

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE**

**INTRODUCTION TO DATABASE**

**SPRING 2022-23** , **Section: I**

**PROJECT ON**

**RAILWAY MANAGEMENT SYSTEM**

**Supervised By**

**NUSRAT JAHAN ANANNYA**

**Submitted By**

|  |  |  |  |
| --- | --- | --- | --- |
| **SERIAL**  **NO** | **Name** | **ID** | **Contribution** |
| **01** | **MD SHAFIN** | **22-48257-2** | **25%** |
| **02** | **ASHFI MUHAMMED Raiyan** | **22-46894-1** | **25%** |
| **03** | **AYASA AKTER MIM** | **22-47024-1** | **25%** |
| **04** | **AHAMMED ULLAH** | **22-46930-1** | **25%** |

**SUBMISSION. 14th May ,2023**

**TABLE OF CONTENTS:**

|  |  |
| --- | --- |
| TOPICS | *Page no*. |
| 1. Title Page | **01** |
| 1. Table of Content | **02** |
| 1. INTRODUCTION | **03** |
| 1. LIST OF TABLE | **03** |
| 1. CASE STUDY | **03** |
| 1. E R DIAGRAM | **04** |
| 1. NORMALIZATION | **05-06** |
| 1. TEMPORARY AND FINAL TABLE | **06** |
| 1. TABLE CREATION | **06-12** |
| 1. TABLE DECORATION | **13-15** |
| 1. QUERY WRITING | **15-19** |
| 1. JOINNING | **20-22** |
| 1. VIEW AND SEQUENCE | **22-24** |
| 1. CONCLUSION | **25** |

**INTRODUCTION**  :

The advanced and essential railway management system makes sure that the railway sector runs smoothly.It involves managing a huge network of rail tracks, automobiles, stations, and passengers.A secure database system is required to carry out this procedure in an effective and efficient manner.One such database management system that offers cutting-edge tools and functionality to successfully manage railroad operations is Oracle 10g.Railway administration can manage vital information like train timetables, passenger statistics, ticket purchases, and other crucial elements with Oracle 10g. The entire effectiveness of railroad operations is increased, and the passenger experience is improved, thanks to this database management system. In this post, we'll examine the many aspects of Oracle 10g's railway management system as well as its advantages for managing railways.

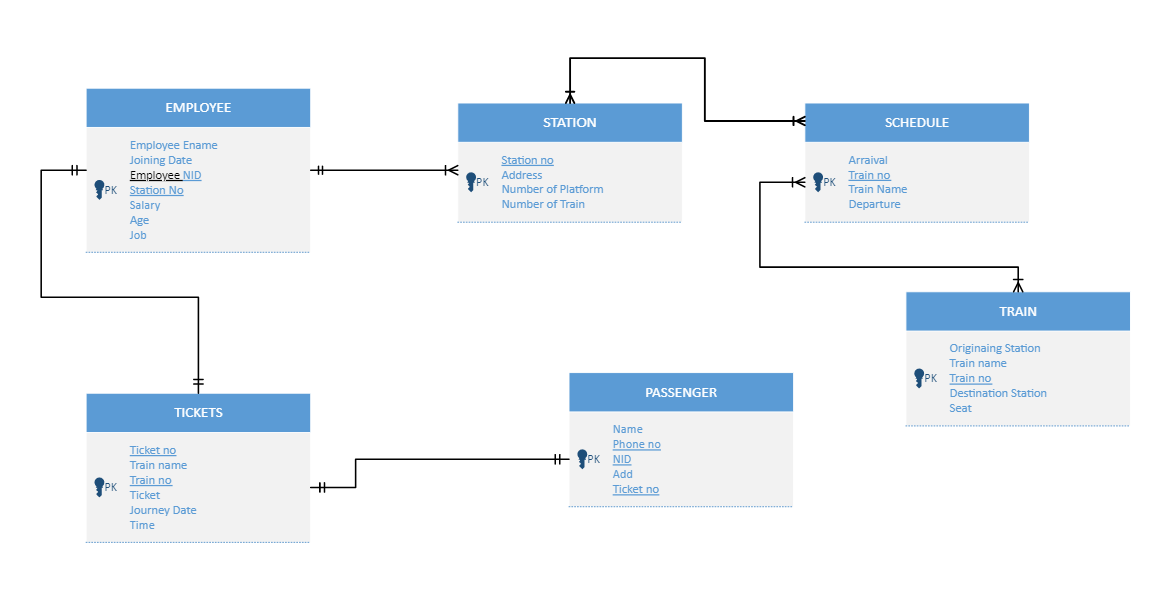
**LIST of TABLE :**

* Station
* Employee
* Tickets
* Passenger
* Trains
* Schedule

**CASE STUDY :**

The railway department of the Bangladeshi government demands that designers create a data management system that will contain all of the stations and the data that comes with it. The database will include the station names, addresses, train and platform numbers, and train and platform counts. This system will record certain important employee data, like NID, Age, Name, Home Address, Salary, Joining Date, and Job.And each of these all-booked tickets will be assigned a unique tickets number together with the following information: issuance date, train name, tickets class type, train number, travel time, journey date, and fee. They will be a valid passenger for train travel after purchasing a ticket. The unique key passenger mobile phone number will be recorded together with the additional information (destination, departure, gender, age, passenger name, NID, tickets no. ), which is required for each passenger in order to reserve.. If there is a delay, the trains will arrive at the scheduled time. Any scheduling system can be managed by employees, who can also keep track of train and schedule information. The name of the train, its unique number, its destination, its start station, and its total number of seats are all important data elements. This system's schedule displays the train number, name, arrival time, departure time, and off day.



