

# **1<sup>a</sup> Prova in Itinere**

SFT – Società Ferrovie Turistiche

Lorenzo Benvenuti / 0046403

## Sommario

<b>Fase 1:</b> Specifica dei requisiti .....	3
<b>Requisiti utente</b> .....	3
<b>Profili e funzioni</b> .....	3
<b>Descrizione treni e linea ferroviaria</b> .....	4
<b>Definizioni</b> .....	4
<b>Analisi del problema</b> .....	5
<b>Considerazioni preliminari</b> .....	5
<b>Fase 2:</b> Progettazione concettuale .....	6
<b>Analisi dei requisiti</b> .....	6
<b>Glossario dei termini</b> .....	7
<b>Considerazioni</b> .....	8
<b>Revisione della traccia</b> .....	9
<b>Operazioni</b> .....	9
<b>Schema E-R</b> .....	10
<b>Fase 3:</b> Progettazione logica .....	11
<b>Tavola dei volumi</b> .....	11
<b>Tavola delle operazioni</b> .....	12
<b>Tavola degli accessi</b> .....	12
<b>Ristrutturazione schema E-R</b> .....	14
<b>Schema E-R ristrutturato</b> .....	15
<b>Schema relazionale</b> .....	16
<b>Fase 4:</b> Progettazione fisica.....	17
<b>Implementazione della base di dati</b> .....	17
<b>Script SQL</b> .....	18

# Fase 1: Specifica dei requisiti

Questa fase iniziale serve per raccogliere tutti i requisiti e le specifiche fornite per realizzare il progetto.

## Requisiti utente

Ci viene richiesto esplicitamente questo:

“Si chiede di realizzare una web application per la vendita biglietti di una società ferroviaria che gestisce una linea turistica. La vendita avverrà on line ed i visitatori potranno acquistare i biglietti prenotando i posti a sedere. La società gestisce una sola linea lunga 54 km con 10 stazioni, ognuna delle quali può essere sia origine che destinazione di traffico. Il costo del biglietto varierà in funzione dei km del viaggio. Sono previste 4 coppie di treni storici al giorno nei giorni festivi, tutto l'anno, e 1 coppia di treni nei giorni feriali dal 1°giugno al 30 settembre. Ogni coppia di treni potrà avere una composizione variabile di carrozze e diverse locomotive. Il sito, per quanto attiene alle pagine di consultazione degli orari dei treni, può avere dei picchi di accesso nella giornata di centinaia di utenti al secondo. L'applicazione dovrà avere 1 backoffice amministrativo per il calcolo della redditività di ciascun treno ed un backoffice di esercizio per la gestione degli orari e delle composizioni dei treni.”

## Profili e funzioni

Vengono indicate le modalità di utilizzo e di funzionamento desiderate:

- **Profilo visitatore**
  - Può leggere le informazioni sulla linea, i paesi attraversati, le stazioni ed i convogli con la storia del materiale rotabile.
- **Profilo utente registrato**
  - Può scegliere la tratta, la data e comprare il biglietto con contestuale prenotazione del posto a sedere. I viaggiatori non possono viaggiare in piedi per motivi di sicurezza.
  - L'acquisto del biglietto avverrà tramite la piattaforma Pay Steam attraverso la chiamata ad una apposita web-api.
  - Il sistema esporrà una web api per la risposta da Pay Steam.
  - Può variare la prenotazione cambiando giorno e posto a sedere.
- **Profilo backoffice amministrativo**
  - Può verificare l'occupazione dei treni e richiedere eventuali treni straordinari al backoffice di esercizio.
  - Può anche cancellare le corse senza prenotazione.
- **Profilo backoffice di esercizio**
  - Può comporre i convogli a partire dal materiale rotabile disponibile.
  - Verifica la disponibilità del materiale rotabile in accordo con gli orari.
  - Costruisce gli orari secondo quanto previsto nei requisiti.
  - Può variare le corse aggiungendone nuove o modificando quelle presenti.
  - Può agire secondo le indicazioni del backoffice amministrativo o autonomamente per ottimizzare gli orari o l'utilizzo del materiale rotabile.

## Descrizione treni e linea ferroviaria

Sono presenti le specifiche tecniche ferroviarie con cui il sistema dovrà operare:

- **Materiale rotabile**
  - o La velocità massima prevista in linea è di 50 km/h.
  - o Il materiale rotabile a disposizione è il seguente:
    - Carrozze serie 1928: B 1, B 2 e B 3, da 36 posti a sedere
    - Carrozze serie 1930: C 6, C 9 da 48 posti a sedere
    - Carrozza serie 1952: C 12 da 52 posti a sedere
    - Bagagliai serie 1910: CD 1 e CD 2 con comparto passeggeri da 12 posti a sedere
    - Automotrici a nafta: AN 56.2 e AN 56.4 da 56 posti a sedere (possono viaggiare isolate senza carrozze al seguito)
- **Linea SFT**
  - o Linea a binario unico.
  - o Le stazioni sono le seguenti:
    1. Torre Spaventa Km 0,000 stazione in comune con Trenitalia
    2. Prato Terra Km 2,700
    3. Rocca Pietrosa Km 7,580
    4. Villa Pietrosa Km 12,680
    5. Villa Santa Maria Km 16,900
    6. Pietra Santa Maria Km 23,950
    7. Castro Marino Km 31,500
    8. Porto Spigola Km 39,500
    9. Porto San Felice Km 46,000
    10. Villa San Felice Km 54,680

## Definizioni

Tutto è corredato da definizioni per rendere più semplice la comprensione e di conseguenza meno ambigue le richieste:

- **Linea:** intero percorso da Torre Spaventa a Villa San Felice
- **Tratta:** percorso tra la stazione di partenza e quella di arrivo (non necessariamente l'intera linea)
- **Sub tratta:** percorso tra una stazione e la successiva
- **Convoglio:** composizione di materiale rotabile
- **Treno:** convoglio in viaggio ad una certa data/ora
- **Traccia oraria:** è l'insieme delle informazioni che sono legate ad un treno nello spazio tempo (località e data/ora lungo il percorso)

## Analisi del problema

Inoltre viene fornita una spiegazione aggiuntiva sul funzionamento della tratta, onde evitare fraintendimenti:

- Un convoglio, per poter partire da una stazione, deve essere fisicamente nella stazione stessa.
- Un convoglio non può partire da una stazione senza esserci arrivato prima.
- Non possono partire dalla stessa stazione due convogli contemporaneamente se non in direzioni opposte.
- Un convoglio giunto alla stazione terminale della linea, deve essere immediatamente disponibile per la programmazione di una nuova corsa (a partire dalla stazione di arrivo).
- Più convogli possono percorrere la linea contemporaneamente in entrambe le direzioni.
- Le sub tratte tra una stazione e l'altra possono essere percorse da un solo convoglio contemporaneamente.

## Considerazioni preliminari

Al termine di questa fase si esprimono delle deduzioni preliminari sui requisiti assegnati. Esse servono per evidenziare dei temi non specificati e fornire quindi tutti gli strumenti necessari per lo step successivo. In questo caso, non potendo chiedere direttamente ulteriori chiarimenti al cliente, verranno adottate osservazioni personali:

- La registrazione dell'utente è di livello elementare. Non sono previste autenticazioni a due fattori (per messaggio, e-mail, ecc.) o modalità più sofisticate.
- L'inserimento dei dati in fase di registrazione avviene tramite protezione crittografica opzionale.
- Le web API descritte comunicano con autenticazione.

## Fase 2: Progettazione concettuale

L'obiettivo di questa fase è analizzare ed organizzare le informazioni possedute per ottenere una buona panoramica di tutti gli attori coinvolti e di tutte le operazioni che tali attori devono compiere all'interno del sistema che dobbiamo sviluppare. Ogni step servirà per eliminare le imprecisioni e le ambiguità (dovute all'utilizzo del linguaggio naturale) ed identificare gli elementi necessari per costituire lo schema E-R finale.

### Analisi dei requisiti

Procediamo nell'analizzare dettagliatamente la traccia fornita per individuare le entità, le relazioni e gli attributi utili alla progettazione. Questo ci aiuterà ad ottenere una scrematura preliminare degli elementi superflui: Legenda (**Entità**, **Relazioni**, **Attributi**)

Si chiede di realizzare una **web application** per la vendita **biglietti** di una **società ferroviaria** che **gestisce** una **linea turistica**.

La vendita avverrà on line ed i **visitatori** potranno **acquistare** i **biglietti** prenotando i **posti a sedere**.

La **società** gestisce una sola **linea** lunga 54 km con 10 **stazioni**, ognuna delle quali può essere sia **origine** che **destinazione** di traffico.

Il **costo** del **biglietto** varierà in funzione dei **km** del viaggio.

Sono previste 4 coppie di **treni storici** al giorno nei **giorni festivi**, tutto l'anno, e 1 coppia di **treni** nei **giorni feriali** dal 1<sup>o</sup> giugno al 30 settembre.

Ogni coppia di **treni** potrà avere una composizione variabile di **carrozze** e diverse **locomotive**.

Il **sito**, per quanto attiene alle pagine di consultazione degli **orari** dei **treni**, può avere dei picchi di accesso nella giornata di centinaia di **utenti** al secondo.

L'**applicazione** dovrà avere 1 **backoffice amministrativo** per il **calcolo della redditività** di ciascun **treno** ed un **backoffice di esercizio** per la **gestione** degli **orari** e delle composizioni dei **treni**.

Per essere più precisi analizziamo anche i profili e le funzioni previsti:

- **Profilo visitatore**
  - Può **leggere** le informazioni sulla **linea**, i **paesi** attraversati, le **stazioni** ed i **convogli** con la storia del **materiale rotabile**.
- **Profilo utente registrato**
  - Può scegliere la **tratta**, la **data** e **comprare** il **biglietto** con contestuale **prenotazione del posto a sedere**. ~~I viaggiatori non possono viaggiare in piedi per motivi di sicurezza.~~
  - L'**acquisto** del **biglietto** avverrà tramite la piattaforma **Pay Steam** attraverso la chiamata ad una apposita **web-api**.
  - Il **sistema** esporrà una **web api** per la risposta da **Pay Steam**.
  - Può variare la **prenotazione** cambiando **giorno** e **posto a sedere**.
- **Profilo backoffice amministrativo**
  - Può **verificare** l'occupazione dei **treni** e **richiedere** eventuali **treni straordinari** al **backoffice di esercizio**.
  - Può anche **cancellare** le **corse** senza prenotazione.
- **Profilo backoffice di esercizio**
  - Può **comporre** i **convogli** a partire dal **materiale rotabile** disponibile.
  - **Verifica** la disponibilità del **materiale rotabile** in accordo con gli **orari**.
  - **Costruisce** gli **orari** secondo quanto previsto nei **requisiti**.
  - Può **variare** le **corse** aggiungendone nuove o modificando quelle presenti.
  - Può agire secondo le indicazioni del **backoffice amministrativo** o autonomamente per **ottimizzare** gli **orari** o l'utilizzo del **materiale rotabile**.

