

```

-- Ticket_Sales_System (Cinema)

-- 17010011005
-- Yunus Emre Alpu

-- Drop tables
DROP TABLE tblMovie CASCADE CONSTRAINTS;
DROP TABLE tblGenre CASCADE CONSTRAINTS;
DROP TABLE Movie_Genre CASCADE CONSTRAINTS;
DROP TABLE tblDirector CASCADE CONSTRAINTS;
DROP TABLE Movie_Director CASCADE CONSTRAINTS;
DROP TABLE tblCast CASCADE CONSTRAINTS;
DROP TABLE Movie_Cast CASCADE CONSTRAINTS;
DROP TABLE tblSaloon CASCADE CONSTRAINTS;
DROP TABLE tblSession CASCADE CONSTRAINTS;
DROP TABLE Saloon_Session CASCADE CONSTRAINTS;
DROP TABLE tblCouch CASCADE CONSTRAINTS;
DROP TABLE tblUser CASCADE CONSTRAINTS;
DROP TABLE tblTicket CASCADE CONSTRAINTS;

-- Alter
ALTER SESSION SET nls_timestamp_format = 'YYYY/MM/DD HH24:MI:SS'; -- Preferences
-> Database -> NLS -> DATE_FORMAT

-- Movie table
CREATE TABLE tblMovie (
    movie_id VARCHAR(10),
    movie_title VARCHAR(50) NOT NULL,
    movie_story VARCHAR(255) NOT NULL,
    movie_release_date DATE NOT NULL,
    movie_runtime NUMBER(5) NOT NULL,
    movie_type VARCHAR(10),

    CHECK(movie_runtime > 0),
    CONSTRAINT PK_Movie PRIMARY KEY(movie_id)
);

-- Genre table
CREATE TABLE tblGenre (
    genre_id VARCHAR(10),
    genre_name VARCHAR(50) NOT NULL,

    CONSTRAINT PK_Genre PRIMARY KEY(genre_id)
);

-- Movie_Genre table
CREATE TABLE Movie_Genre (
    mg_movie_id VARCHAR(10),
    mg_genre_id VARCHAR(10),

    CONSTRAINT FK_Movie_Genre_movie_id FOREIGN KEY(mg_movie_id) REFERENCES
tblMovie(movie_id) ON DELETE CASCADE, -- ON UPDATE CASCADE
    CONSTRAINT FK_Movie_Genre_genre_id FOREIGN KEY(mg_genre_id) REFERENCES
tblGenre(genre_id) ON DELETE CASCADE -- ON UPDATE CASCADE
);

-- Director table
CREATE TABLE tblDirector (
    director_id VARCHAR(10),
    director_first_name VARCHAR(50) NOT NULL,
    director_last_name VARCHAR(50) NOT NULL,

    CONSTRAINT PK_Director PRIMARY KEY(director_id)
);

```

```

-- Movie_Director table
CREATE TABLE Movie_Director (
    md_movie_id VARCHAR(10),
    md_director_id VARCHAR(10),

    CONSTRAINT FK_Movie_Director_movie_id FOREIGN KEY(md_movie_id) REFERENCES
tblMovie(movie_id) ON DELETE CASCADE, -- ON UPDATE CASCADE
    CONSTRAINT FK_Movie_Director_director_id FOREIGN KEY(md_director_id)
REFERENCES tblDirector(director_id) ON DELETE CASCADE -- ON UPDATE CASCADE
);

-- Cast table
CREATE TABLE tblCast (
    cast_id VARCHAR(10),
    cast_first_name VARCHAR(50) NOT NULL,
    cast_last_name VARCHAR(50) NOT NULL,

    CONSTRAINT PK_Cast PRIMARY KEY(cast_id)
);

-- Movie_Cast table
CREATE TABLE Movie_Cast (
    mc_movie_id VARCHAR(10),
    mc_cast_id VARCHAR(10),
    mc_cast_character VARCHAR(50) NOT NULL,

    CONSTRAINT FK_Movie_Cast_movie_id FOREIGN KEY(mc_movie_id) REFERENCES
tblMovie(movie_id) ON DELETE CASCADE, -- ON UPDATE CASCADE
    CONSTRAINT FK_Movie_Cast_cast_id FOREIGN KEY(mc_cast_id) REFERENCES
tblCast(cast_id) ON DELETE CASCADE -- ON UPDATE CASCADE
);

-- Saloon table
CREATE TABLE tblSaloon (
    saloon_id VARCHAR(10),

    CONSTRAINT PK_Saloon PRIMARY KEY(saloon_id)
);

-- Session table
CREATE TABLE tblSession (
    session_id VARCHAR(10),
    session_time DATE NOT NULL,
    session_movie_id VARCHAR(10),

    CONSTRAINT PK_Session PRIMARY KEY(session_id),
    CONSTRAINT FK_Session_Movie_movie_id FOREIGN KEY(session_movie_id)
REFERENCES tblMovie(movie_id) ON DELETE CASCADE -- ON UPDATE CASCADE
);

-- Saloon_Session table
CREATE TABLE Saloon_Session (
    ss_id VARCHAR(10),
    ss_saloon_id VARCHAR(10),
    ss_session_id VARCHAR(10),

    CONSTRAINT PK_Saloon_Session PRIMARY KEY(ss_id),
    CONSTRAINT FK_Saloon_Session_saloon_id FOREIGN KEY(ss_saloon_id)
REFERENCES tblSaloon(saloon_id) ON DELETE CASCADE, -- ON UPDATE CASCADE
    CONSTRAINT FK_Saloon_Session_session_id FOREIGN KEY(ss_session_id)
REFERENCES tblSession(session_id) ON DELETE CASCADE, -- ON UPDATE CASCADE
    CONSTRAINT UC_Saloon_Session UNIQUE(ss_saloon_id, ss_session_id)
);