**Schema Diagram: **

**Normalization :**

**RESERVATION:**  
**UNF** **:** (Name , Phone -no , N-id , Add , Ticket-no , Train-name , Train-no , E-MAIL, Time , Journey date , Class )  
  
**1st NF :** ( Ticket-no , Train-no , Phone-no, N-id , E-MAIL , Train-name ,Class , Journey date , Time , Add , Name )

**2nd NF :**  (, Ticket\_no(FK) Nid , E-MAIL, Phone-no ,Add ,Name )

( Ticket-no(FK), Train-no (FK), , Train-name Time , Journey date , Class )

**3rd NF** : same as second NF**.**

**MANIPULATE** :   
**UNF :** ( Ticket-no , Train-name , Train-no , Time , Journey date , Class , Home address , Age , Job , E-name , N-id , Joining date , Station-no, Salary )

**1st NF :** ( Ticket-no , , Train-no, , Nid , Station-no , Time , Journey date , Class , Age , Job , E-name , Joining date , Salary , Home address, Train-name )

**2NF :** ( Ticket-no , , Train-no(FK) , Train-name Journey date , time , Class)

( Nid , Station-no (FK), Age , Job , E-name , Salary , Home-Address Joining date)

**3rd NF** : same as second NF**.**

**HAS :**

**UNF :** ( Station-no , Address , Number of platform ,Number of train , Train no , Train-name , Departure , Arraival )

**1st NF :** ( Train-no , Station-no , Address ,Train-name, Number of platform , Number of train , Arraival ,Departure )

**2nd NF :** ( Station-no , Address , Number of platform , Number of train )

( Train-no(FK) , Train-name , Arraival ,Departure )

**3rd NF** : same as second NF**.**

**MAINTAIN** :

**UNF :** ( Train-no ,Train-name, Arraival ,Departure ,Originaing-station , Seat , Destination-station , Train-no ,Train-name )

**1st NF :** ( Train-no, Train-no(TRAIN TABLE) ,Train-name , Train-name , Seat, Arraival,,Departure , orginaing-station , Destination-station )

**2nd NF :** ( Train-no , Train-name , Arraival , Departure )

( Train-no(FK) ,Train-name ,Originaing-station,seat,Destination-station )

**3rd NF** : same as second NF**.**

**WORK ON :**

**UNF :** ( Salary , Station-no , Nid , Joining-date ,E-name ,job ,Age , Home-Address , Station-no , Address , number of platform ,Number of train )

**1st NF :** ( Station-no , Nid , Station-no ,Home-adress Salary , Joining-date ,E-name, Job , Age , Address , Number of platform , Number of Train )

**2nd NF :** ( Station-no (FK), Nid , Home-address , Age , Job ,E-name , Salary, joining date )  
 ( Station-no(FK) , Address ,Number of platform , number of Train )

**3rd NF** : same as second NF**.**

**TEMPORARY TABLE :**

1. Phone-no, Nid , Name
2. Ticket-no , Train-name , Train-no , Time , Journey date , Class
3. Ticket-no , Train-name, Train-no , Journey date , Time , Class
4. Home-Address , Nid , Station-no , Age , Job , E-name , Salary , Joining date
5. Station-no , Address , Number of platform ,Number of Train
6. Train-no , Train-name , Arraival ,Departure
7. Train-no , Train-name , Arraival,Departure
8. Seat , Train-no , Train-name , Orginaing station ,Destination-station
9. Station- no , Home-Address ,Age ,Job ,E-name,Salary , Joining date
10. Station-no , Address ,Number of Platform , Number of Train .

**FINAL TABLE :**

1. Phone-no , Nid , Name
2. Ticket-no , Train-name ,Train no ,Time ,Journey date , Class
3. Home-Address , Nid ,Station-no ,Age , Job , E-name , Salary , Joining date
4. Station no , Address ,Number of platform ,Number of Train
5. Train-no , Train-name ,Arraival ,Departure
6. Seat , Train-no , Train-name , Orginasing Station ,Destination station

**TABLE CREATION**

1. **TRAIN TABLE**

CREATE TABLE TRAINS (

Train\_No NUMBER(4) CONSTRAINT PK\_TRAINS PRIMARY KEY,

Train\_Name VARCHAR2(20),

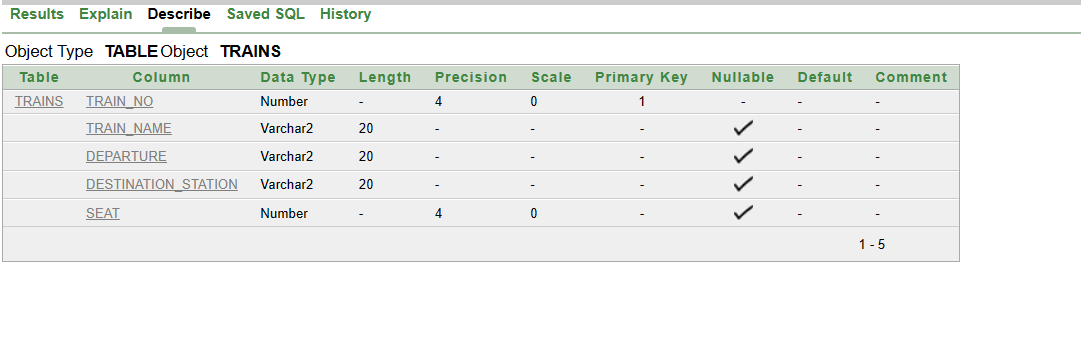
Departure VARCHAR2(20),

DESTINATION\_STATION VARCHAR2(20),

Seat NUMBER(4)

);

DESCRIBE TRAINS;



INSERT INTO TRAINS VALUES

(**1,** 'MOHANAGAR GODHOLI', 'CUMILLA','DHAKA',2000);

INSERT INTO TRAINS VALUES

(**2,** 'UPAKUL EXPRESS', 'DHAKA','NOAKHALI',800);

INSERT INTO TRAINS VALUES

(**3**, 'BARISHAL EXPRESS', 'BARSHAL','DHAKA',2000);

INSERT INTO TRAINS VALUES

(**4,** 'PROVATI', 'BHOLA','BARISHAL',300);

INSERT INTO TRAINS VALUES

(**5,** 'MEGHNA EXPRESS', 'DHAKA','CHATTOGRAM',1350);

SELECT \* FROM TRAINS;

Graphical user interface, table

Description automatically generated

**2. STATION TABLE**

CREATE TABLE STATION

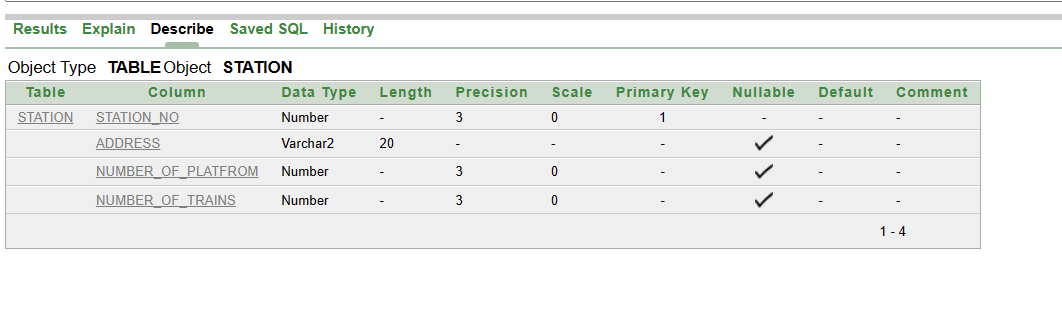
(STATION\_NO NUMBER(3) CONSTRAINT PK\_STATION PRIMARY KEY,

ADDRESS VARCHAR2(20),

NUMBER\_OF\_PLATFROM NUMBER(3),

NUMBER\_OF\_TRAINS NUMBER(3));

