ItaliExpress DB

1 Abstract

ItaliExpress è un azienda italiana che vuole aprire la propria attività di mercato online sul territorio nazionale e come tale necessità di un database per la gestione delle sue risorse.
Grazie a *ItaliExpress* sarà possibile fare acquisti tra una vasta serie di prodotti senza doversi recare in negozio, potendo scegliere se ricevere la consegna direttamente a casa propria oppure specificando durante l'acquisto il punto di ritiro che si preferisce, per poi andarlo a ritirare personalmente.
Ogni utente che desidera fare degli acquisti dovrà registrarsi attraverso la propria email, aggiungendo i propri dati personali e i metodi di pagamento direttamente nel sito. Inoltre per chi lo desidera ci si potrà abbonare all'offerta "Prime" per evitare i costi di spedizione su tutti i prodotti certificati Prime e per poter ricevere le propire consegne in brevissimo tempo. In più verrà fornito l'accesso ad altri servizi esclusivi *ItaliExpress*.
Visualizzando i propri acquisti sarà possibile conoscere la data di presa in carico della spedizione e quella di arrivo al punto di consegna. In caso di necessità sarà anche possibile specificare alcune preferenze di spedizione, in segutio all'avvenuta operazione di acquisto dei propri prodotti.
ItaliExpress assicura ai propri clienti che i fornitori siano accertati mettendo a disposizione la località di produzione ed un recapito telefonico.
In caso di guasto o per via di una determinata motivazione specificata dall'utente sarà possibile effettuare il reso dei prodotti acquistati avendo la garanzia di rimborso della spesa.

2 Analisi dei requisiti

2.1 Descrizione testuale

Nella base di dati sono presenti i dati degli **utenti** registrati, di ogni utente sono noti:

- Nome
- Cognome
- Indirizzo email
- Password
- Numero di telefono
- Indirizzo di residenza per la consegna
- Metodi di pagamento

Un utente può abbonarsi all'offerta "Prime", che avrà durata mensile o annuale in base alla scelta dell'utente. L'utente abbonato non paga i costi di spedizione su tutti i prodotti indicati.
Le **carte di credito** utilizzate dall'utente devono essere provviste di:

- Numero
- Circuito (Mastercard, Visa, . . .)
- Scadenza
- CVV
- Intestatario

I pagamenti vengono autorizzati attraverso un sito esterno, rilevato automaticamente dal circuito.
Ogni **fornitore** deve fornire le informazioni riguardo a:

- Nome
- Numero di telefono
- Indirizzo email
- Partita IVA
- Indirizzo (Via, Numero civico, CAP, Città, Provincia, Stato) dei propri stabilimenti

Gli stabilimenti dei fornitori possono trovarsi anche in paesi esteri fuori dall'Italia.
Di ogni **prodotto**, si vuole conoscere:

- Codice
- Nome
- Prezzo
- Quantità disponibile
- Peso
- Descrizione
- Costo di spedizione
- Offerta prime

Ogni **ordine** deve memorizzare le informazioni riguardo:

- Carrello (Utente e Prodotti ordinati)
- Metodo di pagamento utilizzato
- Data e ora di acquisto
- Indirizzo di consegna
- Preferenze di spedizione (opzionale)

Gli ordini possono essere ritirati dal cliente, in quel caso deve essergli fornita la data e l'ora di arrivo della propria consegna al punto di ritiro scelto oppure possono essere spediti all'indirizzo di residenza dell'utente.
Quando viene richiesta una **spedizione** si vogliono memorizzare le informazioni riguardo:

- Codice
- Data di partenza prevista
- Data di arrivo prevista
- Data effettiva

Per ricevere i propri prodotti interessa conoscere i **punti di consegna**. Per ogni Punto di consegna si vogliono memorizzare le seguenti informazioni:

- Indirizzo (Via, Numero civico, CAP, Città, Provincia, Stato)

Inoltre, se il punto di consegna è un punto di ritiro scelto dall'utente (e quindi non il suo indirizzo di residenza), ci interessa sapere:

- Orario di apertura
- Orario di chiusura

Infine per effettuare un **reso** è necessario specificare:

- Utente
- Prodotto acquistato
- Quantità da rendere
- Motivazione

2.2 Glossario dei termini

Termine	Descrizione	Collegamenti
Utente	Utente registrato al sito	Carrello, CartaDiCredito, Residenza
Prime	Utente dotato di un abbonamento attivo che usufruisce delle agevolazioni di acquisto e spedizione	Entità figlia di Utente
Carrello	Zona di memorizzazione dei prodotti selezionati che non sono ancora stati acquistati	Prodotto, Ordine, Utente
Prodotto	Un bene tangibile, venduto nel sito e acquistabile da qualunque utente	Fornitore, Carrello, Reso

