIGN KEY (cliente)
        REFERENCES public.cliente (partita_iva) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE SET NULL
)

```

```

CREATE TABLE public.referto
(
    codice_referto character(20) COLLATE pg_catalog."default"
NOT NULL,
    nh3 double precision,
    nic double precision,
    rs double precision,
    no3 double precision,
    data_rilascio date,
    cassa character(15) COLLATE pg_catalog."default",
    laboratorio character(50) COLLATE pg_catalog."default",
    via_lab character(50) COLLATE pg_catalog."default",
    cap_lab integer,
    CONSTRAINT referto_pkey PRIMARY KEY (codice_referto),
    CONSTRAINT referto_cassa FOREIGN KEY (cassa)
        REFERENCES public.cassa (barcode) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        DEFERRABLE
        NOT VALID,
    CONSTRAINT referto_laboratorio_via_lab_cap_lab_fkey
FOREIGN KEY (via_lab, laboratorio, cap_lab)
        REFERENCES public.laboratorio (via, nome, cap) MATCH
SIMPLE
        ON UPDATE CASCADE
        ON DELETE SET NULL
)

```

CREATE TABLE public.ricetta

```
(
    id integer NOT NULL,
    CONSTRAINT "Ricetta_pkey" PRIMARY KEY (id)
)
```

CREATE TABLE public.sdoganamento

```
(
    custom_clearance integer NOT NULL,
    ordine integer NOT NULL,
    lotto integer NOT NULL,
    status_precedente character(4) COLLATE
pg_catalog."default",
    CONSTRAINT sdoganamento_pkey PRIMARY KEY
(custom_clearance, ordine, lotto),
    CONSTRAINT sd_cc FOREIGN KEY (custom_clearance, ordine)
REFERENCES public.custom_clearance (id, ordine) MATCH
SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    DEFERRABLE
    NOT VALID,
    CONSTRAINT sd_lotto FOREIGN KEY (lotto)
REFERENCES public.lotto (id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    DEFERRABLE
    NOT VALID
)
```

CREATE TABLE public.spedizione

```
(
    ordine integer NOT NULL,
    mezzo_di_trasporto character(10) COLLATE
pg_catalog."default" NOT NULL,
    data_spedizione date,
    CONSTRAINT spedizione_pkey PRIMARY KEY (ordine,
mezzo_di_trasporto),
    CONSTRAINT sped_ordine FOREIGN KEY (ordine)
REFERENCES public.ordine (codice) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    DEFERRABLE
    NOT VALID,
    CONSTRAINT sped_trasp FOREIGN KEY (mezzo_di_trasporto)
REFERENCES public.mezzo_di_trasporto (targa) MATCH
SIMPLE
    ON UPDATE CASCADE
    ON DELETE SET DEFAULT
    DEFERRABLE
    NOT VALID
)
```

```

CREATE TABLE public.trasferimento2
(
    numero_progressivo integer NOT NULL,
    lotto integer NOT NULL,
    targa_camion character(15) COLLATE pg_catalog."default"
NOT NULL,
    data_di_partenza date,
    CONSTRAINT trasferimento2_pkey PRIMARY KEY
(numero_progressivo, lotto, targa_camion),
    CONSTRAINT trasf_carico FOREIGN KEY (lotto,
numero_progressivo)
REFERENCES public.carico (lotto, numero_progressivo)
MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE SET DEFAULT
    DEFERRABLE
    NOT VALID,
    CONSTRAINT trasf_mezzo FOREIGN KEY (targa_camion)
REFERENCES public.mezzo_di_trasporto (targa) MATCH
SIMPLE
    ON UPDATE CASCADE
    ON DELETE SET DEFAULT
    DEFERRABLE
    NOT VALID
)

```

```

CREATE TABLE public.venditore
(
    partita_iva character varying(30) COLLATE
pg_catalog."default" NOT NULL,
    ragione_sociale character varying(50) COLLATE
pg_catalog."default",
    nazionalita character varying(50) COLLATE
pg_catalog."default",
    citta_sede character varying(50) COLLATE
pg_catalog."default",
    indirizzo character varying(50) COLLATE
pg_catalog."default",
    telefono character(15) COLLATE pg_catalog."default",
    CONSTRAINT venditore_pkey PRIMARY KEY (partita_iva)
)

```

Creazione delle query

OP1: Aggiunta di un lotto, definendo il venditore, il coltivatore e il deposito in cui viene collocato (25 volte al mese).

L'operazione permette di aggiungere un lotto alla relazione LOTTO, inserendo tutti i dati richiesti.

```
CREATE OR REPLACE FUNCTION public.aggiungi_lotto(  
    id integer,  
    trasp character,  
    varieta character,  
    forma character,  
    coltivatore character varying,  
    venditore character varying,  
    status character,  
    dep integer,  
    num_c_disp integer)  
RETURNS void  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
Insert Into "lotto"  
values(  
    id,  
    trasp,  
    varieta,  
    forma,  
    coltivatore,  
    venditore,  
    status,  
    dep,  
    num_c_disp  
);  
end;  
$BODY$;
```


OP2: Aggiunta di una cassa (12500 volte al mese, circa 500 casse a lotto).

Per questa operazione vengono proposte due possibili implementazioni in base ai dati disponibili della cassa da aggiungere. Nel caso in cui si dispongono solo i dati iniziali, prima dell'utilizzo di tale cassa, la funzione è la seguente:

```
CREATE OR REPLACE FUNCTION
public.aggiungi_cassa_con_tdati_iniziali(
    barcode character,
    peso integer,
    tip_imb character,
    data_isp date,
    data_sped date,
    carico integer,
    lotto integer)
RETURNS void
LANGUAGE 'plpgsql'

COST 100
VOLATILE

AS $BODY$
begin
Insert Into "cassa"
values(
    barcode,
    peso,
    tip_imb,
    data_isp,
    data_sped,
    carico,
    lotto
);
end;
$BODY$;
```

Mentre nel caso in cui si hanno a disposizione i dati relativi anche al fine della cassa, come batch e ordine, si utilizza la seguente funzione:

```
CREATE OR REPLACE FUNCTION
public.aggiungi_cassa_con_tutti_i_dati(
    barcode character,
    peso integer,
    tip_imb character,
    data_isp date,
    data_sped date,
    carico integer,
    lotto integer,
```

```
        batch integer,
        ordine integer)
RETURNS void
LANGUAGE 'plpgsql'

COST 100
VOLATILE

AS $BODY$
begin
Insert Into "cassa"
values(
        barcode,
        peso,
        tip_imb,
        data_isp,
        data_sped,
        carico,
        lotto,
        batch,
        ordine
);
end;
$BODY$;
```

OP3: Aggiunta di un carico definendo il mezzo di arrivo (125 volte al mese, circa 5 carichi per lotto):

Si propongono tre versioni: la prima permette di inserire solo i dati relativi al carico, le altre invece permettono di inserire anche il mezzo di trasporto con cui è stato consegnato il carico.

```
CREATE OR REPLACE FUNCTION public.aggiungi_carico(  
    num_prog integer,  
    lotto integer,  
    data_con date,  
    data_quara date,  
    data_tra date,  
    num_cass_iniz integer)  
RETURNS void  
LANGUAGE 'plpgsql'  
  
COST 100  
VOLATILE
```

```
AS $BODY$  
begin  
Insert Into "carico"  
values(  
    num_prog ,  
    lotto ,  
    data_con ,  
    data_quara ,  
    data_tra ,  
    num_cass_iniz  
);  
end;  
$BODY$;
```

```
CREATE OR REPLACE FUNCTION  
public.aggiungi_carico_arrivato_via_mare(  
    id_container character,  
    porto_arrivo character varying,  
    data_partenza date,  
    traga_mezzo character,  
    np_carico integer,  
    lot integer,  
    data_di_controllo date,  
    data_fq date,
```

```

        data_t date,
        num_casse_iniz integer)
RETURNS void
LANGUAGE 'plpgsql'

COST 100
VOLATILE

AS $BODY$
declare trasporto character(5)= (Select "lotto".trasporto from
"lotto" where lot = "lotto".id);

begin
perform("aggiungi_carico"(
        np_carico,
        lot,
        data_di_controllo,
        data_fq,
        data_t,
        num_casse_iniz
));

Insert Into "container" values (
        id_container,
        porto_arrivo,
        data_partenza,
        traga_mezzo,
        np_carico,
        lot
);

if (not exists (select "mezzo_di_trasporto".targa
                    from "mezzo_di_trasporto"
                    where "mezzo_di_trasporto".targa =
traga_mezzo)) then
        perform("aggiungi_nave"(traga_mezzo));
end if;

end;
$BODY$;

```

```

CREATE OR REPLACE FUNCTION
public.aggiungi_carico_arrivato_via_terra(
    data_partenza date,
    traga_mezzo character,
    np_carico integer,
    lot integer,
    data_di_controllo date,
    data_fq date,
    data_t date,
    num_casse_iniz integer)
RETURNS void
LANGUAGE 'plpgsql'

COST 100
VOLATILE

AS $BODY$
begin
perform("aggiungi_carico"(
    np_carico,
    lot,
    data_di_controllo,
    data_fq,
    data_t,
    num_casse_iniz
));

Insert Into "trasferimento2" values (
    np_carico,
    lot,
    traga_mezzo,
    data_partenza
);

if (not exists (select "mezzo_di_trasporto".targa
                    from "mezzo_di_trasporto"
                    where "mezzo_di_trasporto".targa =
traga_mezzo)) then
    perform("aggiungi_nave"(traga_mezzo));
end if;

end;
$BODY$;

```

OP4: Registrazione di un nuovo venditore, (3 volte all'anno):

```
CREATE OR REPLACE FUNCTION public.aggiungi_venditore(  
    p_iva character varying,  
    r_soc character varying,  
    naz character varying,  
    citt_sed character varying,  
    ind character varying,  
    tel character varying)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
Insert Into "venditore"  
values(  
    p_iva,  
    r_soc,  
    naz,  
    citt_sed,  
    ind,  
    tel  
);  
end;  
$BODY$;
```

OP5: Registrazione di un nuovo coltivatore, (8 volte all'anno):

```
CREATE OR REPLACE FUNCTION public.aggiungi_coltivatore(  
    p_iva character varying,  
    r_soc character varying,  
    naz character varying,  
    citt_sed character varying,  
    ind character varying,  
    tel character varying)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
    Insert Into "Coltivatore"  
    values(  
        p_iva,  
        r_soc,  
        naz,  
        citt_sed,  
        ind,  
        tel  
    );  
end;  
$BODY$;
```

OP6: Inserimento di una nave (7 volte al mese):

```
CREATE OR REPLACE FUNCTION public.aggiungi_nave(  
    targa character)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
Insert Into "nave"  
values(  
    targa,  
    "nave"  
);  
end;  
$BODY$;
```

OP7: Aggiunta di un container, indicando la nave con cui è stato trasferito (80 volte al mese):

```
CREATE OR REPLACE FUNCTION public.aggiungi_container(  
    id character,  
    porto_arrivo character varying,  
    data_partenza date,  
    nave character,  
    numero_progressivo integer,  
    lotto integer)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
Insert Into "container"  
values(  
    id,  
    porto_arrivo,  
    data_partenza,  
    nave,  
    numero_progressivo,  
    lotto  
);  
end;  
$BODY$;
```


OP8: Inserimento di un camion (5 volte al mese):

```
CREATE OR REPLACE FUNCTION public.aggiungi_camion(  
    targa character)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
Insert Into "mezzo_di_trasporto"  
values(  
    targa,  
    "camion"  
);  
end;  
$BODY$;
```

OP9: Registrazione di un nuovo cliente (3 volte all'anno):

```
CREATE OR REPLACE FUNCTION public.aggiungi_cliente(  
    p_iva character varying,  
    nome character,  
    ind character,  
    tel character varying)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
Insert Into "cliente"  
values(  
    p_iva,  
    nome,  
    ind,  
    tel  
);  
end;  
$BODY$;
```

OP10: Registrazione di un nuovo ordine e eventuale custom clearance (20 volte al mese):

```
CREATE OR REPLACE FUNCTION public.aggiungi_ordine(  
    codice integer,  
    casse_totali integer,  
    num_batch integer,  
    cliente character,  
    id_custom_c integer)  
RETURNS void  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
insert into "ordine" values(  
    codice,  
    casse_totali ,  
    num_batch ,  
    cliente);  
  
if (id_custom_c <> null) then perform  
aggiungi_custom_clearance(id_custom_c, codice);  
end if;  
  
end;  
$BODY$;
```

OP11: Registrazione di una spedizione di un ordine (20 volte al mese):

```
CREATE OR REPLACE FUNCTION public.aggiungi_spedizione(  
    codice_ordine integer,  
    targa_mezzo character,  
    data_sped date)  
RETURNS void  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
  
insert into spedizione values(  
    codice_ordine ,  
    targa_mezzo,  
    data_sped  
);  
  
end;  
$BODY$;
```

OP11.1: Spedisci casse di un batch.

Per aggiornare le casse si utilizza tale funzione, la quale permette di inserire l'ordine con cui vengono spedite le casse, il relativo batch, che viene inserito anche all'interno della tabella batch, e l'array delle casse che devono essere spedite, specificando la data di spedizione di una cassa e aggiornando il numero di casse disponibili sul lotto.

