A Project Report On

**Railway Reservation System**

Developed By:

**DIRGH PATEL (IT-086)**

**DHRUV PATEL (IT-085)**

Guided By:

**Prof.Vidhi B. Chaudhari**

**Department of Information Technology**

**Faculty of Technology**

**DDU University**



**Department of Information Technology**

**Faculty of Technology, Dharmsinh Desai University**

**College Road, Nadiad – 387001**

**DHARMSINH DESAI UNIVERSITY**

## **NADIAD-387001, GUJARAT**



**CERTIFICATE**

This is to certify that the project entitled “**Railway Reservation System**” is a bonafide report of the work carried out by

1. **Mr.Dirgh patel ,** Student ID No : **18ITUOS123**
2. **Mr.Dhruv patel** , Student ID No : **18ITUON122**

of the Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during the academic year 2020-2021.

### **Prof. Vidhi B. Chaudhari**

(Project Guide)

Department of Information Technology, Faculty of Technology,

Dharmsinh Desai University, Nadiad Date: 31/11/20

**TABLE OF CONTENTS**

**1. SYSTEM OVERVIEW 2**

1.1 Current system 2

1.2 Problems 3

1.3 Advantages of the Proposed system (over current) 3

**2. E-R DIAGRAM 4**

2.1 Entities 4

2.2 Relationships 4

2.3 ER Diagram 5

**3. DATA DICTIONARY 6**

**4. SCHEMA DIAGRAM 9**

**5. DATABASE IMPLEMENTATION** 10

5.1 Create Schema 11

5.2 Insert Data values 15

5.3 Queries 18

5.4 PL/SQL Blocks (Procedures and Functions) 19

5.5 Function 20

5.6 Cursors 21

5.7 Trigger 23

**6. BIBLIOGRAPHY 24**

**1. SYSTEM OVERVIEW**

**1.1 CURRENT SYSTEM**

A Railway Reservation System is one which helps us to reserve or cancel seats in train. When we want to travel from one place to another place. Since it is difficult to maintain a record of all information manually, we have created it with the help of an E-R diagram, many tables. It consists of several steps to be followed and many forms to be maintained i.e for the convenience of the passengers.

**1.2 PROBLEMS**

* Maintenance of a huge amount of Database manually is a very tough due to which passengers are not getting the proper information.
* Passengers had to wait for a long time standing in a queue for the confirmation of the ticket.
* Passengers can't be able to know the proper arrival and departure time which is really very inconvenient.
* It takes a lot of work and time for cancellation.

**1.3 ADVANTAGES OF THE PROPOSED SYSTEM**

Through this system, we try to achieve the below objectives:

* To inform users about their reservation status.
* Available seats and price of ticket.
* Schedules of different trains.
* Provides the details of the various stations from the source to the destination to the passenger.
* To Reduce cost, occur on management.
* To prevent users from getting Invalid Information.
* Train Fare
* Travel Distance.

