UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO DI MATEMATICA

LAUREA TRIENNALE IN INFORMATICA

PROGETTO DI BASI

Basi di Dati per la Gestione di un Sistema di Spedizioni

Lorenzo Soligo 2101057, Pietro Bassi 2137999

Contents

1 Abstract

Questo lavoro sviluppa una base di dati progettata per gestire in modo strutturato e coerente le informazioni relative a pacchi, ai clienti, ai corrieri, alle filiali delle aziende trasportatrici . L'obiettivo principale è fornire un sistema che monitori il trasporto di un pacco, attraverso tutti gli step di consegna, gestendone i relativi dati, garantendo un accesso organizzato e facilitando la loro analisi. Nel contesto della gestione della spedizione di pacchi, la base di dati proposta distingue tra pacchi bundle (come aggregatore di pacchi per un singolo ordine), pacchi assicurati (dotati di tipologie diverse di assicurazione) e pacchi regalo (dotati di una dedica e incartamento non presente nelle altre tipologie). Attributi comuni sono però quelli legati alla tipologia di pacco, costo, dimensioni, date di ordine e di arrivo e un Identificativo.
Il sistema di Tracking lega all'Identificativo del pacco una DataOra come chiave in modo da costruire uno storico degli aggiornamenti di Status e Posizione per ogni pacco che manterrà come informazione l'ultimo aggiornamento del Tracking.
Questa base di dati è progettata per garantire un'archiviazione efficiente e strutturata delle informazioni, migliorando il recupero e l'analisi dei dati. L'organizzazione sistematica delle informazioni contribuisce a un utilizzo più efficace delle risorse, ottimizzando la gestione delle strutture di tracking e supportando i processi decisionali.

2 Analisi dei Requisiti

Questa sezione riassume i requisiti a cui deve sottostare la base di dati.
Pacco. Ogni Pacco è identificato da un codice alfanumerico univoco e contiene le seguenti informazioni:

- **Id** univoco per l'identificazione di un Pacco
- **Tipologia** di prodotto/pacco spedito (per esempio: elettronica, sanitario, gastronomia ...)
- **Dimensioni**
- **Costo**
- **DataOrdine** relativa alla creazione dell'ordine
- **DataPrevista** relativa alla prevista

I pacchi possono essere in: Bundle, Pacchi Assicurati e Pacchi Regalo.

Bundle. Oltre alle informazioni generali, per i bundle

-

IDEA:: la relazione bundle pacco diventa tabella, bundle infatti legherà ad un codice speciale, gli **Id** dei pacchi che quel bando comprende.

Pacchi Assicurati. Oltre alle informazioni generali, per i Pacchi assicurati si aggiunge:

- **Tipo Assicurazione** tra una scelta limitata di tipologie:
  - Reso in 30 giorni
  - Garanzia per un Anno
  - Garanzia per 3 Anni

Pacchi Regalo. Oltre alle informazioni generali, per i Pacchi Regalo si aggiunge:

- **Tipo Incarto** tra una scelta limitata di tipologie:
  - Carta regalo
  - Sacchetto in tela
  - Scatola Speciale
- **Dedica**

Cliente. Ogni Cliente è identificato da una email, come nelle registrazioni ai siti (si è preferito questo rispetto a un parametro Id aggiuntivo) e contiene le seguenti informazioni:

- **Email** univoco per l'identificazione di un cliente
- **Nome** e un **Cognome**
- **Indirizzo** necessario per la spedizione
- **Telefono** presente o meno

Corriere. Ogni Corriere è identificato da un C.F. che lo identifica come lavoratore e contiene le seguenti informazioni:

- **CF** univoco per l'identificazione di un Corriere
- **Agenzia** per cui lavora

- **Grado**, inteso come una sorta di ranking o gerarchia (ispirato da Death Stranding by Hideo Kojima)
- **Mezzo** di consegna
- **Disponibilità** a fare nuove consegne o meno
- **PacchiAttivi** ovvero il numero tutti i pacchi associati ad un corriere e non in stato *"Consegnato"*

Filiale. Ogni Filiale è identificata da un Nome e una città che lo identificano come sede e contiene le seguenti informazioni:

- **Nome** e **Città** univoco per l'identificazione di una filiale (ispirato ai locker Amazon per il ritiro e a Death Stranding)
- **Indirizzo** precisa la posizione
- **Tipo** distingue tra:
  - Locker
  - Punto di Controllo
  - Magazzino

Tracking. Ogni Tracking rappresenta lo storico di un pacco. È identificato dal **Id** del pacco tracciato e dalla **DataOra** del tracciamento che lo identifica come storico e contiene le seguenti informazioni:

- **Id** e **DataOra** univoco per l'identificazione di un Tracking System
- **Note** per specifiche considerazioni, se necessarie
- **Status** distingue tra:
  - Consegnato
  - In Consegna
  - Spedito
  - Autorizzato
  - Fase di Controllo

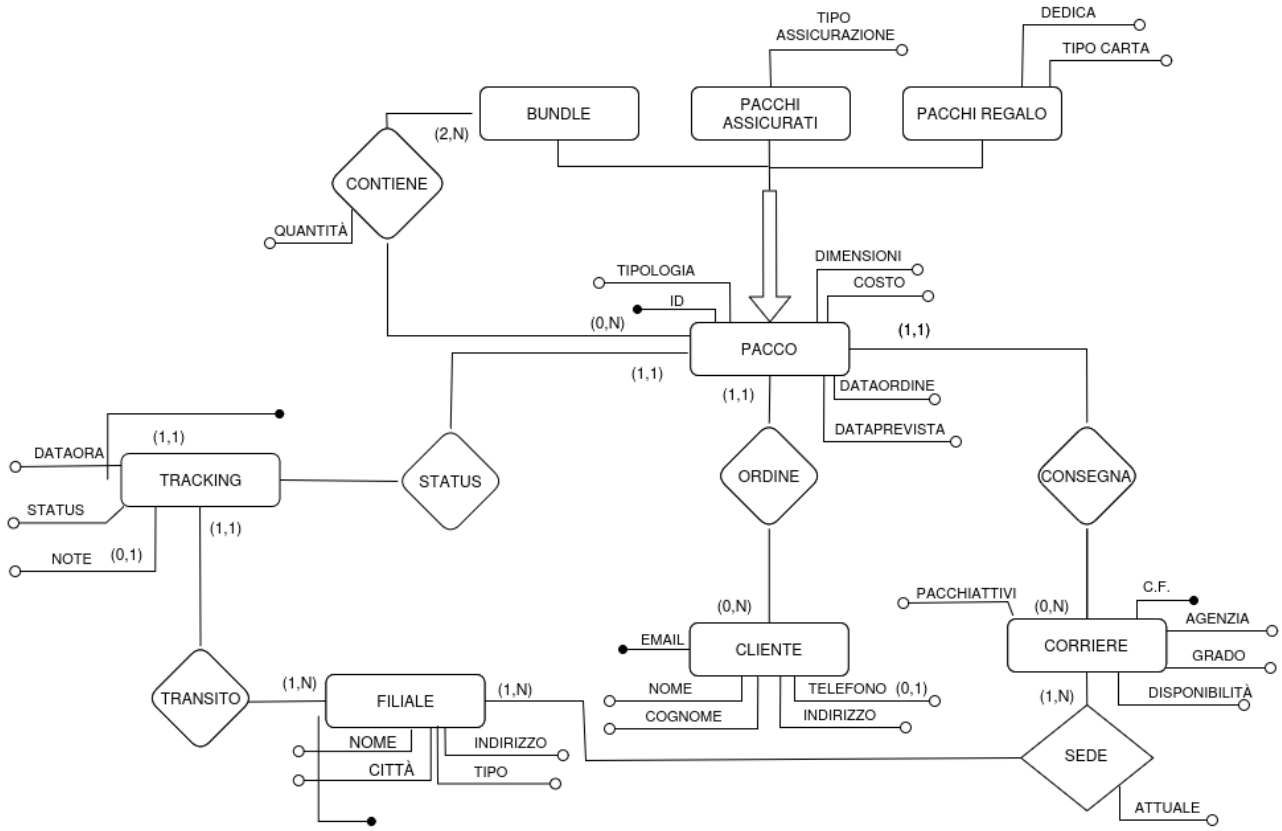

Figure 1: Diagramma ER della Base di Dati relativa al sistema di spedizioni

3 Progettazione Concettuale

Il Diagramma 1 riassume i requisiti della sezione 2.

