#### CASE STUDY

#### OF

#### RAILWAY MANAGEMENT SYSTEM

#### IN

#### RELATIONAL DATABASE DESIGN

#### Submitted to the Department of Computer Science

#### 

1)

CASE STUDY IN RELATIONAL DATABASE DESIGN

TITLE: **RAILWAY MANAGEMENT SYSTEM**

##### STUDENT’S NAME:

##### 

##### Ayush Singla (Team Leader) (2010991512)

##### 

##### Bhavesh Vaidya (2010991517)

##### Bhavika Aggarwal (2010991519)

Aryan Dev Shourie (2010991496)

GUIDE:

##### Dr. Vikas Solanki.

2)

### Abstract

The objective of this thesis is **Railway Management System**. One case study “Railway System” is presented. Input for this case study is taken from its informal specification to a relational schema using entity-relationship modeling and its translation to relational model, to database schema, to implementation of the database, to interactive SQL querying of the installed database (SQL/Oracle). The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers. The objective of this case study is To prevent a moving train from getting into contact with another train or obstruction and thus to provide**safety to the passengers**, the staff and the rolling stock. To maintain a **safe distance** between two trains on the same line in the same direction to **avoid accidents**. To provide protection against trains **collision** and **derailment** at converging junctions and to give a directional indication at diverging junctions. To provide facilities for safely and efficiently carrying out **shunting operations** in marshalling yards. To allow trains to run at a limited speed while **maintaining** and **repairing** the track. To facilitate the flow of traffic and to increase the carrying capacity of the track.

### Acknowledgments

##### I would like to express my gratitude to all of those who made it possible to complete this thesis, in particular to my supervisor Dr. Vikas Solanki. I would also like to thank my family for their understanding and continuous support.

**Contents**

1. [Abstract](#_TOC_250066) 3)
2. Acknowledgments 4)

[Chapter 1: Introduction](#_TOC_250063)  6)

[Chapter 2: Name of Case Study](#_TOC_250061)  11)

* 1. [Case Study INFORMAL DESCRIPTION](#_TOC_250060) 11)

1.4 Terminologies and Symbols of E-R diagram…………………………………………………………. 8)

* 1. [Case Study LOGICAL MODEL(ER Diagram)](#_TOC_250059)  13)
  2. [Case Study Physical schema](#_TOC_250058)  14)

3.2.1 [Case Study INTERACTIVE QUERIES](#_TOC_250056)  14)

Chapter 3 : Conclusion ………………………………………………………………. 16)

1. [Bibliography 17)](#_TOC_250000)

# Chapter 1: Introduction

* 1. Database Management Systems

Database is an organized collection of Data. The data is typically organized to model aspects of reality in a way that supports process requiring information. A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and the end users or application programs, ensuring that the data is consistently organized and remains easily accessible. The DBMS can offer both logical and physical data independence. Database management system is a software which is used to manage the database. For example: MySQL, ORACLE etc are a very popular commercial database which is used in different applications. DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

* 1. Relational Database Management System

A relational database management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a relational database. RDBMS store data in the form of tables, with most commercial relational database management systems using [Structured Query Language](https://searchsqlserver.techtarget.com/definition/SQL) (SQL) to access the database. However, since SQL was invented after the initial development of the relational model, it is not necessary for RDBMS use. The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering a combination of system performance and ease of implementation. The RDBMS typically provides data dictionaries and metadata collections that are useful in data handling. These programmatically support well-defined data structures and relationships. Data storage management is a common capability of the RDBMS, and this has come to be defined by data objects that range from binary large object -- or blob -- strings to stored procedures. Data objects like this extend the scope of basic relational database operations and can be handled in a variety of ways in different RDBMS.

1.3 Feasibility Study

A **feasibility study** is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

1.3.1 Technical Feasibility Study

The technical aspect of a Feasibility Study helps you to determine the efficacy of the proposed project by examining the details of your intended process, including labour and materials, logistics and technology related to producing, delivering and tracking the products or services you intend to develop. The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. A business is considered to be technically feasible if it has the necessary expertise, infrastructure and capital to develop, install and operate the proposed system.

1.3.2 Operational Feasibility Study

Operational Feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. This Railway System project is used to prevent a moving train from getting into contact with another train or obstruction and thus to provide**safety to the passengers**, the staff and the rolling stock. To maintain a **safe distance** between two trains on the same line in the same direction to **avoid accidents**. To provide protection against trains **collision** and **derailment** at converging junctions and to give a directional indication at diverging junctions.

1.3.3 Economical Feasibility Study

The purpose of an Economical Feasibility Study is to demonstrate the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency, other state agencies, and the general public as a whole. The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers.

