

**TEMPLATE
PROJECT WORK**

Corso di Studio	INFORMATICA PER LE AZIENDE DIGITALI (L-31)
Dimensione dell'elaborato	Minimo 6.000 – Massimo 10.000 parole <i>(pari a circa Minimo 12 – Massimo 20 pagine)</i>
Formato del file da caricare in piattaforma	PDF
Nome e Cognome	Samantha De Santis
Numero di matricola	0312401717
Tema n. (Indicare il numero del tema scelto):	1
Titolo del tema (Indicare il titolo del tema scelto):	La digitalizzazione dell'impresa
Traccia del PW n. (Indicare il numero della traccia scelta):	4
Titolo della traccia (Indicare il titolo della traccia scelta):	Progettazione dello schema di persistenza dei dati a supporto dei servizi di un'azienda nel settore dei trasporti
Titolo dell'elaborato (Attribuire un titolo al proprio elaborato progettuale):	Progettazione di un modello di persistenza dei dati per la gestione delle prenotazioni aeree: il caso Luxair.

PARTE PRIMA – DESCRIZIONE DEL PROCESSO

Utilizzo delle conoscenze e abilità derivate dal percorso di studio

(Descrivere quali conoscenze e abilità apprese durante il percorso di studio sono state utilizzate per la redazione dell'elaborato, facendo eventualmente riferimento agli insegnamenti che hanno contribuito a maturarle):

Il mio percorso di studi in Informatica e Tecnologie per lo Sviluppo del Software è iniziato presso il Politecnico di Bari, dove ho frequentato i primi due anni del corso di studi. Successivamente, per conciliare meglio lavoro e studio, ho scelto di proseguire il mio percorso con l'Università Telematica Pegaso. Questa combinazione di esperienze mi ha permesso di consolidare e ampliare in modo significativo le mie competenze nella progettazione e nello sviluppo del software.

Lo sviluppo del Project Work, incentrato sulla progettazione di uno schema di persistenza dei dati per una compagnia aerea (nel mio caso "Luxair"), si basa in particolare sulle solide basi teoriche e pratiche acquisite nel corso "Basi di Dati". Nonostante avessi già seguito un corso analogo al Politecnico di Bari, è stato solo grazie al corso del Professor Sciarrone che ho compreso a fondo l'importanza della progettazione concettuale e logica. In particolare, ho imparato ad utilizzare il Modello Entità-Relazione (E-R) per la modellazione concettuale e il Modello Relazionale per la successiva progettazione logica del database.

Un contributo altrettanto significativo è derivato dai corsi “Modelli e Metodi per la Qualità del Software” e “Ingegneria del Software”.

Il primo mi ha permesso di sviluppare un approccio metodologico alla soluzione delle problematiche progettuali, poiché l'esame prevedeva attività di debugging su un software complesso. Il secondo, invece, focalizzato sull'utilizzo delle metodologie Agile, mi ha insegnato l'importanza cruciale del rispetto delle tempistiche di progetto, grazie all'attenzione posta sulla gestione iterativa e la consegna continua.

Fasi di lavoro e relativi tempi di implementazione per la predisposizione dell'elaborato

(Descrivere le attività svolte in corrispondenza di ciascuna fase di redazione dell'elaborato. Indicare il tempo dedicato alla realizzazione di ciascuna fase, le difficoltà incontrate e come sono state superate):

1. Analisi della traccia e scelta della compagnia (4 giorni).

All'inizio del lavoro ho dedicato i primi giorni a comprendere appieno la traccia e a scegliere la compagnia aerea Luxair come oggetto dell'elaborato. Questa fase ha richiesto attenzione sia per quanto concerne gli aspetti operativi reali della compagnia sia sulla considerazione della gestione dei dati. È stato necessario studiare il funzionamento dei voli, dei biglietti, degli scali e dei servizi offerti ai passeggeri per capire quali informazioni fossero effettivamente rilevanti da rappresentare nel database.

La difficoltà principale è stata tradurre processi aziendali complessi in concetti chiari e modellabili. Ho superato questa difficoltà consultando documentazione ufficiale, siti del settore aereo e guide sulle operazioni di prenotazione e check-in, così da ottenere una visione completa del contesto operativo.

2. Modellazione concettuale ER (6 giorni)

Successivamente mi sono concentrata sullo sviluppo del diagramma ER. Esso rappresenta le principali entità, i relativi attributi e le relazioni che ne intercorrono. In questa fase ho dedicato particolare attenzione alla definizione delle entità, alla scelta degli attributi obbligatori e facoltativi e alla corretta modellazione delle relazioni.

Ho impiegato circa sei giorni, procedendo per successive iterazioni: modellavo il diagramma, lo confrontavo con i requisiti operativi e lo corregevo per risolvere ambiguità o incongruenze. È stato impegnativo analizzare i legami tra le entità con una visione logica, cercando di immedesimarmi nelle diverse procedure, ove possibile, così da comprenderne meglio il significato. In questo passaggio, sono partita da bozze su carta per poi passare all'utilizzo di un software che mi ha permesso la realizzazione definitiva del diagramma.

3. Traduzione in modello relazionale (8 giorni)

Una volta definito il diagramma ER, ho iniziato la traduzione dello stesso in un modello relazionale. Questa fase ha richiesto particolare attenzione, poiché ha comportato la creazione di tutte le tabelle, la definizione delle chiavi primarie ed esterne, la scelta dei vincoli di integrità e la selezione dei tipi di dati più adatti per ciascun attributo. Ho dovuto inoltre gestire relazioni complesse, come la tabella *FlightCrew*, che rappresenta una relazione molti-a-molti tra *Volo* e *PersonaleDiBordo*, assicurandomi che le cardinalità fossero coerente con il diagramma ER. Ho impiegato circa otto giorni per questa fase, poiché non si è trattato solo di scrivere codice SQL, ma anche di valutare attentamente le implicazioni di ogni scelta sul funzionamento reale del database.

4. Normalizzazione e denormalizzazione (5 giorni)

In questa fase ho applicato le forme normali fino alla terza forma normale (3NF) per eliminare ridondanze e garantire la coerenza dei dati. Alcune tabelle, come *CheckIn* e *Biglietto*, sono state leggermente denormalizzate per rendere più agevole l'accesso a informazioni cruciali, come il numero del posto assegnato. Ho dedicato cinque giorni a questa attività, analizzando con attenzione quali attributi spostare o duplicare, valutando il compromesso tra integrità dei dati e praticità d'uso.

5. Ottimizzazione e creazione degli indici (2 giorni)

Ho poi introdotto indici sui campi più utilizzati nelle interrogazioni e nelle ricerche frequenti, come l'e-mail dei passeggeri, il PNR dei biglietti, il numero del volo e la data di partenza. Questa fase ha richiesto circa due giorni, perché ho dovuto valutare attentamente quali indici fossero realmente utili al miglioramento delle prestazioni, evitando quindi di appesantire inutilmente il database.

6. Test e verifica della coerenza (3 giorni)

Dopo aver implementato le tabelle e gli indici, ho eseguito test di coerenza tra le relazioni, verificando il corretto funzionamento dei vincoli di chiave esterna e l'inserimento dei dati simulati. Ho dedicato circa tre giorni a questa fase, correggendo piccoli errori di sintassi, affinando i vincoli e controllando che il modello rispecchiasse fedelmente il diagramma ER concettuale.

7. Valutazione dei campi di applicazione e analisi dei pro e contro (2 giorni)

Infine, ho riflettuto sull'uso del database nel contesto operativo della compagnia aerea. Ho analizzato i principali vantaggi, tra cui la chiarezza della struttura, la gestione integrata dei dati relativi a passeggeri, dei voli e dei biglietti e la possibilità di effettuare query complesse. Ho inoltre considerato criticità come la necessità di aggiornamenti costanti in tempo reale e la gestione delle relazioni multi-a-molti tra voli e personale. Questa fase ha richiesto due giorni di analisi e documentazione.