## Glossario dei termini

Una volta individuati gli elementi utili è buona norma redigere un glossario dei termini più importanti, al fine di identificare eventuali sinonimi, chiarire le ambiguità rimaste ed evidenziare i collegamenti che intercorrono tra di essi:

Termino	Descrizione	Sinonimi	Collegamenti
Amministrativo	Utente che gestisce la linea ferroviaria	Utente	Treno
Esercizio	Utente che si occupa del funzionamento della linea ferroviaria	Utente	Convoglio, Corsa, Orario, Materiale rotabile
Biglietto	Documento che certifica il diritto di viaggiare su un mezzo di trasporto (treno) per una specifica tratta		Corsa, Posto, Treno, Utente
Bagagliaio	Sezione dedicata alla raccolta dei bagagli dei passeggeri	Materiale rotabile	Convoglio, Treno
Carrozza	Vettura del treno dedicata al trasporto dei passeggeri	Materiale rotabile	Convoglio, Treno
Convoglio	Composizione di materiale rotabile		Bagagliaio, Carrozza, Locomotiva, Materiale rotabile
Corsa	Percorso di un treno avente una specifica stazione di partenza e arrivo	Tratta	Linea, Stazione, Treno
Linea	Tratto di ferrovia che collega varie stazioni		Corsa, Stazione, Tratta, Treno
Locomotiva	Motrice del treno	Materiale rotabile	Convoglio, Treno
Materiale rotabile	Insieme di veicoli su binari		Bagagliaio, Carrozza, Convoglio, Locomotiva
Orario	Indicazione di data e ora		Corsa, Stazione, Treno
Posto	Spazio assegnato ad un viaggiatore su un mezzo di trasporto		Biglietto, Utente
Stazione	Tappa della linea ferroviaria in cui vengono caricati e scaricati i passeggeri		Corsa, Linea, Tratta
Tratta	Percorso tra la stazione di partenza e quella di arrivo		Linea, Stazione, Treno
Sub tratta	Percorso tra una stazione e la successiva		Stazione, Tratta, Treno
Treno	Mezzo di trasporto formato da una o più carrozze trainate da una locomotiva in viaggio ad una certa data/ora		Biglietto, Bagagliaio, Carrozza, Convoglio, Corsa, Linea, Locomotiva, Orario, Tratta, Utente
Utente	Persona che utilizza l'applicazione e/o i suoi servizi	Amministrativo, Esercizio, Visitatore	Biglietto, Posto, Treno
Visitatore	Utente che consulta l'applicazione	Utente	Linea, Materiale rotabile, Orario, Stazione
API	Interfaccia che permette la comunicazione tra applicazioni		Pay Steam

## Considerazioni

È necessario interpretare il materiale a disposizione ed evidenziare alcuni punti non trattati per ridurre le incertezze e quindi semplificare le fasi successive di progettazione. Va ancora sottolineato che le deduzioni presentate sono di carattere arbitrario, non avendo la possibilità di dialogo con il cliente per chiarire tali tematiche:

- **Orario turistico:** La linea considerata è di uso turistico e ciò si riflette di conseguenza sull'orario. Si presuppone che esso sia strutturato su base giornaliera e che ogni mattina tutto il materiale rotabile sia disponibile presso la prima stazione. La prima partenza potrà avvenire dalle ore 6:00, mentre l'ultima partenza non sarà possibile oltre le 22:00. Eventuali corse notturne a vuoto per riportare i treni alla posizione iniziale avverranno automaticamente, senza necessità di gestione tramite l'applicazione.
- **Costo del biglietto:** Poiché non specificato nei requisiti, si assume un costo pari a **0,20 €/km**.
- **Binario unico:** La linea è a binario unico e i treni possono incrociarsi soltanto presso le stazioni di scambio. È ammesso che i convogli partano dalla stessa stazione, purché viaggino in direzioni opposte.
- **Stazioni:** Si ipotizza che il numero di stazioni rimanga invariato e che siano tutte sempre operative. Consideriamo i treni immediatamente disponibili dopo l'arrivo in stazione, senza alcun tempo di attesa (contesto ipotetico volto a semplificare la gestione degli scambi). L'informazione sulla condivisione della prima stazione con Trenitalia è ridondante.
- **Partenze dei convogli:** Un convoglio può partire da una stazione solo se vi è giunto in precedenza. L'organizzazione degli spostamenti, inclusi eventuali trasferimenti con treni dedicati, sarà a carico del backoffice di esercizio.
- **Prestazioni del treno:** Si ipotizza che i convogli raggiungano istantaneamente la velocità prevista di 50 km/h e la mantengano costantemente, trascurando i tempi di accelerazione e decelerazione.
- **Convogli fissi:** Una volta configurato un convoglio, la sua composizione resterà invariata fino alla cancellazione o modifica.
- **Composizione dei convogli:** La gestione e composizione dei convogli è affidata al backoffice di esercizio. Non ci sono limiti specifici sul numero massimo o minimo di carrozze, ma ogni treno deve includere almeno un elemento con trazione (locomotiva) e garantire posti a sedere per i passeggeri.
- **Treni storici:** L'indicazione sulla natura "storica" dei treni viene considerata trascurabile. Tutti i treni saranno trattati semplicemente come convogli operativi e catalogati tramite ID.
- **Carico:** Non è specificato il numero massimo di utenti in caso di picchi di accesso, ma si presume che l'infrastruttura sia dimensionata per gestire un elevato numero di accessi simultanei (centinaia al secondo).

