Hotel Database Management System

Team 007

Stuti Singh, Aditi Krishna,

Parth Shah, Abhiram Desai,

Shiwangani Jaiswal

Purpose

- The purpose of the database is to **maintain the data used to track and report** on hotel management activities for an international hotel chain.

- To be used by the hotel's administrative staff only and will not duplicate information.

- **Reason for Hotel DB Management**: Hotels have an intricately woven set of stakeholders (or entities) from chains to employees to customers etc. that are interconnected to each other and needs database management on a high level

Objectives

- Allow a hotel's departments to **generate reports** on customer, employees, hotel locations, event reservations etc.

- Provide information to enhance or **improve marketing efforts** to specific targets like room bookings and restaurant bookings

- Gain **insight for revenue**/budgeting reports based on customer habits and seasons (via dates booked)

- Permit **inventory management** department and facilities department to maintain inventory and coordinate with suppliers

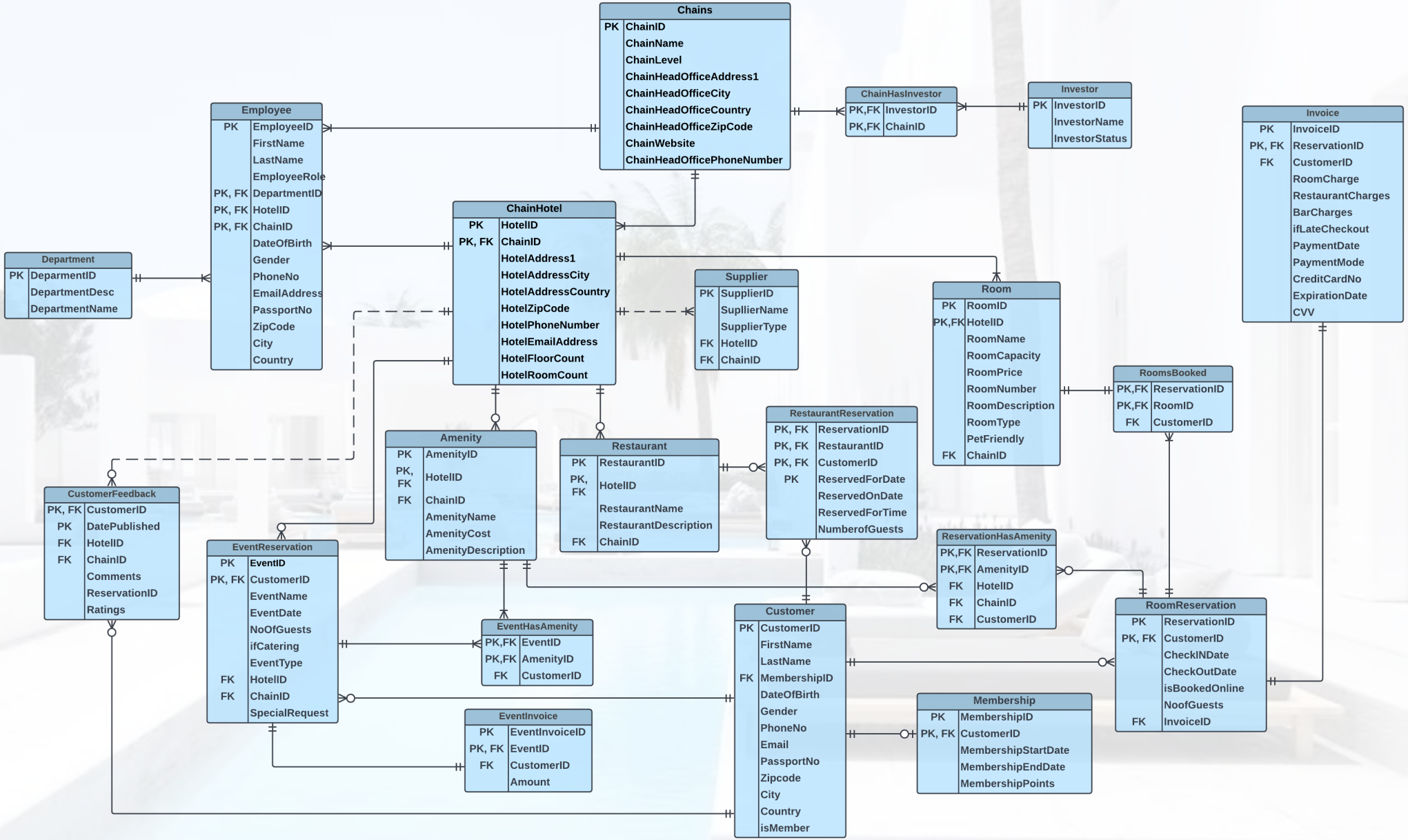
- Allow Chains to **keep track of the performance** of their hotels

- Let Hotels **maintain a record of rooms**, amenities, restaurants and their booking data

Business Rules

- Each parent organization (**chain**) will have **one or many hotels**
- Each **hotel** will have **one or many rooms**
- Each **hotel** will have **one or many employees**
- Each **hotel** will have **zero or many event reservations**
- Each **hotel** will have **zero or many customer feedback**
- Each **employee** will have **one department** they are working for
- Each **department** has **one or many employees**
- Each **customer** will have **zero or one membership**
- Each **customer** will have **zero or many room reservations**
- Each **customer** will **have one or many customer feedback** (for each stay)
- Each **event reservation** will only have **one event invoice**
- Each **event reservation** will have **only zero or more amenities**

Entity Relationship Diagram



Triggers

Customer Entity

Compute Age

Compute Age

Employee Entity

Membership Entity

Compute Membership Status

Update ChainID

Room Entity

Invoice Entity

Invoice Final Amount

Room Charge

Invoice Entity

Constraints

- Employee: Check Age if it is greater than 18
- Customer: Check Customer Age if it is greater than 18
- Event Reservation: Check Guest Number is less than 1,000
- Membership: Check Start Date is before End Date
- Amenity: Check Amenity Charge is greater than 0
- Room: Check Room Charge is greater than 0
- Room Reservation: Check Check-In Date is before Check-Out Date

View 1: Customer Revenue Details

CustomerRevenueDetails | Enter a SQL expression to filter results (use Ctrl+Space)

	Member_Revenue	Non_Member_Revenue	Offline_revenue	online_revenue	GOLD_revenue	Diamond_revenue	Silver_revenue	Platinum_revenue
1	37,603	3,008	21,815	18,796	2,006	21,372	14,225	0

```
Create View [dbo].[CustomerRevenueDetails] AS
SELECT SUM(CASE WHEN ISMEMBER = 1 THEN FINALAMOUNT END) AS Member_Revenue,
SUM(CASE WHEN ISMEMBER = 0 THEN FINALAMOUNT END) AS Non_Member_Revenue,
SUM(CASE WHEN ISBOOKEDONLINE = 0 THEN FINALAMOUNT END) AS Offline_revenue,
SUM(CASE WHEN ISBOOKEDONLINE = 1 THEN FINALAMOUNT END) AS online_revenue,
SUM(CASE WHEN MembershipSTATUS = 'Gold' THEN FinalAmount ELSE 0 END) AS GOLD_revenue,
SUM(CASE WHEN MembershipSTATUS = 'Diamond' THEN FinalAmount ELSE 0 END) AS Diamond_revenue,
SUM(CASE WHEN MembershipSTATUS = 'Silver' THEN FinalAmount ELSE 0 END) AS Silver_revenue,
SUM(CASE WHEN MembershipSTATUS = 'Platinum' THEN FinalAmount ELSE 0 END) AS Platinum_revenue
 FROM RoomReservation
LEFT JOIN
Customer
ON RoomReservation.CustomerID = Customer.CustomerID
LEFT JOIN
Membership
ON Customer.CustomerID = Membership.CustomerID
LEFT JOIN
Invoice
ON RoomReservation.ReservationID = Invoice.ReservationID;
```

View 2: Top Event Chains Details

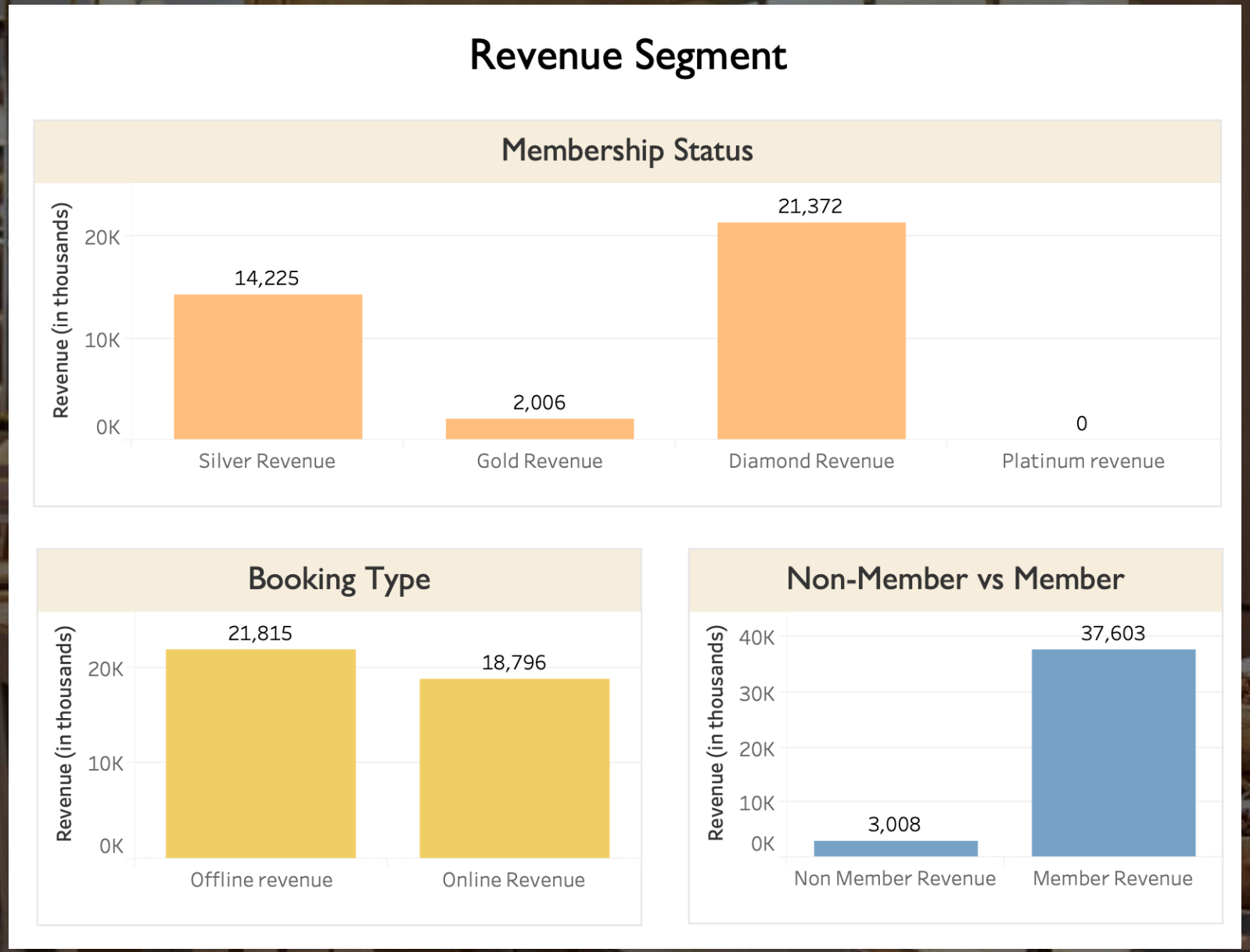
TopEventChainsDetails | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

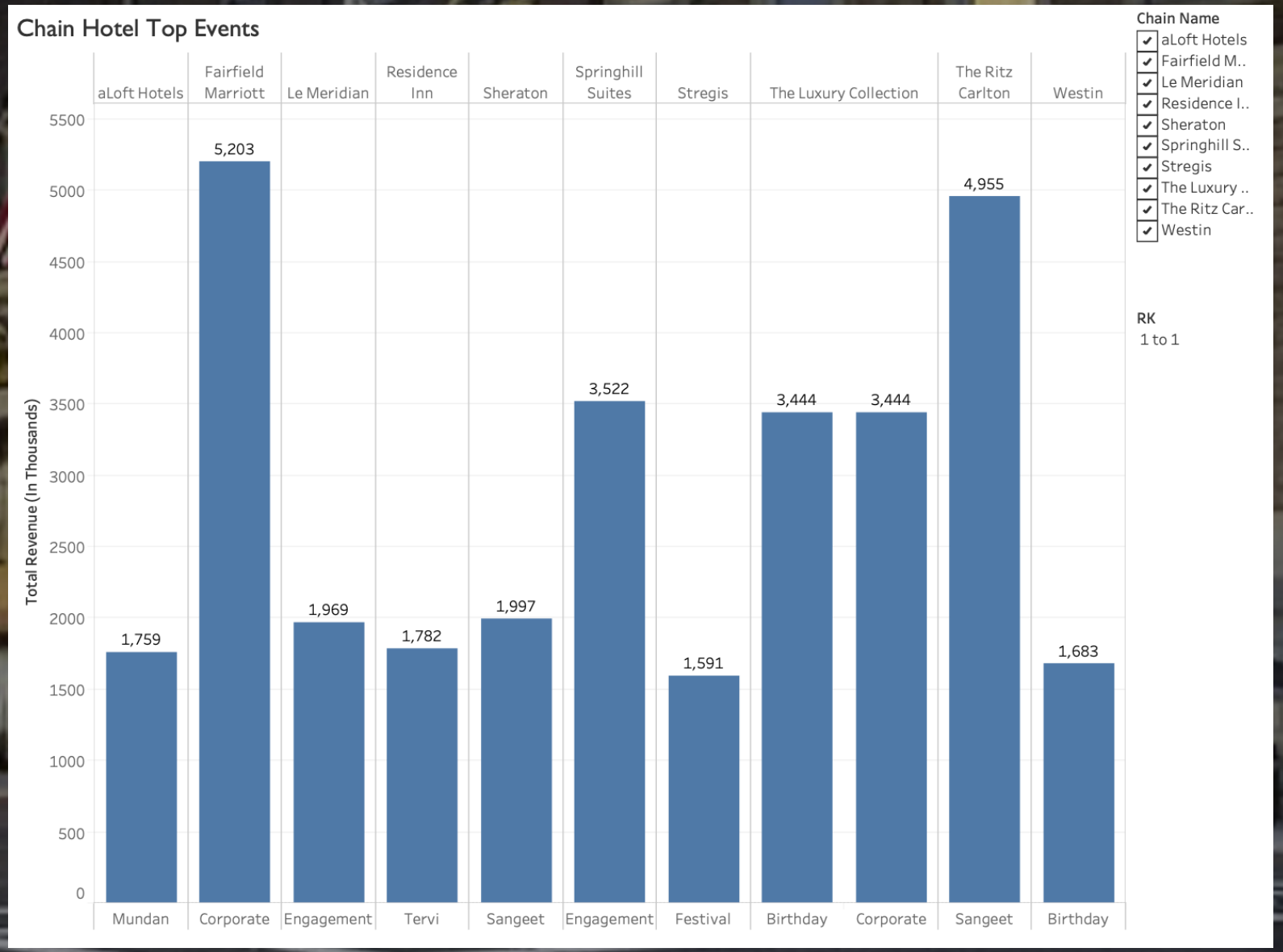
	ChainID	ChainName	ChainLevel	Eventtype	Total Revenue	Total Events
1	10,004	The Luxury Collection	Luxury	Birthday	3444.0000	2
2	10,012	Fairfield Marriott	Select	Corporate	5203.0000	3
3	10,015	Springhill Suites	Select	Engagement	3522.0000	2
4	10,014	The Ritz Carlton	Luxury	Festival	1978.0000	1
5	10,001	The Ritz Carlton	Luxury	Marriage	1903.0000	1
6	10,011	aLoft Hotels	Select	Mundan	1759.0000	1
7	10,001	The Ritz Carlton	Luxury	Sangeet	4955.0000	3
8	10,013	Residence Inn	Longer Stays	Tervi	1782.0000	1

```
Create View [dbo].[TopEventChainsDetails] AS
SELECT  ChainID,ChainName,ChainLevel,Eventtype,[Total Revenue],[Total Events] FROM
(SELECT Chains.ChainID,ChainName,ChainLevel,Eventtype,SUM(AMOUNT) as [Total Revenue], count(1) as [Total Events],
DENSE_RANK() OVER(PARTITION by Eventtype order by SUM(AMOUNT) desc ) AS RK FROM EventReservation
LEFT JOIN
EventInvoice
ON EventReservation.EventID = EventInvoice.EventID
LEFT JOIN
Chains
ON Chains.ChainID = EventReservation.ChainID
GROUP BY Chains.ChainID,Eventtype,ChainName,ChainLevel)c
WHERE RK = 1;
```

Visualization



Visualization



SQL Data Definition Language Statements

```sql
--9) Customer

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Customer](
    [CustomerID] [int] NOT NULL,
    [FirstName] [varchar](50) NULL,
    [LastName] [varchar](50) NULL,
    [Gender] [varchar](15) NULL,
    [PhoneNumber] [bigint] NULL,
    [EmailAddress] [varchar](500) NULL,
    [PassportNo] [varchar](50) NULL,
    [ZipCode] [int] NULL,
    [City] [varchar](50) NULL,
    [Country] [varchar](50) NULL,
    [isMember] [bit] NULL,
    [MembershipID] [int] NULL,
    [DATEOFBIRTH] [date] NULL,
    [CUSTAGE] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [CustomerID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Customer]  WITH CHECK ADD  CONSTRAINT [CheckCustomerAge] CHECK  ((datediff(year,[DATEOFBIRTH],getdate())>(18)))
GO
ALTER TABLE [dbo].[Customer] CHECK CONSTRAINT [CheckCustomerAge]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER [dbo].[ComputeCustAge]
ON [dbo].[Customer]
    AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    UPDATE Customer
    SET CustAge = (DATEDIFF(YEAR, DateOfBirth, GETDATE()))
    FROM Customer;
END;
GO
ALTER TABLE [dbo].[Customer] ENABLE TRIGGER [ComputeCustAge]
GO
```

```sql
---1) CHAINS

USE "007"

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Chains](
    [ChainID] [int] NOT NULL,
    [ChainName] [varchar](50) NOT NULL,
    [ChainLevel] [varchar](50) NULL,
    [ChainHeadOfficeAddress1] [varchar](500) NOT NULL,
    [ChainHeadOfficeCity] [varchar](50) NOT NULL,
    [ChainHeadOfficeCountry] [varchar](50) NOT NULL,
    [ChainHeadOfficeZipcode] [int] NULL,
    [ChainWebsite] [varchar](50) NOT NULL,
    [ChainHeadOfficePhoneNumber] [bigint] NULL,
PRIMARY KEY CLUSTERED
(
    [ChainID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

THANK YOU!