**2. E-R DIAGRAM**

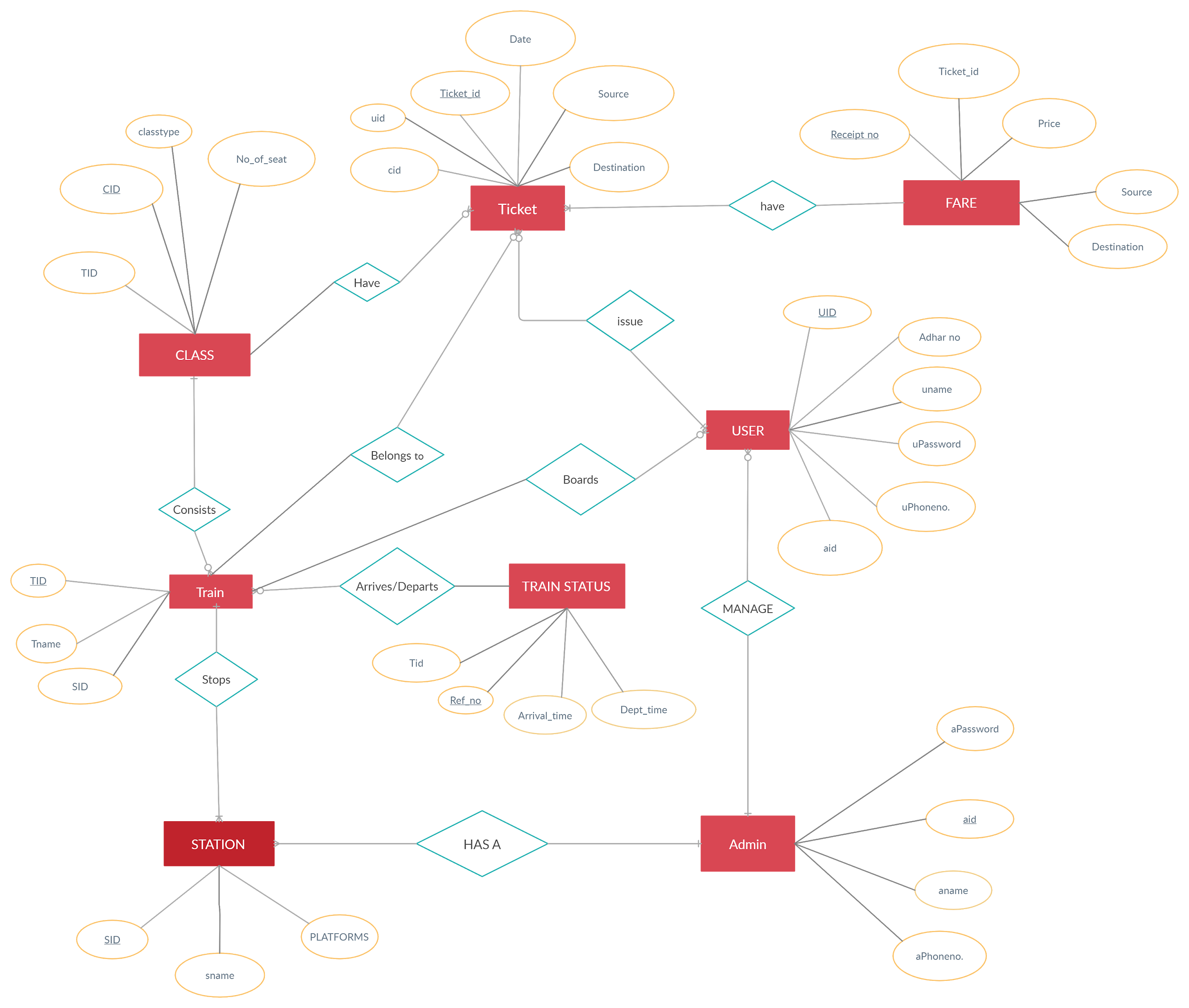
**2.1 ENTITIES & ATTRIBUTES**

* EMPLOYEE: Phone number, Salary, Employee\_id, Name
* USER: Email, Password, Phone number, Name.
* TIME: Arrival and Depart time, Reference number.
* TICKET: Source and Destination, Ticket type, Date, Ticket number,uid
* FARE: Price, Ticket\_id, Receipt number.
* STATION: Station number, id, platform.
* TRAIN: Start and stop station, Train\_no, Train\_name.
* CLASS: Train\_id, type, Seats,classtype.

**2.2 RELATIONSHIP**

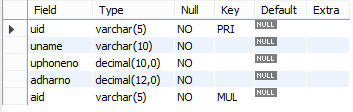
* User: Books: Ticket
* Train: Arrives/Departs: Time
* Train: Consists: Class
* Employee: Helps: Users
* Train : Stops: Station
* Ticket: Belongs to: Train