Si possono fare delle considerazioni anche sulle indicazioni fornite dai profili e dalle funzioni:

- **Backoffice:**
  - o La gestione amministrativa sarà centralizzata. Gli utenti avranno accesso a sezioni specifiche in base ai loro permessi, definiti dalla profilatura.
- **Profili utente:**
  - o **Visitatori:** Possono visualizzare informazioni generiche sulla linea e sui treni. Si presume che le stazioni corrispondano ai nomi dei paesi.
  - o **Utenti registrati:** Dispongono delle stesse funzionalità dei visitatori, con l'aggiunta di quelle espressamente indicate nei requisiti, come la visualizzazione degli orari e la gestione dei pagamenti. Inoltre, possono consultare i dettagli relativi al biglietto acquistato.
  - o **Backoffice amministrativo:** Può monitorare la redditività e l'occupazione dei treni, nonché cancellare specifiche corse (cessazione).
  - o **Backoffice di esercizio:** Si occupa della pianificazione operativa, seguendo le indicazioni del profilo amministrativo (tramite form interno dedicato).

**Nota:** Per semplificare la progettazione preliminare, tutte le informazioni consultabili dagli utenti (la descrizione della linea, la storia del materiale rotabile, le specifiche tecniche dei treni) vengono riassunte nell'attributo "dettagli", mentre gli orari, i periodi feriali e festivi vengono sintetizzati con l'attributo "programma".

## Revisione della traccia

Avendo analizzato nel dettaglio le informazioni disponibili, è possibile adesso riscrivere la traccia per renderla di immediata comprensione. Essa verrà segmentata e suddivisa in una serie di frasi semplici, chiare e standardizzate, in modo da poter visualizzare gli attori in gioco e le operazioni che compiono. Questa pulizia renderà più facile tracciare il diagramma E-R associato:

- Il visitatore legge le informazioni (linea, paesi attraversati, stazioni, convogli, storia del materiale rotabile).
- L'utente registrato sceglie la tratta e la data, compra il biglietto, prenota il posto a sedere, cambia la prenotazione (giorno e posto).
- Il backoffice amministrativo verifica l'occupazione dei treni, richiede i treni straordinari e cancella le corse inutilizzate.
- Il backoffice di esercizio compone i convogli, verifica il materiale rotabile, gestisce gli orari, modifica gli orari, aggiunge le corse.

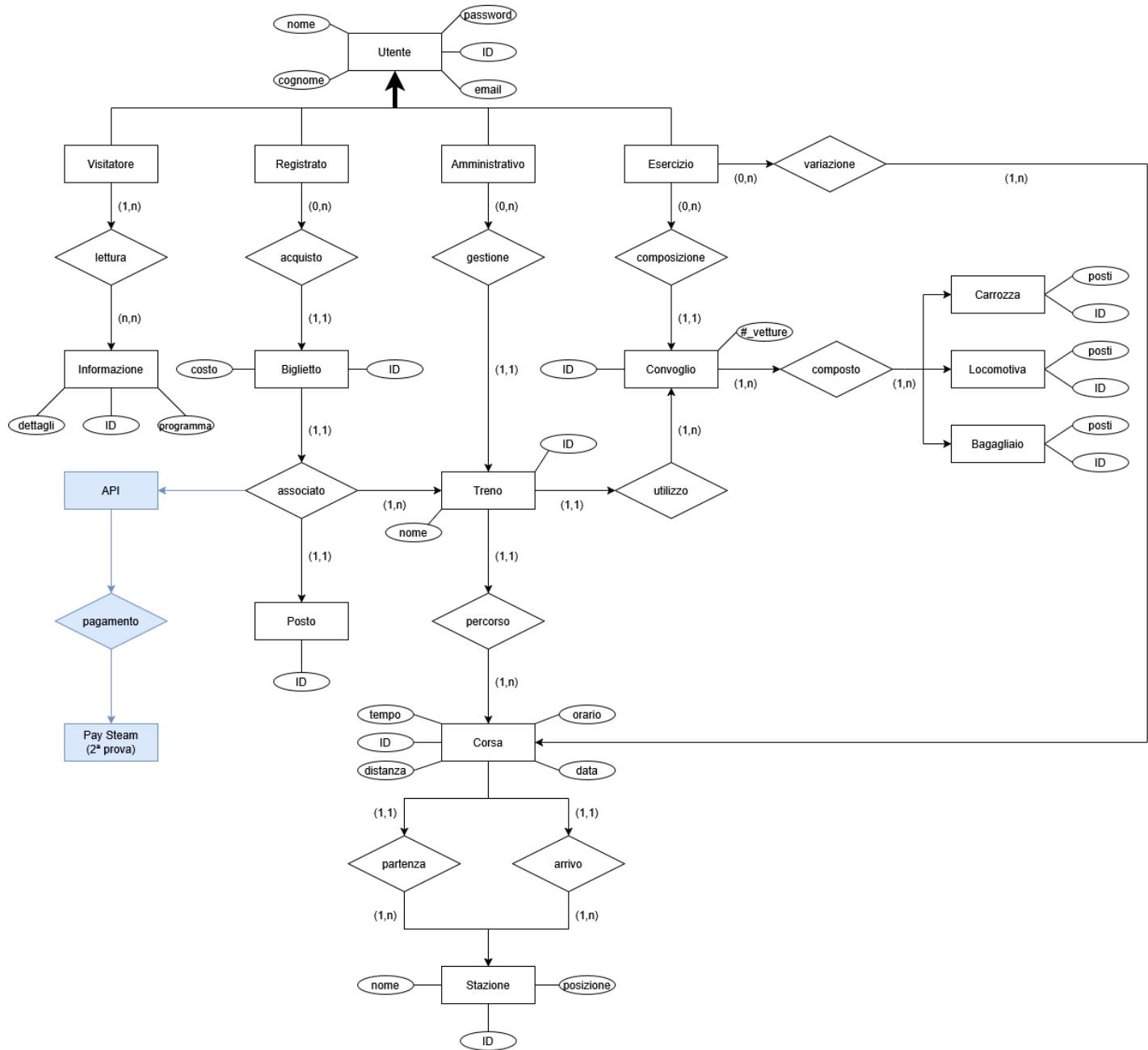
## Operazioni

Le operazioni principali richieste al sistema sono:

- 1) Lettura informazione
- 2) Registrazione utente
- 3) Acquisto biglietto
- 4) Composizione convoglio
- 5) Variazione corsa
- 6) Gestione treno

## Schema E-R

Infine con l'ausilio degli step precedenti si delinea il diagramma entità-relazione, che rappresenta lo schema concettuale iniziale (insieme strutturato di entità, relazioni ed attributi) della base di dati da sviluppare. Si tratta di una prima redazione grezza che verrà raffinata e ristrutturata nei passi successivi:



**Nota:** La parte evidenziata mostra il collegamento tramite API con l'altra applicazione (Pay Steam, 2<sup>a</sup> prova).