Costo spedizione	Il costo della spedizione (nel caso di validità ”Prime” questo non viene calcolato nell’importo totale)	Attributo di Prodotto
Importo	Costo totale dei prodotti salvati nel carrello. Ogni prodotto comprende anche i costi di spedizione	Attributo di Carrello
Fornitore	L’ente che produce e mette in vendita i propri prodotti	Prodotto, Stabilimento
PIVA	La partita IVA è una sequenza di cifre che identifica univocamente un soggetto che esercita un’attività	Attributo di Fornitore
Stabilimento	Una sede di produzione di un fornitore	Fornitore, Reso
Spedizione	Tempo che trascorre tra la presa in carico fino al punto di consegna di uno o più ordini	Ordine
Codice spedizione	Codice univoco dal quale è possibile ottenere le date di spedizione del proprio ordine	Attributo di Spedizione
PuntoDiConsegna	Indirizzo di spedizione	Ordine
PuntoDiRitiro	Esercente che fornisce il proprio stabilimento per la ricezione e il ritiro delle consegne a determinati orari	Entità figlia di PuntoDiConsegna
Reso	La possibilità di ottenere il rimborso su un prodotto acquistato restituendolo al fornitore	Ordine, Prodotto, Stabilimento

2.3 Operazioni

Operazione	**Tipo**	**Frequenza**
Inserimento di un nuovo utente	S	500 al giorno
Inserimento di un nuovo prodotto	S	1000 al mese
Inserimento di un nuovo ordine	S	5000 al giorno
Ricerca di un prodotto	L	5000 all’ora
Ricerca dei dati riguardanti un fornitore	L	200 al giorno
Visualizzazione dei prodotti nel proprio carrello	L	25000 al giorno
Aggiornarmento della quantità disponibile di un prodotto	S	5000 al giorno
Richiesta di reso di un prodotto	S	2000 al mese
Visualizzazione della data di arrivo del proprio ordine	L	500 al giorno
Ricerca della fascia oraria di apertura di un punto di ritiro	L	500 al giorno

3 Progettazione Concettuale

3.1 Lista entità

Se non specificato l’attributo è NOT NULL

- Utente
  - Email: varchar(100) PRIMARY KEY
  - Nome: varchar(50)
  - Cognome: varchar(50)
  - NumeroTelefono: varchar(15)
  - Password: varchar(20)

    L’entità utente si specializza in una sottocategoria con una **generalizzazione parziale**:
  - Utente Prime
    - DataIscrizione: timestamp
    - DataScadenza: timestamp, ($>$ DataIscrizione)
    - Abbonamento: Enumerated, (”MENSILE”, ”ANNUALE”)
- Carrello

  - Id: integer PRIMARY KEY
  - Utente: varchar(100) PRIMARY KEY
  - Importo: decimal, (>= 0)
- Prodotto
  - Codice: varchar(14) PRIMARY KEY
  - Nome: varchar(100)
  - Prezzo: decimal, (>= 0)
  - QuantitàDisponibile: integer, (>= 0)
  - Peso: decimal, (> 0)
  - Descrizione: varchar(5000)
  - CostoSpedizione: decimal, (>= 0)
  - Prime: boolean
- Fornitore
  - PIVA: varchar(11) PRIMARY KEY
  - Nome: varchar(50)
  - Email: varchar(100)
  - NumeroTelefono: varchar(15)
- Stabilimento
  - Via: varchar(100) PRIMARY KEY
  - NumeroCivico: varchar(10) PRIMARY KEY
  - CAP: varchar(5) PRIMARY KEY
  - Città: varchar(100)
  - Stato: varchar(100)
- CartaDiCredito
  - Numero: varchar(16) PRIMARY KEY
  - Circuito: varchar(25)
  - Scadenza: date
  - CVV: varchar(3)
  - Intestatario: varchar(50)
- Ordine
  - Carrello: integer PRIMARY KEY
  - Utente: varchar(100) PRIMARY KEY
  - DataAcquisto: timestamp
  - PreferenzeSpedizione: varchar(500) può essere NULL
- Spedizione:
  - Codice: varchar(13) PRIMARY KEY
  - DataPartenza: timestamp
  - DataArrivo: timestamp (> DataPartenza)
  - DataEffettiva: timestamp può essere NULL
- PuntoDiConsegna
  - Via: varchar(100) PRIMARY KEY
  - NumeroCivico: varchar(10) PRIMARY KEY
  - CAP: varchar(5) PRIMARY KEY
  - Città: varchar(100)

    L'entità Punto di consegna si specializza in due sottocategorie con una **generalizzazione totale**:

  - PuntoDiRitiro

    - OraApertura: time
    - OraChiusura: time, ($>$ OraApertura)
  - Residenza
- Reso
  - Utente: varchar(100) PRIMARY KEY
  - Carrello: integer PRIMARY KEY
  - Ordine: integer PRIMARY KEY
  - Prodotto: varchar(14) PRIMARY KEY
  - Quantità: integer, ($>$ 0)
  - Motivazione: varchar(500)

