

RollingWaves

PROGETTO DI BASI DI DATI

BUSACCA LUCA

FAVARON GIOVANNI

1. ABSTRACT

RollingWaves è una società nata negli anni 90 dalla passione di due ragazzi di Compton, Los Angeles, per lo skateboarding. Questa passione ha portato Luke e John ad aprire un negozio di tavole da skateboard e accessori nel centro di Los Angeles. Il successo del negozio li ha portati a trasformare la loro bottega in una catena di negozi presente in varie città americane e ad aprirsi anche alla vendita di altri prodotti come abbigliamento tecnico, tavole da snowboard e da surf. La loro notorietà è cresciuta a tal punto da creare anche una propria linea di abbigliamento sportivo. Negli ultimi anni, a causa dell'aumento della richiesta dei loro prodotti anche oltreoceano, hanno deciso di introdurre un servizio di e-commerce per la vendita on-line con dei magazzini posti in diversi stati e un servizio di consegna diretta a causa della delicatezza e delle dimensioni dei loro prodotti.

L'azienda per introdurre questo servizio nel mercato deve realizzare una base di dati che gestisca gli ordini e i dati dai clienti, i quali devono preventivamente registrarsi sul sito. Inoltre, la base di dati deve archiviare tutte le informazioni riguardanti le sedi e i dipendenti.

2. ANALISI DEI REQUISITI

2.1 DESCRIZIONE TESTUALE

Si vuole realizzare una base di dati che gestisca le informazioni riguardanti clienti, sedi, magazzini e personale.

I clienti possono fare acquisti sul sito creando un account in cui vengono registrate le seguenti informazioni:

- username
- nome e cognome
- indirizzo di spedizione (via, CAP, città)
- metodo di pagamento preferito
- saldo punti

Ogni prodotto in vendita ha un ID tramite il quale viene identificato in modo univoco, e il cliente ha la possibilità di sceglierne la taglia/misura, vedere la disponibilità in magazzino e il prezzo. Scelti i prodotti di suo gradimento il cliente può aggiungerli al carrello e procede con il pagamento, dopo il quale l'ordine verrà preso in consegna dalla sede e sarà inoltrato al magazzino più vicino alla destinazione di consegna scelta dal cliente. Le informazioni relative all'ordine sono le seguenti:

- numero ordine
- data
- totale
- destinazione
- magazzino partenza ordine
- codice tracking

La merce viene procurata e spedita nei magazzini da dei fornitori di cui si conosce:

- partita iva
- nome
- e-mail
- numero di telefono

Le transazioni dell'azienda vengono gestite dalla sede e si compongono del pagamento delle fatture emesse dai fornitori, del pagamento degli stipendi ai dipendenti e dalle entrate derivate dalla vendita dei prodotti.

Ogni transazione è individuabile da:

- ID
- data
- destinatario
- importo

Le sedi dell'azienda sono amministrate da dei direttori e ci lavorano gli impiegati. Le funzioni di una sede sono quella di smistare i vari ordini ai magazzini, gestire tutta la parte finanziaria, ordinare gli approvvigionamenti di merce e l'assunzione e l'organizzazione del personale. Le sedi sono individuate da:

- ID
- e-mail
- telefono
- indirizzo (via, CAP, civico, città, stato)

Ogni sede gestisce uno o più magazzini, i quali ricevono gli ordini dalle sedi. All'interno dei magazzini, i magazzinieri si occupano di preparare gli ordini per la spedizione caricando la merce che li compone sui mezzi. Gli autisti inseguito consegneranno la merce ai clienti. I magazzini hanno i seguenti attributi:

- ID
- indirizzo

Per quanto riguarda il personale, è composto da dirigenti e impiegati che lavorano nelle sedi e da magazzinieri e autisti che lavorano nei magazzini. Gli impiegati in ufficio si occupano di pratiche, smistamento ordini e di tutte le transazioni mentre i dirigenti si occupano di sovrintendere agli impiegati e gestiscono il personale sia nelle sedi che nei magazzini. I magazzinieri e gli autisti invece lavorano nei magazzini. I primi si occupano di preparare gli ordini, eseguono operazioni di carico e scarico dei mezzi, mentre gli autisti si occupano di eseguire le consegne. Tutti i dipendenti hanno delle caratteristiche comuni quali:

- matricola
- nome e cognome
- età
- stipendio

2.2 GLOSSARIO TERMINI

Termine	Descrizione	Sinonimi	collegamenti
Sede	Edificio dove lavorano impiegati e direttore, gestisce i pagamenti, gli ordini e il personale. È composta principalmente da uffici.	Ufficio, sede amministrativa	Magazzino, ordine, fornitore, transazioni, dipendenti, direttore, impiegato

