create table CUSTOMER  
(  
 CUSTOMERID NUMBER generated as identity  
 primary key,  
 NAME VARCHAR2(50) not null,  
 SURNAME VARCHAR2(50) not null,  
 LOGIN VARCHAR2(50),  
 PASSWORD VARCHAR2(50)  
)

create table RESERVEDTICKETS  
(  
 RESERVEDID NUMBER generated as identity,  
 TICKETID NUMBER  
 constraint TICKETID\_FK  
 references TICKETS  
 on delete cascade,  
 CUSTOMERID NUMBER  
 constraint CUSTOMER\_FK  
 references CUSTOMER,  
 RESERVEDSEATS NUMBER,  
 TOTALPRICE NUMBER,  
 STATUS VARCHAR2(50)  
)  
/  
  
create trigger UPDATETICKETS  
 after insert  
 on RESERVEDTICKETS  
 for each row  
BEGIN  
 update TICKETS  
 set AVALIBLEPLACES=AVALIBLEPLACES - :new.RESERVEDSEATS  
 where TICKETS.TICKETID = :new.TICKETID;  
END;  
/  
  
create trigger DELETETICKETS  
 after delete  
 on RESERVEDTICKETS  
 for each row  
BEGIN  
  
 update TICKETS  
 set AVALIBLEPLACES=AVALIBLEPLACES + :old.RESERVEDSEATS  
 where TICKETS.TICKETID = :old.TICKETID;  
  
END;  
/  
  
create trigger UPDATETICKETSAFTERUPDATE  
 before update  
 on RESERVEDTICKETS  
 for each row  
BEGIN  
 update TICKETS  
 set AVALIBLEPLACES=AVALIBLEPLACES - (:new.RESERVEDSEATS - :old.RESERVEDSEATS)  
 where TICKETS.TICKETID = :new.TICKETID;  
end;  
/

create table TICKETS  
(  
 TICKETID NUMBER generated as identity  
 constraint TICKETID\_PM  
 primary key,  
 AVALIBLEPLACES NUMBER not null,  
 MAXPLACES NUMBER not null,  
 PRICE NUMBER not null,  
 DATEOFDEPARTURE DATE not null,  
 DESTINATIONFROM VARCHAR2(100),  
 DESTINATIONTO VARCHAR2(100),  
 TRAINID NUMBER  
 references TRAINS  
)  
/

create table TRAINS  
(  
 TRAINID NUMBER generated as identity  
 primary key,  
 PHOTO VARCHAR2(200),  
 TRAINNAME VARCHAR2(50) not null  
)  
/

create view TICKETVIEW as  
select TICKETS.TICKETID,  
 TICKETS.AVALIBLEPLACES,  
 TICKETS.MAXPLACES,  
 TICKETS.PRICE,  
 TICKETS.DATEOFDEPARTURE,  
 TICKETS.DESTINATIONFROM,  
 TICKETS.DESTINATIONTO,  
 TRAINS.PHOTO,  
 TRAINS.TRAINNAME  
from Tickets  
 inner join TRAINS on TICKETS.TRAINID = TRAINS.TRAINID  
order by TICKETS.DATEOFDEPARTURE  
/  
  
create view TRAINVIEW as  
select TRAINNAME, TRAINS.TRAINID, PHOTO, count(\*) as NUMBER\_OF\_TICKETS  
from TRAINS  
 inner join TICKETS T on TRAINS.TRAINID = T.TRAINID  
group by TRAINS.TRAINID, TRAINS.PHOTO, TRAINS.TRAINNAME  
/  
  
create view MYACTICETICKETSVIEW as  
select reservedID,  
 ReservedSeats,  
 TotalPrice,  
 Status,  
 T.AVALIBLEPLACES,  
 T.DATEOFDEPARTURE,  
 T.DESTINATIONTO,  
 T.DESTINATIONFROM,  
 T.PRICE,  
 T2.TRAINNAME,  
 T2.PHOTO  
from reservedTickets  
 inner join TICKETS T on reservedTickets.TicketId = T.TICKETID  
 inner join TRAINS T2 on T.TRAINID = T2.TRAINID  
/

insert into Trains  
 (trainName,PHOTO)  
 values  
 ('Shinkansen','https://assets.vogue.com/photos/5b32676b9fb6a34e8cf347f9/master/w\_2560%2Cc\_limit/00-story-image-hello-kitty-bullet-train-japan.jpg');

insert into TRAINS  
 (photo, trainname) VALUES ('https://ocdn.eu/pulscms-transforms/1/fBhk9kpTURBXy82MzI2NWNmNTg0MzhlYjQxZmIzNjljZjI1N2I1ZGQ0MS5qcGeTlQMAzNLNGkDNDsSTBc0DFM0BvJMJpjdiYjdlYgaBoTAB/pendolino.jpg','Pendolino');

insert into Tickets  
(avaliblePlaces, maxPlaces, price, dateOfDeparture, destinationFrom, destinationTo, trainId)  
VALUES  
(200,200,100,'2022-04-10','Tokio','Kawasaki',”podaj id pociągu”);

insert into Tickets  
 (avaliblePlaces, maxPlaces, price, dateOfDeparture, destinationFrom, destinationTo, trainId)  
 VALUES  
 (200,200,100,TO\_DATE('2022-04-1','YYYY-MM-DD'),'Tokio','Sapporo', ,”podaj id pociągu”);

insert into Tickets  
 (avaliblePlaces, maxPlaces, price, dateOfDeparture, destinationFrom, destinationTo, trainId)  
 VALUES  
 (200,200,100,TO\_DATE('2022-06-10','YYYY-MM-DD'),Krakow,'Lublin', ,”podaj id pociągu”);