Risorse e strumenti impiegati

(Descrivere quali risorse - bibliografia, banche dati, ecc. - e strumenti - software, modelli teorici, ecc. - sono stati individuati ed utilizzati per la redazione dell'elaborato. Descrivere, inoltre, i motivi che hanno orientato la scelta delle risorse e degli strumenti, la modalità di individuazione e reperimento delle risorse e degli strumenti, le eventuali difficoltà affrontate nell'individuazione e nell'utilizzo di risorse e strumenti ed il modo in cui sono state superate):

Per la redazione del presente Project Work ho individuato e utilizzato diverse risorse teoriche, strumenti software e riferimenti tecnici, scelti in funzione degli obiettivi dell'elaborato e delle esigenze legate alla progettazione di un modello di persistenza dei dati per il settore del trasporto aereo.

Dal punto di vista teorico, il lavoro si è basato principalmente sulle conoscenze acquisite nel corso di "Basi di Dati", che hanno rappresentato il quadro concettuale di riferimento per l'intero progetto. In particolare, ho applicato i principi relativi alla modellazione concettuale tramite il Modello Entità-Relazione (E-R), alla traduzione in modello relazionale, all'uso delle forme normali fino alla terza forma normale (3NF) e alla definizione dei vincoli di integrità. Questi modelli teorici hanno guidato in modo sistematico tutte le scelte progettuali, dalla definizione delle entità fino all'ottimizzazione delle interrogazioni.

Ho reperito ulteriori risorse informative attraverso la consultazione di documentazione ufficiale e fonti online del settore aeronautico, utilizzate principalmente nella fase iniziale di analisi del contesto operativo. In particolare, ho consultato **siti istituzionali e materiali informativi relativi alle compagnie aeree** per comprendere il funzionamento dei processi di prenotazione, vendita dei biglietti,

gestione dei voli, check-in e assegnazione dei posti. Queste risorse si sono rivelate fondamentali per tradurre i processi reali della compagnia Luxair in una rappresentazione strutturata e coerente all'interno del database.

Un ulteriore elemento di supporto alla fase di analisi è derivato dalla mia esperienza diretta come utente dei servizi Luxair. Vivendo tra l'Italia e il Lussemburgo, ho avuto modo nel tempo di navigare frequentemente il **sito web della compagnia aerea** (www.luxair.lu) per esigenze personali di viaggio. Tuttavia, nel contesto di questo Project Work, la consultazione del sito è avvenuta secondo una prospettiva completamente nuova, orientata non più all'uso del servizio, ma allo studio strutturato dei processi sottostanti.

Inoltre, ho consultato il **sito dell'aeroporto di Lussemburgo** (www.lux-airport.lu) per comprendere meglio il ruolo degli scali, le operazioni aeroportuali e l'organizzazione dei flussi di partenza e arrivo. Queste fonti mi hanno permesso di osservare il funzionamento reale dei servizi e di raccogliere informazioni concrete utili a modellare in modo realistico le entità e le relazioni del database, garantendo una maggiore aderenza del modello progettato ai processi operativi effettivi della compagnia.

Per quanto riguarda gli strumenti software, nella fase di progettazione concettuale sono partita da bozze su carta, utili per una prima definizione delle entità e delle relazioni. Successivamente sono passata all'utilizzo del software **draw.io** per la realizzazione definitiva del diagramma E-R. La scelta di draw.io è stata motivata dalla sua semplicità d'uso, dalla flessibilità nella gestione degli elementi grafici e dal supporto a notazioni standard per la modellazione dei dati. L'impiego di questo software mi ha consentito di rappresentare in modo chiaro e ordinato le interdipendenze tra le entità, di gestire correttamente le cardinalità e di verificare la coerenza complessiva del modello concettuale.

Nella fase di progettazione logica e di implementazione ho utilizzato **PostgreSQL** come Database Management System. La scelta di questo DBMS è stata motivata dalle sue caratteristiche di affidabilità, dal pieno supporto alle transazioni ACID e dalla capacità di gestire relazioni complesse e vincoli di integrità referenziale, aspetti essenziali per un sistema di gestione delle prenotazioni aeree. PostgreSQL mi ha inoltre consentito di definire indici mirati, testare la correttezza dello schema relazionale e valutare le performance delle interrogazioni più frequenti.

Le modalità di individuazione delle risorse e degli strumenti sono state prevalentemente autonome e guidate sia dai contenuti del corso sia dalle esigenze specifiche del progetto. Ho selezionato le risorse teoriche in base alla loro affidabilità e coerenza con il programma di studi, mentre gli strumenti software sono stati scelti per la loro adeguatezza agli obiettivi di modellazione, per la semplicità d'uso e per la compatibilità con il linguaggio SQL adottato.

Durante l'utilizzo delle risorse e degli strumenti non ho riscontrato particolari difficoltà di natura tecnica. Le principali complessità hanno riguardato aspetti concettuali, come la rappresentazione delle relazioni molti-a-molti, la gestione delle denormalizzazioni mirate e la scelta degli indici più appropriati. Tali difficoltà sono state superate attraverso un processo iterativo di revisione del modello, il confronto tra diverse soluzioni progettuali e una verifica costante della coerenza tra modello concettuale, modello relazionale e requisiti operativi del sistema.

Nel complesso, l'insieme delle risorse teoriche e degli strumenti utilizzati mi ha permesso di sviluppare un elaborato strutturato, coerente e metodologicamente solido, garantendo un corretto allineamento tra teoria, progettazione e applicazione pratica.

PARTE SECONDA – PREDISPOSIZIONE DELL’ELABORATO

Obiettivi del progetto

(Descrivere gli obiettivi raggiunti dall’elaborato, indicando in che modo esso risponde a quanto richiesto dalla traccia):

L’obiettivo principale del mio elaborato è stato quello di progettare un modello di persistenza dei dati in grado di supportare efficacemente il processo di gestione e vendita dei biglietti per la compagnia aerea Luxair. Lo scopo è stato tradurre l’intero flusso operativo del servizio in una rappresentazione strutturata, coerente e scalabile, garantendo integrità dei dati, efficienza e possibilità di evoluzione futura del sistema informativo.

Il progetto risponde pienamente alla traccia proposta, in quanto affronta i principali aspetti richiesti. Ho analizzato il contesto operativo della compagnia, approfondendo i processi legati alla vendita dei biglietti, alla gestione delle tratte, agli scali e alle operazioni di check-in. Su questa base, ho elaborato un modello concettuale in grado di rappresentare in modo chiaro e organico il servizio offerto, identificando le principali entità e le relazioni esistenti tra di esse.

Successivamente, il modello concettuale è stato tradotto in un modello relazionale, attraverso la definizione delle tabelle, delle chiavi primarie ed esterne e dei vincoli di integrità, assicurando così coerenza e correttezza dei dati. Ho applicato le forme normali per ridurre le ridondanze e migliorare la qualità complessiva del database, procedendo, ove necessario, a denormalizzazioni mirate al fine di ottimizzare le prestazioni. Sono stati inoltre definiti indici strategici sui campi più rilevanti, così da rendere efficienti le interrogazioni tipiche del sistema.

L’elaborato consente pertanto di ottenere una rappresentazione funzionale e operativa del servizio, supportando attività concrete quali la gestione delle prenotazioni, la verifica dei biglietti, l’analisi delle tratte e il tracciamento dei passeggeri. Gli obiettivi raggiunti riguardano sia la comprensione dei processi reali della compagnia sia la capacità di tradurli in un sistema di gestione dei dati robusto e ottimizzato, rispondendo in modo completo e coerente a quanto indicato dalla traccia.

