Sinema Gösterim Otomasyonu  
  
Tablolar  
  
create table koltuk(

koltukno int not null,

koltuknumarası int,

constraint pkkey6 primary key(koltukno)

)  
  
create table salon(

salonno int not null,

koltuksayisi int,

koltukno int,

constraint pkkey5 primary key(salonno),

constraint fkkey1 foreign key(koltukno) references koltuk(koltukno)

)

create table tur(

turno int not null,

turad varchar(15),

constraint pkkey4 primary key(turno)

)

create table bilet(

biletno int not null,

fiyat int,

tarih date,

aciklama varchar(20),

salonno int,

turno int,

constraint pkkey3 primary key(biletno),

constraint fkkey2 foreign key(salonno) references salon(salonno),

constraint fkkey3 foreign key(turno) references tur(turno)

)

create table satici(

satici\_id int not null,

aciklama varchar2(20),

constraint pkkey1 primary key(satici\_id))  
  
  
create table musteri(

musteri\_id int not null,

ad varchar(15),

soyad varchar(15),

yas int,

cinsiyet varchar(6),

satici\_id int,

biletno int,

constraint pkkey2 primary key(musteri\_id),

constraint fkkey4 foreign key(satici\_id) references satici(satici\_id),

constraint fkkey5 foreign key(biletno) references bilet(biletno)

)

İnsert İşlemleri

insert into koltuk values(1,15)  
insert into salon values(01,45,1)  
insert into tur values(1,'ögrenci')

insert into bilet values(1,10,sysdate,'10 tldir',01,1)  
insert into satici values(1,'satici1')

insert into musteri values(1,'fatih','yildiz',21,'erkek',1,1)

insert into koltuk values(2,18)  
insert into salon values(02,15,2)

insert into tur values(2,'yetişkin')  
insert into bilet values(2,15,sysdate,'15 tldir',02,2)  
insert into satici values(2,'satici2')

insert into musteri values(2,'fatma','yilmaz',25,'bayan',2,2)

Join Sorgu İşlemleri  
  
1-)Bilet tür adi ‘yetişkin’ olan ve salon koltuk sayisi 30’dan fazla olan biletlerin toplam fiyatını getiren sql kod  
  
Select sum(fiyat) from bilet

join tur on bilet.turno = tur.turno

join salon on salon.salonno = bilet.biletno

where tur.turad = 'yetişkin' and salon.koltuksayisi>30  
  
2-) Bilet türü adı ‘ögrenci’ olan tüm müşteri kayıtlarını getiren sql kod   
  
Select distinct musteri.\* from musteri

join bilet on musteri.musteri\_id = bilet.biletno

join tur on tur.turno = bilet.turno

where tur.turad = 'ögrenci'  
  
Procedure  
  
1-) Fiyati en yüksek olan biletin fiyatını donduren procedure  
  
create or replace procedure yuksek\_fiyati\_yazdir

as

yuksekfiyat int:=0;

Begin

select max(fiyat) into yuksekfiyat from bilet;

dbms\_output.put\_line('En yüksek fiyatli bilet fiyati :'||yuksekfiyat);

End yuksek\_fiyati\_yazdir;

begin

yuksek\_fiyati\_yazdir(); *--procedure testi*

end;  
  
2-) Bilet alanları isim ve soyisim olarak musteriler tablosına kaydeden procedure

create table musteriler(

isim varchar(15),

soyisim varchar(15)

)

create or replace procedure musteriadsoyad as

cursor bilgi is select ad,soyad from musteri

join bilet on bilet.biletno = musteri.biletno;

begin

for imlecim in bilgi loop

insert into musteriler values(imlecim.ad,imlecim.soyad);

end loop;

end musteriadsoyad;  
select \* from musteriler *--procedure testi*  
Fonksiyon1-) Müşteri adını musteri\_id girilerek döndüren fonksiyon

create or replace function musteriadbul ( idler in int )

return varchar is

bilgi varchar(25);

Begin

Select ad into bilgi from musteri a where a.musteri\_id=idler;

Return (bilgi);

Exception

when no\_data\_found then

Return 'böyle bir müsteri yok';

End;

select musteriadbul(1) from dual *--test function*  
  
2-) Bilet idsi verilen biletin bilgilerini tablo şeklinde getiren function

create or replace type bilgilerimtype as object

(

musterino int,

musteriad varchar(15),

musteriyas int,

fiyat int,

tarih date,

acıklama varchar(25),

koltuknumarasi int

);

create or replace type bilgilerimigetirtype as table of bilgilerimtype

create or replace function bilgilerim(biletid in int)

return bilgilerimigetirtype

is sonuc bilgilerimigetirtype:=new bilgilerimigetirtype();

begin

for i in (select \* from musteri

join bilet on bilet.biletno = musteri.musteri\_id

join salon on salon.salonno = bilet.salonno

join koltuk on koltuk.koltukno = salon.koltukno

where bilet.biletno=biletid)

loop

sonuc.extend;

sonuc(sonuc.count):=new bilgilerimtype(i.musteri\_id,i.ad,i.yas,i.fiyat,i.tarih,i.aciklama,i.koltuknumarasi);

end loop;

return sonuc;

end;

select \* from table(bilgilerim(2)); *-- test function bilgilerim*  
  
  
  
Trigger   
  
1-) Silenen bilet türünü yeni bir tabloya aktarma yapan tetikleme  
  
create table silinenbiletturu(turad varchar(10));

create or replace trigger schema tureekleme

after delete on tur

begin

insert into silinenbiletturu values(:old.turad)

end;

delete from tur where turno=2; *--trigger testi*

select \* from silinenbiletturu  
  
2-) Bilet fiyatı değiştirildiğinde aradaki farkı ekrana yansıtan tetikleme

create or replace trigger schema bilet\_fiyati\_farkini\_goruntule

before update on bilet

for each row

when (new.biletno>0)

declare

fark int:=0;

begin

fark := :new.fiyat - :old.fiyat;

dbms\_output.put\_line('eski bilet fiyati :' || :old.fiyat);

dbms\_output.put\_line('yeni bilet fiyati :' || :new.fiyat);

dbms\_output.put\_line('bilet fiyat farkı :' || fark);

end;

update bilet set fiyat = fiyat+5 where biletno=2; *--trigger testi*

Exception  
  
1-) Turno’su girilen bilginin kayıtta olmamasında oluşan hata bloğu  
  
Declare

turid int:=&degeri\_al;

bilgi tur%rowtype;

begin

select \* into bilgi from tur where turno=turid;

dbms\_output.put\_line(bilgi.turno||'--'||bilgi.turad);

Exception

when No\_data\_found then

raise\_application\_error(-20111,'..herhangi bir tür kaydı bulunamadi..!');

end;