1.4 Symbol Representations of ER Diagram

1. -> Entity
2. -> Relationship

1. -> Attribute
2. -> Weak Entity
3. -> Weak Entity Relationship

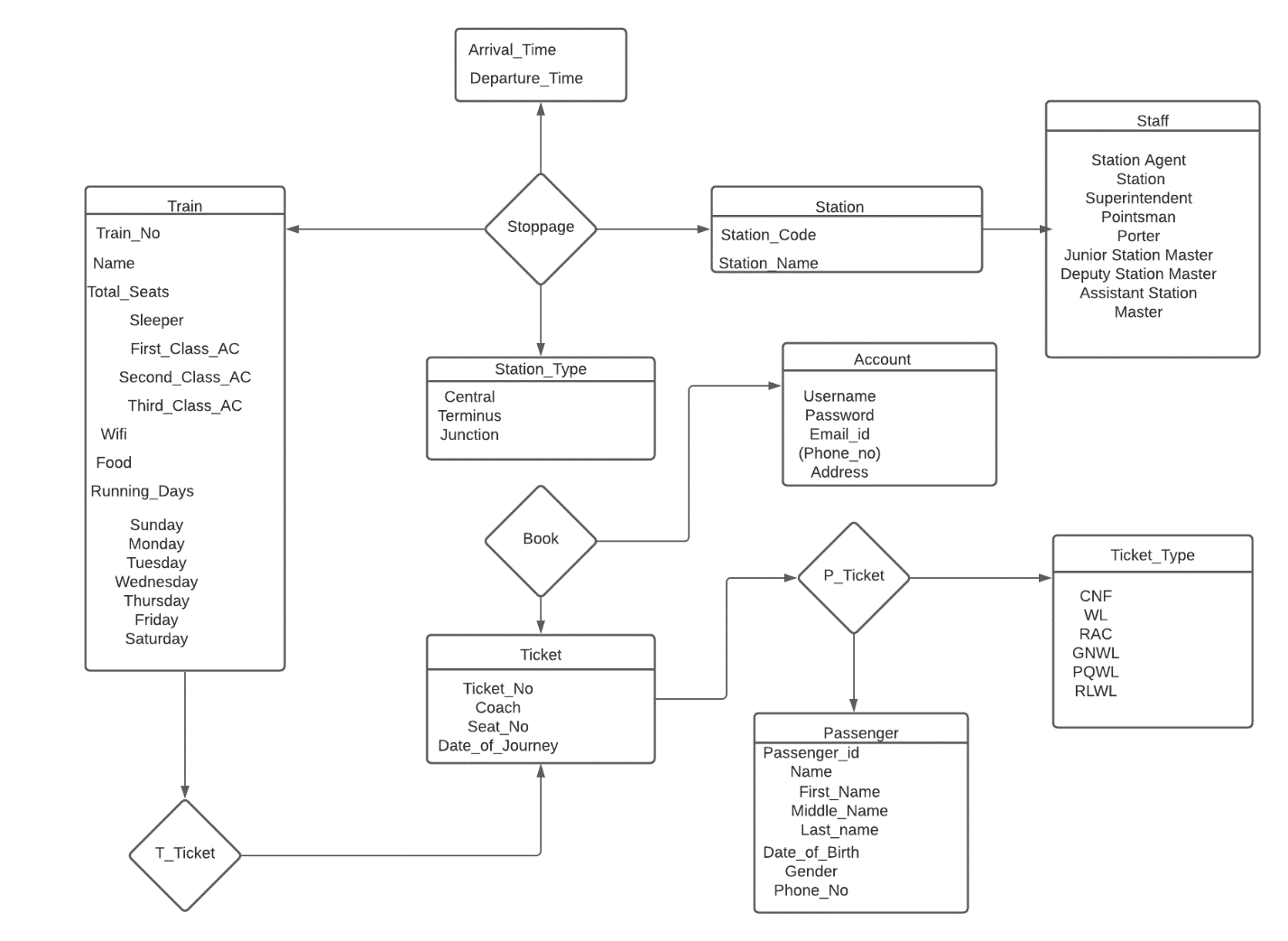
vi)

-> Multivalued Attribute

vii)

-> Key Attribute

* 1. ER Diagram



* 1. Brief introduction of case study

The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers.

* 1. Objective of the case study

To prevent a moving train from getting into contact with another train or obstruction and thus to provide**safety to the passengers**, the staff and the rolling stock. To maintain a **safe distance** between two trains on the same line in the same direction to **avoid accidents**. To provide protection against trains **collision** and **derailment** at converging junctions and to give a directional indication at diverging junctions. To provide facilities for safely and efficiently carrying out **shunting operations** in marshalling yards. To allow trains to run at a limited speed while **maintaining** and **repairing** the track. To facilitate the flow of traffic and to increase the carrying capacity of the track.

The structure of case study:

##### An informal specification of the project in simple English.

1. An Entity-Relationship (ER) model is constructed using **Adobe Illustrator** (name of software)
2. The oracle database schema is produced based on the ER model.

##### Several SQL queries that can be used to query of the installed database are presented.

# Chapter 2: Name of Case study

### Case Study Informal Description

The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers.

**2.1.1 FIRST NORMAL FORM:**

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values. The above schema is in 1NF since all the attributes are atomic and not multivalued. Since a passenger could have multiple phone numbers, it would violate the 1NF rules. Hence we have created a separate table called contact to handle this.

**2.1.2 SECOND NORMAL FORM:**

A table is said to be in 2NF if both the following conditions hold:

-Table is in 1NF (First normal form)

-No non-prime attribute is dependent on the proper subset of any candidate key of

table.

If in Passenger table we consider ticket\_no and first\_name as the candidate key, then

date\_of\_birth would depend only on the name and it would violate the 2NF.

**2.1.3 THIRD NORMAL FORM:**

A table design is said to be in 3NF if both the following conditions hold:

-Table must be in 2NF

-Transitive functional dependency of non-prime attribute on any super key should be

removed.

Our schema follows the above rules and hence is in 3NF.

A trigger has been created which is invoked each time a ticket is cancelled. The trigger helps in increasing the number of seats in a coach after cancellation.

delimiter //

create trigger cancellation

before delete on ticket

for each row

BEGIN

set @trainno=old.train\_no;

set @ticketno=old.ticket\_no;

SET @class = (SELECT p.class

FROM PASSENGER p

WHERE p.ticket\_no = @ticketno);

if @class='first class ac' then

UPDATE Train set Seat\_First\_Class\_AC = Seat\_First\_Class\_AC+1 WHERE Train\_No =

@trainno ;

elseif @class='sleeper' then

UPDATE Train set Seat\_Sleeper = Seat\_Sleeper+1 WHERE Train\_No = @trainno ;

elseif @class='second class ac' then

UPDATE Train set Seat\_Second\_Class\_AC = Seat\_Second\_Class\_AC+1 WHERE

Train\_No = @trainno ;

elseif @class='third class ac' then

UPDATE Train set Third\_Class\_AC = Seat\_Third\_Class\_AC+1 WHERE Train\_No =

@trainno ;

end if;

END//

delimiter ;

**2.1.4** **FUNCTIONAL DEPENDENCIES:-**

TRAIN

`Train\_No`->( `Name`, `Seat\_Sleeper` , `Seat\_First\_Class\_AC`, `Seat\_Second\_Class\_AC`,

`Seat\_Third\_Class\_AC`,`Wifi`,`Food`,`Run\_On\_Sunday``Run\_On\_Monday``Run\_On\_Tuesd

ay` `Run\_On\_Wednesday` `Run\_On\_Thursday` `Run\_On\_Friday`,`Run\_On\_Saturday` )

STOPPAGE

( `Train\_No` , `Station\_Code`) -> (`Arrival\_Time` , `Departure\_Time`)

TICKET

`Ticket\_No`-> (`Train\_No` `Date\_of\_Journey` `Username`)

STATION

`Station\_Code` -> `Station\_Name`

PASSENGER

`Passenger\_Id`-> (`First\_Name` `Last\_Name` `Gender` `Phone\_No` `Ticket\_No` `Age`

`Class`)

ACCOUNT

`Username`->( `Password` `Email\_Id` `Address` )

### Case Study Logical Model

This project is about creating a database about the Railway Management System. The railway management system facilitates the passengers to enquire about the trains available on the basis of source and destination, booking and cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, stations, and passengers. The record of the train includes its number, name, days on which it is available etc. Passengers can book their tickets for the train in which seats are available. For this, passengers have to provide the desired train number and the date for which ticket is to be booked. Before booking a ticket for a passenger, the validity of train number and booking date is checked. Once the train number and booking date are validated, it is checked whether the seat is available. If yes, the ticket is booked with confirm status and corresponding ticket No is generated which is stored along with other details of the passenger. The ticket once booked can be cancelled at any time. For this, the passenger has to provide the ticket ID (the unique key). The ticket ID is searched and the corresponding record is deleted.