## Fase 3: Progettazione logica

Questa fase si occupa di riorganizzare e ristrutturare gli output della progettazione concettuale (in particolare lo schema E-R), con l'obiettivo di ottimizzarli ed adattarli gradualmente alle specifiche di sviluppo della base di dati. Per fare ciò verranno analizzati gli indici di prestazione tramite l'utilizzo di tabelle.

### Tavola dei volumi

Riportiamo tutti i concetti dello schema E-R (entità e relazioni) con il volume previsto a regime. Non avendo delle stime precise sull'occorrenza di tali elementi dalla traccia, procediamo ad assegnare dei valori arbitrari plausibili:

- Per agevolezza, le diverse entità utente (Visitatore, Registrato, Amministrativo, Esercizio) verranno rappresentate da un'unica entità "Utente", alla quale viene assegnato un volume di 3000/gg.
- Le entità e relazioni collegate alle operazioni di modifica (gestione, composizione, variazione, ecc.) hanno un volume leggermente maggiore rispetto alle altre, poiché più modifiche possono interessare lo stesso oggetto.

Concetto	Tipo	Volume/gg
Utente	E	3000
Informazione	E	100
Biglietto	E	2500
Posto	E	2500
Treno	E	10
Convoglio	E	10
Corsa	E	300
Stazione	E	10
Carrozza	E	6
Locomotiva	E	2
Bagagliaio	E	2
lettura	R	1500
acquisto	R	2500
gestione	R	250
composizione	R	50
variazione	R	100
associato	R	2000
percorso	R	300
partenza	R	300
arrivo	R	300
utilizzo	R	20
composto	R	50

## Tavola delle operazioni

Di seguito vengono illustrate le operazioni possibili sul database:

Operazione	Descrizione	Tipo	Frequenza/gg
Op. 1	Lettura informazione	I	1500
Op. 2	Registrazione utente	I	50
Op. 3	Acquisto biglietto	I	2500
Op. 4	Composizione convoglio	I	50
Op. 5	Variazione corsa	I	100
Op. 6	Gestione treno	I	250

## Tavola degli accessi

Essenziale per capire se eliminare o mantenere eventuali ridondanze:

### Lettura informazione

La più semplice operazione poiché richiede la sola lettura dei dati informativi dalla base dati:

Concetto	Costrutto	Accesso	Tipo
Informazione	E	1	L

### Registrazione utente

Prevede una lettura per controllare che non ci siano credenziali corrispondenti a quelle immesse ed una scrittura per registrarle in caso di assenza di duplicati:

Concetto	Costrutto	Accesso	Tipo
Utente	E	1	L
Utente	E	1	S

### Acquisto biglietto

Con questa operazione l'utente registrato sarà in grado di visualizzare e selezionare la tratta, il treno ed il posto a sedere che più desidera:

Concetto	Costrutto	Accesso	Tipo
Utente	E	1	L
Corsa	E	1	L
Treno	E	1	L
Posto	E	1	L
Biglietto	E	1	S
Posto	E	1	S

## **Composizione convoglio**

Il backoffice di esercizio, una volta loggato nell'account, potrà visualizzare lo stato delle varie componenti rotabili, per poi combinarle e registrarle in un convoglio finale, se non fosse già stato creato in precedenza:

<b>Concetto</b>	<b>Costrutto</b>	<b>Accesso</b>	<b>Tipo</b>
Utente	E	1	L
Carrozza	E	1	L
Locomotiva	E	1	L
Bagagliaio	E	1	L
Convoglio	E	1	L
Convoglio	E	1	S

## **Variazione corsa**

Il backoffice di esercizio, una volta loggato nell'account, ha la possibilità di visualizzare le corse attive, se presenti, aggiungerne di nuove, modificarle e salvare il tutto:

<b>Concetto</b>	<b>Costrutto</b>	<b>Accesso</b>	<b>Tipo</b>
Utente	E	1	L
Corsa	E	1	L
Corsa	E	1	S

## **Gestione treno**

Il backoffice amministrativo, una volta loggato nell'account, sarà in grado di controllare i biglietti acquistati (in modo tale da individuare le corse che non sono state prenotate ed eventualmente sopprimerle) e di richiedere treni straordinari. Una volta completate queste azioni, le possibili modifiche vengono registrate:

<b>Concetto</b>	<b>Costrutto</b>	<b>Accesso</b>	<b>Tipo</b>
Utente	E	1	L
Biglietto	E	1	L
Treno	E	1	L
Corsa	E	1	L
Stazione	E	1	L
Treno	E	1	S

## Ristrutturazione schema E-R

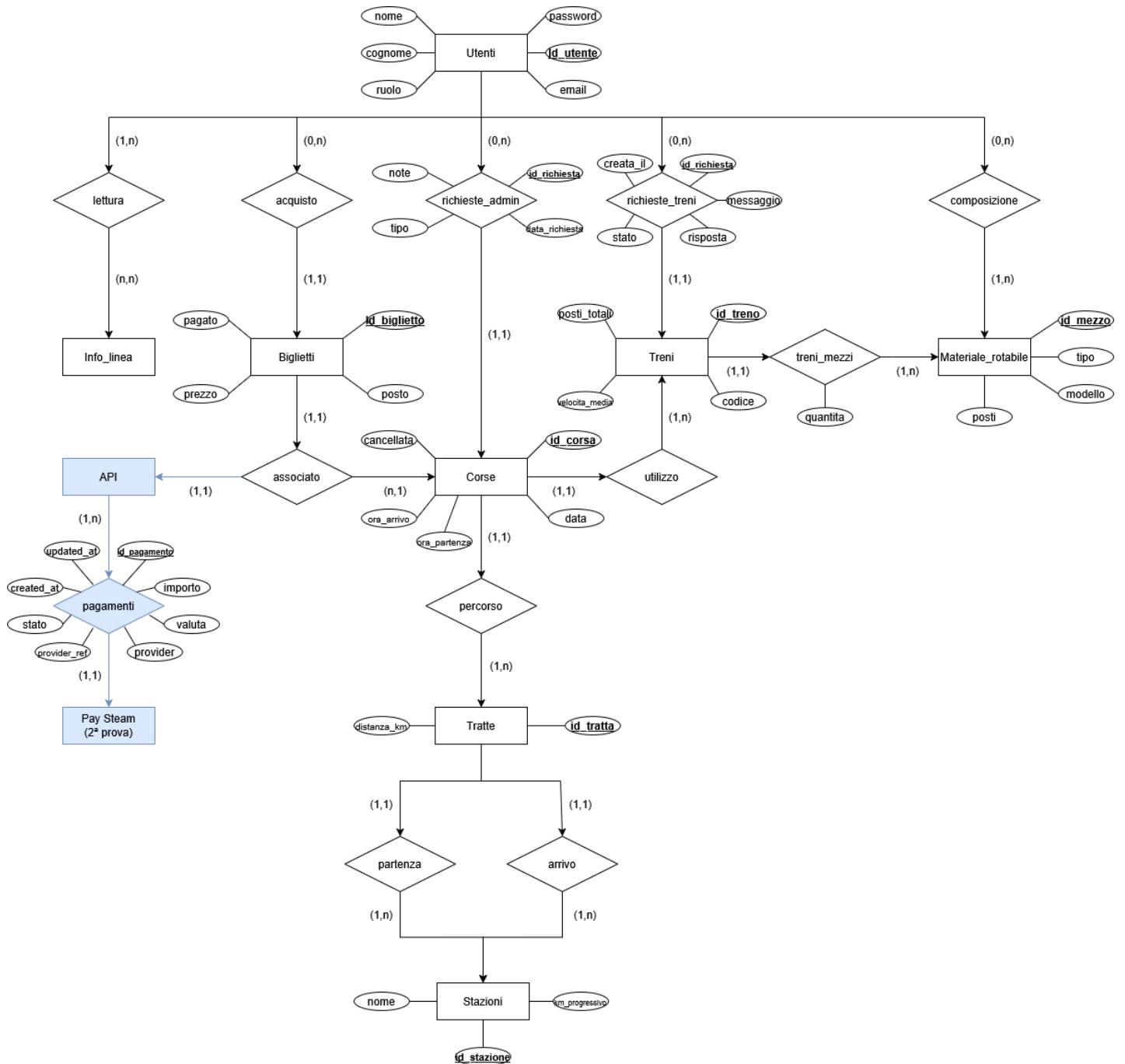
Dopo aver analizzato le ridondanze grazie alle tabelle precedenti, è il momento di completare la ristrutturazione del diagramma tramite l'eliminazione delle generalizzazioni, il partizionamento o accorpamento di entità e relazioni e la scelta degli identificatori primari (primary keys):

- Come già anticipato, si elimina la generalizzazione riassumendo tutti i tipi di utente (Visitatore, Registrato, Amministrativo e Esercizio) nell'entità "Utenti".
- Si accorda l'entità "Posto" a quella "Biglietto" che si trasforma in "Biglietti", facendola diventare un suo attributo. Inoltre si aggiunge l'attributo "pagato" per evidenziare lo stato del biglietto.
- L'entità "Informazione" ed i suoi attributi vengono dedotti dalla descrizione degli altri elementi, quindi essa viene mantenuta nel diagramma per evidenziare solamente la sua presenza, ma nell'atto pratico dello sviluppo dell'applicazione non servirà avere una sezione nel database dedicata alle informazioni, poiché deducibili operativamente tramite riferimenti alle altre tabelle.
- Le relazioni "gestione", "variazione" e "composto" vengono sostituite rispettivamente da "richieste\_admin", "richieste\_treni" e "treni\_mezzi" con i propri attributi, per ottenere un grado di dettaglio maggiore e adattare meglio lo schema ai requisiti della fase di specifica.
- Le entità "Carrozza", "Locomotiva" e "Bagagliaio" vengono accorpate nell'entità sintesi "Materiale\_rotabile", a cui verranno aggiunti gli attributi "tipo" e "modello", in modo tale da poter distinguere facilmente le vetture.
- L'entità "Corsa" diventa "Corse" e sostituisce la posizione di "Treno" del precedente diagramma per avere una maggiore precisione e fare in modo che essa sia collegata direttamente alle richieste dell'utente amministrativo. Viene inoltre inserito l'attributo "cancellata" per evidenziarne lo stato.
- L'entità "Treno" a sua volta diventa "Treni" e prende il posto di "Convoglio", con quest'ultimo che viene accorciato a "treni\_mezzi", rendendo il diagramma più preciso e creando un collegamento diretto con le richieste di treni da parte dell'utente amministrativo.
- Per semplificare la gestione del tragitto dei treni, l'entità "Corse" viene partizionata creando "Tratte", entità corrispondente ai singoli tratti di linea tra due stazioni. Grazie ad essa sarà anche possibile monitorare più accuratamente la velocità dei convogli, in modo da rispettarla (50 km/h).
- Sarà assegnato a tutte le entità uno specifico attributo "id", il quale rappresenterà il loro identificatore primario, grazie alla sua capacità di distinguerle tramite una semplice codifica.
- Per un maggiore dettaglio, l'attributo "orario" verrà scomposto in partenza e arrivo.