DESCRIBE STATION;



INSERT INTO STATION VALUES

(104,'CUMILLA',3,2);

INSERT INTO STATION VALUES

(105,'DHAKA',2,2);

INSERT INTO STATION VALUES

(106,'BARISHAL',6,5);

INSERT INTO STATION VALUES

(107,'BHOLA',NULL,NULL);

INSERT INTO STATION VALUES

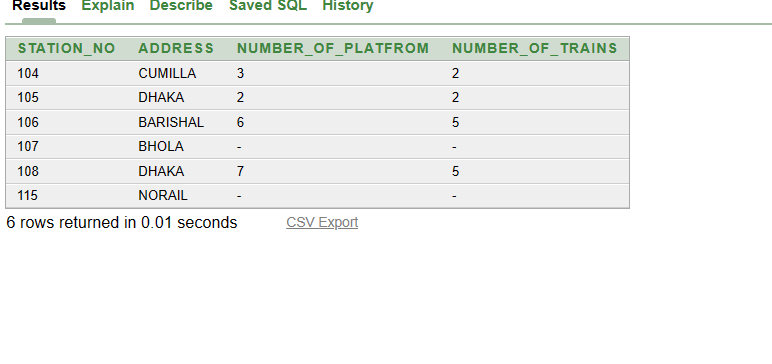
(108,'DHAKA',7,5);

INSERT INTO STATION VALUES

(115,'NORAIL',NULL,NULL);

SELECT \*

FROM STATION;



**3. TICKET TABLES**

CREATE TABLE TICKETS

(TICKET\_NO NUMBER(5) CONSTRAINT PK\_TICKETS PRIMARY KEY,

TRAIN\_NAME VARCHAR2(20),

JOURNEY\_DATE DATE,

TIME VARCHAR2(7),

ISSUE\_DATE DATE,

TRAIN\_NO NUMBER(4) CONSTRAINT FK\_TRAINS REFERENCES TRAINS,

TICKET\_PRICE NUMBER(3));

DESCRIBE TICKETS;

Table

Description automatically generated

INSERT INTO TICKETS VALUES

(TICKETS\_SEQ.NEXTVAL, 'MOHANOGOR GODHULI','28-APR-2023', '06:30','20-APR-2023', 0002, 250);

INSERT INTO TICKETS VALUES

(TICKETS\_SEQ.NEXTVAL, 'UPAKUL EXPRESS','27-APR-2023', '22:00','23-APR-2023', 0001, 300);

INSERT INTO TICKETS VALUES

(TICKETS\_SEQ.NEXTVAL, 'BARISHAL EXPRESS','28-APR-2023', '17.00','21-APR-2023', 0001, 950);

INSERT INTO TICKETS VALUES

(TICKETS\_SEQ.NEXTVAL, 'PROVATI','26-APR-2023', '8.00','18-APR-2023', 0002, 750);

INSERT INTO TICKETS VALUES

(TICKETS\_SEQ.NEXTVAL, 'PAHARIKA','28-APR-2023', '12.00','19-APR-2023', 0001, 200);

INSERT INTO TICKETS VALUES

(TICKETS\_SEQ.NEXTVAL, 'MEGHNA EXPRESS','24-APR-2023', '1.00','16-APR-2023', 0003, 900);

INSERT INTO TICKETS VALUES

(TICKETS\_SEQ.NEXTVAL, 'SIZUKA EXPRESS','26-APR-2023', '8.00','18-APR-2023', 0002, 650);

SELECT \*

FROM TICKETS;

## Graphical user interface, table Description automatically generated

**4. PASSENGER TABLE**

CREATE TABLE PASSENGR (

PHONE\_NO NUMBER(11) CONSTRAINT PK\_PASSENGR PRIMARY KEY,

NAME VARCHAR2(20),

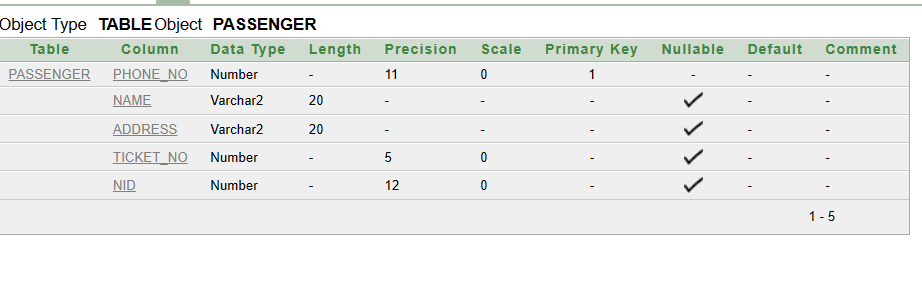
ADDRESS VARCHAR2(20),

TICKET\_NO NUMBER(5) CONSTRAINT FK\_TICKET\_NO REFERENCES TICKETS,

NID NUMBER(12)

);

DESCRIBE PASSENGR;



INSERT INTO PASSENGR VALUES (01767333701,'RAIYAN', 'CUMILLA',10001,356455261235);

INSERT INTO PASSENGR VALUES (01767332541,'AMEDULLA', 'BHOLA',10002,526123535645);

INSERT INTO PASSENGR VALUES (01824463412,'SAFIN','BARISAL',10003,789456123789);

INSERT INTO PASSENGR VALUES (01767313254,'AYASHA', 'NOAKHALI',10004,235613525645);

INSERT INTO PASSENGR VALUES (01940256605,'BUSHRA', 'CHATTOGRAM',10005,554017074255);

INSERT INTO PASSENGR VALUES (01940107080,'NOBEL', 'SYLHET',10006,256452356135);

INSERT INTO PASSENGR VALUES (01901070804,'RUBEL', 'RAJSHAHI',10007,707425555401);

INSERT INTO PASSENGR VALUES (01708049010,'AYMAN', 'NAOGA',10008,401707425555);