Esistono 3 tipi di Pacchi: Bundle, Pacchi Assicurati, Pacchi Regalo. Bundle risulta particolarmente singolare in quanto implementa una struttura ricorsiva. Per natura infatti un bundle è un insieme di elementi, nel caso specifico di pacchi.

Necessita particolari attenzioni anche l'entità 'Tracking'. Questa rappresenta sostanzialmente uno storico degli aggiornamenti nel tempo delle informazioni di status e posizione di un pacco. Per ogni DataOra di un determinato Pacco viene infatti associato uno Status, una Posizione presso le filiali dell'azienda di spedizione e eventualmente anche una Nota come descritto sopra.

La relazione Sede indica rispettivamente la sede di appartenza di un Corriere, specificando se si tratti della sede attuale o passata attraverso l'attributo booleano ATTUALE. Si tratta quindi di una relazione molti a molti che rappresenta lo storico delle sedi a cui è appartenuto un Corriere. Trattandosi di una relazione N a N non è comparsa nell'Analisi dei requisiti e verrà gestita con la creazione di una tabella SEDE.

4 Progettazione Logica

In questa sezione viene illustrato il processo di "traduzione" dello schema concettuale in uno schema logico, con l'obiettivo di rappresentare i dati in modo preciso ed efficiente. Il primo passo consiste nell'analizzare le eventuali ridondanze nel modello, al fine di ottimizzare la struttura complessiva. Successivamente, si procede con l'eliminazione delle due generalizzazioni. Infine, viene presentato il diagramma ristrutturato, con una descrizione delle modifiche apportate.

4.1 Analisi Rindondanze

L'attributo PACCHIATTIVI in CORRIERE esprime il numero di pacchi non ancora consegnati che sono associati a ogni corriere. Ciò corrisponde a una ridondanza dato che si tratta di un numero calcolabile contando, per ogni corriere, il numero di pacchi il cui corrispettivo TRACKING ha l'attributo STATUS diverso da *"Consegnato"*. Questo attributo viene modificato ogni volta che un pacco viene consegnato o associato al corriere. Viene invece visualizzato ogni mezz'ora per monitorare l'operato di ogni corriere. Il numero di pacchi associati al giorno per un corriere è in media di 120[1].
Questo si riassume nelle due seguenti operazioni:

- **Operazione 1 (120 volte al giorno)**: memorizza il numero di pacchi non consegnati associati ad ogni corriere.
- **Operazione 2 (48 volte al giorno)**: visualizza il numero di pacchi non consegnati associati ad ogni corriere

[1]Fonte di riferimento per i "120 pacchi al giorno". 2019

Entità	Descrizione	Attributi	Identificatore
Filiale	Filiale di un'azienda di spedizione	Nome, Città, Tipo, Indirizzo	Nome, Città
Tracking	Storico del Tracking di un pacco	DataOra, IdPacco, Status, Note, NomeCheckPoint, CittàCheckPoint	IdPacco, DataOra
Pacco	Elemento spedito	Id, Tipologia, Dimensioni, Costo, DataOrdine, DataSpedizione	Id
Pacchi Assicurati	Pacco Assicurato	TipoAssicurazione	—
Pacchi Regalo	Pacco Regalo	Dedica, TipoCarta	—
Cliente	Proprietario del Pacco	Email, Nome, Cognome, Indirizzo, Telefono	Email
Corriere	Addetto alla spedizione di un pacco	C.F., Grado, Disponibilità, PacchiAttivi	C.F.