Contestualizzazione

(Descrivere il contesto teorico e quello applicativo dell’elaborato realizzato):

Il mio progetto si inserisce nel contesto più ampio della gestione dei sistemi informativi nel settore dei trasporti, con particolare riferimento al trasporto aereo: un ambito caratterizzato da processi complessi, fortemente interconnessi e ad alta intensità di dati. In questo settore, l’affidabilità delle informazioni e la capacità di gestire grandi volumi di dati in modo strutturato rappresentano elementi imprescindibili per il corretto funzionamento dell’organizzazione e per la qualità del servizio offerto agli utenti finali.

Dal punto di vista teorico, la progettazione di un modello di persistenza dei dati costituisce una fase cruciale nello sviluppo di qualsiasi sistema informativo. Un modello ben progettato consente infatti di garantire integrità, coerenza e affidabilità delle informazioni, elementi fondamentali per qualsiasi organizzazione che eroghi servizi complessi e interconnessi. I principi della teoria dei database relazionali, l’applicazione delle forme normali e l’utilizzo consapevole delle strategie di indicizzazione forniscono gli strumenti necessari per strutturare i dati in maniera logica e rigorosa, assicurando una gestione efficiente e scalabile nel tempo. Tali aspetti risultano particolarmente rilevanti in contesti operativi dinamici, nei quali i dati vengono costantemente aggiornati e interrogati da diversi attori e sistemi.

In termini applicativi, il progetto che ho realizzato si colloca nel contesto operativo di una compagnia aerea, Luxair, dove la gestione dei dati non rappresenta soltanto una problematica tecnica, ma assume un ruolo strategico con un impatto diretto sull'efficienza delle operazioni e sulla qualità del servizio offerto ai passeggeri. Una compagnia aerea deve infatti gestire in modo coordinato un insieme eterogeneo di informazioni relative alle prenotazioni, all'emissione e alla validazione dei biglietti, alla pianificazione dei voli e delle tratte, alla gestione degli scali intermedi, all'assegnazione dei posti, alle operazioni di check-in e al controllo dei pagamenti. Ciascuno di questi processi genera dati che devono essere raccolti, memorizzati e resi accessibili in modo affidabile e tempestivo.

Il modello progettato risponde quindi a una duplice esigenza. Da un lato, soddisfa un'esigenza di tipo **teorico**, applicando in modo concreto i principi della progettazione dei database per ottenere un sistema coerente, normalizzato e strutturato secondo le buone pratiche della disciplina. Dall'altro lato, risponde a un'esigenza **applicativa**, fornendo un supporto concreto ai processi operativi della compagnia aerea. La progettazione del modello Entità-Relazione e la sua successiva trasformazione in uno schema relazionale consentono di rappresentare in maniera chiara e immediata la struttura dei dati, le interdipendenze tra le entità coinvolte e le regole di integrità necessarie per mantenere la consistenza delle informazioni nel tempo.

In questo scenario, un sistema di persistenza dei dati progettato in maniera adeguata consente di supportare efficacemente le attività operative quotidiane, garantendo al contempo la possibilità di effettuare interrogazioni complesse, analisi dei dati e attività di reportistica. La disponibilità di informazioni coerenti e facilmente interrogabili risulta fondamentale non solo per il personale operativo, ma anche per le funzioni di controllo e di pianificazione strategica della compagnia.

Il mio progetto tiene inoltre conto delle complessità specifiche del settore del trasporto aereo, come la gestione dei voli con scali intermedi, la presenza di diverse classi di biglietto, le preferenze dei passeggeri e le relazioni molti-a-molti tra voli e personale di bordo. Questi aspetti hanno richiesto una progettazione attenta, in grado di bilanciare rigore teorico e praticità applicativa. La definizione di indici strategici sui campi più utilizzati e l'attenzione alla normalizzazione, affiancata da denormalizzazioni mirate laddove necessario, assicurano che il database possa sostenere un carico operativo realistico, consentendo interrogazioni rapide, affidabili e coerenti con le esigenze del contesto reale.

Descrizione dei principali aspetti progettuali

(Sviluppare l'elaborato richiesto dalla traccia prescelta):

1. Inquadramento settoriale del servizio offerto dalla compagnia aerea Luxair.

Nel corso della mia analisi ho scelto di focalizzarmi sulla compagnia aerea di bandiera del Lussemburgo, in quanto rappresenta un esempio significativo per la progettazione di un sistema di gestione dei dati. Pur essendo un vettore di dimensioni contenute rispetto ai grandi player internazionali, opera su una rete articolata e presenta una struttura organizzativa complessa, che integra il trasporto passeggeri con attività di handling aeroportuale e servizi turistici. Questa combinazione di ambiti operativi ne fa un caso di studio rilevante per l'analisi e la modellazione dei processi informativi.

Il vettore collega il Lussemburgo a oltre 90 destinazioni in più di 30 Paesi, svolgendo un ruolo strategico nell'economia e nella mobilità del Granducato. Rappresenta uno dei principali strumenti di connessione del Paese con il resto d'Europa e con alcune destinazioni extraeuropee, supportando sia il traffico business sia quello turistico. Nel 2023 ha superato i 2,5 milioni di passeggeri trasportati, un risultato indicativo della crescente rilevanza operativa.

Fondata nel **1961** dalla riorganizzazione della “Luxembourg Airlines Company”, ha operato il primo volo commerciale nel marzo 1962 sulla tratta Lussemburgo–Parigi con un Fokker F27, avviando un percorso di sviluppo progressivo che ha portato all’ampliamento della rete di destinazioni, all’integrazione di nuovi aeromobili e alla modernizzazione della flotta.

Dal punto di vista del mercato, può essere considerata un **vettore full-service regionale**, capace di attrarre sia viaggiatori business, interessati a collegamenti frequenti e affidabili, sia turisti, grazie all’offerta di pacchetti di viaggio e servizi accessori. Questo duplice orientamento comporta una gestione articolata dei dati relativi a passeggeri, prenotazioni, classi di servizio e preferenze individuali, aspetti fondamentali per la progettazione del sistema informativo.

Il panorama operativo include la gestione della flotta, le operazioni di handling aeroportuale, la programmazione dei voli con eventuali scali e una strategia commerciale che combina traffico di linea e offerta turistica. Questi elementi rendono il caso scelto ideale per il progetto di modellazione dei dati, poiché tutti i componenti coinvolti (passeggeri, rotte, aeroporti, voli, biglietti e pagamenti) devono essere strutturati in modo coerente e integrato all’interno di un unico sistema di persistenza dei dati.

2. Il processo di servizio analizzato: la vendita e la gestione dei biglietti

Per quanto riguarda il processo di servizio, ho concentrato la mia attenzione sulla vendita e sulla gestione dei biglietti, che rappresentano il cuore dell’attività operativa di Luxair e il fulcro della traccia su cui si basa il Project Work. Questo processo costituisce il punto di incontro tra le esigenze del passeggero e i sistemi informativi della compagnia, rendendo fondamentale una gestione strutturata e coerente dei dati lungo tutte le sue fasi.

Quando un passeggero decide di volare con Luxair, il suo viaggio inizia ben prima di salire a bordo dell’aeromobile. La prima fase è quella della **ricerca del volo**, che avviene generalmente attraverso il sito ufficiale della compagnia o, in alternativa, tramite un’agenzia di viaggi online. In questa fase il passeggero seleziona le rotte disponibili, le date di partenza e arrivo, gli orari e la classe di viaggio desiderata. Il sistema di prenotazione deve quindi verificare in tempo reale la disponibilità dei posti, tenendo conto sia delle caratteristiche del volo diretto sia di eventuali scali o connessioni, elemento particolarmente rilevante considerando la rete articolata su cui opera Luxair.

