****

**DATA BASE MANAGEMENT SYSTEM**

**Railway Management System**



**Table of Contents:**

****

|  |  |  |
| --- | --- | --- |
| **S. No** | **Section Reference** | **Page Number** |
| **1** | **Introduction of The Database System** | **2** |
| **2** | **ER Diagram Modelling** | **3** |
| **3** | **Entities, Attributes and Relationships** | **4** |
| **4** | **Process of Normalization** | **6** |
| **5** | **Database Schema** | **7** |
| **6** | **Data Definition Language (DDL)** | **8** |
| **7** | **Inserting Values** | **9** |
| **8** | **Sample Table Overviews** | **10** |
| **9** | **Sample Queries** | **13** |
| **10** | **Conclusion** | **15** |

****

**PAGE 1**

**Introduction of the Database System:**

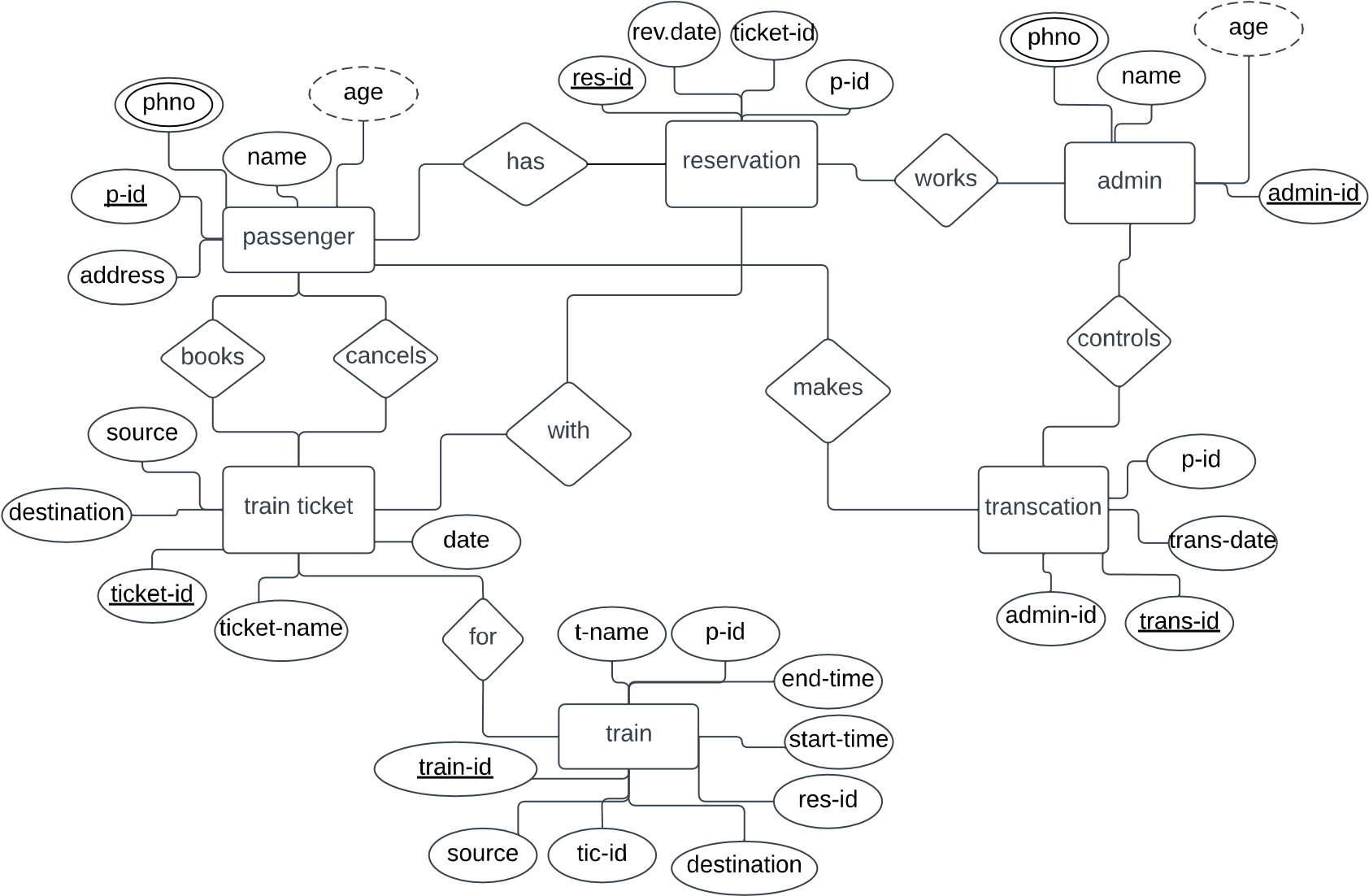
**The booming phenomenon of e-commerce has today assumed an extremely vital and pivotal role in our daily lives. It enhances our convenience by allowing us access to varied resources, all from the comfort of our home.**

A Railway Management System (RMS) is a computerized system designed to manage and store data related to railway operations. It acts as the central repository for information such as train schedules, ticket booking, passenger details, maintenance records, and financial data. The RMS helps to streamline railway operations, increase efficiency, and provide real-time information to stakeholders such as train dispatchers, ticket agents, and maintenance personnel. The database can be accessed by authorized personnel through a user-friendly interface, making it easier to retrieve and analyze data, make informed decisions, and improve overall railway management**.**

**The main ideology is to develop a completely automated, database to work more efficiently and accurately to serve the passengers as much early as possible and reduce the use of man power.**

**PAGE 2**

**ER Diagram Modelling:**



**PAGE 3**

**Entities, Attributes and Relationships:**

**Entity**: An entity is an object that exists. It doesn't have to do anything; it just has to exist. In database administration, an entity can be a single thing, person, place, or object. Data can be stored about such entities. A design tool that allows database administrators to view the relationships between several entities is called the entity relationship diagram (ERD).

**The following Entities are used in our Database:**

**1. Passengers**

**2. Reservation**

**3. Admin**

**4. Ticket**

**5. Transaction**

**a**

**Attribute**:

An attribute defines the information about the entity that needs to be stored. If the entity is an employee, attributes could include name, employee ID, health plan enrolment, and work location. An entity will have zero or more attributes, and each of those attributes apply only to that entity**.**

**Relationship:**

A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

**Following is a detailed description of every Entity, its Attributes and**

**Relations in between them as employed in our Database:**

**PAGE 4**

**1.Passenger:**

Each Passenger entity passenger\_id, name, age,phone-number , address.

* One Passenger can book n tickets and can cancel m tickets.
* N passengers can have m reservations.
* One passenger can make n transactions**.**

**2. Reservation:**

Each Reservation entity is associated with reservation\_id,reservation\_date,passenger\_id, ticket\_id.

* + - 1 train ticket can access for n reservations.
    - 1 passenger can have 1 reservation.
    - M Reservations can be handled by n admins;

**3. Admin:**

Admin Entity consists of Admin\_id(primary key),name of the admin, age of admin,Contactdetails.

* + - N admins works for m reservation compartments.
    - M admin controls n transactions.

**4. Train Ticket:**

Each Train ticket entity associated with ticket\_id, destination, ticket\_name,date.

* One train-ticket has one train name.
* N train tickets can be booked by 1 passenger ,m tickets can be cancelled by 1 passenger.
* 1 train ticket can consists of m reservations.

**5. Transaction:**

Transaction Entity has transaction\_id, transaction\_date, admin\_id, passenger\_id.

* 1 passenger can make n transactions.
* N admins can control m transactions.

**6. Train:**

Train Entity consists of transaction\_id , source , ticket\_id , ticket\_name , date , destination , start\_time , end\_time , passenger\_id , train\_name.

* one train ticket has one train name.

**PAGE5**

**Process of Normalization:**

**Normalization:**

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and update-based anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

There exist three main types of Normal forms, each being associated with a increasing degree of Normalization:

**First Normal Form:**A relation is in first normal form if every attribute in that relation is singled valued attribute**.**

**Ex:** In our database, consider the attribute Serial Number under entitypassengerThe numbers are not multi-valued and are distinct. Hence, it isin 1NF.

**Second Normal Form:**A relation is in second normal form if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

**Ex**: In our database, consider all attributes under the entity Passenger. There exist no partial dependencies throughout them. Hence, in 2NF.

**Third Normal Form:** A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

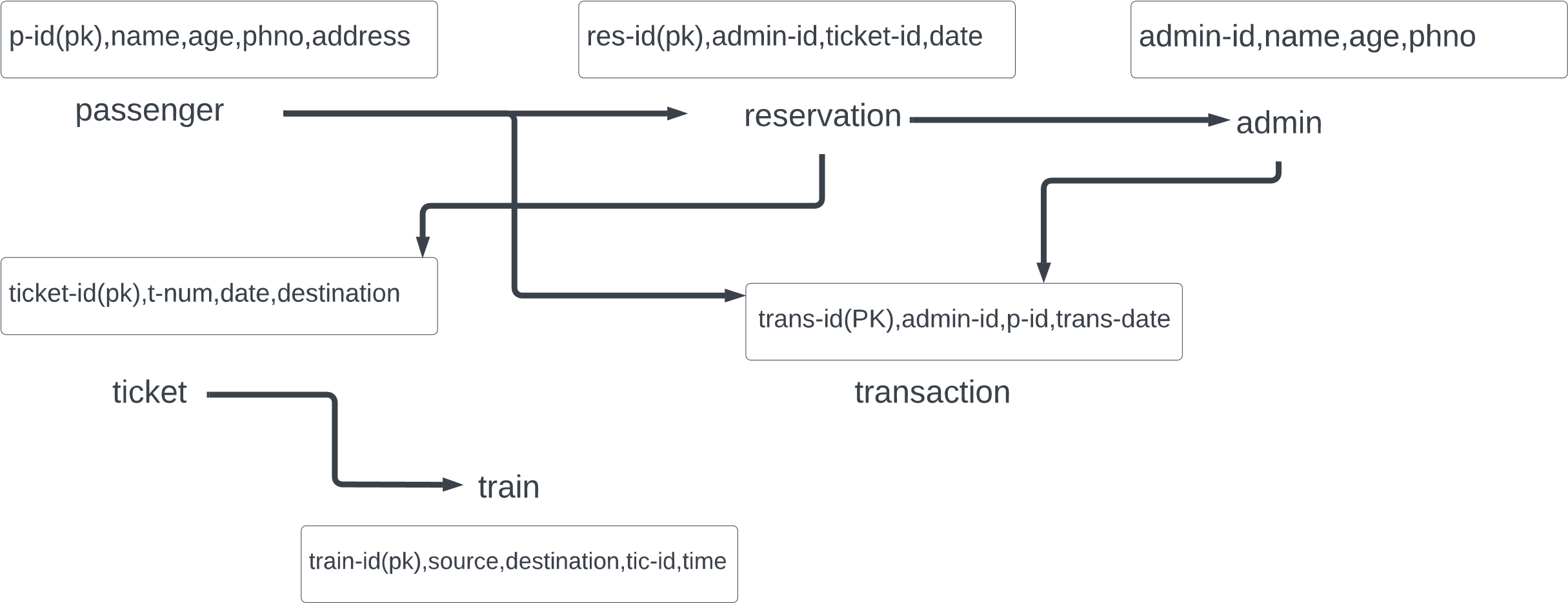
**Ex**: In our database, consider all attributes under the entityPassenger. There exist no transitive dependencies throughout them.Also, they are in 2NF and hence, are also in 3NF.

Apart from the entity cart, in the entire database, there exist no attributes which exhibit any feature of redundancy. Furthermore, each attribute satisfies all the above-mentioned Normal forms, thereby eliminating the need for any further Normalization. Thereby, we can move on to the actual representation of the Database schema.

**PAGE 6**

**Database Schema:**

**The following is an overview of the entire database schema, with individual table schemas defined.**



**PAGE 7**

**Data Definition Language:**

**1.Passenger:**

create table pasenger(p\_id int not null primary key,name varchar(20) not null,age int not null,phno varchar(20),address varchar(20));

**2.Admin:**

create table admin(admin\_id int not null primary key,name varchar(20) not null,age int not null,phno varchar(20));

**3. Train Ticket:**

create table Train\_ticket(ticket\_id int not null primary key,ticket\_name varchar(20) not null,source varchar(20) not null,destination varchar(20) not null,date date);

**4.Transaction:**

create table transaction(trans\_id int not null primary key,trans\_date date,p\_id int,admin\_id int,CONSTRAINT fk\_key foreign key(p\_id) references pasenger(p\_id));

alter table transaction add constraint fk\_key1 foreign key(admin\_id) references admin(admin\_id);

**5.Reservation:**

create table reservation(res\_id int not null primary key,date\_reserve date,p\_id int,admin\_id int,ticket\_id int,CONSTRAINT fk\_key2 foreign key(p\_id) references pasenger(p\_id));

alter table reservation add constraint fk\_key3 foreign key(admin\_id) references admin(admin\_id);

alter table reservation add constraint fk\_key4 foreign key(ticket\_id) references Train\_ticket(ticket\_id);

**6.Train:**

create table Train(train\_id int not null primary key,train\_name char(255),source varchar(20),destination varchar(20),start\_time varchar(20),end\_time varchar(20),p\_id int,admin\_id int,ticket\_id int,CONSTRAINT fk\_key5 foreign key(p\_id) references pasenger(p\_id));

alter table Train add constraint fk\_key6 foreign key(admin\_id) references admin(admin\_id);

alter table Train add constraint fk\_key7 foreign key(ticket\_id) references Train\_ticket(ticket\_id);

**PAGE 8**

**Inserting Values:**

**Following are a few sample queries being inserted into each of the tables to build up a mock database model.**

**1.Passenger:**

Insertintopasenger(p\_id,name,age,phno,addressvalues(1,'Mahesh',19,'901299999','Hyderabad'),(2,'Arun',18,'908737638','JNTU'),(3,'Lokesh',20,'7866907690','Kompally'),(4,'Tejaswini',50,'87868960','Gajwel'),(5,'Anjan',6,'87657687','Nizamabad');

**2.Admin:**

Insertintoadmin(admin\_id,name,age,phno)values(10,'Ram',30,'56789034'),(11,'Kiran',33,'3678290'),(12,'Charan',28,'567890987'),(13,'Chitra',35,'67893098'),(14,'Julan',26,'98765445');

**3.Train Ticket:**

InsertintoTrain\_ticket(ticket\_id,ticket\_name,source,destination,date)values(100,'1TierAC','Hyd','Khammam','20231020'),(101,'2TierAC','Gajwel','Hyd','20230129'),(102,'3TierAC','Nizamabad','Mumbai','20230307'),(103,'Sleeper','Delhi','Kanyakumari','20230601'),(104,'General','Dispur','Gandhinagar','2023-02-15');

**4.Transaction:**

insert into transaction(trans\_id,trans\_date,p\_id,admin\_id)values(74328,'2023-01-01',1,10),(74302,'2023-03-25',2,11),(74380,'2023-06-10',3,10), (74386,'20230419',4,13),(76852,'2023-10-08',5,10);

**5.Reservation:**

insert into reservation(res\_id,date\_reserve,p\_id,admin\_id,ticket\_id)values(200,'2023-1020',1,10,100),(201,'20230129',2,11,101),(202,'2023-03-07',3,12,102),(203,'2023-06-01',4,13,103),(204,'20230215',5,14,104);

**6.Train:**

Insertintotrain(train\_id,train\_name,source,destination,start\_time,end\_time,p\_id,admin\_id,ticket\_id)values(17532,'Howra','Hyd','Khammam','10:15AM','9;30PM',1,10,100),(17861,'Ajanta','Gajwel','Hyd','6:20AM','2:00PM',2,11,101),(18368,'Vandebharat','Nizamabad','Mumbai','7:15AM','11:50PM',3,10,102),(17364,'Rajadhani','Delhi','Kanyakumari','8:30AM','12:30AM',4,13,103),(17296,'Ajmer','Dispur','Gandhinagar','5:45AM','3:30AM',5,10,104);

**PAGE 9**

**Sample Table Overviews:**

**1.Passenger:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p\_id | name | age | phno | address |
| 1 | **Mahesh** | **19** | **901299999** | **Hyderabad** |
| 2 | **Arun** | **18** | **908737638** | **JNTU** |
| 3 | **Lokesh** | **20** | **786690760** | **Kompally** |
| 4 | **Tejaswini** | **50** | **876576833** | **Gajwel** |
| 5 | **Anjan** | **6** | **653552635** | **Nizambad** |

**2.Admin:**

|  |  |  |  |
| --- | --- | --- | --- |
| Admin\_id | name | age | phno |
| 10 | **Ram** | **30** | **56789034** |
| 11 | **Kiran** | **33** | **3678290** |
| 12 | **Charan** | **28** | **567890987** |
| 13 | **Chitra** | **35** | **67893098** |
| 14 | **Julan** | **26** | **98765445** |

**PAGE 10**

**3.Train Ticket:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ticket\_id | ticket-name | source | destination | date |
| 100 | **1Tier-AC** | **Hyd** | **Khammam** | **2023-10-20** |
| 101 | **2Tier-AC** | **Gajwel** | **Hyd** | **2023-01-29** |
| 102 | **3Tier-AC** | **Nizamabad** | **Mumbai** | **2023-03-07** |
| 103 | **Sleeper** | **Delhi** | **Kanyakumari** | **2023-06-01** |
| 104 | **General** | **Dispur** | **Gandhinagar** | **2023-02-15** |

**4.Transaction:**

|  |  |  |  |
| --- | --- | --- | --- |
| trans\_id | trans\_date | p\_id | admin\_id |
| 74302 | **2023-03-25** | **2** | **11** |
| 74328 | **2023-01-01** | **1** | **10** |
| 74380 | **2023-06-10** | **3** | **10** |
| 74386 | **2023-04-19** | **4** | **13** |
| 76852 | **2023-10-08** | **5** | **10** |

**PAGE 11**

**5.Reservation:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| res\_id | date\_reserve | p\_id | admin\_id | ticket\_id |
| 200 | **2023-10-20** | **1** | **10** | **100** |
| 201 | **2023-01-29** | **2** | **11** | **101** |
| 202 | **2023-03-07** | **3** | **12** | **102** |
| 203 | **2023-06-01** | **4** | **13** | **103** |
| 204 | **2023-02-15** | **5** | **14** | **104** |

**6.Train:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| train\_id | train\_name | source | destination | start\_  time | end\_time | p\_id | admin\_  id | ticket\_  id |
| 17296 | **Ajmer** | **Dispur** | **Gandhinagar** | **5:45AM** | **3:30AM** | **5** | **10** | **104** |
| 17364 | **Rajadhani** | **Delhi** | **Kanyakumari** | **8:30AM** | **12:30AM** | **4** | **13** | **103** |
| 17532 | **Howra** | **Hyd** | **Khammam** | **10:15AM** | **9:30AM** | **1** | **10** | **100** |
| 17861 | **Ajanta** | **Gajwel** | **Hyd** | **6:20AM** | **2:00PM** | **2** | **11** | **101** |
| 13368 | **Vandebharat** | **Nizambad** | **Mumbai** | **7:15AM** | **11:50PM** | **3** | **10** | **102** |

**PAGE 12**

**Sample Queries:**

**1.**Create view alexa as selecttt.ticket name,tt.source,t.destination, t.ticket\_id,t.train\_name from train t,train\_ticket tt where t.ticket\_id =tt.ticket\_id;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ticket\_id | Ticket\_name | source | destination | Train\_name |
| 100 | **1Tier-AC** | **Hyd** | **Khammam** | **Howra** |
| 101 | **2Tier-AC** | **Gajwel** | **Hyd** | **Ajanta** |
| 102 | **3Tier-AC** | **Nizamabad** | **Mumbai** | **Vandebharat** |
| 103 | **Sleeper** | **Delhi** | **Kanyakumari** | **Rajadhani** |
| 104 | **General** | **Dispur** | **Gandhinagar** | **Ajmer** |

2.select \* from alexa where ticket\_id<102;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ticket\_id | Ticket\_name | source | destination | Train\_name |
| 100 | **1Tier-AC** | **Hyd** | **Khammam** | **Howra** |
| 101 | **2Tier-AC** | **Gajwel** | **Hyd** | **Ajanta** |

**3.** select p.name,t.train\_name,r.res\_id from pasenger p inner join train t on p.p\_id=t.p\_id inner join reservation r on t.ticket\_id=r.ticket\_id and t.ticket\_id < 103;

|  |  |  |
| --- | --- | --- |
| res\_id | name | Train\_name |
| 200 | **Mahesh** | **Howra** |
| 201 | **Arun** | **Ajanta** |
| 202 | **Lokesh** | **Vandebharat** |

**PAGE 13**

**4.**select pasenger.name,transaction.trans\_date from pasenger inner join transaction on pasenger.p\_id=transaction.p\_id;

|  |  |
| --- | --- |
| name | Trans\_date |
| Mahesh | **2023-01-01** |
| Arun | **2023-03-25** |
| Lokesh | **2023-06-10** |
| Tejaswini | **2023-10-08** |
| Anjan | **2023-10-08** |

**5.** select a.name from admin a,pasenger p where a.age>(select age from pasenger where name='Mahesh') group by a.name;

|  |
| --- |
| name |
| Julan |
| Chitra |
| Charan |
| Kiran |
| Ram |

**PAGE 14**

**Conclusion:**

In our project Railway ManagementSystem we have stored all the information about the Trains scheduled and the users booking tickets and even status of trains, seats etc. This data base 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 inconviniences of going to railway station for each and every query they get. We had considered the most important requriments only, many more features and details cand be added to our project inorder to obtain even more user friendly applications. These applications are already in progress and in future they can be upgraded and may become part of amazing technology.

**PAGE 15**