|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 12 | https://www.toursbylocals.com | | | getDate() |
| חברי הצוות - מספרי ת.ז | | | | |
| 123456789 | | 123456789 | 123456789 | |

## פרויקט בסיסי נתונים – חלק ג'

## תרשים ERD

**התרשים בחלק ב':**

## 

## תרשים מתוקן

**Of a**

**0:N**

* DT
* Rating
* Free text

**Review**

**Response**

* ResponseDT
* FreeTxt
* ID
* First name
* Last name
* First Tour On
* [Language]

**Guide**

**0:1**

**0:N**

**0:N**

**In charge of**

**1:N**

**1:1**

* Tour Num.
* Tour name
* Country
* Area
* [Tour type]
* Currency
* Price
* Activity level

**Tour**

**0:N**

**Contact**

* DT
* Text
* Date Of Tour
* Num. Of People

**0:N**

**0:N**

**0:N**

**wish**

* DT
* Already travelled

**1:N**

**0:N**

**0:N**

**1:1**

**search**

**0:N**

* Email
* User Password
* Name
* Address - Country
* Address - Street
* Address - Number
* Address - City
* Address - Zip Code

**Customer**

**search**

**0:1**

**0:N**

**Runs**

**0:N**

**0:N**

* IP Address
* Search DT
* Date- from
* Date- to
* Country
* [Filter]

**Search Virtual Tour**

* Next tour voucher
* Voucher Currency

**Virtual Tour**

**Details**

* Start Date
* End Date
* **Num. Of Tourists**

**Purchases**

**0:N**

**1:1**

**0:N**

* Order Num.
* Order DT
* Price
* Currency
* Num. of tourists

**Order**

**1:N**

**1:1**

* CC Num.
* Type
* CVV
* Expiry date

**Credit Card**

* Area
* Tour name
* Type

**Search Tour**

**Paid with**

|  |  |
| --- | --- |
| **מרכיב** | **הנחות** |
| הקשר בין "Search" ל- "Costumer" | חיפושים יכולים להתבצע על ידי משתמש מזדמן (0) או לקוח רשום (1). משתמש יכול לא לחפש כלום ועד N חיפושים. |
| שדה RATING בעשות ביקורת | לכל ביקורת נותנים דירוג. הדירוג הזה אינו מחושב ולכן נמצא בERD. דירוג מחושב לכל מדריך או טיול לא נמצא בERD. |
| הוספת שדה Type בישות “Search Tour” | כלל השדות הנכללים בישות זו, נכללים גם כאשר לקוח מעוניין לבצע חיפוש לפי מדריך ולא טיול. לכן שדה זה יקבע האם מדובר בחיפוש טיול או חיפוש מדריך. |
| ישות “Review” היא ישות חלשה של “Customer” | לא ניתן לכתוב ביקורת אם אתה לא לקוח רשום באתר. |
| הישות “Virtual Tour” נורש על ידי “Tour” | טיול וירטואלי הוא הרחבה של טיול, מכיל את כל השדות של טיול ובנוסף שובר הנחה לטיול הבא. |
| הישות “Search Tour” נורש על ידי “Search Virtual Tour” | חיפוש טיול (או מדריך- בהתאם לType) מכיל את השדות שיש לחיפוש טיול וירטואלי ומרחיב אותו עם שדות נוספים, ולכן יורש אותו. |
| הקשר בין ישות ביקורת לישות מדריך | בין ישויות אלו שני קשרים, אחד מתאר את כמות הביקורת שאפשר לתת על כל מדריך והשני את תגובות המדריך לביקורות. |
| Wish list בישות לקוח | זוהי רשימה של העדפות של טיולים. לכל טיול יש קשר להזמנות שבוצעו אליו, ולהזמנה יש קשר ללקוח יחיד שביצע אותה. לכן, הלקוח מקושר |
| הקשר RESPONSE בין ישות ביקורת למדריך | הנחנו כי מדריך יכול להגיב מספר פעמים לכל ביקורת ולכל מדריך יכולות להיות הרבה ביקורת. לכן הקשר הוא רבים לרבים. |

# **עיצוב לוגי מתוקן**

* **TOURS** (TourNum ,TourName ,Country ,Area ,Price , Currency , ActivityLevel ,ID(GUIDES))
* **TOURTYPES** (TourNum(TOURS) , TourType)
* **VIRTUALTOURS**(TourNum (TOURS),NextTourVoucher, VoucherCurrency )
* **GUIDES**(ID, FirstName , LastName , FirstTourON)
* **LANGUAGES** (ID(GUIDES), LanguageName)
* **CUSTOMERS**(Email , Password , Name , Country , City , Street , StreetNum ,ZipCode )
* **SEARCHVIRTUALS**(IP , DT ,FromDate , ToDate , Country , Email(CUSTOMERS))
* **FILTERS**({IP , DT}( SEARCHVIRTUALS), Filter)
* **ORDERS**(OrderNum , OrderDT, Price ,Currency , Email(CUSTOMERS) , CCN(CREDITCARDS))
* **CREDITCARDS**(CCN , CreditCardType , CVV , ExpiryDate )
* **SEARCHTOURS**({IP , DT}( SEARCHVIRTUALS) , Area , TourName , TourType)
* **REVIEWS**(Email(CUSTOMERS) , DT , Rating ,FreeText , OfTour(TOURS))
* **GUIDESRESPONSES**({ReviewDT, Email} (REVIEWS), GuideID(GUIDES), ResponseDT, FreeTxt
* **WISHES**(Email(CUSTOMERS), TourNum (TOURS) , DT, Alredy\_Travelled)
* **Details**(TourNum (TOURS) , OrderNum(ORDERS) , TouristsNum , StartDate , EndDate )
* **TOURSSEARCHES**({IP, DT}( SEARCHVIRTUALS), TourNum(TOURS))
* **GUIDESSEARCHES**({IP, DT}( SEARCHVIRTUALS), ID(GUIDES))
* **CONTACTS**(ID(GUIDES) , Email(CUSTOMER)) , DT, Text , DateOfTour , NumOfPeople)

בהתאם להערות שניתנו תיקנו את מודל הטבלאות אחרי שתיקנו את מודל הERD שלנו בהתאם .

התיקונים מוצגים בעמוד הבא.

**הצגת התיקונים**

* **TOURS** (TourNum ,TourName ,Country ,Area ,Price ,ActivityLevel , Currency ,ID(GUIDES))
* **TOURTYPES** (TourNum(TOURS) , TourType)
* **LANGUAGES** (~~TourNum (TOURS)~~ ID(GUIDES) LanguageName)
* **VIRTUALTOURS**(TourNum (TOURS),NextTourVoucher , VoucherCurrency )
* **CUSTOMERS**(Email , Password , Name , Country , City , Street , StreetNum ,ZipCode )
* **GUIDES**(ID, FirstName , LastName , FirstTourON, ~~Rating, NumberOfReviews, Contact~~)
* **ORDERS**(OrderNum , OrderDT, Price , TouristsNum , Email(CUSTOMERS) , ~~{CCN,Type}(CREDITCARDS)~~ CCN(CREDITCARDS), Currency )
* **CREDITCARDS**(CCN , ~~CreditCardType~~ CreditCardType, CVV , ExpiryDate )
* **SEARCHVIRTUALS**(IP , DT ,FromDate , ToDate,Country , Email(CUSTOMERS))
* **SEARCHTOURS**({IP , DT}( SEARCHVIRTUALS),Area , TourName,TourType)
* **REVIEWS**(Email(CUSTOMERS) , ReviewDT , Rating, FreeTxt , ~~GuideResponse~~ ~~, ID(GUIDES)~~ , OfTour(TOURS))
* **~~TOURSFILTERS~~**~~({IP, DT}( SEARCHVIRTUALS), TourNum(TOURS) ,FilterType ,FilterContent)~~
* **TOURSSEARCHES**({IP, DT}( SEARCHVIRTUALS), TourNum(TOURS))
* **~~GUIDESFILTERS~~**~~({IP, DT}( SEARCHVIRTUALS),ID(GUIDES),FilterType ,FilterContent)~~
* **GUIDESSEARCHES**({IP, DT}( SEARCHVIRTUALS), ID(GUIDES))
* **WISHES**( Email(CUSTOMERS) , TourNum(TOURS), DT, Already\_Travelled)
* **ORDEREDTOURS**(TourNum(TOURS), OrderNum(ORDERS) , StartDate , EndDate)

שלוש טבלאות נוספות:

* **FILTERS**({IP , DT}( SEARCHVIRTUALS), Filter)
* **CONTACTS**(ID(GUIDES) , Email(CUSTOMER)) , DT, Text , DateOfTour , NumOfPeople
* **GUIDESRESPONSES**({ReviewDT, Email} (REVIEWS), GuideID(GUIDES), ResponseDT, FreeTxt

# **פירוט שדות בסיס הנתונים והערכת גודלם**

|  |  |  |  |
| --- | --- | --- | --- |
| **Size (Bytes)** | **Data Type** | **Field** | **Table** |
| 4 | int | TourNum | **TOURS** |
| 202 | varchar(200) | TourName |  |
| 52 | varchar(50) | Country |  |
| 52 | varchar(50) | Area |  |
| 4 | smallmoney | Price |  |
| 5 | varchar(5) | Currency |  |
| 52 | varchar(50) | ActivityLevel |  |
| 4 | int | GuideID(GUIDES) |  |
| **375** |  | **Total Record Size (Bytes)** |  |
| 4 | int | TourNum(TOURS) | **TOURTYPES** |
| 52 | varchar(50) | TourType |  |
| **56** |  | **Total Record Size (Bytes)** |  |
| 4 | int | TourNum (TOURS) | **VIRTUALTOURS** |
| 4 | smallmoney | NextTourVoucher |  |
| 7 | varchar(5) | VoucherCurrency |  |
| **15** |  | **Total Record Size (Bytes)** |  |
| 4 | int | ID | **GUIDES** |
| 52 | varchar(50) | FirstName |  |
| 52 | varchar(50) | LastName |  |
| 3 | Date | FirstTourOn |  |
| **111** |  | **Total Record Size (Bytes)** |  |
| 4 | int | GuideID(GUIDES) | **LANGUAGES** |
| 52 | varchar(50) | LanguageName |  |
| **56** |  | **Total Record Size (Bytes)** |  |
| 52 | varchar(50) | Email | **CUSTOMERS** |
| 52 | varchar(50) | UserPassword |  |
| 52 | varchar(50) | Name |  |
| 52 | varchar(50) | Country |  |
| 52 | Varchar(50) | City |  |
| 52 | varchar(50) | Street |  |
| 4 | int | StreetNum |  |
| 22 | Varchar(20) | ZipCode |  |
| **338** |  | **Total Record Size (Bytes)** |  |
| 52 | Varchar(50) | IP | **SEARCHVIRTUALS** |
| 8 | DateTime | DT |  |
| 3 | Date | FromDate |  |
| 3 | Date | ToDate |  |
| 52 | varchar(50) | Country |  |
| 52 | varchar(50) | Email(CUSTOMERS) |  |
| **170** |  | **Total Record Size (Bytes)** |  |
|  |  |  |  |
| 52 | Varchar(50) | IP(SEARCHVIRTUALS) | **FILTERS** |
| 8 | DateTime | DT(SEARCHVIRTUALS) |  |
| 52 | Varchar(50) | FilterType |  |
| 52 | Varchar(50) | FilterContent |  |
| **164** |  | **Total Record Size (Bytes)** |  |
| 4 | int | OrderNum | **ORDERS** |
| 8 | DateTime | OrderDT |  |
| 4 | smallmoney | Price |  |
| 7 | Varchar(5) | Currency |  |
| 1 | tinyint | TouristsNum |  |
| 52 | varchar(50) | Email(CUSTOMERS) |  |
| 18 | Varchar(16) | CCN(CREDITCARDS) |  |
| **94** |  | **Total Record Size (Bytes)** |  |
| 18 | Varchar(16) | CCN | **CREDITCARDS** |
| 52 | Varchar(50) | CreditCardType |  |
| 5 | Varchar(3) | CVV |  |
| 3 | Date | ExpiryDate |  |
| **78** |  | **Total Record Size (Bytes)** |  |
| 52 | Varchar(50) | IP(SEARCHVIRTUALS) | **SEARCHTOURS** |
| 8 | DateTime | DT(SEARCHVIRTUALS) |  |
| 52 | varchar(50) | Area |  |
| 52 | varchar(50) | TourName |  |
| 52 | varchar(50) | TourType |  |
| **216** |  | **Total Record Size (Bytes)** |  |
| 52 | varchar(50) | Email(CUSTOMERS) | **REVIEWS** |
| 8 | DateTime | ReviewDT |  |
| 4 | Real | Rating |  |
| 2002 | Varchar(2000) | FreeTxt |  |
| 4 | int | OfTour(TOURS) |  |
| **2070** |  | **Total Record Size (Bytes)** |  |
| 8 | DateTime | ReviewDT (REVIEWS) | **GUIDESRESPONSES** |
| 52 | varchar(50) | Email(CUSTOMERS) |
| 4 | int | GuideID(GUIDE) |
| 8 | DateTime | ResponseDT |
| 2002 | Varchar(2000) | FreeTxt |
| 2074 |  | **Total Record Size (Bytes)** |
| 52 | varchar(50) | Email(CUSTOMERS) | **WISHES** |
| 4 | int | TourNum(TOURS) |  |
| 8 | DateTime | DT |  |
| 7 | VarChar(5) | Already\_Travelled |  |
| **71** |  | **Total Record Size (Bytes)** |  |
| 4 | int | TourNum(TOURS) | **ORDEREDTOURS** |
| 4 | int | OrderNum(ORDERS) |  |
| 3 | Date | StartDate |  |
| 3 | Date | EndDate |  |
| **14** |  | **Total Record Size (Bytes)** |  |
| 50 | varchar(50) | IP(SEARCHVIRTUALS) | **TOURSSEARCHES** |
| 8 | DateTime | DT(SEARCHVIRTUALS) |  |
| 4 | int | TourNum(TOURS) |  |
| **62** |  | **Total Record Size (Bytes)** |  |
| 52 | varchar(50) | IP(SEARCHVIRTUALS) | **GUIDESSEARCHES** |
| 8 | DateTime | DT(SEARCHVIRTUALS) |  |
| 4 | int | GuideID(GUIDES) |  |
| **64** |  | **Total Record Size (Bytes)** |  |
| 4 | Int | GuideID(GUIDES) | **CONTACS** |
| 52 | Varchar(50) | Email(CUSTOMERS) |  |
| 8 | DateTime | DT |  |
| 2002 | Varchar(2000) | CustomerMassage |  |
| 3 | Date | DateOfTour |  |
| 1 | tinyInt | NumOfPeople |  |
| **2070** |  | **Total Record Size (Bytes)** |  |

בניית הטבלאות :

create table GUIDES(

ID int not null,

FirstName varchar(50) null,

LastName varchar(50) null,

FirstTourOn date default getDate(),

constraint Guides\_Pk Primary key(ID)

)

create table TOURS (

TourNum int not null,

TourName varchar(200) not null,

Country varchar(50) not null,

Area varchar(50) not null,

Currency varchar(5) default 'USD',

Price SmallMoney null default 0,

ActivityLevel varchar(50) not null,

GuideID int not null,

constraint Price\_Pos check(price>=0 ),

constraint Tours\_Pk Primary key(TourNum),

constraint Tours\_Fk FOREIGN KEY (GuideID) REFERENCES GUIDES (ID)

)

create table TOURTYPES(

TourNum int not null,

TourType varchar(50) Not null,

constraint TOURTYPES\_Pk Primary key(TourNum, TourType),

constraint TOURTYPES\_Fk FOREIGN KEY (TourNum) REFERENCES TOURS (TourNum)

)

create table LANGUAGES(

GuideId int not null,

LanguageName varchar(50) not null,

constraint LANGUAGES\_Pk Primary key(GuideId,LanguageName),

constraint LANGUAGES\_Fk FOREIGN KEY (GuideId) REFERENCES Guides (id)

)

create table VIRTUALTOURS(

TourNum int not null,

NextTourVoucher smallmoney,

VoucherCurrency varchar(5),

constraint VIRTUALTOURS\_Pk Primary key(TourNum),

constraint VIRTUALTOURS\_Fk FOREIGN KEY (TourNum) REFERENCES TOURS ( TourNum)

)

CREATE TABLE Customers (

Email Varchar(50) NOT NULL,

UserPassword Varchar(50) NOT NULL,

Name Varchar(50) NOT NULL,

Country varchar(50) NOT NULL,

City Varchar(50) NOT NULL,

Street varchar(50) NOT NULL,

StreetNum int NOT NULL,

ZipCode varchar(20),

CONSTRAINT pk\_CUSTOMERS PRIMARY KEY (Email),

CONSTRAINT ck\_EMAIL CHECK (Email LIKE '%@%.%'),

)

CREATE TABLE CreditCards (

CCN varchar(16) NOT NULL,

CreditCardType Varchar(50) NOT NULL,

CVV varchar (3) NOT NULL,

ExpiryDate Date NOT NULL,

CONSTRAINT pk\_CreditCards PRIMARY KEY (CCN),

CONSTRAINT ck\_ExpiryDate CHECK (ExpiryDate > getDate())

)

CREATE TABLE Orders (

OrderNum int NOT NULL,

OrderDT DateTime,

Price SmallMoney DEFAULT 0,

Currency varchar(5) default 'usd',

Email varchar(50) NOT NULL,

CCN varchar(16) NOT NULL,

CONSTRAINT pk\_Orders PRIMARY KEY (OrderNum),

CONSTRAINT fk\_Customers FOREIGN KEY (Email) REFERENCES Customers (Email),

CONSTRAINT fk\_CreditCards FOREIGN KEY (CCN) REFERENCES CreditCards (CCN),

CONSTRAINT ck\_NonNegativityPrice CHECK (Price>=0)

)

create table WISHES(

Email varchar(50) not null,

TourNum int not null,

DT dateTime,

Alredy\_Travelled VarChar (5) Default 'NO'

CONSTRAINT pk\_WISHES primary key(Email,TourNum),

CONSTRAINT fk\_WISHES1 FOREIGN KEY (Email) REFERENCES Customers (Email),

CONSTRAINT fk\_WISHES2 FOREIGN KEY ( TourNum) REFERENCES TOURS ( TourNum)

)

create table Details(

OrderNum int not null,

TourNum int not null,

TouristsNum Tinyint DEFAULT 1,

startdate date,

endDate date

CONSTRAINT pk\_ORDERDTOURS primary key(OrderNum,TourNum),

CONSTRAINT fk\_ORDERDTOURS1 FOREIGN KEY (OrderNum) REFERENCES ORDERS (OrderNum),

CONSTRAINT fk\_ORDERDTOURS2 FOREIGN KEY ( TourNum) REFERENCES TOURS ( TourNum)

)

create table SEARCHVIRTUALS(

IP Varchar(50) not null,

DT DateTime not null,

FromDate Date ,

ToDate Date,

Country varchar(50),

Email varchar(50),

constraint Ip\_Good CHECK (IP LIKE '%.%.%.%'),

CONSTRAINT pk\_SEARCHVIRTUALS PRIMARY KEY (IP,DT),

CONSTRAINT fk\_SEARCHVIRTUALS FOREIGN KEY (Email) REFERENCES Customers (Email)

)

create table SEARCHTOURS(

IP Varchar(50) not null,

DT DateTime not null,

Area varchar(50),

TourName varchar(50),

TourType varchar(50),

CONSTRAINT pk\_SEARCHTOURS PRIMARY KEY (IP,DT),

CONSTRAINT fk\_SEARCHTOURS FOREIGN KEY (IP,DT) REFERENCES SEARCHVIRTUALS (IP,DT)

)

create table ToursSearches(

IP varchar(50) not null,

DT DateTime not null,

TourNum int not null

constraint TOURSFILTERS\_PK Primary key(TourNum,IP,DT),

constraint TOURSFILTERS\_Fk1 FOREIGN KEY ( TourNum) REFERENCES TOURS ( TourNum),

constraint TOURSFILTERS\_Fk2 FOREIGN KEY ( IP,DT) REFERENCES SEARCHVIRTUALS ( IP,DT)

)

create table GuidesSearches(

IP varchar(50) not null,

DT DateTime not null,

GuideID int not null

CONSTRAINT pk\_GUIDESFILTERS PRIMARY KEY (IP,DT,GuideID),

CONSTRAINT fk\_GUIDESFILTERS1 FOREIGN KEY (IP,DT) REFERENCES SEARCHVIRTUALS (IP,DT),

CONSTRAINT fk\_GUIDESFILTERS2 FOREIGN KEY ( GuideID) REFERENCES GUIDES ( ID)

)

create Table Filters(

IP varchar(50) not null,

DT DateTime not null,

Filter varChar (100) Default 'ANY',

CONSTRAINT pk\_filters PRIMARY KEY (IP,DT,Filter),

CONSTRAINT fk\_filters1 FOREIGN KEY (IP,DT) REFERENCES SEARCHVIRTUALS (IP,DT)

)

create table REVIEWS(

Email varchar(50) not null,

ReviewDT DateTime ,

Rating real ,

FreeTxt varchar(2000),

OfTour int,

constraint Rating\_lim check(Rating>=0 and Rating<=5),

CONSTRAINT Pk\_REVIEWS PRIMARY KEY (Email,ReviewDT),

CONSTRAINT fk\_REVIEWS1 FOREIGN KEY (Email) REFERENCES Customers(Email) ,

CONSTRAINT fk\_REVIEWS2 FOREIGN KEY ( ofTour) REFERENCES TOURS ( TourNum),

)

create table GUIDESRESPONES(

Email varchar(50) not null,

ReviewDT dateTime not null,

GuideId int not null,

ResponeDT dateTime not null,

FreeTxt VarChar(2000)

constraint Pk\_GuidesRespones primary key( ReviewDT ,GuideId ,ResponeDT),

CONSTRAINT fk\_GUIDESRESPONES1 FOREIGN KEY (Email,ReviewDT) REFERENCES REVIEWS (EMAIL,ReviewDT) ,

CONSTRAINT fk\_GUIDESRESPONES2 FOREIGN KEY ( GuideId) REFERENCES guides (id)

)

create table Contacts(

GuideID int not null,

Email varchar(50) not null,

DT dateTime not null,

CustomerMassage varchar(2000),

DateOfTour date ,

NumOfPeople tinyint

constraint Pk\_Contacts primary key( Email , GuideID ,DT)

CONSTRAINT fk\_Contacts1 FOREIGN KEY (Email) REFERENCES CUSTOMERS (Email) ,

CONSTRAINT fk\_Contacts2 FOREIGN KEY ( GuideId) REFERENCES guides ( id)

)

CREATE TABLE COUNTRIES(

Country varChar(50) Not null,

constraint pk\_country primary Key (Country),

)

ALTER TABLE TOURS

ADD CONSTRAINT fk\_countries FOREIGN key(Country) references COUNTRIES (Country)

CREATE TABLE LANGUAGESLIST(

LANGUAGENAME varChar(50) Not null,

constraint pk\_LANGUAGES primary Key (LANGUAGENAME),

)

ALTER TABLE Languages

ADD CONSTRAINT fk\_Languages FOREIGN key(LanguageName) references LANGUAGESLIST (LANGUAGENAME)

drop table Contacts

drop table GUIDESRESPONES

drop table REVIEWS

drop table Filters

drop table GuidesSearches

drop table ToursSearches

drop table SEARCHTOURS

drop table SEARCHVIRTUALS

drop table Details

drop table WISHES

drop table Orders

drop table CreditCards

drop table VIRTUALTOURS

drop table LANGUAGES

drop table TOURTYPES

drop table TOURS

drop table GUIDES

drop table Customers

drop table LANGUAGESLIST

drop table COUNTRIES

## מטלה 1 – שאילתות

# שתי שאילתות SELECT ללא קינון

|  |  |
| --- | --- |
| **שאילתה מספר 1** | |
| **היגיון עסקי** | לחברה ישנן מטרות עסקיות ברורות לכל שנה. היא מגדירה את המדינות הרווחיות שלה על ידי חישוב סך מחיר ההזמנות של כל מדינה בשנתיים האחרונות. מדינות רווחיות הן מדינות שסך מחיר ההזמנות שלהן בשנתיים האחרונות גדול מ-1500. לכן, שאילתא זו מחזירה את המדינות שעומדות בתנאי זה, ומסדרת אותן על פי הרווחיות שלהן. |
| **SQL** | select t.Country, Count (\*) as [Tours ordered], [Total orders price]=SUM(o.Price)  from tours as t join details as d on t.TourNum=d.TourNum join orders as o on o.OrderNum=d.OrderNum  where year(GETDATE())-year(d.startdate)<2  group by t.Country  having SUM(o.Price)>1500  order by 3 desc |
| **פלט** |  |

|  |  |
| --- | --- |
| **שאילתה מספר 2** | |
| **היגיון עסקי** |  |
| **SQL** |  |
| **פלט** |  |

# שתי שאילתות SELECT מקוננות

|  |  |
| --- | --- |
| **שאילתה מספר 3** | |
| **היגיון עסקי** | מתוך כלל הטיולים, שאילתא זו מחזירה רשימה (LIST) של כל הטיולים שלא הוזמנו מאז תחילת הקורונה, כלומר מחודש מרץ של שנת 2019. |
| **SQL** | Select TourNum  From Tours  Where TourNum NOT IN (  Select distinct D.TourNum  From Orders as O Join Details as D on O.OrderNum=D.OrderNum  Where YEAR (OrderDT) >= 2019 AND MONTH (OrderDT) >= 3) |
| **פלט** | **סה"כ 510 רשומות** |

|  |  |
| --- | --- |
| **שאילתה מספר 4** | |
| **היגיון עסקי** | השאילתא מראה את הלקוחות שהזמינו טיולים וירטואליים ואת סך מחיר השוברים שקיבלו מכל ההזמנות, מסודרים לפי כמות הטיולים שהוזמנו על ידי כל לקוח. לקוחות אלו מקבלים שובר הנחה מכל הזמנה כך נוכל לדעת כמה החברה חייבת לכל לקוח. |
| **SQL** | Select C.Name, C.Email, C.Country, Sum ([Tours per order]) as [Number of tours], Sum ([Total Voucher]) as [Total voucher per customer]  From Customers as C Join Orders as O on C.Email=O.Email Join (  Select D.OrderNum, Count (D.TourNum) as [Tours per order], Sum (V.NextTourVoucher) as [Total Voucher]  From VirtualTours as V Join Details as D on V.TourNum=D.TourNum  Group By D.OrderNum) as [Tours per order] on O.OrderNum=[Tours per order].OrderNum  Group By C.Name, C.Email, C.Country  Order By Sum ([Tours per order]) DESC |
| **פלט** | **סה"כ 67 רשומות** |

# שאילתות מקוננות עם מרכיבים נוספים

|  |  |
| --- | --- |
| **שאילתה מספר 5** | |
| **היגיון עסקי** | החברה מעוניינת לבדוק את יעילות פילטור החיפוש. שאילתא זו יוצרת עמודה חדשה בטבלת FILTERS בשם STATUS ומעדכנת אותה בהתאם לכמות הפעמים שכל פילטר שומש. |
| **SQL** | Alter Table Filters  Add Status Varchar (40) Default 'Weak'  Update Filters  Set Filters.Status = status\_update.status  from(  Select Filter, CASE  When count (\*) > 10 THEN 'Popular'  When count (\*) < 5 THEN 'Weak'  Else 'Regular'  End as Status  From Filters  Group By Filter) as status\_update Join Filters on status\_update.Filter=Filters.Filter |
| **פלט** |  |

|  |  |
| --- | --- |
| **שאילתה מספר 6** | |
| **היגיון עסקי** | שאילתא זו מבצעת חיתוך (INTERSECT) בין הלקוחות שביצעו יותר הזמנות מממוצע ההזמנות ללקוח עם הלקוחות שביצעו יותר חיפושים מכמות החיפושים הממוצעת ללקוח. לקוחות אלו פוטנציאליים להיות לקוחות מובילים בחברה. |
| **SQL** | Select Email From (  (Select Email, count (O.OrderNum) as [Total orders]  From Orders as O  Group By Email) as [Orders per Customer] cross Join (  Select Average = (Count (\*)/Count(distinct Email))  From Orders) as [Average orders] )  Where [Total orders] > Average    INTERSECT  Select Email  From (  (Select Email, count (\*) as [Num of searches]  From SEARCHVIRTUALS  Group By Email ) as [Top Searchers] cross Join (  Select Average = (Count (\*)/Count(distinct Email))  From SEARCHVIRTUALS) as [Average searches] )  Where [Num of searches] > Average |
| **פלט** |  |

## מטלה 2 - יישומי כלים מתקדמים

View

|  |  |
| --- | --- |
| **VIEW** | |
| **היגיון עסקי** | הטבלה מיועדת למנהלי החשבונות של החברה אשר מנהלים את כל ענייני התשלום של הלקוחות. הטבלה מציגה את נתוני ההזמנה, פרטי הלקוח המזמין ואת פרטי כרטיס האשראי באמצעותו בוצעה ההזמנה.  הערה- במימוש פעולה זו הגדרנו להראות רק לקוחות שמחיר הזמנתם גבוה מ1500. |
| **SQL** | drop view PaymentsDetails\_view  CREATE VIEW PaymentsDetails\_view as  SELECT O.OrderDT , o.OrderNum , O.Email, CU.Name , CC.CCN , CC.CreditCardType , CC.CVV , CC.ExpiryDate ,  o.Price  FROM CreditCards as CC join Orders as O on CC.CCN=O.CCN  join Customers as CU on O.Email=CU.Email |
| **הפעלה** | Select \*  From PaymentsDetails\_view  Where PaymentsDetails\_view.Price > 1500  Order By PaymentsDetails\_view.Price DESC |
| **מימוש** |  |

# פונקציות

|  |  |
| --- | --- |
| **SCALAR FUNCTION** | |
| **היגיון עסקי** | הלקוחות יכולים לתת ביקורת על מדריכי הטיולים ובתוך הביקורת לתת דירוג בין 0 ל- 5.  הפונקציה מקבלת בפלט מזהה של מדריך ומחזירה את הדירוג הסופי שלו שהוא ממוצע כלל הדירוגים של כלל הלקוחות. |
| **SQL** | CREATE FUNCTION getGuideAvgReview ( @ID int )  RETURNS Real  AS BEGIN  DECLARE @Score Real    SELECT @Score = avg (r.Rating)  FROM reviews as r join tours as t on r.OfTour=t.TourNum join guides as g on g.ID=t.GuideID  WHERE g.ID = @ID  RETURN @Score  END |
| **הפעלה** | select g.FirstName, g.ID,[Average review]=dbo.getGuideAvgReview(g.ID)  from GUIDES as g  order by 3 des |
| **מימוש** | **10 המדריכים בעלי הדירוג המירבי** |

# 

|  |  |
| --- | --- |
| **TABULAR FUNCTION** | |
| **היגיון עסקי** | פונקציה המקבלת כקלט שם של שפה, ומחזירה טבלה שבה כל המדריכים שדוברים את השפה הזו. הפלט הוא מזהה המדריך והשם המלא שלו. |
| **SQL** | drop function Guides\_languge  CREATE FUNCTION Guides\_languge ( @Language varchar (40))  RETURNS TABLE  AS RETURN  SELECT [Full name]=g.FirstName+' '+g.LastName,g.ID  FROM GUIDES as g join LANGUAGES as l on l.GuideId=g.ID  WHERE l.LanguageName=@Language |
| **הפעלה** | select [Full name],[ID]  from Guides\_languge('English') |
| **מימוש** |  |

SP- פרוצדורה שמורה

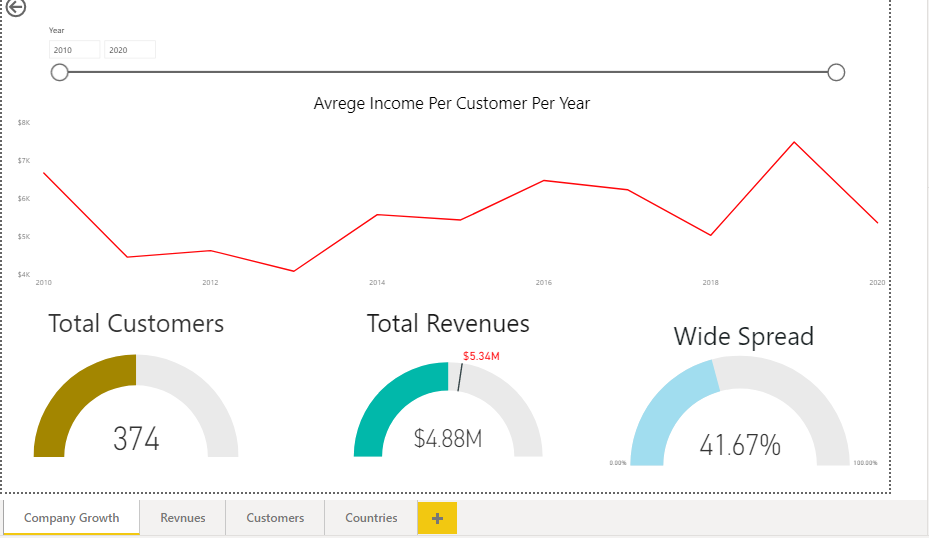
|  |  |
| --- | --- |
| **SP** | |
| **היגיון עסקי** | פרוצדורה אשר מקבלת כתובת מייל של לקוח, ומציגה את שם הלקוח אשר מקושר למייל ואת כל הטיולים שהזמין, מחירם ותאריך ההזמנה.  נרצה להשתמש בפרוצדורה שמורה, מכיוון שבטבלת הזמנות ישנם פרטים של כרטיס האשראי ממנו ביצע הלקוח את ההזמנה ועל כן לא נרצה שעובדים בחברה, אשר יבצעו את החיפוש, יהיו חשופים למידע זה. |
| **SQL** | CREATE PROCEDURE SP\_TripBY\_CustomerInfo @Email varchar(50)  as  SELECT c.Name ,t.TourNum,o.Price,d.startdate,d.endDate  FROM CUSTOMERS as c join Orders as o on c.Email=o.Email join Details as d on d.OrderNum=o.OrderNum join tours as t on t.TourNum=d.TourNum  WHERE c.Email = @Email |
| **הפעלה** | EXECUTE SP\_TripBY\_CustomerInfo '9PE0X6OFWIZ5T9X@KUSA0AHL.5SCWAF' |
| **מימוש** |  |

TRIGGER

|  |  |
| --- | --- |
| **Trigger** | |
| **היגיון עסקי** | TRIGGER זה נועד לעדכן את מחיר ההזמנה בטבלת הזמנות בכל פעם שמעדכנים/מוסיפים/מוחקים רשומה בטבלת DETAILS. הטריגר סוכם את סכום ההזמנה החדש לאחר השינויים ומעדכן בטבלה. |
| **SQL** | CREATE TRIGGER Update\_Tota  ON Details  FOR INSERT, UPDATE, DELETE  AS  UPDATE ORDERS  SET Price = (  SELECT Sum (T.Price)  FROM Details as D JOIN Tours as T  ON D.TourNum = T.TourNum  WHERE D.OrderNum = Orders.OrderNum  )  WHERE OrderNum IN (  SELECT DISTINCT OrderNum from INSERTED  UNION  SELECT DISTINCT OrderNum from DELETED  ) |
| **הפעלה** | Insert into Details Values (10020, 102138, 12, '2014-10-09', '2014-10-11') |
| **מימוש** | לפני:    אחרי: |

## מטלה 3 - כלים להצגת נתונים

# דו"ח עסקי



## מטלה 4 - יישומים מתקדמים

דו"ח המושתת על שאילתה מקוננת מורכבת

|  |  |
| --- | --- |
| **דו"ח המושתת על שאילתה מקוננת מורכבת** | |
| **היגיון עסקי** | דו"ח זה מציג בצורה מפורטת את תפוקתם של מדריכי החברה מרגע שהתחילו לעבוד בחברה ועד היום. הדו"ח נעזר בשתי טבלאות VIEW שנועדו להציג את מספר הלקוחות שחיפשו מדריך מסוים, מספר הביקורת שקיבל מלקוחות ואת הדירוג שלו מבלי לפגוע ברגישות המידע ובפרטיות הלקוחות. |
| **בניית פונקציית עזר לחישוב דירוג של מדריך** | --פונקציה שמחשבת דירוג של מדריך  CREATE FUNCTION getGuideAvgReview ( @ID int )  RETURNS Real  AS BEGIN  DECLARE @Score Real    SELECT @Score = avg (r.Rating)  FROM reviews as r join tours as t on r.OfTour=t.TourNum join guides as g on g.ID=t.GuideID  WHERE g.ID = @ID  RETURN @Score  END |
| **בניית 2 טבלאות VIEWS כעזר להצגת הנתונים** | --הצגת מספר הלקוחות שחיפשו מדריך מסוים  CREATE VIEW V\_GuidesSearches as  SELECT G.GuideID, count (distinct S.Email) as Searches  From SEARCHVIRTUALS as S Join GuidesSearches as G on S.IP = G.IP AND S.DT = G.DT  Group By G.GuideID  -- הצגת מספר הביקורת שקיבל מדריך ואת הדירוג שלו  CREATE VIEW V\_GuidesReviews as  Select T.GuideID, count (\*) as [Number of reviews], Rating = dbo.getGuideAvgReview(T.GuideID)  From REVIEWS as R Join Tours as T on R.OfTour = T.TourNum  Group By T.GuideID |
| **הפעלה** | SELECT  Guide = G.ID,  Name = G.FirstName+ ' ' + G.LastName,  [Time in firm in days] = DATEDIFF (dd, G.FirstTourON, getDate() ),  Rating = GR.Rating,  [Revieus] = GR.[Number of reviews],  Languages = (Select count (L.LanguageName)  From LANGUAGES as L  Where L.GuideId=G.ID  Group By L.GuideId) ,  Tours = (Select count (T.TourNum)  From Tours as T  Where T.GuideID=G.ID  Group By T.GuideID),  Searches = GS.Searches  FROM V\_GuidesReviews as GR Join  Guides as G on GR.GuideID = G.ID JOIN  V\_GuidesSearches AS GS ON G.ID = GS.GuideID  Order By GR.Rating DESC |
| **פלט** |  |

שילוב מערכתי של מספר כלים

|  |  |
| --- | --- |
| **Function+View+SP** | |
| **היגיון עסקי** | החברה שלנו היא כלל עולמית ולכן צריכה להתאים את מחירי הטיולים על פי המדינה בה מתרחש הטיול. השילוב המערכתי שיצרנו עוזר לה להבין את המחירים הממוצעים של טיולים בכל שנה וכך לדעת להתאים את המחיר הנכון לכל טיול. בנוסף היא יכולה לראות שוני במחירים ביחס לשנים קודמות ולהתעדכן בהתאם. |
| **בניית פונקציית עזר לחישוב מחיר ממוצע של טיול במדינה מסוימת בשנה מסוימת** | CREATE FUNCTION F\_AVG\_PRICE\_PER\_YEAR\_PER\_COUNTRY( @COUNTRY VARCHAR(50),@YEAR VARCHAR(4) )  RETURNS Real  AS BEGIN  DECLARE @PRICE Real    SELECT @PRICE = avg (o.Price)  FROM TOURS as t join Details as d on t.TourNum=d.TourNum join Orders as o on o.OrderNum=d.OrderNum  WHERE @COUNTRY=t.Country and year(d.startdate)=@year  RETURN @PRICE  END |
| **בניית טבלת VIEW אשר מציגה את המדינות בהן הזמינו טיול בשנה מסוימת** | create view V\_YEAR AS  SELECT DISTINCT YEAR=YEAR(D.startdate),t.Country  FROM Details AS D join TOURS as t on t.TourNum=d.TourNum |
| **SP פרוצדורה שמורה לה מכניסים מדינה והיא מחזירה שנים בהן טיילו בה ואת המחיר הממוצע לאותה שנה** | --drop PROCEDURE SP\_AVG\_TOURS\_PRICE\_PERCOUNTRY  CREATE PROCEDURE SP\_AVG\_TOURS\_PRICE\_PERCOUNTRY @COUNTRY VARCHAR(50)  as  SELECT YEAR=v.YEAR, [Average price per tour]=dbo.F\_AVG\_PRICE\_PER\_YEAR\_PER\_COUNTRY(@COUNTRY,v.YEAR)  FROM V\_YEAR as v  where v.Country=@COUNTRY |
| **הפעלה לדוגמא עבור בלגיה** | EXECUTE SP\_AVG\_TOURS\_PRICE\_PERCOUNTRY 'Belgium' |
| **פלט** |  |