Table 1: Tabella delle Entità

Relazione	Descrizione	Componente	Attributi
Sede	Sede di un Corriere	Corriere, Filiale	Attuale
Transito	Posizione Pacco	Tracking, Filiale	
Status	Informazioni sul tracciamento del pacco	Pacco, Tracking	
Contiene	Composizione dei Bundle	Pacco,Bundle	Quantità
Ordine	Ordine di un pacco di un cliente	Cliente, Pacco	
Consegna	Consegna di un pacco da parte di un Corriere	Corriere, Pacco	

Table 2: Tabella delle Relazioni

Assumendo i seguenti volumi nella base di dati:

Concetto	Costrutto	Volume
Corriere	E	1000
Tracking	E	120 000
Pacco	E	120 000
Consegna	R	120 000
Status	R	120 000

Table 3: Tabella dei Volume

la seguente analisi serve per stabilire se sia utile o meno tenere l'attributo ridondante PACCHIATTIVI in CORRIERE.

Con Ridondanza: Analizziamo prima il costo totale con ridondanza.

Operazione 1:

Concetto	Costrutto	Accessi	Tipo	Accessi al giorno
Tracking	E	1	S	120
Pacco	E	1	S	120
Status	R	1	S	120
Corriere	E	1	S	120

Table 4: Tabella dei Volume

Operazione 2:

Concetto	Costrutto	Accessi	Tipo	Accessi al giorno
Corriere	E	1	L	48

Table 5: Tabella dei Volume

Assumendo costo doppio per gli accessi in scrittura rispetto a quelli in lettura, il costo totale con ridondanza è: Costo totale con ridondanza = 120*4*2+48*1= 1080+48=1128

Senza Ridondanza: Analizziamo prima il costo totale senza ridondanza.

- **Operazione 1:**

Concetto	Costrutto	Accessi	Tipo	Accessi al giorno
Tracking	E	1	S	120
Pacco	E	1	S	120
Status	R	1	S	120

Table 6: Tabella dei Volume

- **Operazione 2(Con circa 120 000/1000=120 pacchi consegnati al giorno):**

Concetto	Costrutto	Accessi	Tipo	Accessi al giorno
Corriere	E	1	L	x48
Pacco	E	120	L	x48
Consegna	R	120	L	x48

Table 7: Tabella dei Volume

Assumendo costo doppio per gli accessi in scrittura rispetto a quelli in lettura, il costo totale senza ridondanza è: Costo totale senza ridondanza = 120*3*2+48*1+120*48*2= 720+48+11520=12288

Conclusione: Il costo totale con ridondanza è 1128, mentre il costo totale senza ridondanza è 12288.

L'analisi suggerisce dunque di mantenere l'attributo ridondante PACCHIATTIVI in CORRIERE, in quanto il costo totale con ridondanza è significativamente inferiore rispetto al costo totale senza ridondanza, rendendo così gli accessi ottimizati.
Inoltre, l'attributo ridondante PACCHIATTIVI in CORRIERE è utile per monitorare l'operato di ogni corriere e per ottimizzare le operazioni di consegna.

4.2 Eliminazione delle Generalizzazioni

Le generalizzazioni descritte in Sezione 3 vengono eliminate attraverso una ristrutturazione dello schema concettuale, con l'obiettivo di semplificare la successiva implementazione del modello relazionale e ridurre la presenza di valori nulli. Le modifiche vengono applicate come segue:

PACCHI: La generalizzazione parziale di PACCHI viene rimossa tramite tre operazioni distinte che si occupano delle tre entità figlie, realizzando dunque una soluzione "ibrida".
Per PACCHI ASSICURATI e per PACCHI REGALO si è deciso di sostituire la generalizzazione con l'accorpamento delle due entità figlie nel padre. Le due entità figlie verranno dunque eliminate e la loro funzione sarà sostituita da degli attributi che saranno attribuiti all'entità padre. Questa scelta si giustifica di fronte al basso livello di "profondità" e "annidamento" degli attributi delle entità figlie in questione. Infatti queste entità non hanno relazioni con altre entità che non siano quella padre. Ne consegue dunque che l'introduzione di ulteriori attributi nel padre produce una quantità di valori NULL limitata, che non produce un effetto a cascata su ulteriori entità. Infatti in questo caso la presenza di valori NULL negli attributi sostituivi sarà dunque limitata a soli tre attributi.
L'entità figlia BUNDLE dispone già di per sé di una relazione con l'entità padre e merita dunque un trattamento differente. Il vincolo originario che un PACCO ASSICURATO, non possa essere anche PACCO REGALO e viceversa verrà quindi mantenuto grazie alla creazione di check che controllano che gli stati dei parametri nel padre siano coerenti: se sono *not null* alcuni lo dovranno essere gli altri.
Incorporarla direttamente nella relazione padre renderebbe la sua gestione particolarmente complessa, con la necessità dell'aggiunta di numerosi attributi per mantenere l'insieme delle caratteristiche originali. Questo corrisponderebbe tra l'altro a un numero rilevante di attributi che si troverebbero poi molto spesso a essere messi a NULL (nel caso di un non-BUNDLE). Un'altra opzione consisterebbe nell'aggiungere un'altra relazione tra BUNDLE e PACCHI, riducendo in questo caso il numero di potenziali NULL. In questo caso però si avrebbero però