Magazzino	Edificio in cui si svolge l'attività logistica e la preparazione degli ordini da spedire.	Deposito	Sede, ordine, merce, autista, magazziniere, operazioni su magazzino
Ordine	È composto dai prodotti acquistati dal cliente e viene gestito da una sede e preparato da un magazzino.	ordinativo	Magazzino, sede, merce, account
Merce	È l'insieme di tutti i prodotti disponibili alla vendita e presenti in magazzino.	Prodotti, articoli, beni	Ordine, magazzino, specifiche, fornitore, account
Fornitore	Sono i produttori che inviano i beni che dopo vengono venduti.	produttori	Merce, sede, pagamento fatture
Transazioni	Sono tutte le operazioni finanziarie che vengo gestite dalla sede.	pagamenti	Sede, stipendio, incasso, pagamento fatture
Fatture	Sono le fatture emesse dai fornitori che hanno procurato la merce da vendere.		Transazioni, fornitore
Incasso	Sono le entrate derivanti dalle vendite degli articoli.	entrate	Transazioni, cliente
Stipendio	sono i soldi che vengo erogati ai dipendenti come compenso del lavoro svolto.	retribuzione	Transazioni dipendenti
Account	È il punto di contatto tra azienda e cliente che permette a quest'ultimo di fare acquisti.		Cliente, merce, ordini, autista
Cliente	È la persona fisica che acquista beni dall'azienda.	acquirente	Account, incasso
Operazioni di magazzino	Sono tutte le attività logistiche che avvengono all'interno del magazzino.	Operazioni logistiche	Magazzino, magazziniere
Carico	Consiste nel caricare i mezzi che portano a destinazione gli ordini dei clienti.		Operazioni su magazzino
Scarico	Attività che consiste dello scarico della merce inviata dai fornitori.		Operazioni su magazzino
Dipendenti	Sono tutti i soggetti che lavorano e percepiscono uno stipendio dall'azienda.		Sede, stipendio
Direttore	È un dipendente che lavora in una sede e ne è anche il responsabile, si occupa del personale e della supervisione dei dipendenti.	responsabile	Sede, dipendenti
Impiegato	È un dipendente che lavora in una sede, si occupa di inoltrare gli ordini ai magazzini e delle transazioni.		Sede, dipendenti
Magazziniere	È un dipendente che lavora nel magazzino e si occupa di caricare, scaricare e preparare gli ordini.		Magazzino, operazioni su magazzino, dipendenti
Autista	È un dipendente che si occupa di consegnare gli ordini ai clienti.		Magazzino, cliente, dipendenti

2.3 STRUTTURAZIONE DEI REQUISITI

FRASI RELATIVE A SEDE
Le sedi dell'azienda sono amministrate da dei direttori e ci lavorano gli impiegati. Le funzioni di una sede sono quella di smistare i vari ordini ai magazzini, gestire tutta la parte finanziaria, ordinare gli approvvigionamenti di merce e l'assunzione e l'organizzazione del personale.

FRASI RELATIVE A MAGAZZINO

Ogni sede gestisce uno o più **magazzini**, i quali ricevono gli ordini dalle sedi. All'interno dei magazzini, i magazzinieri si occupano di preparare gli ordini per la spedizione caricando la merce che li compone sui mezzi. Gli autisti inseguito consegneranno la merce ai clienti.

FRASI RELATIVE A CLIENTE

I clienti possono fare acquisti sul sito creando un account in cui vengono registrate le seguenti informazioni: [...]

FRASI RELATIVE A MERCE

Ogni prodotto in vendita ha un ID tramite il quale viene identificato in modo univoco, e il cliente ha la possibilità di sceglierne la taglia/misura, vedere la disponibilità in magazzino e il prezzo. Scelti i prodotti di suo gradimento il cliente può aggiungerli al carrello e procede con il pagamento, dopo il quale l'ordine verrà preso in consegna dalla sede e sarà inoltrato al magazzino più vicino alla destinazione di consegna scelta dal cliente.

FRASI RELATIVE A TRANSAZIONI

Le transazioni dell'azienda vengono gestite dalla sede e si compongono del pagamento delle fatture emesse dai fornitori, del pagamento degli stipendi ai dipendenti e dalle entrate derivate dalla vendita dei prodotti.

FRASI RELATIVE A PERSONALE

Per quanto riguarda il personale è composto da dirigenti e impiegati che lavorano nelle sedi e da magazzinieri e autisti che lavorano nei magazzini. Gli impiegati in ufficio si occupano di pratiche, smistamento ordini e di tutte le transazioni mentre i dirigenti si occupano di sovrintendere agli impiegati e gestiscono il personale sia nelle sedi che nei magazzini. I magazzinieri e gli autisti invece lavorano nei magazzini e i primi si occupano di preparare gli ordini, eseguono operazioni di carico e scarico dei mezzi, mentre gli autisti si occupano di eseguire le consegne.

3.PROGETTAZIONE CONCETTUALE

3.1 LISTA DELLE ENTITA'

Le identità della base di dati sono le seguenti:

- Sede: indica una sede amministrativa
 - ID: char (4) PRIMARY KEY (identificatore)
 - e-mail: varchar (50) NOT NULL
 - telefono: varchar (12) NOT NULL
 - indirizzo:
 - via: varchar (60) NOT NULL
 - città: varchar (25) NOT NULL
 - CAP varchar (10) NOT NULL
- Dipendenti: sono tutte le persone fisiche che lavorano per l'azienda
 - Matricola: char (6) PRIMARY KEY (identificatore)
 - Nome: varchar (15)
 - Cognome: varchar (15)
 - Et : int check (eta > 15 AND eta < 120)
 - Stipendio: float NOT NULL

L'entit  dipendenti si suddivide tramite una generalizzazione totale in:

- Direttore
- Impiegato
- Magazziniere
- Autista

- Magazzino: Sono gli edifici dove si svolgono le operazioni logistiche
 - ID: char (4) PRIMARY KEY (identificatore)
 - indirizzo:
 - via: varchar (60) NOT NULL
 - citt : varchar (25) NOT NULL
 - CAP varchar (10) NOT NULL

- Cliente: Informazioni relative ad ogni cliente che acquista nel sito
 - username: varchar (10) PRIMARY KEY (identificatore)
 - Nome: varchar (15) NOT NULL
 - Cognome: varchar (15) NOT NULL
 - Saldo Punti: int check (saldoPunti >= 0),
 - Via: varchar (60) NOT NULL
 - Cap: varchar (10) NOT NULL
 - Città: varchar (25) NOT NULL
 - Metodo Pagamento: varchar (50)

- Ordine: Sono gli ordini di merce fatti dai clienti
 - Numero Ordine: char (11) PRIMARY KEY (identificatore)
 - Totale: float check (totale > 0)
 - Partenza: char (4) NOT NULL
 - Destinazione: varchar (10) NOT NULL
 - tracking: varchar (12) NOT NULL
 - Data: date NOT NULL
 - Annullato: bool NOT NULL

- Operazioni su Magazzino: sono le operazioni di carico e scarico che si fanno in magazzino
 - Magazzino: char (4) (identificatore esterno)
 - Data e ora: timestamp
 - Operazioni: varchar (7) NOT NULL

- Fornitore: sono i produttori che riforniscono la merce
 - Partita IVA: char (11) PRIMARY KEY
 - Nome: varchar (20) NOT NULL
 - e-mail: varchar (50) NOT NULL
 - telefono: varchar (12) NOT NULL

- Merce: sono i prodotti disponibili alla vendita
 - ID: char (5) PRIMARY KEY
 - Marca: varchar (20)
 - Taglia: varchar (6) NOT NULL
 - Prezzo: float NOT NULL

Merce è una generalizzazione totale di tutti i vari prodotti in vendita.