INSERT INTO PASSENGR VALUES (01824463482,'SHAFIN','BARISAL',10009,123456789123);

SELECT \*  
FROM PASSENGR;

Table

Description automatically generated

**05. SEHEDULE TABLE:**

CREATE TABLE SCHEDULE

(TRAIN\_NO NUMBER(4) CONSTRAINT FK\_TRAINS2 REFERENCES TRAINS,

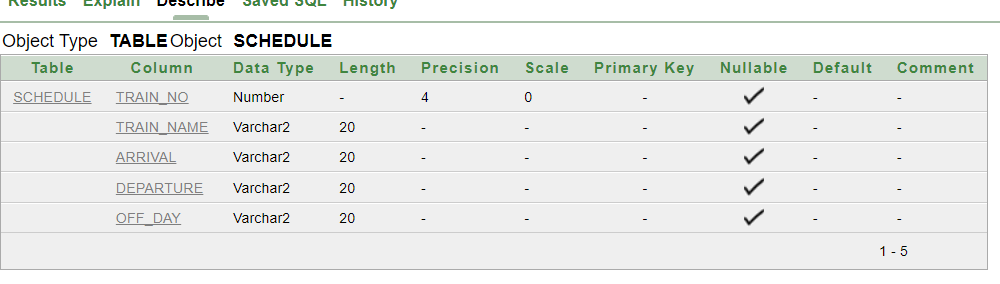
TRAIN\_NAME VARCHAR2(20),

ARRIVAL VARCHAR2(20),

DEPARTURE VARCHAR2(20),

OFF\_DAY VARCHAR(20));

DESC SCHEDULE;



INSERT INTO SCHEDULE VALUES

(001,'EKOTA\_EXPRESS','08:30','09:00','TUESDAY');

INSERT INTO SCHEDULE VALUES

(002,'TISTA\_EXPRESS','14:10','14:40','SUNDAY');

INSERT INTO SCHEDULE VALUES

(003,'PARABAT\_EXPRESS','18:10','18:40','TUESDAY');

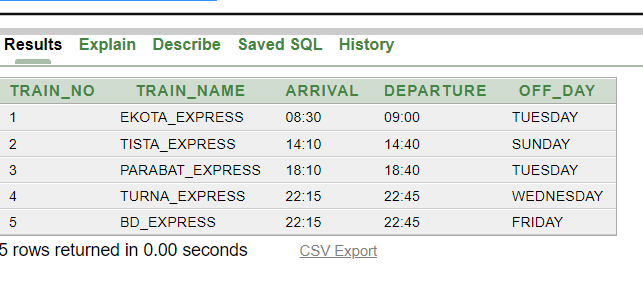
INSERT INTO SCHEDULE VALUES

(004,'TURNA\_EXPRESS','22:15','22:45','WEDNESDAY');

INSERT INTO SCHEDULE VALUES

(005,'BD\_EXPRESS','22:15','22:45','FRIDAY');

SELECT \*  
FROM SCHEDULE;



**06.EMPLOYEE TABLE:**

CREATE TABLE EMPLOYEE

(NID NUMBER(12) CONSTRAINT PK\_EMPLOYEE PRIMARY KEY,

ENAME VARCHAR2(20),

JOB VARCHAR2(20),

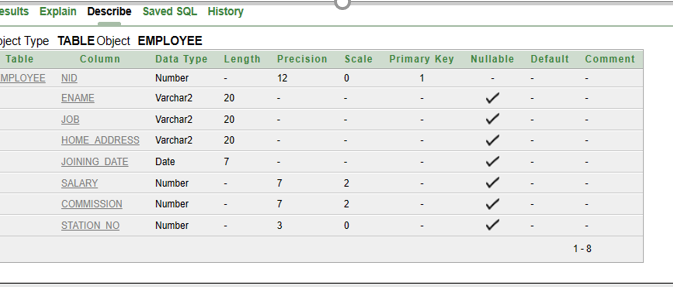
HOME\_ADDRESS VARCHAR2(20),

JOINING\_DATE DATE, SALARY NUMBER(7,2),

COMMISSION NUMBER(7,2),

STATION\_NO NUMBER(3) CONSTRAINT FK\_STATION REFERENCES STATION);

DESC EMPLOYEE;



INSERT INTO EMPLOYEE VALUES ('0101010101','RAIYAN','STATION MASTER','BANANI,DHAKA','22-JAN-1988',20500,NULL,104);

INSERT INTO EMPLOYEE VALUES ('909090909','SAFIN','GANDY DANCER','BARISHAL','19-FEB-1999',11500,NULL,105);

INSERT INTO EMPLOYEE VALUES ('0202020202','AHMEDULLAH','T.T.','NIMTOLA, JAMALPUR','03-SEP-1994',10500,1050.5,106);

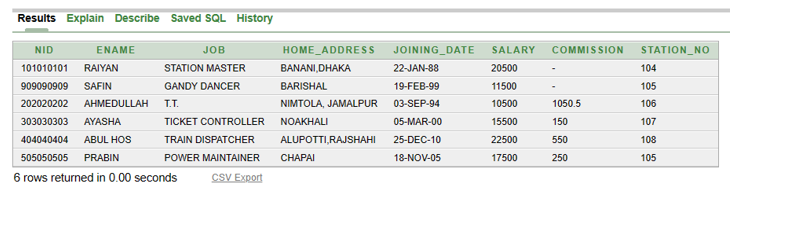
INSERT INTO EMPLOYEE VALUES ('0303030303','AYASHA','TICKET CONTROLLER','NOAKHALI','05-MAR-2000',15500,150,107);

INSERT INTO EMPLOYEE VALUES (0404040404,'ABUL HOS','TRAIN DISPATCHER','ALUPOTTI,RAJSHAHI','25-DEC-2010',22500,550,108);

INSERT INTO EMPLOYEE VALUES (0505050505,'PRABIN','POWER MAINTAINER','CHAPAI','18-NOV-2005',17500,250,105);

SELECT \*

FROM EMPLOYEE;



**UPDATE SCHEDULE TABLE:**

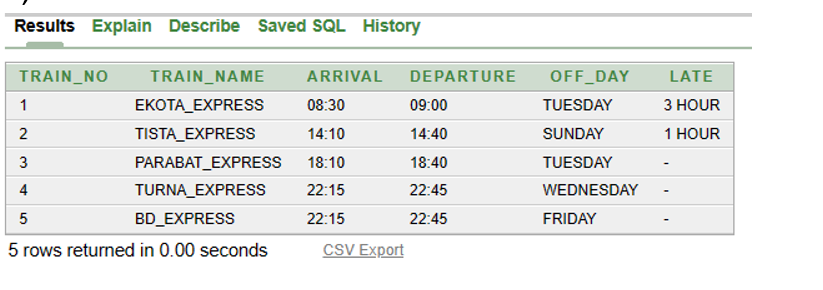
ALTER TABLE SCHEDULE ADD(LATE VARCHAR2(10));

UPDATE SCHEDULE SET LATE='3 HOUR' WHERE TRAIN\_NAME='EKOTA\_EXPRESS'

UPDATE SCHEDULE SET LATE='1 HOUR' WHERE TRAIN\_NAME='TISTA\_EXPRESS'

SELECT \*

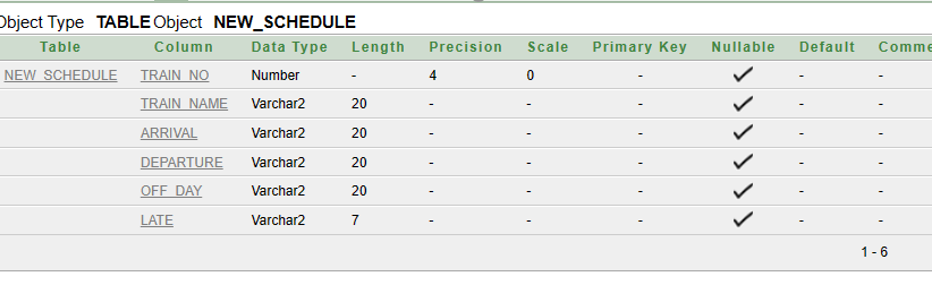
FROM SCHEDULE;



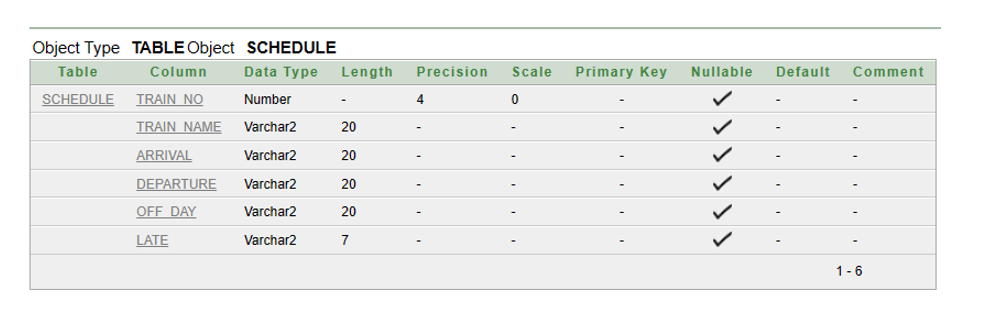
**TABLE NAME CHANGE:**

RENAME SCHEDULE TO NEW\_SCHEDULE

DESCRIBE NEW\_SCHEDULE;

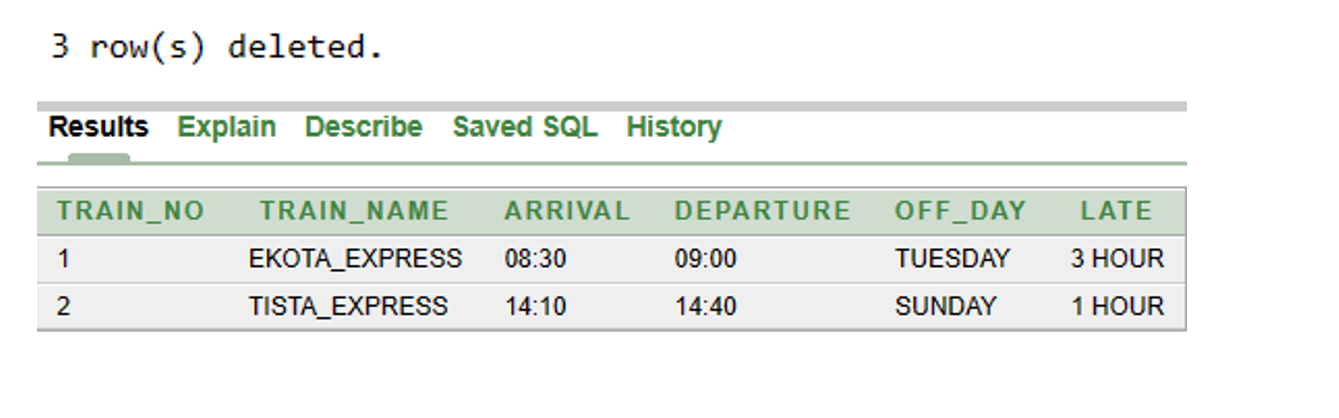


|  |
| --- |
| RENAME NEW\_SCHEDULE TO SCHEDULE  DESCRIBE SCHEDULE; |



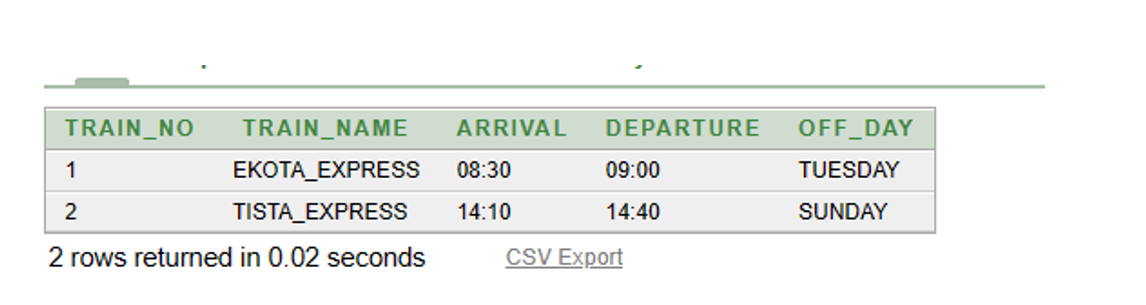
|  |
| --- |
| DELETE FROM SCHEDULE WHERE LATE IS NULL; |