Una volta individuata la soluzione di viaggio più adatta, il passeggero procede con la fase di **prenotazione**. In questo momento inserisce i propri dati personali, come nome, cognome e contatti, ed eventualmente specifica preferenze aggiuntive, ad esempio bagagli extra, scelta del posto o servizi accessori. Il sistema crea così una prenotazione temporanea, assimilabile a un “carrello”, che rimane valida per un intervallo di tempo limitato. Successivamente entra in gioco la fase di pagamento, che può avvenire tramite diversi metodi, come carta di credito, PayPal, Apple Pay o bonifico bancario.

Al termine del pagamento, il sistema conferma la prenotazione, emette il biglietto elettronico, assegna un codice PNR (Passenger Name Record) e aggiorna lo stato del posto, che viene contrassegnato come venduto o riservato.

Il processo, tuttavia, non si esaurisce con l’emissione del biglietto. In molti casi, infatti, il passeggero può decidere di **modificare la prenotazione**, ad esempio cambiando la data del volo, aggiungendo bagagli, modificando la classe di viaggio o selezionando un volo alternativo. In queste situazioni, il sistema informativo di Luxair deve gestire correttamente vincoli e disponibilità, verificando se le modifiche sono consentite e calcolando eventuali penali o supplementi. Solo dopo queste verifiche il sistema aggiorna in modo coerente la prenotazione e i dati associati.

Avvicinandosi alla data del volo, il passeggero ha la possibilità di effettuare il **check-in**, che può essere svolto online oppure direttamente in aeroporto. Durante questa fase, Luxair registra ufficialmente la presenza del passeggero sul volo, assegna il posto definitivo e genera la carta d'imbarco. Il check-in rappresenta un momento cruciale per la compagnia, poiché consente di gestire in modo efficiente l'imbarco, il controllo dei bagagli, le procedure di sicurezza e la composizione finale della lista dei passeggeri.

Infine, al momento dell'imbarco, il passeggero sale a bordo dell'aeromobile e viene registrato come presente. In questa fase il sistema informativo deve garantire una perfetta sincronizzazione tra i dati di prenotazione, quelli relativi al check-in e le informazioni utilizzate ai gate di imbarco, al fine di evitare discrepanze o errori operativi.

Nel complesso, il processo di vendita e gestione dei biglietti in Luxair richiede un sistema informativo robusto e affidabile, capace di gestire un elevato numero di entità interconnesse, tra cui passeggeri, prenotazioni, biglietti, pagamenti, voli, aeromobili e aeroporti. È inoltre necessaria una solida **governance dei dati**, che consenta di tracciare le modifiche, garantire la sicurezza delle transazioni e mantenere la coerenza delle informazioni lungo l'intero ciclo di vita del viaggio. Queste considerazioni hanno costituito la base informativa e concettuale per lo sviluppo del database che ho realizzato nell'ambito del mio Project Work.

3. Indicazione del DBMS selezionato con relativa motivazione della scelta

Per lo sviluppo del modello di persistenza dei dati a supporto del sistema di vendita e gestione dei biglietti di Luxair, ho scelto **PostgreSQL** come Database Management System (DBMS). La decisione deriva da una valutazione attenta di fattori tecnici e funzionali legati alle esigenze di un sistema informativo complesso, in cui affidabilità, coerenza dei dati e prestazioni rappresentano requisiti fondamentali. Un elemento determinante è la piena conformità alle proprietà **ACID** (Atomicità, Consistenza, Isolamento e Durabilità), indispensabili per applicazioni critiche come i sistemi di prenotazione aerea. Ogni operazione relativa alla creazione di prenotazioni, emissione di biglietti o registrazione di pagamenti viene gestita in modo sicuro e coerente. Grazie al supporto avanzato alle transazioni, PostgreSQL garantisce che operazioni correlate vengano trattate come un'unica unità logica, riducendo il rischio di errori quali doppie prenotazioni o dati parzialmente registrati.

Altro aspetto rilevante è il supporto alla modellazione di **relazioni complesse** e ai **vincoli di integrità referenziale**. PostgreSQL consente di definire chiavi primarie, chiavi esterne, vincoli di unicità e controlli di integrità, strumenti fondamentali per rappresentare correttamente entità interconnesse come passeggeri, prenotazioni, voli, tratte, biglietti, check-in e pagamenti, assicurando il controllo della concorrenza e della consistenza dei dati.

Dal punto di vista delle performance e della scalabilità, il DBMS offre funzionalità avanzate quali **indici ottimizzati**, **viste materializzate**, **partizionamento** delle tabelle e **ottimizzazione** delle query, consentendo di gestire grandi volumi di dati e carichi elevati, come quelli che si verificano durante periodi di alta richiesta o accessi simultanei da più canali di vendita.

Infine, PostgreSQL è ampiamente supportato e documentato, con una comunità attiva e numerosi strumenti per gestione, monitoraggio e backup, facilitando lo sviluppo, la manutenzione e l'integrazione con applicazioni esterne come portali web, app mobile o sistemi di reportistica aziendale.

Quindi la scelta di PostgreSQL consente di sviluppare un sistema di gestione dei biglietti strutturato, coerente e sicuro, capace di rappresentare fedelmente i processi reali della compagnia e di garantire affidabilità, scalabilità e buone prestazioni nel tempo.

4. Sviluppo del modello concettuale ER che rappresenta le principali entità con relativi attributi e relazioni della compagnia Luxair.

Durante lo sviluppo del modello concettuale E-R ho realizzato un diagramma volto a rappresentare in maniera chiara le principali entità coinvolte nel sistema di gestione della compagnia aerea Luxair e le relazioni esistenti tra di esse. In una fase iniziale di progettazione, il diagramma includeva anche gli attributi associati a ciascuna entità, al fine di comprendere in modo completo la struttura informativa del sistema e di analizzare nel dettaglio le dipendenze tra i dati. Tuttavia, nella versione finale del diagramma ho scelto di non rappresentare graficamente gli attributi, privilegiando una maggiore leggibilità e una visione più immediata delle relazioni concettuali tra le entità.

Questa scelta è stata motivata dall'esigenza di evitare un'eccessiva densità informativa all'interno del diagramma, che avrebbe potuto comprometterne la chiarezza complessiva. In un sistema articolato come quello di una compagnia aerea, infatti, l'elevato numero di attributi rischia di rendere il diagramma poco leggibile e di distogliere l'attenzione dagli aspetti concettuali fondamentali, ovvero le entità e le relazioni che le collegano. Gli attributi sono stati comunque considerati e definiti con attenzione nelle fasi successive di progettazione logica e implementazione del database.

La rappresentazione grafica del modello E-R mi ha permesso di visualizzare in modo concreto come i dati si relazionano tra loro e quali percorsi informativi essi seguono all'interno del sistema, facilitando la comprensione delle dipendenze e dei vincoli esistenti tra le diverse entità. Nel diagramma ho evidenziato le **chiavi primarie** sottolineandole, così da distinguere immediatamente dagli altri elementi e rendere più chiara l'identificazione univoca di ciascuna entità.