- Transazioni: sono le entrate e le uscite dell'azienda
 - ID: char (12) PRIMARY KEY
 - Importo: float check (importo > 0)
 - Destinatario: char (4) NOT NULL
 - Data: date NOT NULL

Transazioni è una generalizzazione totale di

 - Fatture
 - Incasso
 - stipendio

3.2 LISTA DELLE RELAZIONI

Magazzino – Magazziniere : **Afferenza**

Ciascun magazzino è affarito da uno o più magazzinieri (1,N)

Ciascun magazziniere affarisce a un singolo magazzino (1,1)

Magazzino – Autista : **Afferenza**

Ciascun magazzino è affarito da uno o più autisti (1,N)

Ciascun Autista affarisce a un singolo magazzino (1,1)

Magazzino – Operazioni_su_Magazzino : **Svolgimento**

In ciascun magazzino vengono svolte una o più operazioni di magazzino (1,N)

Ciascuna operazione di magazzino viene svolta su un singolo magazzino (1,1)

Magazzino – Sede : **Gestione**

Un magazzino è gestito da una singola sede (1,1)

Ciascuna sede gestisce uno o più magazzini (1,N)

Magazzino – Merce : **Deposito**

In un magazzino sono depositate svariate merci (1,N)

Ciascun tipo di merce è depositata in uno o più magazzini (1,N)

Magazzino – Ordine : **Partenza**

Da un magazzino possono partire uno o più ordini (0,N)

Un ordine parte da un solo magazzino (1,1)

Ordine – Cliente : **Prenotazione**

Un ordine è prenotato da un solo cliente (1,1)

Un cliente può prenotare uno o più ordini (0,N)

Ordine – Sede: **Evasione**

Un ordine viene evaso da una sola sede (1,1)

Una sede può evadere uno o più ordini (0,N)

Ordine – Merce: **Composizione**

Un ordine è composto da una o più merci (1,N)

Una o più merci compongono un ordine (1,N)

Merce – Cliente : **Aggiunto al carrello**

Una o più merci possono essere aggiunte al carrello dal cliente (0,N)

Un cliente può aggiungere al carrello una o più merci (0,N)

Merce – Fornitore: **Rifornimento**

Una o più merci vengono rifornite da diversi fornitori (1,N)

Un fornitore rifornisce una o più merci (1,N)

Fornitore – Fatture: **Esecuzione**

Ad un fornitore vengono eseguite una o più fatture (1,N)

Una fattura viene eseguita ad un singolo fornitore (1,1)

Fornitore – Sede: **Prenotazione**

Ad un fornitore può essere eseguita una prenotazione da una o più sedi (0,N)

Una sede prenota da uno o più fornitori (1,N)

Sede – Transazioni: **Esecuzione**

Una sede esegue una o più transazioni (1,N)

Una transazione viene eseguita da una sola sede (1,1)

Sede – Dipendente: **Assunzione**

Una sede può assumere uno o più dipendenti (0,N)

Un dipendente viene assunto da una sola sede (1,1)

Sede – Direttore: **Amministrazione**

Una sede viene diretta da un solo direttore (1,1)

Un direttore amministra una sola sede (1,1)

Sede – Impiegato: **Afferenza**

Una sede è afferita da uno o più Impiegati (1,N)

Un Impiegato afferisce una sola sede (1,1)

Dipendente – Stipendio: **Retribuzione**

A un dipendente vengono retribuiti diversi stipendi (1,N)

Uno stipendio viene retribuito a un singolo dipendente (1,1)

Cliente – Autista: **Consegna**

Un cliente può ricevere consegne da diversi autisti (0,N)
 Un autista può consegnare a uno o più clienti (0,N)