3.2 Tabella delle relazioni

Relazione	Entità coinvolte	Descrizione	Attributi
Acquisto	Carrello (0,1) Ordine (1,1)	Un utente può decidere di acquistare i prodotti che ha all'interno del carrello (facendo quindi un ordine relativo a quel carrello), oppure no e quindi non procedere all'acquisto	Data e ora: TIMESTAMP
Cliente	Ordine (0,N) Reso (1,1)	Un utente può decidere di effettuare il reso di uno o più prodotti, in tal caso l'ordine permette di risalire al cliente che ha acquistato il prodotto da restituire	Nessuno
Attività	Fornitore (1,N) Prodotto (1,1)	Un fornitore può mettere in vendita un certo quantitativo di prodotti che quindi saranno riconducibili a lui	Nessuno
Consegna	Ordine (1,1) Spedizione (1,N)	Una volta effettuato un ordine, va organizzata una spedizione affinché venga consegnato all'utente	Nessuno
Località	Fornitore (1,N) Stabilimento (1,1)	Un fornitore avrà a disposizione uno o più stabilimenti che metteranno a disposizione i prodotti pronti per la vendita	Nessuno
Località	Residenza (1,N) Utente (0,1)	Un utente può registrare al più un indirizzo per la residenza a cui potrà far arrivare le spedizioni richieste	Nessuno
Metodo di pagamento	CartaDiCredito (1,1) Utente (1,N)	Un utente deve registrare uno o più metodi di pagamento, per poter effettuare degli acquisti	Nessuno
Possiede	Carrello (1,1) Utente (1,1)	Un utente avrà a disposizione un carrello in cui mettere tutti i prodotti che intende acquistare	Nessuno
ProdottiSalvati	Carrello (0,N) Prodotto (1,1)	Un carrello al suo interno avrà una lista di prodotti che l'utente intende acquistare	Quantità: INTEGER $>$ 0
Rifiuto	Prodotto (0,N) Reso (1,1)	E' possibile effettuare il reso del prodotto se, ad esempio, non rispetta la descrizione con cui è stato presentato	Nessuno
Transazione	CartaDiCredito (0,N) Ordine (1,1)	Una carta di credito può essere utilizzata per zero o più ordini, un ordine è sempre associato ad un metodo di pagamento	Codice: VARCHAR
Trasporto	Reso (1,1) Stabilimento (0,N)	Un prodotto che viene restituito va riportato in uno stabilimento che lo prenderà in gestione per smaltirlo	Nessuno
Trasporto	Ordine (1,1) PuntoDiConsegna (0,N)	Una volta effettuato l'ordine, esso va consegnato all'indirizzo scelto dall'utente o, alternativamente, ad un punto di ritiro	Nessuno

