**A**

**Report**

**Submitted for**

**RAILWAY STALLS MANAGEMENT SYSTEM**

**Submitted by:**

**VARTIKA THAPA (102203900)**

**SIDDHANTH NAGRATH (102203686)**

**ARNAV CHAUDHARY (102203024)**

**DEVANSH (102203)**

**BE Second Year**

**Batch: 2026**

Submitted to- **Mr. ANIL VASHISHT**



**DATABASE MANAGEMENT SYSTEM**

**TIET, Patiala**

**Jan-May 2024**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Title** | **Page no.** |
| **1.** | **Introduction** | **3** |
| **2.** | **ER Diagram** | **5** |
| **3.** | **ER to Table** | **6** |
| **4.** | **Normalization** | **7** |
| **5.** | **SQL/ PL-SQL** | **9** |
| **6.** | **Conclusion** | **17** |
| **7.** | **References** | **18** |

**INTRODUCTION**

The project endeavors to revolutionize the railway station experience by offering an integrated platform that provides comprehensive stall information and facilitates passenger orders. As railway stations witness a continuous surge in foot traffic, efficient management of stalls and catering services becomes pivotal. This project addresses this need by amalgamating databases for platforms, trains, stalls, passengers, and a catalog of stall items.

**Tables and Relationships:**

**1.**  Platforms Table: This table manages information about various platforms within the station, distinguishing between arrival and departure platforms. It serves as a foundational reference for locating stalls within the station.

**2.** Trains Table: Capturing details such as train name, type, arrival, departure times, and stop duration, this table aids in scheduling and coordinating stall operations according to train schedules.

**3.** Stalls Table: The focal point of the project, this table contains stall details including name, type, location (referencing the Platforms table), contact number, owner name, and status (open, closed, or under maintenance). Its relationship with the Platforms table ensures accurate stall location mapping.

**4.** Passengers Table: Storing passenger information such as ID, name, age, gender, and contact number, this table facilitates personalized services and order tracking.

**5.** Catalog Table: This table contains item details available at stalls, including stall ID (referencing the Stalls table), item name, type, price, and availability status. It enables passengers to browse stall offerings and place orders efficiently.

**Community Benefits:**

**1.** Enhanced Passenger Experience: By providing detailed stall information and facilitating orders, the project simplifies the station experience for passengers, reducing time spent searching for stalls and waiting in queues.

**2.** Increased Efficiency: Stall owners can better manage their inventory and staffing levels based on passenger flow and train schedules, leading to improved service efficiency and reduced wastage.

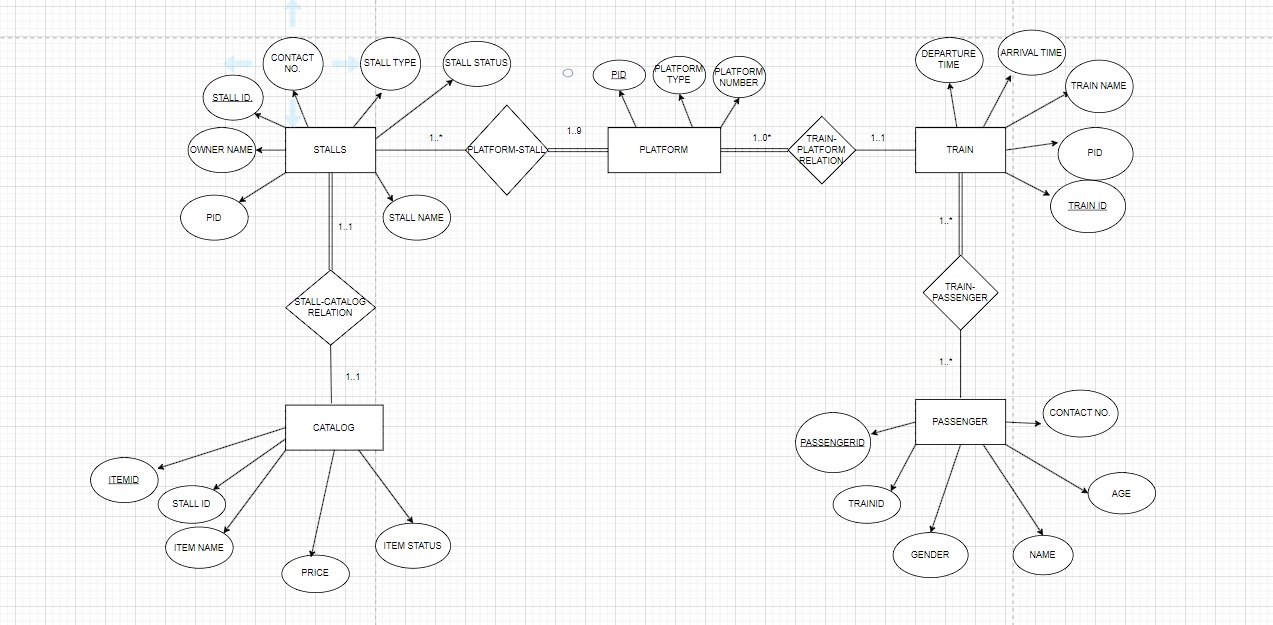
**3.** Economic Opportunities: The project creates opportunities for small businesses by promoting their stalls to a wider audience of railway passengers, thereby fostering local economic development.

**4.** Data-Driven Decision Making: Railway authorities can leverage the collected data to analyze passenger preferences, optimize stall placements, and tailor services to meet evolving demands, thereby improving overall station management.

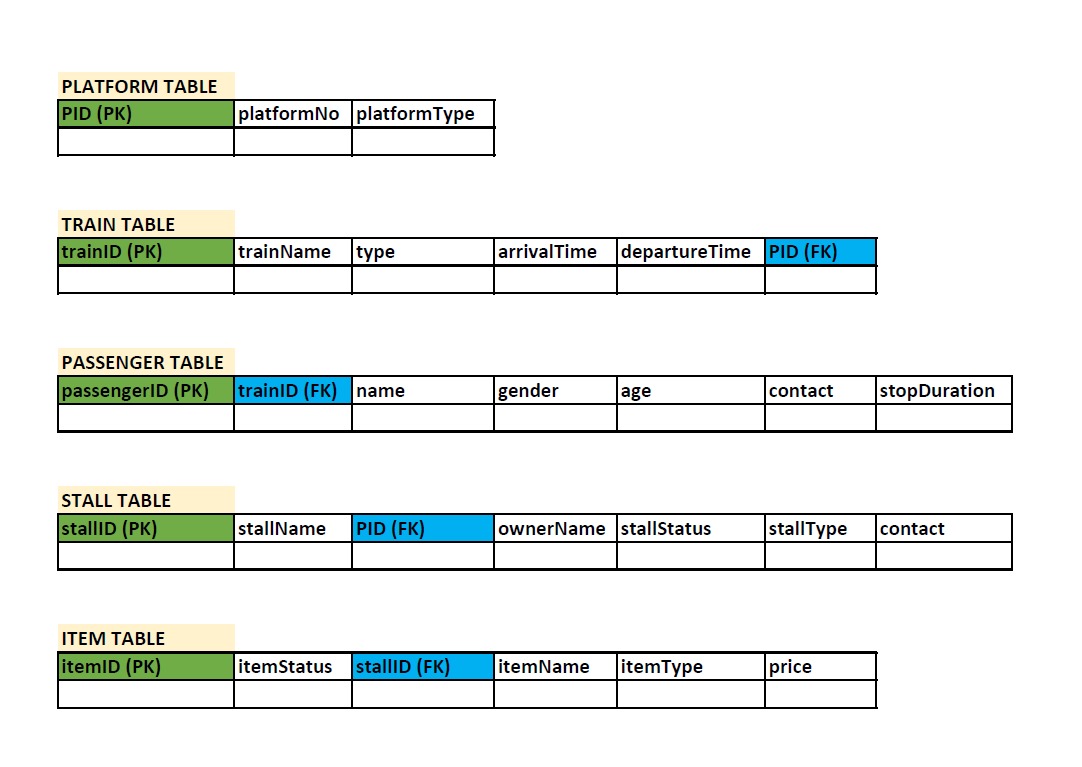
**5.** Accessibility: By centralizing stall information and order placement, the project ensures accessibility for passengers with diverse needs, including those with disabilities or language barriers, fostering inclusivity within the station environment.

By centralizing this information, passengers can easily access details about stalls available at the station and the items they offer. Furthermore, passengers can place orders seamlessly, enhancing their overall experience while navigating through the railway station. This project not only benefits passengers but also aids stall owners in managing their inventory effectively and catering to the diverse needs of travelers.

**ER DIAGRAM**

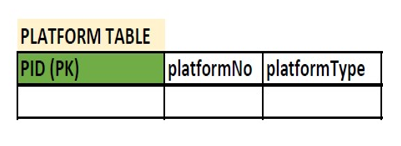


**ER TO TABLE**

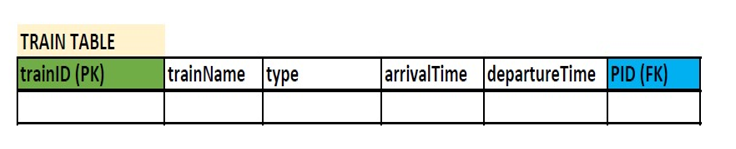


**NORMALIZATION:**

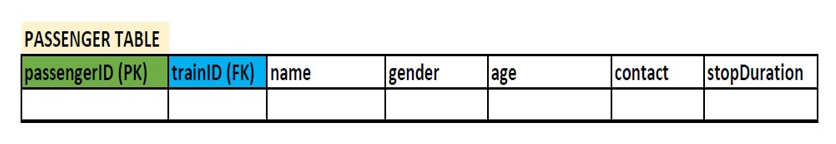
**Platform Table (Already in 3NF)**

****

**Train Table (Already in 3NF)**

****

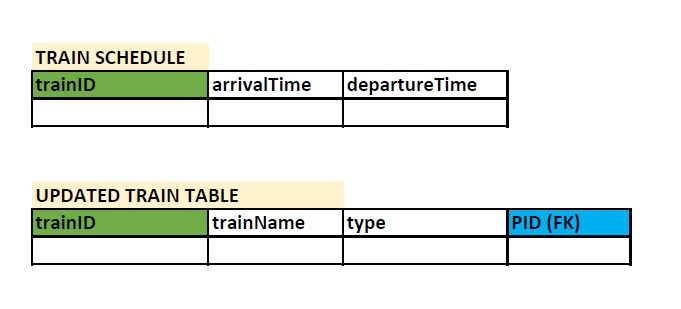
**Passengers Table (In 2NF)**

****

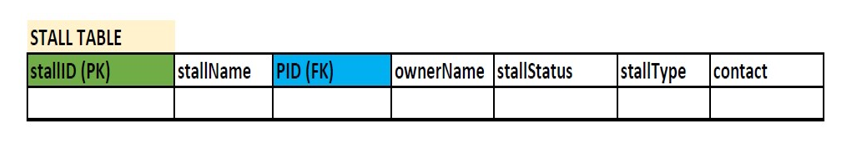
This is not in 3NF because:  
passengerID -> trainID -> stopDuration (Linear Relational Dependence)

To convert it into 3NF:  
1. Altering the Train Table by removing arrivalTime and departureTime.

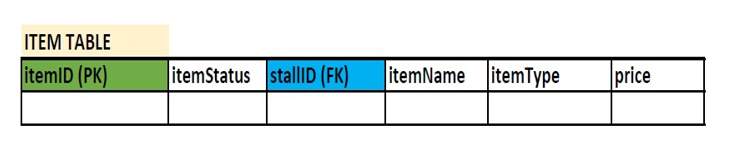
2. Creating a new ‘Train Schedule’ Table as shown in the picture below.



**Stall Table (Already in 3NF)**

****

**Catalog Table (Already in 3NF)**

****

**SQL**

**Creation of Tables**

**Table Platforms :**

CREATE TABLE Platforms (

PlatformID NUMBER PRIMARY KEY,

PlatformNumber VARCHAR2(10) NOT NULL,

PlatformType VARCHAR2(50) CHECK (PlatformType IN (‘Arrival’, ‘Departure’))

);

**Table Trains:**

CREATE TABLE Trains (

TrainID NUMBER PRIMARY KEY,

TrainName VARCHAR2(100) NOT NULL,

TrainType VARCHAR2(50),

ArrivalTime TIMESTAMP,

DepartureTime TIMESTAMP,

PlatformID NUMBER REFERENCES Platforms(PlatformID)

);

**Table Passengers:**

CREATE TABLE Passengers (

PassengerID NUMBER PRIMARY KEY,

Name VARCHAR2(100) NOT NULL,

Age NUMBER,

Gender VARCHAR2(10),

ContactNumber VARCHAR2(20),

TrainID NUMBER REFERENCES Trains(TrainID)

);

**Table Stalls:**

CREATE TABLE Stalls (

StallID NUMBER PRIMARY KEY,

StallName VARCHAR2(100) NOT NULL,

StallType VARCHAR2(50) NOT NULL,

Location NUMBER REFERENCES Platforms(PlatformID),

ContactNumber VARCHAR2(20),

OwnerName VARCHAR2(100),

StallStatus VARCHAR2(50) CHECK (StallStatus IN (‘Open’, ‘Closed’, ‘Under Maintenance’))

);

**Table Catalog:**

CREATE TABLE Catalog (

ItemID NUMBER PRIMARY KEY,

StallID NUMBER REFERENCES Stalls(StallID),

ItemName VARCHAR2(100) NOT NULL,

ItemType VARCHAR2(50),

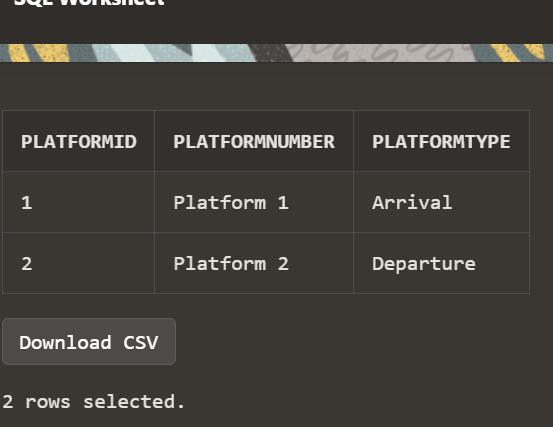
Price NUMBER,

ItemStatus VARCHAR2(100) CHECK (ItemStatus IN (‘IN Stock’, ‘Out of Stock’))

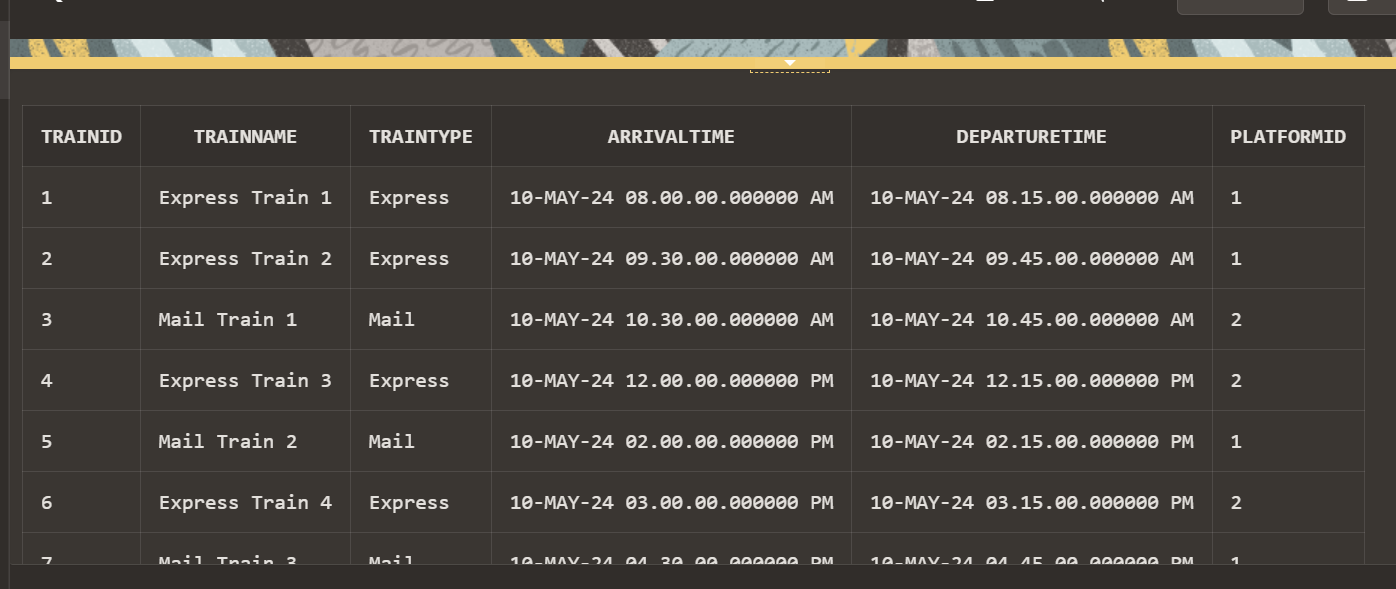
);

**After inserting Dummy Values :**

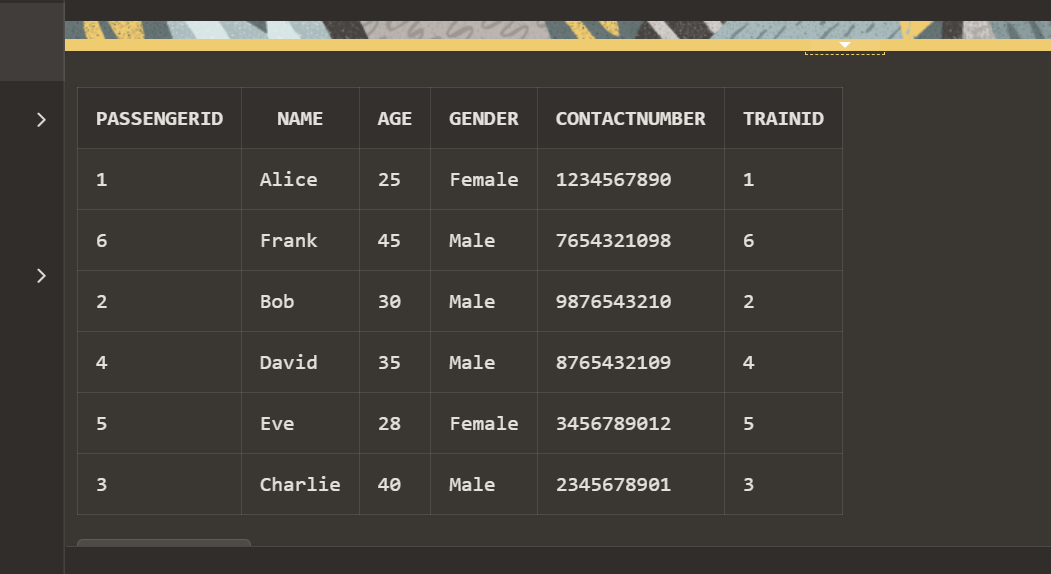
**Table Platforms**

****

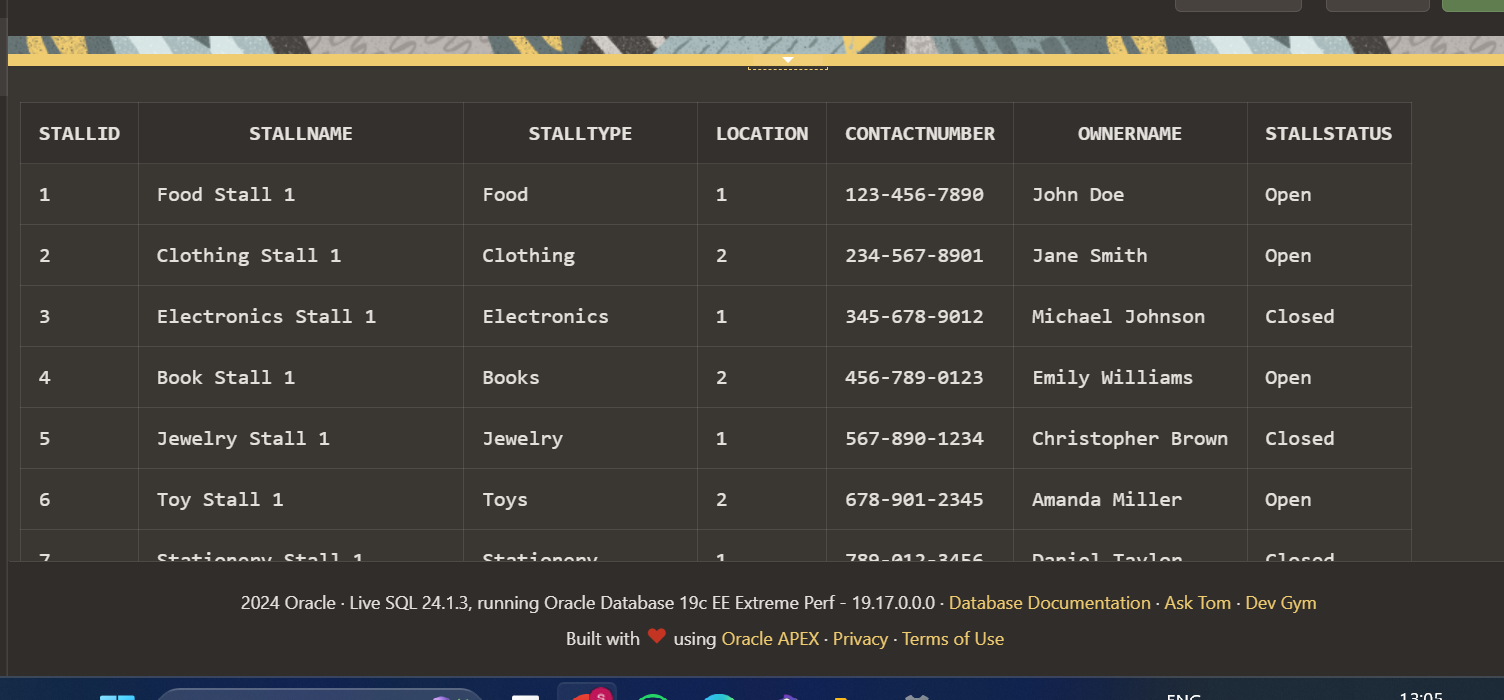
**Table Trains**

****

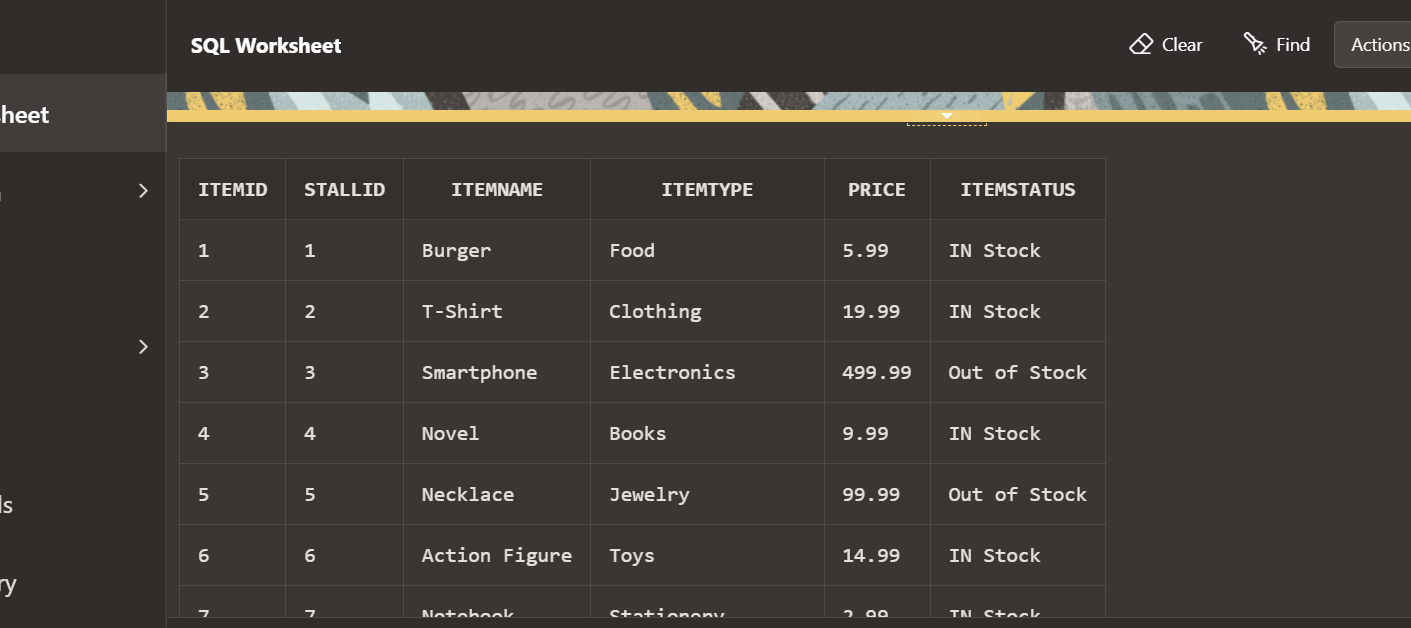
**Table Passengers**

****

**Table Stalls**

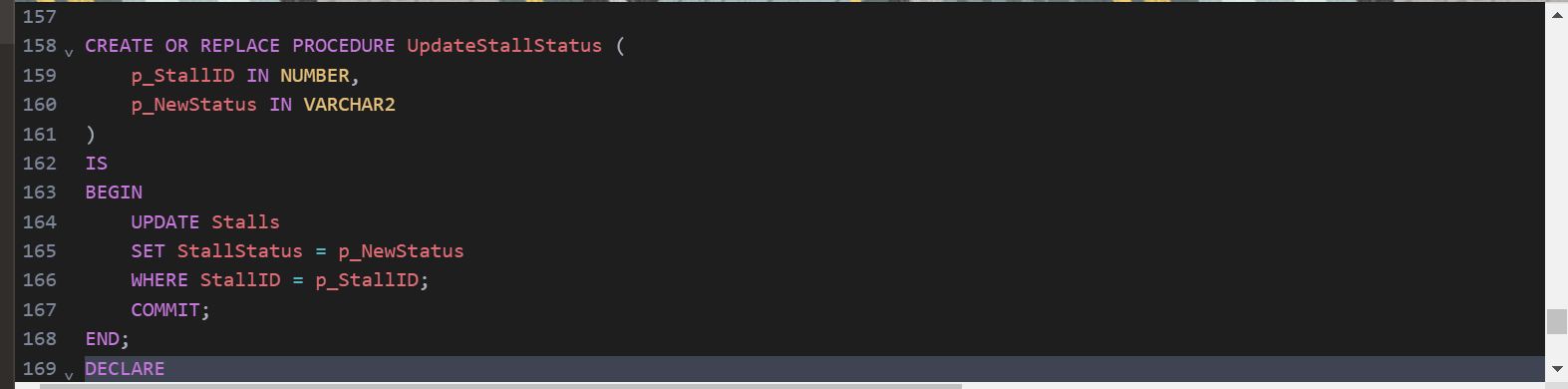
****

**Table Catalog**

****

**PL/SQL**

1. **Procedure to Update Stall Status**