**2.3 ER DIAGRAM**

****

**3. DATA DICTIONARY**

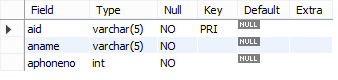
**USER1**

****

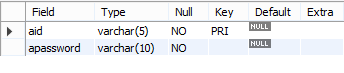
**USER2**

****

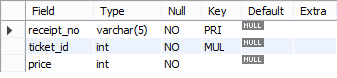
**ADMIN1**

****

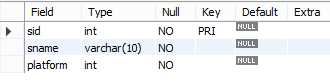
**ADMIN2**

****

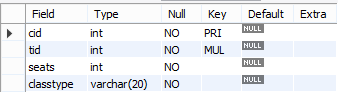
**FARE**

****

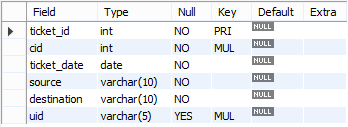
**STATION**

****

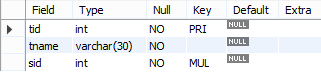
**CLASS**

****

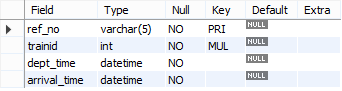
**TICKET**

****

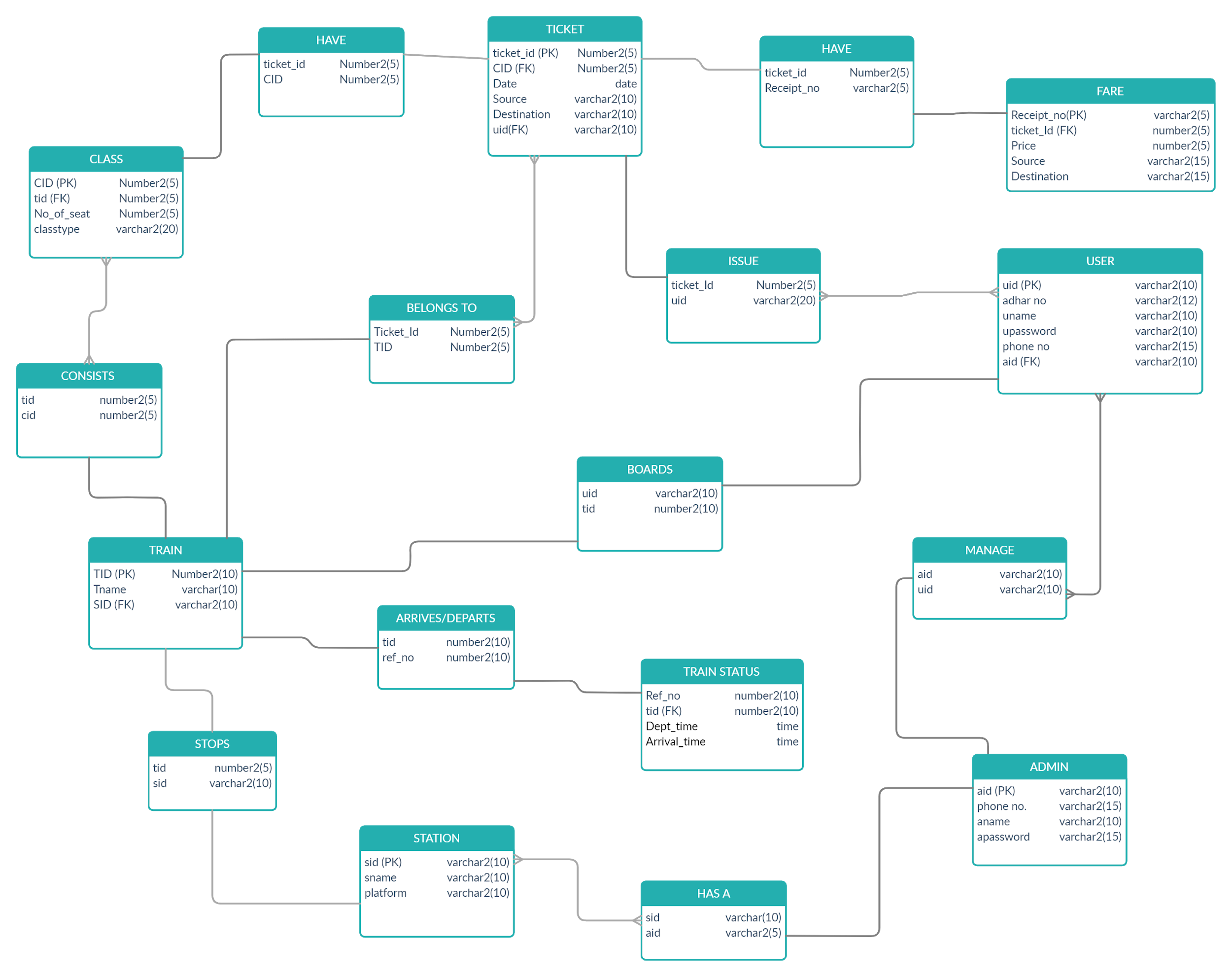
**TRAIN**

****

**TRAIN STATUS**

****

**4. SCHEMA DIAGRAM**

****

**5. DATABASE IMPLEMENTATION**

**5.1 CREATE SCHEMA**

**1.ADMIN1**

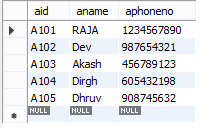
CREATE TABLE `railway`.`admin1` (

`aid` VARCHAR(5) NOT NULL,

`aname` VARCHAR(5) NOT NULL,

`aphoneno` INT(10) NOT NULL,

PRIMARY KEY (`aid`));