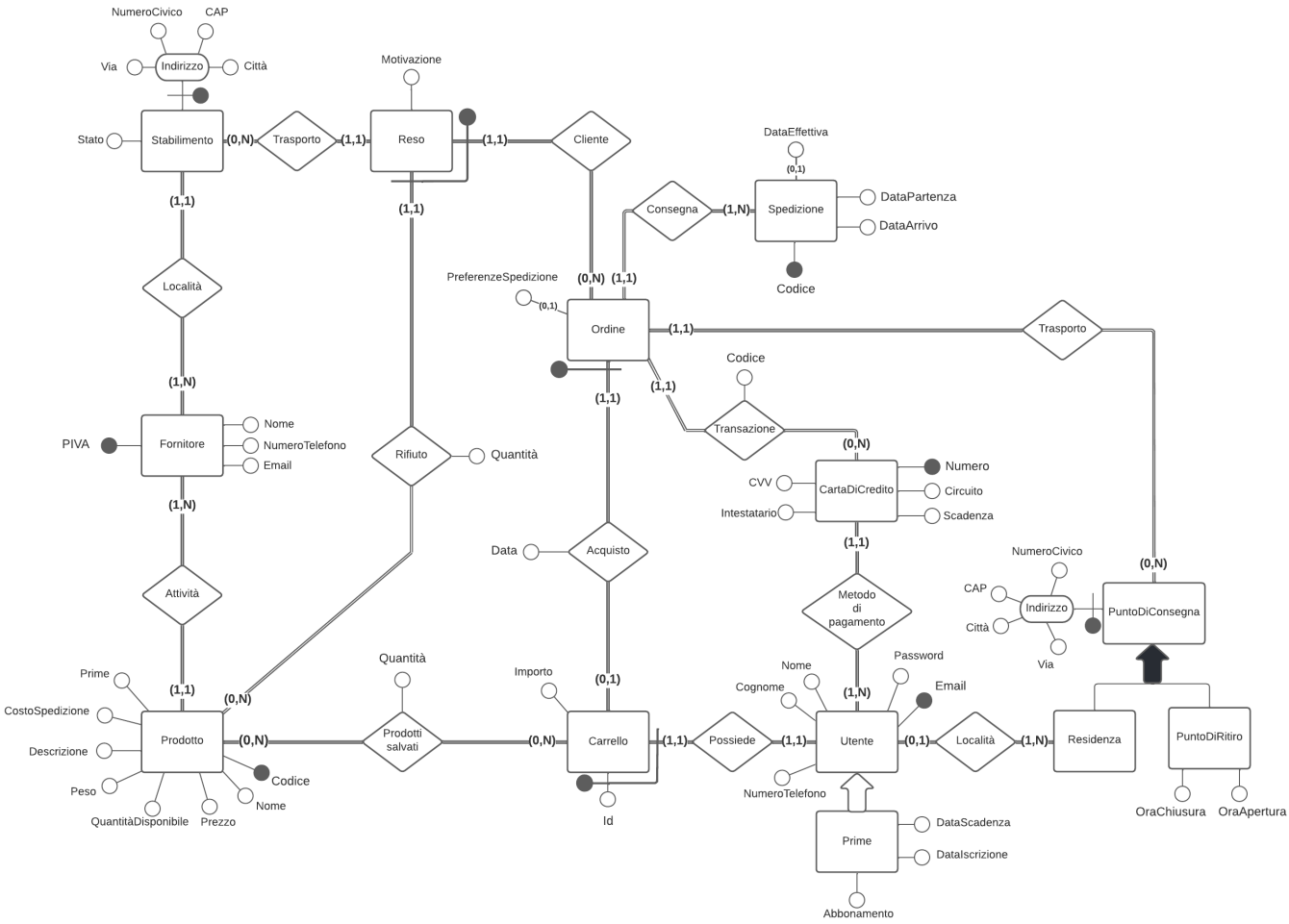
Vincoli non rappresentabili tramite schema E-R:

- Un cliente può effettuare il reso solo dei prodotti che ha precedentemente acquistato e che gli sono stati consegnati
- L'attributo importo dell'entità ordine è uguale alla somma dei prezzi dei prodotti presenti nel carrello più le relative spese di spedizione dei singoli prodotti

Vincoli di derivazione:

- Il valore degli attributi DataIscrizione e DataScadenza dell'entità utente prime devono essere coerenti con il tipo di abbonamento dell'utente

Schema Concettuale



4 Progettazione Logica

4.1 Ristrutturazione

4.1.1 Ananlisi delle ridondanze

L'attributo "**Importo**" dell'entità Carrello è ridondante in quanto si potrebbe calcolare facendo la somma dei prezzi di tutti i prodotti e del costo delle loro spedizioni presenti nel carrello.

Concetto	Tipo	Volume
Prodotto	Entità	2000000 (2M)
Carrello	Entità	250000 (250K)

ProdottiSalvati	Relazione	500000 (500K)

- Operazione 1: Aggiungere un nuovo prodotto al carrello (10000 al giorno)
- Operazione 2: Visualizzare l'importo totale del carrello (25000 al giorno)

PRESENZA RIDONDANZA
Operazione 1:

Concetto	**Costrutto**	**Accessi**	**Tipo**	
ProdottiSalvati	Relazione	1	S	x 10000
Prodotto	Entità	1	L	x 10000
Carrello	Entità	1	L	x 10000
Carrello	Entità	1	S	x 10000

Operazione 2:

Concetto	**Costrutto**	**Accessi**	**Tipo**	
Carrello	Entità	1	L	x 25000

Perciò tenendo conto del fatto che le operazioni di scrittura valgono il doppio rispetto a quelle di lettura.
COSTO TOTALE: 10000*2 + 10000 + 10000 + 10000*2 + 25000 = 85000 (85K)

ASSENZA RIDONDANZA

Concetto	**Costrutto**	**Accessi**	**Tipo**	
ProdottiSalvati	Relazione	1	S	x 10000

Operazione 2:
Sono presenti in media 500000 / 250000 = 2 prodotti a carrello

Concetto	**Costrutto**	**Accessi**	**Tipo**	
Carrello	Entità	1	L	x 25000
ProdottiSalvati	Relazione	2	L	x 25000
Prodotto	Entità	2	L	x 25000

COSTO TOTALE: 10000*2 + 10000 + 25000*2 + 25000*2 = 130000 (130K)

Dall'analisi eseguita risulta quindi che la presenza dell'attributo importo può risparmiare quotidianamente il costo computazionale di (130K - 85K) = 45K operazioni, perciò verrà mantenuto.