```

```

-- Couch table
CREATE TABLE tblCouch(
    couch_id VARCHAR(10),

    CONSTRAINT PK_Couch PRIMARY KEY(couch_id)
);

-- User table
CREATE TABLE tblUser (
    user_id VARCHAR(10),
    user_first_name VARCHAR(50) NOT NULL,
    user_last_name VARCHAR(50) NOT NULL,
    user_age DATE NOT NULL,

    CONSTRAINT PK_User PRIMARY KEY(user_id)
);

-- Ticket table
CREATE TABLE tblTicket (
    ticket_user_id VARCHAR(10),
    ticket_saloon_session_id VARCHAR(10),
    ticket_couch_id VARCHAR(10),
    ticket_price NUMBER(5,2),

    CHECK(ticket_price > 0),
    CONSTRAINT FK_Ticket_user_id FOREIGN KEY(ticket_user_id) REFERENCES
tblUser(user_id) ON DELETE CASCADE, -- ON UPDATE CASCADE
    CONSTRAINT FK_Ticket_session_id FOREIGN KEY(ticket_saloon_session_id)
REFERENCES Saloon_Session(ss_id) ON DELETE CASCADE, -- ON UPDATE CASCADE
    CONSTRAINT FK_Ticket_couch_id FOREIGN KEY(ticket_couch_id) REFERENCES
tblCouch(couch_id) ON DELETE CASCADE, -- ON UPDATE CASCADE,
    CONSTRAINT UC_Ticket UNIQUE(ticket_saloon_session_id, ticket_couch_id)
);

-- Insert data (tblMovie)
INSERT INTO tblMovie(movie_id, movie_title, movie_story, movie_release_date,
movie_runtime, movie_type)
VALUES('tt0898367', 'The Road', 'In a dangerous post-apocalyptic world, an
ailing father defends his son as they slowly travel to the sea.', TO_DATE('
2009/11/25', 'YYYY/MM/DD'), 111, 'Full');

INSERT INTO tblMovie(movie_id, movie_title, movie_story, movie_release_date,
movie_runtime, movie_type)
VALUES('tt3553976', 'Captain Fantastic', 'In the forests of the Pacific
Northwest, a father devoted to raising his six kids with a rigorous physical and
intellectual education is forced to leave his paradise and enter the world,
challenging his idea of what it means to be a parent.', TO_DATE(' 2016/11/11',
'YYYY/MM/DD'), 118, 'Full');

INSERT INTO tblMovie(movie_id, movie_title, movie_story, movie_release_date,
movie_runtime, movie_type)
VALUES('tt3472226', 'Kung Fury', 'In 1985, Kung Fury, the toughest martial
artist cop in Miami, goes back in time to kill the worst criminal of all time -
Kung Führer, a.k.a. Adolf Hitler.', TO_DATE(' 2015/05/28', 'YYYY/MM/DD'), 32,
'Full');

INSERT INTO tblMovie(movie_id, movie_title, movie_story, movie_release_date,
movie_runtime, movie_type)
VALUES('tt0112691', 'A Close Shave', 'Wallaces whirlwind romance with the owner
of the local wool shop puts his head in a spin; Gromit is framed for sheep-
rustling in a fiendish criminal plot.', TO_DATE(' 1998/07/26', 'YYYY/MM/DD'),
30, 'Full');

```

```

INSERT INTO tblMovie(movie_id, movie_title, movie_story, movie_release_date,
movie_runtime, movie_type)
VALUES('tt4154796', 'Avengers Endgame', 'After the devastating events of
Avengers: Sonsuzluk Savaşı (2018), the universe is in ruins. With the help of
remaining allies, the Avengers assemble once more in order to reverse Thanos
actions and restore balance to the universe.', TO_DATE(' 2019/04/25',
'YYYY/MM/DD'), 181, 'Short');

```

-- Insert data (tblGenre)

```

INSERT INTO tblGenre(genre_id, genre_name) VALUES('Act', 'Action');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Adv', 'Adventure');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Com', 'Comedy');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Cri', 'Crime');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Dra', 'Drama');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Epi', 'Epicl');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Hor', 'Horror');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Mus', 'Musicals');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Sci', 'Science Fiction');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('War', 'War');
INSERT INTO tblGenre(genre_id, genre_name) VALUES('Wes', 'Westerns');

```

-- Insert data (Movie-Genre)

```

INSERT INTO Movie_Genre(mg_movie_id, mg_genre_id) VALUES('tt0898367', 'Dra');
INSERT INTO Movie_Genre(mg_movie_id, mg_genre_id) VALUES('tt3553976', 'Dra');
INSERT INTO Movie_Genre(mg_movie_id, mg_genre_id) VALUES('tt3553976', 'Com');
INSERT INTO Movie_Genre(mg_movie_id, mg_genre_id) VALUES('tt4154796', 'Act');
INSERT INTO Movie_Genre(mg_movie_id, mg_genre_id) VALUES('tt4154796', 'Adv');
INSERT INTO Movie_Genre(mg_movie_id, mg_genre_id) VALUES('tt4154796', 'Dra');

```

-- Insert data (tblDirector)

```

INSERT INTO tblDirector(director_id, director_first_name, director_last_name)
VALUES('JhnHlct', 'John', 'Hillcoat');
INSERT INTO tblDirector(director_id, director_first_name, director_last_name)
VALUES('MtRs', 'Matt', 'Ross');

```

-- Insert data (Movie-Director)

```

INSERT INTO Movie_Director(md_movie_id, md_director_id) VALUES('tt0898367',
'JhnHlct');
INSERT INTO Movie_Director(md_movie_id, md_director_id) VALUES('tt3553976',
'MtRs');

```

-- Insert data (tblCast)

```

INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name) VALUES('VgMrtsn',
'Viggo', 'Mortensen');
INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name) VALUES('KdSmMPH',
'Kodi', 'Smit-McPhee');
INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name) VALUES('ChrzThr',
'Charlize', 'Theron');
INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name) VALUES('GrgMcKy',
'George', 'MacKay');
INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name) VALUES('SmnthSl',
'Samantha', 'Isler');
INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name) VALUES('AnlsBs',
'Annalise', 'Basso');
INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name)
VALUES('NchlsHmltn', 'Nicholas', 'Hamilton');
INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name) VALUES('ShrCrks',
'Shree', 'Crooks');
INSERT INTO tblCast(cast_id, cast_first_name, cast_last_name)
VALUES('ChrlShtwl', 'Charlie', 'Shotwell');

```

-- Insert data (Movie-Cast)

```

INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt0898367', 'VgMrtsn', 'Man');