due relazioni e due entità per gestire una struttura che corrisponde a una semplice entità con una relazione ricorsiva con un attributo. Abbiamo dunque deciso di optare per quest'ultima operazione di traduzione che comporta una notevole semplificazione e una gestione efficiente delle risorse disponibili.

Sono inoltre necessari due ulteriori controlli del vincolo nella tabella BUNDLE e in TRACKING.In BUNDLE per evitare che un pacco possa contenere se stesso, quindi *IdBUndle* e *IdContenuto* non possono essere uguali.

In TRACKING invece è necessario che tutte le *DataOra* siano successive alla *DataOra* di creazione del pacco.

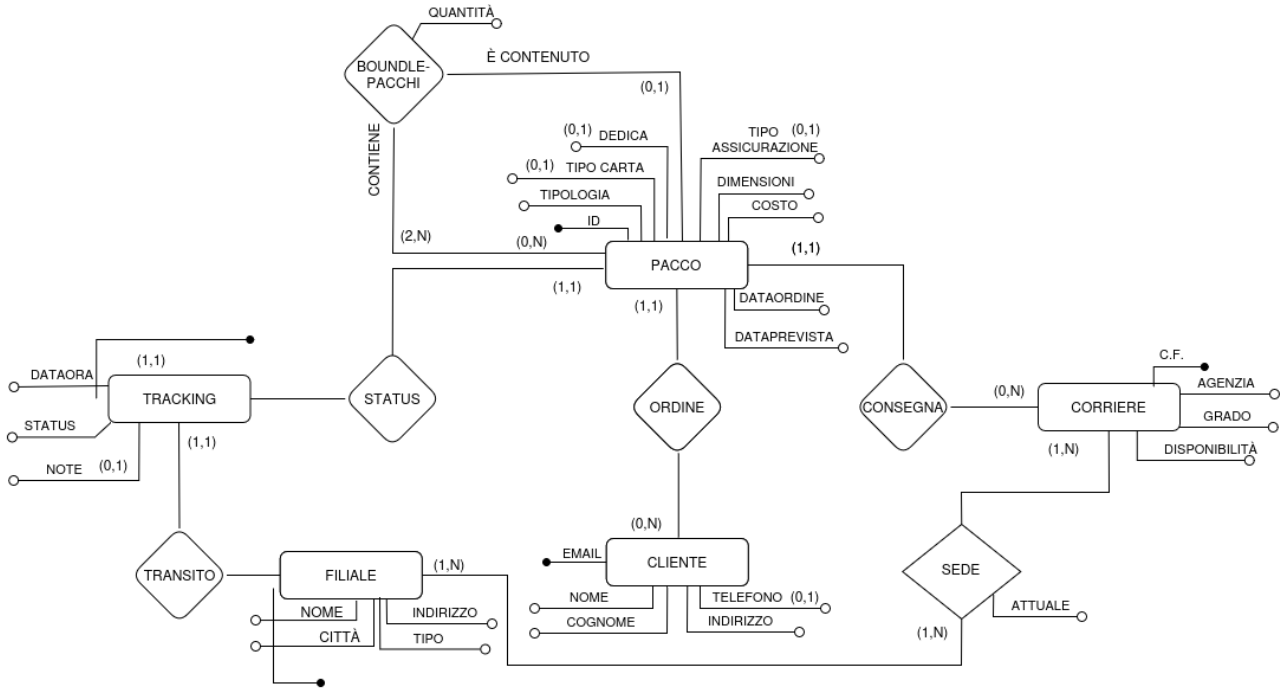

Figure 2: Diagramma della Base di Dati dopo l'eliminazione delle Generalizzazioni