```
CREATE OR REPLACE FUNCTION  
public.spedisci_casse_di_un_batch_v_3(  
    elenco_casse character[],  
    id_batch integer,  
    codice_ordine integer,  
    ricetta integer)  
RETURNS void  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
declare data_sped date = (select data_spedizione from  
spedizione where "spedizione".ordine = codice_ordine);  
  
declare lotto_cassa int;
```

```

begin

if (not exists (select * from batch where numero_progressivo =
id_batch and ordine = codice_ordine)) then Insert into batch
values (id_batch,codice_ordine,ricetta ); end if;

for i in 1..cardinality(elenco_casse) loop
    lotto_cassa =(select "cassa".lotto from cassa where
"cassa".barcode = elenco_casse[i]);

    update "cassa"
    set batch = id_batch, ordine = codice_ordine,
data_spedizione = data_sped
    where "cassa".barcode = elenco_casse[i];

    update "lotto"
    set num_casse_disp = (select num_casse_disp
                        from "lotto", "cassa"
                        where "cassa".lotto =
"lotto".id and "cassa".barcode = elenco_casse[i]) -1
    where lotto.id = lotto_cassa;
end loop;
end;
$BODY$;

```

OP12: Registrazione di un nuova ricetta (10 volte al mese):

```
CREATE OR REPLACE FUNCTION public.aggiungi_ricetta(  
    id_ricetta integer,  
    numero_lotto integer[],  
    casse_per_lotto integer[])  
RETURNS void  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
  
insert into ricetta values(  
    id_ricetta  
);  
  
for i in 1..cardinality(numero_lotto) loop  
    insert into "dettaglio_ricetta" values(  
        id_ricetta ,  
        numero_lotto [i] ,  
        casse_per_lotto [i]  
    );  
end loop;  
  
end;  
$BODY$;
```

OP13: Registrazione di un custom clearance (8 volte al mese):

```
CREATE OR REPLACE FUNCTION public.aggiungi_custom_clearance(  
    id integer,  
    numero_ordine integer)  
RETURNS void  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
insert into "custom_clearance" values(id, numero_ordine);  
end;  
$BODY$;
```

OP14: Modifica del tipo di una cassa (400 volte al mese):

```
CREATE OR REPLACE FUNCTION public.split_cassa(  
    barcode_ca character,  
    data_split date)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
  
    for i in 1..2 loop  
        insert into cassa_split values(barcode_ca , i,  
data_split);  
    end loop;  
  
    update "cassa" set tipo_imballaggio = 'C96', peso = 100;  
end;  
$BODY$;
```

OP15: Modifica di status di un lotto (10 volte al mese):

```
CREATE OR REPLACE FUNCTION public.sdogana_lotti(  
    codice_c_c integer,  
    codice_ord integer,  
    id_lotto integer[])  
RETURNS void  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
declare status_prec character(5);  
  
begin  
  
for i in 1..cardinality(id_lotto) loop  
    status_prec = (select status  
                                from "lotto"  
                                where  
"lotto".id = id_lotto[i]);  
  
    insert into "sdoganamento" values (  
        codice_c_c ,  
        codice_ord,  
        id_lotto [i],  
        status_prec  
    );  
  
    update "lotto" set status = 'FC' where id_lotto [i] =  
"lotto".id;  
  
end loop;  
end;  
$BODY$;
```

OP16: Cancellazione di un cliente (1 volta all'anno):

```
CREATE OR REPLACE FUNCTION public.cancellazione_cliente(  
    piva character varying)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
BEGIN  
DELETE FROM "cliente"  
WHERE piva=partita_iva;  
END  
$BODY$;
```

OP17: Cancellazione di un venditore (1 volta all'anno):

```
CREATE OR REPLACE FUNCTION public.cancellazione_venditore(  
    piva character varying)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
BEGIN  
DELETE FROM "venditore"  
WHERE piva=partita_iva;  
END  
$BODY$;
```


OP18: Cancellazione di un coltivatore (3 volte all'anno):

```
CREATE OR REPLACE FUNCTION public.cancellazione_coltivatore(  
    piva character varying)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
BEGIN  
DELETE FROM "coltivatore"  
WHERE piva=partita_iva;  
END  
$BODY$;
```

OP19: Cancellazione di una ricetta (5 al mese):

```
CREATE OR REPLACE FUNCTION public.cancellazione_ricetta(  
    ricetta integer)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
begin  
delete from "ricetta" where "ricetta".id = ricetta;  
end;  
$BODY$;
```

OP20: Stampare la quantità di casse, raggruppate per lotti, presenti nei magazzini (10 volte al mese):

```
CREATE OR REPLACE FUNCTION public.stampa_giacenza_magazzini(  
    )  
    RETURNS TABLE(id_deposito integer, lotto integer,  
num_casse_lotto integer)  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
begin  
return query(  
    select "deposito".id_deposito, "lotto".id,  
"lotto".num_casse_disp  
    from lotto, deposito  
    where "lotto".deposito = "deposito".id_deposito  
);  
end;  
$BODY$;
```

OP21: Stampare la quantità in kg e in numero di casse di tabacco arrivato in magazzino in un intervallo di tempo raggruppato per lotto (4 volte al mese):

```
CREATE OR REPLACE FUNCTION public.stampa_casse_arrivate(  
    data_inizio date,  
    data_fine date)  
RETURNS TABLE(lotto integer, num_casse bigint, kg bigint)  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
begin  
return query(  
    select "carico".lotto, sum("carico".num_casse_iniz) as  
num_casse_iniz, sum("carico".num_casse_iniz)*200 as kg_totali  
    from carico  
    where "carico".data_di_controllo > data_inizio and  
"carico".data_di_controllo < data_fine  
    group by "carico".lotto  
    order by "carico".lotto  
  
);  
end;  
$BODY$;
```

OP22: Stampare il numero di casse di tabacco spedito, in un determinato intervallo, raggruppato per lotto (4 volte al mese):

```
CREATE OR REPLACE FUNCTION public.stampa_num_casse_inviare(  
    data_inizio date,  
    data_fine date)  
RETURNS TABLE(lotto integer, num_casse bigint)  
LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
begin  
  
return query(  
    select "cassa".lotto, count("cassa".barcode)  
    from cassa  
    where data_spedizione > data_inizio and data_spedizione <  
data_fine  
    group by "cassa".lotto  
    order by "cassa".lotto  
);  
end;  
$BODY$;
```

OP23: Stampare la quantità in kg e in numero di casse di tabacco, raggruppato per lotto, che ha cambiato di status, stampando anche lo status di partenza(3 volte al mese):

