

Arab Academy for Science, Technology and Maritime Transport  
College of Computing & Information Technology  
Database Management Systems

# Domestic Tourism System

documentation file

|               |   |                   |
|---------------|---|-------------------|
| Supervised by | Dr. Mohamed Kholief<br>Eng. Mohamed Galal |                   |
| By            | Abdallah Mohamed Elsayed Aboudeif         | Reg. no. 19235004 |
|               | Mohamed Masoud Mohamed Ibrahim            | Reg. no. 20135025 |
|               | Hazem Magdy Hammad                        | Reg. no. 20135026 |

## Contents:

### 1 INTRODUCTION

### 2 Business logic

### 3 Roles summary

### 4 Constraints on business

### 5 Constrains on tables

### 6 Examples for created tables

### 7 Queries examples:

### 8 Database diagram

---

## 1 INTRODUCTION

While countries often tend to focus on international tourism due to the revenue earned through exports; domestic tourism remains the leading form of tourism, representing an important tool for regional economic growth and development.

With over 50% of the global population now categorized as “middle class”, more and more people can afford to travel. Research suggests that domestic tourism demand picks up at an income level lower than international travel.

Absolute size of Travel & Tourism’s domestic spending in 2017, domestic tourism represented 73% of the total global tourism spend (US\$3,971 billion). While there are significant variations between countries.

Egypt is one of the most prominent tourist countries in the world, it is distinguished by the abundance of tourist attractions of all kinds, the spread of temples, museums, monuments, historical and artistic buildings and vast gardens on its land, and its possession of strong infrastructure based on serving the tourism sector, including Hotel rooms, villages, tourist resorts, tourism companies and airline offices.

## 2 Business logic

Domestic tourism system logic stands for offering attractive short to medium travels that aims to spend vacations visiting one place or more inside one or more city, this structure of business requires collecting and storing tourists’ data, offering hostels and transportation for tourists, offering tourist guides for tourists’ groups to help them know about tourist places and coordinate their travels. Marketing is an important part of this business, attracting tourists is made by advertising on a travel at a known media beside to that it’s possible to deal with popular entities by giving their members a discount to encourage their clients or members have travels with our company. It is what we can call a Partner and it’s another important source for customers. Also, it is recommended to store hostel and transport owners and advertising companies’ data as a contact to continuously update their services prices and availability.

## 3 Roles summary

Business roles is roles can be summary in these points:

- Business goal is to make a profit by making travels inside the country
- Business has an information system consists of Microsoft SQL Server database and Java desktop application supported with JAVA-FX GUI
- Advertising companies make advertisements for our company
- Agents are responsible for working on system
- Agents publish advertisements for a travel on many types of media
- Agents store tourists and their partners on system as they have their information and update it
- Agents stores themselves, tourist guides, tourism places, advertisements and its companies, transports and their owners, hostels and their owners, campaigns
- Agents create travels and register tourism places, tourist guides, transports, campaigns, hostels and tourists in it
- Tourists have balance and it’s decreased by a travel price as they registered in it discounted by their partner discount (if any)

- Guides have a daily rate that calculated as a cost in every travel they registered on according to number of days they worked on
- Every campaign has a publish cost that calculated as a cost on its travel
- Advertisements have a design cost that doesn't calculated as a travel cost while one advertisement consider a business asset and it can be used in many travels for years
- Every hostel has its price and its updateable according to price changes, this price is calculated as a cost in every travel it registered on according to number of nights and rooms
- Transports have its price and its updateable according to price changes, this price is calculated as a cost in every travel they registered on according to number of days it used on
- Places have its ticket price, and it is calculated as a cost in every travel it registered on according to tourists' number
- Tourist Guides are responsible for guides tourists in every travel they registered on
- System should show a travel report showing travel data, registered members numbers, costs, profits and net profit

#### 4 Constraints on business

- Tourist, agent and guide cannot be registered on database more than one time
- It's mandatory to store all person's kind mobile numbers as it's the main way to connect with them
- Advertisements cannot be registered directly in a travel as it have a design cost that consider as on-time-buy and it could be published many times for one travel
- Travel elements cities should be known clearly (not a part of address) to help agents make local compatibility between travel elements
- Partners cannot have a discount more than 100% or less than 0%
- Instead of deleting temporary unused elements, elements have [active/idle] state to ignore idle elements registration on travels while its data still stored
- Travels are created with no elements, and it is eligible for have one or many elements in every type making it flexible to fit all domestic tourist travel types
- Total cost of every registered element is calculated and stored in element's registration table to not affected by future element's price changes

#### 5 Constrains on tables

- A travel period can be one day or more, so travel has start date and end date
- Service Provider table have a class attribute which stores the type of Service Provider's child and it make sure it's a type from its four children [hostel owner, transport owner, partner and advertising company]
- Service provider, Guide, Agent, Place and Tourist tables have a cityID column which make sure its number is an id for a city stored in City table
- Transport, Hostel, Tourist and Advertisement tables have its company id as a foreign key attribute stored in it, so it's easy to track its companies if any

- As most of system data is Arabic, database collation set to 'Arabic\_CI\_AI' and 'NCHAR' data type was used with the little English data
- Person's NID attribute made as unique to prevent person's multi-recording
- Person's mobile attribute made as unique to prevent person's mobile conflict
- Person's NID and mobile constraints as NOT NULL attributes
- Every database table have an automatic generated 'create Date' attribute which stores its date-time creation
- Most of repeated expected value and NOT NULL attributes have a default value ex. [idle, gender, cost, ...]
- Hotel degrees have a check constraint to make sure it is between 0 and 7
- Discount value fore partner must be between 0 and 1

## 6 Examples for created tables

- Tourist table is a good example, as its cod shows

```
CREATE TABLE Tourist
(
  id INT IDENTITY(1,1),
  prtID INT NULL,
  info NVARCHAR(MAX),
  balance MONEY NOT NULL DEFAULT 0,
  [name] VARCHAR(75) NOT NULL,
  NID CHAR(14) NOT NULL UNIQUE,
  gender BIT NOT NULL DEFAULT 0,
  mobile CHAR(11) NOT NULL UNIQUE,
  birthDate DATE NOT NULL,
  email NVARCHAR(75),
  cityID INT NULL,
  localAdd VARCHAR(100),
  creatDate SMALLDATETIME NOT NULL DEFAULT GETDATE(),
  idle BIT NOT NULL DEFAULT 0,
  CONSTRAINT FK_turPrtID FOREIGN KEY (prtID) REFERENCES ServiceProvider(id) ON DELETE SET NULL,
  CONSTRAINT FK_turCityID FOREIGN KEY (cityID) REFERENCES City(id) ON DELETE SET NULL,
  CONSTRAINT PK_Tourist PRIMARY KEY (id)
);
```

Tourist table is consisting of:

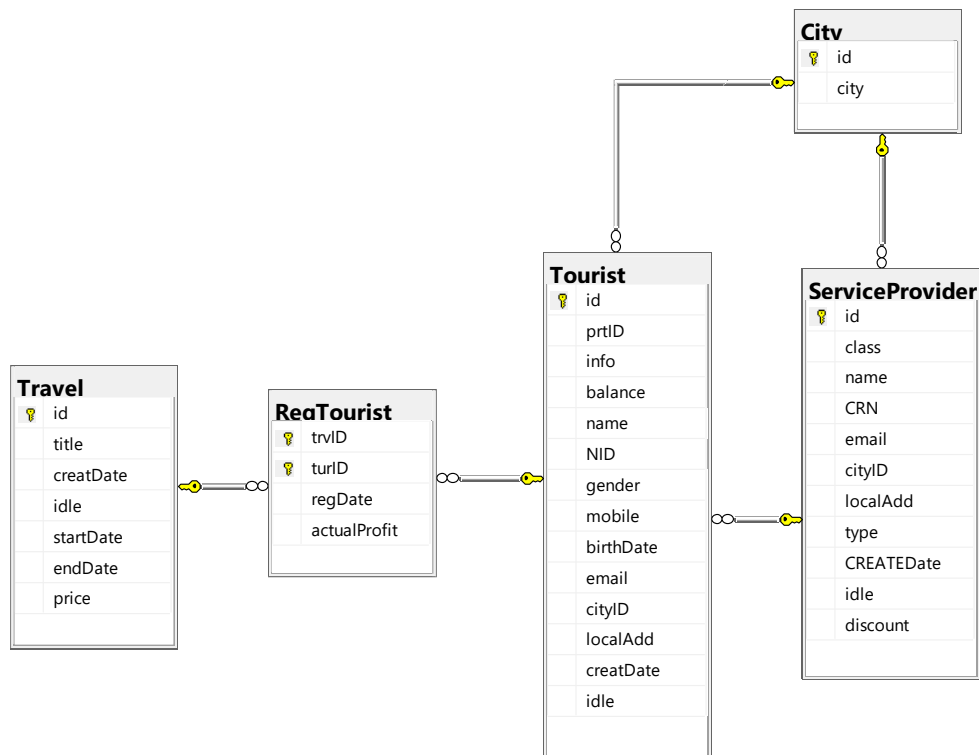
|    |           |                         |                           |                                     |
|----|-----------|-------------------------|---------------------------|-------------------------------------|
| 1  | id        | integer                 | Tourist unique identifier | Primary key / Auto increment        |
| 2  | prtID     | integer                 | Partner ID                | Foreign key for Partner primary key |
| 3  | info      | Unicode variable String | Additional information    | NULL                                |
| 4  | balance   | money                   | Tourist current balance   | NOT NULL / Default value is zero    |
| 5  | name      | variable String         | Tourist name              | NOT NULL                            |
| 6  | NID       | string                  | National ID number        | UNIQUE / NOT NULL / 14 characters   |
| 7  | gender    | bit                     | 0=male – 1=female         | NOT NULL / Default value is zero    |
| 8  | mobile    | string                  | National mobile number    | UNIQUE / NOT NULL / 11 characters   |
| 9  | birthDate | Date                    | Tourist date of birth     | NOT NULL                            |
| 10 | email     | Unicode variable String | Tourist email             | Max is 75 characters                |
| 11 | cityID    | Integer                 | Tourist city              | Foreign key for City primary key    |

|    |           |           |                           |                                  |
|----|-----------|-----------|---------------------------|----------------------------------|
| 12 | localAdd  | String    | Tourist address           | Max is 100 characters            |
| 13 | creatDate | Date-Time | Tourist registration date | NOT NULL / Auto generated        |
| 14 | idle      | bit       | 0=Active – 1=idle         | NOT NULL / Default value is zero |

As shown Tourist table has two one-to-many relations represented as a foreign keys in its table:

- 1- One to many relationship with City table
- 2- One to many relationship with Partner [ represented as Service Provider on schema ]

In addition to that Tourist has a many-to-many relationship with Travel table which doesn't appear in this table as it represented in RegTourist table which stores Tourist and Travel ids



- Another example is Advertisement table:

```

CREATE TABLE Advertisement
(
    id INT IDENTITY(1,1),
    companyID INT NULL,
    info VARCHAR(MAX),
    designCost MONEY NOT NULL DEFAULT 0,
    CREATEDate SMALLDATETIME NOT NULL DEFAULT GETDATE(),
    idle BIT NOT NULL DEFAULT 0,
    CONSTRAINT FK_adCompanyID FOREIGN KEY (companyID) REFERENCES ServiceProvider(id) ON DELETE SET NULL,
    CONSTRAINT PK_Advertisement PRIMARY KEY(id)
)
  
```

);

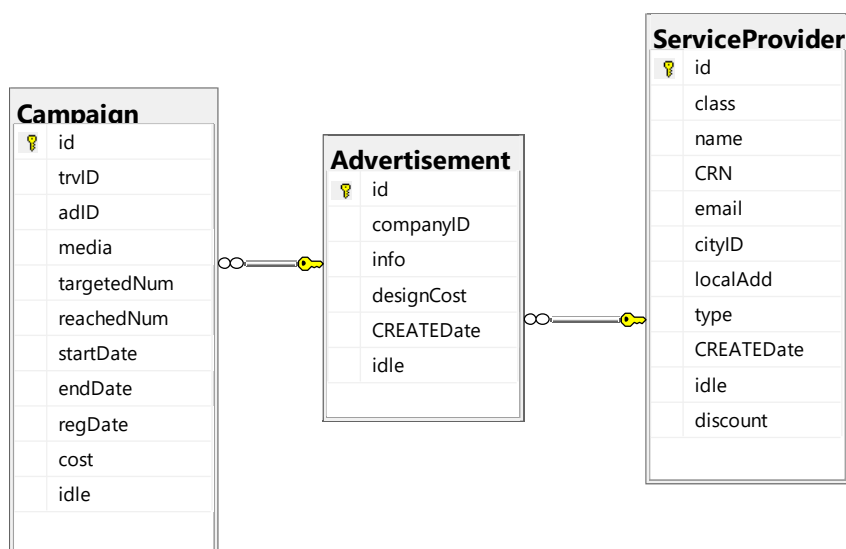
Advertisement table is consisting of:

|   |            |                         |                                     |                                     |
|---|------------|-------------------------|-------------------------------------|-------------------------------------|
| 1 | id         | integer                 | Unique identifier                   | Primary key / Auto increment        |
| 2 | companyID  | integer                 | Advertising company ID              | Foreign key for company primary key |
| 3 | info       | Unicode variable String | Information about the advertisement | NULL                                |
| 4 | designCost | money                   | Advertisement design cost           | NOT NULL / Default value is zero    |
| 5 | creatDate  | Date-Time               | Advertisement registration date     | NOT NULL / Auto generated           |
| 6 | idle       | bit                     | 0=Active – 1=idle                   | NOT NULL / Default value is zero    |

As shown Advertisement table has one one-to-many relationship represented as a foreign key in its table:

- 1- One to many relationship with Advertising Company [ represented as Service Provider on schema]

In addition to that Advertisement has a many-to-one relationship with Campaign table which doesn't appear in this table as it represented in Campaign table which stores Advertisement id



## 7 Queries examples:

- Advertisement table insertion table:

| Name       | Data type     | Nullable | Default   | Auto increment |
|------------|---------------|----------|-----------|----------------|
| id         | int           | False    |           | True           |
| companyID  | int           | True     |           | False          |
| info       | varchar(MAX)  | True     |           | False          |
| designCost | money         | False    | 0         | False          |
| CREATEDate | smalldatetime | False    | getdate() | False          |
| idle       | bit           | False    | 0         | False          |

Advertisement table has an auto increment primary key 'id' that doesn't need to insert manually, also there is CREATEDate attribute that doesn't need to insert manually since it has auto insert date-time the function getdate() as a default value

```
INSERT INTO Advertisement (companyID,info,designCost,idle)
VALUES (NULL,'1 إعلان شرم والغردقة',450,0);
```

It's possible to ignore 'idle' attribute since it have a default value 0 but we didn't in this example because application insertion query should enable creating both idle and active objects

- RegTourist table insertion and deletion query:

| Name         | Data type     | Nullable | Default   | Auto increment column |
|--------------|---------------|----------|-----------|-----------------------|
| trvID        | int           | False    |           | False                 |
| turID        | int           | False    |           | False                 |
| regDate      | smalldatetime | False    | getdate() | False                 |
| actualProfit | money         | False    | 0         | False                 |

Tourists are stored in Tourist table and Travels too have its table Travel, but when it comes to registering tourist in a travel, we found ourselves in front of Many-To-Many relationship that require making a separated table to represents these relationships.

Tourist has balance attribute of money datatype, and some tourists have a partner that has a discount deal with our company, if we want to register a tourist in a travel we should first check if the tourist balance is greater than or equal to travel price after discount if any. If tourist balance is enough then we must discount the actual price from tourist balance and add it to the actual price attribute in the registration record. If tourist balance is not enough we should return a message show why registration process failed.

The blow code is created to do that:

```
DECLARE @turprft AS MONEY = (SELECT price FROM Travel WHERE id = 1)-
ISNULL((SELECT discount FROM ServiceProvider WHERE id = (SELECT prtID FROM
Tourist WHERE id = 16)),0)*
(SELECT price FROM Travel WHERE id = 1)
IF (SELECT balance FROM Tourist WHERE id = 16) >= @turprft
BEGIN
INSERT INTO RegTourist (trvID,turID,actualProfit) VALUES (1,16,@turprft)
```



```
UPDATE Tourist SET balance = (balance-@turprft) WHERE id = 16
END
ELSE
SELECT 'Not enough balance'
```

The same thing should be followed when this registration cancel happen. First we should return the actual price to tourist balance then delete the registration record.

```
UPDATE Tourist SET balance = (balance+(SELECT actualProfit FROM RegTourist WHERE trvID = 1 AND
turID = 1)) WHERE id = 1
DELETE RegTourist WHERE trvID = 1 AND turID = 1
```

- Report SQL query:

The main goal of the system and business to make a profit in addition to monitoring travels process. This needs to make some queries to collect and show travels financial and numerical information using aggregation functions.

One problem should be considered is that prices and costs always going to change make report information changed continuously. The solution of this problem was to store total costs and actual prices in registration record so we can back to it to generate correct reports any time.

This query calculates the total cost of registered hostel for a travel:

```
-- التكلفة الإجمالية لأماكن الإقامة المحجوزة
SELECT SUM(totalCost) FROM RegHostel WHERE trvID = 2;
```

And this one find the number of registered tourists for a travel:

```
-- إجمالي عدد السياح المشتركين
SELECT COUNT(trvID) FROM RegTourist WHERE trvID = 2;
```

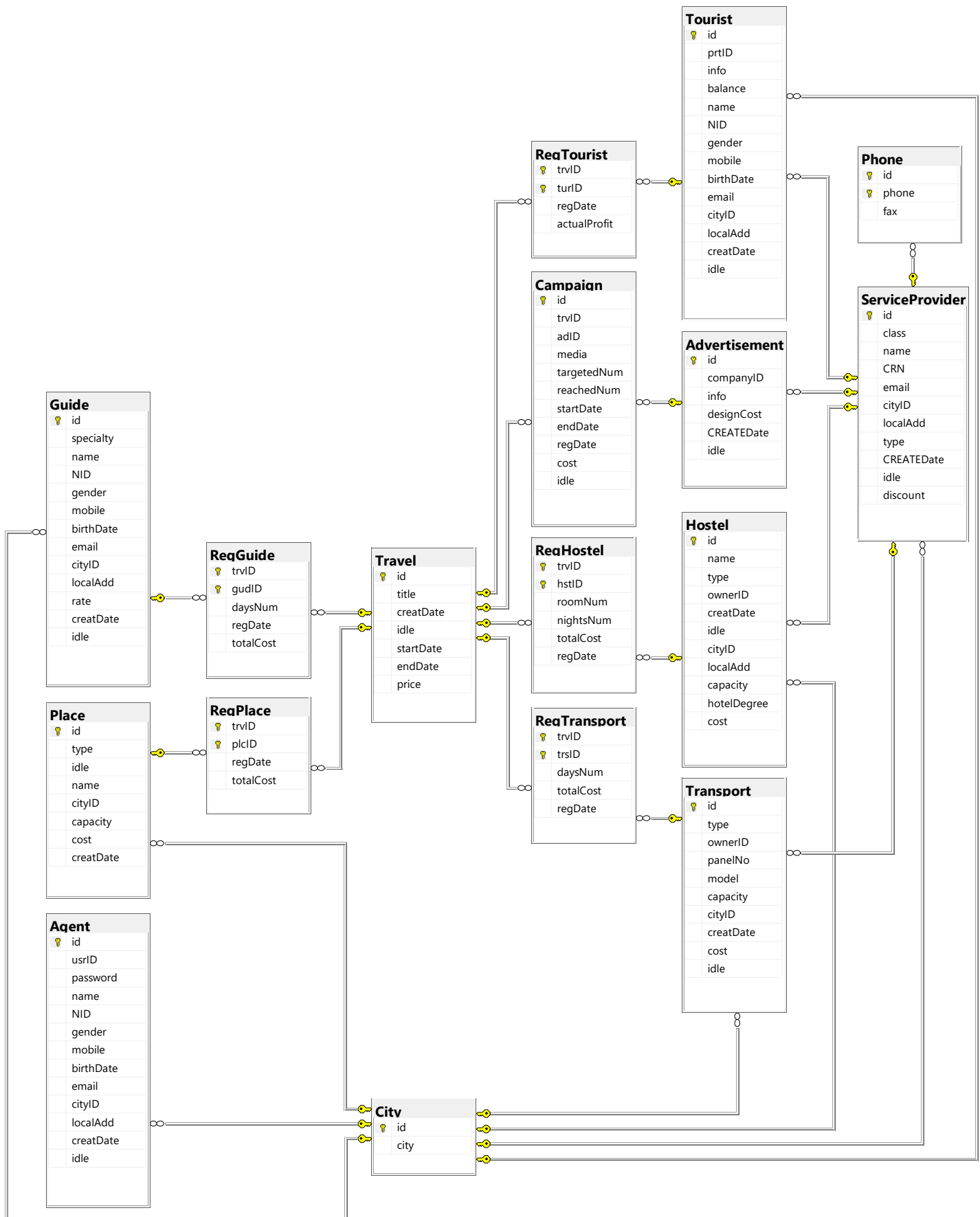
This one return the total costs of a travel:

```
-- إجمالي تكلفة الرحلة
DECLARE @id AS int = 2
DECLARE @cost AS MONEY = (SELECT SUM(totalCost) FROM RegTransport WHERE trvID = @id) +
    (SELECT SUM(totalCost) FROM RegHostel WHERE trvID = @id) +
    ((SELECT SUM(cost) FROM Place WHERE id in
        (SELECT plcID FROM RegPlace WHERE trvID = 2))*
    (SELECT COUNT(trvID) FROM RegTourist WHERE trvID = 2)) +
    (SELECT SUM(totalCost) FROM RegGuide WHERE trvID = @id) +
    (SELECT SUM(cost) FROM Campaign WHERE trvID = @id)

SELECT @cost
```

## 8 Database diagram

## 1- Entity Relationship diagram [Schema]



2- Entity Relationship diagram [Logical]