**DELETE DATA:**

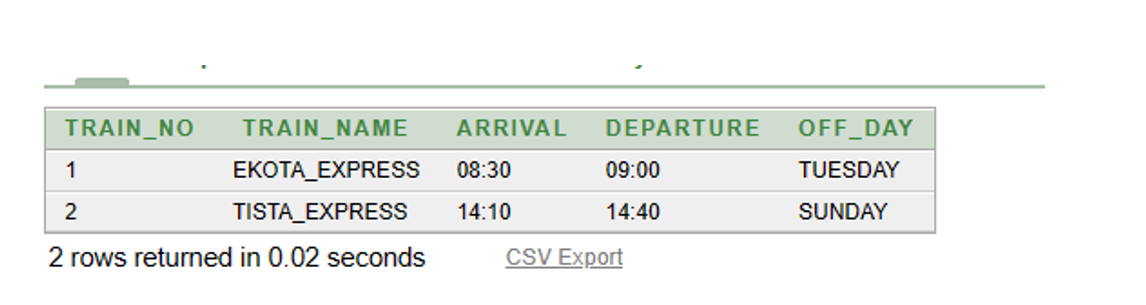


|  |
| --- |
| ALTER TABLE SCHEDULE DROP COLUMN LATE |

**TABLE DROP:**



|  |
| --- |
| ALTER TABLE SCHEDULE RENAME COLUMN ARRIVAL TO DESTINATION\_TIME |

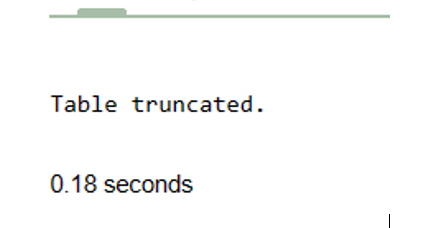
**RENAME TABLE COLUMN HEADING**

|  |
| --- |
| ALTER TABLE SCHEDULE RENAME COLUMN DESTINATION\_TIME TO ARRIVAL |

****

**TRUNCATE:**

|  |
| --- |
| TRUNCATE TABLE SCHEDULE |



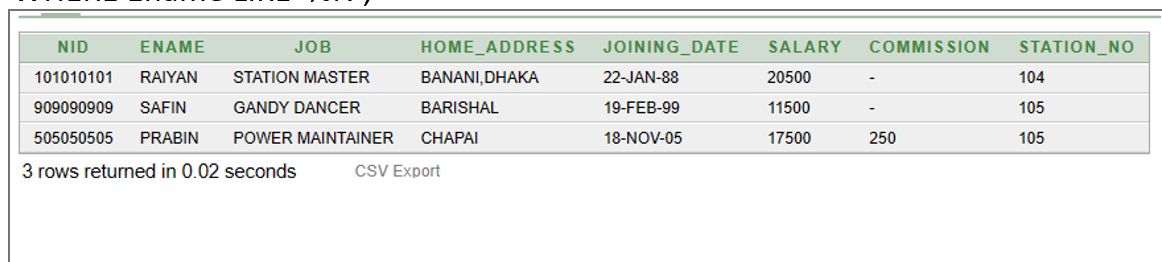
**QUERY WRITING:**

**01.**

**Employees whose Ename ends with ‘N’?**

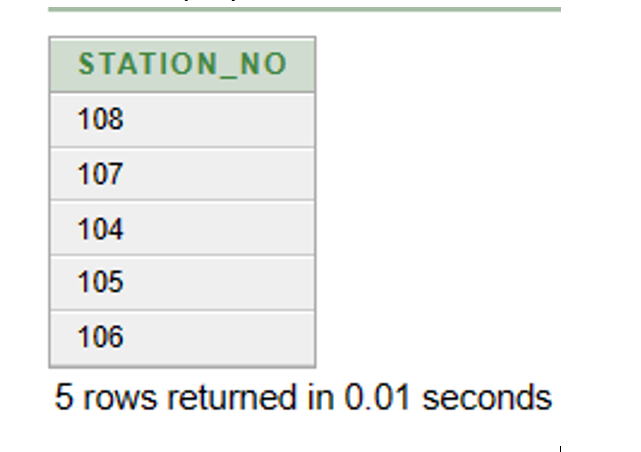
SELECT \* FROM Employee

WHERE Ename LIKE '%N';



**02. Unique values of STATION NUMBER ?**

|  |
| --- |
| SELECT DISTINCT STATION\_NO  FROM STATION; |



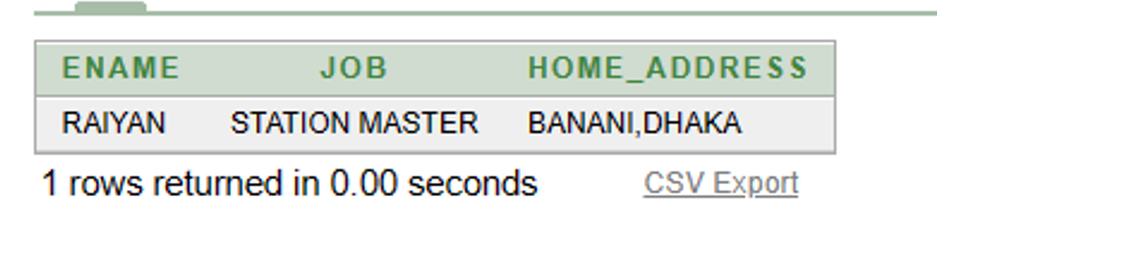
**Single row function query:**

**1. Show employee name, job, home address where employee name is 'RAIYAN’.**

SELECT ENAME, JOB, HOME\_ADDRESS

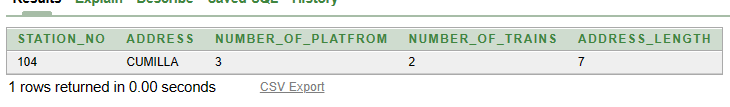
FROM EMPLOYEE

WHERE ENAME='RAIYAN';



**02.Write a single-row function query for the STATION table that retrieves all columns of a specific row and calculates the length of the ADDRESS column using the LENGTH function.**

|  |
| --- |
| SELECT  STATION\_NO,  ADDRESS,  NUMBER\_OF\_PLATFROM,  NUMBER\_OF\_TRAINS,  LENGTH(ADDRESS) AS ADDRESS\_LENGTH  FROM  STATION  WHERE  STATION\_NO = 104; |



