Mini Project Report

on

TRAVEL COMPANY MANAGEMANT SYSTEM

Report Submitted to

Jawaharlal Nehru Technological University Anantapur, Ananthapuramu

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

S ABHISHEK	**21121A3541**
SM AFREEN	**21121A3542**
SM YASWANTH	**21121A3543**
S BINDU SREE	**21121A3544**
T KUSHAL KUMAR	**21121A3545**

Department of Information Technology

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu, Approved by AICTE, Accredited by NBA & NAAC)
Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

2022-2023

DEPARTMENT OF INFORMATION TECHNOLOGY

VISION

To become a nationally recognized quality education center in the domain of Computer Science and Information Technology through teaching, training, learning, research and consultancy.

MISSION

- The Department offers undergraduate program in Information Technology to produce high quality information technologists and software engineers by disseminating knowledge through contemporary curriculum, competent faculty and adopting effective teaching-learning methodologies.
- Igniting passion among students for research and innovation by exposing them to real time systems and problems
- Developing technical and life skills in diverse community of students with modern training methods to solve problems in Software Industry.
- Inculcating values to practice engineering in adherence to code of ethics in multicultural and multi discipline teams.

PROGRAM EDUCATIONAL OBJECTIVES

After few years of graduation, the graduates of B. Tech. (IT) Program will be:

1. Enrolled or completed higher education in the core or allied areas of Computer Science and Information Technology or management.
2. Successful entrepreneurial or technical career in the core or allied areas of Computer Science and Information Technology.
3. Continued to learn and to adapt to the world of constantly evolving technologies in the core or allied areas of Computer Science and Information Technology.

PROGRAM OUTCOMES

On successful completion of the Program, the graduates of B. Tech. (IT) Program will be able to:

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES

On successful completion of the program, the graduates of B.Tech. (IT) program will be able to:

PSO1: Design and develop database systems, apply data analytics techniques, and use advanced databases for data storage, processing and retrieval.

PSO2: Apply network security techniques and tools for the development of highly secure systems.

PSO3: Analyze, design and develop efficient algorithms and software applications to deploy in secure environment to support contemporary services using programming languages, tools and technologies.

PSO4: Apply concepts of computer vision and artificial intelligent for the development of efficient intelligent systems and applications.

Institute Vision and Mission

VISION

To be one of the Nation's premier Engineering Colleges by achieving the highest order of excellence in Teaching and Research.

MISSION

- To foster intellectual curiosity, pursuit and dissemination of knowledge.
- To explore students' potential through academic freedom and integrity.
- To promote technical mastery and nurture skilled professionals to face competition in ever increasing complex world.

TRAVEL COMPANY MANAGEMANT SYSTEM

ABSTRACT:

The travel industry is a dynamic and fast-paced sector that relies heavily on efficient management and organization to provide exceptional services to customers. A well-designed and structured database plays a crucial role in ensuring the smooth functioning of a travel company. In the modern digital era, travel companies need to effectively store, manage, and retrieve vast amounts of data related to various aspects of their operations. From customer information and bookings to destinations, transportation options, and payment details, a centralized and structured database is essential for streamlining processes and enhancing the customer experience.

INTRODUCTION:

The Travel Company Database is designed to support the efficient management and organization of travel-related data for a travel company. The database design will consist of several tables, each serving a specific purpose. The "Passenger" table will store customer details such as customer ID, name, contact information, and travel preferences. The "Booking" table will maintain information about bookings made by customers, including booking ID, customer ID, travel dates, and payment status. To manage various travel destinations, the "Destination" table will store details about different locations, including destination ID, name, description, and attractions. The "Package" table will link destinations to specific travel packages, containing package ID, destination ID, package details, and pricing information. Furthermore, the "Passenger" table will track passenger payments, recording payment ID, passenger ID, payment date, and amount. The "Review" table will allow customers to provide feedback and reviews about their travel experiences, including review ID, customer ID, destination ID, and review details. this database design provides a solid foundation for managing travel-related data efficiently. It enables easy retrieval of customer information, booking management, transportation details, payment tracking and customer reviews within the travel company.

TRAVEL MANAGEMANT SYSTEM

- **Administrator:**

  This module provides administrator related functionality, and the administrator manages all the doubts of the customers related problems, manages the whole operations like to add, delete edit and view of the data related to the places, travels, routes and bookings problems in the application.
- **Travels:** This module provides the details of various travel agencies, a user can select the appropriate agency depending on convenience and accessibility.

- **Routes:**

  This module provides information related to the various routes connecting sources and destinations. For each route information such as source, destination, fare, reservation details, pick up, points etc. are provided. Only the administrator can add, delete, edit and manage the data. Users can only view the information.

- **Reservations:**

  The module provides the functionalities that allow an user to book tickets or cancel previously booked tickets, the module maintains the details of all the reservations made so far and allows the administrator to either confirm or reject the bookings.

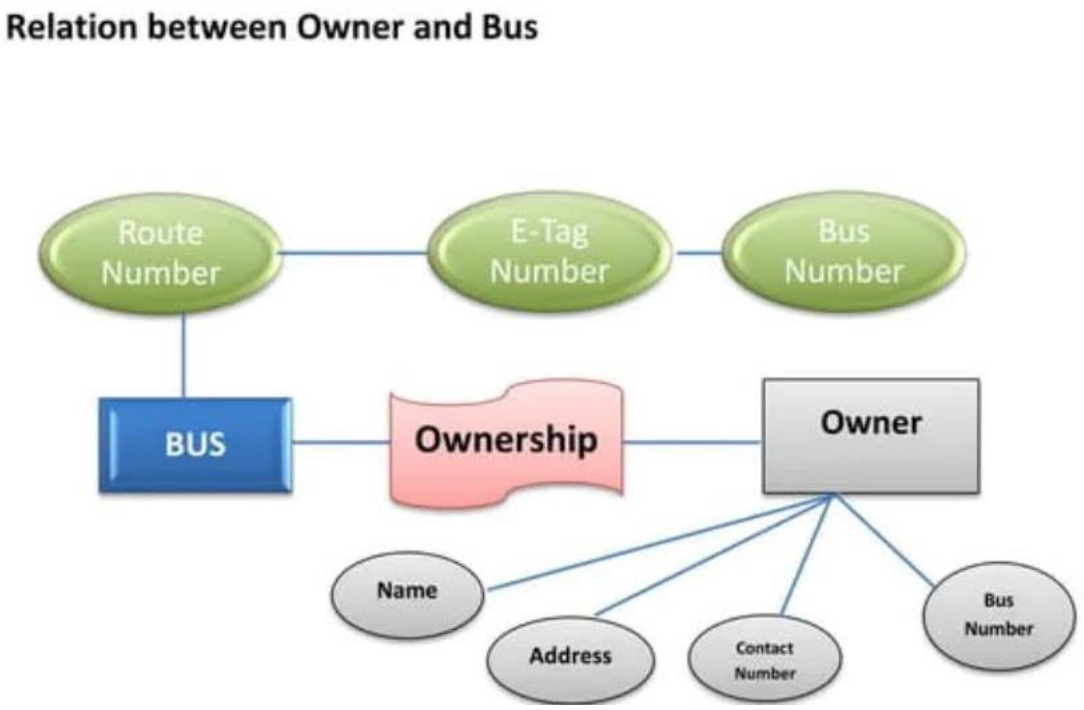
- **Testimonials:**

  Users of this application can post their opinions, complaints and suggestions regarding this portal and serves the administrator. accordingly, the administrator can take various steps to act on the complaints and suggestions.

ENTITIES: Bus, Owner, Government, Agent, Passenger.

ER-DIAGRAM FOR OWNER AND BUS:

The Entity-Relationship(ER) diagram between owner and bus , it represents logical structure of a database system, depicting the entities ,attributes, and cardinalities between them . based on the provided information for the travel management system, here are some details included in the ER-diagram.



Entities:

- Bus
- Owner

Key Attribute:

-Bus-Number: The bus number is unique number should be provided by the RTO.

List of Attributes :

-**Route Number**: is an identifying numeric or alphanumeric designation assigned by a highway authority to a particular stretch.

-E-Tag Number: is a unique number ,it's a road map number.

-Name: Owner's name.

-**Address:** Owner's address.

-Contact number: Owner's number.

❖ BUS TABLE:

The bus relation contains the fields like bus no, bus name, route name and the table were created and structure will be displayed.

Schema:

```
SQL>create table bus(bus no numeric(5),busname varchar(10),routename varchar(10));

SQL>Table created.
SQL> desc bus;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BUSNO                                              NUMBER(5)
 BUSNAME                                            VARCHAR2(10)
 ROUTENAME                                          VARCHAR2(10)
```

❖ OWNER TABLE:

The owner relation contains the fields like owner name, owner no, owner address, and mobile number and with respective bus number and the table was created , and the structure will be displayed.

Schema:

SQL>create table owner(ownerno number(5),ownername varchar(15),oaddress varchar(15),moblieno number(15),busno number(15));

SQL>Table created.

SQL> desc owner;

Name	Null?	Type
OWNERNO		NUMBER(5)
OWNERNAME		VARCHAR2(15)
OADDRESS		VARCHAR2(15)
MOBLIENO		NUMBER (15)
BUSNO		NUMBER (15)

Relationship between bus table and owner table:

Each owner is associated with bus, and each bus has one owner. In the database, you can add a foreign key column in the "Owner" table that references the primary key of the "Bus" table. This foreign key establishes the relationship between the two tables.

TABLE ENTRIES:

BUS TABLE:

The table 1 contains the information about the bus no, bus name, and route name were displayed. The passengers will check the bus details ,and take the bus arrangements.

SQL>select *from bus;

BUSNO	BUSNAME	ROUTENAME
5312	ML travels	Tirupati
6711	KL travels	Bengaluru
8923	AL travels	Kurnool
7654	Noor travels	Hyderabad
9875	SR travels	Mumbai

OWNER TABLE:

The below table consists of company owner details like owner name, address, mobile no and the bus number and as am I discussed above .

SQL>select *from owner;

OWNERNO	OWNERNAME	OADDRESS	MOBLIENO	BUSNO
6563533	Nagaraj	Anantapur	9835634333	5312
6436535	Maruti	Kurnool	9836354764	6711
8975363	Yashoda	Anantapur	7875366537	5443
4633373	Vani	Kurnool	8975427252	9825
4363539	Yash	Bengaluru	9160085672	6527
8725252	Vincent	Kurnool	8142788262	8756

ER-DIAGRAM FOR OWNER AND GOVERNMENT:

The Entity-Relationship(ER) diagram between owner and government represents logical structure of a database system, depicting the entities ,attributes, and cardinalities between them . based on the provided information for the travel management system, here are some details can be included in the ER-diagram.

Entities:

-Government

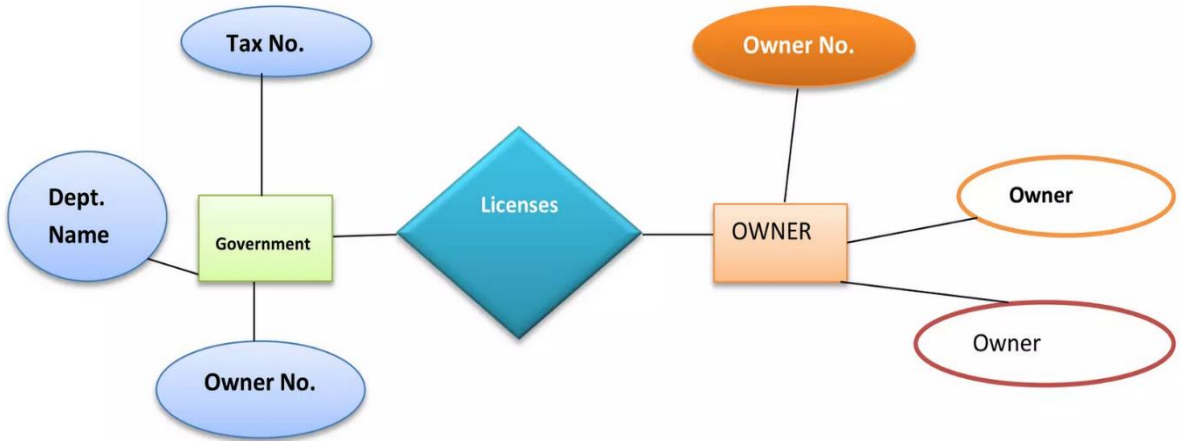
-Owner

Key Attribute: Owner Number

List of Attributes :

- **Tax No:** The bus owner should pay the tax and the RTO issued the Tax No respect to bus number.
- **Dept Name:** The vehicle should be on which department like consumer vehicle or business vehicle.
- **Owner NO:** The Owner's having some particular number on which department.
- **Owner:** Owner details.

Relation between Owner and Government:-



Relationship between Owner entity and Government entity:

Government-relation-owner:

If the travel company has only one owner and interacts with a single government entity, a one-to-one relationship can be established. In this case, you can add a foreign key column in the "Owner" table that references the primary key of the "Government" table. This foreign key establishes the relationship between the owner and the specific government entity.

OWNER TABLE:

The owner relation contains the fields like owner name, owner no, owner address, mobile number and respective bus number then the table will be created, and the structure will be displayed.

Schema:

SQL>create table owner(ownerno number(5),ownername varchar(15),oaddress varchar(15),moblieno number(15),busno number(15));

SQL>Table created.

SQL> desc owner;

Name	Null?	Type
OWNERNO		NUMBER(5)
OWNERNAME		VARCHAR2(15)
OADDRESS		VARCHAR2(15)
MOBLIENO		NUMBER (15)
BUSNO		NUMBER (15)

❖ GOVT TABLE:

The govt relation contains the fields like dept name, owner name, tax no, and mobile number and with respective bus number and the table was created , and the structure will be displayed.

Schema:

```
SQL>   create table govt(deptname varchar(15),ownername varchar(15),taxno number(15));

SQL>Table created.

SQL> desc govt;

Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
DEPTNAME                                           VARCHAR2(15)
OWNERNAME                                          VARCHAR2(15)
TAXNO                                              NUMBER(15)
```

TABLE ENTRIES:

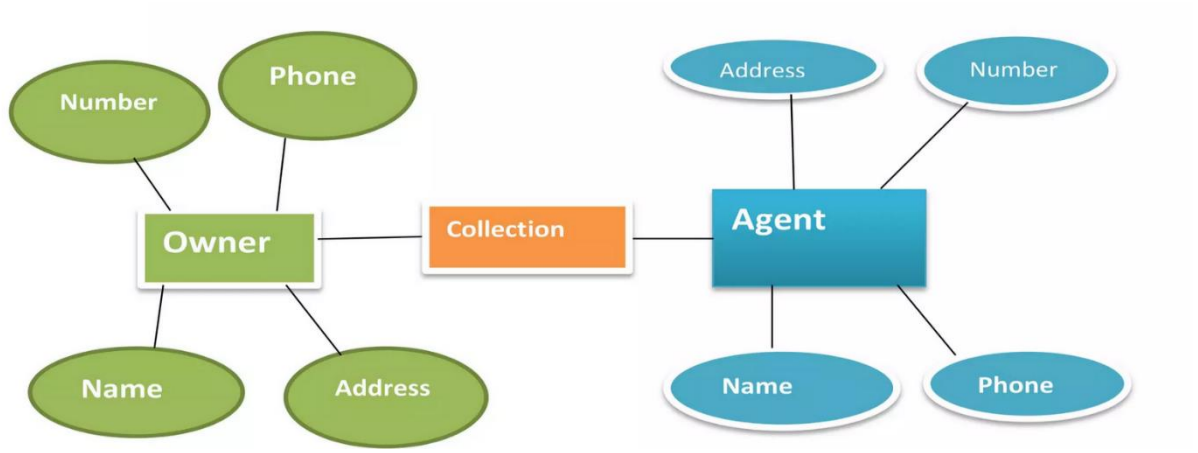
The below table contains the dept name , and owner name. the company owners should pay the tax to the income tax department , according to the tax number and department , and the tax invoice is generated by the database.

SQL>select *from govt;

DEPTNAME	OWNERNAME	TAXNO
Trustier	Nagaraju	987657
Trustier	Maruthi	736353
Trustier	Yashoda	635353
Trustier	Vani	983373
Trustier	Yash	983629
Trustier	Vincent	865363

ER-DIAGRAM FOR OWNER AND AGENT:

The Entity-Relationship(ER) diagram between owner and agent , it represents logical structure of a database system. It depicting the entities ,attributes, and cardinalities between them . Based on the provided information for the travel management system, here are some details included in the ER-diagram.



Entities:

1. Owner
2. Agent

List of Attributes :

- **Phone:** Owner's Phone Number.
- **Address:** Owner's Address.
- **Name:** Owner's Name.
- **AAddress:** Agent's Address.
- **AName:** Agent's Name.
- **ANumber:**Agent's Number.

Relationship between Owner entity and Agent entity:

In this scenario, each owner is associated with exactly one agent, and each agent is assigned to multiple owners. This relationship can be represented by adding a foreign key column in the "Owner" table that references the primary key of the "Agent" table. This foreign key establishes the relationship between the owner and the assigned agent.

OWNER TABLE:

The owner relation contains the fields like owner name, owner no, owner address, and mobile number and with respective bus number and the table was created , and the structure will be displayed.

Schema:

SQL>create table owner(ownerno number(5),ownername varchar(15),oaddress varchar(15),moblieno number(15),busno number(15));

SQL>Table created.

SQL> desc owner;

Name	Null?	Type
OWNERNO		NUMBER(5)
OWNERNAME		VARCHAR2(15)
OADDRESS		VARCHAR2(15)
MOBLIENO		NUMBER (15)
BUSNO		NUMBER (15)

❖ AGENT TABLE:

The agent relation contains the fields like agent name, agent address, and mobile number and with respective bus number and the table was created , and the structure will be displayed.

Schema:

SQL> create table agent(a name varchar(10),address varchar(15),ano number(10));

SQL>Table created.

SQL> desc agent;

Name	Null?	Type
ANAME		VARCHAR2(10)
AADDRESS		VARCHAR2(15)
ANO		NUMBER(15)

TABLE ENTRIES:

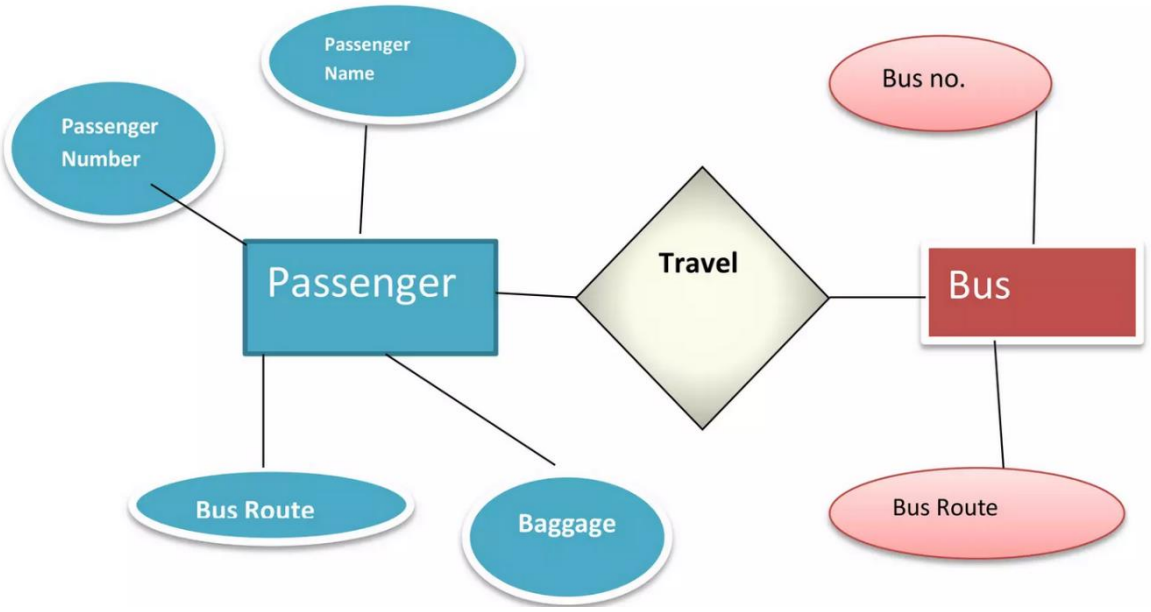
The table consists of the agent's details , the table helps us the give the details to the passengers who's don't the bus and ticket details respectively.

SQL> select *from agent;

ANAME	AADDRESS	ANO	ROUTENAME
Mokshith	Anantapur	783	Tirupati
Raja	Kurnool	876	Bengaluru
Tanveez	Anantapur	764	Kurnool
Abhi	Tirupati	346	Hyderabad
Giri	Tirupati	433	Mumbai
Rocky	Kurnool	453	Chennai

ER-DIAGRAM FOR PASSENEGR AND BUS:

The Entity-Relationship(ER) diagram between passenger and bus , it represents logical structure of a database system, depicting the entities ,attributes, and cardinalities between them . based on the provided information for the travel management system, here are some details can be included in the ER-diagram.



Key Attribute: Bus Route

List Of Attributes:

- Passenger Name: The Database stores the passenger's name.
- Passenger No: The database will give the respective id to reach destination place.
- Bus Route: The Bus team members will store the bus route details , where did the bus go to place.

- Baggage: The passenger's luggage.
- Bus Number: Unique identity of bus.

Relationship between bus entity and passenger entity:

In this scenario, each passenger is associated with exactly one bus, and each bus can accommodate on more passengers. This relationship can be represented by adding a foreign key column in the "Passenger number" table that references the primary key of the "Bus" table. This foreign key establishes the relationship between the passenger and the assigned bus.

❖ BUS TABLE:

The bus relation contains the fields like bus no, bus name, route name and the table were created and structure will be displayed.

Schema:

SQL>create table bus(bus no numeric(5),busname varchar(10),routename varchar(10));

SQL>Table created.
SQL> desc bus;

Name	Null?	Type
BUSNO		NUMBER(5)
BUSNAME		VARCHAR2(10)
ROUTENAME		VARCHAR2(10)

❖ PASSENSGER TABLE:

The passenger relation contains the fields like name, passenger no, route name and mobile number and with respective and the table was created, and the structure will be displayed.

SQL>create table passenger(psgno numeric(10),psgname varchar(10),psgaddress varchar(15),routename varchar(15));
Table created.

desc passeneger;

Name	Null?	Type
PSGNO		NUMBER(10)
PSGNAME		VARCHAR2(10)
PSGADDRESS		VARCHAR2(15)
ROUTENAME		VARCHAR2(15)
AMOUNT		NUMBER(15)

Schema:

SQL> create table agent(a name varchar(10),address varchar(15),ano number(10));

SQL>Table created.

SQL> desc agent;

Name	Null?	Type
ANAME		VARCHAR2(10)
AADDRESS		VARCHAR2(15)
ANO		NUMBER(15)

TABLE ENTRIES:

The below table contains whose passengers going to his/her booking destination place with some other details.

SQL> select *from passeneger;

PSGNO	PSGNAME	PSGADDRESS	ROUTENAME
674647	Yaswanth	Anantapur	Tirupati
536754	Vincent	Anantapur	Tirupati
894573	prasad	Bathalapalli	Bengaluru
458732	Santhosh	Dharam avaram	Tirupati
982358	Kushal	Madanapalli	Tirupati
436367	Eswar Sai	Kurnool	Tirupati
320987	Asif khan	Anantapur	Bengaluru
278964	Shravani	Bathalapalli	Bengaluru
563536	Usha	Anantapur	Bengaluru
453363	Thulasi	Anantapur	Tirupati
873633	Renuka	Kurnool	Kurnool
683733	Naveen	Anantapur	Mumbai
435353	Sandhya	Bathalapalli	Mumbai
673373	Navya	Kurnool	Bengaluru

JOIN:

To query the passengers traveling on a specific bus in a travel company database, you can use a SQL query. Assuming you have tables named "Passenger" and "Bus" with appropriate columns, and there is a foreign key column "Route name " in the "Passenger" table referencing the primary key "Route name " in the "Bus" table, you can use the following query:

SQL>select psgno,psgname,busno,busname from passeneger p inner join bus b on(p.routename=b.routename)

- The passenger ID generated by the system.
- The Route name matches in bus details of route name.
- The system generates the amount.
- The database will share agent's details.

PSGNO	PSGNAME	BUSNO	BUSNAME
674647	Yaswanth	5312	ML Travels
536754	Vincent	5312	ML Travels
894573	Prasad	6711	KL Travels
458732	Santhosh	6711	KL Travels
982358	Kushal	5312	ML Travels
436367	Eswarsai	5312	ML Travels
320987	Asifkhan	5312	ML Travels
278964	Shravani	6711	K L Travels
563536	Usha	6711	KL Travels
453363	Thulasi	5312	ML Travels
873633	Renuka	8923	AL Travels
683733	Naveen	9875	SR Travels
435353	Sandhya	9875	SR Travels
673373	Navya	8923	AL Travels

QUERY:

To query the owners of a travel company who need to pay taxes to the government for a specific bus, you can use a SQL query. Assuming you have tables named "Owner", "Bus", and "Government" with appropriate columns, and there are foreign key columns "Bus no" in the "Bus" table referencing the primary key "Bus no" in the "Owner" table, and "Bus no" in the "Owner" table referencing the primary key "Bus no" in the "Government" table, you can use the following query:

SQL> select *from govt;

DEPTNAME	OWNERNAME	TAXNO	BUSNO	PAYTAX
trustier	Nagaraj	987657	5312	60000
trustier	Maruti	736353	6711	70000
touristed	Jagan	986760	9878	90000

SQL> select *from owner;

OWNERNO	OWNERNAME	OADDRESS	MOBLIENO	BUSNO
6563533	Nagaraj	Anantapur	9835634333	5312
436535	Maruti	Kurnool	9836354764	6711
676546	Nagur	Dharamvaram	7875466554	9868

SQL>select deptname,paytax,ownerno,moblieno from govt g inner join owner o on(g.busno=o.busno);

DEPTNAME	PAYTAX	OWNERNO	MOBLIENO
Trousiter	60000	6563533	9835634333
Trustier	70000	436535	9836354764

.SQL>select deptname,paytax,ownerno,moblieno from govt g right outer join owner o on(g.busno=o.busno)

;

DEPTNAME	PAYTAX	OWNERNO	MOBLIENO
Trustier	60000	6563533	9835634333
Trustier	70000	436535	9836354764
Trustier	90000		

Query: Display the name and route name of the passenger who is in Tirupati.

SQL> select psgname,routename from passeneger where amount>500 group by psgname,routename having routename='tirupathi';

PSGNAME	ROUTENAME
Eswarsai	Tirupati
Vincent	Tirupati
Kushal	Tirupati
Yaswanth	Tirupati
Asifkhan	Tirupati

The above query is based upon whose passengers going to his places, so whether the passengers will be going to Tirupathi and display them.

QUERY:

To query the passengers who need to meet the agent on a specific bus in a travel company database, you can use a SQL query. Assuming you have tables named "Passenger", and "Agent" with appropriate columns, and there are foreign key columns "Route name" in the "Passenger" table referencing the primary key "Route name" in the "Agent" table, you can use the following query:

SQL>select psgno,psgname,aname,ano from passeneger p inner join agent b on(p.routename=b.routename);

PSGNO	PSGNAME	ANAME	ANO
674647	Yaswanth	Mokshith	783
536754	Vincent	Mokshith	783
894573	Prasad	Raja	876
458732	Santhosh	Raja	876
982358	kushal	Mokshith	783
436367	Eswarsai	Mokshith	783
320987	Asifkhan	Mokshith	783
278964	Sharvani	Raja	876
873633	Renuka	Tanveez	764
435353	Sandhya	Giri	433
683733	Naveen	Abhi	346
563536	Usha	Mokshith	783

Query: Display the name and route name of the passenger who is not in Tirupati

SQL> select psgname,routename from passeneger where amount>500 group by psgname,routename having routename='bengaluru';

PSGNAME	ROUTENAME
Santhosh	Bengaluru
prasad	Bengaluru
Shravani	Bengaluru

The above query is based upon whose passengers going to his places, so whether the passengers will be going to Bengaluru and display them.

Query: What is the total amount spent for the specific destination place?

SQL>select routename,sum(amount) as tmt from passeneger group by routename;

ROUTENAME	TMT
Bengaluru	2800
Tirupati	3740

The above query is based upon we calculate the total amount for the Bengaluru and Tirupathi.

Query: How many passengers visited each destination place?

SQL>select *from passenger count(passenger);

ROUTENAME	COUNT
Bengaluru	4
Tirupati	6
Kurnool	3
Hyderabad	4
Mumbai	2

Query: List out the order format of amount for passenger database table.

Order by:

SQL> select *from passenger order by routename;

PSGNO	PSGNAME	PSGADDRESS	ROUTENAME	AMOUNT
458732	Santhosh	Dharam avaram	Bengaluru	900
278964	Shravani	bathalapall	Bengaluru	1000
894573	Prasad	bathalapall	Bengaluru	900
320987	Asifkhan	Anantapur	Tirupati	690
982358	Kushal	madanapalli	Tirupati	700
436367	Eswarsai	Kurnool	Tirupati	650
536754	Vincent	Anantapur	Tirupati	900
674647	Yaswanth	Anantapur	Tirupati	800

PL/SQL Code:

```
CREATE OR REPLACE PROCEDURE add_psgno_psgname(
  p_psgno NUMBER,
  p_psgname VARCHAR2
) AS
BEGIN
  INSERT INTO passeneger (PSGNO, PSGNAME) VALUES (p_psgno, p_psgname);
  COMMIT;
END;
/
CREATE OR REPLACE PROCEDURE update_passenger_amount(
  p_psgno NUMBER,
  p_amount NUMBER
) AS
BEGIN
  UPDATE passeneger SET AMOUNT = p_amount WHERE PSGNO = p_psgno;
  COMMIT;
END;
/
CREATE OR REPLACE FUNCTION get_passenger_count RETURN NUMBER AS
  v_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_count FROM passeneger;
  RETURN v_count;
END;
/
CREATE OR REPLACE FUNCTION get_passengers_in_tirupathi RETURN SYS_REFCURSOR AS
  v_result SYS_REFCURSOR;
BEGIN
  OPEN v_result FOR
```

```
    SELECT * FROM passeneger WHERE ROUTENAME = 'tirupathi';
  RETURN v_result;
END;
/
CREATE OR REPLACE FUNCTION get_passengers_in_bengaluru RETURN SYS_REFCURSOR AS
  v_result SYS_REFCURSOR;
BEGIN
  OPEN v_result FOR
    SELECT * FROM passeneger WHERE ROUTENAME = 'bengaluru';
  RETURN v_result;
END;
/
CREATE OR REPLACE PROCEDURE add_passenger(
  p_psgno NUMBER,
  p_psgname VARCHAR2,
  p_psgaddress VARCHAR2,
  p_routename VARCHAR2,
  p_amount NUMBER
) AS
BEGIN
  INSERT INTO passeneger (PSGNO, PSGNAME, PSGADDRESS, ROUTENAME, AMOUNT)
  VALUES (p_psgno, p_psgname, p_psgaddress, p_routename, p_amount);
  COMMIT;
END;
/
DECLARE
  choice NUMBER;
  psgno NUMBER;
  psgname VARCHAR2(50);
  amount NUMBER;
```

```
  psg_count NUMBER;
  psg_cursor SYS_REFCURSOR;
  psg_row passeneger%ROWTYPE;
BEGIN
  -- Prompt user for query type
  DBMS_OUTPUT.PUT_LINE('Select the type of query:');
  DBMS_OUTPUT.PUT_LINE('1. Add the psgno and psgname');
  DBMS_OUTPUT.PUT_LINE('2. Update the amount of selected passenger');
  DBMS_OUTPUT.PUT_LINE('3. Show the passenger count');
  DBMS_OUTPUT.PUT_LINE('4. Show the passengers who are in Tirupathi');
  DBMS_OUTPUT.PUT_LINE('5. Show the passengers who are in Bengaluru');
  DBMS_OUTPUT.PUT_LINE('6. Add one more passenger details');
  DBMS_OUTPUT.PUT('Enter your choice (1-6): ');
  choice := &choice;

  -- Execute the selected query
  CASE choice
    WHEN 1 THEN
      DBMS_OUTPUT.PUT('Enter PSGNO: ');
      psgno := &psgno;
      DBMS_OUTPUT.PUT('Enter PSGNAME: ');
      psgname := '&psgname';
      add_psgno_psgname(psgno, psgname);
    WHEN 2 THEN
      DBMS_OUTPUT.PUT('Enter PSGNO: ');
      psgno := &psgno;
      DBMS_OUTPUT.PUT('Enter AMOUNT: ');
      amount := &amount;
      update_passenger_amount(psgno, amount);
    WHEN 3 THEN
```

```
    psg_count := get_passenger_count();
    DBMS_OUTPUT.PUT_LINE('Passenger Count: ' || psg_count);
  WHEN 4 THEN
    psg_cursor := get_passengers_in_tirupathi();
    LOOP
      FETCH psg_cursor INTO psg_row;
      EXIT WHEN psg_cursor%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('PSGNO: ' || psg_row.PSGNO || ', PSGNAME: ' || psg_row.PSGNAME);
    END LOOP;
    CLOSE psg_cursor;
  WHEN 5 THEN
    psg_cursor := get_passengers_in_bengaluru();
    LOOP
      FETCH psg_cursor INTO psg_row;
      EXIT WHEN psg_cursor%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('PSGNO: ' || psg_row.PSGNO || ', PSGNAME: ' || psg_row.PSGNAME);
    END LOOP;
    CLOSE psg_cursor;
  WHEN 6 THEN
    DBMS_OUTPUT.PUT('Enter PSGNO: ');
    psgno := &psgno;
    DBMS_OUTPUT.PUT('Enter PSGNAME: ');
    psgname := '&psgname';
    DBMS_OUTPUT.PUT('Enter PSGADDRESS: ');
    psgname := '&psgaddress';
    DBMS_OUTPUT.PUT('Enter ROUTENAME: ');
    psgname := '&routename';
    DBMS_OUTPUT.PUT('Enter AMOUNT: ');
    amount := &amount;
    add_passenger(psgno, psgname, psgaddress, routename, amount);
```

```
    ELSE
      DBMS_OUTPUT.PUT_LINE('Invalid choice.');
  END CASE;
END;
/
```

OUTPUT:

Select the type of query:

1. Add the psgno and psgname.
2. Update the amount of selected passenger.
3. Show the passenger count.
4. Show the passengers who are in Tirupathi.
5. Show the passengers who are in Bengaluru.
6. Add one more passenger details.
7. Enter your choice (1-6):

1

```
Enter value for psgno: 84748
Enter value for psgname:  jhanvi.
old   92:   VALUES (&psgno, '&psgname',)
new   92:   VALUES (84748, ' jhanvi ')
1 row created.
  SQL> /
```

Select the type of query:

1. Add the psgno and psgname.
2. Update the amount of selected passenger.
3. Show the passenger count.
4. Show the passengers who are in Tirupathi.
5. Show the passengers who are in Bengaluru.
6. Add one more passenger details.
7. Enter your choice (1-6):

ROUTENAME	COUNT
Bengaluru	4
Tirupati	6
Kurnool	3
Hyderabad	4
Mumbai	2

SQL> /

Select the type of query:

1. Add the psgno and psgname.
2. Update the amount of selected passenger.
3. Show the passenger count.
4. Show the passengers who are in Tirupathi.
5. Show the passengers who are in Bengaluru.
6. Add one more passenger details.
7. Enter your choice (1-6):

end

Conclusion:

A database management system (DBMS) is crucial for application in a travel company. It provides a robust foundation for data storage, organization, and retrieval. Key benefits include streamlined operations, efficient data management, improved customer service, and enhanced data integrity and security. Implementing a DBMS application for a travel company enables effective management of passenger details, bookings, routes, and other essential information, leading to a more efficient and reliable system overall.

REFERENCES:

1. Elmasri, R., & Navathe, S. B. Fundamentals of Database Systems. Pearson, (2019)..
2. Coronel, C., Morris, S., & Rob P. Database Systems: Design, Implementation, & Management. Cengage Learning, (2016).
3. Saha, S. K., & Ray, S. A Comprehensive Travel Management System (CHMS) Design and Development. International Journal of Emerging Technology and Advanced Engineering, 6(12), 85-91 (2016).
4. Akter, S., & Rahman, M. S. Travel company Management System: A Perspective View. International Journal of Computer Science and Information Security, 14(2), 27-33,(2016).
5. Gupta, R., & Anand, A. (2019). Travel Management System Using Database Technology: An Empirical Study. International Journal of Advanced Research in Computer Science, 10(5), 257-261.

Web Links:

- https://www.slideshare.net/AwaisAliAlHussaini/bilal-travel-dbms
- https://studentprojectguide.com/project-report/database-design/travel-agency-management-system-table-design/
- https://github.com/praveenhonavar/Travel-Agency-Management-System
- https://github.com/praveenhonavar/Travel-Agency-Management-System