4.1.2 Eliminazione delle generalizzazioni

Generalizzazione	**Risoluzione**

Utente <— Prime	L'entità Prime viene accorpata in Utente, per distinguere un utente standard da uno prime non viene aggiunto alcun attributo. Gli attributi Abbonamento, DataIscrizione e DataScadenza diventano perciò opzionali e assumeranno valore NULL per tutti gli utenti che non si sono mai abbonati. Gli utenti prime con un abbonamento scaduto verranno considerati come degli utenti standard
PuntoDiConsegna <— Residenza, PuntoDiRitiro	Le entità Residenza e PuntoDiRitiro ereditano l'attributo composto Indirizzo da PuntoDiConsegna. Perciò si rimuove l'entità padre e vengono create due nuove relazioni di Trasporto tra Ordine (0,1) - Residenza (0,N) e Ordine (0,1) - PuntoDiRitiro (0,N)

4.1.3 Scelta degli identificatori primari

Per l'entità Carrello è stato deciso di utilizzare come chiave primaria un attributo composto formato da:

- L'attributo Utente, che è anche chiave esterna della relazione Carrello - Utente.
- L'attributo Id.

Perciò attraverso l'identificatore primario ogni volta che un utente acquista il proprio carrello sarà possibile risalire ai prodotti di quell'ordine e ad ogni acquisto effettuato verrà aggiornato il valore di Id per creare un nuovo carrello vuoto.

Durante la fase di progettazione logica si è notato che le entità Utente, Reso e Ordine per poter far riferimento ad uno specifico indirizzo dovevano aggiungere quattro nuovi attributi (NumeroCivico, Via, Citta, CAP) aumentando la ridondanza e rendendo più difficile l'aggiornamento delle tabelle.

Si è deciso perciò di creare una nuova entità **Indirizzo**:

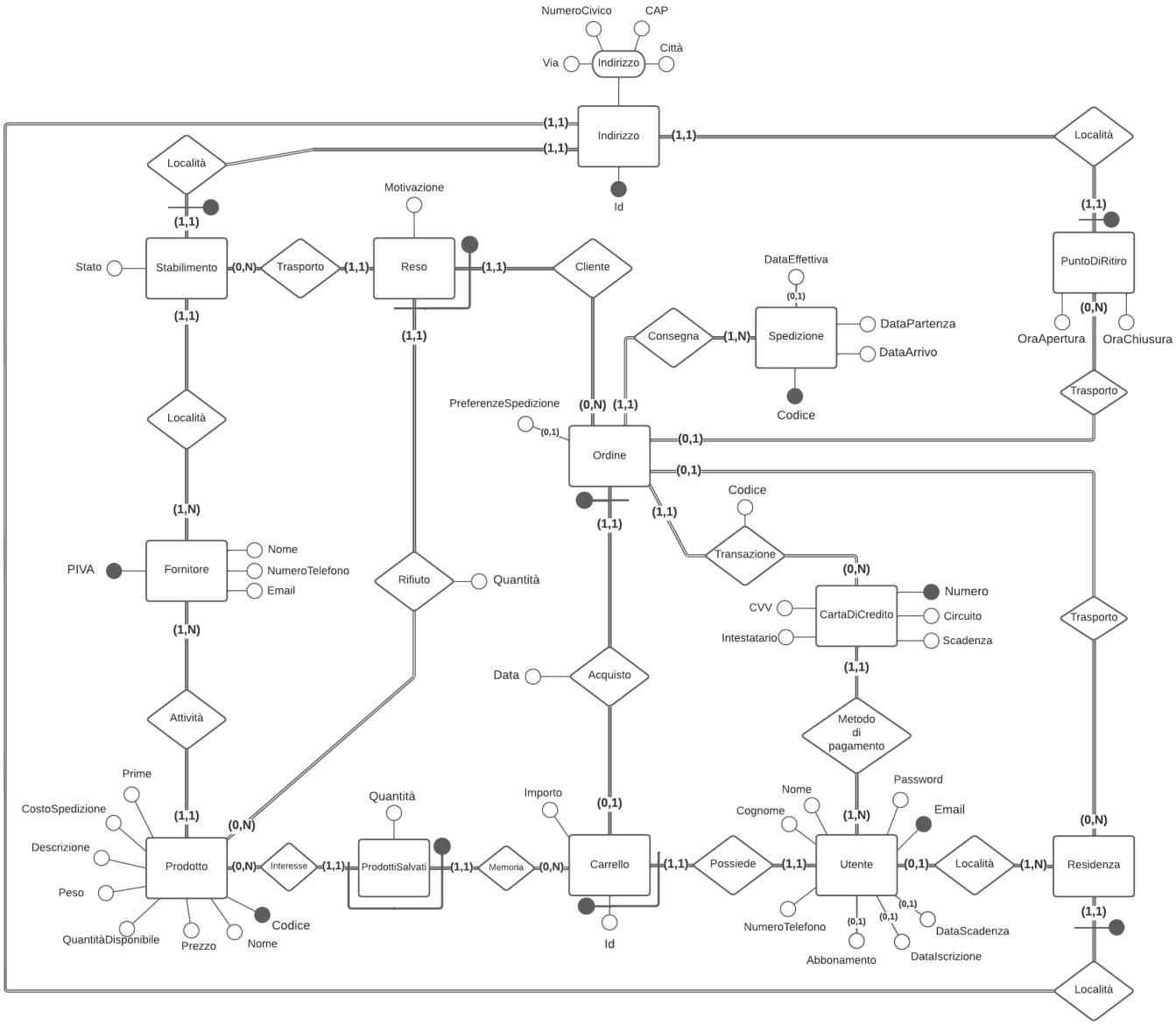
- Id: Integer PRIMARY KEY
- Via: varchar(100)
- NumeroCivico: varchar(10)
- Citta: varchar(100)
- CAP: varchar(5)