****

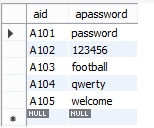
**2. ADMIN2**

CREATE TABLE `railway`.`admin2` (

`aid` VARCHAR(5) NOT NULL,

`password` VARCHAR(10) NOT NULL,

PRIMARY KEY (`aid`));

****

**3.USER1**

CREATE TABLE `railway`.`user1` (

`uid` VARCHAR(5) NOT NULL,

`uname` VARCHAR(10) NOT NULL,

`uphoneno` INT(10) NOT NULL,

`adharno` INT(10) NOT NULL,

`aid` VARCHAR(5) NOT NULL,

PRIMARY KEY (`uid`),

INDEX `aid\_idx` (`aid` ASC) VISIBLE,

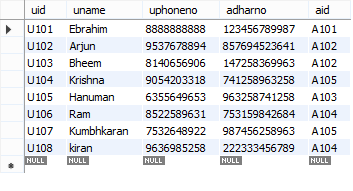
CONSTRAINT `aid`

FOREIGN KEY (`aid`)

REFERENCES `railway`.`admin1` (`aid`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

****

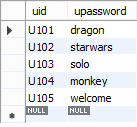
**4. USER2**

CREATE TABLE `railway`.`user2` (

`uid` INT NOT NULL,

`upassword` VARCHAR(10) NOT NULL,

PRIMARY KEY (`uid`));



**5. FARE**

CREATE TABLE `railway`.`fare` (

`receipt\_no` VARCHAR(5) NOT NULL,

`ticket\_id` INT(5) NOT NULL,

`price` INT(5) NOT NULL,

PRIMARY KEY (`receipt\_no`),

INDEX `ticket\_id\_idx` (`ticket\_id` ASC) VISIBLE,

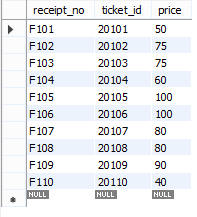
CONSTRAINT `ticket\_id`

FOREIGN KEY (`ticket\_id`)

REFERENCES `railway`.`ticket` (`ticket\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);



**6. TICKET**

CREATE TABLE `railway`.`ticket` (

`ticket\_id` INT(5) NOT NULL,

`cid` INT(5) NOT NULL,

`ticket\_date` DATE NOT NULL,

`source` VARCHAR(10) NOT NULL,

`destination` VARCHAR(10) NOT NULL,

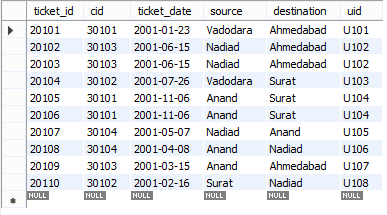
PRIMARY KEY (`ticket\_id`),

INDEX `cid\_idx` (`cid` ASC) VISIBLE,

CONSTRAINT `cid`

FOREIGN KEY (`cid`)

REFERENCES `railway`.`class` (`cid`)



**7. CLASS**

CREATE TABLE `railway`.`class` (

`cid` INT(5) NOT NULL,

`tid` INT(5) NOT NULL,

`seats` INT(5) NOT NULL,

PRIMARY KEY (`cid`),

INDEX `tid\_idx` (`tid` ASC) VISIBLE,

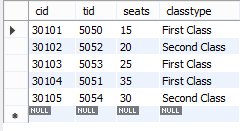
CONSTRAINT `tid`

FOREIGN KEY (`tid`)

REFERENCES `railway`.`train` (`tid`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);