4.3 Schema Relazionale

Lo schema ristrutturato in Figura **??** contiene solamente costrutti mappabili in corrispettivi dello schema relazionale detto anche schema logico. Lo schema logico è rappresentato a seguire, dove l'asterisco dopo il nome degli attributi indica quelli che ammettono valori nulli.

- **Pacco**(ID, Tipologia, Dimensioni, Costo, DataOraOrdine, DataOraPrevista, Cliente, Corriere, TipoAssicurazione*, Dedica*, TipoCarta*)
  - Pacco.Cliente → Cliente.Email
  - Pacco.Corriere → Corriere.C.F.

- **Bundle-Pacchi**(IdBundle, IdContenuto, Quantità)
  - Bundle.IdBundle → Pacco.Id
  - Bundle.IdContenuto → Pacco.Id
- **Corriere**(C.F., Agenzia, Grado, Disponibilità, Pacchi Attivi)
- **Filiale**(Nome, Città, Tipo, Indirizzo)
- **Tracking**(IdPacco, DataOra, Status, Note*, NomeCheckPoint, CittàCheckPoint)
  - Tracking.IdPacco → Pacco.Id
  - Tracking.(NomeCheckPoint, CittàCheckPoint) → Filiale.(Nome, Città)
- **Cliente**(Email, Nome, Cognome, Indirizzo, Telefono*)
- **Sede**(Corriere, SedeNome, SedeCittà, Attuale)
  - Sede.Corriere → Corriere.C.F.
  - Sede.(SedeNome, SedeCittà) → Filiale.(Nome, Città)

5 Implementazione in PostgreSQL e Definizione delle Query

5.1 Definizione delle Query

1. **QUERY 1** Media del costo dei pacchi di un corriere di un certo grado che sono passati per una filiale di un certo tipo

```sql
SELECT    corriere, AVG(p.costo) AS costo_medio
FROM    pacco AS p
JOIN    tracking AS t ON t.id_pacco = p.id
JOIN    filiale AS f ON t.città_check_point = f.città AND t.nome_check_point=f.nome
JOIN    corriere AS c ON p.corriere = c.cf
WHERE    f.tipo = 'Locker'    AND c.grado = 'Porter'
GROUP BY   corriere;
```

2. **QUERY 2** Numero pacchi per bundle con garanzia 3 anni

```sql
SELECT b.id_bundle, COUNT(*) as n_garanzia_tre_anni
FROM bundle AS b
JOIN pacco AS p ON b.id_contenuto = p.id
WHERE p.tipo_assicurazione = 'Garanzia per 3 anni'
GROUP BY b.id_bundle
```

3. **QUERY 3** Corrieri che hanno un tempo di consegna medio dei loro pacchi superiore al tempo medio di consegna di un pacco:

```sql
SELECT
    p.corriere,
    AVG(h.differenza) AS media_corriere
FROM (
    SELECT
        t1.id_pacco,
        p1.corriere,
        MAX(t1.data_ora) - p1.dataora_ordine AS differenza
    FROM
        tracking AS t1
    JOIN
        pacco AS p1 ON p1.id = t1.id_pacco
    WHERE t1.status = 'Consegnato'
    GROUP BY
        t1.id_pacco, p1.dataora_ordine, p1.corriere
) AS h
JOIN pacco AS p ON p.id = h.id_pacco
GROUP BY
    p.corriere
HAVING AVG(h.differenza) > (
    SELECT AVG(differenza)
    FROM (
        SELECT
            t1.id_pacco,
            MAX(t1.data_ora) - p2.dataora_ordine AS differenza
        FROM
            tracking AS t1
        JOIN
            pacco AS p2 ON p2.id = t1.id_pacco
        WHERE t1.status = 'Consegnato'
        GROUP BY
            t1.id_pacco, p2.dataora_ordine
    ) AS m
) ORDER BY media_corriere DESC;
```