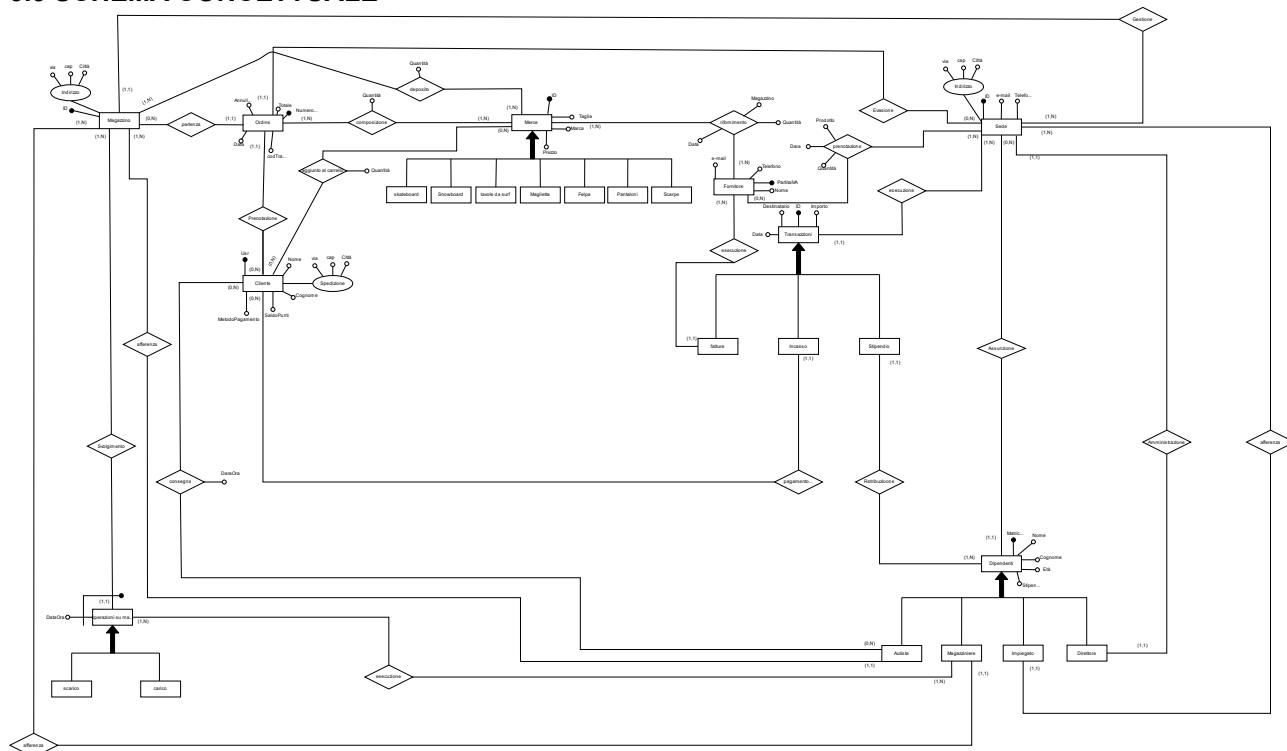
Cliente – Incasso **Pagamento ordine**

Un cliente paga tutti gli ordini effettuati (0,N)
 Un ordine viene pagato da un solo cliente (1,1)

Magazziniere – Operazioni Su Magazzino: **Esecuzione**

Un magazziniere esegue una o più operazioni di magazzino (1,N)
 Un'operazione di magazzino viene eseguita da uno o più magazzinieri (1,N)

3.3 SCHEMA CONCETTUALE



4.PROGETTAZIONE LOGICA

4.1 ANALISI DELLE RIDONDANZE

Analizzando lo schema ER si può notare che l'attributo "Totale" nella relazione "Ordine" genera una ridondanza poiché sarebbe possibile calcolarlo sommando per ogni ordine il costo di ogni prodotto moltiplicato per la quantità di acquisto del prodotto nell'ordine.

Per capire se questa ridondanza porta un vantaggio oppure no è necessario dunque esaminare il costo degli accessi e delle scritture per l'attributo "Totale" e confrontarlo con il costo in accesso e scrittura delle operazioni necessarie qualora questo non ci fosse.

Concetto	Tipo	Volume
Ordine	Entità	100000
Composizione_Ordine	Relazione	1000000
Merce	Entità	200

L'attributo "Totale di fatto è ridondante. Vediamo nel seguito se conviene mantenerlo oppure rimuoverlo.

OPERAZIONE 1 (200 volte al giorno) Memorizzare un ordine con relativa composizione

OPERAZIONE 2 (50 volte al giorno) Stampare tutti i dati relativi agli ordini incluso il totale (supponendo che in media un ordine è composto da 2 articoli.)

A questo punto bisogna capire quanto costa eseguire le operazioni richieste con e senza l'attributo Totale.

CON RIDONDANZA:

OPERAZIONE 1

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Ordine	Entità	1	S

Composizione_Ordine	Relazione	2 (in media)	S
---------------------	-----------	--------------	---

OPERAZIONE 2

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Ordine	Entità	1	L

Assumendo doppio il costo degli accessi in scrittura rispetto a quelli in lettura ottengo dunque:

$$200 * 2 + 200 * 2 * 2 + 50 * 1 = 1250$$

SENZA RIDONDANZA:

OPERAZIONE 1

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Ordine	Entità	1	S
Composizione_Ordine	Relazione	2 (in media)	S

OPERAZIONE 2

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Ordine	Entità	1	L
Composizione_Ordine	Relazione	2 (in media)	L
Merce	Entità	2 (in media)	L

Assumendo doppio il costo degli accessi in scrittura rispetto a quelli in lettura ottengo dunque:

$$200 * 2 + 200 * 2 * 2 + 50 + 50 * 2 + 50 * 2 = 1450$$

E' dunque evidente che, in questo caso, è conveniente mantenere la ridondanza per effettuare le operazioni descritte.

4.2 ELIMINAZIONE DELLE GENERALIZZAZIONI

La gerarchia **MERCE** è una generalizzazione totale dei prodotti venduti dall'azienda. Poiché tutte le merci hanno attributi molto simili tra loro e tutte quante sono identificate da un singolo attributo ID si è scelto di accorpate tutte le entità figlie nell'entità padre. A questo punto si è scelto di aggiungere un attributo **Tipo** per distinguere i diversi tipi di prodotti(Skateboard, Snowboard, Tavole da Surf, Magliette, Pantaloi, Scarpe).

La gerarchia **OPERAZIONI_SU_MAGAZZINO** è una generalizzazione totale delle operazioni che vengono svolte in magazzino (Carico e Scarico) e poiché le operazioni vengono comunque svolte dagli stessi dipendenti e quindi a differenziarle è solo il loro tipo si è scelto di accorpate le entità figlie nell'entità padre aggiungendo un attributo **Tipo** che ne specifica, appunto, il tipo di operazione svolta.

La gerarchia **TRANSAZIONI** è una generalizzazione totale delle transazioni(Entrata, Stipendio, Fattura) di cui ciascuna sede si occupa. Essendo le varie transazioni di diverso tipo, esse sono collegate ad entità diverse. Si è scelto, dunque, di mantenerle separate optando per una eliminazione dell'entità padre, trasferendo così tutti i suoi attributi e le sue relazioni a ciascuna delle entità figlie.