```
CREATE OR REPLACE FUNCTION public.stampa_casse_sdoganate(  
    )  
    RETURNS TABLE(lotto integer, num_casse bigint,  
status_precedente character)  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
begin  
  
return query(  
select "cassa".lotto, count("cassa".barcode),  
"sdoganamento".status_precedente  
from cassa, sdoganamento  
where "cassa".lotto = "sdoganamento".lotto  
group by "cassa".lotto, "sdoganamento".status_precedente  
order by "cassa".lotto, "sdoganamento".status_precedente  
);  
  
end;  
$BODY$;
```

OP24: Stampare la quantità in kg e in numero di casse di tabacco, raggruppato per lotto, andato **in trattamento** in un intervallo di tempo (1 volta al mese):

```
CREATE OR REPLACE FUNCTION public.stampa_num_casse_trattate(  
    data_inizio date,  
    data_fine date)  
    RETURNS TABLE(lotto integer, num_casse bigint, kg_totali  
    bigint)  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
begin  
  
return query(  
    select "carico".lotto, sum("carico".num_casse_iniz),  
sum("carico".num_casse_iniz)*200  
    from carico  
    where data_trattamento > data_inizio and data_trattamento  
< data_fine  
    group by "carico".lotto  
    order by "carico".lotto  
);  
end;  
$BODY$;
```

OP25: Stampare la quantità in kg e in numero di casse della merce, raggruppata per lotto, **ispezionate** in un intervallo di tempo (1 volta al mese):

```
CREATE OR REPLACE FUNCTION public.stampa_numero_split(  
    split integer)  
    RETURNS TABLE(numero_split integer)  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
BEGIN  
return query  
(select count(*)  
    from cassa_split  
    where extract(month from data_split) = yy);  
end  
$BODY$;
```

OP26: Stampare il numero di analisi fatte in un anno con cadenza mensile (1 volta al mese):

```
CREATE OR REPLACE FUNCTION public.stampa_numero_analisi(  
    anno integer)  
    RETURNS TABLE(mese double precision, numero_referti  
bigint)  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
BEGIN  
return query  
(select extract (month from data_rilascio), count(*)  
from referto  
where extract(year from data_rilascio) = anno  
group by extract (month from data_rilascio));  
end  
$BODY$;
```

OP27: Stampare il numero di analisi fatte in un anno con cadenza mensile (1 volta al mese):

```
CREATE OR REPLACE FUNCTION public.stampa_numero_split(  
    anno integer)  
    RETURNS TABLE(mese double precision, numero_referti  
bigint)  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
BEGIN  
return query  
(select extract (month from data_split), count(*)  
from "cassa_split"  
where extract(year from data_split) = anno  
group by extract (month from data_split));  
end  
;  
$BODY$;
```


OP28: (100 al mese) **Stampare un report** dell'intero iter della cassa nel magazzino comprensivo di:

```
CREATE OR REPLACE FUNCTION public.stampa_iter_cassa(  
    cassa_cercata character)  
    RETURNS TABLE(barcode character, partizionamento integer,  
peso integer, tipo_imballaggio character, carico integer,  
data_di_controllo date, data_fine_quarantena date,  
data_trattamento date, data_ispezione date, data_referto date,  
codice_referto character, nh3 double precision, nic double  
precision, rs double precision, no3 double precision, lotto  
integer, status character, origine character varying,  
provenienza character varying, ordine integer, batch integer,  
mezzo_di_trasporto character, data_spedizione date)  
    LANGUAGE 'plpgsql'  
  
    COST 100  
    VOLATILE  
    ROWS 1000  
  
AS $BODY$  
begin  
return query (  
    select "cassa".barcode,  
        "cassa_split".partizionamento,  
        "cassa".peso,  
        "cassa".tipo_imballaggio,  
        "cassa".carico,  
        "carico".data_di_controllo,  
        "carico".data_fine_quarantena,  
        "carico".data_trattamento,  
        "cassa".data_ispezione,  
        "referto".data_rilascio,  
        "referto".codice_referto,  
        "referto".nh3,  
        "referto".nic,  
        "referto".rs,  
        "referto".no3,  
        "cassa".lotto,  
        "lotto".status,  
        "coltivatore".nazionalita as origine,  
        "venditore".nazionalita as provenienza,  
        "cassa".ordine,  
        "cassa".batch,  
        "spedizione".mezzo_di_trasporto,  
        "spedizione".data_spedizione  
  
        from ("cassa" join "carico" on "cassa".carico =  
"carico".numero_progressivo and "cassa".lotto =
```

```

"carico".lotto) left outer join
            "cassa_split" on "cassa".barcode =
"cassa_split".cassa left outer join
            "referto" on "cassa".barcode = "referto".cassa
join
            "lotto" on "cassa".lotto = "lotto".id left
outer join
            "coltivatore" on "lotto".coltivatore =
"coltivatore".partita_iva left outer join
            "venditore" on "lotto".venditore =
"venditore".partita_iva left outer join
            "spedizione" on "cassa".ordine =
"spedizione".ordine

        where "cassa".barcode = cassa_cercata

    );