**8. TRAIN**

CREATE TABLE `railway`.`train` (

`tid` INT(5) NOT NULL,

`tname` VARCHAR(10) NOT NULL,

`sid` INT(5) NOT NULL,

PRIMARY KEY (`tid`),

INDEX `sid\_idx` (`sid` ASC) VISIBLE,

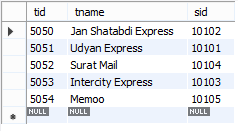
CONSTRAINT `sid`

FOREIGN KEY (`sid`)

REFERENCES `railway`.`station` (`sid`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);



**9. STATION**

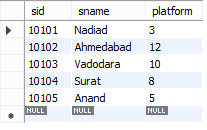
CREATE TABLE `railway`.`station` (

`sid` INT(5) NOT NULL,

`sname` VARCHAR(10) NOT NULL,

`platform` INT NOT NULL,

PRIMARY KEY (`sid`));

****

**10. TRAIN STATUS**

CREATE TABLE `railway`.`train\_status` (

`ref\_no` INT NOT NULL,

`trainid` INT(5) NOT NULL,

`dept\_time` DATETIME NOT NULL,

`arrival\_time` DATETIME NOT NULL,

PRIMARY KEY (`ref\_no`),

INDEX `trainid\_idx` (`trainid` ASC) VISIBLE,

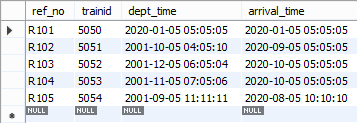
CONSTRAINT `trainid`

FOREIGN KEY (`trainid`)

REFERENCES `railway`.`train` (`tid`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

****

**5.2 INSERT DATA VALUES**

**ADMIN1 - ADMIN2:**

insert into admin1 values("A101","RAJA",1234567890);

insert into admin2 values("A101","password");

insert into admin1 values("A102","Dev",987654321);

insert into admin2 values("A102","123456");

insert into admin1 values("A103","Akash",456789123);

insert into admin2 values("A103","football");

insert into admin1 values("A104","Dirgh",605432198);

insert into admin2 values("A104","qwerty");

insert into admin1 values("A105","Dhruv",908745632);

insert into admin2 values("A105","welcome");

select \* from admin1;

select \* from admin2;

delete from admin1 where aname = "RAJA";

delete from admin1 where aname = "RAJA";

**USER1:**

insert into user1 values("U101","Ebrahim","8888888888","123456789987","A101");

insert into user1 values("U102","Arjun","9537678894","857694523641","A102");

insert into user1 values("U103","Bheem","8140656906","147258369963","A102");

insert into user1 values("U104","Krishna","9054203318","741258963258","A105");

insert into user1 values("U105","Hanuman","6355649653","963258741258","A103");

insert into user1 values("U106","Ram","8522589631","753159842684","A104");

insert into user1 values("U107","Kumbhkaran","7532648922","987456258963","A105");

insert into user1 values("U108","kiran","9636985258","222333456789","A104");

**USER2:**

insert into user2 values("U101","dragon");

insert into user2 values("U102","starwars");

insert into user2 values("U103","solo");

insert into user2 values("U104","monkey");

insert into user2 values("U105","welcome");

select \* from user2;

**STATION:**

insert into station values(10101,"Nadiad",3);

insert into station values(10102,"Ahmedabad",12);

insert into station values(10103,"Vadodara",10);

insert into station values(10104,"Surat",8);

insert into station values(10105,"Anand",5);

select \* from station;

**TRAIN:**

insert into train values(5050,"Jan Shatabdi Express",10102);

insert into train values(5051,"Udyan Express",10101);

insert into train values(5052,"Surat Mail",10104);

insert into train values(5053,"Intercity Express",10103);

insert into train values(5054,"Memoo",10105);

insert into train values(5055,"Chennai Express",10101);

insert into train values(5056,"Madras Express",10102);

select \* from train;

**CLASS:**

insert into class values(30101,5050,15,"First Class");

insert into class values(30102,5052,20,"Second Class");

insert into class values(30103,5053,25,"First Class");

insert into class values(30104,5051,35,"First Class");

insert into class values(30105,5054,30,"Second Class");