La gerarchia **DIPENDENTE** è una generalizzazione totale delle mansioni dei dipendenti(Autista, Magazziniere, Impiegato, Direttre) che lavorano presso le varie sedi dell'azienda. Ogni tipologia di dipendente svolge mansioni diverse ed è quindi collegata a relazioni diverse. Per questo motivo si è scelto di eliminare l'entità padre trasferendo tutti i suoi attributi e le sue relazioni a ciascuna delle entità figlie.

4.3 ELIMINAZIONE DEGLI ATTRIBUTI COMPOSTI

A causa di impossibilità di rappresentare attributi composti nel modello relazionale è necessario partizionare questi ultimi in più attributi.

Gli attributi Indirizzo delle relazioni Sede e Magazzino vengono partizioni nei tre attributi Via, Città e CAP.

Lo stesso vale per l'attributo composto Spedizione dell'entità Account, che viene dunque partizionato negli stessi tre attributi descritti per le precedenti entità.

4.4 SCELTA DEGLI IDENTIFICATORI PRIMARI

Si è scelto di identificare sedi e magazzini con un ID. In alternativa sarebbe stato possibile identificare i due attributi tramite il loro indirizzo. Questo tipo di identificazione avrebbe però riportato in svariate tabelle alla ripetizione di attributi quali Via CAP e città generando così svariate ridondanze e una maggiore occupazione di spazio.

i dipendenti sono identificati da un numero di matricola. Così facendo è possibile che due o più dipendenti possano possedere lo stesso nome cognome, cosa che altrimenti, optando a mantenere quegli attributi come identificatori, non sarebbe stata possibile.

Anche la merce è stata identificata per mezzo di un ID poiché non era possibile identificarla per mezzo della marca, visto che più prodotti possono avere la stessa marca, né, tantomeno, per mezzo di taglia o prezzo. Gli account sono stati identificati da un nome utente (User) per un ragionamento analogo a quello fatto per sedi e magazzini. In alternativa sarebbe stato possibile identificare ogni cliente tramite attributi quali nome, cognome e indirizzo.

Le operazioni svolte in magazzino sono identificate da un attributo dataOra accompagnato dalla chiave esterna ai magazzini. Questo perché quest'entità è molto vicina ai magazzini in quanto in magazzini diversi possono essere eseguite operazioni dello stesso tipo alla stessa data e alla stessa ora.

L'entità Fornitore è identificata dall'attributo PartitaIVA in quanto non possono esistere due partite IVA identiche.

Transazione e Ordine invece sono identificate rispettivamente da un numeroOrdine e da un ID in quanto ogni tupla ha caratteristiche che possono risultare comuni ad altre tuple e necessitavano dunque di un identificatore che li rendesse unici.

4.5 TRADUZIONE VERSO IL MODELLO RELAZIONALE

SEDE (ID, Via, CAP, Città, e-mail, Telefono, Direttore)

MAGAZZINO (ID, Via, CAP, Città, Gestore)

DIPENDENTE (Matricola, Nome, Cognome, Età, Stipendio, Impiego, SedeAssunzione, sedeAfferita, MagazzinoAfferito)

CLIENTE (Username, Nome, Cognome, SaldoPunti, Via, Città, Cap, MetodoPagamento)

ORDINE (NumeroOrdine, Totale, Partenza, Destinazione, GestitoDaSede, Tracking, Annullato, Data)

OPERAZIONI_MAGAZZINO (DataOra, Magazzino, TipoOperazione)

FORNITORE (PartitaIVA, Nome, e-mail, Telefono)

MERCE (ID, Tipo, Marca, Taglia, Prezzo)

ESECUZIONE_OPERAZIONI_MAGAZZINO (Magazzino, Dipendente, DataOra)

COMPOSIZIONE_ORDINE (Ordine, Prodotto, Quantità)

CARRELLO (Cliente, Merce, Quantità)

CONSEGNA (Dipendente, Cliente, DataOra)

DEPOSITO (Prodotto, Magazzino, Quantità)

MERCE_PRENOTATA (Sede, Fornitore, Prodotto, Quantità, Data)

RIFORNIMENTO_MERCE (Magazzino, Fornitore, Prodotto, Quantità, Data)

STIPENDIO (ID, Importo, Data, Destinatario, Trasmittente)

4.6 VINCOLI DI INTEGRITA' REFERENZIALE

Sede.Direttore → Dipendente.Matricola

Magazzino.Gestore → Sede.ID

Dipendente.SedeAssunzione → Sede.ID

Dipendente.SedeAfferita → Sede.ID

Dipendente:MagazzinoAfferito → Magazzino.ID

Ordine.Partenza → Magazzino.ID

Ordine.Destinazione → Cliente(User)