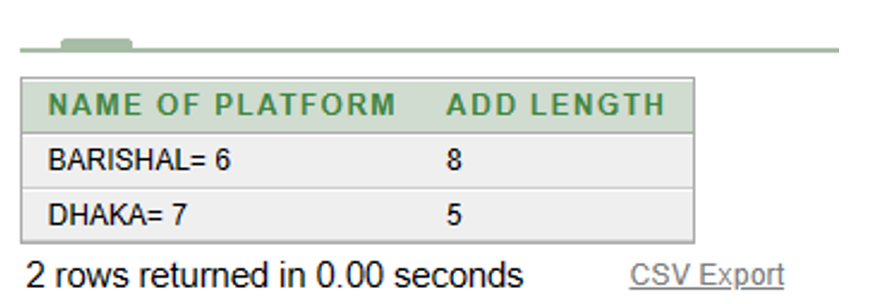
**Multiple row function query :**

**Show name of station and number of platform as "NAME OF PLATFORM" and show the length of address where the number of trains is greater than 4**

SELECT ADDRESS ||'= '|| NUMBER\_OF\_PLATFROM AS "NAME OF PLATFORM", LENGTH(ADDRESS) AS "ADD LENGTH"

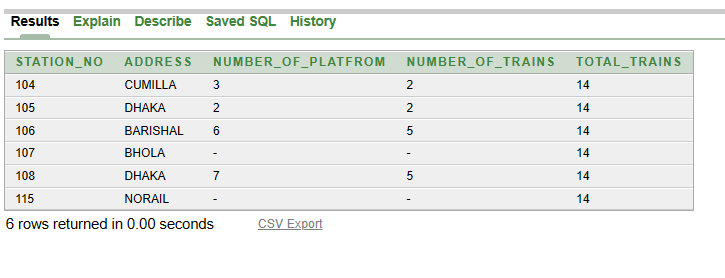
FROM STATION

WHERE NUMBER\_OF\_TRAINS>4;

****

**Uses a subquery to calculate the total number of trains across all stations and renames the result to TOTAL\_TRAINS.**

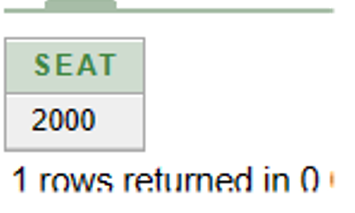
|  |
| --- |
| SELECT STATION\_NO, ADDRESS, NUMBER\_OF\_PLATFROM, NUMBER\_OF\_TRAINS,  (SELECT SUM(NUMBER\_OF\_TRAINS)  FROM STATION) AS TOTAL\_TRAINS  FROM STATION; |



|  |
| --- |
| SELECT SEAT  FROM TRAINS  WHERE TRAIN\_NO=  (SELECT TRAIN\_NO FROM TICKETS WHERE TICKET\_NO=10002); |

**Single row sub queries** :

**Show the number of seats on the train of ticket number 10002?**



**Write a subquery to find the maximum number of seats for all trains?**

|  |
| --- |
| SELECT Train\_No,Train\_Name, Departure,DESTINATION\_STATION,Seat,  (SELECT MAX(Seat) FROM TRAINS) AS MAX\_SEATS  FROM TRAINS  WHERE Train\_No = 0001; |

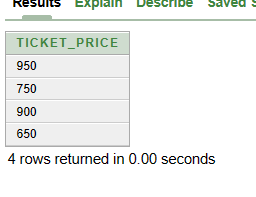


**Multiple row sub query:**

SELECT TICKET\_PRICE

FROM TICKETS

WHERE TICKET\_PRICE > (SELECT AVG(TICKET\_PRICE) FROM TICKETS);



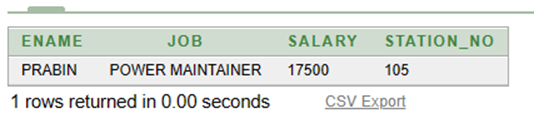
**02.Write a SQL query to retrieve the name, job, salary, and station number of all employees whose salary is greater than the average salary of employees working in the same station.?**

**SELECT ENAME, JOB, SALARY, STATION\_NO**

**FROM EMPLOYEE e**

**WHERE SALARY > (SELECT AVG(SALARY) FROM**

**EMPLOYEE WHERE STATION\_NO = e.STATION\_NO);**

****

**JOINING QUERIES**

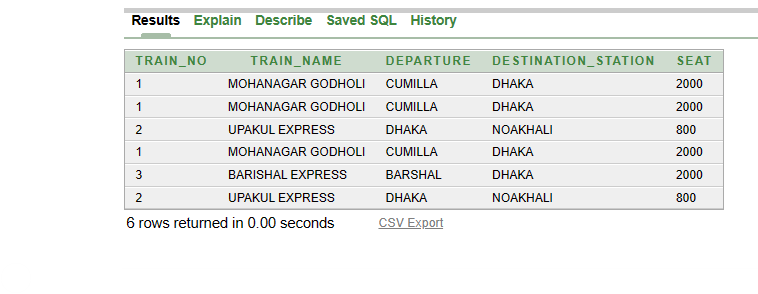
1. **EQUIJOIN:**

SELECT T.Train\_No, T.Train\_Name, T.Departure, T.DESTINATION\_STATION, T.Seat

FROM TICKETS TK

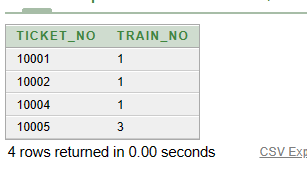
JOIN TRAINS T

ON TK.Train\_No = T.Train\_No;



|  |
| --- |
| **Select ticket no. of trains has seats number between 1800 to 2000?**  SELECT TK.TICKET\_NO, TR.TRAIN\_NO FROM TICKETS TK, TRAINS TR WHERE  TR.SEAT BETWEEN 1800 AND 2000 AND TR.TRAIN\_NO=TK.TRAIN\_NO; |

**02. Non-equijoining:**

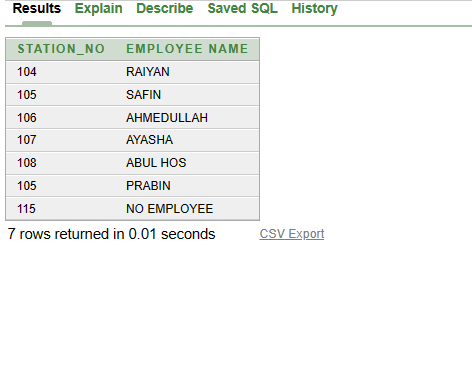


**03.Left outer join:**

**Show all stations working employee name if there is no employee then show "NO EMPLOYEE".?**

SELECT S.STATION\_NO,NVL(E.ENAME,'NO EMPLOYEE')AS "EMPLOYEE NAME"

FROM STATION S LEFT OUTER JOIN EMPLOYEE E ON (E.STATION\_NO=S.STATION\_NO);