Dotata delle seguenti relazioni

Relazione	Entità coinvolte	Descrizione	Attributi
Località	Indirizzo(1,1) Residenza(1,1)	L'indirizzo di consegna che la identifica	Nessuno
Località	Indirizzo(1,1) PuntoDiRitiro(1,1)	L'indirizzo di consegna che la identifica	Nessuno
Località	Indirizzo(1,1) Stabilimento(1,1)	L'indirizzo di consegna che la identifica	Nessuno

Le entità Stabilimento, PuntoDiRitiro e Residenza avranno come chiave primaria l'Id al loro indirizzo.

Schema E-R ristrutturato



4.2 Crezione delle tabelle

(A→B indica che B è chiave esterna di A)
Indirizzo(Id, Via, NumeroCivico, Città, CAP)
Stabilimento(Id → Indirizzo.Id, Stato, Fornitore → Fornitore.PIVA)
Fornitore(PIVA, Nome, NumeroTelefono, email)
Prodotto(Codice, Nome, Prezzo, Prime, CostoSpedizione, Descrizione, Peso, QuantitaDisponibile, Fornitore→Fornitore.PIVA
Reso(Prodotto → Prodotto.Codice, Ordine → Ordine.Carrello, Utente → Ordine.Utente, Motivazione, Indirizzo → Stabilimento.Id)
ProdottiSalvati(Prodotto → Prodotto.Codice, Carrello → Carrello.Id, Utente → Carrello.Utente, Quantita')
Spedizione(Codice, DataPartenza, DataArrivo, DataEffettiva)
Ordine(Carrello → Carrello.Id, Utente → Carrello.Utente, Data, PreferenzeSpedizione, CartaDiCredito → CartaDiCredito.Numero, CodiceTransazione, CodiceSpedizione → Spedizione.codice, Residenza → Residenza.id, PuntoDiRitiro → PuntoDiRitiro.id)
Carrello(Id, Utente → Utente.Email, Importo)
Utente(Email, Password, Nome, Cognome, NumeroTelefono, Abbonamento, DataIscrizione, DataScadenza, Residenza → Residenza)
CartaDiCredito(Numero, Circuito, Scadenza, CVV, Intestatario, Utente → Utente.Email)
Residenza(Id → Indirizzo.Id)
PuntoDiRitiro(Id → Indirizzo.Id, OrarioApertura, OrarioChiusura)

5 Query e indici

5.1 Query

1. Selezionare l'email ed il numero totale di resi di ogni utente.

```
SELECT Utente.Email, COUNT(*) as NumeroResi
FROM Reso, Utente
WHERE Reso.Utente = Utente.Email
GROUP BY Utente.Email
ORDER BY NumeroResi DESC
```

	email [PK] character varying (100)	numeroresi bigint
1	utente2@unipd.it	2
2	utente11@studente.unipd.it	2
3	utente1@studente.unipd.it	2
4	utente6@studente.unipd.it	1
5	utente4@studente.unipd.it	1

2. Selezionare le informazioni dei fornitori che emettono il maggior numero di prodotti prime, i primi 5.

```
SELECT Nome, NumeroTelefono, Email, ProdottiPrime
FROM Fornitore JOIN
(
    SELECT Fornitore as PIVA, COUNT(*) as ProdottiPrime
    FROM Prodotto
    WHERE Prime = TRUE
    GROUP BY PIVA
    ORDER BY ProdottiPrime DESC
    LIMIT(5)
) as FornitorePrime
ON Fornitore.PIVA = FornitorePrime.PIVA
```

	nome character varying (50)	numerotelefono character varying (15)	email character varying (100)	prodottiprime bigint
1	Fornitore3	+390987654323	fornitore3@italiexpress.com	4
2	Fornitore5	+390987654325	fornitore5@italiexpress.com	4
3	Fornitore6	+390987654326	fornitore6@italiexpress.com	4
4	Fornitore7	+390987654327	fornitore7@italiexpress.com	4
5	Fornitore8	+390987654328	fornitore8@italiexpress.com	4