Ordine.Gestito_da_sede → Sede.ID

Operazioni_Magazzino.Magazzino → Magazzino.ID

Esecuzione_Operazioni_Magazzino.Magazzino → Magazzino.ID

Esecuzione_Operazioni_Magazzino.Dipendente → Dipendente.Matricola

Composizione_Ordine.Ordine → Ordine.NumeroOrdine

Composizione_Ordine.Prodotto → Merce.ID

Carrello.Cliente → Cliente.User

Carrello.Merce → Merce.ID

Consegna.Dipendente → Dipendente.Matricola

Consegna.Cliente → Cliente.User

Deposito.Prodotto → Merce.ID

Deposito.Magazzino → Magazzino.ID

Merce_Prenotata.Sede → Sede.ID

Merce_Prenotata.Fornitore → Fornitore.PartitaIVA

Merce_Prenotata.Prodotto → Merce.ID

Rifornimento_Merce.Magazzino → Magazzino.ID

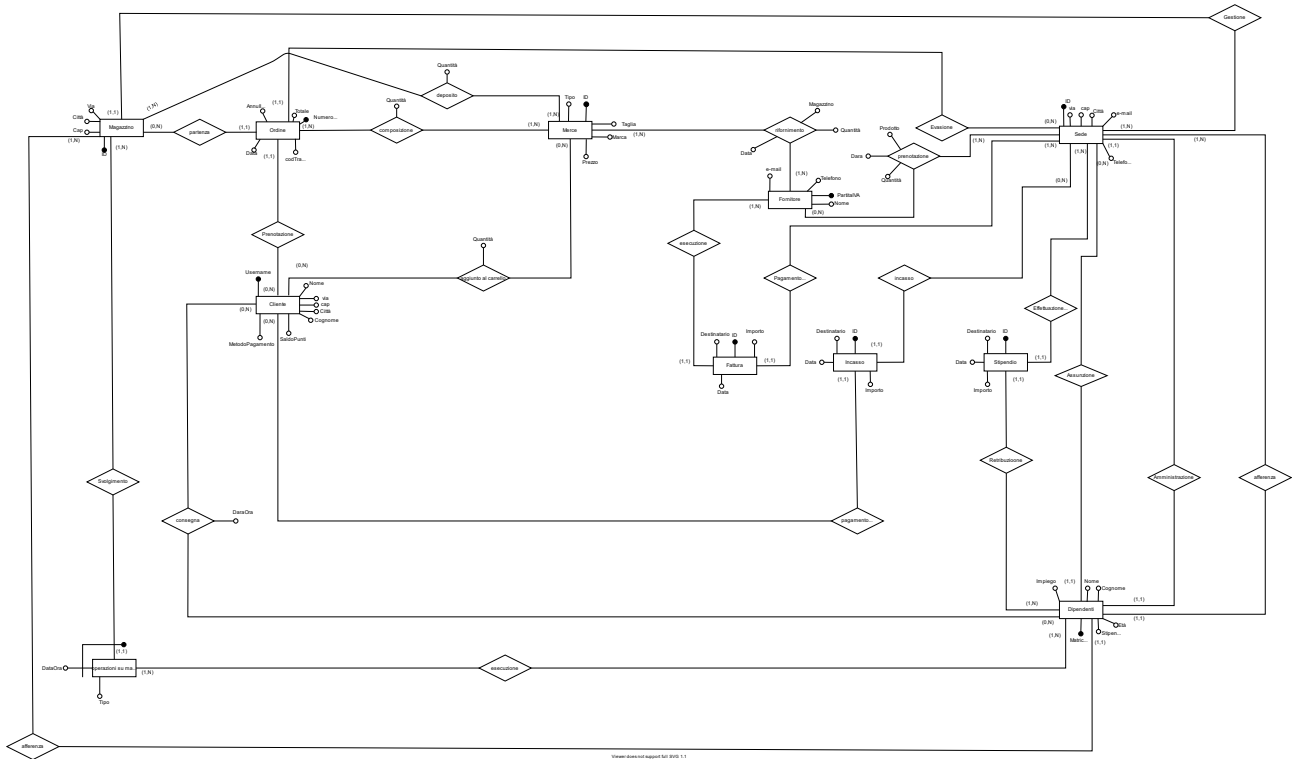
Rifornimento_Merce.Fornitore → Fornitore.PartitaIVA

Stipendio.Destinatario → Dipendente.Matricola

Stipendio.Trasmittente → Sede.ID

Fattura.Destinatario → Fornitore.PartitaIVA
 Fattura.Trasmittente → Sede.ID
 Entrata.Destinatario → Sede.ID
 Entrata.Trasmittente → Cliente.User

4.7 SCHEMA E-R RISTRUTTURATO



5. QUERY E INDICI

5.1 Query

1. Scontare del 20% tutti i prodotti invenduti durante l'anno corrente (2020), mostrare il risultato ordinando il prezzo in ordine decrescente.

```
update merce
set prezzo = prezzo-(prezzo*20)/100
where ID in (select ID
            from merce
            EXCEPT
            select prodotto
            from composizione_ordine join ordine on ordine.numeroOrdine = composizione_ordine.ordine
            where data > '2019-12-31'
            );
select *
from merce
order by prezzo desc;
```

Progetto RollingWaves/postgres@NewServer					
Data Output					
	id [PK] character (5)	tipo character varying (15)	marca character varying (20)	taglia character varying (6)	prezzo double precision
24	SK012	Skateboard	Zero	7.25	87.96000000000001
25	SK002	Skateboard	Jart	8	85.592
26	SH011	Scarpe	Globe	42	84.99
27	SH021	Scarpe	Etnies	41.5	80
28	SH018	Scarpe	Etnies	40	80
29	SK010	Skateboard	RollingWaves	7.75	79.99
30	SK007	Skateboard	Element	8.5	79.96000000000001
31	SH005	Scarpe	DC shoes	41.5	74.9
32	SH010	Scarpe	Globe	41	67.99199999999999
33	SH014	Scarpe	Globe	45	67.99199999999999
34	SH015	Scarpe	Globe	46	67.99199999999999
35	SH012	Scarpe	Globe	43	67.99199999999999
36	SH009	Scarpe	Globe	40	67.99199999999999
37	SH013	Scarpe	Globe	44	67.99199999999999
38	SH022	Scarpe	Etnies	42	64
39	SH016	Scarpe	Etnies	38	64

2. Mostrare a schermo il miglior autista (quello che ha effettuato più consegne) dell'azienda indicandone nome, cognome, matricola e la sede nella quale egli è impiegato.

```
select dipendente.matricola, dipendente.nome, dipendente.cognome, sede.id as Sede
from(
  select max(dipendente) as matricola
  from(
    select dipendente, count(Cliente)
    from consegna
    group by dipendente) as ID) as migliorAutista, dipendente, magazzino, sede
where migliorAutista.matricola=Dipendente.matricola and
dipendente.magazzinoAfferito = magazzino.id and
magazzino.gestore = sede.id;
```

Data Output				
	matricola character (6)	nome character varying (15)	cognome character varying (15)	sede character (4)
1	950615	Woody	Allen	NY01

