Abdul-Rahman Abdel-Fattah 210046015
Data Warehousing - Phase 3
DSEG 733: Advances Data Management Systems
Dr. Aiman Erbad

Phase 1: Initial Schema

The first phase of this three phase assignment involved coming up with the star schema for the given ER design that is shown in Figure 1. By creating the schema for the ER design, we are able to get an idea about the fact table, the multiple dimension tables, the attributes in each of the tables, and the links between all the tables. Figure 2 shows the initial schema that I came up to use for this assignment.

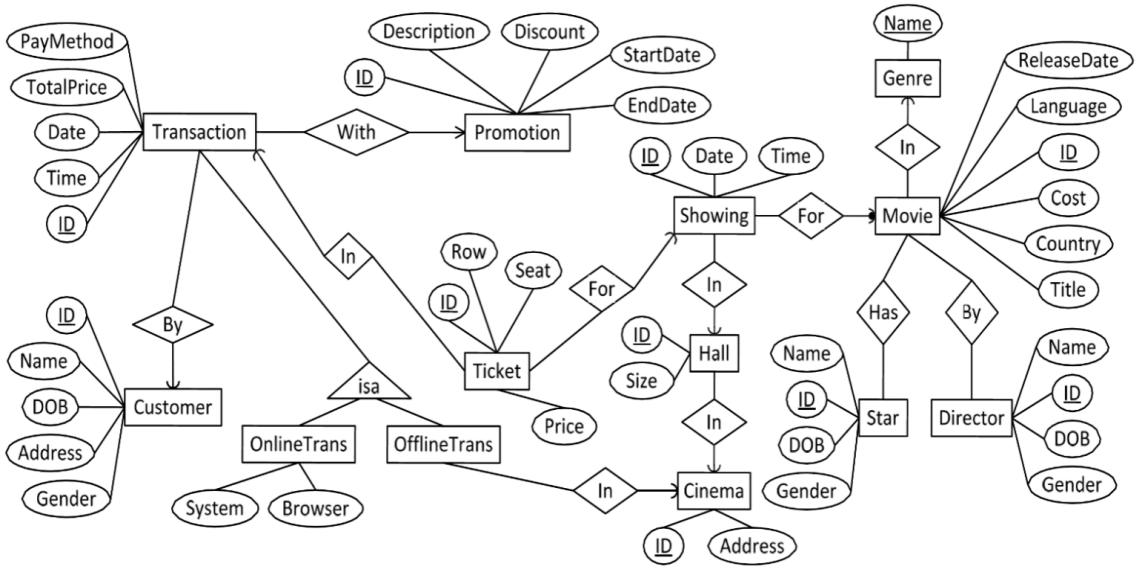


Figure 1. The ER Design for the assignment

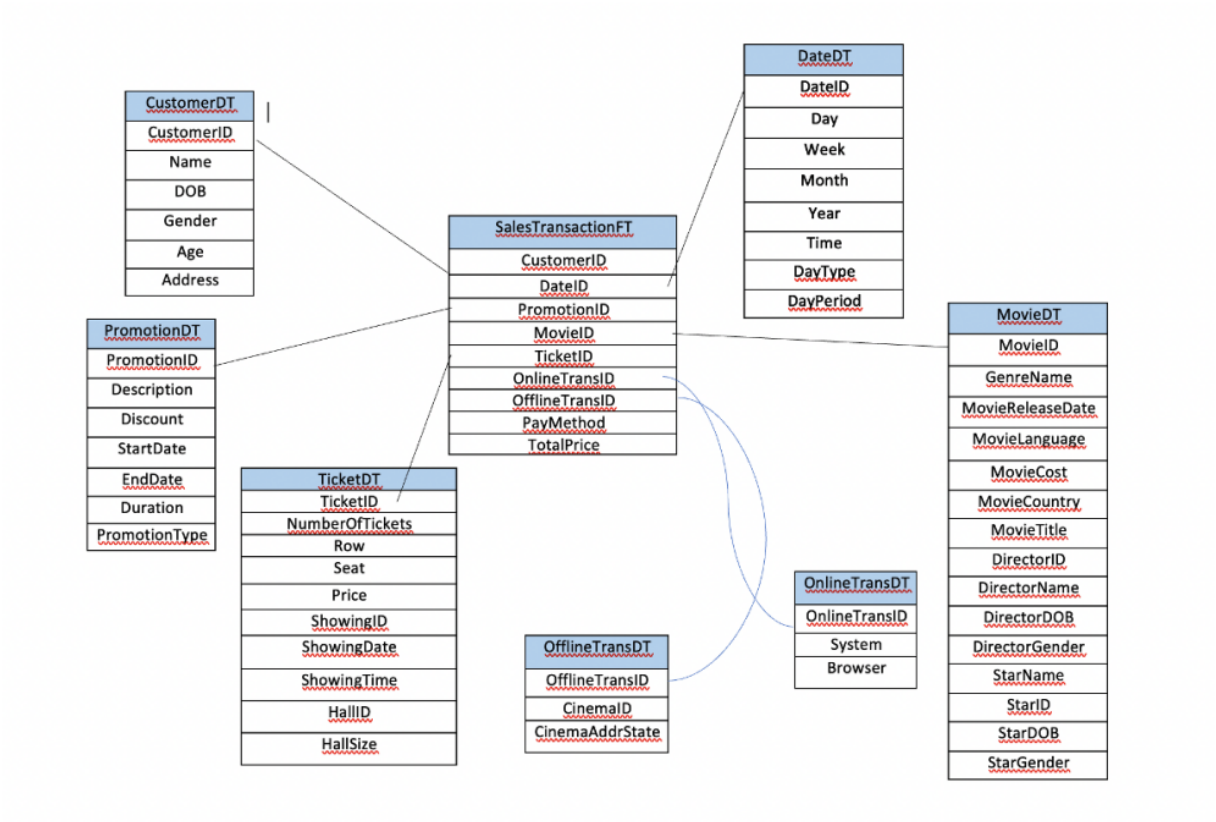


Figure 2: Initial Schema

Phase 2: Revised Schema + SQL Queries

A couple of changes were made to Figure 2 to result in Figure 3. Instead of having separate dimension tables for Online Transaction and Offline Transaction, I felt that it would be better to represent this into a single transaction-type dimension table. The ‘Transaction_Type’ column deals with whether it is an online or offline transaction. Since a transaction can be one or the other, there will be NULL inputs in this dimension table. For example, if the customer buys their ticket at the cinema, the browser attribute will be NULL.

I also decided to remove the MovieDT. One of my reasons for doing so is because it contained many attributes that were irrelevant to what was needed from us later on in this phase - this helps any potential data issues . Some of the attributes in this dimension table in phase 1 were also moved to other dimension tables or the fact table for this current phase. A couple of attributes were removed from the Customer dimension table - leaving only the important attributes needed.

Another key difference between the two designs is the distinction of ‘Time’ and ‘ShowingTime’. ShowingTime holds the time of the showing that the ticket was bought for, whereas ‘Time’ stores the time in which the ticket was bought.

By doing the above changes, I was able to reduce the overall number of tables from 8 to 6. Some might argue that this design could be represented as a snowflake schema (like the sample solution posted by the course TA), I ended up going with the star schema given its simplicity when writing queries.

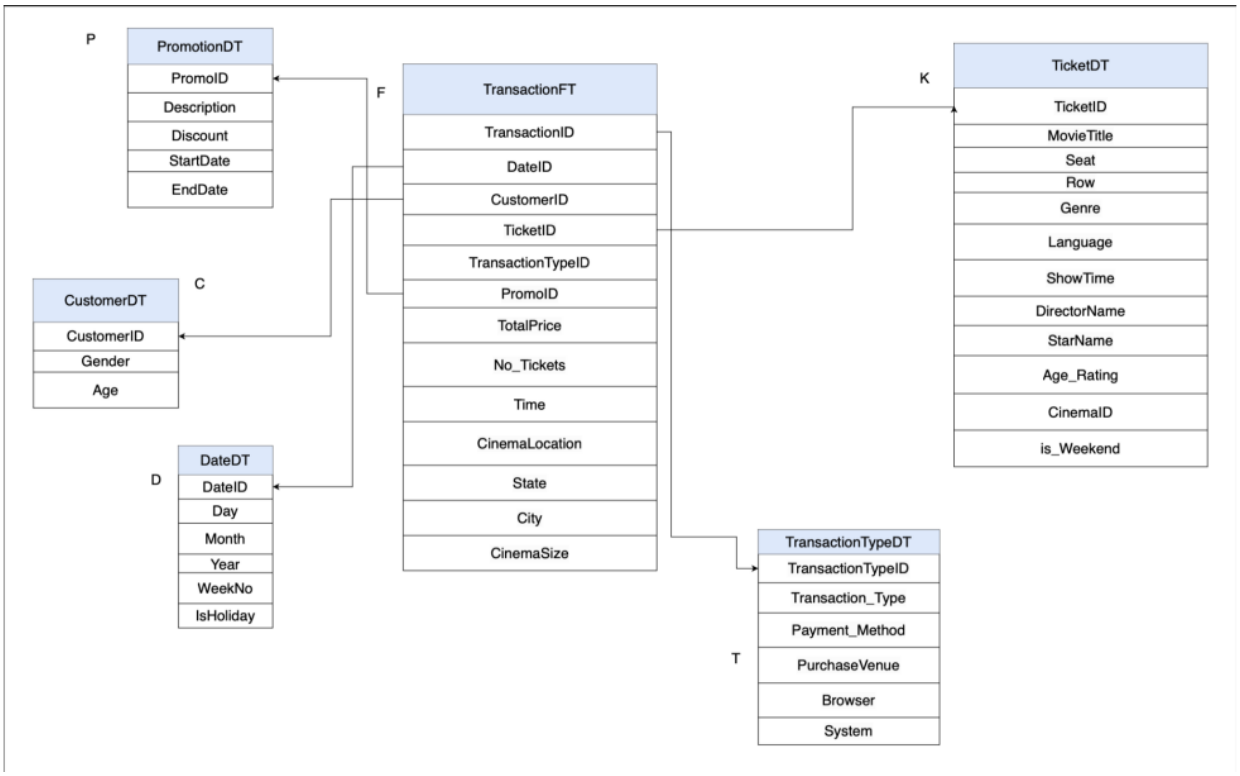


Figure 3. Revised Schema

Phase 2 also started to dive into the queries for the creation of the tables and generating pivot tables - with the purpose of gaining analysis about the information we have at our hands.

```
CREATE TABLE [dbo].[DateDT] (
        DateID   INT NOT NULL,
        Date_Day          INT NOT NULL,
        Date_WeekNo INT NOT NULL,
        Date_Month  INT NOT NULL,
        Date_Year   INT NOT NULL,
        is_holiday  INT NOT NULL,
        CONSTRAINT PK_DateDT PRIMARY KEY CLUSTERED (Date_id ASC)
);

CREATE TABLE [dbo].[CustomerDT] (
        CustomerID INT NOT NULL,
        Age       INT NOT NULL,
        Gender   NVARCHAR (50) NOT NULL,
        CONSTRAINT PK_CustomerDT PRIMARY KEY CLUSTERED (Customer_id ASC)




CREATE TABLE [dbo].[TicketDT] (
        TicketID INT NOT NULL,
        MovieTitle   NVARCHAR (50) NOT NULL,
        Seat      INT NOT NULL,
        Row       NVARCHAR (50) NOT NULL,
        Genre     NVARCHAR (50) NOT NULL,
        Language          NVARCHAR (50) NOT NULL,
        ShowTime          NVARCHAR (50) NOT NULL,
        DirectorName NVARCHAR (50) NOT NULL,
        StarName          NVARCHAR (50) NOT NULL,
        AgeRating         NVARCHAR (50) NULL,
        CinemaID           INT NOT NULL,
        is_weekend    INT NOT NULL,
        CONSTRAINT PK_TicketDT PRIMARY KEY CLUSTERED (Ticket_id ASC)
);

CREATE TABLE [dbo].[TransactionTypeDT] (
        TransactionTypeID  INT NOT NULL,
        Transaction_Type   NVARCHAR (50) NOT NULL,
        Payment_Method NVARCHAR (50) NOT NULL,
        System            NVARCHAR (50) NULL,
        Browser           NVARCHAR (50) NULL,
        Purchase_Venue    NVARCHAR (50) NULL,
        CONSTRAINT PK_TransactionTypeDT PRIMARY KEY CLUSTERED (TransactionType_id ASC)
);
```

```
CREATE TABLE [dbo].[PromotionDT] (
        PromotionID     INT     NOT NULL,
        Description NVARCHAR (50) NOT NULL,
        Discount NVARCHAR (50) NOT NULL,
        Start_Date      DATE            NOT NULL,
        End_Date        DATE            NOT NULL,
        CONSTRAINT PK_PromotionDT PRIMARY KEY CLUSTERED (Promotion_id ASC)
);
```

```
CREATE TABLE [dbo].[TransactionFT] (
        TransactionID   INT NOT NULL
        DateID          INT NOT NULL,
        CustomerID      INT NOT NULL,
        TicketID        INT NOT NULL,
        TransactionTypeID INT NOT NULL,
        PromotionID     INT NOT NULL,
        TotalPrice      INT NOT NULL,
        No_tickets      INT NOT NULL,
        Time            TIME (7) NOT NULL,
        CinemaLocation  NVARCHAR (50) NOT NULL,
        States          NVARCHAR (50) NOT NULL,
        City        NVARCHAR (50) NOT NULL,
        CinemaSize      NVARCHAR (50) NOT NULL,
        CONSTRAINT [FK_TransactionFT_CustomerDT] FOREIGN KEY ([CustomerID]) REFERENCES
[dbo].[CustomerDT] ([Customer_id]),
        CONSTRAINT [FK_TransactionFT_DateDT] FOREIGN KEY ([DateID]) REFERENCES [dbo].[DateDT]
([Date_id]),
```

```
        CONSTRAINT [FK_TransactionFT_PromotionDT] FOREIGN KEY ([PromotionID]) REFERENCES
 [dbo].[PromotionDT] ([Promotion_id]),
        CONSTRAINT [FK_TransactionFT_TransactionTypeDT] FOREIGN KEY ([TransactionTypeID])
 REFERENCES [dbo].[TransactionTypeDT] ([TransactionType_id])
 );
```

Question 1: Produce a pivot table for total sales, with two dimensions, namely, (i) the years of the transactions, and (ii) the genders of the customers.	SELECT Gender, Date_year, SUM(TotalPrice) FROM C, F, D WHERE T.DateID=D.DateID AND F.CustomerID= C.CustomerID Group by Date_year, Gender;
Question 2: Produce a pivot table for total sales, with two dimensions, namely, (i) the months of the transactions with ROLLUP to years, and (ii) whether or not the transactions are made online or offline.	SELECT Date_Month, Date_Year, Transaction_Type, SUM(TotalPrice) AS 'Total Sales' FROM F,T, D WHERE F.DateID=D.DateID AND F.TransactionTypeID=T.TransactionTypeID GROUP BY Transaction_Type, Rollup(Date_Year, Date_Month)

	`` ORDER BY Date_Month, Date_Year ; ```
Question 3: Produce a pivot table for total sales, with two dimensions, namely, (i) the genres of the movies, and (ii) whether the movies are shown on weekdays or weekends.	``` SELECT Genre, Case When is_weekend=0 then 'Weekday' Else 'Weekend' END as 'is_Weekend?', SUM(TotalPrice) AS 'Total Sales' FROM K, F , D WHERE F.DateID=D.DateID AND F.TicketID=K.Ticket_ID Group by Genre, is_weekend; ```
Question 4: Produce a pivot table on total sales in 2018, with two dimensions, namely, (i) the genders of the customers, and (ii) the types of promotions (if any) associated with the transactions.	``` SELECT Gender, Description, SUM(TotalPrice) AS 'Total Sales' FROM C, F, D, P WHERE F.DateID=D.DateID AND D.Date_Year= 2018 AND F.CustomerID=C.CustomerID AND F.PromoID=P.PromoID Group by Gender, Description; ```
Question 5: Produce a pivot table on total sales in 2018, with two dimensions, namely, (i) the genders of the customers, and (ii) the numbers of tickets bought in each transaction.	``` SELECT Gender, No_tickets, SUM(TotalPrice) AS 'Total Sales' FROM F, C, D WHERE F.CustomerID=C.CustomerID AND F.DateID=D.DateID AND D.Date_Year= 2018 GROUP BY Gender, No_Tickets; ```
Question 6: Produce a pivot table on total number of tickets sold in 2018, with two dimensions, namely, (i) the genders of the customers, and (ii) whether the movie is shown in the morning, in the afternoon, or at night.	``` SELECT Gender, SUM(No_tickets) AS 'Total Tickets Sold', 'Morning' as morning FROM F, C, K, D WHERE F.CustomerID=C.CustomerID AND F.TicketID=K.TicketID AND F.DateID=D.DateID AND D.Date_Year= 2018 And K.ShowTime LIKE '06%' OR K.ShowTime LIKE '07%' OR K.ShowTime LIKE '08%' OR K.ShowTime LIKE '09%' OR K.ShowTime LIKE '10%' OR K.ShowTime LIKE '11%' GROUP BY GENDER; UNION ALL SELECT Gender, SUM(No_tickets) AS 'Total Tickets Sold', 'Afternoon' as afternoon FROM F, C, K, D WHERE F.CustomerID=C.CustomerID AND F.TicketID=K.TicketID AND F.DateID=D.DateID AND D.Date_Year= 2018 And K.ShowTime LIKE '12%' OR K.ShowTime LIKE '13%' ```

	OR K.ShowTime LIKE '14%' OR K.ShowTime LIKE '15%' OR K.ShowTime LIKE '16%' OR K.ShowTime LIKE '17%' GROUP BY GENDER; UNION ALL SELECT Gender, SUM(No_tickets) AS 'Total Tickets Sold', 'Night' as night FROM F, C, K, D WHERE F.CustomerID=C.CustomerID AND F.TicketID=K.TicketID AND F.DateID=D.DateID AND D.Date_Year= 2018 And K.ShowTime LIKE '18%' OR K.ShowTime LIKE '19%' OR K.ShowTime LIKE '20%' OR K.ShowTime LIKE '21%' OR K.ShowTime LIKE '22%' OR K.ShowTime LIKE '23%' OR K.ShowTime LIKE '00%' OR K.ShowTime LIKE '01%' GROUP BY Gender; (I think this can be done in another way i.e. if ShowTime contains "Evening" etc rather than an actual time.
Question 7: Produce a pivot table on total sales from 2015 to 2018 for movies directed by Mohamed Khan, with two dimensions, namely, (i) the years of transactions, and (ii) the states in which the cinemas are located.	SELECT Date_Year, States, SUM(TotalPrice) AS 'Total Sales' FROM F, D, K WHERE F.DateID=D.DateID AND (Date_Year Between 2015 AND 2018) AND (K.DirectorName= 'Mohamed Khan') AND F.TicketID=K.TicketID GROUP BY Date_Year, State;
Question 8: Produce a pivot table on total sales for movies where Omar Sharif were casted in, with two dimensions, namely, (i) genres of the movies, and (ii) the genders of customers.	SELECT Gender,Genre , SUM(TotalPrice) AS 'Total Sales' FROM F, C,K WHERE F.TicketID=K.TicketID AND F.CustomerID=C.CustomerID AND Star_name='Omar Sharif' GROUP BY Gender, Genre;
Question 9: Produce a pivot table on total sales for offline transactions in 2018, with two dimensions, namely, (i) the states in which the cinemas are located, and (ii) whether the movie is shown in a small-size, mid-size, or	SELECT CinemaSize, State, SUM(TotalPrice) AS 'Total Sales' FROM K, F, C, D, T WHERE F.TicketID=K.Ticket_id AND F.CustomerID=C.CustomerID AND F.DateID=D.DateID AND T.Transaction_type='Offline' AND Date_Year=2018 AND F.TransactionTypeID=T.TransactionTypeID GROUP BY CinemaSize, State;

large-size hall. (You can define your own categorization of small-size, mid-size, and large-size halls.)	
Question 10: Produce a pivot table on the total sales from 2015 to 2018, with two dimensions, namely, (i) the genders of the customers, and (ii) the ages of the customers at the time of ticket purchase, with ROLLUP to age groups. (You can define your own categorization of age groups, e.g., [1, 10], [11, 20], [21, 30], etc.	SELECT Gender, Age, SUM(TotalPrice) AS 'Total Sales', CASE WHEN AGE BETWEEN 1 AND 10 THEN 'CHILD' WHEN AGE BETWEEN 11 AND 20 THEN 'TEENAGER' WHEN AGE BETWEEN 21 AND 59 THEN 'ADULT' WHEN AGE > 60 THEN 'SENIOR' END AS Age_Group FROM F, C, D WHERE F.CustomerID=C.CustomerID AND F.DateID=D.DateID AND (Date_Year Between 2015 AND 2018) GROUP BY Gender, Rollup(Age_Group, Age);
Question 11: For each city, rank the cinemas in the city in descending order of total sales in 2018.	SELECT CinemaID, State, SUM(TotalPrice) AS 'Total Sales', RANK() OVER (PARTITION BY States ORDER BY SUM(TotalPrice)) AS RANK FROM F, K, D WHERE F.DateID=D.DateID AND Date_Year=2018 AND F.TicketID=K.TicketID GROUP BY State, CinemaID;
Question 12: For each director, rank his/her movies in descending orders of total sales for customers with ages under 40 (at the time of ticket purchases).	SELECT DirectorName, MovieTitle, SUM(TotalPrice) AS 'Total Sales', RANK() OVER (PARTITION BY DirectorName ORDER BY SUM(TotalPrice)) AS RANK FROM F, K, C WHERE F.CustomerID=C.CustomerID AND F.TicketID=F.ticketID AND Age <40
Question 13: Consider the online transactions made with various browsers, for cinemas in different states. For each city, rank the browsers in descending order of the total numbers of transactions made.	SELECT States , Browser, Count(TransactionID) AS 'Total Transaction', RANK()OVER(PARTITION BY States ORDER BY Count(TransactionID)) as RANK FROM T, F WHERE F.TransactionTypeID=T.TransactionTypeID AND Browser != 'NULL' GROUP BY State, Browser;
Question 14: Find the top 10 movies in 2018 (in terms of the total number of	SELECT TOP 10 MovieTitle, Gender, SUM(TotalPrice) AS 'Total Sales', RANK() OVER(PARTITION BY GENDER ORDER BY

tickets sold) for male and female customers, respectively.	SUM(TotalPrice) DESC) FROM K, F, D, C WHERE F.TicketID=K.TicketID AND F.DateID=D.DateID AND D.Date_Year=2018 AND F.CustomerID=C.Customer_id GROUP BY Gender, MovieTitle;
Question 15: For each city, find the top 5 cinemas in terms of the total number of tickets sold from 2014 to 2018.	SELECT TOP 5 City, CinemaLocation, SUM(No_tickets) AS 'Total Sales', RANK() OVER(PARTITION BY City ORDER BY SUM(No_Tickets)) FROM F, D WHERE F.DateID=D.DateID AND DATE_YEAR BETWEEN 2014 AND 2018 GROUP BY CinemaLocation, City;
Question 16: Compute the 8-week moving average of total sales, for each week in 2018.	SELECT Date_WeekNo, SUM(TotalPrice) AS 'Total Sales', AVG(SUM(TotalPrice)) OVER (ORDER BY Date_WeekNo ROWS BETWEEN 7 PRECEDING AND CURRENT ROW) AS 'MOVING AVERAGE' FROM F, D WHERE F.DateID=D.DateID AND Date_Year=2018 GROUP BY Date_WeekNo;
Question 17: Compute largest three 4-week moving averages of total sales, among the weeks in 2018.	SELECT Date_WeekNo, SUM(TotalPrice) AS 'Total Sales', AVG(SUM(TotalPrice)) OVER (ORDER BY Date_WeekNo ROWS BETWEEN 7 PRECEDING AND CURRENT ROW) AS 'MOVING AVERAGE' FROM F, D WHERE F.DateID=D.DateID AND Date_Year=2018 GROUP BY Date_WeekNo;
Question 18: For each city, compute the largest 4-week moving average of total sales from 2010 to 2018.	SELECT City, Date_WeekNo, SUM(TotalPrice) AS 'Total Sales', AVG(SUM(TotalPrice)) OVER (ORDER BY Date_WeekNo ROWS BETWEEN 3 PRECEDING AND CURRENT ROW) AS 'MOVING AVERAGE', Rank() OVER(PARTITION BY CITY ORDER BY SUM(TotalPrice)) as 'R' FROM F, D WHERE F.DateID=D.DateID AND D.Date_Year=2018 GROUP BY City, CinemaSize,Date_WeekNo ORDER BY 'R') WHERE R=1

Please note that these queries might have been changed for phase 3 of the assignment.

Phase 3: Implementation

Table Generation:

TransactionTypeDT

Query | Query History

```
CREATE TABLE TransactionTypeDT
(
    TransactionTypeID   INT PRIMARY KEY NOT NULL,
    TransactionType VARCHAR(512),
    Browser VARCHAR(512),
    System  VARCHAR(512)
);

INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('1', 'Online', 'Chrome', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('2', 'Online', 'Safari', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('3', 'Online', 'Firefox', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('4', 'Online', 'Safari', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('5', 'Online', 'Firefox', 'Linux');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('6', 'Online', 'Chrome', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('7', 'Online', 'Firefox', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('8', 'Online', 'Safari', 'Linux');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('9', 'Online', 'Firefox', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('10', 'Online', 'Chrome', 'Linux');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('11', 'Online', 'Chrome', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('12', 'Online', 'Firefox', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('13', 'Online', 'Firefox', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('14', 'Online', 'Firefox', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('15', 'Online', 'Safari', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('16', 'Online', 'Chrome', 'Linux');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('17', 'Online', 'Safari', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('18', 'Online', 'Safari', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('19', 'Online', 'Safari', 'MacOS');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('20', 'Online', 'Firefox', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('21', 'Online', 'Chrome', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('22', 'Online', 'Firefox', 'Windows');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('23', 'Offline', 'N/A', 'N/A');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('24', 'Offline', 'N/A', 'N/A');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('25', 'Offline', 'N/A', 'N/A');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('26', 'Offline', 'N/A', 'N/A');
INSERT INTO TransactionTypeDT (TransactionTypeID, TransactionType, Browser, System) VALUES ('27', 'Offline', 'N/A', 'N/A');
```

Data Output | Messages | Notifications

PromoDT

```
CREATE TABLE PromoDT
(
    PromoID INT PRIMARY KEY NOT NULL,
    Description VARCHAR(512),
    Discount    INT,
    StartDate   DATE,
    EndDate DATE
);

INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('1', 'medium', '98', '2/27/2014', '3/14/2014');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('2', 'medium', '97', '11/3/2016', '1/16/2017');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('3', 'medium', '68', '3/2/2012', '1/28/2011');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('4', 'small', '21', '10/12/2012', '1/8/2014');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('5', 'small', '90', '4/1/2012', '12/13/2011');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('6', 'small', '69', '4/1/2010', '6/4/2016');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('7', 'large', '55', '3/9/2010', '8/30/2015');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('8', 'medium', '22', '9/18/2017', '8/26/2013');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('9', 'large', '19', '11/19/2017', '7/28/2011');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('10', 'small', '3', '1/27/2010', '11/21/2017');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('11', 'medium', '57', '6/30/2017', '8/29/2016');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('12', 'medium', '25', '4/16/2010', '3/21/2015');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('13', 'large', '26', '4/14/2012', '11/19/2011');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('14', 'small', '85', '5/19/2013', '5/11/2012');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('15', 'medium', '15', '6/12/2010', '11/18/2018');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('16', 'medium', '67', '12/11/2013', '8/6/2012');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('17', 'medium', '86', '4/26/2015', '6/29/2017');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('18', 'medium', '54', '7/9/2016', '10/17/2011');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('19', 'medium', '36', '8/21/2016', '4/14/2013');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('20', 'medium', '89', '7/13/2013', '6/7/2014');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('21', 'medium', '87', '1/15/2014', '10/11/2017');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('22', 'large', '12', '10/15/2015', '5/27/2010');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('23', 'large', '96', '8/31/2014', '11/21/2010');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('24', 'medium', '5', '12/21/2014', '9/20/2017');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('25', 'large', '86', '5/6/2012', '8/5/2011');
INSERT INTO PromoDT (PromoID, Description, Discount, StartDate, EndDate) VALUES ('26', 'medium', '39', '1/9/2010', '9/15/2012');
```

Data Output | Messages | Notifications

DateDT

Query   Query History

```sql
/*script generated for free with OnlineDataGenerator available at: https://www.onlinedatagenerator.com */
CREATE TABLE DateDT(DateID INT PRIMARY KEY NOT NULL,Day INT,Month INT,Year INT,WeekNo INT,IsHoliday BOOLEAN);
INSERT INTO DateDT VALUES('1',10,4,2013,6,'True');
INSERT INTO DateDT VALUES('2',17,4,2013,5,'False');
INSERT INTO DateDT VALUES('3',19,5,2017,6,'True');
INSERT INTO DateDT VALUES('4',30,4,2018,6,'False');
INSERT INTO DateDT VALUES('5',22,3,2013,3,'True');
INSERT INTO DateDT VALUES('6',20,5,2014,4,'True');
INSERT INTO DateDT VALUES('7',4,8,2013,1,'True');
INSERT INTO DateDT VALUES('8',2,7,2010,4,'True');
INSERT INTO DateDT VALUES('9',8,8,2015,3,'True');
INSERT INTO DateDT VALUES('10',8,6,2018,5,'True');
INSERT INTO DateDT VALUES('11',10,11,2012,6,'False');
INSERT INTO DateDT VALUES('12',3,11,2010,1,'True');
INSERT INTO DateDT VALUES('13',5,10,2018,3,'True');
INSERT INTO DateDT VALUES('14',30,1,2016,7,'True');
INSERT INTO DateDT VALUES('15',12,3,2013,1,'True');
INSERT INTO DateDT VALUES('16',2,7,2010,3,'True');
INSERT INTO DateDT VALUES('17',6,6,2017,1,'True');
INSERT INTO DateDT VALUES('18',5,5,2014,4,'True');
INSERT INTO DateDT VALUES('19',4,6,2018,4,'False');
INSERT INTO DateDT VALUES('20',13,9,2012,3,'True');
INSERT INTO DateDT VALUES('21',6,9,2010,2,'False');
INSERT INTO DateDT VALUES('22',1,6,2012,4,'True');
INSERT INTO DateDT VALUES('23',14,10,2015,5,'False');
```

TicketDT

```sql
CREATE TABLE TicketDT
(
    TicketID     INT PRIMARY KEY NOT NULL,
    MovieTitle   VARCHAR(512),
    Seat     INT,
    Row INT,
    Genre    VARCHAR(512),
    Language     VARCHAR(512),
    ShowTime     VARCHAR(512),
    DirectorName     VARCHAR(512),
    StarName     VARCHAR(512),
    IsWeekend    BOOLEAN
);

INSERT INTO TicketDT (TicketID, MovieTitle, Seat, Row, Genre, Language, ShowTime, DirectorName, StarName, IsWeekend) VALUES ('
INSERT INTO TicketDT (TicketID, MovieTitle, Seat, Row, Genre, Language, ShowTime, DirectorName, StarName, IsWeekend) VALUES ('
INSERT INTO TicketDT (TicketID, MovieTitle, Seat, Row, Genre, Language, ShowTime, DirectorName, StarName, IsWeekend) VALUES ('
INSERT INTO TicketDT (TicketID, MovieTitle, Seat, Row, Genre, Language, ShowTime, DirectorName, StarName, IsWeekend) VALUES ('
INSERT INTO TicketDT (TicketID, MovieTitle, Seat, Row, Genre, Language, ShowTime, DirectorName, StarName, IsWeekend) VALUES ('
INSERT INTO TicketDT (TicketID, MovieTitle, Seat, Row, Genre, Language, ShowTime, DirectorName, StarName, IsWeekend) VALUES ('
```

CustomerDT

```sql
/*script generated for free with OnlineDataGenerator available at: https://www.onlinedatagenerator.com */
CREATE TABLE CustomerDT(CustomerID INT PRIMARY KEY NOT NULL,CustomerName VARCHAR(500),Gender VARCHAR(500),Age INT);

INSERT INTO CustomerDT VALUES('1','Gabriel Tanner','Male',94);
INSERT INTO CustomerDT VALUES('2','Allison Silva','Female',31);
INSERT INTO CustomerDT VALUES('3','Anabelle Wilson','Female',84);
INSERT INTO CustomerDT VALUES('4','Sofie Shields','Female',14);
INSERT INTO CustomerDT VALUES('5','Jessica Rose','Female',59);
INSERT INTO CustomerDT VALUES('6','Zara Andrews','Female',75);
INSERT INTO CustomerDT VALUES('7','Daphne Bentley','Female',51);
INSERT INTO CustomerDT VALUES('8','Jade Oakley','Female',81);
INSERT INTO CustomerDT VALUES('9','Roger Osmond','Male',49);
INSERT INTO CustomerDT VALUES('10','Ronald Ingham','Male',48);
INSERT INTO CustomerDT VALUES('11','Noah Gordon','Male',71);
INSERT INTO CustomerDT VALUES('12','Cherish Samuel','Female',30);
INSERT INTO CustomerDT VALUES('13','Makenzie Price','Female',15);
INSERT INTO CustomerDT VALUES('14','Lexi Everett','Female',85);
INSERT INTO CustomerDT VALUES('15','Eryn Norman','Female',88);
INSERT INTO CustomerDT VALUES('16','Mike Moran','Male',87);
INSERT INTO CustomerDT VALUES('17','Joseph Tindall','Male',13);
INSERT INTO CustomerDT VALUES('18','Angelina Ramsey','Female',51);
INSERT INTO CustomerDT VALUES('19','David Nelson','Male',70);
INSERT INTO CustomerDT VALUES('20','Rufus Stone','Male',99);
INSERT INTO CustomerDT VALUES('21','Mike Collins','Male',46);
```

TransactionFT

```
CREATE TABLE TransactionFT
(
    TransactionTypeID   INT NOT NULL,
    DateID  INT NOT NULL,
    CustomerID  INT NOT NULL,
    TicketID    INT NOT NULL,
    PromoID INT NOT NULL,
    TotalPrice  INT NOT NULL,
    NoTickets   INT NOT NULL,
    Time    VARCHAR(512) NOT NULL,
    CinemaState VARCHAR(512) NOT NULL,
    CinemaCity  VARCHAR(512) NOT NULL,
    CinemaSize  VARCHAR(512) NOT NULL,
    CONSTRAINT fk_TransactionTypeDT FOREIGN KEY (TransactionTypeID) REFERENCES TransactionTypeDT (TransactionTypeID),
    CONSTRAINT fk_CustomerDT FOREIGN KEY (CustomerID) REFERENCES CustomerDT (CustomerID),
    CONSTRAINT fk_TicketDT FOREIGN KEY (TicketID) REFERENCES TicketDT (TicketID),
    CONSTRAINT fk_DateDT FOREIGN KEY (DateID) REFERENCES DateDT (DateID),
    CONSTRAINT fk_PromoDT FOREIGN KEY (PromoID) REFERENCES PromoDT (PromoID)
);

INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('1'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('2'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('3'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('4'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('5'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('6'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('7'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('8'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('9'
INSERT INTO TransactionFT (TransactionTypeID, DateID, CustomerID, TicketID, PromoID, TotalPrice, NoTickets, Time, CinemaState, CinemaCity, CinemaSize) VALUES ('
```

Data Output | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 5 secs 924 msec.
```

Total rows: 0 of 0 | Query complete 00:00:05.924 | Ln 11, Col 39

Table Population:

TransactionTypeDT:

Query Query History

```
SELECT * FROM TransactionTypeDT
```

Data Output Messages Notifications

	transactiontypeid [PK] integer	transactiontype character varying (512)	browser character varying (512)	system character varying (512)
1	1	Online	Chrome	MacOS
2	2	Online	Safari	MacOS
3	3	Online	Firefox	Windows
4	4	Online	Safari	Windows
5	5	Online	Firefox	Linux
6	6	Online	Chrome	Windows
7	7	Online	Firefox	MacOS
8	8	Online	Safari	Linux
9	9	Online	Firefox	MacOS
10	10	Online	Chrome	Linux
11	11	Online	Chrome	Windows
12	12	Online	Firefox	MacOS
13	13	Online	Firefox	MacOS
14	14	Online	Firefox	Windows
15	15	Online	Safari	MacOS
16	16	Online	Chrome	Linux
17	17	Online	Safari	MacOS
18	18	Online	Safari	Windows
19	19	Online	Safari	MacOS
20	20	Online	Firefox	Windows

PromoDT:

	promoid [PK] integer	description character varying (512)	discount integer	startdate date	enddate date
1	1	medium	98	2014-02-27	2014-03-14
2	2	medium	97	2016-11-03	2017-01-16
3	3	medium	68	2012-03-02	2011-01-28
4	4	small	21	2012-10-12	2014-01-08
5	5	small	90	2012-04-01	2011-12-13
6	6	small	69	2010-04-01	2016-06-04
7	7	large	55	2010-03-09	2015-08-30
8	8	medium	22	2017-09-18	2013-08-26
9	9	large	19	2017-11-19	2011-07-28
10	10	small	3	2010-01-27	2017-11-21
11	11	medium	57	2017-06-30	2016-08-29
12	12	medium	25	2010-04-16	2015-03-21
13	13	large	26	2012-04-14	2011-11-19
14	14	small	85	2013-05-19	2012-05-11
15	15	medium	15	2010-06-12	2018-11-18
16	16	medium	67	2013-12-11	2012-08-06
17	17	medium	86	2015-04-26	2017-06-29
18	18	medium	54	2016-07-09	2011-10-17
19	19	medium	36	2016-08-21	2013-04-14
20	20	medium	89	2013-07-13	2014-06-07

Total rows: 1000 of 100000 Query complete 00:00:00.347 Ln 2, Col 22

TicketDT:

Query Query History

```
SELECT * FROM TicketDT;
```

Data Output Messages Notifications

	ticketid [PK] integer	movietitle character varying (512)	seat integer	row integer	genre character varying (512)	language character varying (512)	showtime character varying (512)	directorname character vary
1	1	The Notebook	4	1	Horror	Romanian	Morning	Carol Wheele
2	2	Brimstone	6	4	Animation	Japanese	Morning	Christine Dick
3	3	The Godfather	8	5	Crime	Swedish	Afternoon	Barney Warne
4	4	The Notebook	3	14	Action	Persian	Afternoon	Domenic Glas
5	5	The Notebook	3	1	Animation	Italian	Night	Emmanuelle
6	6	Star Wars	11	3	Family	Turkish	Morning	Aleksandra A
7	7	Blood Diamond	4	11	History	Bhojpuri	Morning	Scarlett Stew

CustomerDT:

Data Output | Messages | Notifications

	customerid [PK] integer	customername character varying (500)	gender character varying (500)	age integer
1	1	Gabriel Tanner	Male	94
2	2	Allison Silva	Female	31
3	3	Anabelle Wilson	Female	84
4	4	Sofie Shields	Female	14
5	5	Jessica Rose	Female	59
6	6	Zara Andrews	Female	75
7	7	Daphne Bentley	Female	51
8	8	Jade Oakley	Female	81
9	9	Roger Osmond	Male	49
10	10	Ronald Ingham	Male	48
11	11	Noah Gordon	Male	71
12	12	Cherish Samuel	Female	30
13	13	Makenzie Price	Female	15
14	14	Lexi Everett	Female	85
15	15	Eryn Norman	Female	88
16	16	Mike Moran	Male	87
17	17	Joseph Tindall	Male	13
18	18	Angelina Ramsey	Female	51
19	19	David Nelson	Male	70
20	20	Rufus Stone	Male	99

Total rows: 1000 of 100000 | Query complete 00:00:00.085 | Ln 1, Col 26

DateDT

```
1  SELECT * FROM DateDT;
```

Data Output | Messages | Notifications

	dateid [PK] integer	day integer	month integer	year integer	weekno integer	isholiday boolean
1	1	10	4	2013	6	true
2	2	17	4	2013	5	false
3	3	19	5	2017	6	true
4	4	30	4	2018	6	false
5	5	22	3	2013	3	true
6	6	20	5	2014	4	true
7	7	4	8	2013	1	true
8	8	2	7	2010	4	true
9	9	8	8	2015	3	true
10	10	8	6	2018	5	true
11	11	10	11	2012	6	false
12	12	3	11	2010	1	true
13	13	5	10	2018	3	true
14	14	30	1	2016	7	true
15	15	12	3	2013	1	true
16	16	2	7	2010	3	true
17	17	6	6	2017	1	true
18	18	5	5	2014	4	true
19	19	4	6	2018	4	false
20	20	13	9	2012	3	true

TransactionFT

Query | Query History

```
1  SELECT * FROM TransactionFT;
```

Data Output | Messages | Notifications

	transactiontypeid integer	dateid integer	customerid integer	ticketid integer	promoid integer	totalprice integer	notickets integer	time character varying (512)	cinemastate character varying (512)	cinemacity character varying (512)	cinemasize character varying (512)
1	1	1	1	1	1	259	2	10:18 AM	Rhode Island	Scottsdale	Large
2	2	2	2	2	2	627	9	11:53 AM	Pennsylvania	Bellevue	Small
3	3	3	3	3	3	2174	9	8:06 AM	Delaware	Reno	Medium
4	4	4	4	4	4	1061	2	9:10 AM	Kansas	Ontario	Small
5	5	5	5	5	5	2086	2	3:57 AM	Florida	London	Small
6	6	6	6	6	6	2355	8	9:40 AM	Minnesota	Saint Paul	Medium
7	7	7	7	7	7	2223	6	8:06 AM	North Carolina	Bridgeport	Medium
8	8	8	8	8	8	920	7	9:18 PM	Missouri	New Orleans	Small
9	9	9	9	9	9	844	7	4:53 AM	South Carolina	San Jose	Large
10	10	10	10	10	10	260	9	4:15 PM	West Virginia	Nashville	Medium
11	11	11	11	11	11	2480	5	4:09 AM	Vermont	Laredo	Medium
12	12	12	12	12	12	1354	9	4:17 PM	Hawaii	Philadelphia	Large
13	13	13	13	13	13	1891	7	7:39 AM	Kentucky	Norfolk	Large
14	14	14	14	14	14	2173	6	4:20 AM	New Hampshire	Stockton	Small
15	15	15	15	15	15	2400	1	1:45 PM	Minnesota	Boston	Medium
16	16	16	16	16	16	1044	3	8:47 AM	New Jersey	Lakewood	Medium
17	17	17	17	17	17	1872	6	7:27 AM	Colorado	Omaha	Medium
18	18	18	18	18	18	486	3	11:18 AM	North Dakota	Norfolk	Large
19	19	19	19	19	19	1006	2	12:12 AM	Maine	London	Large
20	20	20	20	20	20	335	5	2:28 AM	New Hampshire	Sacramento	Medium

Query Results:

1. Produce a pivot table for total sales, with two dimensions, namely, (i) the “Year”s of the transactions, and (ii) the “Gender”s of the customers.

Query Query History

```
1  --Query 1
2  SELECT Gender, Year,SUM(TotalPrice)
3  FROM CustomerDT,TransactionFT, DateDT
4  WHERE TransactionFT.DateID=DateDT.DateID AND TransactionFT.CustomerID=
5  CustomerDT.CustomerID
6  Group by Year, Gender;
```

Data Output Messages Notifications

	gender character varying (500)	year integer	sum bigint
1	Female	2011	7662020
2	Female	2012	7597460
3	Male	2012	7849797
4	Female	2016	7539418
5	Male	2015	7660544
6	Female	2010	7594107
7	Female	2015	7634462
8	Female	2013	7633223
9	Male	2013	7713109
10	Female	2014	7649834
11	Female	2018	7639436
12	Female	2017	7540893
13	Male	2017	7612704
14	Male	2018	7572909
15	Male	2014	7458164
16	Male	2010	7583913
17	Male	2011	7759373
18	Male	2016	7577526

2. Produce a pivot table for total sales, with two dimensions, namely, (i) the months of the transactions with ROLLUP to “Year”s, and (ii) whether or not the transactions are made online or offline.

Query Query History

```
--Query 2
SELECT Month, Year,TransactionType,SUM(TotalPrice)
FROM TransactionTypeDT,TransactionFT, DateDT
WHERE TransactionFT.DateID=DateDT.DateID AND TransactionFT.TransactionTypeID=
TransactionTypeDT.TransactionTypeID
Group by TransactionType, ROLLUP(Year,Month);
```

Data Output Messages Notifications

	month integer	year integer	transactiontype character varying (512)	sum bigint
1	5	2010	Offline	1349150
2	8	2011	Offline	1366430
3	10	2011	Offline	1419830
4	10	2017	Offline	1397388
5	1	2010	Offline	1342014
6	8	2014	Offline	1346710
7	5	2016	Offline	1319567
8	4	2018	Online	1061
9	10	2016	Offline	1427102
10	10	2018	Online	1891
11	6	2018	Offline	1358412
12	7	2010	Online	1964
13	11	2010	Offline	1502241
14	5	2018	Offline	1437710
15	7	2017	Offline	1312070
16	9	2015	Offline	1380114
17	7	2014	Offline	1370436
18	9	2014	Offline	1411338
19	7	2010	Offline	1306157
20	10	2018	Offline	1429900

Total rows: 135 of 135 Query complete 00:00:00.604 Ln 6, Col 46

3. Produce a pivot table for total sales, with two dimensions, namely, (i) the genres of the movies, and (ii) whether the movies are shown on weekdays or weekends.

Query Query History

```
--Query 3
SELECT Genre, IsWeekend,SUM(TotalPrice)
FROM TicketDT,TransactionFT
WHERE TransactionFT.TicketID=TicketDT.TicketID
Group by Genre, IsWeekend;
```

Data Output Messages Notifications

	genre character varying (512)	isweekend boolean	sum bigint
1	Action	false	4563791
2	Action	true	4522394
3	Adventure	false	4613976
4	Adventure	true	4686822
5	Animation	false	4705278
6	Animation	true	4576412
7	Biography	false	4364456
8	Biography	true	4646878
9	Comedy	false	4441483
10	Comedy	true	4612946
11	Crime	false	4632120
12	Crime	true	4602693
13	Drama	false	4612494
14	Drama	true	4594668
15	Family	false	4555074
16	Family	true	4590338
17	Fantasy	false	4523121
18	Fantasy	true	4573044
19	History	false	4542391
20	History	true	4502332

Total rows: 30 of 30 Query complete 00:00:00.616 Ln 5, Col 26

4. Produce a pivot table on total sales in 2018, with two dimensions, namely, (i) the “Gender”s of the customers, and (ii) the types of promotions (if any) associated with the transactions.

Query Query History

```
1  --Query 4
2  SELECT Gender, Description,SUM(TotalPrice)
3  FROM PromoDT,TransactionFT,CustomerDT,DateDT
4  WHERE  TransactionFT.DateID=DateDT.DateID AND Year= 2018 AND TransactionFT.CustomerID=CustomerDT.CustomerID AND TransactionFT.PromoID=PromoDT.PromoID
5  Group by Gender, Description;
```

Data Output Messages Notifications

	gender character varying (500)	description character varying (512)	sum bigint
1	Male	medium	2568371
2	Female	large	2552466
3	Male	large	2500843
4	Male	small	2503695
5	Female	small	2525973
6	Female	medium	2560997

5. Produce a pivot table on total sales in 2018, with two dimensions, namely, (i) the Genders of the customers, and (ii) the numbers of tickets bought in each transaction.

Query Query History

```
1  --Query 5
2  SELECT Gender, NoTickets,SUM(TotalPrice)
3  FROM TransactionFT,CustomerDT,DateDT
4  WHERE  TransactionFT.DateID=DateDT.DateID AND Year= 2018 AND TransactionFT.CustomerID=CustomerDT.CustomerID
5  Group by Gender, NoTickets;
```

Data Output Messages Notifications

	gender character varying (500)	notickets integer	sum bigint
1	Male	2	817318
2	Male	1	821394
3	Female	8	859889
4	Female	5	827752
5	Female	9	899662
6	Male	6	873403
7	Female	2	828008
8	Male	5	858656
9	Male	7	863517
10	Female	7	888861
11	Male	4	847150
12	Male	8	855092
13	Male	9	839122
14	Female	1	814032
15	Female	4	835184
16	Female	6	846642
17	Male	3	797257
18	Female	3	839406

Total rows: 18 of 18 Query complete 00:00:00.277 Ln 5, Col 28

6. Produce a pivot table on the total number of tickets sold in 2018, with two dimensions, namely, (i) the Genders of the customers, and (ii) whether the movie is shown in the morning, in the afternoon, or at night.

Query Query History

```
--Query 6
SELECT Gender, ShowTime,SUM(TotalPrice)
FROM TransactionFT,CustomerDT,TicketDT,DateDT
WHERE  TransactionFT.TicketID=TicketDT.TicketID AND Year= 2018 AND TransactionFT.CustomerID=CustomerDT.CustomerID AND TransactionFT.DateID=DateDT.DateID
Group by Gender, ShowTime;
```

Data Output Messages Notifications

	gender character varying (500)	showtime character varying (512)	sum bigint
1	Female	Afternoon	2559483
2	Female	Morning	2507001
3	Female	Night	2572952
4	Male	Afternoon	2559206
5	Male	Morning	2476848
6	Male	Night	2536855

7. Produce a pivot table on total sales from 2015 to 2018 for movies directed by Mohamed Khan, with two dimensions, namely, (i) the “Year”s of transactions, and (ii) the states in which the cinemas are located.

Query Query History

```
--Query 7
SELECT Year, CinemaState,SUM(TotalPrice)
FROM TransactionFT,TicketDT,DateDT
WHERE  TransactionFT.TicketID=TicketDT.TicketID AND  TransactionFT.DateID=DateDT.DateID AND (YEAR Between 2015 and 2018) AND DirectorName = 'Mohamed Khan'
Group by Year, CinemaState;
```

Data Output Messages Notifications

	year integer	cinemastate character varying (512)	sum bigint
1	2016	Florida	1803
2	2016	Michigan	1666
3	2016	New Jersey	1621
4	2017	Missouri	855
5	2018	Pennsylvania	593

8. Produce a pivot table on total sales for movies where Omar Sharif was cast in, with two dimensions, namely, (i) genres of the movies, and (ii) the Genders of customers.

Query Query History

```
--Query 8
SELECT Gender, Genre,SUM(TotalPrice)
FROM TransactionFT,TicketDT,CustomerDT
WHERE  TransactionFT.TicketID=TicketDT.TicketID AND  TransactionFT.CustomerID=CustomerDT.CustomerID AND   StarName = 'Omar Sharif'
Group by Genre, Gender;
```

Data Output Messages Notifications

	gender character varying (500)	genre character varying (512)	sum bigint
1	Male	Animation	1066
2	Male	Fantasy	2032
3	Female	History	1186
4	Female	Horror	1806
5	Male	Musical	3911
6	Female	Sci-Fi	3186
7	Male	Sci-Fi	2309
8	Female	Thriller	3893

9. Produce a pivot table on total sales for offline transactions in 2018, with two dimensions, namely, (i) the states in which the cinemas are located, and (ii) whether the movie is shown in a small size, mid-size, or large size hall. (You can define your own categorization of small-size, mid-size, and large-size halls.)

Query Query History

```
--Query 9
SELECT CinemaState, CinemaSize,SUM(TotalPrice)
FROM TransactionFT,TicketDT,TransactionTypeDT,DateDT
WHERE  TransactionFT.TicketID=TicketDT.TicketID AND  TransactionFT.TransactionTypeID=TransactionTypeDT.TransactionTypeID
AND TransactionFT.DateID=DateDT.DateID AND TransactionType = 'Offline' AND Year = 2018
Group by CinemaState, CinemaSize;
```

Data Output Messages Notifications

	cinemastate character varying (512)	cinemasize character varying (512)	sum bigint
1	Alabama	Large	118719
2	Alabama	Medium	107240
3	Alabama	Small	109457
4	Alaska	Large	110872
5	Alaska	Medium	110115
6	Alaska	Small	99729
7	Arizona	Large	92642
8	Arizona	Medium	70577
9	Arizona	Small	82425
10	Arkansas	Large	79443
11	Arkansas	Medium	101257
12	Arkansas	Small	95088
13	California	Large	118918
14	California	Medium	102061
15	California	Small	87466
16	Colorado	Large	104911
17	Colorado	Medium	103848
18	Colorado	Small	108014
19	Connecticut	Large	114336
20	Connecticut	Medium	94214

Data saved successfully.

10. Produce a pivot table on the total sales from 2015 to 2018, with two dimensions, namely, (i) the Genders of the customers, and (ii) the Ages of the customers at the time of ticket purchase, with ROLLUP to “Age” groups. (You can define your own categorization of “Age” groups, e.g., [1, 10], [11, 20], [21,30], etc.

Query Query History

```
--Query 10
SELECT Gender, Age, SUM(TotalPrice),
    CASE WHEN AGE BETWEEN 1 AND 10
        THEN 'CHILD'
    WHEN AGE BETWEEN 11 AND 20 THEN
        'TEENAGER'
    WHEN AGE BETWEEN 21 AND 59 THEN
        'ADULT'
    WHEN AGE > 60 THEN 'SENIOR'
END AS Age_Group
FROM TransactionFT,CustomerDT,DateDT
WHERE TransactionFT.CustomerID=CustomerDT.CustomerID
AND TransactionFT.DateID=DateDT.DateID  AND (Year Between 2015 AND 2018)
Group by Gender, ROLLUP(Age_Group,Age);
```

Data Output Messages Notifications

	gender character varying (500)	age integer	sum bigint	age_group text
1	Male	29	300708	ADULT
2	Male	97	326748	SENIOR
3	Female	87	307946	SENIOR
4	Male	82	321512	SENIOR
5	Female	24	294179	ADULT
6	Male	35	304305	ADULT
7	Male	42	295483	ADULT
8	Male	85	326575	SENIOR
9	Male	26	284441	ADULT
10	Male	30	352950	ADULT
11	Male	89	286329	SENIOR
12	Male	80	309473	SENIOR
13	Female	51	317626	ADULT
14	Male	55	310465	ADULT
15	Male	21	337695	ADULT
16	Male	95	379407	SENIOR
17	Male	7	300249	CHILD
18	Female	39	301906	ADULT
19	Male	16	306569	TEENAGER
20	Female	88	319184	SENIOR

11. For each city, rank the cinemas in the city in descending order of total sales in 2018.

Query Query History

```
1  --Query 11
2  SELECT CinemaState, CinemaCity,SUM(TotalPrice) ,
3  RANK() OVER ( PARTITION BY CinemaCity ORDER BY
4  SUM(TotalPrice)DESC) AS RANK
5  FROM TransactionFT, TicketDT, DateDT
6  WHERE TransactionFT.DateID=DateDT.DateID AND Year=2018
7  AND TransactionFT.TicketID=TicketDT.TicketID
8  GROUP BY CinemaState,CinemaCity;
```

Data Output Messages Notifications

	cinemastate character varying (512)	cinemacity character varying (512)	sum bigint	rank bigint
1	Tennessee	Albuquerque	7283	1
2	West Virginia	Albuquerque	7098	2
3	South Carolina	Albuquerque	6832	3
4	Pennsylvania	Albuquerque	5746	4
5	Colorado	Albuquerque	5204	5
6	Maryland	Albuquerque	4604	6
7	Oklahoma	Albuquerque	3978	7
8	Texas	Albuquerque	3751	8
9	Rhode Island	Albuquerque	3743	9
10	Kentucky	Albuquerque	3667	10
11	South Dakota	Albuquerque	3629	11
12	Utah	Albuquerque	3609	12
13	Vermont	Albuquerque	3409	13
14	Michigan	Albuquerque	3401	14
15	Alabama	Albuquerque	3384	15
16	Kansas	Albuquerque	3365	16
17	Illinois	Albuquerque	3320	17
18	Maine	Albuquerque	3010	18
19	New Hampshire	Albuquerque	2853	19
20	Missouri	Albuquerque	2747	20
21	Wyoming	Albuquerque	2695	21
22	Massachusetts	Albuquerque	2512	22
23	Nevada	Albuquerque	2482	23
24	New Mexico	Albuquerque	2455	24
25	Iowa	Albuquerque	2447	25
26	Alaska	Albuquerque	2275	26

Total rows: 1000 of 4651 Query complete 00:00:00.107 Ln 7, Col 45

12. For each director, rank his/her movies in descending orders of total sales for customers with Ages under 40 (at the time of ticket purchases).

Query | Query History

```
--Query 12
SELECT DirectorName, MovieTitle, SUM(TotalPrice),
RANK() OVER (PARTITION BY DirectorName ORDER
BY SUM(TotalPrice)DESC) AS RANK
FROM TransactionFT, TicketDT, CustomerDT
WHERE TransactionFT.CustomerID=CustomerDT.CustomerID
AND TransactionFT.TicketID=TicketDT.TicketID
AND Age <40
Group By DirectorName,MovieTitle;
```

Data Output | Messages | Notifications

	directorname character varying (512)	movietitle character varying (512)	sum bigint	rank bigint
54	Abbey Weasley	Rambo III	396	1
55	Abbey Windsor	Thor : Ragnarok	2065	1
56	Abbey Yard	Allied	1979	1
57	Abdul Adams	Terminator I	1862	1
58	Abdul Adler	American Pie	2464	1
59	Abdul Alldridge	Avengers Infinity War	312	1
60	Abdul Allington	Iron Man	967	1
61	Abdul Amstead	American Pie	1769	1
62	Abdul Anderson	Romeo and Juliet	1282	1
63	Abdul Andrews	Batman Begins	1658	1
64	Abdul Archer	Iron Man	462	1
65	Abdul Atkinson	Octav	1511	1
66	Abdul Ballard	SING	397	1
67	Abdul Bell	Harry Potter and The Chamber Of Secrets	1373	1
68	Abdul Bingham	Chronicles of Narnia	2457	1
69	Abdul Bingham	Iron Man	1964	2
70	Abdul Booth	The Godfather	666	1
71	Abdul Briggs	Transformers	1286	1

Total rows: 1000 of 36556 | Query complete 00:00:00.299 | Ln 9, Col 34

13. Consider the online transactions made with various browsers, for cinemas in different states. For each city, rank the browsers in descending order of the total numbers of transactions made.

Query | Query History

```
--Query 13
SELECT
  CinemaCity,
  Browser,
  COUNT(*) AS TotalTransactions,
  RANK() OVER (PARTITION BY CinemaCity ORDER BY COUNT(*) DESC) AS BrowserRank
FROM
  TransactionFT,TransactionTypeDT
WHERE
  Browser != 'N/A'
  AND TransactionFT.TransactionTypeID = TransactionTypeDT.TransactionTypeID
GROUP BY
  CinemaCity,
  Browser
ORDER BY
  CinemaCity,
  BrowserRank;
```

Data Output | Messages | Notifications

	cinemacity character varying (512)	browser character varying (512)	totaltransactions bigint	browserrank bigint
1	Bellevue	Safari	1	1
2	Boston	Safari	1	1
3	Bridgeport	Firefox	1	1
4	Lakewood	Chrome	1	1
5	Laredo	Chrome	1	1
6	London	Firefox	1	1
7	London	Safari	1	1
8	Milwaukee	Chrome	1	1
9	Nashville	Chrome	1	1
10	New Orleans	Safari	1	1
11	Norfolk	Firefox	1	1
12	Norfolk	Safari	1	1
13	Omaha	Safari	1	1
14	Ontario	Safari	1	1
15	Philadelphia	Firefox	1	1
16	Reno	Firefox	1	1
17	Sacramento	Firefox	1	1
18	Saint Paul	Chrome	1	1
19	San Francisco	Firefox	1	1

Total rows: 22 of 22 | Query complete 00:00:00.117

14. Find the top 10 movies in 2018 (in terms of the total number of tickets sold) for male and female customers, respectively.

```
WITH TempV AS
(SELECT MovieTitle,Gender,SUM(NoTickets) as "TotalTicketsSold"
,RANK () OVER(PARTITION BY Gender ORDER BY SUM(NoTickets) DESC) AS R
FROM TransactionFT, DateDT, CustomerDT, TicketDT
WHERE TransactionFT.DateID = DateDT.DateID AND TransactionFT.CustomerID = CustomerDT.CustomerID AND TransactionFT.TicketID = TicketDT.TicketID AND Year = 2018
Group by MovieTitle,Gender)
SELECT * FROM TempV WHERE R <= 10;
```

Data Output | Messages | Notifications

	movietitle character varying (512)	gender character varying (500)	TotalTicketsSold bigint	r bigint
1	Friends With Kids	Female	638	1
2	Rambo II	Female	624	2
3	The Dark Tower	Female	607	3
4	Romeo and Juliet	Female	606	4
5	Avatar	Female	601	5
6	Iron Man	Female	594	6
7	Pretty Woman	Female	590	7
8	The Dark Knight	Female	589	8
9	Chronicles of Narnia	Female	588	9
10	Thor : Ragnarok	Female	588	9
11	Shutter Island	Male	656	1
12	Black Panther	Male	640	2
13	Casablanca	Male	626	3
14	The Godfather	Male	619	4
15	Harry Potter and The Sorcerers Stone	Male	611	5
16	Interstellar	Male	609	6
17	Chocolat	Male	603	7
18	Snatch	Male	601	8
19	Finding Dory	Male	595	9
20	Brimstone	Male	588	10

Total rows: 20 of 20 | Query complete 00:00:00.353 | Ln 9, Col 35

15. For each city, find the top 5 cinemas in terms of the total number of tickets sold from 2014 to 2018.

Query | Query History

```
--Query 15
SELECT CinemaCity, CinemaState, SUM(NoTickets) AS TotalTicketsSold,RANK() OVER(PARTITION BY CinemaCity ORDER BY
SUM(NoTickets)DESC)
FROM TransactionFT,DateDT
WHERE TransactionFT.DateID = DateDT.DateID AND(Year Between 2014 and 2018)
Group By CinemaCity, CinemaState
ORDER By CinemaCity, TotalTicketsSold
LIMIT 5;
```

Data Output | Messages | Notifications

	cinemacity character varying (512)	cinemastate character varying (512)	totalticketssold bigint	rank bigint
1	Albuquerque	Maine	13	50
2	Albuquerque	Idaho	20	49
3	Albuquerque	Washington	21	48
4	Albuquerque	Wyoming	25	47
5	Albuquerque	Connecticut	27	46

16. Compute the 8-week moving average of total sales, for each week in 2018.

Query Query History

```
--Query 16
SELECT WeekNo, SUM(TotalPrice) AS "Total
Sales", AVG(SUM(TotalPrice)) OVER (ORDER BY
WeekNo ROWS BETWEEN 7 PRECEDING AND
CURRENT ROW) AS "MOVING AVERAGE"
FROM TransactionFT, DateDT
WHERE TransactionFT.DateID=DateDT.DateID AND Year =2018
GROUP BY WeekNo;
```

Data Output Messages Notifications

	weekno integer	Total Sales bigint	MOVING AVERAGE numeric
1	1	2192687	2192687.000000000000
2	2	2161059	2176873.000000000000
3	3	2172426	2175390.666666666667
4	4	2217845	2186004.250000000000
5	5	2140054	2176814.200000000000
6	6	2127792	2168643.833333333333
7	7	2200482	2173192.142857142857

17. Compute the largest three 4-week moving averages of total sales, among the weeks in 2018.

Query Query History

```
--Query 17

WITH TempV AS
(SELECT WeekNo, Sum(TotalPrice),AVG(SUM(TotalPrice)) OVER (ORDER BY WeekNo ROWS BETWEEN 3 PRECEDING AND CURRENT ROW)
 AS A
 FROM TransactionFT, DateDT
 WHERE TransactionFT.DateID = DateDT.DateID AND Year = 2018
 Group By WeekNo),
 TempV2 AS
 (SELECT WeekNo, A, RANK() OVER(ORDER BY A DESC) AS R FROM TempV)
 SELECT * FROM TempV2
 WHERE R<=3;
```

Data Output Messages Notifications

	weekno integer	a numeric	r bigint
1	1	2192687.000000000000	1
2	4	2186004.250000000000	2
3	2	2176873.000000000000	3

18. For each city, compute the largest 4-week moving average of total sales from 2010 to 2018.

Query | Query History

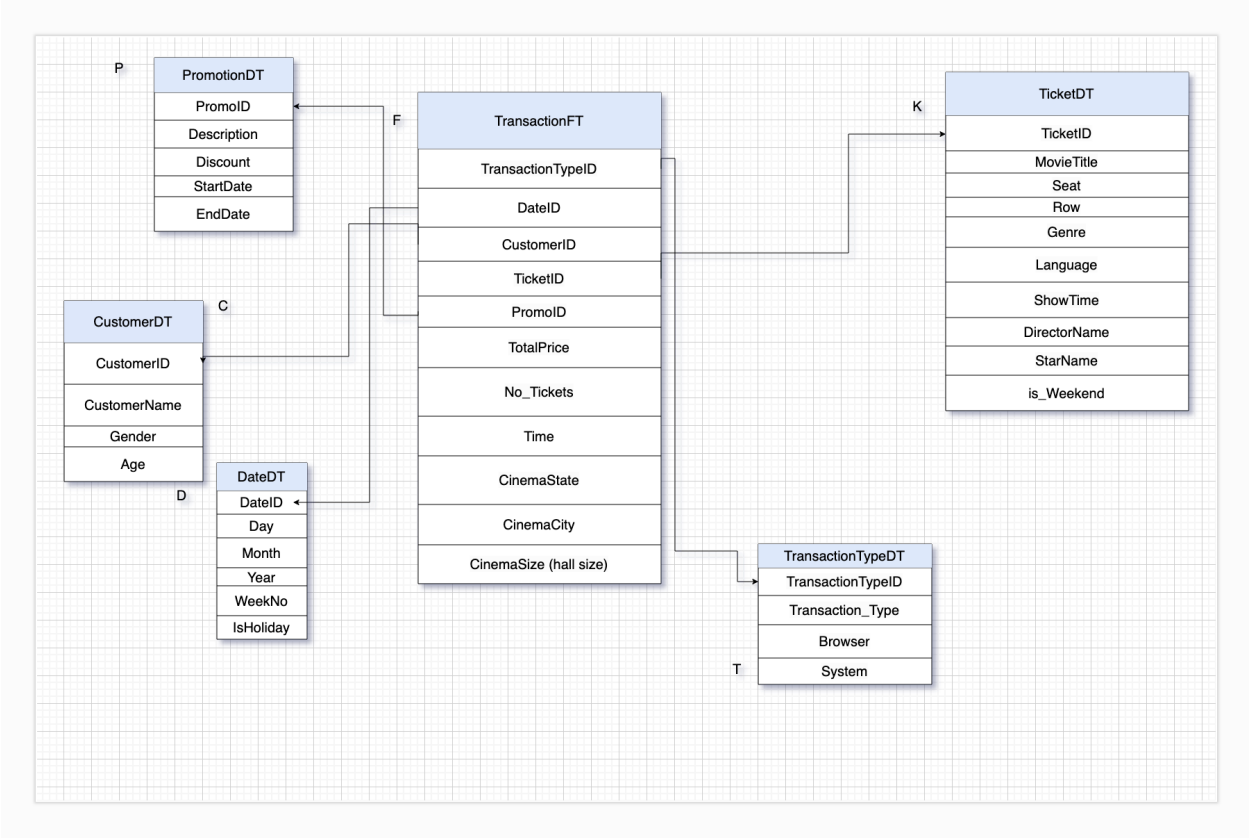
```
--Query 18

WITH TempV AS
(SELECT CinemaCity,WeekNo,SUM(TotalPrice), AVG (SUM(TotalPrice))OVER (PARTITION BY CinemaCity ORDER BY WeekNo ROWS BETWEEN 3 PRECEDING AND CURRENT ROW) AS A
FROM TransactionFT,TransactionTypeDT,DateDT
WHERE TransactionFT.DateID=DateDT.DateID AND TransactionFT.TransactionTypeID=TransactionTypeDT.TransactionTypeID AND Year >= 2010 AND Year <= 2018 GROUP BY CinemaCity
    (SELECT CinemaCity,WeekNo,A,
    RANK() OVER (PARTITION BY CinemaCity ORDER BY A DESC) AS R FROM TempV)
SELECT * FROM TempV2
WHERE R =1 ;
```

Data Output | Messages | Notifications

	cinemacity character varying (512)	weekno integer	a numeric	r bigint
1	Albuquerque	3	178652.333333333333	1
2	Amarillo	1	199894.000000000000	1
3	Anaheim	7	185798.500000000000	1
4	Arlington	1	218114.000000000000	1
5	Atlanta	5	179981.250000000000	1
6	Bakersfield	1	198454.000000000000	1
7	Baltimore	6	202861.000000000000	1
8	Bellevue	6	187481.750000000000	1
9	Berlin	6	187123.000000000000	1
10	Berna	7	188080.750000000000	1
11	Boston	1	223298.000000000000	1
12	Bridgeport	7	187760.250000000000	1
13	Bucharest	5	169527.500000000000	1
14	Charlotte	2	202244.500000000000	1
15	Chicago	1	213146.000000000000	1
16	Cincinnati	1	192796.000000000000	1
17	Colorado Springs	6	190144.250000000000	1
18	Columbus	1	200380.000000000000	1
19	Dallas	5	191025.500000000000	1
20	Denver	3	187045.666666666667	1
21	Detroit	1	205801.000000000000	1
22	El Paso	1	193025.000000000000	1
23	Escondido	4	194907.250000000000	1

Total rows: 107 of 107 | Query complete 00:00:00.147 | Ln 10, Col 14

Schema that I based all of above on:



Just some redundancy changes.

ReadMe:

Just need to copy queries from the file.