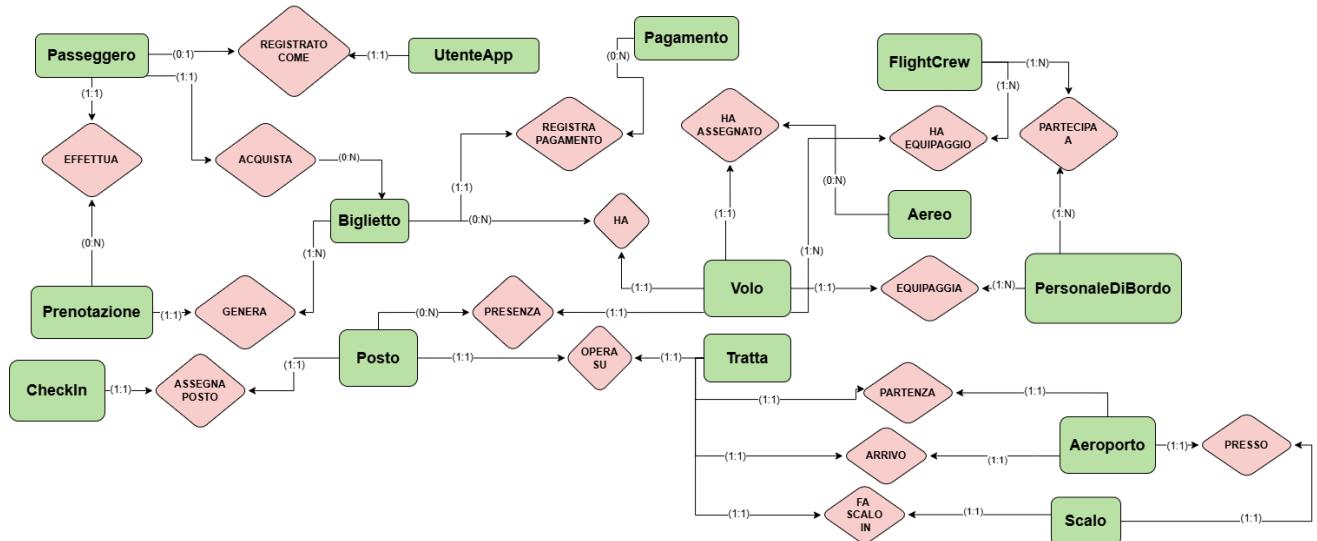
Per la modellazione delle relazioni ho utilizzato i **rombi**, assegnando loro nomi significativi che descrivono il tipo di interazione tra le entità coinvolte. Ad esempio, la relazione "effettua" collega l'entità Passeggero all'entità Prenotazione, mentre la relazione "assegna" mette in relazione il Check-In con il Posto. In questo modo, il diagramma non si limita a mostrare i collegamenti strutturali, ma comunica anche il significato semantico delle interazioni tra le diverse componenti del sistema.

Particolare attenzione è stata dedicata alla definizione delle **cardinalità**, che sono state indicate in modo esplicito per chiarire la molteplicità dei legami tra le entità. Ad esempio, un Passeggero può effettuare una o più Prenotazioni, mentre ogni Prenotazione è associata a un singolo passeggero.

Sebbene nel dominio applicativo del trasporto aereo il personale di bordo operi fisicamente sugli aeromobili, dal punto di vista informativo e gestionale la relazione risulta più correttamente modellata a livello di volo. Per questo motivo, nel modello concettuale ho scelto di rappresentare la relazione molti-a-molti tra *Volo* e *PersonaleDiBordo* tramite l'entità associativa *FlightCrew*. Tale scelta progettuale mi consente di descrivere in modo più fedele la realtà operativa della compagnia, in cui l'equipaggio viene assegnato ai singoli voli in base a pianificazione, turnazione e disponibilità. Questa soluzione permette inoltre di gestire correttamente la partecipazione di uno stesso membro dell'equipaggio a più voli nel tempo e di rappresentare in modo chiaro la composizione dell'equipaggio per ciascun volo, migliorando la coerenza e la flessibilità del modello dati.

Dal punto di vista grafico, ho scelto di disporre le entità in modo orizzontale e distribuito nello spazio, anziché in un'unica disposizione verticale, così da ridurre la sovrapposizione delle linee di collegamento e migliorare la leggibilità complessiva del diagramma. Questa impostazione ha facilitato l'individuazione delle relazioni principali e ha reso più chiara la struttura concettuale del sistema.

Il diagramma E-R ha svolto un ruolo fondamentale nelle fasi successive di sviluppo del database. Esso ha rappresentato la base per la progettazione dello schema relazionale, guidandomi nella definizione delle tabelle, delle chiavi primarie ed esterne e dei vincoli di integrità referenziale. Inoltre, il modello concettuale mi ha supportato nelle scelte di normalizzazione e nella gestione di eventuali ridondanze, preparando in modo strutturato la fase di implementazione fisica e l'ottimizzazione delle interrogazioni.



5. Traduzione del modello ER in un modello relazionale, con definizione di tabelle, chiavi primarie, chiavi esterne e vincoli di integrità.

```

• -- 1. TABELLA PASSEGGERO
CREATE TABLE Passeggero (
    PassengerID INT PRIMARY KEY,
    Nome VARCHAR(50) NOT NULL,
    Cognome VARCHAR(50) NOT NULL,
    DataNascita DATE NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Telefono VARCHAR(20),
    Fedelta INT DEFAULT 0,
    Preferenze TEXT
);

• -- 2. TABELLA UTENTE APP
CREATE TABLE UtenteApp (
    AppUserID INT PRIMARY KEY,
    PassengerID INT UNIQUE NOT NULL,
    Username VARCHAR(50) UNIQUE NOT NULL,
    PasswordHash VARCHAR(255) NOT NULL,
    UltimoAccesso TIMESTAMP,
    PreferenzeApp TEXT,
    FOREIGN KEY (PassengerID) REFERENCES Passeggero(PassengerID)
        ON DELETE CASCADE
);

• -- 3. TABELLA PRENOTAZIONE
CREATE TABLE Prenotazione (
    BookingID INT PRIMARY KEY,
    PassengerID INT NOT NULL,
    DataPrenotazione TIMESTAMP NOT NULL,
    StatoPrenotazione VARCHAR(20) NOT NULL,
    OriginePrenotazione VARCHAR(20) NOT NULL,
    FOREIGN KEY (PassengerID) REFERENCES Passeggero(PassengerID)
        ON DELETE CASCADE
);

```

```

● -- 4. TABELLA AEROPORTO
CREATE TABLE Aeroporto (
    AirportID INT PRIMARY KEY,
    Nome VARCHAR(50) NOT NULL,
    Citta VARCHAR(50) NOT NULL,
    Paese VARCHAR(50) NOT NULL,
    CodiceIATA CHAR(3) UNIQUE NOT NULL,
    CodiceICAO CHAR(4) UNIQUE NOT NULL
);

● -- 5. TABELLA AEREO
CREATE TABLE Aereo (
    AircraftID INT PRIMARY KEY,
    Modello VARCHAR(50) NOT NULL,
    NumeroMatricola VARCHAR(20) UNIQUE NOT NULL,
    CapacitaPosti INT NOT NULL,
    Tipo VARCHAR(20)
);

● -- 6. TABELLA TRATTA
CREATE TABLE Tratta (
    RouteID INT PRIMARY KEY,
    AeroportoPartenzaID INT NOT NULL,
    AeroportoArrivoID INT NOT NULL,
    Distanza DECIMAL(8,2),
    FOREIGN KEY (AeroportoPartenzaID) REFERENCES Aeroporto(AirportID),
    FOREIGN KEY (AeroportoArrivoID) REFERENCES Aeroporto(AirportID)
);

● -- 7. TABELLA SCALO
CREATE TABLE Scalo (
    StopoverID INT PRIMARY KEY,
    RouteID INT NOT NULL,
    AeroportoID INT NOT NULL,
    OraArrivoPrevista TIMESTAMP,
    OraPartenzaPrevista TIMESTAMP,
    DurataStimata TIME,
    FOREIGN KEY (RouteID) REFERENCES Tratta(RouteID) ON DELETE CASCADE,
    FOREIGN KEY (AeroportoID) REFERENCES Aeroporto(AirportID)
);

● -- 8. TABELLA VOLO
CREATE TABLE Volo (
    FlightID INT PRIMARY KEY,
    NumeroVolo VARCHAR(10) NOT NULL,
    RouteID INT NOT NULL,
    AircraftID INT NOT NULL,
    DataPartenza DATE NOT NULL,
    OraPartenza TIME NOT NULL,
    Durata TIME,
    StatoVolo VARCHAR(20) NOT NULL,
    FOREIGN KEY (RouteID) REFERENCES Tratta(RouteID),
    FOREIGN KEY (AircraftID) REFERENCES Aereo(AircraftID)
);

● -- 9. TABELLA PERSONALE DI BORDO
CREATE TABLE PersonaleDiBordo (
    CrewID INT PRIMARY KEY,
    Nome VARCHAR(50) NOT NULL,
    Cognome VARCHAR(50) NOT NULL,
    Ruolo VARCHAR(30) NOT NULL,
    Certificazioni TEXT
);