### Case Study Physical Model (e.g.)

### Train Table:-

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Datatype** | **Size** | **Constraint** |
| Train\_No | INT | - | PRIMARY KEY, NOT NULL |
| Name | VARCHAR | 25 | NOT NULL |
| Seat\_Sleeper | INT | - | NOT NULL |
| Seat\_First\_Class\_AC | INT | - | NOT NULL |
| Seat\_Second\_Class\_AC | INT | - | NOT NULL |
| Seat\_Third\_Class\_AC | INT | - | NOT NULL |
| Wifi | CHAR | 1 | NOT NULL |
| Food | CHAR | 1 | NOT NULL |
| Run\_On\_Sunday | CHAR | 1 | NOT NULL |
| Run\_On\_Monday | CHAR | 1 | NOT NULL |
| Run\_On\_Tuesday | CHAR | 1 | NOT NULL |
| Run\_On\_Wednesday | CHAR | 1 | NOT NULL |
| Run\_On\_Thursday | CHAR | 1 | NOT NULL |
| Run\_On\_Friday | CHAR | 1 | NOT NULL |
| Run\_On\_Saturday | CHAR | 1 | NOT NULL |

CREATE TABLE `Train` (

`Train\_No` int(6) NOT NULL DEFAULT '0',

`Name` varchar(25) NOT NULL,

`Seat\_Sleeper` int(4) NOT NULL,

`Seat\_First\_Class\_AC` int(4) NOT NULL,

`Seat\_Second\_Class\_AC` int(4) NOT NULL,

`Seat\_Third\_Class\_AC` int(4) NOT NULL,

`Wifi` char(1) NOT NULL,

`Food` char(1) NOT NULL,

`Run\_On\_Sunday` char(1) NOT NULL,

`Run\_On\_Monday` char(1) NOT NULL,

`Run\_On\_Tuesday` char(1) NOT NULL,

`Run\_On\_Wednesday` char(1) NOT NULL,

`Run\_On\_Thursday` char(1) NOT NULL,

`Run\_On\_Friday` char(1) NOT NULL,

`Run\_On\_Saturday` char(1) NOT NULL,

PRIMARY KEY (`Train\_No`)