**TICKET:**

insert into ticket values(20101,30101,'2001-01-23',"Vadodara","Ahmedabad","U101");

insert into ticket values(20102,30103,'2001-06-15',"Nadiad","Ahmedabad","U102");

insert into ticket values(20103,30103,'2001-06-15',"Nadiad","Ahmedabad","U102");

insert into ticket values(20104,30102,'2001-07-26',"Vadodara","Surat","U103");

insert into ticket values(20105,30101,'2001-11-06',"Anand","Surat","U104");

insert into ticket values(20106,30101,'2001-11-06',"Anand","Surat","U104");

insert into ticket values(20107,30104,'2001-05-07',"Nadiad","Anand","U105");

insert into ticket values(20108,30104,'2001-04-08',"Anand","Nadiad","U106");

insert into ticket values(20109,30103,'2001-03-15',"Anand","Ahmedabad","U107");

insert into ticket values(20110,30102,'2001-02-16',"Surat","Nadiad","U108");

select \* from ticket;

**FARE:**

insert into fare values("F101",20101,50);

insert into fare values("F102",20102,75);

insert into fare values("F103",20103,75);

insert into fare values("F104",20104,60);

insert into fare values("F105",20105,100);

insert into fare values("F106",20106,100);

insert into fare values("F107",20107,80);

insert into fare values("F108",20108,80);

insert into fare values("F109",20109,90);

insert into fare values("F110",20110,40);

select \* from fare;

**TRAIN\_STATUS**:

insert into train\_status values ("R101",5050,'2020-01-05 05-05-05','2020-01-05 05-05-05');

insert into train\_status values ("R102",5051,'2001-10-05 04-05-10','2020-09-05 05-05-05');

insert into train\_status values ("R103",5052,'2001-12-05 06-05-04','2020-10-05 05-05-05');

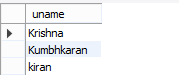
insert into train\_status values ("R104",5053,'2001-11-05 07-05-06','2020-10-05 05-05-05');

insert into train\_status values ("R105",5054,'2001-09-05 11-11-11','2020-08-05 10-10-10');

**5.3 QUERIES**

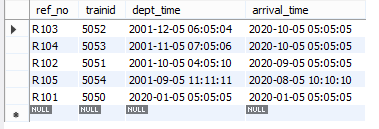
**1. List name of users whose name start with**

select uname from user1 where uname like 'd%';



**2. List name of train status in descending order.**

select \* from train\_status order by arrival\_time desc;



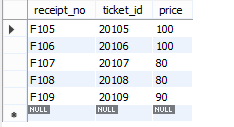
**3. Count total no of seats from a train having first-class as class type.**

select sum(seats) as total seats from class where classtype="first-class";



**4. List all prices ranging from**

select \* from fare where price between 80 and 100;



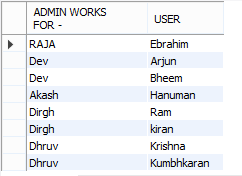
**5. Find the minimum and maximum cost of a ticket.**

select min(price) as "Min\_price", max(price) as "Max\_price" from fare;



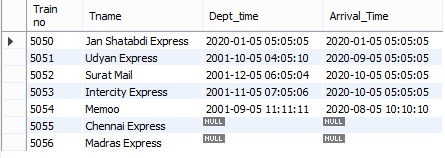
**6. Display all admin which works for appropriate users**.

SELECT B.ANAME AS "ADMIN WORKS FOR -" ,A.UNAME AS "USER " FROM USERS A INNER JOIN ADMIN B ON A.UID=B.AID;



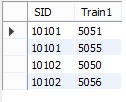
**7. List all Train name and its arrival and departure time by using train & status table.**

SELECT T.tid as "Train no" , T.Tname, TS.Dept\_time, TS.Arrival\_Time FROM Train T left JOIN TRAIN\_STATUS TS ON T.TID = TS.trainid ORDER BY T.Tid;



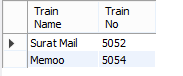
**8. Details of Trains Which can come at the same station**.