**Nota:** Tutte le modifiche apportate per ottenere il diagramma ristrutturato sono finalizzate a semplificare la fase di sviluppo operativo delle prove. Infatti, tale schema servirà come linea guida definitiva per strutturare il database che gestirà le informazioni ed il traffico della prova in itinere.

## Schema E-R ristrutturato

Ecco lo schema E-R finale dell'applicazione, ristrutturato per rispettare i parametri illustrati nei passaggi precedenti:



## Schema relazionale

Infine sintetizziamo lo schema logico definitivo. Vengono applicate ulteriori ottimizzazioni per renderlo più performante e riportate le foreign keys:

**Utenti**(id\_utente, nome, cognome, email, password, ruolo)

**Biglietti**(id\_biglietto, id\_utente, id\_corsa, posto, prezzo, pagato)

**Treni**(id\_treno, codice, velocita\_media, posti\_totali)

**Treni\_mezzi**(id\_treno, id\_mezzo, quantita)

**Materiale\_rotabile**(id\_mezzo, tipo, modello, posti)

**Corse**(id\_corsa, id\_treno, id\_tratta, data, ora\_partenza, ora\_arrivo, cancellata)

**Tratte**(id\_tratta, id\_stazione\_partenza, id\_stazione\_arrivo, distanza\_km)

**Stazioni**(id\_stazione, nome, km\_progressivo)

**Pagamenti**(id\_pagamento, id\_corsa, importo, valuta, provider, provider\_ref, stato, created\_at, updated\_at)

**Richieste\_admin**(id\_richiesta, data\_richiesta, tipo, note)

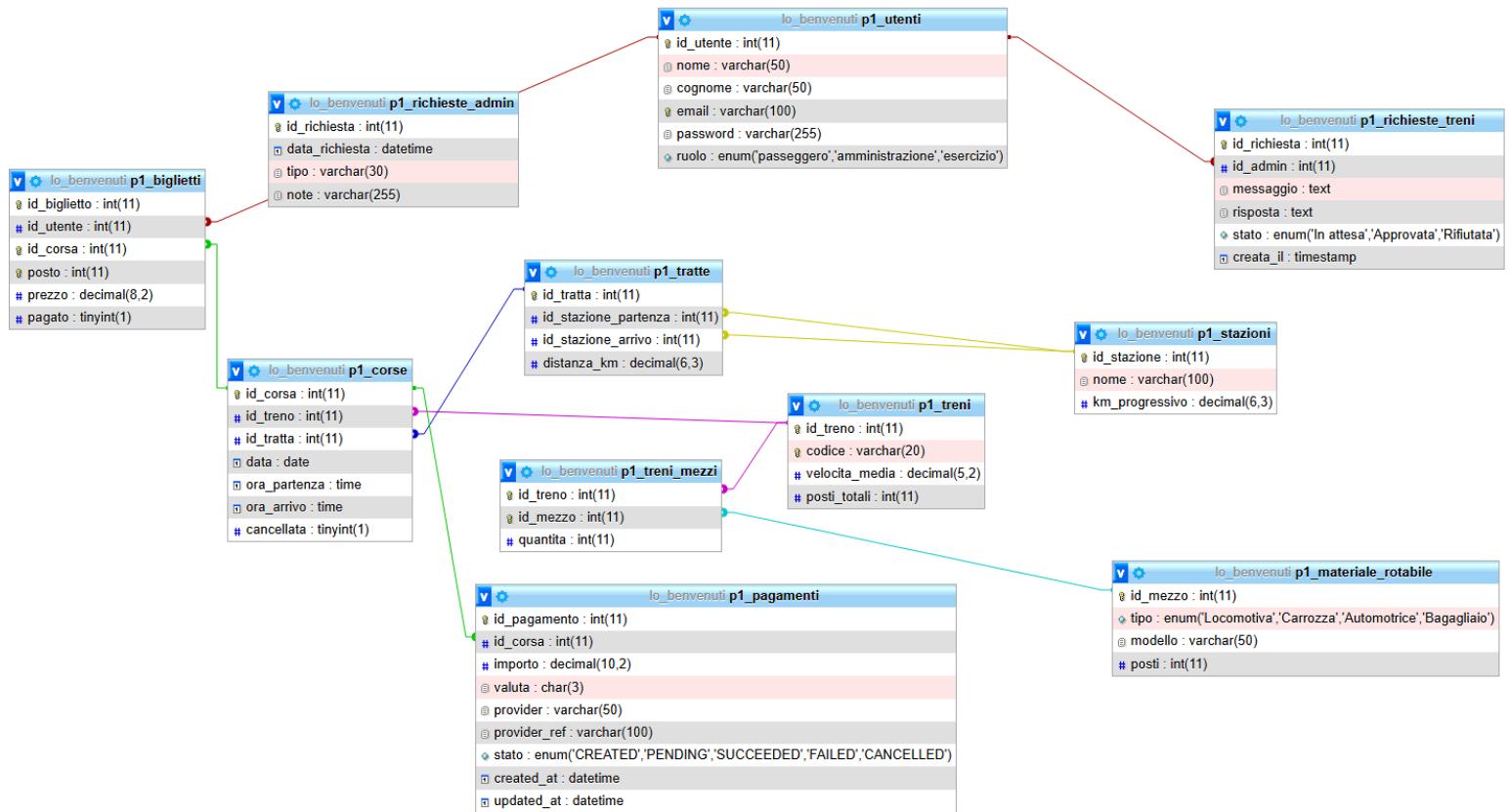
**Richieste\_treni**(id\_richiesta, id\_admin, messaggio, risposta, stato, creata\_il)

## Fase 4: Progettazione fisica

Questa è l'ultima fase del progetto. Essa ha lo scopo di tradurre gli schemi sviluppati in precedenza nel database effettivo e di illustrare gli strumenti operativi che permettono la sua creazione. Ha come finalità l'adattamento della base di dati ad un contesto pratico.

### Implementazione della base di dati

Viene riportato l'hardcopy di phpmyAdmin delle tabelle che popolano il nostro database:



## Script SQL

Per completezza allegiamo il codice SQL delle tabelle appena mostrate:

```
-- SCHEMA SFT

CREATE TABLE IF NOT EXISTS p1_utenti (
    id_utente INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    cognome VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    ruolo ENUM('passeggero','amministrazione','esercizio') NOT NULL DEFAULT 'passeggero'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p1_stazioni (
    id_stazione INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    km_progressivo DECIMAL(6,3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p1_treni (
    id_treno INT AUTO_INCREMENT PRIMARY KEY,
    codice VARCHAR(20) NOT NULL UNIQUE,
    velocita_media DECIMAL(5,2) NOT NULL DEFAULT 50.00,
    posti_totali INT NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p1_materiale_rotabile (
    id_mezzo INT AUTO_INCREMENT PRIMARY KEY,
    tipo ENUM('Locomotiva','Carrozza','Automotrice','Bagagliaio') NOT NULL,
    modello VARCHAR(50) NOT NULL,
    posti INT NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p1_treni_mezzi (
    id_treno INT NOT NULL,
    id_mezzo INT NOT NULL,
    quantita INT NOT NULL DEFAULT 1,
    PRIMARY KEY (id_treno,id_mezzo),
    CONSTRAINT fk_tm_treno FOREIGN KEY (id_treno) REFERENCES p1_treni(id_treno) ON DELETE CASCADE,
    CONSTRAINT fk_tm_mezzo FOREIGN KEY (id_mezzo) REFERENCES p1_materiale_rotabile(id_mezzo) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```

CREATE TABLE IF NOT EXISTS p1_tratte (
    id_tratta INT AUTO_INCREMENT PRIMARY KEY,
    id_stazione_partenza INT NOT NULL,
    id_stazione_arrivo INT NOT NULL,
    distanza_km DECIMAL(6,3) NOT NULL,
    CONSTRAINT fk_tratta_sp FOREIGN KEY (id_stazione_partenza) REFERENCES p1_stazioni(id_stazione),
    CONSTRAINT fk_tratta_sa FOREIGN KEY (id_stazione_arrivo) REFERENCES p1_stazioni(id_stazione)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p1_corse (
    id_corse INT AUTO_INCREMENT PRIMARY KEY,
    id_treno INT NOT NULL,
    id_tratta INT NOT NULL,
    data DATE NOT NULL,
    ora_partenza TIME NOT NULL,
    ora_arrivo TIME NOT NULL,
    cancellata TINYINT(1) NOT NULL DEFAULT 0,
    CONSTRAINT fk_corsa_treno FOREIGN KEY (id_treno) REFERENCES p1_treni(id_treno),
    CONSTRAINT fk_corsa_tratta FOREIGN KEY (id_tratta) REFERENCES p1_tratte(id_tratta)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p1_biglietti (
    id_biglietto INT AUTO_INCREMENT PRIMARY KEY,
    id_utente INT NOT NULL,
    id_corse INT NOT NULL,
    posto INT NOT NULL,
    prezzo DECIMAL(8,2) NOT NULL,
    pagato tinyint(1) NOT NULL DEFAULT '0',
    CONSTRAINT fk_bigl_utente FOREIGN KEY (id_utente) REFERENCES p1_utenti(id_utente),
    CONSTRAINT fk_bigl_corsa FOREIGN KEY (id_corse) REFERENCES p1_corse(id_corse),
    UNIQUE KEY uq_posto (id_corse, posto)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p1_richieste_admin (
    id_richiesta INT AUTO_INCREMENT PRIMARY KEY,
    data_richiesta DATETIME NOT NULL,
    tipo VARCHAR(30) NOT NULL,
    note VARCHAR(255)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE IF NOT EXISTS p1_richieste_treni (
    id_richiesta INT AUTO_INCREMENT PRIMARY KEY,
    id_admin INT NOT NULL,
    messaggio TEXT NOT NULL,
    risposta TEXT DEFAULT NULL,
    stato ENUM('In attesa','Approvata','Rifiutata') DEFAULT 'In attesa',
    creata_il TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (id_admin) REFERENCES p1_utenti(id_utente)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p1_pagamenti (
    id_pagamento INT AUTO_INCREMENT PRIMARY KEY,
    id_corsa INT NOT NULL,
    importo DECIMAL(10,2) NOT NULL,
    valuta CHAR(3) NOT NULL DEFAULT 'EUR',
    provider VARCHAR(50) NOT NULL,
    provider_ref VARCHAR(100) DEFAULT NULL,
    stato ENUM('CREATED','PENDING','SUCCEEDED','FAILED','CANCELLED') NOT NULL DEFAULT 'CREATED',
    created_at DATETIME NOT NULL,
    updated_at DATETIME NOT NULL,
    CONSTRAINT fk_pagamento_corsa FOREIGN KEY (id_corsa) REFERENCES p1_corse(id_corsa)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```