```

```

INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt0898367', 'KdSmMPH', 'Boy');
INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt0898367', 'ChrzThr', 'Woman');
INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt3553976', 'VgMrtsn', 'Ben');
INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt3553976', 'GrgMcKy', 'Bodevan');
INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt3553976', 'SmnthSl', 'Kielyr');
INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt3553976', 'AnlsBs', 'Vespyr');
INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt3553976', 'NchlsHmltn', 'Rellian');
INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt3553976', 'ShrCrks', 'Zaja');
INSERT INTO Movie_Cast(mc_movie_id, mc_cast_id, mc_cast_character)
VALUES('tt3553976', 'ChrlShtwl', 'Nai');

-- Insert data (tblSaloon)
Insert INTO tblSaloon(saloon_id) VALUES('Saloon 1');
Insert INTO tblSaloon(saloon_id) VALUES('Saloon 2');
Insert INTO tblSaloon(saloon_id) VALUES('Saloon 3');

-- Insert data (tblSession)
INSERT INTO tblSession(session_id, session_time, session_movie_id )
VALUES('Mor9Pck1', TO_DATE( '2020/04/27 09:00:00', 'YYYY/MM/DD HH24:MI:SS') ,
'tt0898367');
INSERT INTO tblSession(session_id, session_time, session_movie_id )
VALUES('Mor9Pck2', TO_DATE( '2020/04/27 09:00:00', 'YYYY/MM/DD HH24:MI:SS') ,
'tt3553976');
INSERT INTO tblSession(session_id, session_time, session_movie_id )
VALUES('Ev11Pck2', TO_DATE( '2020/04/30 23:00:00', 'YYYY/MM/DD HH24:MI:SS') ,
'tt3553976');
INSERT INTO tblSession(session_id, session_time, session_movie_id )
VALUES('Ev11Pck1', TO_DATE( '2020/04/30 23:00:00', 'YYYY/MM/DD HH24:MI:SS') ,
'tt4154796');

-- Insert data (Saloon_Session)
INSERT INTO Saloon_Session(ss_id, ss_saloon_id, ss_session_id) VALUES('Pck1',
'Saloon 1', 'Mor9Pck1');
INSERT INTO Saloon_Session(ss_id, ss_saloon_id, ss_session_id) VALUES('Pck2',
'Saloon 2', 'Mor9Pck2');
INSERT INTO Saloon_Session(ss_id, ss_saloon_id, ss_session_id) VALUES('Pck3',
'Saloon 3', 'Mor9Pck2');
INSERT INTO Saloon_Session(ss_id, ss_saloon_id, ss_session_id) VALUES('Pck4',
'Saloon 1', 'Ev11Pck2');
INSERT INTO Saloon_Session(ss_id, ss_saloon_id, ss_session_id) VALUES('Pck5',
'Saloon 2', 'Ev11Pck1');
-- INSERT INTO Saloon_Session(ss_id, ss_saloon_id, ss_session_id) VALUES('Pck4',
'Saloon 3', 'Mor9Pck2'); error -> same saloon and same session

-- Insert data (tblCouch)
INSERT INTO tblCouch(couch_id) VALUES('A1');
INSERT INTO tblCouch(couch_id) VALUES('A2');
INSERT INTO tblCouch(couch_id) VALUES('A3');
INSERT INTO tblCouch(couch_id) VALUES('A4');
INSERT INTO tblCouch(couch_id) VALUES('A5');
INSERT INTO tblCouch(couch_id) VALUES('A6');
INSERT INTO tblCouch(couch_id) VALUES('A7');
INSERT INTO tblCouch(couch_id) VALUES('A8');
INSERT INTO tblCouch(couch_id) VALUES('A9');
INSERT INTO tblCouch(couch_id) VALUES('A10');

```

```

-- Insert data (tblUser)
INSERT INTO tblUser(user_id, user_first_name, user_last_name, user_age)
VALUES('User1YEA', 'Yunus Emre', 'Alpu', TO_DATE('1999/04/18', 'YYYY/MM/DD'));
INSERT INTO tblUser(user_id, user_first_name, user_last_name, user_age)
VALUES('User2YA', 'Yusuf', 'Alpu', TO_DATE('2001/11/25', 'YYYY/MM/DD'));
INSERT INTO tblUser(user_id, user_first_name, user_last_name, user_age)
VALUES('User3HA', 'Huseyin', 'Ari', TO_DATE('1999/06/25', 'YYYY/MM/DD'));
INSERT INTO tblUser(user_id, user_first_name, user_last_name, user_age)
VALUES('User4MA', 'Mesut', 'Ates', TO_DATE('1999/01/07', 'YYYY/MM/DD'));
INSERT INTO tblUser(user_id, user_first_name, user_last_name, user_age)
VALUES('User5AA1', 'Ali', 'Al', TO_DATE('2004/01/07', 'YYYY/MM/DD'));

-- Insert data (tblTicket)
INSERT INTO tblTicket(ticket_user_id, ticket_saloon_session_id, ticket_couch_id,
ticket_price)
VALUES('User1YEA', 'Pck1', 'A1', 20);
INSERT INTO tblTicket(ticket_user_id, ticket_saloon_session_id, ticket_couch_id,
ticket_price)
VALUES('User2YA', 'Pck1', 'A2', 20);
INSERT INTO tblTicket(ticket_user_id, ticket_saloon_session_id, ticket_couch_id,
ticket_price)
VALUES('User3HA', 'Pck2', 'A1', 20);
INSERT INTO tblTicket(ticket_user_id, ticket_saloon_session_id, ticket_couch_id,
ticket_price)
VALUES('User4MA', 'Pck2', 'A2', 20);
INSERT INTO tblTicket(ticket_user_id, ticket_saloon_session_id, ticket_couch_id,
ticket_price)
VALUES('User5AA1', 'Pck2', 'A3', 20);
INSERT INTO tblTicket(ticket_user_id, ticket_saloon_session_id, ticket_couch_id,
ticket_price)
VALUES('User5AA1', 'Pck5', 'A3', 20);
INSERT INTO tblTicket(ticket_user_id, ticket_saloon_session_id, ticket_couch_id,
ticket_price)
VALUES('User1YEA', 'Pck5', 'A1', 20);
/*
  INSERT INTO tblTicket(ticket_user_id, ticket_saloon_session_id,
ticket_couch_id, ticket_price)
VALUES('User4MA', 'Pck2', 'A2', 20); error x
*/

-- Select data
SELECT * FROM tblMovie;
SELECT * FROM tblGenre;
SELECT * FROM Movie_Genre;
SELECT * FROM tblDirector;
SELECT * FROM Movie_Director;
SELECT * FROM tblCast;
SELECT * FROM Movie_Cast;
SELECT * FROM tblSaloon;
SELECT * FROM tblSession;
SELECT * FROM Saloon_Session;
SELECT * FROM tblCouch;
SELECT * FROM tblUser;
SELECT * FROM tblTicket;

-- Join
-- 18 yasindan kucuk kullanicilari ve ucretlerini yazan query
SELECT userr.user_id, userr.user_age, ticket.ticket_price
FROM tblUser userr INNER JOIN tblTicket ticket
ON userr.user_id = ticket.ticket_user_id WHERE (TRUNC(SYSDATE-
userr.user_age)/365.25) < 18;

-- oyuncularin hangi filmlerde ve hangi karakteri canlandirdigini yazan query
SELECT movies.movie_title, (SELECT plyr.cast_first_name FROM tblCast plyr WHERE

```

```

cast_id = castt.mc_cast_id), (SELECT plyr.cast_last_name FROM tblCast plyr WHERE
cast_id = castt.mc_cast_id) , castt.mc_cast_character
FROM tblMovie movies INNER JOIN Movie_Cast castt ON movies.movie_id =
castt.mc_movie_id;

-- PL/SQL
-- Drop
DROP PROCEDURE session_timeout;
DROP FUNCTION total_price;
DROP TYPE rsvrrcdTicket;
DROP TYPE rcrd_row;
DROP FUNCTION ticketFunc;
DROP TRIGGER user_price;
DROP TRIGGER movie_price;

-- procedures
-- zaman asimina ugramis seanslari silen procedure
CREATE OR REPLACE PROCEDURE session_timeout
AS
BEGIN
    DELETE FROM tblSession WHERE (TRUNC(SYSDATE - session_time)) > 0; --
YYYY/MM/DD HH24:MI:SS (NLS Settings)
EXCEPTION
    WHEN no_data_found THEN
        DBMS_OUTPUT.put_line(' - Data not found. ');
    WHEN others THEN
        DBMS_OUTPUT.put_line(' - Error ');
END;
/

-- filmin suresine gore filmin tipini yazan procedure
-- SET SERVEROUTPUT ON; -- SQL Developer
DECLARE
    c_movie_id tblMovie.movie_id%TYPE;
    c_movie_title tblMovie.movie_title%TYPE;
    c_movie_runtime tblMovie.movie_runtime%TYPE;
    c_movie_type tblMovie.movie_type%TYPE;
    CURSOR movieList IS
        SELECT movie_id, movie_title, movie_runtime, movie_type FROM tblMovie;
BEGIN
    OPEN movieList;
    LOOP
        FETCH movieList INTO c_movie_id, c_movie_title, c_movie_runtime,
c_movie_type;
        EXIT WHEN movieList%NOTFOUND;
        IF ((c_movie_runtime < 40) AND (c_movie_type != 'Short')) THEN
            UPDATE tblMovie SET movie_type = 'Short' WHERE movie_id = c_movie_id;

        ELSIF ((c_movie_runtime > 40) AND (c_movie_type != 'Full')) THEN
            UPDATE tblMovie SET movie_type = 'Full' WHERE movie_id = c_movie_id;

        END IF;
        DBMS_OUTPUT.put_line(c_movie_id || ' - ' || c_movie_title || ' - ' ||
c_movie_runtime || ' - ' || c_movie_type);
    END LOOP;
    CLOSE movieList;
END;

-- functions
-- toplam kazanci hesaplayan fonksiyon
CREATE FUNCTION total_price
RETURN NUMBER AS
tot_price NUMBER := 0;
BEGIN

```

```

SELECT SUM(ticket_price) INTO tot_price FROM tblTicket;
RETURN tot_price;
EXCEPTION
    WHEN no_data_found THEN
        DBMS_OUTPUT.put_line(' - Data not found.');
```

-- koltuk numarasina gore bilet bilgilerini ve kullanıcı bilgilerini getiren function

--Create Database Type as Object

```

CREATE OR REPLACE TYPE rsvrrcdTicket AS OBJECT(
    tckt_userid    VARCHAR(10),
    tckt_sln_ss    VARCHAR(10),
    tckt_cchid     VARCHAR(10),
    tckt_prc       NUMBER(5,2),
    usr_fname      VARCHAR(50),
    usr_lname      VARCHAR(50)
);
/

--Create table of the Type
CREATE OR REPLACE TYPE rcrd_row AS TABLE OF rsvrrcdTicket;
/

-- Koltuk numaralarına gore tablo donduren function
CREATE OR REPLACE FUNCTION ticketFunc(cch_id IN VARCHAR) RETURN rcrd_row
IS
    reslt rcrd_row := NEW rcrd_row();

BEGIN
    FOR rw IN (SELECT tck.ticket_user_id, usr.user_first_name, usr.user_last_name,
tck.ticket_saloonsession_id, tck.ticket_couch_id, tck.ticket_price FROM
tblTicket tck INNER JOIN tblUser usr ON tck.ticket_user_id = usr.user_id WHERE
ticket_couch_id = cch_id)
    LOOP
        reslt.extend;
        reslt(reslt.count) := new rsvrrcdTicket(
            tckt_userid => rw.ticket_user_id,  usr_fname => rw.user_first_name,  usr_lname
=> rw.user_last_name, tckt_sln_ss =>rw.ticket_saloonsession_id,  tckt_cchid =>
rw.ticket_couch_id, tckt_prc => rw.ticket_price
        );
    END LOOP;
    RETURN reslt;

END;
/

-- Triggers
-- 18 yasından küçüklere %20 indirim
CREATE OR REPLACE TRIGGER user_price
BEFORE INSERT OR UPDATE ON tblTicket
REFERENCING NEW AS n OLD AS o
FOR EACH ROW
DECLARE
    birthday DATE;
    u_age NUMBER;
    t_price NUMBER(5,2);
BEGIN
    SELECT user_age INTO birthday FROM tblUser WHERE user_id = :n.ticket_user_id;
    u_age := (TRUNC((SYSDATE - birthday)/365.25));
```



```

    IF (u_age < 18) THEN
        t_price := :n.ticket_price - :n.ticket_price * 0.2;
        :n.ticket_price := t_price;
        DBMS_OUTPUT.put_line(' - Age < 18 : Discount %20');
    END IF;
END;

-- 2 saatten uzun filmlerin ucretine %50 zam
CREATE OR REPLACE TRIGGER movie_price
BEFORE INSERT OR UPDATE ON tblTicket
REFERENCING NEW AS n OLD AS o
FOR EACH ROW
DECLARE
    ses_id VARCHAR(10); -- session id
    mov_id VARCHAR(10); -- movie id
    mov_runtime NUMBER; -- movie runtime
    t_price NUMBER(5,2);
BEGIN
    SELECT ss_session_id INTO ses_id FROM Saloon_Session WHERE ss_id =
:n.ticket_saloon_session_id;
    SELECT session_movie_id INTO mov_id FROM tblSession WHERE session_id =
ses_id;
    SELECT movie_runtime INTO mov_runtime FROM tblMovie WHERE movie_id = mov_id;

    IF (mov_runtime > 120) THEN
        t_price := :n.ticket_price + :n.ticket_price * 0.5;
        :n.ticket_price := t_price;
        DBMS_OUTPUT.put_line(' - Movie Time ');
    END IF;
END;

-- Run
-- EXECUTE session_timeout -- SQL DEVELOPER
BEGIN
    session_timeout;
END;
SELECT total_price FROM dual;
SELECT * FROM TABLE(ticketFunc('A3'));

```