SELECT A.SID,A.TID AS Train1 FROM Train A, Train B WHERE A.TID != B.TID AND A.SID = B.SID ORDER BY A.SID;



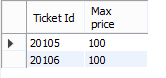
**9. Trains having second class compartment.**

select train.tname as "Train Name",train.tid as "Train No" from train,class where class.classtype = "Second Class" and train.tid = class.tid



**10. Display ticket id whose price is the maximum among all.**

select ticket\_id as "Ticket Id",price as "Max price" from fare f1 where price = (select max(price) from fare f2);



**5.4 PL/SQL**

**THIS PL/SQL GIVE USER ID, ADMIN ID,ADHAR NO AND PHONE NUMBER OF KRISHNA.**

declare

v\_user user1 % rowtype;

begin

select username,userid,uphoneno,adharno,aid into v\_user.username,v\_user.userid,v\_user.uphoneno,v\_user.adharno,v\_user.aid from user1 where username='Krishna';

dbms\_output.put\_line(v\_user.username||' '||v\_user.userid||' '||' '||v\_user.uphoneno||' '||v\_user.adharno||' '||v\_user.aid);

End;



**5.5 FUNCTION**

**A function which counts the total number of seats.**

CREATE OR REPLACE FUNCTION totalSeats RETURN

number IS

total number(3) := 0;

BEGIN

SELECT sum(seats) into total FROM class;

RETURN total;

END;

BEGIN

dbms\_output.put\_line('Total seats =' || totalSeats);

END;

/



**5.6 CURSORS**

**This Cursor Gives The Information About People AdharCard number And Their PHONE NUMBER whose NAME starts with either ‘A’ or ‘K ’ .**

declare

d1 user1.adharno % type;

d2 user1.uphoneno % type;

cursor c1 is select adharno,uphoneno from user1 where

username like 'A%'

or username like 'K%';

begin

dbms\_output.put\_line('Adhar no'||' '||'Phone Number');

open c1;

loop

fetch c1 into d1,d2;

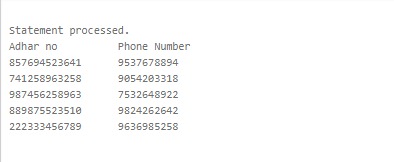
exit when c1 % notfound;

dbms\_output.put\_line(d1||' '||d2);

end loop;

close c1;

End;



**5.7 TRIGGERS**

**1.This trigger updates price value by 200 whenever new values is Inserted into fare.**

create trigger farex

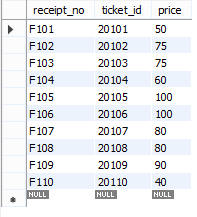
before insert

on fare

for each row

set new.price = new.price + 200;

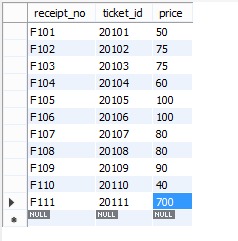
**BEFORE:**



insert into ticket values(20111,30103,'2001-02-16',"Surat","Nadiad","U108");

insert into fare values('F111',20111,500);

**AFTER:**



**2. This updates the arrival time of train and creates a new table which consists of train’s previous and current arrival time.**

CREATE TABLE Timechanges (

id INT AUTO\_INCREMENT PRIMARY KEY,

trainid INT,

previoustime datetime,

currenttime datetime,

changedAt TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP

);

DROP TRIGGER IF EXISTS `railway`.`train\_status\_AFTER\_UPDATE`;

DELIMITER $$

USE `railway`$$

CREATE DEFINER=`root`@`localhost` TRIGGER `train\_status\_AFTER\_UPDATE` AFTER UPDATE ON `train\_status` FOR EACH ROW BEGIN

IF OLD.arrival\_time <> new.arrival\_time THEN

INSERT INTO Timechanges(trainid, previoustime, currenttime)

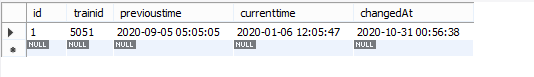
VALUES(old.trainid, old.arrival\_time, new.arrival\_time);

END IF;

END$$

DELIMITER ;

update train\_status SET arrival\_time='2020-01-06 12-05-47' where trainid=5051;



**6. BIBLIOGRAPHY**

* For the implementation of this Railway Reservation System project we referred to many websites and books.
* We created the ER Diagram and Schema Diagram on “<https://creately.com/>”.

**Reference Website:**

1. [**https://www.w3schools.com**](https://www.w3schools.com)
2. [**https://www.mysqltutorial.org**](https://www.mysqltutorial.org)
3. [**https://stackoverflow.com/**](https://stackoverflow.com/)
4. **https://www.w3resource.com**

**Submitted By,**

IT085- Dhruv Patel

IT086- Dirgh Patel