3. Selezionare gli utenti che hanno acquistato almeno un prodotto realizzato da un fornitore che possiede almeno uno stabilimento all'estero e la città della loro residenza.

```
DROP VIEW IF EXISTS FornitoreEstero;
CREATE VIEW FornitoreEstero as
SELECT PIVA
FROM Fornitore, Stabilimento
WHERE Stabilimento.Fornitore = Fornitore.PIVA
AND Stato <> 'IT';

SELECT Nome, Cognome, Citta
FROM Utente, Indirizzo,
(
    SELECT DISTINCT ProdottiAcquistati.Utente
    FROM Prodotto JOIN
    (
        SELECT ProdottiSalvati.Prodotto, ProdottiSalvati.Utente
        FROM Ordine, Carrello, ProdottiSalvati
        WHERE Ordine.Carrello = Carrello.id
        AND Ordine.Utente = Carrello.Utente
        AND Carrello.id = ProdottiSalvati.Carrello
        AND Carrello.Utente = ProdottiSalvati.Utente
    ) as ProdottiAcquistati
    ON ProdottiAcquistati.Prodotto = Prodotto.Codice
    WHERE Prodotto.Fornitore IN (SELECT * FROM FornitoreEstero)
) as Utenti
WHERE Utente.Email = Utenti.Utente
AND Utente.Residenza = Indirizzo.id;
```

	nome character varying (50)	cognome character varying (50)	citta character varying (100)
1	utente	1	Padova
2	utente	10	Padova
3	utente	13	Legnaro
4	utente	2	Padova
5	utente	2	Padova
6	utente	4	Padova
7	utente	6	Padova
8	utente	7	Padova

4. Selezionare i prodotti, la quantità, l'ordine e il codice di spedizione delle ultime 5 spedizioni effettuate all'indirizzo "Via Trieste, 63, 35121, Padova".

```
SELECT Nome, Quantita, ProdottiAcquistati.Ordine, ProdottiAcquistati.Utente,
       ProdottiAcquistati.CodiceSpedizione
FROM Prodotto JOIN
(
    SELECT Prodotto as Codice, Quantita, InfoAcquisto.Carrello as Ordine,
           InfoAcquisto.Utente, InfoAcquisto.Codice as CodiceSpedizione
    FROM ProdottiSalvati JOIN
    (
```

```
            SELECT Carrello, Utente, Codice
            FROM Ordine JOIN Spedizione
            ON Ordine.CodiceSpedizione = Spedizione.Codice
            WHERE Ordine.PuntoDiRitiro =
            (
                SELECT Id
                FROM Indirizzo
                WHERE Via = 'Via Trieste'
                AND NumeroCivico = '63'
                AND CAP = '35121'
                AND Citta = 'Padova'
            )
            ORDER BY DataEffettiva
            LIMIT(5)
        ) as InfoAcquisto
        ON InfoAcquisto.Utente = ProdottiSalvati.Utente
        AND InfoAcquisto.Carrello = ProdottiSalvati.Carrello
    ) as ProdottiAcquistati
    ON Prodotto.Codice = ProdottiAcquistati.Codice
```

	nome character varying (100)	quantita integer	ordine integer	utente character varying (100)	codicespedizione character varying (13)
1	Prodotto 5	5	1	utente7@studente.unipd.it	SP023
2	Prodotto 2	2	1	utente7@studente.unipd.it	SP023
3	Prodotto 3	2	1	utente17@studente.unipd.it	SP011
4	Prodotto 3	2	1	utente17@studente.unipd.it	SP011
5	Prodotto 3	2	1	utente1@unipd.it	SP001
6	Prodotto 1	1	1	utente1@unipd.it	SP001
7	Prodotto 1	1	1	utente4@unipd.it	SP023
8	Prodotto 2	2	1	utente5@unipd.it	SP016
9	Prodotto 3	3	1	utente5@unipd.it	SP016