### 3.2.1) Case Study Interactive Queries (e.g)

##### Query 1: Find the phone number of the user whose email ID is [ajitesh@pes.edu](mailto:ajitesh@pes.edu)

##### SELECT Phone\_no FROM Contact WHERE username IN (SELECT Username FROM account WHERE Email\_id=’ajitesh@pes.edu’);

##### Query 2: Find the time at which the train leaves the New Delhi Station.

CREATE VIEW f(Departure\_time) AS

SELECT Departure\_time

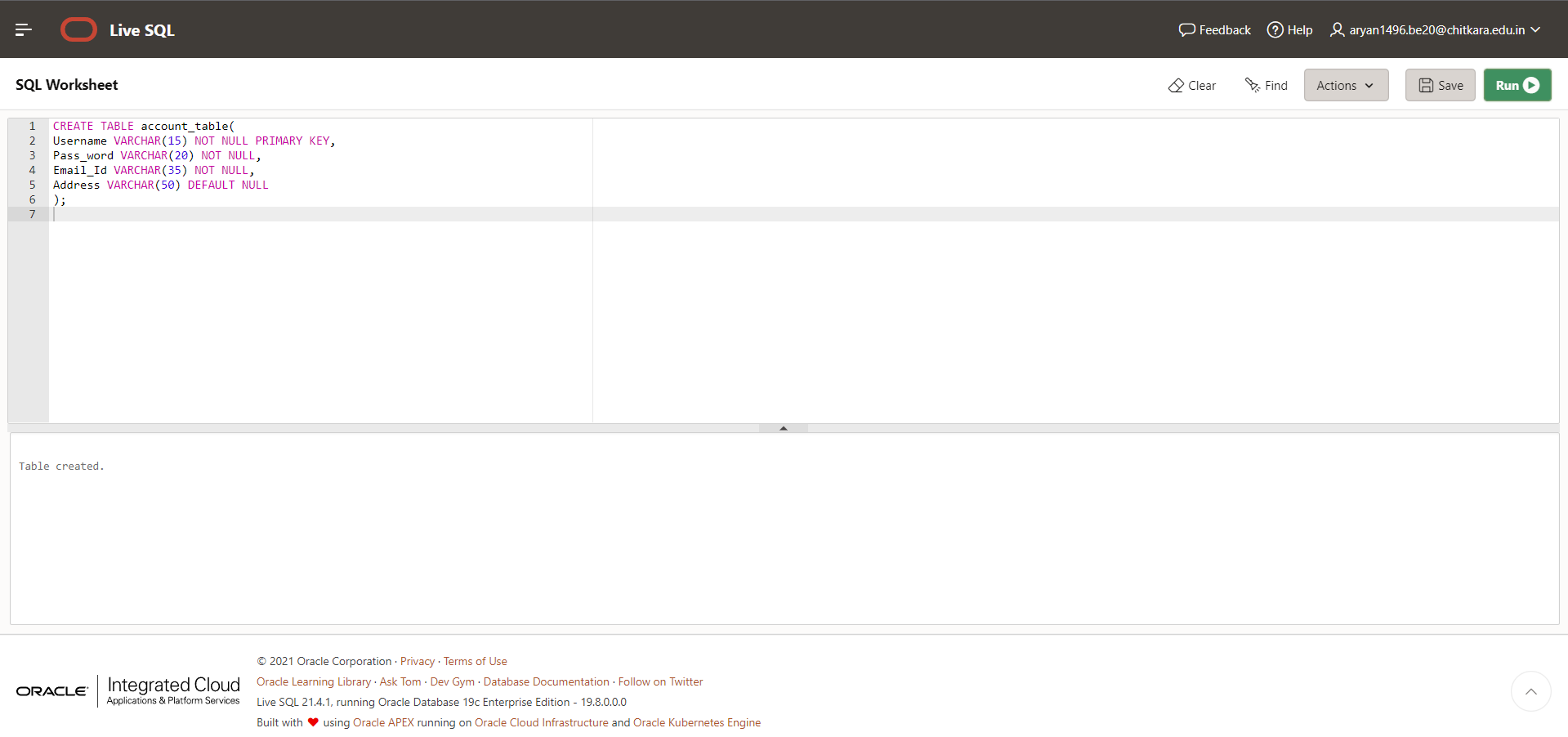
FROM Stoppage

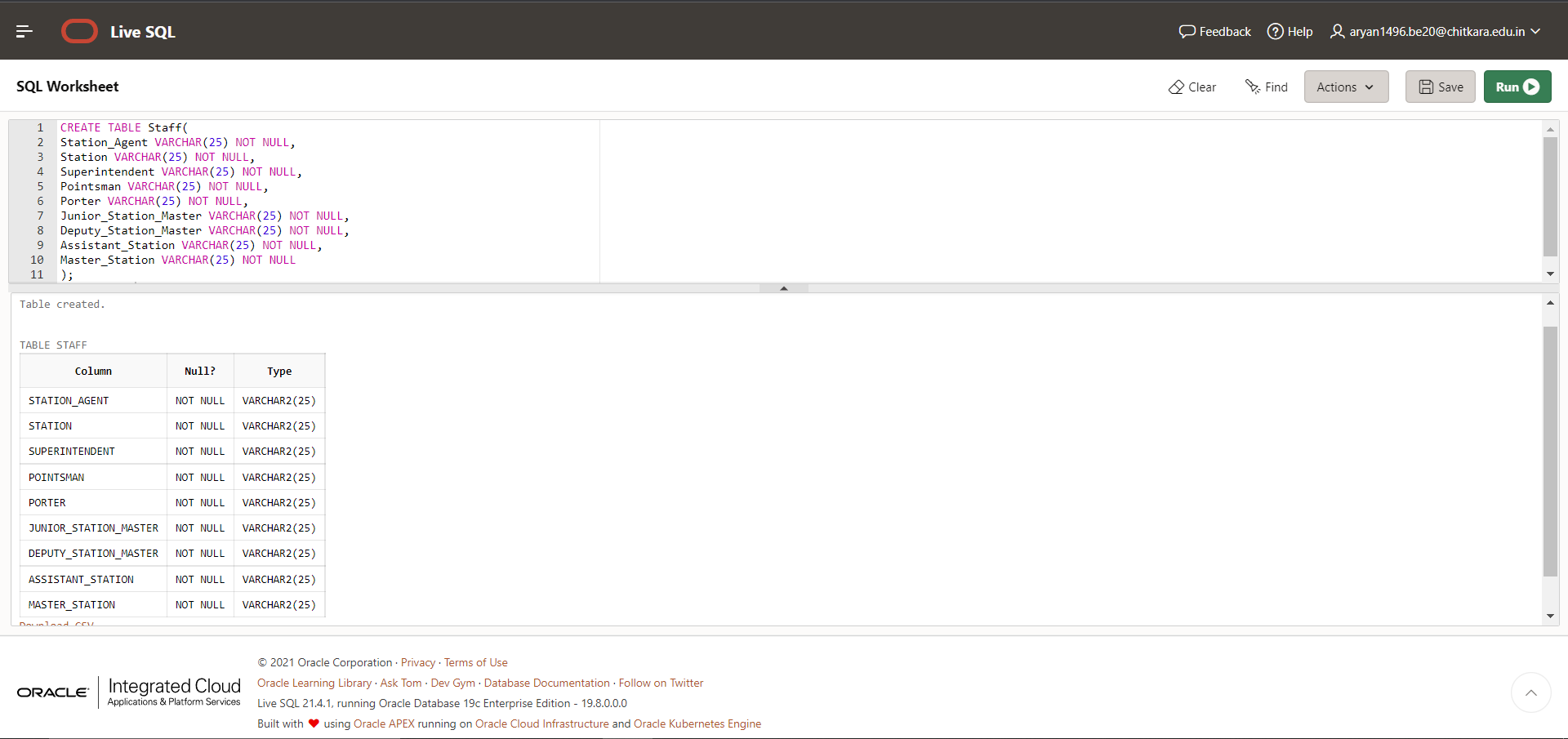
WHERE Station\_Code IN (SELECT Station\_code FROM station WHERE Station\_Name=’New Delhi’);