end;
$BODY$;

```

ESECUZIONE DELLE OPERAZIONI PRINCIPALI

OP1: Aggiunta di un lotto, definendo il venditore, il coltivatore e il deposito in cui viene collocato (25 volte al mese);

Si analizza il modo con cui agisce la seguente operazione supponendo di partire dall'istanza sottostante per la relazione LOTTO:

	id [PK] integer	trasporto character (5)	varietà character (3)	forma character (15)	coltivatore character varying (30)	venditore character varying (30)	status character (5)	deposito integer	num_casse_disp integer
204	460	mare	DAC	strips	gd154478512	zs082725179	FC	1	826
205	463	mare	DAC	strips	dr884294131	jz560990273	FC	2	426
206	466	terra	DAC	strips	js315239785	fu795111503	EXEU	2	45
207	469	mare	LAC	stem	st199509486	yg603950868	FC	2	299
208	476	mare	LAC	stem	gj051748372	kb478310551	FC	1	910
209	480	mare	FCV	loose leaves	lz864591953	bb755896434	EXEU	2	369
210	483	terra	LAC	scraps2	cv274811955	bb755896434	EXEU	2	786
211	484	mare	LAC	strips	cf127702793	rh052274090	EXEU	2	322
212	488	mare	FCV	stem	rm732336300	bt527846326	EXEU	3	753
213	489	mare	LAC	stem	li275108066	qk732864515	EXEU	1	32
214	490	mare	LAC	loose leaves	fb685420703	ah562776935	FC	3	573
215	491	mare	LAC	stem	pv200578104	rh052274090	EXEU	1	692
216	492	terra	FCV	scraps2	rx994150965	yg603950868	FC	2	18
217	494	terra	FCV	scraps2	cv274811955	sa881177869	EXEU	2	471
218	495	terra	FCV	scraps2	of483314399	zs082725179	EXEU	3	372
219	496	terra	LAC	finest	ay678175718	rh052274090	FC	1	433
220	499	terra	DAC	finest	iz200984380	jd070375642	EXEU	2	748
221	500	mare	LAC	stem	gd154478512	fu795111503	FC	1	400

Si vogliono inserire i dati relativi a un nuovo Lotto. Tale operazione è eseguibile invocando la funzione sottostante, definita nella sezione “Creazione delle query”:

`SELECT (aggiungi_lotto('536', 'terra', 'FCV', 'strips', 'im726749209', 'qk732864515', 'FC', '1', '827'));`

La relazione LOTTO viene aggiornata con i dati del lotto inserito, inserendo la riga evidenziata:

	id [PK] integer	trasporto character (5)	varietà character (3)	forma character (15)	coltivatore character varying (30)	venditore character varying (30)	status character (5)	deposito integer	num_casse_disp integer
205	463	mare	DAC	strips	dr884294131	jz560990273	FC	2	426
206	466	terra	DAC	strips	js315239785	fu795111503	EXEU	2	45
207	469	mare	LAC	stem	st199509486	yg603950868	FC	2	299
208	476	mare	LAC	stem	gj051748372	kb478310551	FC	1	910
209	480	mare	FCV	loose leaves	lz864591953	bb755896434	EXEU	2	369
210	483	terra	LAC	scraps2	cv274811955	bb755896434	EXEU	2	786
211	484	mare	LAC	strips	cf127702793	rh052274090	EXEU	2	322
212	488	mare	FCV	stem	rm732336300	bt527846326	EXEU	3	753
213	489	mare	LAC	stem	li275108066	qk732864515	EXEU	1	32
214	490	mare	LAC	loose leaves	fb685420703	ah562776935	FC	3	573
215	491	mare	LAC	stem	pv200578104	rh052274090	EXEU	1	692
216	492	terra	FCV	scraps2	rx994150965	yg603950868	FC	2	18
217	494	terra	FCV	scraps2	cv274811955	sa881177869	EXEU	2	471
218	495	terra	FCV	scraps2	of483314399	zs082725179	EXEU	3	372
219	496	terra	LAC	finest	ay678175718	rh052274090	FC	1	433
220	499	terra	DAC	finest	iz200984380	jd070375642	EXEU	2	748
221	500	mare	LAC	stem	gd154478512	fu795111503	FC	1	400
222	536	terra	FCV	strips	im726749209	qk732864515	FC	1	827

In questo caso, sia il venditore che il coltivatore sono già presenti nelle relative relazioni, pertanto non si devono modificare altre tabelle.

OP2: Aggiunta di una cassa (12500 volte al mese, circa 500 casse a lotto);

Per questa operazione vengono proposte due possibili implementazioni in base ai dati disponibili della cassa da aggiungere. Supponendo che lo schema di partenza della relazione CASSA è la seguente:

	barcode [PK] character (15)	peso integer	tipo_imballaggio character (4)	data_ispezione date	data_spedizione date	carico integer	lotto integer	batch integer	ordine integer
521	JYWGH6G6E17J	200	C48	[null]	[null]	1	361	1	288
522	JYZ8HVVW875JI	200	C48	[null]	[null]	3	124	0	255
523	K02SCT3DXXP8	200	C48	[null]	2020-04-24	4	296	0	635
524	K040FLUF83TH	200	C48	[null]	2020-06-01	2	197	0	387
525	K0HSHGR7DGDF	200	C48	2022-12-27	[null]	3	120	0	868
526	K10T3VMY38BD	200	C48	[null]	[null]	1	439	0	500
527	K39NKHS5EJS0	200	C48	[null]	2020-12-29	7	238	0	576
528	K3XRCPPSL9TB	200	C48	[null]	[null]	1	291	0	868
529	K3YCHSWWFNJW	200	C48	[null]	[null]	1	397	0	618
530	K3Z5Y34JW327	200	C48	[null]	[null]	3	436	1	631
531	K53Q5ZNQ8MGX	200	C48	[null]	2020-02-01	8	280	0	749
532	KAMPAFOTTQTZ	200	C48	[null]	2020-04-25	3	432	1	778
533	KE9EV5FBF406	200	C48	2021-05-12	2020-11-22	0	244	0	425
534	KH0WFMFEFMHZ	200	C48	[null]	2020-10-08	2	291	2	953
535	KI517ID8KVIA	200	C48	[null]	[null]	4	134	0	513
536	KKWYNB7FNFZP	200	C48	[null]	[null]	0	184	2	953
537	KL0M7LI4ZKZ5	200	C48	2021-05-20	2020-01-27	0	204	0	214
538	KLW14IO4S70U	200	C48	2022-05-30	2020-07-05	1	179	0	298

Nel caso in cui si dispongono solo i dati iniziali, prima dell'utilizzo di tale cassa, si utilizza la prima versione della funzione:

SELECT (aggiungi_cassa_con_tdati_iniziali('KBTC0OOCDXMT','200', 'C48', '16/04/2020',
'2', '24'));

	barcode [PK] character (15)	peso integer	tipo_imballaggio character (4)	data_ispezione date	data_spedizione date	carico integer	lotto integer	batch integer	ordine integer
521	JYWGH6G6E17J	200	C48	[null]	[null]	1	361	1	288
522	JYZ8HVVW875JI	200	C48	[null]	[null]	3	124	0	255
523	K02SCT3DXXP8	200	C48	[null]	2020-04-24	4	296	0	635
524	K040FLUF83TH	200	C48	[null]	2020-06-01	2	197	0	387
525	K0HSHGR7DGDF	200	C48	2022-12-27	[null]	3	120	0	868
526	K10T3VMY38BD	200	C48	[null]	[null]	1	439	0	500
527	K39NKHS5EJS0	200	C48	[null]	2020-12-29	7	238	0	576
528	K3XRCPPSL9TB	200	C48	[null]	[null]	1	291	0	868
529	K3YCHSWWFNJW	200	C48	[null]	[null]	1	397	0	618
530	K3Z5Y34JW327	200	C48	[null]	[null]	3	436	1	631
531	K53Q5ZNQ8MGX	200	C48	[null]	2020-02-01	8	280	0	749
532	KAMPAFOTTQTZ	200	C48	[null]	2020-04-25	3	432	1	778
533	KBTC0OOCDXMT	200	C48	2020-04-16	[null]	2	24	[null]	[null]
534	KE9EV5FBF406	200	C48	2021-05-12	2020-11-22	0	244	0	425
535	KH0WFMFEFMHZ	200	C48	[null]	2020-10-08	2	291	2	953
536	KI517ID8KVIA	200	C48	[null]	[null]	4	134	0	513
537	KKWYNB7FNFZP	200	C48	[null]	[null]	0	184	2	953
538	KL0M7LI4ZKZ5	200	C48	2021-05-20	2020-01-27	0	204	0	214
539	KLW14IO4S70U	200	C48	2022-05-30	2020-07-05	1	179	0	298

Nel caso in cui si dispongono anche i dati relativi al batch e all'ordine per i quali la cassa viene utilizzata, la funzione che si adopera è la seconda versione:

SELECT (aggiungi_cassa_con_tutti_i_dati('KAULOLZKPB5G','200', 'C48', '19/11/2020',
'17/12/2020', '0', '24', '0', '722'));

	barcode [PK] character (15)	peso integer	tipo_imballaggio character (4)	data_ispezione date	data_spedizione date	carico integer	lotto integer	batch integer	ordine integer
527	K39NKHSEJS0	200	C48	[null]	2020-12-29	7	238	0	576
528	K3XRCPPSL9TB	200	C48	[null]	[null]	1	291	0	868
529	K3YCHSWWFNJV	200	C48	[null]	[null]	1	397	0	618
530	K3Z5Y34JW327	200	C48	[null]	[null]	3	436	1	631
531	K53Q5ZNQ8MGX	200	C48	[null]	2020-02-01	8	280	0	749
532	KAMPAFOTTQTZ	200	C48	[null]	2020-04-25	3	432	1	778
533	KAULOLZKPB5G	200	C48	2020-11-19	2020-12-17	0	24	0	722
534	KE9EV5FBF4O6	200	C48	2021-05-12	2020-11-22	0	244	0	425
535	KH0WFMFEFMHZ	200	C48	[null]	2020-10-08	2	291	2	953
536	KI517ID8KVIA	200	C48	[null]	[null]	4	134	0	513
537	KKWYNB7FNFZP	200	C48	[null]	[null]	0	184	2	953
538	KL0M7LI4ZKZ5	200	C48	2021-05-20	2020-01-27	0	204	0	214
539	KLW14IO4S7OU	200	C48	2022-05-30	2020-07-05	1	179	0	298
540	KNEA289Y21NC	200	C48	[null]	2020-05-04	5	189	1	507
541	KNRW7FF8GSLB	200	C48	[null]	2020-04-02	0	450	0	321
542	K06CGDL76YG8	200	C48	2020-10-29	2020-06-01	3	434	1	953
543	KON8CI4T81OR	200	C48	[null]	2020-05-04	4	26	0	421
544	KPDP6IGSN7NV	200	C48	[null]	2020-03-15	1	391	1	818
545	KTBD8YJ8QC08	200	C48	[null]	[null]	0	76	0	126

OP4: Registrazione di un nuovo venditore, (3 volte all'anno);

Supponendo di partire dalla tabella “venditore” nello stato mostrato di seguito:

	partita_iva [PK] character varying (30)	ragione_sociale character varying (50)	nazionalita character varying (50)	citta_sede character varying (50)	indirizzo character varying (50)	telefono character (15)
5	cm393713304	Thai Fund, Inc. (The)	France	La Plaine-Saint-Denis	701 Pleasure Drive	496-534-2847
6	fq921097786	Tetra Tech, Inc.	France	Vitry-sur-Seine	607 Texas Road	479-930-6397
7	fu795111503	Universal Stainless & Alloy P...	France	Orléans	150 Walton Trail	551-933-1124
8	go649624700	OHA Investment Corporation	Poland	Kraków	7447 High Crossing Lane	871-746-6142
9	jd070375642	Waters Corporation	United States	Pasadena	6452 Shoshone Circle	626-346-9597
10	jw382177527	Dupont Fabros Technology, I...	France	Alençon	883 Jackson Circle	791-325-7109
11	jz560990273	Atossa Genetics Inc.	Brazil	Maragogipe	78339 Loftsgordon Court	107-166-6625
12	kb478310551	Northern Trust Corporation	France	Lattes	45 2nd Terrace	852-420-8322
13	partita_iva	ragione_sociale	nazionalita	citta_sede	indirizzo	telefono
14	qk732864515	Superior Uniform Group, Inc.	Poland	Cedry Wielkie	9987 Carberry Terrace	632-595-2501
15	rh052274090	MFS Multimarket Income Tr...	United States	Harrisburg	44885 Artisan Circle	717-528-7502
16	sa881177869	Lam Research Corporation	Brazil	Iraty	0 Westerfield Court	491-894-1108
17	uw816264005	Baltimore Gas & Electric Co...	Poland	Bielawy	6 Esch Way	936-479-9403
18	xb235487850	Denbury Resources Inc.	Brazil	Arcos	42878 Hanson Drive	732-242-0269
19	yc617519055	Koppers Holdings Inc.	France	Roanne	46650 Morrow Point	439-266-6133
20	yg603950868	Blackrock Municipal 2018 T...	Brazil	Baturité	89403 Grasskamp Lane	577-537-2232
21	zs082725179	iPath US Treasury Flatteners ...	United States	Columbus	0064 Glacier Hill Way	614-503-2776

Si vogliono inserire i dati relativi al venditore con partita_iva : ho936481560. Tale operazione è eseguibile invocando la funzione definita nella sezione “creazione delle query”:

```
SELECT (aggiungi_venditore('ho936481560','fumatul_dauneaza','Romania',
                           'Bucuresti','piata libertati','403-589-6489'));
```

La relazione VENDITORE viene aggiornata con i dati del venditore inserito, inserendo la riga evidenziata:

	partita_iva [PK] character varying (30)	ragione_sociale character varying (50)	nazionalita character varying (50)	citta_sede character varying (50)	indirizzo character varying (50)	telefono character (15)
6	fq921097786	Tetra Tech, Inc.	France	Vitry-sur-Seine	607 Texas Road	479-930-6397
7	fu795111503	Universal Stainless & Alloy P...	France	Orléans	150 Walton Trail	551-933-1124
8	go649624700	OHA Investment Corporation	Poland	Kraków	7447 High Crossing Lane	871-746-6142
9	ho936481560	fumatul_dauneaza	Romania	Bucuresti	piata libertati	403-589-6489
10	jd070375642	Waters Corporation	United States	Pasadena	6452 Shoshone Circle	626-346-9597
11	jw382177527	Dupont Fabros Technology, I...	France	Alençon	883 Jackson Circle	791-325-7109
12	jz560990273	Atossa Genetics Inc.	Brazil	Maragogipe	78339 Loftsgordon Court	107-166-6625
13	kb478310551	Northern Trust Corporation	France	Lattes	45 2nd Terrace	852-420-8322
14	partita_iva	ragione_sociale	nazionalita	citta_sede	indirizzo	telefono
15	qk732864515	Superior Uniform Group, Inc.	Poland	Cedry Wielkie	9987 Carberry Terrace	632-595-2501
16	rh052274090	MFS Multimarket Income Tr...	United States	Harrisburg	44885 Artisan Circle	717-528-7502
17	sa881177869	Lam Research Corporation	Brazil	Iratí	0 Westerfield Court	491-894-1108
18	uw816264005	Baltimore Gas & Electric Co...	Poland	Bielawy	6 Esch Way	936-479-9403
19	xb235487850	Denbury Resources Inc.	Brazil	Arcos	42878 Hanson Drive	732-242-0269
20	yc617519055	Koppers Holdings Inc.	France	Roanne	46650 Morrow Point	439-266-6133
21	yg603950868	Blackrock Municipal 2018 T...	Brazil	Baturité	89403 Grasskamp Lane	577-537-2232
22	zs082725179	iPath US Treasury Flatteners ...	United States	Columbus	0064 Glacier Hill Way	614-503-2776

OP5: Registrazione di un nuovo coltivatore, (8 volte all'anno);

Supponendo di partire dalla tabella “coltivatore” nello stato mostrato di seguito:

	partita_iva [PK] character varying (30)	ragione_sociale character varying	nazionalita character varying (50)	citta_sede character varying (50)	indirizzo character varying (50)	telefono character (15)
83	so772982076	UMH Properties, Inc.	France	Aix-les-Bains	68930 Bluestem Plaza	403-607-4301
84	st199509486	Grand Canyon Educati...	France	Quimperlé	559 Trailsway Lane	349-993-2139
85	td713974858	DRDGOLD Limited	United States	Charlotte	3382 Holy Cross Alley	704-646-4866
86	tt334690535	John Hancock Financi...	Brazil	Araçuaí	1 Goodland Street	894-276-1197
87	tu980792486	Easterly Acquisition Co...	Brazil	Matelândia	53548 Union Lane	623-415-5191
88	tw595798527	Summit Therapeutics p...	Brazil	Cruz Alta	65260 Hovde Lane	989-398-4075
89	ty539666123	SigmaTron Internation...	France	Saint-Quentin	9 Oakridge Junction	311-243-4298
90	ul253899468	Eaton Vance High Inco...	Brazil	Guaratinguetá	2 Bashford Street	672-119-0865
91	uw473176678	Stanley Black & Decker,...	Brazil	Nazaré	2 Corscot Alley	837-894-9349
92	vv242797859	Antero Midstream GP LP	Brazil	Ipojuca	4 Sherman Place	156-180-7875
93	wg818399183	Heron Therapeutics, In...	Brazil	João Pessoa	74835 Quincy Plaza	798-394-9048
94	wn028440978	Shineco, Inc.	France	Yvetot	1072 Reinke Junction	205-377-5844
95	wt029580174	Vanguard Intermediate...	France	Besançon	53 Melby Parkway	368-677-2828
96	wz360208053	Ocean Rig UDW Inc.	Brazil	São Roque	5 Waywood Lane	336-370-9585
97	xm614039782	Vanguard Russell 2000...	Poland	Aleksandrów	15 Brickson Park Street	273-489-1035
98	yb054998671	Titan Pharmaceuticals,...	France	Caen	140 Little Fleur Avenue	508-500-2588
99	yv983108557	Sunshine Bancorp, Inc.	Poland	Borne Sulnowo	410 Morrow Park	783-790-7212
100	za698381697	Qwest Corporation	Poland	Ludwin	271 Monterey Alley	120-443-8870
101	zd650032720	Vical Incorporated	United States	Akron	844 Rigney Avenue	330-437-7766

Si vogliono inserire i dati al Coltivatore con partita_iva : zo946294037. Tale operazione è eseguibile invocando la funzione definita nella sezione “creazione delle query”:

```
SELECT (aggiungi_coltivatore('zo946294037','cultivare','Romania',
'galati','strada lucefarului 3',
'400-892-6598'));
```

La relazione COLTIVATORE viene aggiornata con i dati del venditore inserito, inserendo la riga evidenziata:

	partita_iva [PK] character varying (30)	ragione_sociale character varying	nazionalita character varying (50)	citta_sede character varying (50)	indirizzo character varying (50)	telefono character (15)
85	td713974858	DRDGOLD Limited	United States	Charlotte	3382 Holy Cross Alley	704-646-4866
86	tt334690535	John Hancock Financi...	Brazil	Araçuaí	1 Goodland Street	894-276-1197
87	tu980792486	Easterly Acquisition Co...	Brazil	Matelândia	53548 Union Lane	623-415-5191
88	tw595798527	Summit Therapeutics p...	Brazil	Cruz Alta	65260 Hovde Lane	989-398-4075
89	ty539666123	SigmaTron Internation...	France	Saint-Quentin	9 Oakridge Junction	311-243-4298
90	ul253899468	Eaton Vance High Inco...	Brazil	Guaratinguetá	2 Bashford Street	672-119-0865
91	uw473176678	Stanley Black & Decker,...	Brazil	Nazaré	2 Corscot Alley	837-894-9349
92	vv242797859	Antero Midstream GP LP	Brazil	Ipojuca	4 Sherman Place	156-180-7875
93	wg818399183	Heron Therapeutics, In...	Brazil	João Pessoa	74835 Quincy Plaza	798-394-9048
94	wn028440978	Shineco, Inc.	France	Yvetot	1072 Reinke Junction	205-377-5844
95	wt029580174	Vanguard Intermediate...	France	Besançon	53 Melby Parkway	368-677-2828
96	wz360208053	Ocean Rig UDW Inc.	Brazil	São Roque	5 Waywood Lane	336-370-9585
97	xm614039782	Vanguard Russell 2000...	Poland	Aleksandrów	15 Brickson Park Street	273-489-1035
98	yb054998671	Titan Pharmaceuticals,...	France	Caen	140 Little Fleur Avenue	508-500-2588
99	yv983108557	Sunshine Bancorp, Inc.	Poland	Borne Sulinowo	410 Morrow Park	783-790-7212
100	za698381697	Qwest Corporation	Poland	Ludwin	271 Monterey Alley	120-443-8870
101	zd650032720	Vical Incorporated	United States	Akron	844 Rigney Avenue	330-437-7766
102	zo946294037	cultivare	Romania	galati	strada lucefarului 3	400-892-6598

OP 15: Modifica status di un lotto, (10 volte al mese);

Supponendo di partire dalla tabella “lotto” nello stato mostrato di seguito:

	id [PK] integer	trasporto character (5)	varietà character (3)	forma character (15)	coltivatore character varying (30)	venditore character varying (30)	status character (5)	deposito integer	num_casse_disp integer
1	3	mare	FCV	strips	bu284095694	ah562776935	EXEU	1	172
2	8	terra	LAC	scraps2	jo818268424	fu795111503	EXEU	3	112
3	10	mare	DAC	stem2	cv274811955	zs082725179	EXEU	1	730
4	11	mare	LAC	strips	vv242797859	fq921097786	EXEU	2	290
5	12	mare	FCV	scraps2	qz494483388	zs082725179	EXEU	2	625
6	16	mare	DAC	scraps2	at683309929	sa881177869	EXEU	3	781
7	21	mare	LAC	loose leaves	of483314399	fq921097786	EXEU	1	364
8	24	terra	DAC	stem	rf736562272	ah562776935	EXEU	3	118
9	25	mare	DAC	strips	so772982076	jz560990273	FC	2	425
10	26	mare	DAC	loose leaves	qt726523635	sa881177869	EXEU	3	994
11	28	mare	LAC	scraps2	ty539666123	fq921097786	FC	2	29
12	29	terra	FCV	scraps2	dy039337515	jw382177527	FC	1	509
13	30	terra	FCV	scraps1	wg818399183	ah562776935	EXEU	2	174

```
UPDATE public.lotto
SET status='FC'
WHERE id=8;
```

La relazione LOTTO viene aggiornata modificando i dati dello status, modificando la riga evidenziata.

	id [PK] integer	trasporto character (5)	varietà character (3)	forma character (15)	coltivatore character varying (30)	venditore character varying (30)	status character (5)	deposito integer	num_casse_disp integer
1	3	mare	FCV	strips	bu284095694	ah562776935	EXEU	1	172
2	8	terra	LAC	scraps2	jo818268424	fu795111503	FC	3	112
3	10	mare	DAC	stem2	cv274811955	zs082725179	EXEU	1	730
4	11	mare	LAC	strips	vv242797859	fq921097786	EXEU	2	290
5	12	mare	FCV	scraps2	qz494483388	zs082725179	EXEU	2	625
6	16	mare	DAC	scraps2	at683309929	sa881177869	EXEU	3	781
7	21	mare	LAC	loose leaves	of483314399	fq921097786	EXEU	1	364
8	24	terra	DAC	stem	rf736562272	ah562776935	EXEU	3	118
9	25	mare	DAC	strips	so772982076	jz560990273	FC	2	425
10	26	mare	DAC	loose leaves	qt726523635	sa881177869	EXEU	3	994
11	28	mare	LAC	scraps2	ty539666123	fq921097786	FC	2	29

OP28: (100 al mese) **Stampare un report dell'intero iter della cassa** nel magazzino comprensivo di:

- Lotto di appartenenza (comprensivo di status corrente);
- Data di arrivo (eventuale di fine quarantena e di trattamento);
- Nazione di Origine e provenienza del tabacco;
- La data di ispezione, se ispezionata;
- L'eventuale referto;
- Ordine con cui è stata spedita (con data e mezzo di spedizione).

Questa operazione è tra le più complesse in quanto prevede un join tra più tabelle per descrivere il corretto iter di una cassa. Di seguito sono riportati i join necessari:

```
FROM ("cassa" join "carico" on "cassa".carico = "carico".numero_progressivo and "cassa".lotto =  
"carico".lotto) left outer join  
"cassa_split" on "cassa".barcode = "cassa_split".cassa left outer join  
"referto" on "cassa".barcode = "referto".cassa join  
"lotto" on "cassa".lotto = "lotto".id left outer join  
"coltivatore" on "lotto".coltivatore = "coltivatore".partita_iva left outer join  
"venditore" on "lotto".venditore = "venditore".partita_iva left outer join  
"spedizione" on "cassa".ordine = "spedizione".ordine
```

I risultati seguenti sono ottenuti dalla ricerca di una cassa che non ha subito split in magazzino (figura 1) e dalla ricerca di una cassa che ha subito uno split (figura 2):

```
SELECT public.stampa_iter_cassa(  
    '00681K428VPP'  
)
```



The screenshot shows a database query interface. The top part displays the SQL query: `SELECT public.stampa_iter_cassa('00681K428VPP')`. Below the query, there are tabs for "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is selected, showing a table with one record. The table has a header row with the text "stampa_iter_cassa record" and a lock icon. The data row contains a long string representing the output of the query.

	stampa_iter_cassa record
1	("00681K428VPP ", "200", "C48 ", "0,2021-09-26,2021-09-28,2021-09-29,,,,,,211", "FC ", "Poland,Brazil,7,0,,")

Figura 1


```
SELECT public.stampa_iter_cassa(
    '02C4UD5LTCZI'
)
```



The screenshot shows a PostgreSQL query editor with the following query:

```
1 SELECT public.stampa_iter_cassa(
2     '02C4UD5LTCZI'
3 )
```

The results are displayed in a table named 'stampa_iter_cassa' with the following data:

id	stampa_iter_cassa
1	("02C4UD5LTCZI", "0,200;C48", "3,2021-11-09,2021-11-11,2021-11-12,,2020-01-12;BSLOIRKAPXWV", "0.82,4.51,15.63,0.13,195;FC", "France,France,990,0;79 DCM 38", "2020...
2	("02C4UD5LTCZI", "1,200;C48", "3,2021-11-09,2021-11-11,2021-11-12,,2020-01-12;BSLOIRKAPXWV", "0.82,4.51,15.63,0.13,195;FC", "France,France,990,0;79 DCM 38", "2020...

Figura 2

I valori riportati sono elencati nel seguente ordine:

- barcode
- partizionamento
- peso
- tipo_imballaggio
- numero_progressivo_carico_di_appartenenza
- data_di_controllo
- data_fine_quarantena
- data_trattamento
- data_ispezione
- data_referto
- codice_referto
- %nh3
- %nic
- %rs
- %no3
- lotto_di_appartenenza
- status_corrente_del_lotto
- origine
- provenienza
- ordine
- batch
- mezzo_di_trasporto
- data_spedizione