5. Selezionare l'email, il tipo di abbonamento e la carta di credito più usata degli utenti che hanno una spesa totale maggiore di 500 euro.

```
DROP VIEW IF EXISTS OrdiniPerCircuto;
CREATE VIEW OrdiniPerCircuto as
SELECT Circuito, Ordine.Utente, COUNT(*) as OrdiniEffettuati
FROM Ordine, CartaDiCredito
WHERE Ordine.CartaDiCredito = CartaDiCredito.Numero
GROUP BY Circuito, Ordine.Utente;

SELECT UtenteDatoImportoTotale.Email, UtenteDatoImportoTotale.Abbonamento,
OrdiniPerCircuto.Circuito, UtenteDatoImportoTotale.ImportoTotale
FROM OrdiniPerCircuto,
(
    SELECT Utente.Email as Email, Abbonamento, SpesaUtente.ImportoTotale
    FROM Utente JOIN
    (
        SELECT Carrello.Utente as Email, SUM(Importo) as ImportoTotale
        FROM Ordine, Carrello
```

```
            WHERE Ordine.Utente = Carrello.Utente
            AND Ordine.Carrello = Carrello.Id
            GROUP BY Carrello.Utente
            HAVING SUM(Importo) >= 500
        ) as SpesaUtente
        ON Utente.Email = SpesaUtente.Email
    ) as UtenteDatoImportoTotale
    WHERE UtenteDatoImportoTotale.Email = OrdiniPerCircuto.Utente
    AND OrdiniPerCircuto.OrdiniEffettuati =
    (
        SELECT MAX(OrdiniEffettuati)
        FROM OrdiniPerCircuto
        WHERE OrdiniPerCircuto.Utente = UtenteDatoImportoTotale.Email
    )
```

	email character varying (100)	abbonamento tipoabbonamento	circuito character varying (25)	importototale numeric
1	utente1@studente.unipd.it	MENSILE	Visa	978.47
2	utente7@studente.unipd.it	ANNUALE	Visa	666.30
3	utente13@studente.unipd.it	ANNUALE	Visa	844.97
4	utente6@studente.unipd.it	ANNUALE	Visa	1701.69
5	utente2@studente.unipd.it	ANNUALE	MasterCard	616.41
6	utente4@studente.unipd.it	MENSILE	MasterCard	601.95

5.2 Indici

La ricerca di un prodotto è un operazione che viene eseguita molto spesso in lettura dato che la loro ricerca viene eseguita dagli utenti e anche vengono inseriti automaticamente tra i prodotti consigliati e nella home del sito. Quindi per diminuire il più possibile la latenza nella ricerca e nel caricamento delle pagine si è deciso di indicizzare l'attributo Nome della tabella Prodotto nell'indice NomiProdotti.

```
CREATE INDEX NomiProdotti ON Prodotto(Nome);
```

6 Codice C++

6.1 Descrizione dell'utilizzo del codice

Il codice C++ per l'esecuzione delle query consiste in un unico file .cpp compilabile con il comando

```
g++ main.cpp -L dependencies\lib -lpq -o main
```

A questo punto il comando necessario per l'esecuzione del programma è **./main**.
Prima che il programma esegui le query si cerca di stabilire la connessione con il database attraverso le credenziali e la password di un utente registrato al sistema di gestione basi di dati e solo nel caso la connessione vada a buon fine si procederà con la scrittura e l'esecuzione delle query riportate nel punto 5 per l'interrogazione al database, i risultati verranno stampati a video in formato tabulare.
La query numero 4 richiederà l'inserimento di alcuni parametri da parte dell'utente per la ricerca:

- Via
- Numero civico
- CAP
- Città

Si fa notare che qualora i parametri non vengano inseriti come riportato dal programma nel momento dell'esecuzione, la query potrebbe non restituire i risultati attesi.

6.2 Documentazione codice

Funzioni e pezzi di codice interessanti utilizzati dal programma:

```
void checkResults(PGresult* res, const PGconn* conn)
```

Controlla che il risultato della query contenga la flag di validità altrimenti stamperà a video il messaggio di errore.

```
char conninfo[250];

sprintf(conninfo, "user=%s password=%s dbname=%s hostaddr=%s port=%d",
                  PG_USER, PG_PASS, PG_DB, PG_HOST, PG_PORT);

PGconn *conn = PQconnectdb(conninfo);

if(PQstatus(conn) != CONNECTION_OK) {
    cout<<"Errore di connessione"<<std::endl<<PQerrorMessage(conn);
    PQfinish(conn);
}
```

Cerca di stabilire la connessione al database inviando verso un determinato server (host:porta) le credenziali di accesso al database specificato. Se la connessione non va a buon fine verrà mostrato un messaggio di errore altrimenti la computazione continuerà.

```
res = PQexec(conn, query_1.c_str());
```

Prende come paramentri di input la connessione al database e la query e riceve il risultato associato.

```
template<typename T> void printElement(T t, const int& width)
```

Template di funzione che permette di tabulizzare i risultati di una query utlizzando la libreria 'iomanip'.

```
const char* params[] = {via.c_str(), ncv.c_str(), cap.c_str(), citta.c_str()};

res = PQexecParams(conn, query_4.c_str(), 4, NULL, params, NULL, NULL, 0);
```

Questa parte del programma si occupa dell'inserimento di dati da parte dell'utente via tastiera. La funzione riceve in input la connessione al database, la query e i dati necessari per la processazione dei parametri ovvero il numero totale e il loro valore.