3. Si desidera conoscere quali fornitori devono essere contattati da ciascuna sede per la prenotazione delle merci. La merce da prenotare è quella che ha una quantità in deposito ≤ 10 . Del fornitore si desidera conoscere:
il nome, la mail e la partita IVA in modo da poterlo contattare ed effettuare il pagamento ordinare il tutto in modo che le stesse sedi compaiano in righe consecutive

```
select sede.id as Sede, merce.id, merce.tipo, deposito.quantita, fornitore.nome, fornitore.email, fornitore.partitaIVA
from merce, deposito, fornitore, magazzino, sede
where prodotto = merce.id
and fornitore.nome = merce.marca
and deposito.magazzino = magazzino.id
and deposito.quantita <= 10
and magazzino.gestore = sede.id
order by sede.id;
```

Data Output							
	sede character (4)	id character (5)	tipo character varying (15)	quantita integer	nome character varying (20)	email character varying (50)	partitaiva character (11)
1	AM01	SN004	Snowboard	0	Burton	Burton@shop.com	00455570713
2	AM01	SK003	Skateboard	10	Jart	Jart@shop.com	01663820502
3	BA01	SK006	Skateboard	2	Element	Element@shop.com	02337350421
4	BA01	SK002	Skateboard	3	Jart	Jart@shop.com	01663820502
5	BE01	SU001	Tavola da surf	7	Slater	Slater@shop.com	02156380426
6	CT01	SK003	Skateboard	0	Jart	Jart@shop.com	01663820502
7	LA01	SK001	Skateboard	2	Jart	Jart@shop.com	01663820502
8	LA01	SK006	Skateboard	5	Element	Element@shop.com	02337350421
9	LO01	SN002	Snowboard	9	Salomon	Salomon@shop.com	02607430218
10	MI01	SK002	Skateboard	6	Jart	Jart@shop.com	01663820502
11	NY01	SH017	Scarpe	7	Etnies	Etnies@shop.com	03721820409
12	NY01	SH022	Scarpe	0	Etnies	Etnies@shop.com	03721820409
13	PA01	SH003	Scarpe	8	DC shoes	DC-shoes@shop.com	96042280782
14	TO01	SN001	Snowboard	0	Salomon	Salomon@shop.com	02607430218
15	TO01	SN002	Snowboard	10	Salomon	Salomon@shop.com	02607430218

4. calcolare il bilancio annuo (2020) di ciascuna sede

```

drop view if exists entrate_2020;
create view entrate_2020(sede, importo) as
select gestito_da_sede as sede, sum(totale) as importo
from ordine
where data > '2019-12-31' and data < '2021-01-01'
group by gestito_da_sede;

drop view if exists uscite_2020;
create view uscite_2020(sede, importo) as
select sede, sum(importo) as importo
from(
    select trasmittente as sede, sum(importo) as importo
    from stipendio
    where data > '2019-12-31' and data < '2021-01-01'
    group by sede
    UNION
    select trasmittente as sede, sum(importo) as importo
    from fattura
    where data > '2019-12-31' and data < '2021-01-01'
    group by sede
) as uscite
group by sede;

select *
from Entrate_2020
where sede not in (select sede from Uscite_2020 )
UNION
select *
from Uscite_2020
where sede not in (select sede from Entrate_2020)
UNION
select Entrate_2020.sede, Entrate_2020.importo - Uscite_2020.importo
from Entrate_2020,Uscite_2020
where Entrate_2020.sede = Uscite_2020.sede;

```

Data Output

	sede character (4)	importo double precision
1	LA01	-4436.5
2	LO01	10273.38
3	BE01	9251.31
4	PA01	50.88
5	AM01	-5162.740000000001
6	TO01	862.88
7	BA01	-1988.5799999999995
8	NY01	-23436.149999999998
9	CT01	-8684.089999999998
10	BO01	9678.509999999998
11	MI01	-18088.04

5. mostrate per ogni amministratore la sede amministrata e l'elenco dei dipendenti che lavorano in ciascuna sede, il tutto ordinato per sede in ordine alfabetico. In particolare:
- della sede verranno mostrati l'id e la città in cui essa si trova;
 - del direttore verrà mostrata la matricola, il nome e il cognome;
 - dei dipendenti verranno mostrati la matricola, il nome, il cognome e la mansione svolta.

```

select sede.id as sede, sede.citta as stabilimento,
       direttore, dipendente.nome as nomeDirettore, dipendente.cognome as cognomeDirettore,
       R1.matricola as impiegato, R1.nome as nomeImpiegato, R1.cognome as cognomeImpiegato, R1.impiego
from sede, dipendente, dipendente as R1
where sede.direttore = dipendente.matricola
      and R1.sedeafferita = sede.id
UNION
select sede.id as sede, sede.citta as stabilimento,
       direttore, dipendente.nome as nomeDirettore, dipendente.cognome as cognomeDirettore,
       R1.matricola as impiegato, R1.nome as nomeImpiegato, R1.cognome as cognomeImpiegato, R1.impiego
from sede, dipendente, dipendente as R1, magazzino
where sede.direttore = dipendente.matricola
      and R1.magazzinoafferito = magazzino.id
      and magazzino.gestore = sede.id
EXCEPT
select sede.id as sede, sede.citta as stabilimento,
       direttore, dipendente.nome as nomeDirettore, dipendente.cognome as cognomeDirettore,
       R1.matricola as impiegato, R1.nome as nomeImpiegato, R1.cognome as cognomeImpiegato, R1.impiego
from sede, dipendente, dipendente as R1
where direttore = R1.matricola
order by stabilimento;