```

```

● -- 10. TABELLA FLIGHTCREW
CREATE TABLE FlightCrew (
    FlightID INT NOT NULL,
    CrewID INT NOT NULL,
    PRIMARY KEY (FlightID, CrewID),
    FOREIGN KEY (FlightID) REFERENCES Volo(FlightID) ON DELETE CASCADE,
    FOREIGN KEY (CrewID) REFERENCES PersonaleDiBordo(CrewID) ON DELETE CASCADE
);

● -- 11. TABELLA BIGLIETTO
CREATE TABLE Biglietto (
    TicketID INT PRIMARY KEY,
    BookingID INT NOT NULL,
    FlightID INT NOT NULL,
    PNR VARCHAR(10) UNIQUE NOT NULL,
    Classe VARCHAR(20) NOT NULL,
    StatoBiglietto VARCHAR(20) NOT NULL,
    BagaglioIncluso VARCHAR(50),
    ServiziExtra TEXT,
    PassengerID INT,
    FOREIGN KEY (BookingID) REFERENCES Prenotazione(BookingID) ON DELETE CASCADE,
    FOREIGN KEY (FlightID) REFERENCES Volo(FlightID) ON DELETE CASCADE,
    FOREIGN KEY (PassengerID) REFERENCES Passeggero(PassengerID)
);

● -- 12. TABELLA PAGAMENTO
CREATE TABLE Pagamento (
    PaymentID INT PRIMARY KEY,
    TicketID INT NOT NULL,
    MetodoPagamento VARCHAR(20) NOT NULL,
    DataPagamento TIMESTAMP NOT NULL,
    Importo DECIMAL(10,2) NOT NULL,
    StatoPagamento VARCHAR(20) NOT NULL,
    FOREIGN KEY (TicketID) REFERENCES Biglietto(TicketID) ON DELETE CASCADE
);

```

```

● -- 13. TABELLA POSTO
CREATE TABLE Posto (
    SeatID INT PRIMARY KEY,
    FlightID INT NOT NULL,
    NumeroPosto VARCHAR(5) NOT NULL,
    Classe VARCHAR(20) NOT NULL,
    Stato VARCHAR(20) NOT NULL,
    FOREIGN KEY (FlightID) REFERENCES Volo(FlightID) ON DELETE CASCADE
);

● -- 14. TABELLA CHECK-IN
CREATE TABLE CheckIn (
    CheckInID INT PRIMARY KEY,
    TicketID INT UNIQUE NOT NULL,
    TipoCheckIn VARCHAR(20) NOT NULL,
    DataCheckIn TIMESTAMP NOT NULL,
    PostoAssegnato INT,
    CartaImbarcoGenerata BOOLEAN DEFAULT FALSE,
    NumeroPosto VARCHAR(5),
    FOREIGN KEY (TicketID) REFERENCES Biglietto(TicketID) ON DELETE CASCADE,
    FOREIGN KEY (PostoAssegnato) REFERENCES Posto(SeatID)
);

```

6. Applicazione delle forme normali per eliminare ridondanze e migliorare la coerenza dei dati. Eventuale denormalizzazione per ottimizzare delle prestazioni.

Nel corso della progettazione del database ho applicato in modo sistematico i principi della **normalizzazione**, con l'obiettivo di eliminare ridondanze, ridurre il rischio di anomalie di inserimento, aggiornamento e cancellazione e garantire la coerenza dei dati. L'analisi è stata condotta considerando progressivamente la Prima, la Seconda e la Terza Forma Normale, valutando per ciascuna tabella il rispetto dei requisiti teorici e le eventuali implicazioni operative.

Analisi delle forme normali:

La **Prima Forma Normale (1NF)** richiede che tutti gli attributi siano atomici, ovvero che ciascun campo contenga un solo valore. Nel modello progettato, tutte le tabelle rispettano questo requisito: non sono presenti campi di tipo array o attributi multivalore e ogni dato è memorizzato in una colonna specifica, garantendo una struttura chiara e facilmente interrogabile.

La **Seconda Forma Normale (2NF)** richiede che le tabelle siano in 1NF e che ogni attributo non chiave dipenda dall'intera chiave primaria. La maggior parte delle tabelle, come *Passeggero*, *Aeroporto*, *Aereo* o *PersonaleDiBordo*, presenta chiavi primarie semplici e soddisfa automaticamente la 2NF. Particolare attenzione è stata dedicata alla tabella *FlightCrew*, con chiave primaria composta da *FlightID* e *CrewID*, che rappresenta una relazione molti-a-molti tra *Volo* e *PersonaleDiBordo*. In questo caso, gli attributi dipendono dall'associazione nel suo insieme, garantendo la conformità alla 2NF.

La **Terza Forma Normale (3NF)** richiede che non esistano dipendenze transitive tra attributi non chiave. Le tabelle *Prenotazione*, *Biglietto*, *Volo* e *Scalo* rispettano questo requisito, poiché ogni attributo descrive direttamente la chiave primaria.

Nella tabella *CheckIn* è emersa una possibile ridondanza legata al posto assegnato. L'attributo *PostoAssegnato* fa riferimento a *SeatID*, associato a un volo, e il recupero del numero di posto avrebbe richiesto un join aggiuntivo con *Posto*, aumentando la complessità delle interrogazioni. Per questo motivo è stata introdotta una denormalizzazione mirata, aggiungendo l'attributo *NumeroPosto* direttamente in *CheckIn*, consentendo un accesso rapido all'informazione più rilevante senza compromettere significativamente la coerenza dei dati.

In conclusione, l'applicazione delle forme normali fino alla 3NF ha prodotto un modello coerente, privo di ridondanze superflue e facilmente manutenibile. Le denormalizzazioni introdotte sono motivate da esigenze operative e orientate all'ottimizzazione delle prestazioni, mantenendo elevata integrità e affidabilità del database.

7. Definizione di indici e strategie per migliorare le performance delle interrogazioni.

Nel modello relazionale sviluppato per Luxair, l'obiettivo non è stato solo garantire coerenza e integrità dei dati, ma anche assicurare interrogazioni **rapide** ed **efficienti**. In un contesto operativo, il sistema deve rispondere prontamente a richieste di passeggeri, personale di terra e sistemi di prenotazione esterni. La creazione di indici rappresenta uno strumento fondamentale per migliorare le performance delle query, soprattutto su tabelle soggette a rapida crescita come *Biglietto*, *Volo* e *CheckIn*. Gli indici consentono al DBMS di individuare velocemente le righe che soddisfano una condizione, evitando la scansione completa della tabella e riducendo i tempi di esecuzione.

La definizione degli indici richiede attenzione, poiché ogni indice occupa spazio e può influire sulle operazioni di scrittura (INSERT, UPDATE, DELETE). Per questo, nel progetto è stata adottata una strategia selettiva, creando indici solo sui campi effettivamente coinvolti nelle interrogazioni più frequenti e rilevanti dal punto di vista operativo.

Nel modello che ho progettato, le interrogazioni più frequenti riguardano in particolare:

- la **ricerca dei biglietti associati a uno specifico passeggero**, tipicamente effettuata tramite l'identificativo del passeggero o il codice PNR;
- l'**accesso rapido alle informazioni di check-in**, come il posto assegnato a un determinato biglietto;

- la **consultazione dei voli** in base alla data di partenza o al numero di volo, operazione fondamentale sia per il personale operativo sia per i sistemi di prenotazione;
- la **verifica dei pagamenti** associati ai biglietti, necessaria per controllare lo stato delle transazioni e la validità delle prenotazioni;
- la **consultazione degli scali** relativi a una determinata tratta o a uno specifico aeroporto.

Per rispondere in modo efficiente a queste esigenze, ho definito indici mirati su attributi quali *Email* del passeggero, *PassengerID*, *PNR* del biglietto, *NumeroVolo*, *DataPartenza*, *TicketID* e sugli identificativi delle relazioni tra le tabelle. In alcuni casi, sono stati utilizzati anche indici composti, che permettono di ottimizzare interrogazioni basate su più criteri di ricerca contemporaneamente.

Questa strategia di indicizzazione consente al mio database di supportare in modo efficace le operazioni tipiche del sistema, migliorando le performance complessive senza compromettere la manutenibilità e la coerenza dei dati. In tal modo, il modello progettato risulta non solo corretto dal punto di vista teorico, ma anche adeguato a sostenere un carico operativo realistico, in linea con le esigenze di una compagnia aerea come Luxair.

- Per la tabella **Passeggero** è stato opportuno creare indici sui campi *Email* e sulla combinazione di *Nome* e *Cognome*, poiché le ricerche sui passeggeri avvengono spesso a partire da queste informazioni. L'indice sull'email consente un accesso rapido ai dati del passeggero, mentre l'indice composto su nome e cognome ottimizza le interrogazioni effettuate in fase di consultazione o assistenza.

```
-- INDICI PER MIGLIORARE LE PERFORMANCE
-- Passeggero
CREATE INDEX idx_Passeggero_Email ON Passeggero(Email);
CREATE INDEX idx_Passeggero_NomeCognome ON Passeggero(Nome, Cognome);
```

- La tabella **Biglietto**, che rappresenta una delle tabelle più rilevanti e potenzialmente più estese del database, beneficia della creazione di indici sui campi *PassengerID*, *PNR* e *StatoBiglietto*. Questa scelta consente di recuperare rapidamente tutti i biglietti associati a un determinato passeggero, di individuare uno specifico biglietto tramite il codice PNR e di filtrare in modo efficiente i biglietti in base al loro stato, ad esempio attivi o venduti.

```
-- Biglietto
CREATE INDEX idx_Biglietto_PassengerID ON Biglietto(PassengerID);
CREATE INDEX idx_Biglietto_PNR ON Biglietto(PNR);
CREATE INDEX idx_Biglietto_Stato ON Biglietto(StatoBiglietto);
CREATE INDEX idx_Biglietto_PassengerID_FlightID ON Biglietto(PassengerID, FlightID);
```

- La tabella **CheckIn**, che include il campo denormalizzato *NumeroPosto*, è stata indicizzata insieme a *TicketID* al fine di consentire un accesso rapido alle informazioni relative al posto assegnato a uno specifico biglietto. Questa scelta permette di recuperare il dato in modo diretto, evitando join aggiuntivi con la tabella Posto e migliorando così l'efficienza delle interrogazioni più frequenti.

```
-- CheckIn
CREATE INDEX idx_CheckIn_TicketID ON CheckIn(TicketID);
CREATE INDEX idx_CheckIn_NumeroPosto ON CheckIn(NumeroPosto);
```

- Nella tabella **Volo** risulta vantaggioso definire indici sui campi *NumeroVolo* e *DataPartenza*, sia singolarmente sia in forma combinata. Questa scelta è motivata dal fatto che molte interrogazioni operative richiedono di individuare uno specifico volo in base al suo identificativo e alla data di partenza, consentendo così ricerche più rapide ed efficienti.

```
-- Volo
CREATE INDEX idx_Volo_NumeroVolo ON Volo(NumeroVolo);
CREATE INDEX idx_Volo_DataPartenza ON Volo(DataPartenza);
CREATE INDEX idx_Volo_NumeroVolo_Data ON Volo(NumeroVolo, DataPartenza);
```

- La tabella **Scalo**, collegata alle entità *Tratta* e *Aeroporto*, beneficia della creazione di indici sui campi *RouteID* e *AeroportoID*. Questa scelta consente di velocizzare le interrogazioni relative ai voli che prevedono determinati scali, così come la consultazione degli scali associati a uno specifico aeroporto.

```
-- Scalo
CREATE INDEX idx_Scalo_RouteID ON Scalo(RouteID);
CREATE INDEX idx_Scalo_AeroportoID ON Scalo(AeroportoID);
```

- Analogamente, la tabella **Pagamento** trae beneficio dall'indicizzazione dei campi *TicketID* e *DataPagamento*, poiché consente sia di individuare rapidamente i pagamenti associati a uno specifico biglietto, sia di estrarre in modo efficiente report basati su criteri temporali.

```
-- Pagamento
CREATE INDEX idx_Pagamento_TicketID ON Pagamento(TicketID);
CREATE INDEX idx_Pagamento_DataPagamento ON Pagamento(DataPagamento);
```

- Infine, la tabella **Posto** beneficia della definizione di un indice composto sui campi (*FlightID*, *NumeroPosto*), in quanto consente di individuare rapidamente lo stato di un posto specifico all'interno di un determinato volo, operazione frequente durante le fasi di prenotazione e check-in.

```
-- Posto
CREATE INDEX idx_Posto_FlightID_Numeron ON Posto(FlightID, NumeroPosto);
```

8. Definizione di almeno cinque query SQL rappresentative di operazioni tipiche, come ricerca biglietti disponibili, storico prenotazioni di un cliente e verifica validità di un biglietto.

Al fine di completare il Project Work e dimostrare l'effettiva operatività del modello di persistenza dei dati progettato, ho definito una serie di **query SQL** rappresentative delle principali operazioni che un sistema informativo a supporto di una compagnia aerea come Luxair deve essere in grado di gestire. Le interrogazioni selezionate simulano scenari realistici di utilizzo del database e fanno riferimento a funzionalità centrali del servizio, quali la ricerca dei biglietti disponibili, la consultazione dello storico delle prenotazioni di un passeggero e la verifica della validità di un biglietto.

Le query sono state progettate tenendo conto della struttura del modello relazionale, dei vincoli di integrità definiti e degli indici introdotti per ottimizzare le prestazioni, in modo da garantire interrogazioni efficienti e coerenti con i processi operativi reali della compagnia.

- Query 1 – Ricerca voli disponibili per tratta e data

Questa query restituisce l'elenco dei voli programmati per una specifica tratta e data. Mostra informazioni quali il numero del volo, le città di partenza e arrivo, la data e l'orario di partenza

e lo stato del volo. Filtra i voli in base al codice IATA degli aeroporti di partenza e arrivo e considera solo i voli con stato “Programmato”.

```
• -- Inizio Query.
-- Query 1 - Ricerca voli disponibili per tratta e data
SELECT
    v.FlightID,
    v.NumeroVolo,
    ap_partenza.Citta AS CittaPartenza,
    ap_arrivo.Citta AS CittaArrivo,
    v.DataPartenza,
    v.OraPartenza,
    v.StatoVolo
FROM Volo v
JOIN Tratta t ON v.RouteID = t.RouteID
JOIN Aeroporto ap_partenza ON t.AeroportoPartenzaID = ap_partenza.AirportID
JOIN Aeroporto ap_arrivo ON t.AeroportoArrivoID = ap_arrivo.AirportID
WHERE ap_partenza.CodiceIATA = 'LUX'
    AND ap_arrivo.CodiceIATA = 'FCO'
    AND v.DataPartenza = '2025-07-15'
    AND v.StatoVolo = 'Programmato';
```

- **Query 2 – Biglietti disponibili per un volo specifico**

La query elenca tutti i biglietti ancora disponibili per un determinato volo. Include informazioni come l’ID del biglietto, il PNR, la classe e lo stato del biglietto. Filtra i risultati in base all’ID del volo e allo stato “Disponibile”.

```
• -- Query 2 - Biglietti disponibili per un volo specifico
SELECT
    b.TicketID,
    b.PNR,
    b.Classe,
    b.StatoBiglietto
FROM Biglietto b
WHERE b.FlightID = 101
    AND b.StatoBiglietto = 'Disponibile';
```