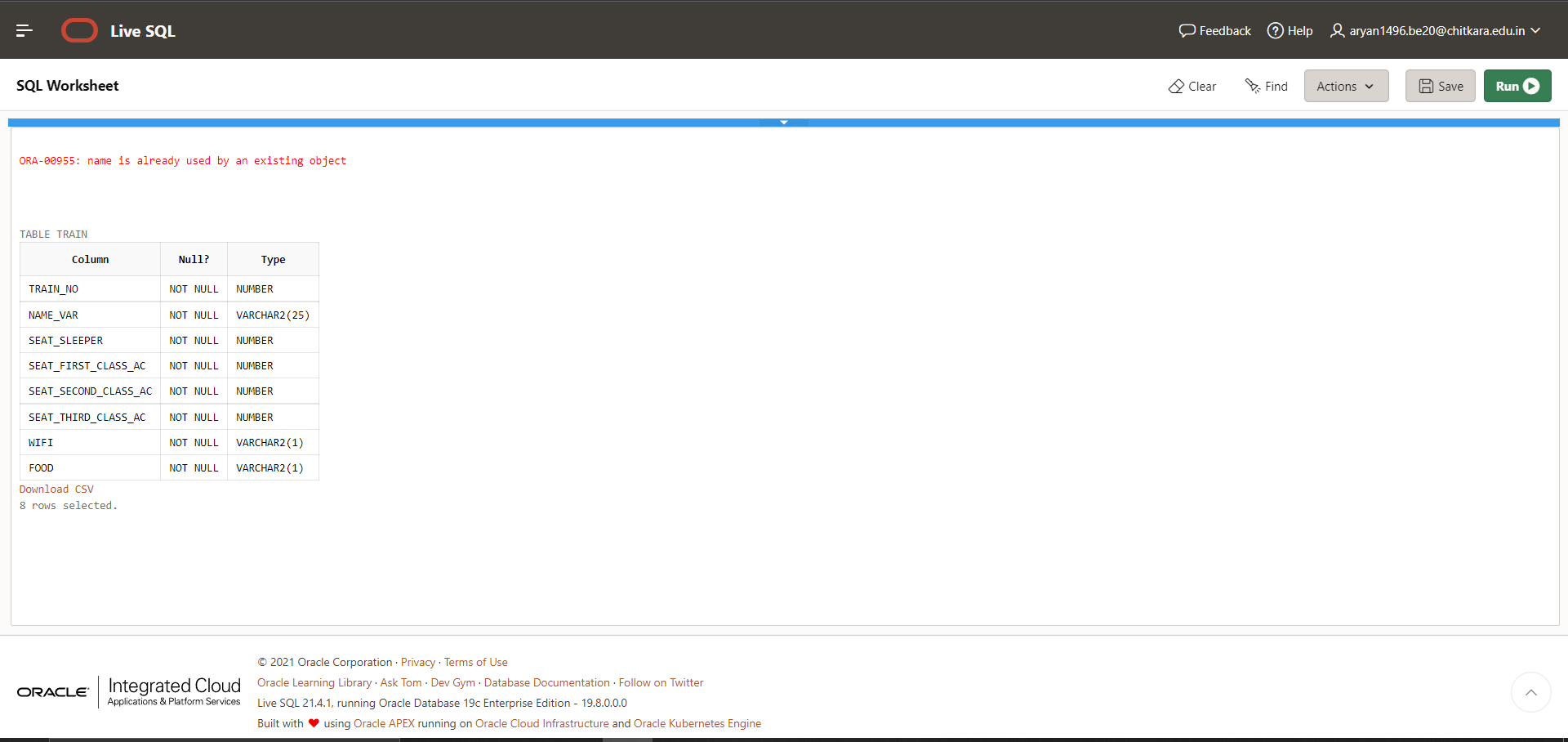
SELECT \* FROM f;

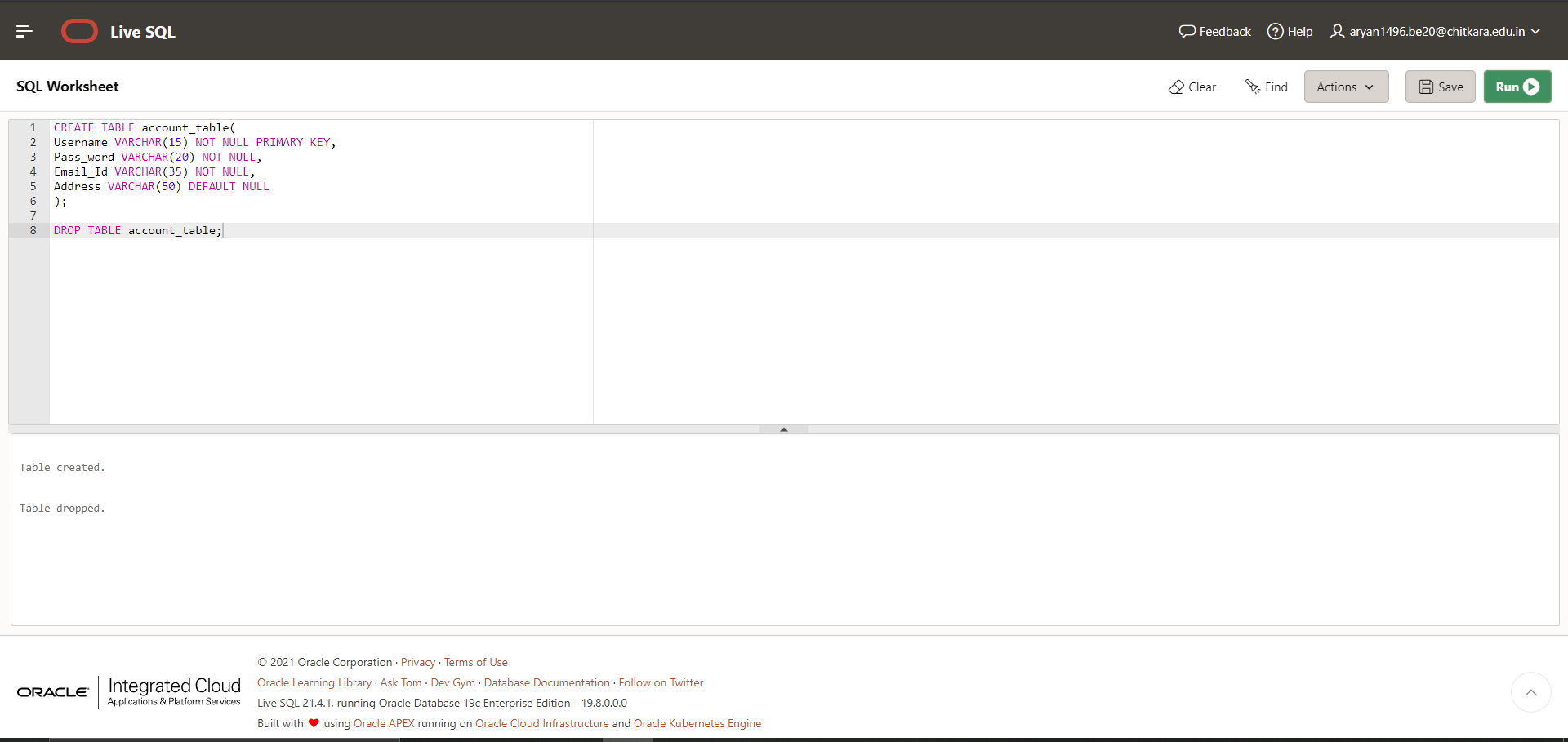
SELECT MAX(Departure\_time) FROM f;

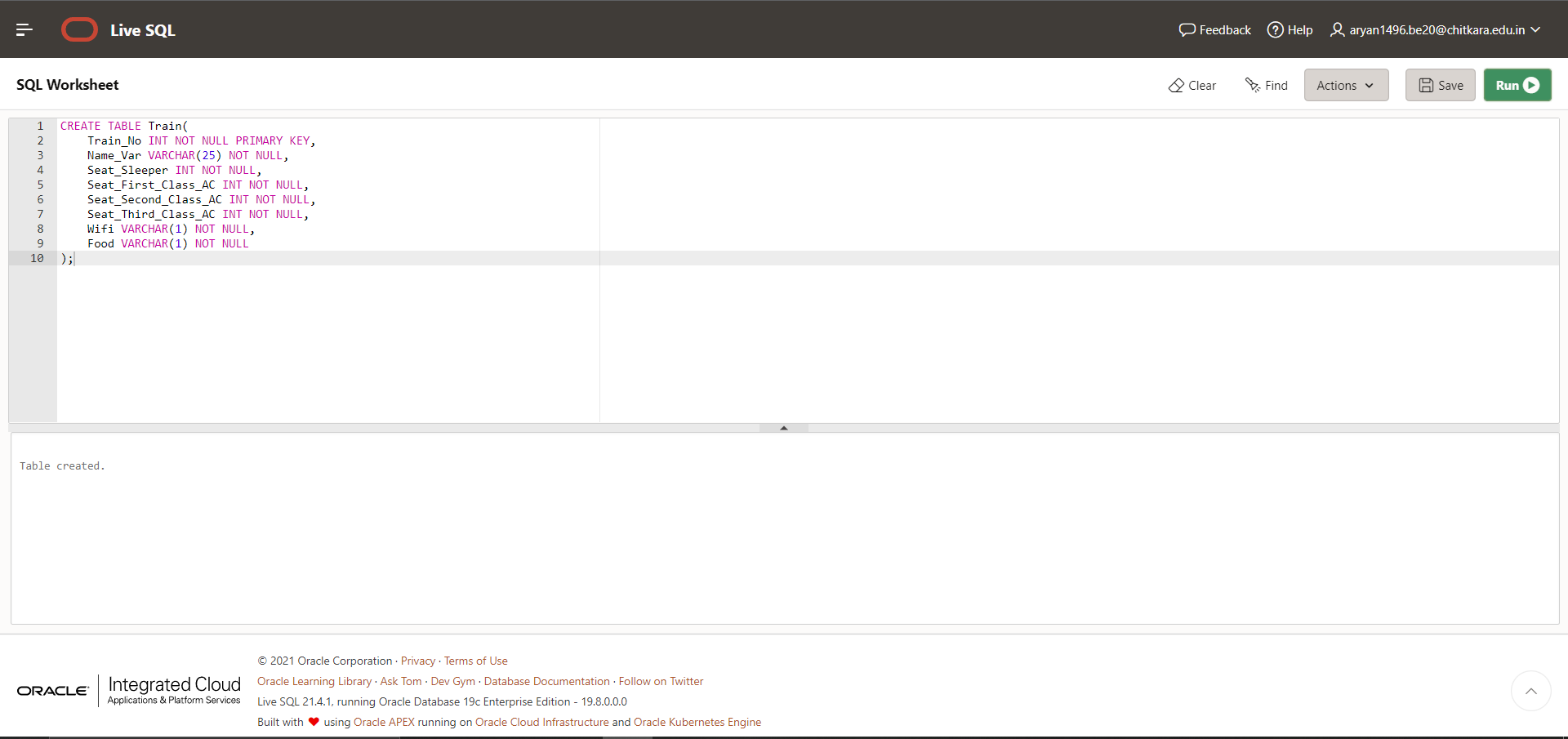
### 3.2.2 Output Screen Shots

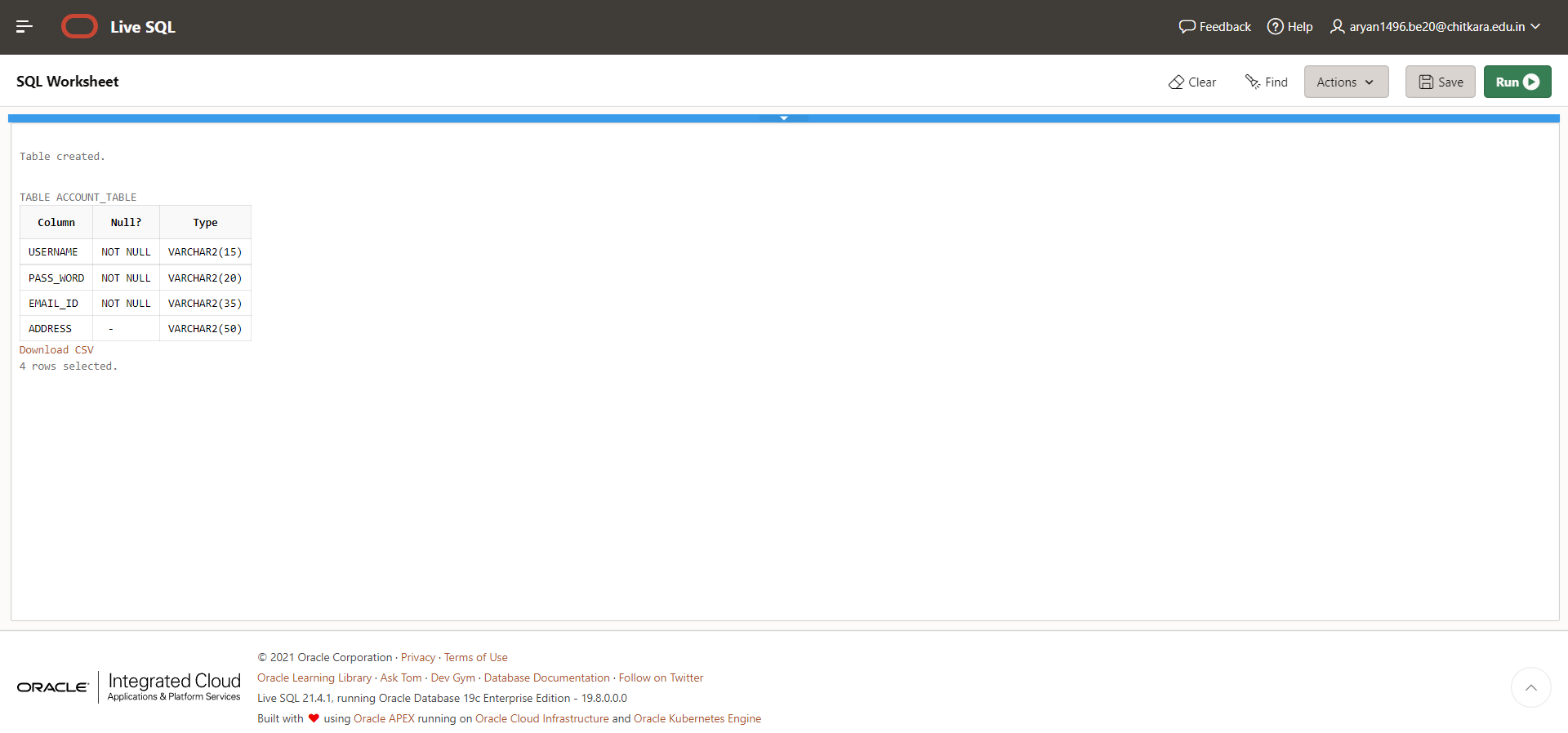


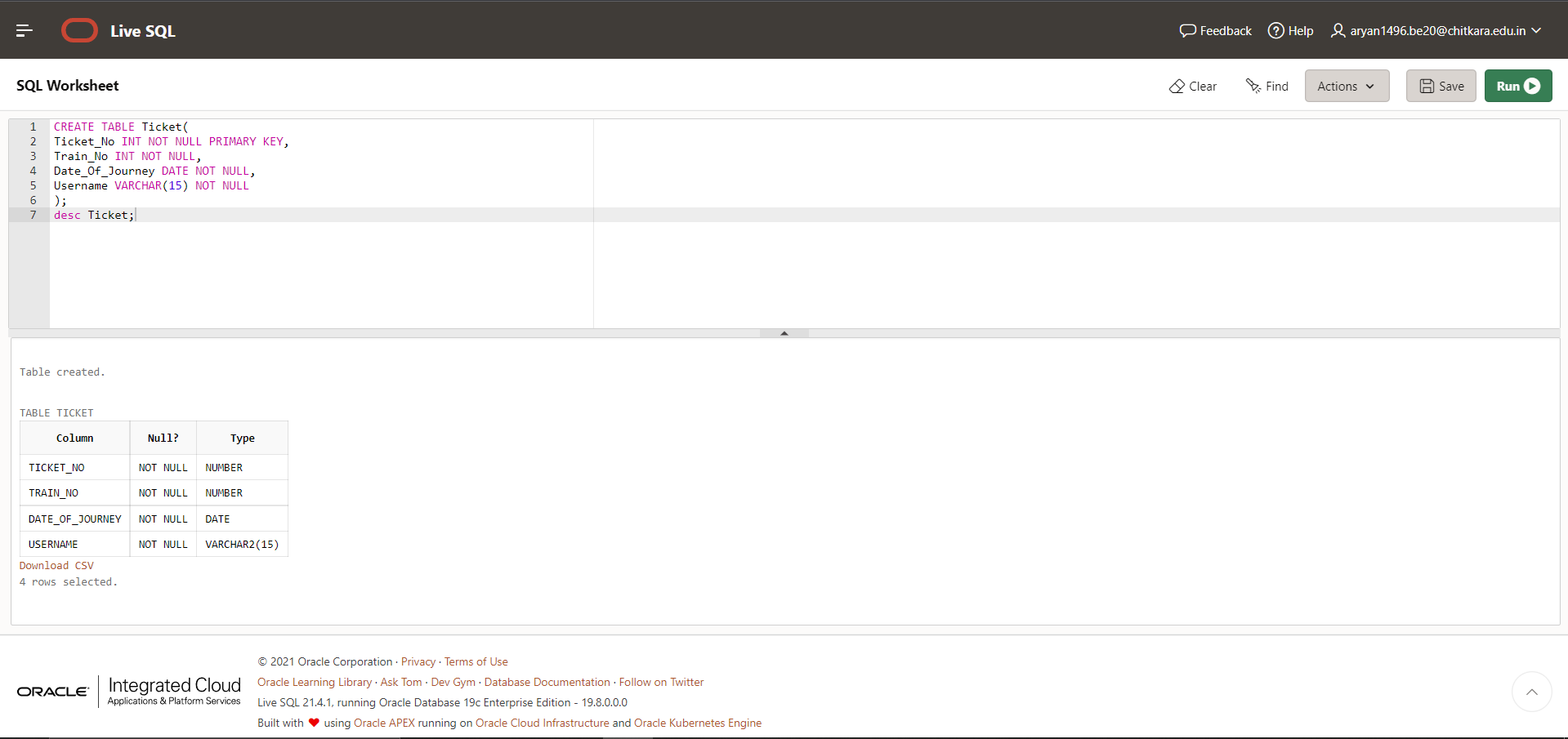


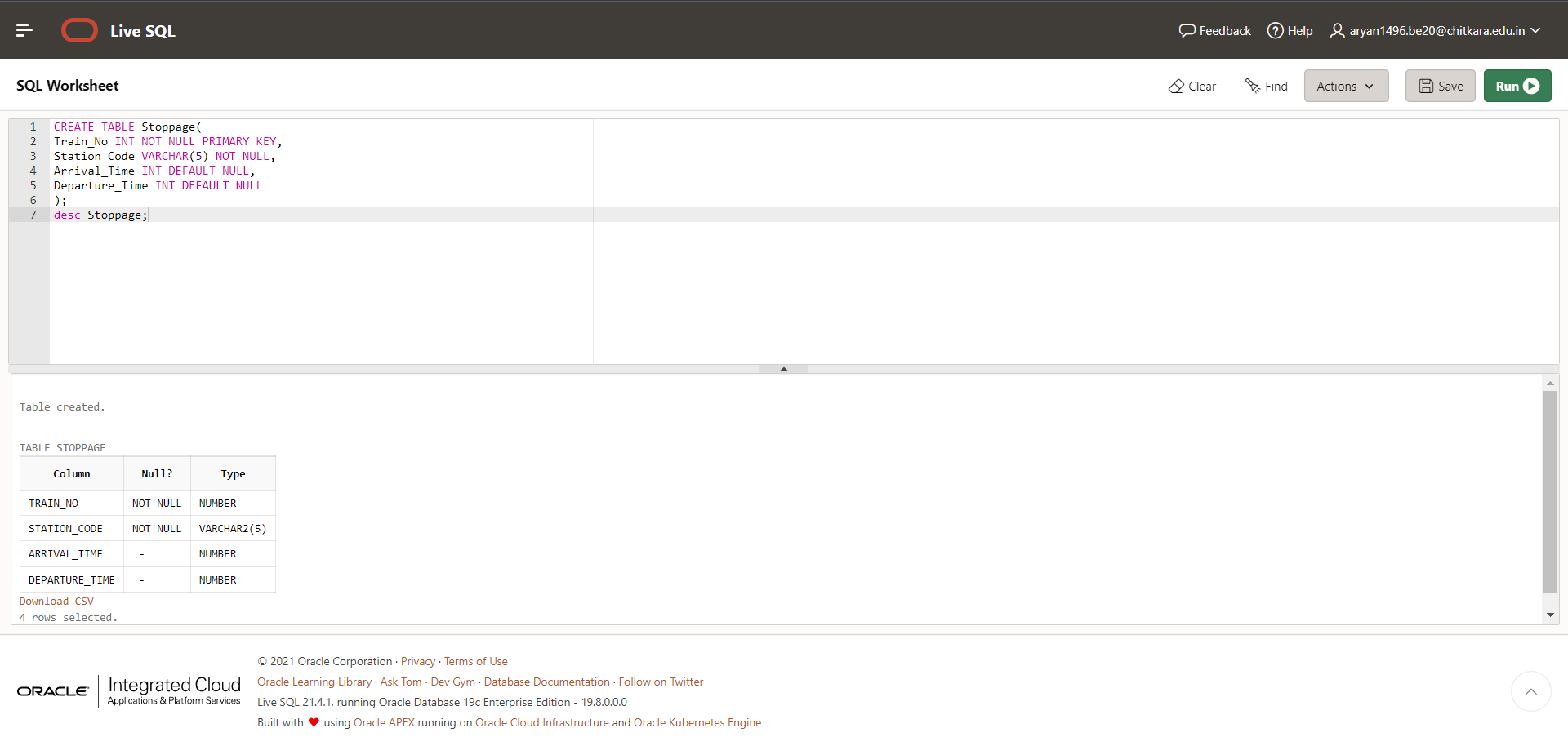


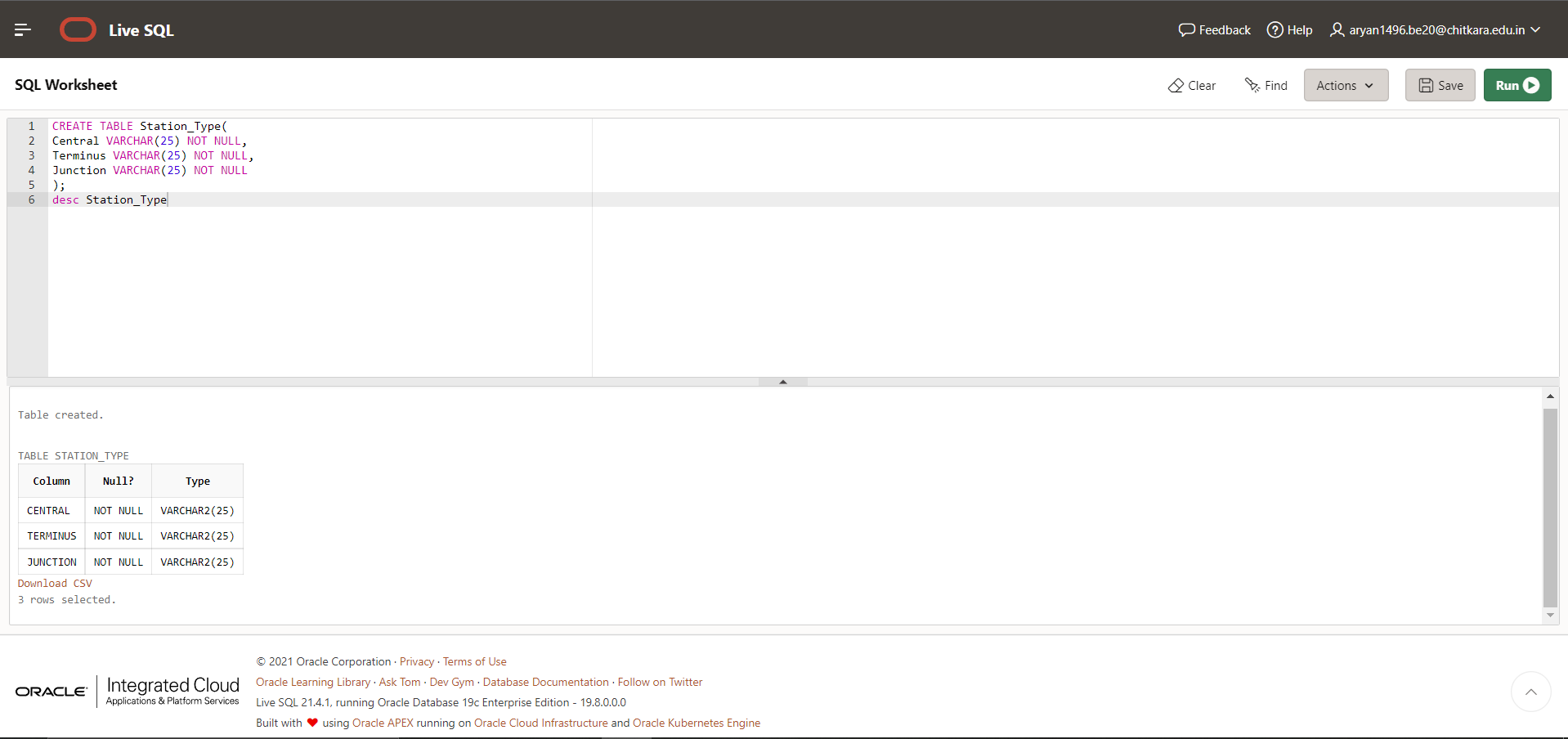












**Chapter 3: Conclusion and Future Work**

This Database is helpful for the applications which facilitate passengers to book the train tickets and check the details of trains and their status from their place itself. It avoids inconveniences of going to the Railway Station for each and every query they get.

Our system can successfully give information on any train, find trains running between two stations, book tickets and cancel tickets. This system could be used for official train booking. However the several other features could be added like booking meals on trains etc. Also the payment gateways have to be implemented to make sure the transactions happen securely.

## Bibliography

## <https://www.javatpoint.com/dbms-tutorial>

## <https://searchdatamanagement.techtarget.com/definition/RDBMS-relational-database-management-system>

## <https://gateoverflow.in/>

## Appendix

## The roles of different members in the Team are described as below:-

## Bhavika Aggarwal – Normalization, Database Queries and Research.

## Aryan Dev Shourie – Research, Database Queries and E.R. Diagram.

## Ayush Singla – Normalization, Database Queries.

## Bhavesh Vaidya – Research, Database Queries.