4. **QUERY 4** Corrieri con bundle con pacchi con garanzia a 3 anni e costo maggiore di 300

```sql
SELECT    corriere, AVG(p.costo) AS costo_medio
FROM    pacco AS p
JOIN    tracking AS t ON t.id_pacco = p.id
JOIN    filiale AS f ON t.città_check_point = f.città AND t.nome_check_point=f.nome
JOIN    corriere AS c ON p.corriere = c.cf
WHERE    f.tipo = 'Locker'    AND c.grado = 'Porter'
GROUP BY   corriere;
```

5. **QUERY 5** Ultimo aggiornamento del tracking di un pacco

```sql
SELECT t1.id_pacco, t1.status
FROM tracking t1
INNER JOIN (
    SELECT id_pacco, MAX(data_ora) as max_data_ora
    FROM tracking
    GROUP BY id_pacco
) t2 ON t1.id_pacco = t2.id_pacco AND t1.data_ora = t2.max_data_ora
```

5.2 Creazione degli Indici

Supponendo di voler ottimizzare la query num 5, si deve considerare:

- Condizione del JOIN: `t.id_pacco = p.id`
- Condizione del JOIN: `t.città_check_point = f.città AND t.nome_check_point = f.nome`
- Condizione del JOIN: `p.corriere = c.cf`
- Condizione del WHERE: `f.tipo = 'Locker'AND c.grado = 'Porter'`
- GROUP BY sulle colonne: `corriere`

Per il punto 1 è opportuno creare due indici hash su entrambe le colonne del join:

```sql
CREATE INDEX idx_id_hash ON pacco USING HASH (id);
CREATE INDEX idx_id_pacco ON tracking USING HASH (id_pacco);
```

Per il punto 2 è opportuno creare quattro indici hash per tutte le condizioni del join:

```
CREATE INDEX idx_città_check_point ON tracking USING HASH (città_check_point);
CREATE INDEX idx_nome_check_point ON tracking USING HASH (nome_check_point);
CREATE INDEX idx_nome  ON filiale USING HASH (nome);
CREATE INDEX idx_città  ON filiale USING HASH (città );
```

Per il punto 3 è opportuno creare due indici hash su entrambe le colonne del join:

```
CREATE INDEX idx_corriere  ON pacco USING HASH (corriere);
CREATE INDEX idx_cf  ON corriere USING HASH (cf);
```

Per il punto 4 è opportuno creare due indici per i due attributi rispetto a cui il WHERE valuta la sua condizione:

```
CREATE INDEX idx_tipo ON filiale USING HASH (tipo);
CREATE INDEX idx_grado ON corriere USING HASH (grado);
```

Il punto 5 è un GROUP BY e potrebbe quindi essere ottimizzato tramite il seguente indice B+Tree:

```
CREATE INDEX pacco_pkey_corriere ON pacco USING BTREE (corriere);
```

Visto che non si tratta di un attributo che è chiave primaria di una entità POSTGRE non ha già creato l'indice corrispondente.

6 Applicazione Software

Il codice code.c permette di connettersi al database PostGres su Supabase, scelto perchè permette di lavorare in condivisione.
Una volta effettuata la connessione, il programma permette di eseguire 5 query diverse, a scelta dell'utente.
Tre delle 5 query sono state scritte in modo da essere parametrizzate, in modo da evitare SQL injection, questo grazie agli *enum* che restringono la scelta dei parametri a solo quelli possibili.

- Media del costo dei pacchi di un corriere di un certo grado che sono passati per una filiale di un certo tipo

  - Parametrizzabile per il tipo di filiale e grado del corriere.

- Numero pacchi per bundle con un determinato tipo di assicurazione
  - Parametrizzabile per il tipo di assicurazione.
- Corrieri con tempi di consegna maggiori alla media.
- Corrieri con bundle con pacchi assicurati o regalo e costo specifici.
  - Parametrizzabile per tutti i valori *Assicurati e Regalo* e per il Costo.
- Status attuale dei pacchi