- **Query 3 – Storico prenotazioni di un passeggero**

Questa query mostra lo storico completo delle prenotazioni di un passeggero, includendo dettagli sulla prenotazione, sui biglietti e sui voli associati. I risultati comprendono il numero del volo, la data di partenza, il PNR, la classe del biglietto e lo stato della prenotazione, ordinati dalla prenotazione più recente alla più vecchia.

```
• -- Query 3 - Storico prenotazioni di un passeggero
SELECT
    p.BookingID,
    p.DataPrenotazione,
    p.StatoPrenotazione,
    v.NumeroVolo,
    v.DataPartenza,
    b.PNR,
    b.Classe,
    b.StatoBiglietto
FROM Prenotazione p
JOIN Biglietto b ON p.BookingID = b.BookingID
JOIN Volo v ON b.FlightID = v.FlightID
WHERE p.PassengerID = 10
ORDER BY p.DataPrenotazione DESC;
```

- **Query 4 – Prenotazioni future di un passeggero**

La query restituisce tutte le prenotazioni future di un passeggero per voli ancora disponibili. Fornisce informazioni simili allo storico delle prenotazioni (numero del volo, data di partenza, PNR, classe del biglietto) ma filtra solo i voli con data di partenza uguale o successiva alla data corrente e con biglietti nello stato “Disponibile”.

```
④ -- Query 4 – Prenotazioni future di un passeggero (solo voli ancora disponibili)
SELECT
    p.BookingID,
    p.DataPrenotazione,
    v.NumeroVolo,
    v.DataPartenza,
    b.PNR,
    b.Classe,
    b.StatoBiglietto
FROM Prenotazione p
JOIN Biglietto b ON p.BookingID = b.BookingID
JOIN Volo v ON b.FlightID = v.FlightID
WHERE p.PassengerID = 10
    AND b.StatoBiglietto = 'Disponibile'
    AND v.DataPartenza >= CURRENT_DATE
ORDER BY v.DataPartenza ASC;
```

- **Query 5 – Verifica validità di un biglietto tramite PNR**

Questa query permette di verificare lo stato di un biglietto conoscendone il PNR. Restituisce il PNR, lo stato del biglietto, il numero del volo, la data di partenza e lo stato del volo, consentendo di controllare rapidamente la validità e lo stato attuale del biglietto.

```
④ -- Query 5 – Verifica validità di un biglietto tramite PNR
SELECT
    b.PNR,
    b.StatoBiglietto,
    v.NumeroVolo,
    v.DataPartenza,
    v.StatoVolo
FROM Biglietto b
JOIN Volo v ON b.FlightID = v.FlightID
WHERE b.PNR = 'ABCD1234';
```

- **Aggiornamento dati per query 4 – Adeguamento delle date di partenza dei voli**

Poiché la query 4 restituisce solo le prenotazioni future con voli ancora disponibili, i dati di test esistenti, contenendo date nel passato rispetto alla data corrente, non producevano risultati. Per consentire il corretto funzionamento della query, ho aggiornato il volo specifico impostando una data di partenza **futura**. In questo modo la query può restituire correttamente le prenotazioni future del passeggero.

```
④ -- Aggiornamento necessario per far funzionare la query 4.
-- La query 4 filtra solo i voli con DataPartenza >= CURRENT_DATE.
-- Poiché i dati di test inseriti potevano essere nel passato rispetto alla data corrente,
-- aggiorno la data dei voli a una data futura per garantire che la query restituisca risultati.
UPDATE Volo
SET DataPartenza = '2026-07-15'
WHERE FlightID = 101;
```

Campi di applicazione

(Descrivere gli ambiti di applicazione dell'elaborato progettuale e i vantaggi derivanti della sua applicazione):

L'elaborato progettuale trova applicazione nell'ambito dei sistemi di gestione dei biglietti aerei, con un focus particolare sulle compagnie di trasporto come Luxair. Il modello sviluppato consente di rappresentare in maniera completa tutte le fasi del ciclo di vita di un biglietto, dalla prenotazione alla vendita, fino al check-in e alla gestione dei posti, includendo anche aspetti relativi ai servizi extra, ai bagagli, al personale di bordo e agli scali. Grazie alla strutturazione relazionale delle informazioni, è possibile gestire in modo efficace sia le prenotazioni singole sia quelle multiple associate a un passeggero, permettendo di generare biglietti elettronici con codici PNR univoci e di registrare i pagamenti effettuati attraverso diversi metodi, come carta di credito, PayPal o bonifico bancario.

Il modello consente inoltre di organizzare e consultare facilmente i dati relativi ai voli, alle tratte e agli aeroporti, comprese informazioni su eventuali scali intermedi, durata dei voli e stato operativo. La gestione del check-in risulta semplificata grazie all'assegnazione diretta dei posti ai biglietti, con la possibilità di generare carte d'imbarco elettroniche. Inoltre, il sistema permette di tracciare il personale di bordo e i servizi aggiuntivi richiesti dai passeggeri, offrendo un quadro completo delle operazioni.

L'applicazione di questo modello comporta vantaggi concreti dal punto di vista operativo e gestionale. La normalizzazione garantisce coerenza e integrità dei dati, mentre alcune denormalizzazioni mirate, come la duplicazione del PassengerID in Biglietto e del NumeroPosto in CheckIn, consentono di ottimizzare le interrogazioni più frequenti e migliorare le performance. Il modello è inoltre scalabile e flessibile, potendo essere esteso per integrare nuove funzionalità, come programmi fedeltà avanzati, promozioni o gestione dei bagagli speciali, senza compromettere la struttura di base.

Valutazione dei risultati

(Descrivere le potenzialità e i limiti ai quali i risultati dell'elaborato sono potenzialmente esposti):

L'elaborato progettuale rappresenta un modello completo e coerente per la gestione dei biglietti aerei, in grado di garantire sia l'integrità dei dati sia l'efficienza operativa. La normalizzazione fino alla terza forma normale assicura che le informazioni siano coerenti e prive di ridondanze non necessarie, mentre l'introduzione mirata di denormalizzazioni e indici consente di velocizzare le interrogazioni più frequenti, come la ricerca dei biglietti di un passeggero o l'individuazione dei posti assegnati durante il check-in. Questo approccio rende il sistema adatto a gestire grandi quantità di dati, permettendo di ottenere rapidamente informazioni utili sia a livello gestionale sia per la produzione di report.

Tuttavia, il mio modello presenta alcune limitazioni intrinseche. Le denormalizzazioni, pur migliorando le performance, introducono campi duplicati che richiedono attenzione per mantenere la coerenza in caso di aggiornamenti o modifiche dei dati. Inoltre, il modello non prevede meccanismi avanzati di storicizzazione, come il versioning dei biglietti o dei voli modificati, né gestisce automaticamente scenari complessi legati a servizi aggiuntivi personalizzati per i passeggeri. Su scale molto grandi, con un numero elevato di voli, biglietti e passeggeri, possono emergere criticità sulle prestazioni, che richiederebbero ulteriori strategie di ottimizzazione, come il partizionamento delle tabelle o il caching dei dati più statici.

In conclusione, l'elaborato dimostra come sia possibile progettare un sistema per la gestione dei biglietti aerei di Luxair che coniughi integrità dei dati, efficienza operativa e scalabilità, offrendo una base solida per future evoluzioni e garantendo al contempo affidabilità e coerenza nelle operazioni quotidiane.

Il codice sorgente e i file del progetto sono disponibili nella repository pubblica al seguente link:
https://github.com/supersamy22/ProjectWork_Luxair.git.