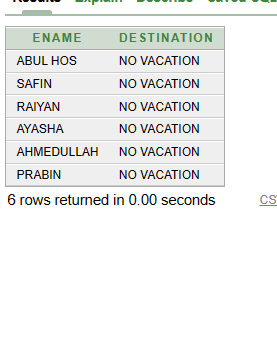


**Right outer join:**

**SHOW THE EMPLOYEE NAME AND DESTINATION WHO ARE TRAVELING DHAKA IF SOME DID NOT TRAVEL THEN MENTION THEM " NO VACATION"?**

SELECT E.ENAME,NVL(P.ADDRESS,'NO VACATION')AS "DESTINATION" FROM PASSENGR P RIGHT

OUTER JOIN EMPLOYEE E ON (E.NID=P.NID);



**Full outer join**

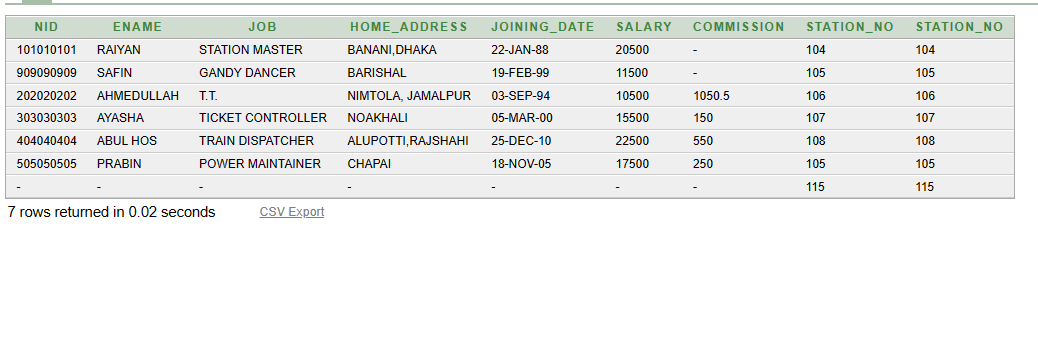
**Question Display all the stations and if there is any station without employees show NO EMPLOYEE?**

SELECT E.NID, E.ENAME, E.JOB, E.HOME\_ADDRESS, E.JOINING\_DATE, E.SALARY, E.COMMISSION, S.STATION\_NO, S.STATION\_NO

FROM EMPLOYEE E

FULL OUTER JOIN STATION S

ON E.STATION\_NO = S.STATION\_NO;

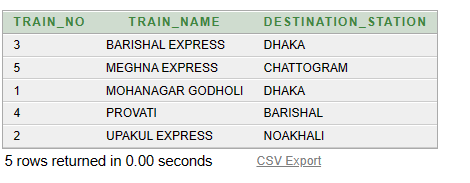


**SELF JOIN:**

**Display trains that has the same destination station??**

SELECT A.TRAIN\_NO, A.TRAIN\_NAME, A.DESTINATION\_STATION FROM TRAINS A, TRAINS B WHERE (A.TRAIN\_NAME=B.TRAIN\_NAME) AND

(A.DESTINATION\_STATION=B.DESTINATION\_STATION) ORDER BY A.TRAIN\_NAME;

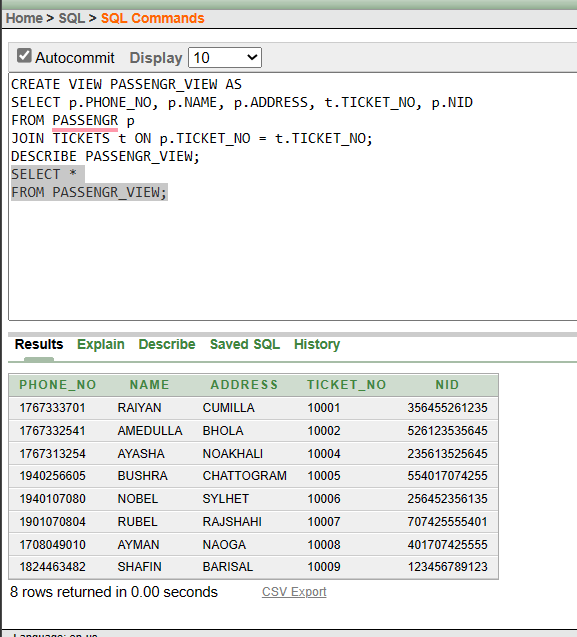


**Creating View:**

In SQL, a view is a virtual table based on the result-set of an SQL statement.A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.Add SQL statements and functions to a view and present the data as if the data were coming from one single table.

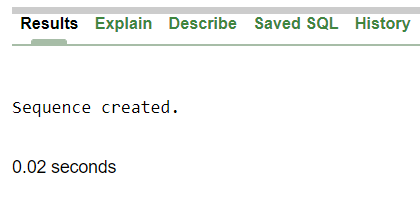
A view is created with the CREATE VIEW statement.

**CREATE VIEW PASSENGR\_VIEW AS  
SELECT p.PHONE\_NO, p.NAME, p.ADDRESS, t.TICKET\_NO, p.NID  
FROM PASSENGR p  
JOIN TICKETS t ON p.TICKET\_NO = t.TICKET\_NO;  
DESCRIBE PASSENGR\_VIEW;  
SELECT \*  
FROM PASSENGR\_VIEW;**



**Create Sequence:**

|  |
| --- |
| CREATE SEQUENCE TICKETS\_SEQ  MINVALUE 10001  MAXVALUE 99999  START WITH 10001  NOCACHE  NOCYCLE  INCREMENT BY 1;  DESCRIBE TICKETS\_SEQ; |



**Relational Algebra:**

01.Find the names of the employees who earn more than 20500

Ans: π e\_name(σe\_salary > 20500 (employee))

02.Find the name of the passenger where ticket number is 1001

Ans: π pass\_name(σtic\_nu = ‘1001’ (passenger))

**CONCLUSION :**

In conclusion, databases are vital to efficient railway management because they make it possible to handle huge amounts of data like schedules, passenger and freight records, maintenance schedules, and employee information. Modern railway management systems make use of this data to improve operations, safety, and customer satisfaction. With the development of technology, railway management has advanced, enabling the collection and analysis of enormous volumes of data. Railway management may streamline operations and take knowledgeable decisions by utilizing databases. So, in order to enhance operations and maintain a competitive edge, railroad firms must give top priority to trustworthy and advanced database systems.

THE END-