****

Call the procedure and display the Stall Status

A screenshot of a computer

Description automatically generated

1. **Procedure to fetch Passenger Details**



Display the Passenger Details

A screenshot of a computer

Description automatically generated

1. **Function to calculate the total prices of the items**

**A black rectangular object with a black border

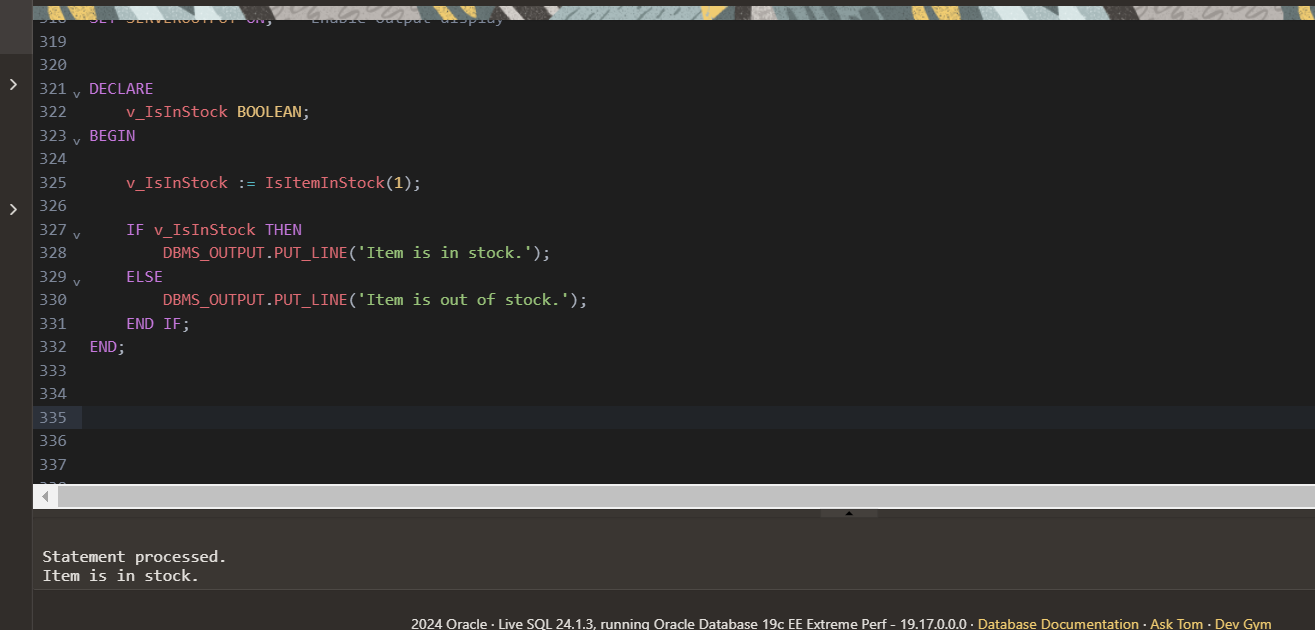
Description automatically generated**

1. **Function to check whether item is in stock or not**

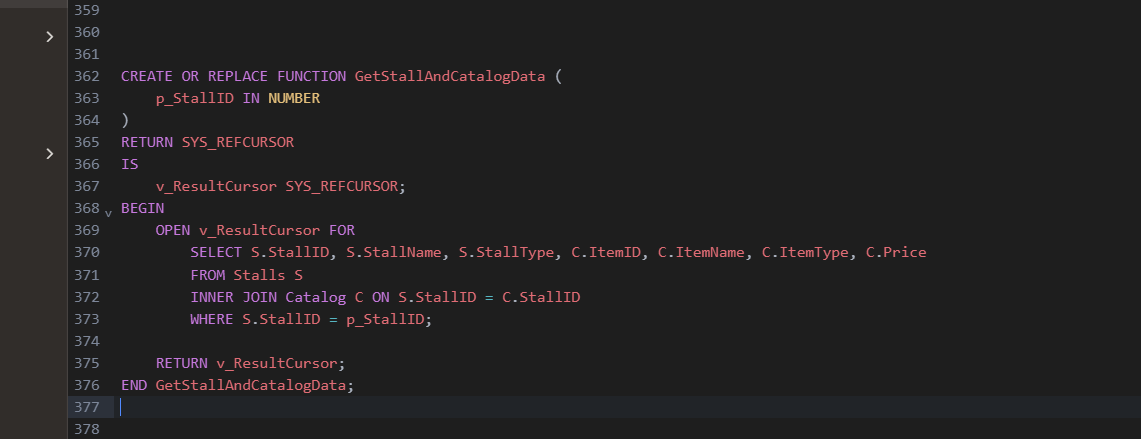
**A screenshot of a computer

Description automatically generated**

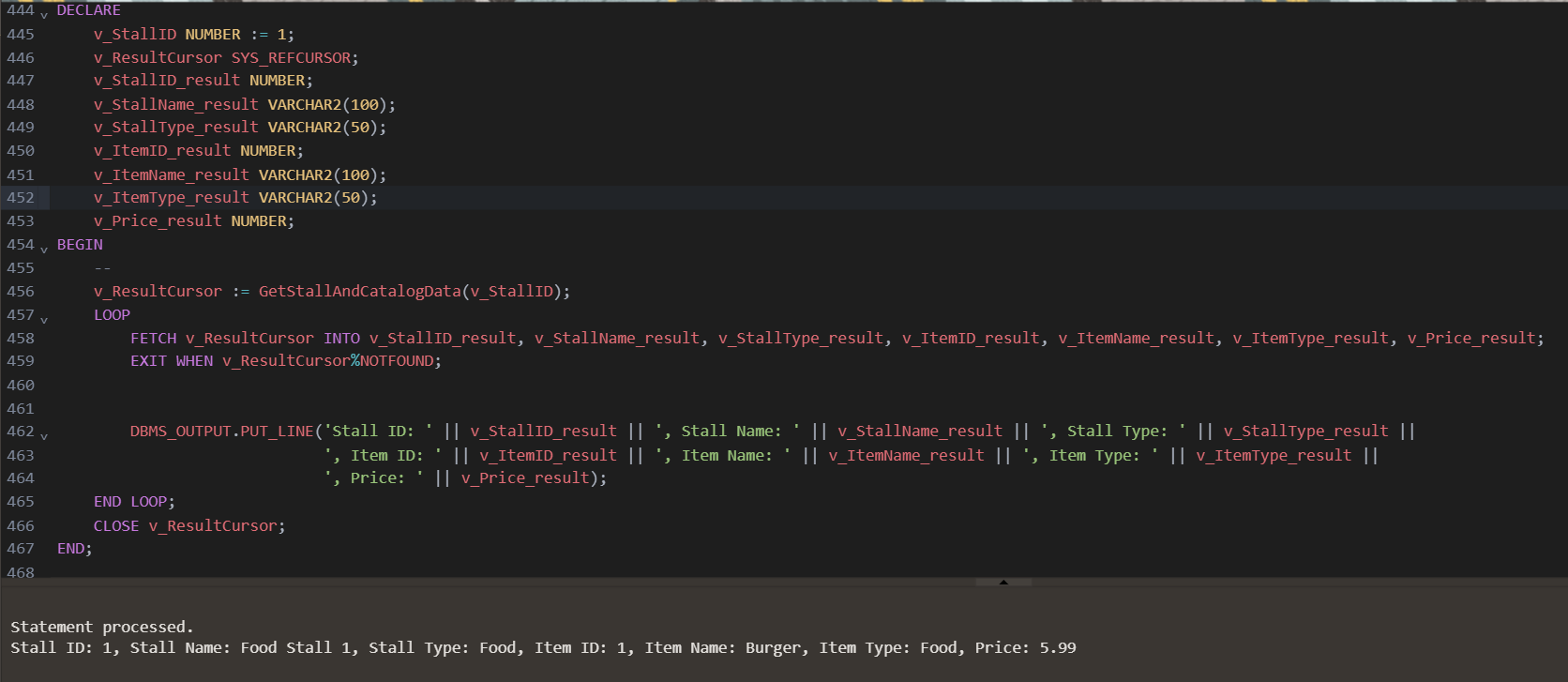
Call the function and display whether item is in stock or not



1. **Function to fetch data from tables Stall and Catalog using joins**

****

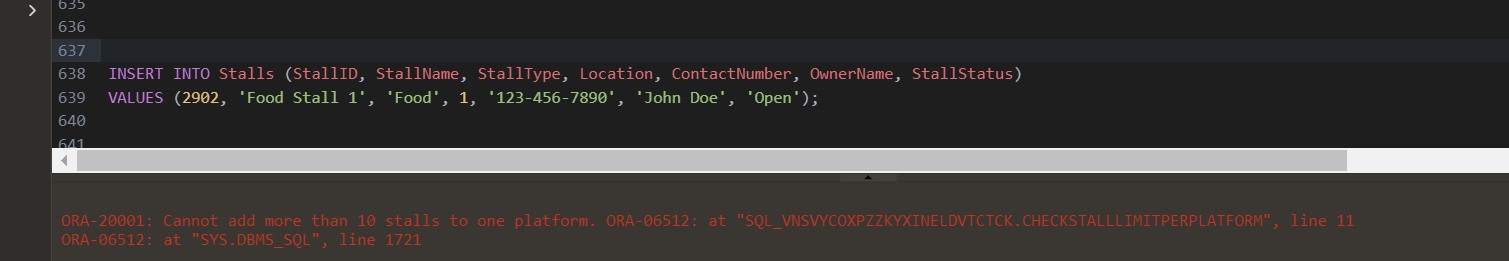
Call the function and display details from the two tables



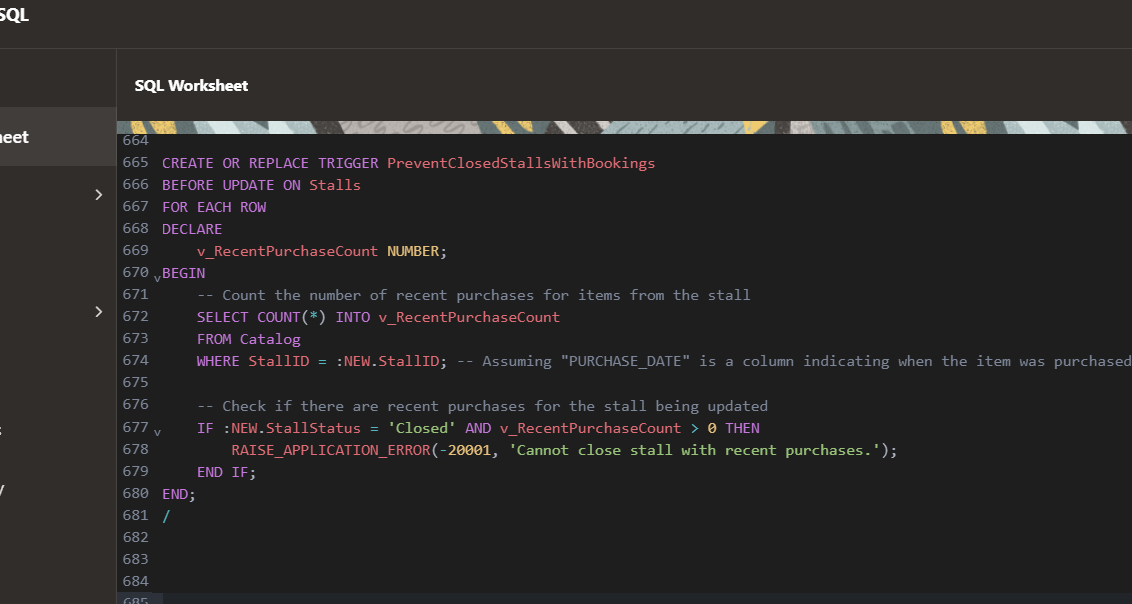
1. **Trigger to check not more than 10 stalls can be installed in a platform**

****

Insert a record into stalls table



1. **Trigger to prevent closing the stalls with recent bookings**



**CONCLUSION**

The project aims to create a robust railway management system, integrating data on stalls, passengers, platforms, trains, and feedback. Its significance lies in facilitating efficient coordination between stakeholders, optimizing operations, and improving passenger experience. By analysing feedback, the system enables data-driven decision-making to enhance services and address concerns promptly. Future prospects include integrating real-time data feeds for train schedules and passenger traffic, predictive analytics for resource allocation, and adapting to emerging trends like e-commerce integration and personalized services. However, to remain effective amidst evolving demands, continuous refinement and expansion are essential. The system must evolve to meet the growing needs of the railway industry and society, ensuring its relevance and efficacy in optimizing railway operations and enhancing passenger satisfaction.

**REFERENCES**

1. Oracle
2. claude.ai by Anthropic (LLM)
3. draw.io (for the ER Diagram)
4. Database System Concepts by Silberscatz, Abraham
5. Simplified Approach to DBMS by Prateek Bhatia