```

Data Output									
	sede character (4)	stabilimento character varying (25)	direttore character (6)	nomedirettore character varying (15)	cognomedirettore character varying (15)	Implegato character (6)	nomempleado character varying (15)	cognomempleado character varying (15)	impiego character varying
1	AM01	Amsterdam	002777	Luigi	Bianchi	877127	Maria	Spinello	autista
2	AM01	Amsterdam	002777	Luigi	Bianchi	560333	Jorre	Dik	magazziniere
3	AM01	Amsterdam	002777	Luigi	Bianchi	276664	Annelien	Corvers	magazziniere
4	AM01	Amsterdam	002777	Luigi	Bianchi	388986	Frank	De Boer	autista
5	AM01	Amsterdam	002777	Luigi	Bianchi	268262	Lydyan	Zomers	impiegato
6	BA01	Barcellona	000023	Lucas	Vasquez	974991	Juan	Mata	autista
7	BA01	Barcellona	000023	Lucas	Vasquez	104446	Eduardo	Mendosa	autista
8	BA01	Barcellona	000023	Lucas	Vasquez	795477	Francisco	Franco	autista
9	BA01	Barcellona	000023	Lucas	Vasquez	712122	Vincent	Van Gogh	magazziniere
10	BA01	Barcellona	000023	Lucas	Vasquez	804719	Esteban	Martinez	magazziniere
11	BA01	Barcellona	000023	Lucas	Vasquez	429904	Alejandro	Gonzalez	autista
12	BA01	Barcellona	000023	Lucas	Vasquez	169836	Luis	Sal	magazziniere
13	BA01	Barcellona	000023	Lucas	Vasquez	737117	Pablo	Picasso	magazziniere
14	BA01	Barcellona	000023	Lucas	Vasquez	536881	Jose	Alvarez	impiegato
15	BA01	Barcellona	000023	Lucas	Vasquez	082050	Denise	Godoi	impiegato
16	BE01	Berlino	000753	Bastian	Schweinsteiger	035771	Egon	Schiele	autista
17	BE01	Berlino	000753	Bastian	Schweinsteiger	926256	Tanja	Schroeder	magazziniere
18	BE01	Berlino	000753	Bastian	Schweinsteiger	756439	Dustin	Adler	impiegato
19	BE01	Berlino	000753	Bastian	Schweinsteiger	746049	Walfgang	Patersen	autista

6. Mostrare la top 5 degli articoli più venduti durante l'anno corrente (2020). In particolare si mostrino: L'ID, il tipo, e la marca del prodotto e quanti pezzi ne sono stati venduti

```
SELECT merce.id, merce.tipo, merce.marca, COUNT(*) AS quantita
FROM merce, composizione_ordine, ordine
WHERE merce.id = composizione_ordine.prodotto
      AND composizione_ordine.ordine = ordine.numeroOrdine
      AND ordine.data > '2019-12-31' AND ordine.data < '2021-01-01'
      AND ordine annullato = 'false'
GROUP BY (merce.id, merce.tipo, merce.marca)
ORDER BY quantita DESC
LIMIT 5;
```

Data Output				
	id [PK] character (5)	tipo character varying (15)	marca character varying (20)	quantita bigint
1	SU005	Tavola da surf	RollingWaves	2
2	TS007	T-shirt	Murder	2
3	SN007	Snowboard	Nitro	2
4	SU004	Tavola da surf	Firewire	2
5	TS008	T-shirt	Murder	2

5.2 INDICI

1. l'operazione di controllo merci in deposito è essenziale sia per la prenotazione di merci che per la disponibilità all'acquisto. Motivo per cui l'operazione viene eseguita svariate volte ogni giorno.

Per questo motivo si è quindi scelto di creare un indice per l'attributo quantità della tabella Deposito.

L'obiettivo è dunque quello di velocizzare le operazioni di visualizzazione della disponibilità delle merci in deposito.

```
drop index if exists quantita_in_magazzino;  
create index quantita_in_magazzino on deposito(quantita);
```

2. La data in cui è stato effettuato un ordine è un dato molto importante sia per il cliente che per le sedi in quanto viene data al cliente la possibilità di filtrare lo storico ordini in base alla data. Lo stesso vale per le sedi.

Per questo motivo si è quindi scelto di creare un indice per l'attributo data della tabella Ordine.

L'obiettivo è dunque quello di velocizzare le operazioni di ricerca di ordini filtrati in base ad una data, ad esempio quelli dell'ultimo mese o anno o ancora quelli successivi o precedenti ad una certa data.

```
drop index if exists data;  
create index data on ordine(data);
```

6.NOTE

Si è scelto di creare un file .cpp per ogni query.

Questa scelta è stata presa sulla base del fatto che alcune query sono state rese interattive mediante l'inserimento di un attributo, mentre altre no.

Per questo motivo le prime query citate mostrano come risultato nella relazione l'unione dei risultati di ciascuna query qualora l'utente inserisse tutti i valori possibili.

Per quanto riguarda l'esecuzione degli script:

Come primo passo è necessario creare un database in PostgreSQL ed importare l'intero codice di creazione delle tabelle e popolamento delle tabelle. (è possibile inserire anche gli indici e al massimo una query ma non tutte le query insieme).

A questo punto è necessario inserire all'interno dei file .cpp il nome del database alla macro

#define PG_DB = "RollingWaves"

Sostituendo RollingWaves con il nome assegnato al database precedentemente creato.

In alternativa è possibile chiamare il database **RollingWaves** così da non dover modificare quel campo.

Il passaggio successivo richiede ancora una modifica ai file .cpp. Bisognerà infatti inserire alla macro

#define PG_PASS "Password"

La password per accedere al server PostgreSQL che è stata impostata dall'utente.

Come ultimo passo precedente all'esecuzione sarà necessaria la compilazione di ciascun file .cpp.

Durante l'esecuzione le query interattive chiedono di inserire un valore. A questo proposito viene fornito un elenco dei valori ammissibili. Qualora venisse inserito un valore non ammissibile verrà fatto notare a schermo l'errore di inserimento con la richiesta di inserire nuovamente il valore, o altrimenti premere q e mandare in input premendo invio per uscire dal programma. Qualora venisse inserito un valore accettabile lo script manderà in output la query corrispondente e terminerà. Le query non interattive, dopo essere state eseguite mostrano semplicemente a schermo i risultati desiderati e lo script termina.