# Introduzione

Descrizione del progetto

Progetto individuale volto alla creazione di un’applicazione web con database relazionale alla base. Per la progettazione si useranno i seguenti linguaggi di programmazione:

* SQL, per implementare il database su Oracle SQL Developer;
* Java, per la costruzione del backEnd su Eclipse IDE;
* HTML, per la costruzione del frontEnd nel framework Angular

Obiettivo

Lo scopo del progetto è creare un'applicazione web con la quale gli utenti potranno acquistare biglietti aerei. Il sistema si basa su un database relazionale con le sue funzioni di gestione voli e prenotazione. Avremo un database che supporterà la prenotazione su molteplici tratte in tutto il mondo, centinaia di voli per varie compagnie aeree.

# Fase 1 - Database

Diagramma Entità-Relazione

Ho utilizzato il software di disegno grafico diagrams.net per la rappresentazione dell’architettura del Diagramma Entità-Relazione.

Immagine che contiene schermata, design, Simmetria

Descrizione generata automaticamente

Logica del diagramma E/R

Il database sarà sviluppato per supportare le prenotazioni effettuate da un cliente e relative a un volo. La logica del diagramma parte proprio dalla tabella **cliente**, contenente i dati del medesimo che effettuerà la **prenotazione** e riconosciuto univocamente dalla PK “id\_cliente”. Da questo id, la tabella prenotazione riconoscerà il cliente e lo collegherà al **volo** acquistato, a sua volta riconosciuto univocamente da un “id\_volo”. La tabella volo terrà conto dell’aereo con il quale verrà effetuato e della compagnia aerea che lo organizza. Le entità **aereo** e **compagnia\_aerea** sono anch’esse legate da una relazione, in quanto la compagnia aerea è in grado di possedere un aereo. Tornando all’entità principale del diagramma, nonché volo, questa necessita di avere un aeroporto di origine, uno di destinazione e che vengano menzionati gli eventuali aeroporti in cui l’aeroporto può fare scalo. Per non creare tabelle pressochè identiche, ho optato per raccogliere le caratteristiche di un aeroporto nell’entità **aeroporto** e creare tre tabelle ponte, ovvero **aeroporto\_origine**, **aeroporto\_destinazione** e **aeroporto\_scali**. Le chiavi primarie di queste entità (id\_aeroporto\_....) costituiscono foreign key necessarie a volo per capire la tratta da affrontare. Inoltre, passandomi id\_volo come chiave esterna, ho creato una tabella statistiche che raccoglierà statistiche utili relative a ciascun volo.

Modello relazionale

Ho costruito le tabelle normalizzate derivanti dal diagramma E/R su Microsoft Excel.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

Implementazione del DB su Oracle SQL Developer

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Implementazione Trigger

1. TRIGGER STATO AEREO

Attraverso questo trigger verifico se lo stato dell’aereo (tab= AEREO; col=STATO) è cambiato da “funzionante”(1) a “non funzionante”(2). A condizione avverata, cerco un aereo che risulti funzionante e successivamente modifico i voli collegati all’aereo il cui stato è cambiato, assegnando il nuovo aereo trovato nel passaggio precedente.

Immagine che contiene testo, schermata, numero, software

Descrizione generata automaticamente

1. TRIGGER PRIMO VOLO

Grazie a questo trigger…

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

1. TRIGGER AGGIORNA STATISTICHE

Con questo trigger…

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente