

**DOKUZ EYLUL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPT. OF COMPUTER ENGINEERING**

HOTEL MANAGEMENT SYSTEM

# CME 3201 Database Management Systems

# Term Project Report

**Phase V**

2019-2020 FALL

2016510036-Türkü Su KAPLAN

2016510082-Mustafa Çağatay ÖNAL

IZMIR

26.12.2019

1-INTRODUCTION

Hotel management system is used for keeping records of available rooms and maintenance of these records for reservations.

System keeps the hotel information such as; hotel name, location, hotel’s review, capacity and available rooms for specific dates, prices and room type.

User should reach the hotel room he/she wants within some dates by searching and customer information will be kept in the database.

2-OVERVIEW

System should be helpful for maintaining and handling hotel room reservations. Main functionalities of this system consist of;

* Online reservation for guests
* Check-in/check-out
* Customer/Employee information
* Facilities
* Room information (status, date, occupancy)
* Billing system

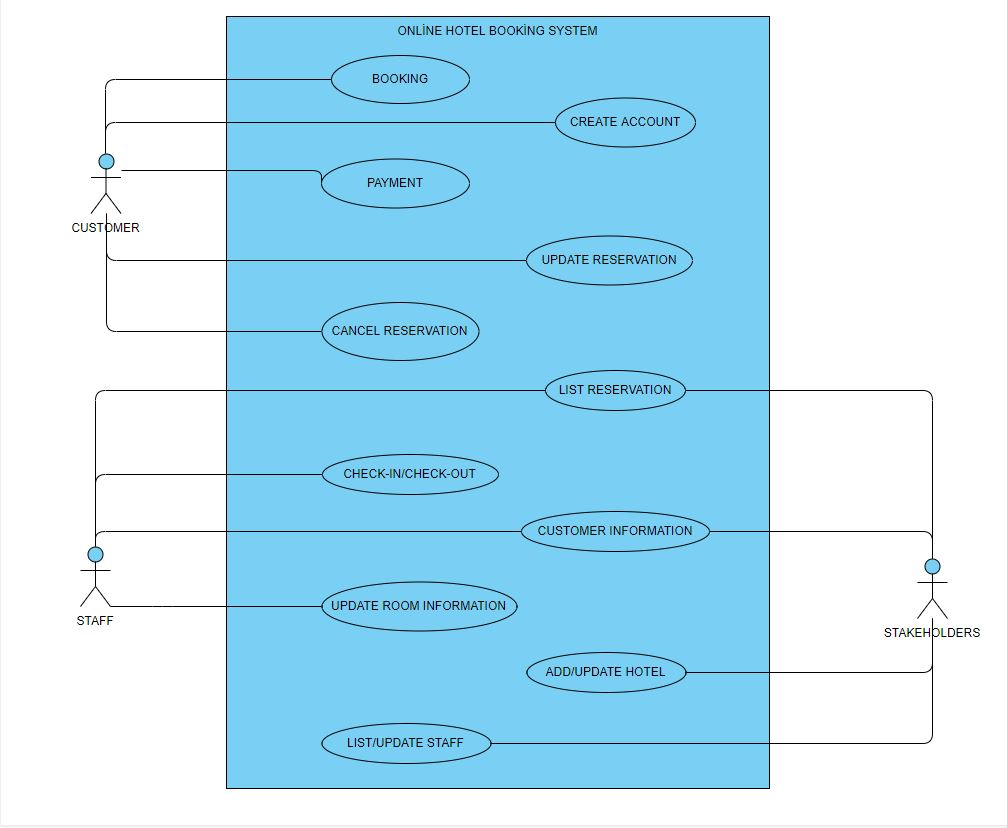
System has three different modules: Customer, Management and Administration Module.

Customer module will have the functionality of doing reservation online by using date and location. The hotel rooms that are available for this customer’s search will be shown.

User can select and reserve the room according to his/her needs.

Management module’s functions is to list all the rooms and their status. Manager can see its hotel’s booked rooms and guest information. Updates on rooms can be made within management module.

Administration module is the main module for hotel database system. Adding or updating a hotel can be made from this module. Hotel name, location, reviews about the hotel and its facilities are some of the entities/attributes that transactions can occur from.



*Use-case diagram*

3 – ASSUMPTIONS/CONSTRAINTS/RISKS

3.1 Assumptions:

1. System will be used by customers for booking hotel rooms, stakeholder that update/modify their hotel room information and hotel staff for editing/updating.,
2. When a customer uses the system for searching a hotel room, search results will show the hotel’s price, rating, facility info and availability during check-in/check-out dates.
3. Customer should be able to cancel the reservation he/she made within its account.
4. System should work in every browser possible (Internet Explorer, Google Chrome, Firefox, Safari, Opera etc...) and it should work anytime of the day.
5. System’s hardware requirements consist of any physical drive that takes input from end-user, processing, saving information and managing it.
6. System should allow multiple transactions at a time.

3.2 Constraints:

User needs a good Internet connection to connect to the system.

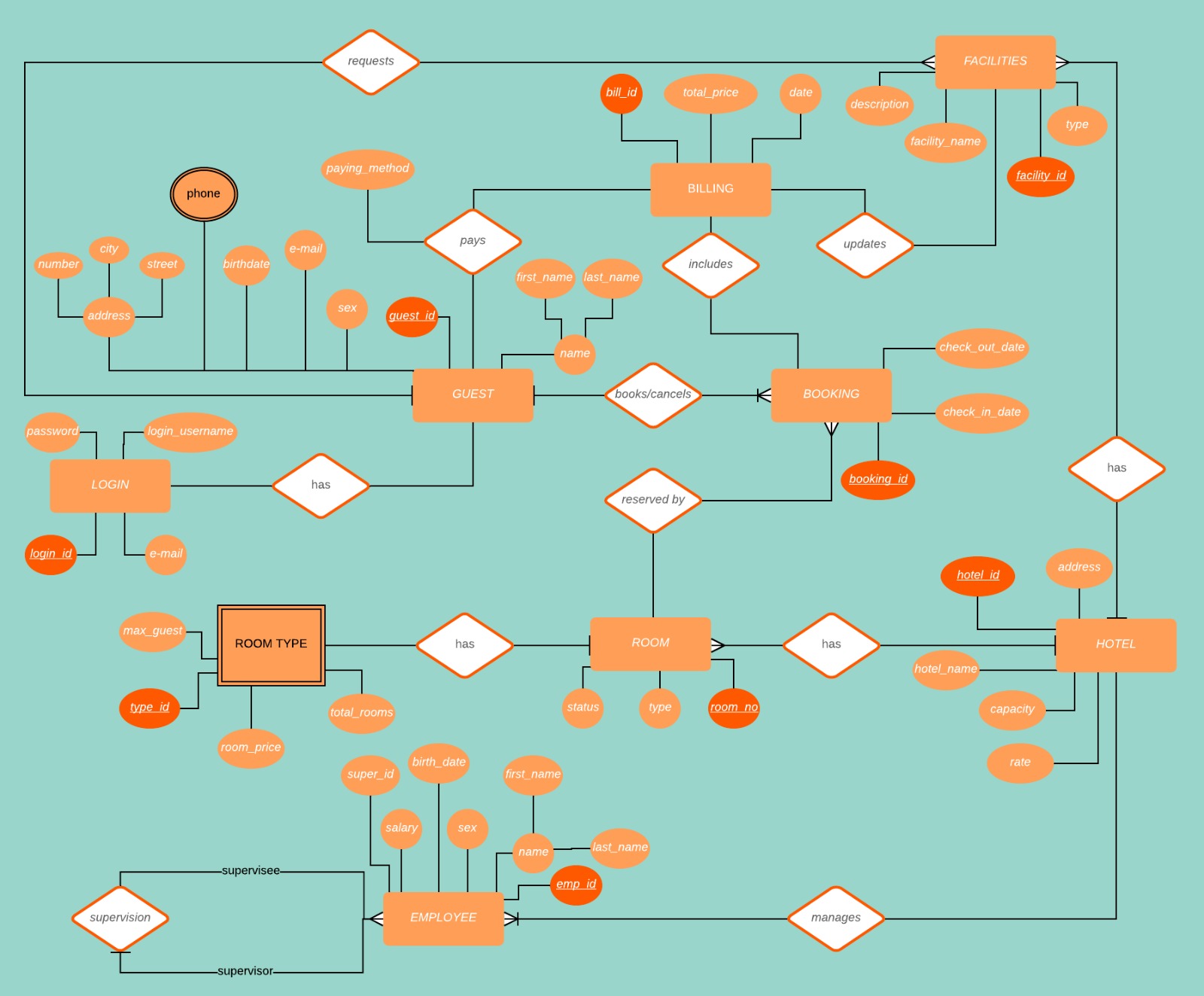
**3.3 Risks:**

Risks of the system can be categorized such as:

* Size of the database risks: can cause slower search / insert time if data is bigger than expected
* End-user characteristics: problems with the user-system communication
* Hotel management/staff characteristics: risks associated with system-staff communication

5 – DETAILED SYSTEM DESIGN

5.1 ENTITY-RELATIONSHIP DIAGRAM:



**5.2 RELATIONAL ALGEBRA EXPRESSIONS**

* Selects names from guests where sex is 'male' and 'age' is > 50

SELECT guest.guest.name

FROM hotel\_booking\_app.guest

WHERE guest.sex = ‘male’ && guest.birthdate < 1959

σguest\_name(sex = "male" & birthdate < “1959”(GUEST)

* Select all guests that booked an hotel

SELECT guest.\*

FROM hotel\_booking\_app.guest, hotel\_booking\_app.booking

WHERE guest.guest\_ID = booking.guest\_ID

πguest.\*(GUEST)

* Get all employees where salary is more than $1500

SELECT employee.emp\_ID, employee.emp\_name

FROM hotel\_booking\_app.employee

HAVING salary > 1500

πemp\_name, guest\_id(σsalary>1500(EMPLOYEE))

* Get name column of guests

SELECT guest.guest\_name

FROM hotel\_booking\_app.guest

πguest\_name(GUEST)

* Select all employees that work under Hilton Hotel’s supervisor

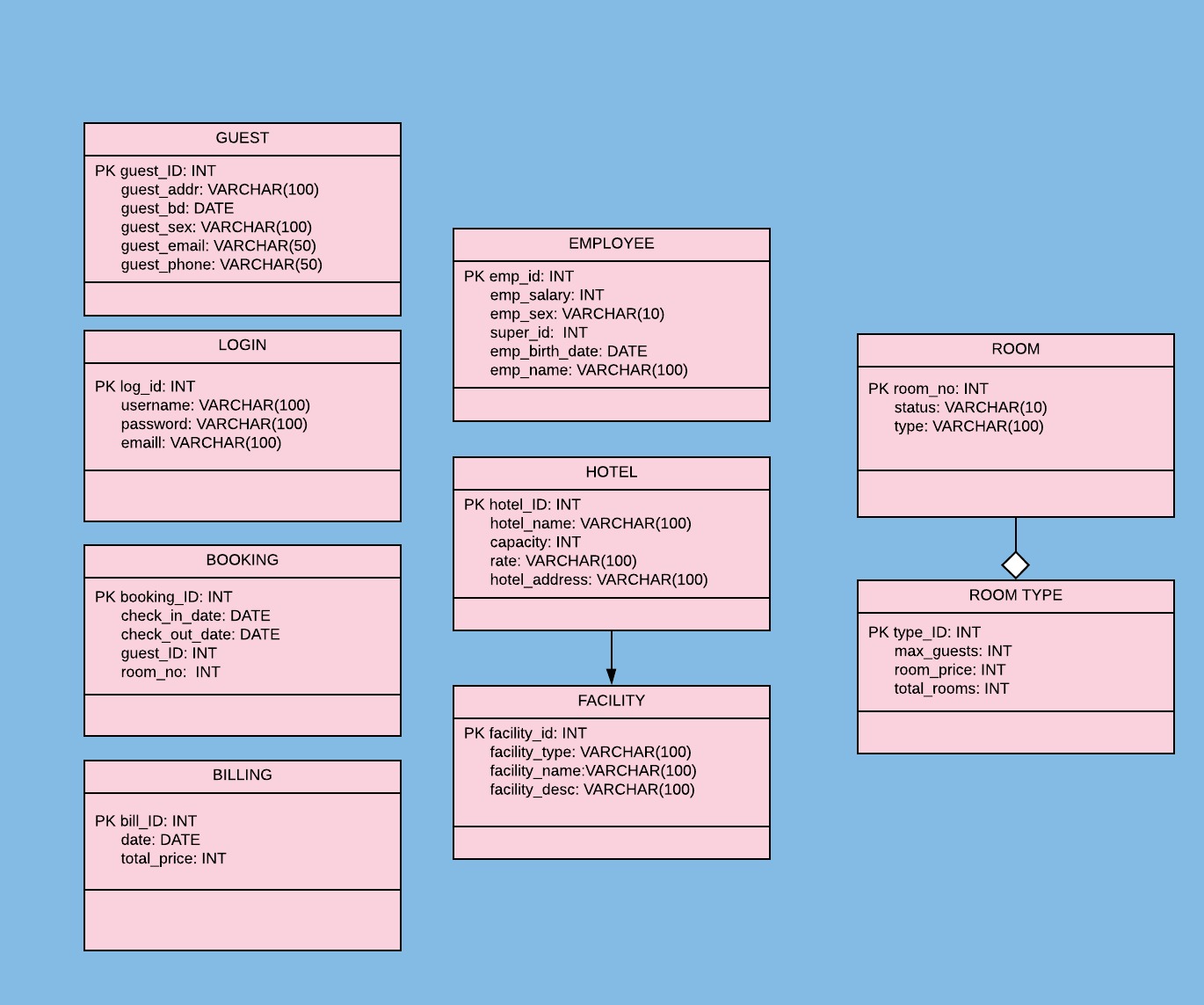
SELECT Employee.employee\_name FROM hotel\_booking\_app.Employee

WHERE Employee.super\_id IN (

SELECT Hotel.super\_id FROM Hotel )

πemployee\_name(σhotel\_name=”Hilton”(Hotel⨝superciver\_id=employee\_id Superviser))

5.3 CLASS DIAGRAM



**5.4 CRUD MATRIX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Relations/Operations | CREATE | UPDATE | READ | DELETE |
| Administration | X | X | X | X |
| Management |  | X | X | X |
| Guest |  |  | X |  |

**5.5 DATABASE SCHEMA**

duvar, bina içeren bir resim

Açıklama otomatik olarak oluşturuldu

APPENDIX A: SQL Statements

CREATE DATABASE [HotelBookingApp]

CONTAINMENT = NONE

ON PRIMARY

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT

/\*\*\*\*\*\* Object: Schema [hotel\_booking\_app] Script Date: 25.12.2019 22:24:53 \*\*\*\*\*\*/

CREATE SCHEMA [hotel\_booking\_app]

CREATE TABLE [dbo].[Hotel](

[hotel\_id] [int] IDENTITY(1,1) NOT NULL,

[hotel\_name] [text] NOT NULL,

[hotel\_address] [text] NULL,

[capacity] [int] NULL,

[hotel\_city] [text] NULL,

CONSTRAINT [PK\_Hotel] PRIMARY KEY CLUSTERED

(

[hotel\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: View [dbo].[HotelViewLessThan600] Script Date: 25.12.2019 22:24:53 \*\*\*\*\*\*/

CREATE VIEW [dbo].[HotelViewLessThan600]

AS

SELECT hotel\_name, hotel\_address, capacity

FROM dbo.Hotel

WHERE (capacity < 600)

GO

CREATE TABLE [dbo].[Guest](

[guest\_id] [int] IDENTITY(1,1) NOT NULL,

[first\_name] [varchar](100) NOT NULL,

[last\_name] [varchar](100) NULL,

[sex] [varchar](100) NULL,

[email] [varchar](100) NULL,

[address\_number] [int] NULL,

[address\_city] [varchar](100) NULL,

[address\_street] [varchar](100) NULL,

[phone] [varchar](100) NULL,

[birtdate] [date] NULL,

[login\_id] [int] NULL,

CONSTRAINT [PK\_Guest] PRIMARY KEY CLUSTERED

(

[guest\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE VIEW [dbo].[GroupListView]

AS

SELECT first\_name, last\_name, login\_id

FROM dbo.Guest

GO

CREATE VIEW [dbo].[HotelsInTurkey] as

SELECT hotel\_name, hotel\_address

FROM dbo.Hotel

WHERE hotel\_city LIKE 'Turkey';

GO

CREATE TABLE [dbo].[Employee](

[emp\_id] [int] IDENTITY(1,1) NOT NULL,

[emp\_first\_name] [varchar](100) NOT NULL,

[emp\_last\_name] [varchar](100) NOT NULL,

[emp\_sex] [varchar](100) NOT NULL,

[emp\_birthdate] [date] NULL,

[emp\_salary] [int] NOT NULL,

[super\_id] [int] NULL,

[hotel\_id] [int] NULL,

CONSTRAINT [PK\_Employee] PRIMARY KEY CLUSTERED

(

[emp\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE VIEW [dbo].[NormalEmployees]

AS

SELECT emp\_id, emp\_first\_name, emp\_last\_name

FROM dbo.Employee

WHERE (super\_id IS NOT NULL)

GO

CREATE TABLE [dbo].[\_\_EFMigrationsHistory](

[MigrationId] [nvarchar](150) NOT NULL,

[ProductVersion] [nvarchar](32) NOT NULL,

CONSTRAINT [PK\_\_\_EFMigrationsHistory] PRIMARY KEY CLUSTERED

(

[MigrationId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Billing](

[bill\_id] [int] IDENTITY(1,1) NOT NULL,

[booking\_id] [int] NULL,

[total\_price] [int] NULL,

[bill\_date] [date] NULL,

CONSTRAINT [PK\_Billing] PRIMARY KEY CLUSTERED

(

[bill\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Booking](

[booking\_id] [int] IDENTITY(1,1) NOT NULL,

[guest\_id] [int] NULL,

[check\_in\_date] [date] NOT NULL,

[check\_out\_date] [date] NOT NULL,

CONSTRAINT [PK\_Booking] PRIMARY KEY CLUSTERED

(

[booking\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Facilities] Script Date: 25.12.2019 22:24:53 \*\*\*\*\*\*/

GO

CREATE TABLE [dbo].[Facilities](

[facility\_id] [int] IDENTITY(1,1) NOT NULL,

[facility\_name] [varchar](100) NOT NULL,

[facility\_desc] [varchar](100) NULL,

[facility\_price] [int] NULL,

[hotel\_id] [int] NULL,

CONSTRAINT [PK\_facility] PRIMARY KEY CLUSTERED

(

[facility\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Hotel\_Rating](

[hotelrating\_id] [int] IDENTITY(1,1) NOT NULL,

[hotel\_id] [int] NULL,

[guest\_id] [int] NULL,

[rate] [int] NULL,

CONSTRAINT [PK\_Hotel\_Rating] PRIMARY KEY CLUSTERED

(

[hotelrating\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Login](

[login\_id] [int] IDENTITY(1,1) NOT NULL,

[password] [text] NOT NULL,

[username] [text] NOT NULL,

[email] [text] NOT NULL,

CONSTRAINT [PK\_Login] PRIMARY KEY CLUSTERED

(

[login\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

CREATE TABLE [dbo].[Room](

[room\_id] [int] IDENTITY(1,1) NOT NULL,

[room\_status] [text] NULL,

[hotel\_id] [int] NULL,

[guest\_id] [int] NULL,

CONSTRAINT [PK\_Room] PRIMARY KEY CLUSTERED

(

[room\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

CREATE TABLE [dbo].[RoomType](

[type\_id] [int] IDENTITY(1,1) NOT NULL,

[maxguest] [int] NULL,

[room\_price] [int] NULL,

[room\_id] [int] NULL,

[room\_type] [int] NULL,

CONSTRAINT [PK\_RoomType] PRIMARY KEY CLUSTERED

(

[type\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Billing] WITH CHECK ADD CONSTRAINT [fk\_booking] FOREIGN KEY([booking\_id])

REFERENCES [dbo].[Booking] ([booking\_id])

GO

ALTER TABLE [dbo].[Billing] CHECK CONSTRAINT [fk\_booking]

GO

ALTER TABLE [dbo].[Booking] WITH CHECK ADD CONSTRAINT [fk\_guest] FOREIGN KEY([guest\_id])

REFERENCES [dbo].[Guest] ([guest\_id])

GO

ALTER TABLE [dbo].[Booking] CHECK CONSTRAINT [fk\_guest]

GO

ALTER TABLE [dbo].[Employee] WITH CHECK ADD FOREIGN KEY([hotel\_id])

REFERENCES [dbo].[Hotel] ([hotel\_id])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[Facilities] WITH CHECK ADD CONSTRAINT [fk\_facility] FOREIGN KEY([hotel\_id])

REFERENCES [dbo].[Hotel] ([hotel\_id])

GO

ALTER TABLE [dbo].[Facilities] CHECK CONSTRAINT [fk\_facility]

GO

ALTER TABLE [dbo].[Guest] WITH CHECK ADD CONSTRAINT [fk\_login] FOREIGN KEY([login\_id])

REFERENCES [dbo].[Login] ([login\_id])

GO

ALTER TABLE [dbo].[Guest] CHECK CONSTRAINT [fk\_login]

GO

ALTER TABLE [dbo].[Hotel\_Rating] WITH CHECK ADD CONSTRAINT [fk\_guest\_rtng] FOREIGN KEY([guest\_id])

REFERENCES [dbo].[Guest] ([guest\_id])

GO

ALTER TABLE [dbo].[Hotel\_Rating] CHECK CONSTRAINT [fk\_guest\_rtng]

GO

ALTER TABLE [dbo].[Hotel\_Rating] WITH CHECK ADD CONSTRAINT [fk\_hotel\_rtng] FOREIGN KEY([hotel\_id])

REFERENCES [dbo].[Hotel] ([hotel\_id])

GO

ALTER TABLE [dbo].[Hotel\_Rating] CHECK CONSTRAINT [fk\_hotel\_rtng]

GO

ALTER TABLE [dbo].[Room] WITH CHECK ADD CONSTRAINT [fk\_room\_guest] FOREIGN KEY([guest\_id])

REFERENCES [dbo].[Guest] ([guest\_id])

GO

ALTER TABLE [dbo].[Room] CHECK CONSTRAINT [fk\_room\_guest]

GO

ALTER TABLE [dbo].[Room] WITH CHECK ADD CONSTRAINT [fk\_room\_hotel] FOREIGN KEY([hotel\_id])

REFERENCES [dbo].[Hotel] ([hotel\_id])

GO

ALTER TABLE [dbo].[Room] CHECK CONSTRAINT [fk\_room\_hotel]

GO

ALTER TABLE [dbo].[RoomType] WITH CHECK ADD CONSTRAINT [fk\_roomtyperoom] FOREIGN KEY([room\_id])

REFERENCES [dbo].[Room] ([room\_id])

GO

ALTER TABLE [dbo].[RoomType] CHECK CONSTRAINT [fk\_roomtyperoom]

GO

ALTER TABLE [dbo].[Hotel\_Rating] WITH CHECK ADD CONSTRAINT [check\_rate] CHECK (([rate]<(6)))

GO

ALTER TABLE [dbo].[Hotel\_Rating] CHECK CONSTRAINT [check\_rate]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ChangeManagerSalary] Script Date: 25.12.2019 22:24:53 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <hotel\_booking\_admin>

-- Create date: <25/12/2019>

-- Description: <HotelBookingApplication>

-- =============================================

CREATE PROCEDURE [dbo].[ChangeManagerSalary]

AS

BEGIN

UPDATE dbo.Employee SET dbo.Employee.emp\_salary = 5000 WHERE dbo.Employee.super\_id IS NULL;

END

GO

-- =============================================

-- Author: <hotel\_booking\_admin>

-- Create date: <25/12/2019>

-- Description: <HotelBookingApplication>

-- =============================================

CREATE PROCEDURE [dbo].[DeleteGuest]

@GuestID nvarchar(10)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

DELETE FROM dbo.Guest

WHERE guest\_id = @GuestID

RETURN @@ROWCOUNT

END

GO

-- =============================================

-- Author: <hotel\_booking\_admin>

-- Create date: <25/12/2019>

-- Description: <HotelBookingApplication>

-- =============================================

CREATE PROCEDURE [dbo].[GetRoom\_Hotel]

@HotelID INT

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT dbo.Room.room\_id, dbo.Room.room\_status FROM dbo.Room where room.hotel\_id = @HotelID;

END

GO

-- =============================================

-- Author: <hotel\_booking\_admin>

-- Create date: <25/12/2019>

-- Description: <HotelBookingApplication>

-- =============================================

CREATE PROCEDURE [dbo].[GetSpecificCity]

@CityName text

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

Select \* From dbo.Hotel where hotel\_city like @CityName

-- Insert statements for procedure here

END

GO

-- =============================================

-- Author: <hotel\_booking\_admin>

-- Create date: <25/12/2019>

-- Description: <HotelBookingApplication>

-- =============================================

CREATE PROCEDURE [dbo].[GuestCount]

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

SELECT COUNT(\*) FROM dbo.Guest;

END

GO

-- =============================================

-- Author: <hotel\_booking\_admin>

-- Create date: <25/12/2019>

-- Description: <HotelBookingApplication>

-- =============================================

CREATE PROCEDURE [dbo].[RaiseSalaryBy50]

@EmployeeID INT

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

UPDATE dbo.Employee SET emp\_salary = emp\_salary \* 1.5

WHERE Employee.emp\_id = @EmployeeID

END

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]

Begin DesignProperties =

Begin PaneConfigurations =

Begin PaneConfiguration = 0

NumPanes = 4

Configuration = "(H (1[40] 4[20] 2[20] 3) )"

End

Begin PaneConfiguration = 1

NumPanes = 3

Configuration = "(H (1 [50] 4 [25] 3))"

End

Begin PaneConfiguration = 2

NumPanes = 3

Configuration = "(H (1 [50] 2 [25] 3))"

End

Begin PaneConfiguration = 3

NumPanes = 3

Configuration = "(H (4 [30] 2 [40] 3))"

End

Begin PaneConfiguration = 4

NumPanes = 2

Configuration = "(H (1 [56] 3))"

End

Begin PaneConfiguration = 5

NumPanes = 2

Configuration = "(H (2 [66] 3))"

End

Begin PaneConfiguration = 6

NumPanes = 2

Configuration = "(H (4 [50] 3))"

End

Begin PaneConfiguration = 7

NumPanes = 1

Configuration = "(V (3))"

End

Begin PaneConfiguration = 8

NumPanes = 3

Configuration = "(H (1[56] 4[18] 2) )"

End

Begin PaneConfiguration = 9

NumPanes = 2

Configuration = "(H (1 [75] 4))"

End

Begin PaneConfiguration = 10

NumPanes = 2

Configuration = "(H (1[66] 2) )"

End

Begin PaneConfiguration = 11

NumPanes = 2

Configuration = "(H (4 [60] 2))"

End

Begin PaneConfiguration = 12

NumPanes = 1

Configuration = "(H (1) )"

End

Begin PaneConfiguration = 13

NumPanes = 1

Configuration = "(V (4))"

End

Begin PaneConfiguration = 14

NumPanes = 1

Configuration = "(V (2))"

End

ActivePaneConfig = 0

End

Begin DiagramPane =

Begin Origin =

Top = 0

Left = 0

End

Begin Tables =

Begin Table = "Guest"

Begin Extent =

Top = 7

Left = 48

Bottom = 170

Right = 251

End

DisplayFlags = 280

TopColumn = 7

End

End

End

Begin SQLPane =

End

Begin DataPane =

Begin ParameterDefaults = ""

End

End

Begin CriteriaPane =

Begin ColumnWidths = 11

Column = 1440

Alias = 900

Table = 1176

Output = 720

Append = 1400

NewValue = 1170

SortType = 1356

SortOrder = 1416

GroupBy = 1356

Filter = 1356

Or = 1350

Or = 1350

Or = 1350

End

End

End

' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'GroupListView'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'GroupListView'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]

Begin DesignProperties =

Begin PaneConfigurations =

Begin PaneConfiguration = 0

NumPanes = 4

Configuration = "(H (1[40] 4[20] 2[20] 3) )"

End

Begin PaneConfiguration = 1

NumPanes = 3

Configuration = "(H (1 [50] 4 [25] 3))"

End

Begin PaneConfiguration = 2

NumPanes = 3

Configuration = "(H (1 [50] 2 [25] 3))"

End

Begin PaneConfiguration = 3

NumPanes = 3

Configuration = "(H (4 [30] 2 [40] 3))"

End

Begin PaneConfiguration = 4

NumPanes = 2

Configuration = "(H (1 [56] 3))"

End

Begin PaneConfiguration = 5

NumPanes = 2

Configuration = "(H (2 [66] 3))"

End

Begin PaneConfiguration = 6

NumPanes = 2

Configuration = "(H (4 [50] 3))"

End

Begin PaneConfiguration = 7

NumPanes = 1

Configuration = "(V (3))"

End

Begin PaneConfiguration = 8

NumPanes = 3

Configuration = "(H (1[56] 4[18] 2) )"

End

Begin PaneConfiguration = 9

NumPanes = 2

Configuration = "(H (1 [75] 4))"

End

Begin PaneConfiguration = 10

NumPanes = 2

Configuration = "(H (1[66] 2) )"

End

Begin PaneConfiguration = 11

NumPanes = 2

Configuration = "(H (4 [60] 2))"

End

Begin PaneConfiguration = 12

NumPanes = 1

Configuration = "(H (1) )"

End

Begin PaneConfiguration = 13

NumPanes = 1

Configuration = "(V (4))"

End

Begin PaneConfiguration = 14

NumPanes = 1

Configuration = "(V (2))"

End

ActivePaneConfig = 0

End

Begin DiagramPane =

Begin Origin =

Top = 0

Left = 0

End

Begin Tables =

Begin Table = "Hotel"

Begin Extent =

Top = 7

Left = 48

Bottom = 170

Right = 242

End

DisplayFlags = 280

TopColumn = 1

End

End

End

Begin SQLPane =

End

Begin DataPane =

Begin ParameterDefaults = ""

End

End

Begin CriteriaPane =

Begin ColumnWidths = 11

Column = 1440

Alias = 900

Table = 1170

Output = 720

Append = 1400

NewValue = 1170

SortType = 1350

SortOrder = 1410

GroupBy = 1350

Filter = 1350

Or = 1350

Or = 1350

Or = 1350

End

End

End

' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'HotelViewLessThan600'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'HotelViewLessThan600'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]

Begin DesignProperties =

Begin PaneConfigurations =

Begin PaneConfiguration = 0

NumPanes = 4

Configuration = "(H (1[40] 4[20] 2[20] 3) )"

End

Begin PaneConfiguration = 1

NumPanes = 3

Configuration = "(H (1 [50] 4 [25] 3))"

End

Begin PaneConfiguration = 2

NumPanes = 3

Configuration = "(H (1 [50] 2 [25] 3))"

End

Begin PaneConfiguration = 3

NumPanes = 3

Configuration = "(H (4 [30] 2 [40] 3))"

End

Begin PaneConfiguration = 4

NumPanes = 2

Configuration = "(H (1 [56] 3))"

End

Begin PaneConfiguration = 5

NumPanes = 2

Configuration = "(H (2 [66] 3))"

End

Begin PaneConfiguration = 6

NumPanes = 2

Configuration = "(H (4 [50] 3))"

End

Begin PaneConfiguration = 7

NumPanes = 1

Configuration = "(V (3))"

End

Begin PaneConfiguration = 8

NumPanes = 3

Configuration = "(H (1[56] 4[18] 2) )"

End

Begin PaneConfiguration = 9

NumPanes = 2

Configuration = "(H (1 [75] 4))"

End

Begin PaneConfiguration = 10

NumPanes = 2

Configuration = "(H (1[66] 2) )"

End

Begin PaneConfiguration = 11

NumPanes = 2

Configuration = "(H (4 [60] 2))"

End

Begin PaneConfiguration = 12

NumPanes = 1

Configuration = "(H (1) )"

End

Begin PaneConfiguration = 13

NumPanes = 1

Configuration = "(V (4))"

End

Begin PaneConfiguration = 14

NumPanes = 1

Configuration = "(V (2))"

End

ActivePaneConfig = 0

End

Begin DiagramPane =

Begin Origin =

Top = 0

Left = 0

End

Begin Tables =

Begin Table = "Employee"

Begin Extent =

Top = 7

Left = 48

Bottom = 170

Right = 247

End

DisplayFlags = 280

TopColumn = 3

End

End

End

Begin SQLPane =

End

Begin DataPane =

Begin ParameterDefaults = ""

End

End

Begin CriteriaPane =

Begin ColumnWidths = 11

Column = 1440

Alias = 900

Table = 1170

Output = 720

Append = 1400

NewValue = 1170

SortType = 1350

SortOrder = 1410

GroupBy = 1350

Filter = 1872

Or = 1350

Or = 1350

Or = 1350

End

End

End

' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'NormalEmployees'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'NormalEmployees'

GO

USE [master]

GO

ALTER DATABASE [HotelBookingApp] SET READ\_WRITE

GO

Complex Sql Queries:

--find 5 top booking hotels

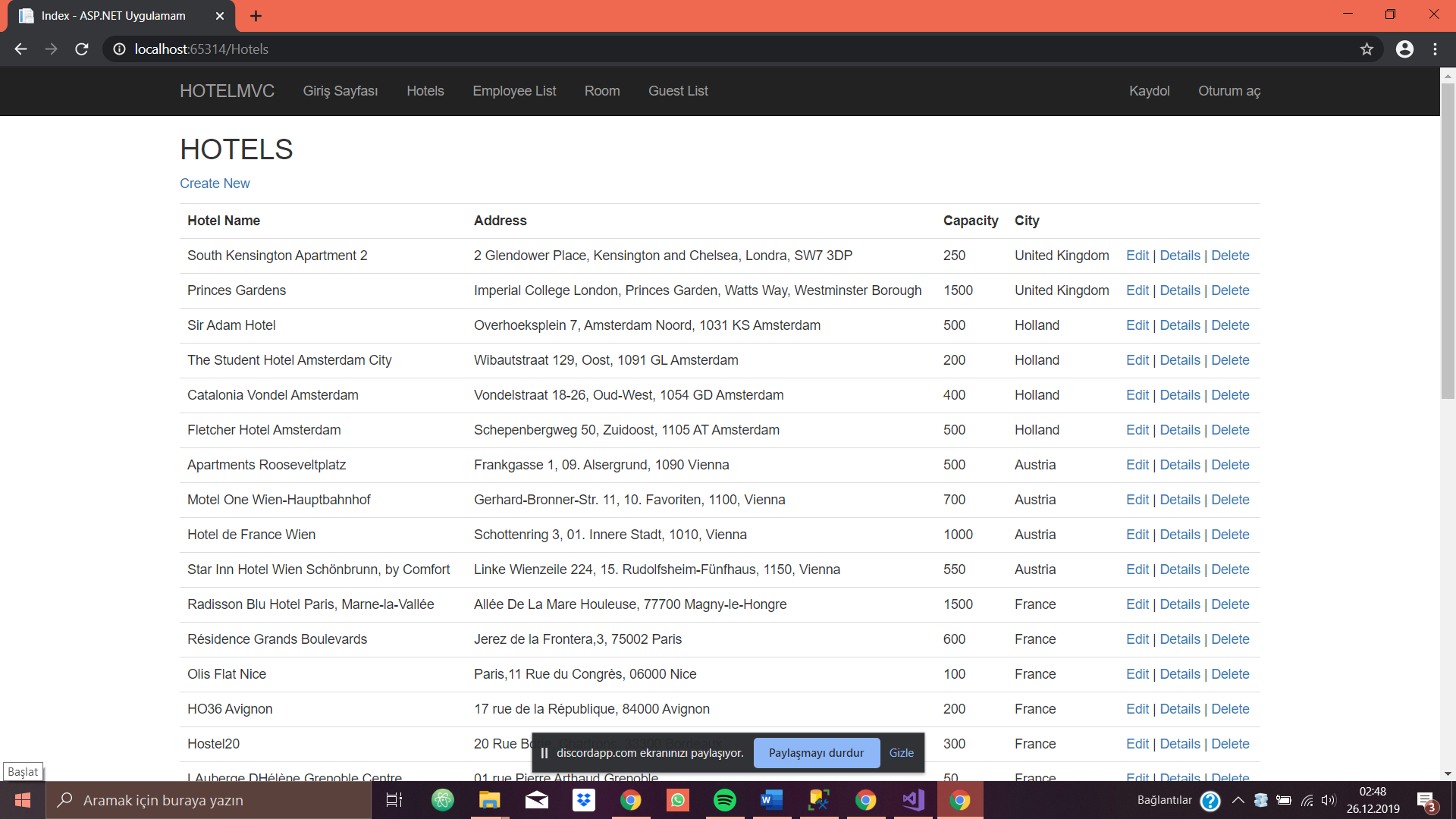
SELECT h.\* FROM hotel AS h WHERE h.hotel\_id IN (SELECT TOP 5 r.hotel\_id FROM guest g, room r WHERE g.guest\_id = r.guest\_id GROUP BY r.hotel\_id ORDER BY COUNT(g.guest\_id) ASC );

--find second highest salary of employee

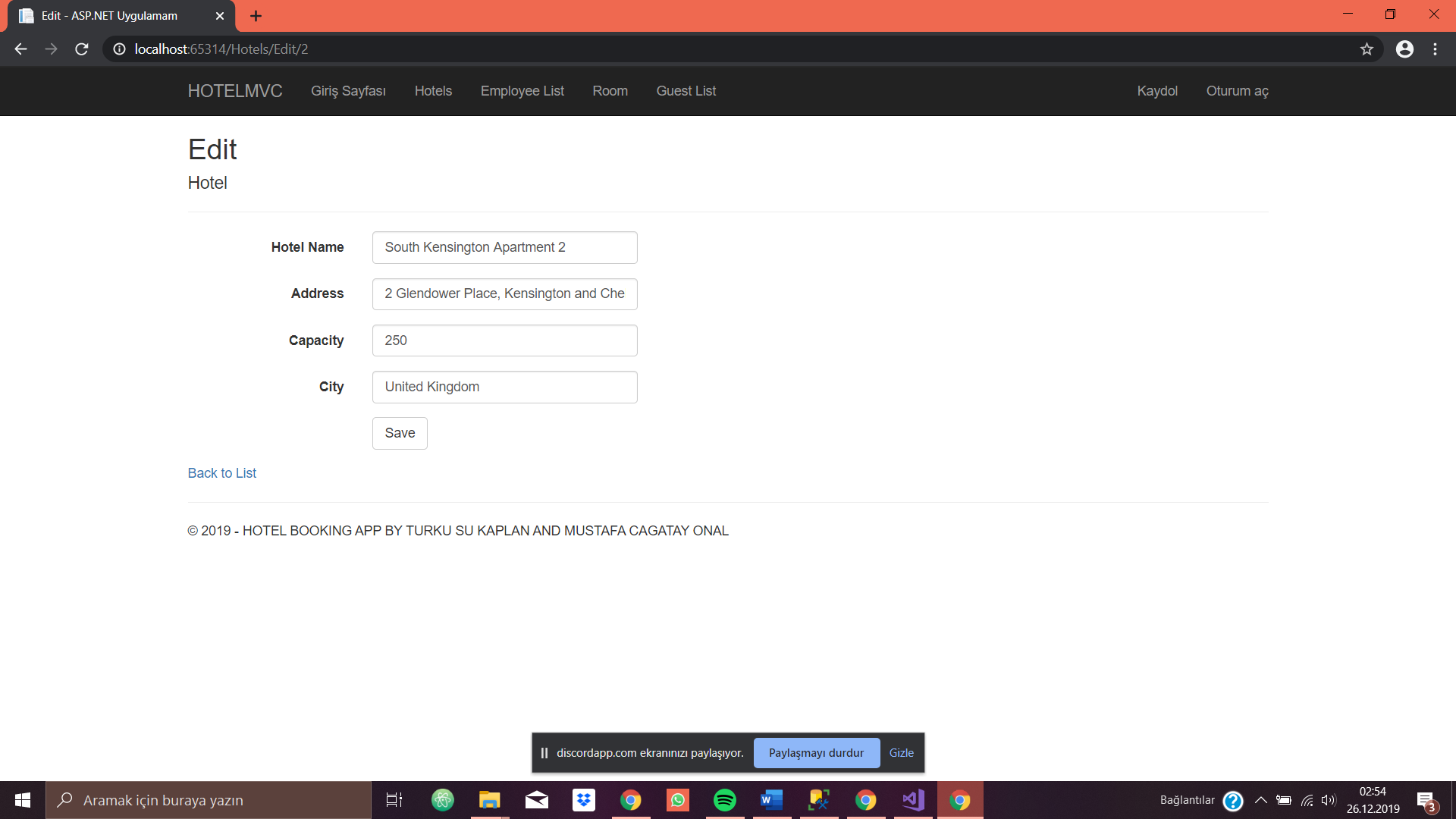
SELECT MIN(emp\_salary)FROM(SELECT DISTINCT emp\_salary FROM Employee ORDER BY emp\_salary DESC)WHERE ROWNUM<=2;

Appendix B: Screenshots

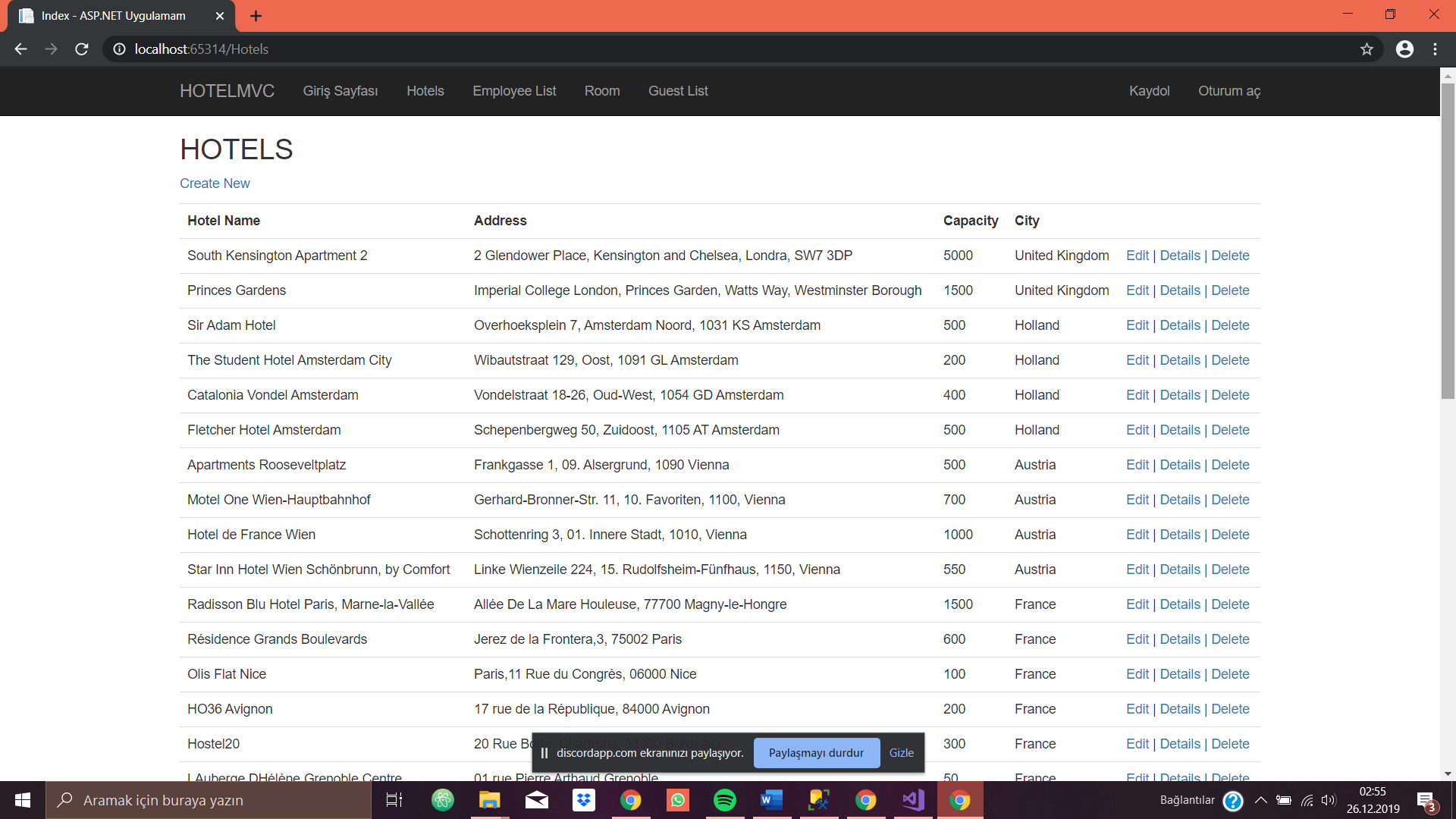
* Hotel list:

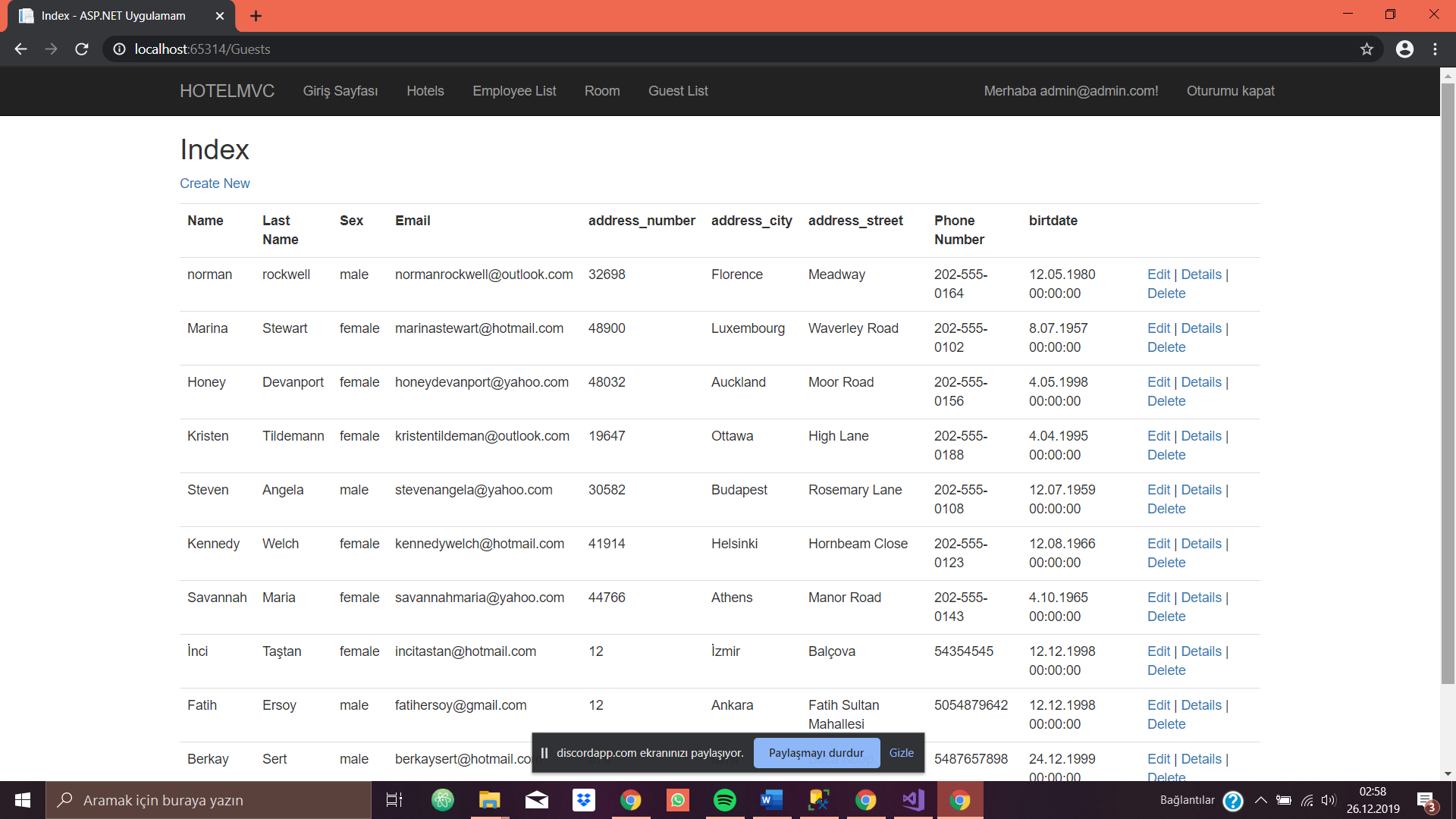


* Edit operation:



* After edit operation:





* Create room operation:

