Roms Booking

Data Dictionary

2021-01-22

KODLUYORUZ
geleceği kodluyoruz >_

Table of Contents

Legend


Primary key

Primary key disabled


User-defined primary key


Unique key


Unique key disabled


User-defined unique key


Active trigger


Disabled trigger


Primary key relation


User-defined primary key relation


Foreign key relation


User-defined foreign key relation


Input


Output


Input/Output

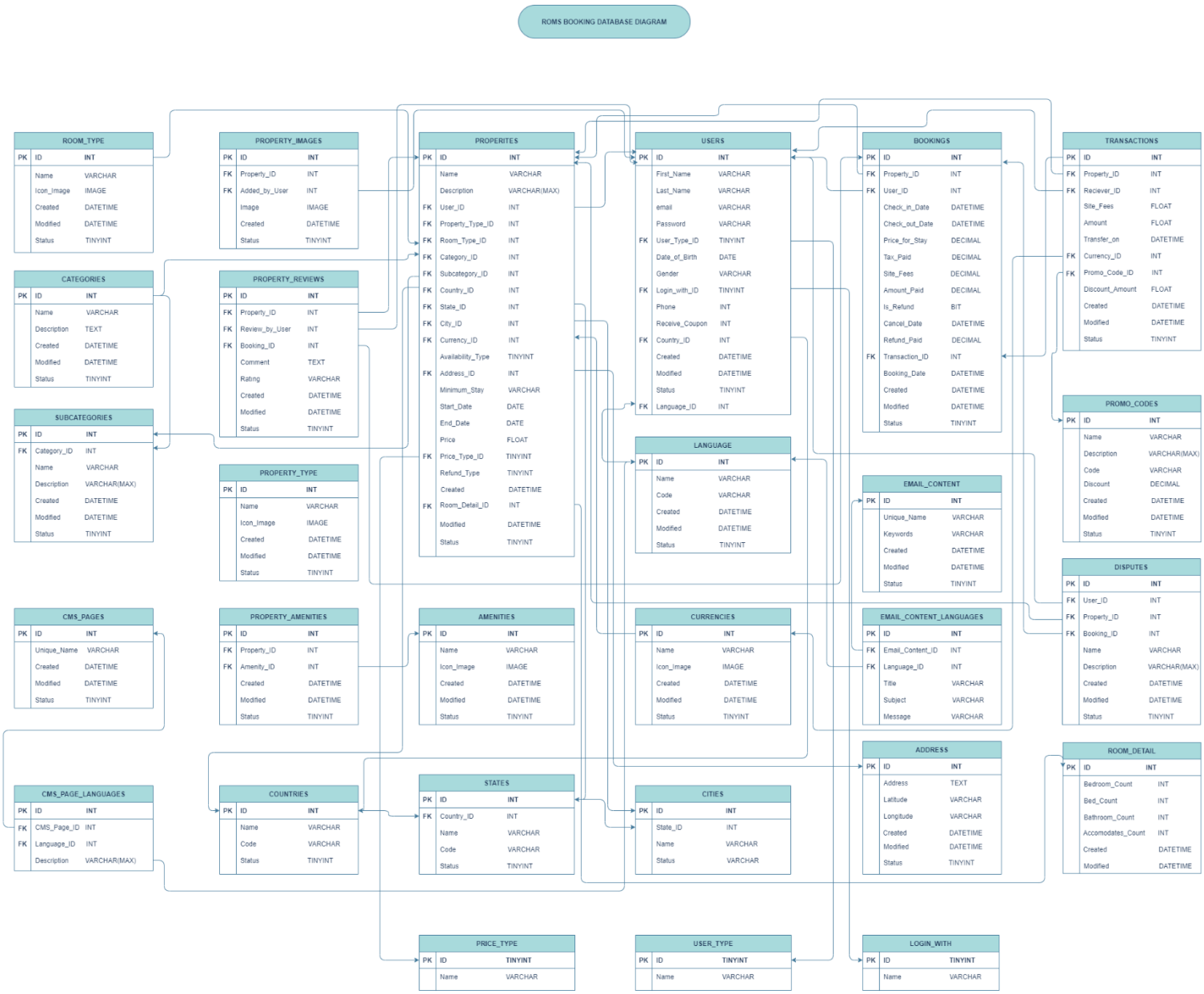

Nullable

1. Roms Bookings

This documentation is – Roms Booking- Microsoft SQL Server database. The Roms Booking database is organized with reference to the Airbnb site. Airbnb Booking database has been designed by taking into account the data needed by customers during their booking process.

This database is monitored by Melek Laçin, Özge Özcan, Rukiye Demirci and Şule Kaya.

2. Booking Entity Relationships



3. Booking

3.1. Tables

3.1.1. Table: Address

Detailed address information of properties are stored at Address table.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for address records. Identity/Auto increment column
	Address	NVARCHAR(500)		Address details
	Latitude	NVARCHAR(200)		Latitude of the address
	Longitude	NVARCHAR(200)		Longitude of the address
	Created	DATETIME		Created date and time
	Modified	DATETIME		Modified date and time
	Status	STATUS		Status of the address

Links to

	Table	Join	Title/Name/Description
	Properties	Properties.Address_ID=Address_ID	FK_ Address_ID Foreign key constraint referencing Address.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
Properties

3.1.2. Table: Amenities

It contains information about Amenities.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for amenities records.
	Name	NVARCHAR(200)		Amenity Name
	Icon_Image	VARCHAR(MAX)		Amenity icon
	Created	DATETIME		Date Created pated system
	Modified	DATETIME		The date when Amenity renewed or corrected the system
	Status	TINYINT		Status of the Amenity

Links to

	Table	Join	Title/Name/Description
	Property_Amenities	PROPERTY_AMENITIES.AMENITY_ID=AMENTIES.ID	FK_Property_ Amenties.AMENITY_ID Foreign key constraint referencing AMENTIES.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
 AMENTIES
 Property_Amenities

3.1.3. Table: Bookings

The user's booking information UserID,amount paid,booking date is stored Bookings table.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for bookings records. Identity/Auto increment column
	Property_ID	INT		Property information Foreign key to Properties table
	UserID	INT		User information ID Foreign key to Users table
	Check_In_Date	DATETIME		Check in date
	Check_Out_Date	DATETIME		Check out date
	Price_For_Stay	FLOAT		Price for stay
	Tax_Paıd	FLOAT		Tax
	Site_Fees	FLOAT		Site fees
	Amound_Paid	FLOAT		Total payment
	Is_Refund	BIT		Refers to refund 1=No refund,0=Yes refund.
	Cancel_Date	DATETIME		Refers to whether user has any discount coupons 1=Yes,0=No
	Refund_Paid	FLOAT		Refers to the amount to be repaid.
	Transaction_ID	INT		Transaction information ID Foreign key to Transactions table
	Booking_Date	DATETIME		Booking date
	Created	DATETIME		Date bookings pated users
	Modified	DATETIME		The date when booking renewed or corrected the users
	Status	TINYINT		Status of the booking

Linked From

	Table	Join	Title/Name/Description
	Propeties	Bookings.Property_ID=Property.ID	FK_ Bookings.Property_ID Foreign key constraint referencing Property.ID
	Users	Bookings_UserID=Users.ID	FK_ Bookings_UserID Foreign key constraint referencing Users.ID
	Transactions	Bookings.Transaction_ID=Transactions.ID	FK_ Bookings.Transaction_ID Foreign key constraint referencing Transactions.ID

Links to

	Table	Join	Title/Name/Description
	Properties	Bookings.ID=Property_Reviews.Booking_ID	FK Property_Reviews.Booking_ID Foreign key constraint referencing Bookings.ID
	Disputes	Bookings.ID = Disputes.Booking_ID	FK_ Disputes.Booking_ID Foreign key constraint referencing Bookings.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
Bookings
Property_Reviews
Disputes

3.1.4. Table: Categories

Keeping information of category types for properties and subcategories table

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for categories records Identity/ Auto increment column
	NAME	NVARCHAR(200)	N	Category name
	DESCRIPTION	NVARCHAR(MAX)	N	Defining categories
	CREATED	DATETIME	N	The creation date of categories
	MODIFIED	DATETIME	N	Update date ıf categories
	STATUS	TINYINT		Categories status

Linked From

	Table	Join	Title/Name/Description
	Subcategories	Categories.ID = Subcategories.Category_ID	FK_Subcategories_Categories_Category_ID Foreign key constraint referencing Categories.ID
	Properties	Categories.ID = Properties.Category_ID	FK_Properties_Categories_Category_ID Foreign key constraint referencing Categories.ID

Unique Keys

	Columns	Name/Description
	ID	PK_Categories_ID Primary key(clustered) constraint

Used By

Name
Categories
Subcategories
Properties

3.1.5. Table: Cities

It contains information about cities

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for cities records. Identity/Auto increment column
	State_ID	INT		State_ID Foreign key to States table
	Name	NVARCHAR(200)		C,ty
	Status	TINYINT		Status of the city

Linked From

	Table	Join	Title/Name/Description
	States	Cities.State_ID=States. ID	FK_ Cities.State_ID Foreign key constraint referencing States. ID

Links to

	Table	Join	Title/Name/Description
	Properties	Cities. ID=Properties.City_ID	FK_ Properties.City_ID Foreign key constraint referencing Cities. ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
Cities
Properties

3.1.6. Table: Cms Page Languages

It is the table that keeps the information in which languages the CMS pages are available.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for CMS Pages Language records. Identity/Auto increment column
	CMS_PAGE_ID	INT		Unique identification number for CMS pages types. Foreign key to CMS Page table
	LANGUAGE_ID	INT		Unique identification number for language on the CMS pages.

				Foreign key to Language table
	DESCRIPTION	NVARCHAR(MAX)	N	Defining to CMS pages

Links To

	Table	Join	Title/Name/Description
	CMS_PAGES	CMS_Page_Language.CMS_ Page_ID = CMS_Page.ID	FK_CMS_Page_Language_CMS_Page_ID_CMS_Page_ID Foreign key constraint referencing CMS_Pages.ID
	LANGUAGE	CMS_Page_Language. Language_ID = Language.ID	FK_ CMS_Page_Language_Language_ID Foreign key constraint referencing Language.ID

Unique Keys

	Columns	Name/Description
	ID	PK_CMS_Page_Language_ID Primary key (clustered) index

Uses

Name
CMS_Page_Language
CMS_Page
Language

3.1.7. Table: Cms Pages

It is a tablet hat keeps information about content management pages

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for CMS Pages records. Identity/Auto increment column
	NAME	NVARCHAR(200)	N	Cms pages the name
	CREATED	DATETIME	N	The names of the types of CMS pages
	MODIFIED	DATETIME	N	Date of creation of CMS pages
	STATUS	TINYINT		Indicates of status of CMS pages

Linked From

	Table	Join	Title/Name/Description
	CMS_Page_Language	CMS_Pages.ID = CMS_Pages_Language_ CMS_Page_ID	FK_CMS_Page_Language_CMS_Page_CMS _Page_ID Foreign key constraint referencing CMS_Pages.ID

Unique Keys

	Columns	Name/Description
	ID	PK_CMS_Pages_ID Primary key (clustered) constraint

Used By

Name
CMS_Pages
CMS_Page_Language

3.1.8. Table: Countries

It contains information about countries.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for cities records. Identity/Auto increment column
	Name	NVARCHAR(200)		Country Name
	Code	NVARCHAR(10)		Country Code
	Status	TINYINT		Status of the country

Links to

	Table	Join	Title/Name/Description
	STATES	STATES.COUNTRY_ID=COUNTRIES.ID	FK_States.COUNTRY_ID Foreign key constraint referencing COUNTRIES.ID
	USERS	USERS.COUNTRY_ID=COUNTRIES.ID	FK_USERS.COUNTRY_ID Foreign key constraint referencing COUNTRIES.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
COUNTRIES
STATES
USERS

3.1.9. Table: Currencies

It contains information about currencies.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for currencies records. Identity/Auto increment column
	Name	NVARCHAR(200)		Currency name
	Icon_Image	VARCHAR(MAX)		Currency icon
	Created	DATETIME		Date currency pated system
	Modified	DATETIME		The date when currency renewed or corrected the system
	Status	TINYINT		Status of the currency

Links to

	Table	Join	Title/Name/Description
	Properties	Currencies.ID=Properties.Currency_ID	FK_ Properties.Currency_ID Foreign key constraint referencing Currencies.ID
	Transactions	Currencies.ID = Transactions.Currency_ID	FK_ Transactions.Currency_ID Foreign key constraint referencing Currencies.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
Currencies
Properties
Transactions

3.1.10. Table: Disputes

Details of disputes of the properties are stored at Disputes table.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for disputes records. Identity/Auto increment column
	USERID	INT		USERID Foreign key to Users table
	PROPERTY_ID	INT		Property ID Foreign Key to Properties table
	BOOKING_ID	INT		Booking ID Foreign Key to Bookings table
	NAME	NVARCHAR(200)		Short description for dispute
	DESCRIPTION	NVARCHAR(MAX)		Description of dispute
	CREATED	DATETIME		Created date and time
	MODIFIED	DATETIME		Modified date and time
	STATUS	TINYINT		Status of the dispute

Linked from

	Table	Join	Title/Name/Description
	Properties	Disputes.Property_ID=Properties.ID	FK Disputes.Property_ID Foreign key constraint referencing Properties.ID
	Users	Disputes.USERID=Users.ID	FK_Disputes.USERID Foreign key constraint referencing Users.ID
	Bookings	Disputes.Booking_ID=Bookings.ID	FK_ Disputes.Booking_ID Foreign key constraint referencing Bookings.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

3.1.11. Table: Email Content

It contains information about emails.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for email content records. Identity/Auto increment column
	Unique_Name	NVARCHAR(200)		Email content name
	Keywords	NVARCHAR(200)		Email keywords
	Created	DATETIME		Date email contents pated system
	Modified	DATETIME		The date when email contents renewed or corrected the system
	Status	TINYINT		Status of the email content

Links to

	Table	Join	Title/Name/Description
	Email_Content_Languages	Email_Content.ID= Email_Content_Languages.Email_Content_ID	FK_Email_Content_Languages.Email_Content_ID Foreign key constraint referencing Email_Content.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
 Email_Content
 Email_Content_Languages

3.1.12. Table: Email Content Languages

It contains information about cities

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for email records. Identity/Auto increment column
	Email_content_id	INT		Email content id, Foreign Key to Email content table
	Language_id	INT		Language id, Foreign Key to Language table
	Title	NVARCHAR(100)		Title of the email
	Subject	NVARCHAR(100)		Subject of the email
	Message	NVARCHAR(500)		Message

Linked From

	Table	Join	Title/Name/Description
	Email_content	Email_content_ID=Email_content.ID	FK_ Email_content_ID Foreign key constraint referencing Email_content. ID
	Language	Email_content_language.Language_id=Language.id	FK_Language_ID Foreign key constraint referencing Language.id

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

3.1.13. Table: Language

Information on languages spoken in the countries and in-system software languages can be found in the Language table.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for language records. Identity/Auto increment column
	Name	NVARCHAR(200)		Language name
	Code	NVARCHAR(20)		Refers to the abbreviated name of the language.
	Created	DATETIME		Date language pated system
	Modified	DATETIME		The date when language renewed or corrected the system
	Status	TINYINT		Status of the language

Links to

	Table	Join	Title/Name/Description
	Users	Language.ID=Users.Language_ID	FK_Users.Language_ID Foreign key constraint referencing Language.ID
	Email_Content_Languages	Language.ID = Email_Content_Languages. Language_ID	FK_ Email_Content_Languages. Language_ID Foreign key constraint referencing Language.ID
	Cms_Page_Language	Language.ID = Cms_Page_Language. Language_ID	FK_ Cms_Page_Language. Language_ID Foreign key constraint referencing Language.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
Language
Users
Email_Content_Languages
Cms_Page_Languages

3.1.14. Table: Login With

Information that the keeps track of which tools users log in

Columns

	Name	Data Type	N	Description/Attributes
	ID	TINYINT		Primary key for price type records. Identity/Auto increment column
	NAME	NVARCHAR(50)		Names of log in

Linked From

	Table	Join	Title/Name/Description
	Users	Log_With.ID = Users.Log_With_ID	FK_Users_Log_With_ID Foreign key constraint referencing Log_With.ID

Unique Keys

	Columns	Name/Description
	ID	PK_Log_With_ID Primary key (clustered) constraint

Used By

Name
 Log_With
 Users

3.1.15. Table: Price Type

It is a table contains information about payment types

Columns

	Name	Data Type	N	Description/Attributes
	ID	TINYINT		Primary key for price type records. Identity/Auto increment column
	NAME	NVARCHAR(50)		Names of payment types

Linked From

	Table	Join	Title/Name/Description
	Properties	Price_Type.ID = Properties.Price_Type_ID	FK_Properties_Price_Type_ID Foreign key constraint referencing Price_Type.ID

Unique Keys

	Columns	Name/Description
	ID	PK_Price_Type_ID Primary key (clustered) constraint

Used By

Name
 Price_Types
 Properties

3.1.16. Table: Properties

This table give information about property of about everything users can choose.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for properties records. Identity/Auto increment column
	NAME	NVARCHAR(200)		Property Name
	Description	NVARCHAR(MAX)		Descriptions of Properties
	User_ID	INT		User information ID Foreign key to Users table
	Property_type_ID	INT		Property Type information ID Foreign key to Property_type table
	Room_type_ID	INT		Room Type information ID Foreign key to Room_type table
	Category_ID	INT		Category information ID Foreign key to Category table
	Subcategory_ID	INT		Subcategory information ID Foreign key to Subcategory table
	Country_ID	INT		Country information ID Foreign key to Country table
	State_ID	INT		State information ID Foreign key to State table.
	City_ID	INT		State information ID Foreign key to STATE table
	Currency_ID	INT		Currency information ID Foreign key to Currency table
	Room_Detail_ID	INT		Room detail information ID

				Foreign key to Room_detail table
	Addres_ID	INT		Address information ID Foreign key to Address table
	AVAILABİLİTY	TINYINT		Refers to availability 1=Yes available,0=No not available
	Start_DATE	DATE		Date of start use this property
	End_DATE	DATE		Date of end use this property
	Price	FLOAT		Price for this properties
	Price_Type_ID	TINYINT		User's price type
	Refund_Type	TINYINT		Refund type about this property
	Minimum_Stay	INT		User minimum stay time(as day)
	Created	DATETIME		Date properties pated users
	Modified	DATETIME		The date when propertiesrenewed or corrected the users
	Status	TINYINT		Status of the property

Linked From

	Table	Join	Title/Name/Description
	User_ID	PROPERTIES.USER_ID=USER.ID	FK_ Properties.User_ID Foreign key constraint referencing USER.ID
	Property_type_ID	PROPERTIES.Property_Type_ID=PROPERTY_TYPE.ID	FK_ Properties. Property_type_ID Foreign key constraint referencing PROPERTY_TYPE.ID
	Room_type_ID	PROPERTIES.Room_Type_ID=ROOM_TYPE.ID	FK_ Properties. Room_type_ID Foreign key constraint referencing ROOM_TYPE.ID
	Category_ID	PROPERTIES.Category_ID=CATEGORIES.ID	FK_ Properties. Category_ID

			Foreign key constraint referencing CATEGORIES.ID
	Subcategory_ID	PROPERTIES.Sub_Category_ID=SUBCATEGORIES.ID	FK_ Properties. Subcategory_ID Foreign key constraint referencing SUBCATEGORIES.ID
	Country_ID	PROPERTIES.COUNTRY_ID=COUNTRIES.ID	FK_ Properties. Country_ID Foreign key constraint referencing COUNTRIES.ID
	State_ID	PROPERTIES.STATE_ID=STATE.ID	FK_ Properties. State_ID Foreign key constraint referencing STATE.ID
	City_ID	PROPERTIES.CITY_ID=CITIES.ID	FK_ Properties. City_ID Foreign key constraint referencing CITIES.ID
	Currency_ID	PROPERTIES.CURRENCY_ID=CURRENCIES.ID	FK_ Properties. Currency_ID Foreign key constraint referencing CURRENCIES.ID
	Room_Detail_ID	PROPERTIES.ROOM_DETAIL_ID=ROOM_DETAIL.ID	FK_ Properties. Room_Detail_ID Foreign key constraint referencing ROOM_DETAIL.ID
	Address_ID	PROPERTIES.Address_ID=ADDRESS.ID	FK_ Properties. Address_ID Foreign key constraint referencing ADDRESS.ID
	Price_Type_ID	PROPERTIES.Price_Type_ID=PRICE_TYPE.ID	FK_ Properties. Price_Type_ID Foreign key constraint referencing PRICE_TYPE.ID

Links to

	Table	Join	Title/Name/Description
	Property_Amenities	PROPERTIES.ID=PROPERTY_AMENITIES.PROPERTY_ID	FK_ Property_Amenities.PROPERTY_TYPE Foreign key constraint referencing PROPERTIES.ID
	Property_Images	PROPERTIES.ID=PROPERTY_IMAGES.PROPERTY_ID	FK_ Property_Images.PROPERTY_TYPE Foreign key constraint referencing PROPERTIES.ID
	TRANSACTIONS	PROPERTIES.ID=TRANSACTIONS.PROPERTY_ID	FK_ TRANSACTIONS.PROPERTY_TYPE Foreign key constraint referencing PROPERTIES.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
PROPERTIES
Property_Amenities
Property_Images
TRANSACTIONS

3.1.17. Table: Promo Codes

Details of the promotion codes are stored by table Promo_codes.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for promo codes' records. Identity/Auto increment column
	Name	NVARCHAR(200)		Promo code name
	Description	NVARCHAR(MAX)		Code description
	Code	VARCHAR(10)		Promo Code
	Discount	FLOAT		Discount amount
	Created	DATETIME		Created date and time
	Modified	DATETIME		Modified date and time
	Status	TINYINT		Status of the promo code

Links to

	Table	Join	Title/Name/Description
	Transactions	Promo_codes.ID = Transactions.Promo_code_ID	FK_ Transactions.Promo_code_ID Foreign key constraint referencing Promo_codes.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
 Transactions

3.1.18. Table: Property Amenities

This table give information about Properties of Amenities. It has connections with PROPERTIES and AMENTIES table.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for property of amenties.
	PROPERTY_ID	INT		It links Properties table
	AMENITY_ID	INT		It links Amentiy Table
	Created	DATETIME		Date language pated system
	Modified	DATETIME		The date when language renewed or corrected the system
	Status	TINYINT		Status of the language

Linked From

	Table	Join	Title/Name/Description
	PROPERTIES	PROPERTY_AMENTIES.PROPERTY_ID=PROPERTIES.ID	FK_Property_Amenities.PROPERTY_ID Foreign key constraint referencing PROPERTIES.ID
	AMENTIES	PROPERTY_AMENTIES.AMENITY_ID=AMENTIES.ID	FK_Property_Amenities.AMENITY_ID Foreign key constraint referencing AMENTIES.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
PROPERTY_AMENITIES

3.1.19. Table: Property Images

It is the table where the image information of the property table is kept.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for property images records. Identity/Auto increment column
	PROPERTY_ID	INT		Unique identification number for properties information. Foreign key to properties table
	ADDED_BY_USER	NVARCHAR(200)	N	To indicate which user added
	IMAGE	NVARCHAR(MAX)	N	Properties the pictures
	CREATED	DATETIME	N	The date the property image was created
	STATUS	TINYINT	N	Property images for status

4.

Linked From

	Table	Join	Title/Name/Description
	Properties	Property_Images.Property_ID= Properties.ID	FK_Property_Images_Property_ID Foreign key constraint referencing Properties.ID

Unique Keys

	Columns	Name/ Description
	ID	PK_Property_Images_ID Primary key (clustered) index
	ADDED_BY_USER	IX_Property_Images_ ADDED_BY_USER Nonclustered index

Uses

Name
 Property_Images
 Properties

3.1.20. Table: Property Reviews

Table with user's opinions

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for property review records. Identity/Auto increment column
	PROPERTY_ID	INT		Unique identification number for properties information. Foreign key to properties table
	REVIEW_BY_USER	NVARCHAR(200)	N	Names of users who post comments
	BOOKING_ID	INT		Unique identification number for bookings information. Foreign key to bookings table
	COMMENT	NVARCHAR(MAX)	N	Users comments

	RATING	TINYINT	N	The vote by users on the evaluation results
	CREATED	DATETIME	N	Date users pated comments
	MODIFIED	DATETIME	N	The date when users renewed or corrected the comment
	STATUS	TINYINT		Status of the comments

Links To

	Table	Join	Title/Name/Description
	Properties	Property_Review.Propert _ID = Properties.ID	FK_Property_Review_Property_Propety_I D Foreign key constraint referencing Properties.ID
	Booking	Property_Review.Booking _ID = Booking.ID	FK_ Property_Review_Bookıngs_Booking_ID Foreign key constraint referencing Booking.ID

Unique Keys

	Columns	Name/Description
	ID	PK_Propert_Review_ID Primary key (clustered) constraint
	REVIEW_BY_USER	IX_Property_Review_REVIEW_BY_USER Nonclustered index

Uses

Name
 Property_Images
 Properties

Bookings

3.1.21. Table: Property Type

It contains information about property type.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for property type records.
	Name	NVARCHAR(200)		Property Type Name
	Icon_Image	VARCHAR(MAX)		Property Type icon Image Url
	Created	DATETIME		Date Created pated system
	Modified	DATETIME		The date when property type renewed or corrected the system
	Status	TINYINT		Status of the property type

Links to

	Table	Join	Title/Name/Description
	PROPERTIES	PROPERTIES.PROPERTY_TYPE.ID= PROPERTY_TYPE.ID	FK_PROPERTIES.PROPERTY_TYPE_ID Foreign key constraint referencing PROPERTY_TYPE.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
PROPERTY_TYPE
PROPERTIES

3.1.22. Table: Room Detail

It contains information about rooms' details like number of bedroom, number of bathroom etc.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for room detail. Identity/Auto increment column
	Bedroom_count	INT		Number of bedrooms
	Bed_count	INT		Number of beds
	Bathroom_count	INT		Number of bathrooms
	Accomodates_count	INT		Number of accomodates
	Created	DATETIME		Created date and time
	Modified	DATETIME		Modified date and time

Links to

	Table	Join	Title/Name/Description
	Properties	Room_detail.ID=Properties.Room_Detail_ID	FK_ Room_detail.ID Foreign key constraint referencing Properties.Room_Detail_ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
 Properties

3.1.23. Table: Room Type

Room information for properties table.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for categories records Identity/ Auto increment column
	NAME	NVARCHAR(100)	N	Room name
	ICON_IMAGE	NVARCHAR(MAX)	N	Icon of room images
	CREATED	DATETIME	N	The date the room was put into service
	MODIFIED	DATETIME	N	Renovation or modification date in the room
	STATUS	TINYINT		Room condition

Linked From

	Table	Join	Title/Name/Description
	Properties	Properties.Room_Type_ID = Room_Type.ID	FK_Properties_Room_Type_Room_Type_ID

			Foreign key constraint referencing Room_Type.ID

Unique keys

	Columns	Name/Description
	ID	PK_Room_Type_ID Primary key (clustered) constraint
	Name	IX_Room_Type_Name Nonclustered index

Used by

Name
Room_Type
Properties

3.1.24. Table: States

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for state record
	COUNTRYID	INT		It use for linking CONTRIES table
	NAME	INT		Name of the STATES
	CODE	DATETIME		Code of the STATES
	STATUS	DATETIME		Status of the STATES

Linked From

	Table	Join	Title/Name/Description
	COUNTRIES	STATE.COUNTRY_ID=COUNTRY.ID	FK_State.COUNTRY_ID Foreign key constraint referencing COUNTRIES.ID

Links to

	Table	Join	Title/Name/Description
	CITIES	STATE.ID=CITIES.STATE_ID	FK_ cities.State_ID Foreign key constraint referencing State.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
STATES
CITIES

3.1.25. Table: Subcategories

The table where the subcategory information about the category is kept

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for subcategories records.

				Identity/Auto increment column
	CATEGORY_ID	INT		Unique identification number of category types. Foreign key to category_ID int Categories table
	NAME	NVARCHAR(200)	N	Subcategory name
	DESCRIPTION	NVARCHAR(MAX)	N	Defining for subcategories
	CREATED	DATETIME	N	Cretion date for subcategories
	MODIFIED	DATETIME	N	Renewal date for subcategories
	STATUS	TINYINT		The status for subcatefories

Links To

	Table	Join	Title/Name/Description
	Categories	Subcategories.Categor_ID = Categories.ID	FK_Subcategories_Categories_Category_ID Foreign key constraint referencing Categories.ID

Linked From

	Table	Join	Title/Name/Description
	Properties	Subcategories.ID = Properties.Subcategory_ID	FK_Properties_Subcategories_Subcategory_ID Foreign key constraint referencing Subcategories.ID

Unique Keys

	Columns	Name/ Description
	ID	PK_Subcategories_ID Primary key (clustered) constraint

Uses

Name
Subcategories
Categories

Used By

Name
Subcategories
Properties

3.1.26. Table: Transactions

To store transactions' details like site fees, amount of the stay or discount and their processing dates, we use table 'Transactions'.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for transactions ID. Identity/Auto increment column
	PROPERTY_ID	INT		Property id, Foreign Key to Properties table
	USERID	INT		Userid, Foreign Key to Users table
	SITE_FEES	FLOAT		Email address
	AMOUNT	FLOAT		Amount of the room or house
	TRANFER_ON	DATETIME		Date of transfer
	CURRENCY_ID	INT		Currency id, Foreign Key to Currencies table
	PROMO_CODE_ID	INT		Promocode id, Foreign Key to Promo codes table
	DISCOUNT_AMOUNT	FLOAT		Amount of discount

	CREATED	DATETIME		Created date and time
	MODIFIED	DATETIME		Modified date and time
	STATUS	TINYINT		Status of the transaction

Linked From

	Table	Join	Title/Name/Description
	Properties	Transactions.Property_ID=Properties.ID	FK_ Transactions.Property_ID Foreign key constraint referencing Properties Properties.ID
	Users	Transactions.USERID=Users.ID	FK_ Transactions.USERID Foreign key constraİnt referencing Users.ID
	Currencies	Transactions.Currency_ID=Currencies.ID	FK_ Transactions.Currency_ID Foreign key constraint referencing Currencies.ID
	Promo_codes	Transactions.Promo_code_id=Promo_codes.ID	FK_ Transactions.Promo_code_id Foreign key constraint referencing Promo_codes.ID

Links to

	Table	Join	Title/Name/Description
	Bookings	Transactions.ID=Bookings.Transaction_ID	FK TRANSACTION_ID Foreign key constraint referencing Transactions.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
Bookings

3.1.27. Table: User Type

Indicate the types of users.

Columns

	Name	Data Type	N	Description/Attributes
	ID	TINYINT		Primary key for user type records. Identity/Auto increment column
	NAME	NVARCHAR(50)		Names of user type

Linked From

	Table	Join	Title/Name/Description
	Users	User_Type.ID = Users.User_Type_ID	FK_Users_User_Type_ID Foreign key constraint referencing User_Type.ID

Unique Keys

	Columns	Name/Description
	ID	PK_User_Type_ID Primary key (clustered) constraint

Used By

Name
User_Type
Users

3.1.28. Table: Users

The user's information such as name, surname, the phone is stored Users table.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Primary key for user records. Identity/Auto increment column
	First_Name	NVARCHAR(200)		User real name
	Last_Name	NVARCHAR(200)		User surname
	Email	NVARCHAR(200)		Email address
	Password	NVARCHAR(30)		Users' password
	User_Type_ID	TINYINT		Old user,new user,active or inactive Foreign key to User_Type table
	Birthday	DATE		Date of birth
	Gender	CHAR(7)		Male and Female
	Login_With_ID	TINYINT		Network login Foreign key to Login_with table
	Phone	CHAR(12)		Contact number with the user
	Receive_Coupon	INT		Refers to whether user has any discount coupons 1=Yes,0=No
	Country_ID	INT		It refers country ıd in database Foreign key to Countries table
	Created	DATETIME		Date users pated users
	Modified	DATETIME		The date when users renewed or corrected the users
	Status	TINYINT		Status of the users
	Language_ID	INT		It refers language ıd in database Foreign key to Languages table

Linked From

	Table	Join	Title/Name/Description
	User_Type	Users.User_Type_ID=User_Type.ID	FK_ Users.User_Type_ID Foreign key constraint referencing User_Type.ID
	Login_with	Users.Login_With_ID=Login_with.ID	FK_ Users.Login_With_ID Foreign key constraint referencing Login_with.ID
	Countries	Users.Country_ID=Countries.ID	FK_ Users.Country_ID Foreign key constraint referencing Countries.ID
	Language	Users.Language_ID=Language.ID	FK_ Users.Language_ID Foreign key constraint referencing Language.ID

Links to

	Table	Join	Title/Name/Description
	Properties	Users.ID=Properties.User_ID	FK Properties.User_ID Foreign key constraint referencing Users.ID
	Bookings	Users.ID = Bookings.User_ID	FK_ Bookings.User_ID Foreign key constraint referencing Users.ID
	Disputes	Users.ID = Disputes.User_ID	FK_ Disputes.User_ID Foreign key constraint referencing Users.ID
	Transactions	Users.ID =Transactions.UserID	FK_ Transactions.UserID Foreign key constraint referencing Users.ID

Unique Keys

	Columns	Name/Description
	ID	PK_ID Primary key (clustered) constraint

Used By

Name
Users
Properties
Bookings

3.2. Views

3.2.1. View: Reservation Information By Month

Joins transactions,users tables with booking table.

Columns

	Name	Data Type	N	Description/Attributes
	B.BOOKING_DATE	DATETIME		Booking date in the booking table
	T.AMOUNT	FLOAT		Amount column in the transactions
	T.SITE_ FEES	FLOAT		Site fees column in the transactions
	U.ID	INT		ID of the users table
	B.ID	INT		ID of the booking table

3.2.2. View: Reservation Features Detailed

Joins subcategories,properties,property_type,categorıes tables with booking table.

Columns

	Name	Data Type	N	Description/Attributes
	C.NAME_	NVARCHAR(200)		Category name
	S.NAME	NVARCHAR(200)		Subcategoris name
	P.NAME	NVARCHAR(200)		Properties name
	PT.NAME	NVARCHAR(200)		Property type name
	B.BOOKING_DATE	DATETIME		Booking date in the booking table
	B.ID	INT		ID of the booking table
	B.PRICE_FOR_STAY	FLOAT		Prıce fors tay column in the bookin table
	B.SITE_FES	FLOAT		Site Fees column in the booking table

3.2.3. View: VW_PET_PROPERTY

The table shows the properties which allows pets and the country of these properties.

Columns

	Name	Data Type	N	Description/Attributes
	Country	NVARCHAR(200)		The country of the property
	Property	NVARCHAR(200)		Property type
	Amenity	NVARCHAR(200)		‘Pets live on this property’

3.2.4. View: dbo.YoungUsers

It shows the ıd,name, last name,gender and age.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Auto increment column
	First_Name	NVARCHAR(200)		
	Last_Name	NVARCHAR(200)		
	Gender	CHAR(7)		
	Age	INT		

3.2.5. View: dbo.AMOUNT_PAID_500PLUS

The table shows the name from Properties table, the id,name, last name and gender from Users table,id from Booking table and name from Country table.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Auto increment column
	Property_Name	NVARCHAR(200)		
	User_ID	INT		
	First_Name	NVARCHAR(200)		
	Last_Name	NVARCHAR(200)		
	Gender	CHAR(7)		
	Booking_ID	INT		
	Country_Name	NVARCHAR(200)		

3.2.6. View: dbo.BOOKING_AND_CUSTOMERS_LIST

The table shows id,first name and last name from Users table,check in-check out day and amount paid from Bookings table. Includes bookings made by users and payment information to them.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Auto increment column
	First_Name	NVARCHAR(200)		
	Last_Name	NVARCHAR(200)		
	Check_in_Date	DATETIME		
	Check_out_Date	DATETIME		
	Amount_Paid	FLOAT		

3.2.7. View: Booking_Count

The table shows id and booking count from view of Booking and Customers List.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Auto increment column
	Booking_Count	INT		

3.2.8. View: UK_MESSAGES

Contains information from e-mails sent by UK-English.

Columns

	Name	Data Type	N	Description/Attributes
	ID	INT		Auto increment column
	Subject	NVARCHAR(100)		
	Keywords	NVARCHAR(MAX)		
	Message	NVARCHAR(500)		
	Name	NVARCHAR(200)		

3.3. Procedures

3.3.1. Procedure: Reservation By Months

Input/Output

	Name	Data Type	N	Description/Attributes
	BASAY	INT		Input parameter for the stored procedure booking Inserted Enter a valid basay from the booking table
	ENDAY	INT		Input parameter for the stored procedure booking Inserted Enter a valid enday from the booking table
	BOOKING_DATE	INT		output parameter for the stored procedure booking Inserted Enter a valid booking_date name column of the booking table

3.3.2. Procedure: User Login Control

Input/Output

	Name	Data Type	N	Description/Attributes
	FIRSTNAME	NVARCHAR(200)		Input parameter for the stored procedure booking Inserted Enter a valid basay from the booking table
	SIFRE	INT		Input parameter for the stored procedure booking Inserted Enter a valid enday from the booking table
	VAR	INT		output parameter for the stored procedure

3.3.3. Procedure: Search by dates

Input/Output

	Name	Data Type	N	Description/Attributes
	START_DATE	DATE		Input parameter for the stored procedure properties Inserted Enter a valid start_date from the properties table
	END_DATE	DATE		Input parameter for the stored procedure properties Inserted Enter a valid end_date from the properties table
	KONUM1	NVARCHAR(200)		output parameter for the stored procedure countries Inserted Enter a valid konum1 name column of the countries table
	KONUM2	NVARCHAR(200)		output parameter for the stored procedure cities Inserted

				Enter a valid konum2 name column of the cities table
	KONUM3	NVARCHAR(200)		output parameter for the stored procedure states Inserted Enter a valid konum3 name column of the states table

3.3.4. Procedure: Transaction Inserted

Input/Output

	Name	Data Type	N	Description/Attributes
	ID	INT		Input parameter for the stored procedure Transactions Inserted Enter a valid ID from the transactions table
	USER_ID	INT		Input parameter for the stored procedure Transactions Inserted Enter a valid user_id from the transactions table
	TRANSFER_ON	DATETIME		Input parameter for the stored procedure Transactions Inserted Enter a valid transfer on from the transactions table
	AMOUNT	FLOAT		Input parameter for the stored procedure Transactions Inserted Enter a valid amount from the transactions table
	STATUS	TINYINT		Input parameter for the stored procedure Transactions Inserted Enter a valid status from the transactions table

3.3.5. Procedure: stpInsertPromoCode

It is used for inserting new promo code informations.

Input/Output

	Name	Data Type	Description/Attributes
@	ID	INT	Input parameter for the stored procedure for ID
@	NAME	NVARCHAR(200)	Input parameter for the stored procedure for Name of the promo code
@	DESCRIPTION	NVARCHAR(MAX)	Input parameter for the stored procedure for Description of the promo code
@	CODE	VARCHAR(10)	Input parameter for the stored procedure for Code
@	DISCOUNT	FLOAT	Input parameter for the stored procedure amount of the discount
@	CREATED	DATETIME	Input parameter for the stored procedure created datetime of the promo code
@	MODIFIED	DATETIME	Input parameter for the stored procedure modified datetime of the promo code
@	STATUS	TINYINT	Input parameter for the stored procedure to status

3.3.6. Procedure: dbo.sp_amenities_availability

It is used to search easily the key word and show the related informations regarding the key Word.

Input/Output

	Name	Data Type	Description
@	Amenity_Name	NVARCHAR(200)	Amenity name that is searched by keyword
@	Property_ID	INT	Output parameter
@	Property_Status	TINYINT	Output parameter
@	Property_Address_ID	INT	Output parameter

@	Property_Description	NVARCHAR(MAX)	Output parameter
@	Property_Price	FLOAT	Output parameter
@	Country_name	NVARCHAR(200)	Output parameter

3.3.7. Procedure: dbo.sum_salaries

It is used for finding users by their Surname.

Output

	Name	Data Type	Description
@	p_sum	INTEGER	Output parameter for the stored procedure dbo.sum_salaries. User can see this value on screen. It finds by this stored procedure

3.3.8. Procedure: dbo.CheckRoomAvailability

Input/Output

It used for finding available room acoording bed,bedroom,bathroom count. Values enter by user

	Name	Data Type	Description/Attributes
@	Created	DATETIME	Input parameter for the stored procedure for a created time. User enter a valid date and time
@	Modified	DATETIME	Input parameter for the stored procedure for a created time. User enter a valid date and time
@	BedCount	INT	Input parameter for the stored procedure to calculate total Bed Count according user
@	BedRoomCount	INT	Input parameter for the stored procedure to calculate total Bedroom Count according user

@	BathroomCount	INT	Input parameter for the stored procedure to calculate total Bathroom Count according user

3.3.9. Procedure: dbo.SalesTarget

If the total, non-canceled bookings, and income from bookings made in 2017 if is over 5000, the response is checked. Shown as total sales.

Input/Output

	Name	Data Type	Description/Attributes
@	TotalSales	INT	Input parameter for the stored procedure SalesTarget. Enter a valid target sales amount.
@	Sales	INT	Input parameter for the stored procedure to calculate TotalSales.

3.3.10. Procedure: dbo.spPeople_GetbyLastName

It is used for finding users by their Surname.

Input/Output

	Name	Data Type	Description
@	LastName	NVARCHAR(200)	Input parameter for the stored procedure spPeople_GetbyLastName. Enter a valid LastName from Users.

3.3.11. Procedure: dbo.spYoungUsers

It shows the ıd,name, last name,gender and between the ages of 20 and 40 in the View: Users.YoungUsers.

Input/Output

	Name	Data Type	Description
@	ID	INT	Output parameter
@	First_Name	NVARCHAR(200)	Output parameter
@	Last_Name	NVARCHAR(200)	Output parameter
@	Gender	CHAR(7)	Output parameter
@	Age	INT	Output parameter

3.3.12. Procedure: AddUser

Adds new user

Input/Output

	Name	Data Type	N	Description/Attributes
	FIRSTNAME	NVARCHAR(200)		Input parameter for the stored procedure Add User Enter a valid firstname from the users table
	LASTNAME	NVARCHAR(200)		Input parameter for the stored procedure Add User Enter a valid lastname from the users table
	EMAIL	NVARCHAR(200)		Input parameter for the stored procedure Add User Enter a valid email from the users table
	PASSWORD	NVARCHAR(200)		Input parameter for the stored procedure Add User Enter a valid password from the users table

3.4. Functions

3.4.1. Function: Customers' Bookings certain Interval

Table value function returning the showing customers who booked in a certain price range.

Input/Output

	Name	Data Type	N	Description/Attributes
	Returns	Table type		
	ENAZ	INT		Input parameter for the table valued function valid enaz from booking table

	ENCOK	INT		ınput parameter for the table valued function valid encok from booking table

3.4.2. Function: Reservation Information by month

Table value function returning reservation information.

Input/Output

	Name	Data Type	N	Description/Attributes
	Returns	Table type		
	AY	INT		Input parameter for the table valued function valid enaz from booking table

3.4.3. Function: tbldispute

Table value function returns the informations of the disputes

Input/Output

	Name	Data Type	N	Description/Attributes
	Returns	Table type		
	Fist_name	NVARCHAR(200)		Output parameter for the table valued function valid First_name from USERS table
	Email	NVARCHAR(200)		Output parameter for the table valued function valid Email from USERS table
	Phone	CHAR(12)		Output parameter for the table valued function valid Phone from USERS table
	Dispute_name	NVARCHAR(200)		Output parameter for the table valued function valid Name from DISPUTES table
	Dispute_description	NVARCHAR(MAX)		Output parameter for the table valued function valid Description from DISPUTES table
	Property_Description	NVARCHAR(MAX)		Output parameter for the table valued function valid Description from PROPERTIES table

3.4.4. Function: Fn_PT_REVIEW

According property type,Table value function returning the showing properties,user review,user rating and star sembol using rating.

Input/Output

	Name	Data Type	N	Description/Attributes
	Returns	Table type		
	Property_type	NVARCHAR(15)		Input parameter for the table valued function valid property_type from Property_Type table
	NAME	NVARCHAR(500)		Output parameter for the table valued function valid NAME from PROPERTY_TYPE table
	COMMENT	NVARCHAR(MAX)		Output parameter for the table valued function valid COMMENT from PROPERTY_REVIEW table
	RATING OUTPUT	NVARCHAR		Output parameter for the table valued function valid RATING from PROPERTY_REVIEW table.According this value count RATING OUTPUT shows * symbol.

3.4.5. Function: fn_FindingAvailableLocation

Table value function returning available location information according countryname and first letter of city.

Input/Output

	Name	Data Type	N	Description/Attributes
	Returns	Table type		
	ulkeAd	Nvarchar(50)		Input and output parameter(as country name) for the table valued function valid from countries,states and cities table
	statename	Nvarchar(200)		Output parameter for the table valued function valid NAME from STATES table
	cityname	Nvarchar(200)		Output parameter for the table valued function valid NAME from CITIES table

3.4.6. Function: fn_Bookings

Table value function returns the information of the bookings.

Input/Output

	Name	Data Type	N	Description/Attributes
	Returns	INT		
	Fist_name	NVARCHAR(200)		Output parameter for the table valued function valid First_name from USERS table
	Last_Name	NVARCHAR(200)		Output parameter for the table valued function valid Last_Name USERS table

	ID	INT		Output parameter for the table valued function valid ID from Bookings table

3.4.7. Function: fn_CategoryProperties

Table value function returns the information of the properties.

Input/Output

	Name	Data Type	N	Description/Attributes
	Returns	INT		
	Name	NVARCHAR(200)		Output parameter for the table valued function valid name from properties table
	Property_Count	INT		Output parameter for the table valued function valid Property_Count properties table

3.5. Triggers

3.5.1. Trigger: Reservation update

Automatically updates upon reservation cancellation or creation.

Input/Output

	Name	Data Type	N	Description/Attributes
	ID	INT		Input parameter valid ID from booing table

3.5.2. Trigger: EventsLog

Used to view objects created in the system.

Input/Output

	Name	Data Type	N	Description/Attributes
	DATA	XML		Input parameter for the triggerEventsLog Inserted Enter a valid data

4. Database Backup

Weekly Database Backup Strategy

EN: Since our project has an international character, it has a database that has a lot of data that grows a lot and its area grows with various operations performed during the day. For this reason, full backup can be made daily and 48 times a day (every half hour) backup process can be done. In addition, backup files that take unnecessary memory can be deleted by performing Maintenance Clean-up process weekly (once a week, including on Sundays). However, we believe that it is more appropriate for us to do the Shrink process when needed without tying it to a certain routine. Because regularly reducing space in memory can cause the loss of parts that are important to us. For this reason, we aim to make shrink operations as much as we need when needed.

TR: Projemiz uluslararası nitelik taşıdığı için alanı çokça büyüyen çokça veriye sahip olan ve alanı gün içinde yapılan çeşitli işlemler ile büyüyen bir veri tabanına sahiptir. Bu nedenle günlük olarak Full backup ve günde 48 defa (yarım saatte bir aralıklarla) yedekleme işlemi yapılabilir. Ayrıca haftalık olarak (Pazar günleri olmak üzere haftada bir şeklinde) Maintanance Clean-up işlemi yapılarak hafızada gereksiz olarak yer tutan backup dosyalarıda silinebilir. Ancak Shrink işlemini belli bir rutine bağlamaksızın ihtiyaç duyulduğunda yapılmasının bizim için daha uygun olduğu kanaatindeyiz. Çünkü düzenli olarak hafızada alan eksiltmek bizim için önemli olan kısımların kaybına sebep olabilir. Bu sebeple ihtiyaç duyduğumuzda ihtiyacımız kadar boyutta shrink işlemi yapmayı hedeflemekteyiz.

FULL BACKUP 1. GÜN	FULL BACKUP 2. GÜN	FULL BACKUP 3. GÜN	FULL BACKUP 4. GÜN	FULL BACKUP 5. GÜN	FULL BACKUP 6. GÜN	FULL BACKUP 7. GÜN
DIFFERENTIAL BACKUP 06.30	DIFFERENTIAL BACKUP 06.30	DIFFERENTIAL BACKUP 06.30	DIFFERENTIAL BACKUP 06.30	DIFFERENTIAL BACKUP 06.30	DIFFERENTIAL BACKUP 06.30	DIFFERENTIAL BACKUP 06.30
DIFFERENTIAL BACKUP 07:00	DIFFERENTIAL BACKUP 07:00	DIFFERENTIAL BACKUP 07:00	DIFFERENTIAL BACKUP 07:00	DIFFERENTIAL BACKUP 07:00	DIFFERENTIAL BACKUP 07:00	DIFFERENTIAL BACKUP 07:00
DIFFERENTIAL BACKUP 07:30	DIFFERENTIAL BACKUP 07:30	DIFFERENTIAL BACKUP 07:30	DIFFERENTIAL BACKUP 07:30	DIFFERENTIAL BACKUP 07:30	DIFFERENTIAL BACKUP 07:30	DIFFERENTIAL BACKUP 07:30
DIFFERENTIAL BACKUP 08:00	DIFFERENTIAL BACKUP 08:00	DIFFERENTIAL BACKUP 08:00	DIFFERENTIAL BACKUP 08:00	DIFFERENTIAL BACKUP 08:00	DIFFERENTIAL BACKUP 08:00	DIFFERENTIAL BACKUP 08:00
DIFFERENTIAL BACKUP 08:30	DIFFERENTIAL BACKUP 08:30	DIFFERENTIAL BACKUP 08:30	DIFFERENTIAL BACKUP 08:30	DIFFERENTIAL BACKUP 08:30	DIFFERENTIAL BACKUP 08:30	DIFFERENTIAL BACKUP 08:30
DIFFERENTIAL BACKUP 09:00	DIFFERENTIAL BACKUP 09:00	DIFFERENTIAL BACKUP 09:00	DIFFERENTIAL BACKUP 09:00	DIFFERENTIAL BACKUP 09:00	DIFFERENTIAL BACKUP 09:00	DIFFERENTIAL BACKUP 09:00
DIFFERENTIAL BACKUP	DIFFERENTIAL BACKUP	DIFFERENTIAL BACKUP	DIFFERENTIAL BACKUP	DIFFERENTIAL BACKUP	DIFFERENTIAL BACKUP	DIFFERENTIAL BACKUP

09:30	09:30	09:30	09:30	09:30	09:30	09:30
DIFFERENTIAL BACKUP 10:00	DIFFERENTIAL BACKUP 10:00	DIFFERENTIAL BACKUP 10:00	DIFFERENTIAL BACKUP 10:00	DIFFERENTIAL BACKUP 10:00	DIFFERENTIAL BACKUP 10:00	DIFFERENTIAL BACKUP 10:00
DIFFERENTIAL BACKUP 10:30	DIFFERENTIAL BACKUP 10:30	DIFFERENTIAL BACKUP 10:30	DIFFERENTIAL BACKUP 10:30	DIFFERENTIAL BACKUP 10:30	DIFFERENTIAL BACKUP 10:30	DIFFERENTIAL BACKUP 10:30
DIFFERENTIAL BACKUP 11:00	DIFFERENTIAL BACKUP 11:00	DIFFERENTIAL BACKUP 11:00	DIFFERENTIAL BACKUP 11:00	DIFFERENTIAL BACKUP 11:00	DIFFERENTIAL BACKUP 11:00	DIFFERENTIAL BACKUP 11:00
DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30
DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30	DIFFERENTIAL BACKUP 11:30
DIFFERENTIAL BACKUP 12:00	DIFFERENTIAL BACKUP 12:00	DIFFERENTIAL BACKUP 12:00	DIFFERENTIAL BACKUP 12:00	DIFFERENTIAL BACKUP 12:00	DIFFERENTIAL BACKUP 12:00	DIFFERENTIAL BACKUP 12:00
DIFFERENTIAL BACKUP 12:30	DIFFERENTIAL BACKUP 12:30	DIFFERENTIAL BACKUP 12:30	DIFFERENTIAL BACKUP 12:30	DIFFERENTIAL BACKUP 12:30	DIFFERENTIAL BACKUP 12:30	DIFFERENTIAL BACKUP 12:30
DIFFERENTIAL BACKUP 13:00	DIFFERENTIAL BACKUP 13:00	DIFFERENTIAL BACKUP 13:00	DIFFERENTIAL BACKUP 13:00	DIFFERENTIAL BACKUP 13:00	DIFFERENTIAL BACKUP 13:00	DIFFERENTIAL BACKUP 13:00
DIFFERENTIAL BACKUP 13:30	DIFFERENTIAL BACKUP 13:30	DIFFERENTIAL BACKUP 13:30	DIFFERENTIAL BACKUP 13:30	DIFFERENTIAL BACKUP 13:30	DIFFERENTIAL BACKUP 13:30	DIFFERENTIAL BACKUP 13:30
DIFFERENTIAL BACKUP 14:00	DIFFERENTIAL BACKUP 14:00	DIFFERENTIAL BACKUP 14:00	DIFFERENTIAL BACKUP 14:00	DIFFERENTIAL BACKUP 14:00	DIFFERENTIAL BACKUP 14:00	DIFFERENTIAL BACKUP 14:00
DIFFERENTIAL BACKUP 14:30	DIFFERENTIAL BACKUP 14:30	DIFFERENTIAL BACKUP 14:30	DIFFERENTIAL BACKUP 14:30	DIFFERENTIAL BACKUP 14:30	DIFFERENTIAL BACKUP 14:30	DIFFERENTIAL BACKUP 14:30
DIFFERENTIAL BACKUP 15:30	DIFFERENTIAL BACKUP 15:30	DIFFERENTIAL BACKUP 15:30	DIFFERENTIAL BACKUP 15:30	DIFFERENTIAL BACKUP 15:30	DIFFERENTIAL BACKUP 15:30	DIFFERENTIAL BACKUP 15:30
DIFFERENTIAL BACKUP 16:00	DIFFERENTIAL BACKUP 16:00	DIFFERENTIAL BACKUP 16:00	DIFFERENTIAL BACKUP 16:00	DIFFERENTIAL BACKUP 16:00	DIFFERENTIAL BACKUP 16:00	DIFFERENTIAL BACKUP 16:00
DIFFERENTIAL BACKUP 16:30	DIFFERENTIAL BACKUP 16:30	DIFFERENTIAL BACKUP 16:30	DIFFERENTIAL BACKUP 16:30	DIFFERENTIAL BACKUP 16:30	DIFFERENTIAL BACKUP 16:30	DIFFERENTIAL BACKUP 16:30
DIFFERENTIAL BACKUP 17:00	DIFFERENTIAL BACKUP 17:00	DIFFERENTIAL BACKUP 17:00	DIFFERENTIAL BACKUP 17:00	DIFFERENTIAL BACKUP 17:00	DIFFERENTIAL BACKUP 17:00	DIFFERENTIAL BACKUP 17:00
DIFFERENTIAL BACKUP 17:30	DIFFERENTIAL BACKUP 17:30	DIFFERENTIAL BACKUP 17:30	DIFFERENTIAL BACKUP 17:30	DIFFERENTIAL BACKUP 17:30	DIFFERENTIAL BACKUP 17:30	DIFFERENTIAL BACKUP 17:30

DIFFERENTIAL BACKUP 18:00	DIFFERENTIAL BACKUP 18:00	DIFFERENTIAL BACKUP 18:00	DIFFERENTIAL BACKUP 18:00	DIFFERENTIAL BACKUP 18:00	DIFFERENTIAL BACKUP 18:00	DIFFERENTIAL BACKUP 18:00
DIFFERENTIAL BACKUP 18:30	DIFFERENTIAL BACKUP 18:30	DIFFERENTIAL BACKUP 18:30	DIFFERENTIAL BACKUP 18:30	DIFFERENTIAL BACKUP 18:30	DIFFERENTIAL BACKUP 18:30	DIFFERENTIAL BACKUP 18:30
DIFFERENTIAL BACKUP 19:00	DIFFERENTIAL BACKUP 19:00	DIFFERENTIAL BACKUP 19:00	DIFFERENTIAL BACKUP 19:00	DIFFERENTIAL BACKUP 19:00	DIFFERENTIAL BACKUP 19:00	DIFFERENTIAL BACKUP 19:00
DIFFERENTIAL BACKUP 19:30	DIFFERENTIAL BACKUP 19:30	DIFFERENTIAL BACKUP 19:30	DIFFERENTIAL BACKUP 19:30	DIFFERENTIAL BACKUP 19:30	DIFFERENTIAL BACKUP 19:30	DIFFERENTIAL BACKUP 19:30
DIFFERENTIAL BACKUP 20:30	DIFFERENTIAL BACKUP 20:30	DIFFERENTIAL BACKUP 20:30	DIFFERENTIAL BACKUP 20:30	DIFFERENTIAL BACKUP 20:30	DIFFERENTIAL BACKUP 20:30	DIFFERENTIAL BACKUP 20:30
DIFFERENTIAL BACKUP 21:00	DIFFERENTIAL BACKUP 21:00	DIFFERENTIAL BACKUP 21:00	DIFFERENTIAL BACKUP 21:00	DIFFERENTIAL BACKUP 21:00	DIFFERENTIAL BACKUP 21:00	DIFFERENTIAL BACKUP 21:00
DIFFERENTIAL BACKUP 21:30	DIFFERENTIAL BACKUP 21:30	DIFFERENTIAL BACKUP 21:30	DIFFERENTIAL BACKUP 21:30	DIFFERENTIAL BACKUP 21:30	DIFFERENTIAL BACKUP 21:30	DIFFERENTIAL BACKUP 21:30
DIFFERENTIAL BACKUP 22:00	DIFFERENTIAL BACKUP 22:00	DIFFERENTIAL BACKUP 22:00	DIFFERENTIAL BACKUP 22:00	DIFFERENTIAL BACKUP 22:00	DIFFERENTIAL BACKUP 22:00	DIFFERENTIAL BACKUP 22:00
DIFFERENTIAL BACKUP 22:30	DIFFERENTIAL BACKUP 22:30	DIFFERENTIAL BACKUP 22:30	DIFFERENTIAL BACKUP 22:30	DIFFERENTIAL BACKUP 22:30	DIFFERENTIAL BACKUP 22:30	DIFFERENTIAL BACKUP 22:30
DIFFERENTIAL BACKUP 23:00	DIFFERENTIAL BACKUP 23:00	DIFFERENTIAL BACKUP 23:00	DIFFERENTIAL BACKUP 23:00	DIFFERENTIAL BACKUP 23:00	DIFFERENTIAL BACKUP 23:00	DIFFERENTIAL BACKUP 23:00
DIFFERENTIAL BACKUP 23:30	DIFFERENTIAL BACKUP 23:30	DIFFERENTIAL BACKUP 23:30	DIFFERENTIAL BACKUP 23:30	DIFFERENTIAL BACKUP 23:30	DIFFERENTIAL BACKUP 23:30	DIFFERENTIAL BACKUP 23:30
DIFFERENTIAL BACKUP 00:00	DIFFERENTIAL BACKUP 00:00	DIFFERENTIAL BACKUP 00:00	DIFFERENTIAL BACKUP 00:00	DIFFERENTIAL BACKUP 00:00	DIFFERENTIAL BACKUP 00:00	DIFFERENTIAL BACKUP 00:00
DIFFERENTIAL BACKUP 00:30	DIFFERENTIAL BACKUP 00:30	DIFFERENTIAL BACKUP 00:30	DIFFERENTIAL BACKUP 00:30	DIFFERENTIAL BACKUP 00:30	DIFFERENTIAL BACKUP 00:30	DIFFERENTIAL BACKUP 00:30
DIFFERENTIAL BACKUP 01:00	DIFFERENTIAL BACKUP 01:00	DIFFERENTIAL BACKUP 01:00	DIFFERENTIAL BACKUP 01:00	DIFFERENTIAL BACKUP 01:00	DIFFERENTIAL BACKUP 01:00	DIFFERENTIAL BACKUP 01:00
DIFFERENTIAL BACKUP 01:30	DIFFERENTIAL BACKUP 01:30	DIFFERENTIAL BACKUP 01:30	DIFFERENTIAL BACKUP 01:30	DIFFERENTIAL BACKUP 01:30	DIFFERENTIAL BACKUP 01:30	DIFFERENTIAL BACKUP 01:30
DIFFERENTIAL BACKUP 02:00	DIFFERENTIAL BACKUP 02:00	DIFFERENTIAL BACKUP 02:00	DIFFERENTIAL BACKUP 02:00	DIFFERENTIAL BACKUP 02:00	DIFFERENTIAL BACKUP 02:00	DIFFERENTIAL BACKUP 02:00
DIFFERENTIAL BACKUP 02:30	DIFFERENTIAL BACKUP 02:30	DIFFERENTIAL BACKUP 02:30	DIFFERENTIAL BACKUP 02:30	DIFFERENTIAL BACKUP 02:30	DIFFERENTIAL BACKUP 02:30	DIFFERENTIAL BACKUP 02:30

DIFFERENTIAL BACKUP 03:00	DIFFERENTIAL BACKUP 03:00	DIFFERENTIAL BACKUP 03:00	DIFFERENTIAL BACKUP 03:00	DIFFERENTIAL BACKUP 03:00	DIFFERENTIAL BACKUP 03:00	DIFFERENTIAL BACKUP 03:00
DIFFERENTIAL BACKUP 03:30	DIFFERENTIAL BACKUP 03:30	DIFFERENTIAL BACKUP 03:30	DIFFERENTIAL BACKUP 03:30	DIFFERENTIAL BACKUP 03:30	DIFFERENTIAL BACKUP 03:30	DIFFERENTIAL BACKUP 03:30
DIFFERENTIAL BACKUP 04:00	DIFFERENTIAL BACKUP 04:00	DIFFERENTIAL BACKUP 04:00	DIFFERENTIAL BACKUP 04:00	DIFFERENTIAL BACKUP 04:00	DIFFERENTIAL BACKUP 04:00	DIFFERENTIAL BACKUP 04:00
DIFFERENTIAL BACKUP 04:30	DIFFERENTIAL BACKUP 04:30	DIFFERENTIAL BACKUP 04:30	DIFFERENTIAL BACKUP 04:30	DIFFERENTIAL BACKUP 04:30	DIFFERENTIAL BACKUP 04:30	DIFFERENTIAL BACKUP 04:30
DIFFERENTIAL BACKUP 05:00	DIFFERENTIAL BACKUP 05:00	DIFFERENTIAL BACKUP 05:00	DIFFERENTIAL BACKUP 05:00	DIFFERENTIAL BACKUP 05:00	DIFFERENTIAL BACKUP 05:00	DIFFERENTIAL BACKUP 05:00
DIFFERENTIAL BACKUP 05:30	DIFFERENTIAL BACKUP 05:30	DIFFERENTIAL BACKUP 05:30	DIFFERENTIAL BACKUP 05:30	DIFFERENTIAL BACKUP 05:30	DIFFERENTIAL BACKUP 05:30	DIFFERENTIAL BACKUP 05:30
DIFFERENTIAL BACKUP 06:00	DIFFERENTIAL BACKUP 06:00	DIFFERENTIAL BACKUP 06:00	DIFFERENTIAL BACKUP 06:00	DIFFERENTIAL BACKUP 06:00	DIFFERENTIAL BACKUP 06:00	DIFFERENTIAL BACKUP 06:00