

DM - Pretest

1. Create a database named **dbPretest**
2. Create the following tables in Database **dbPretest**

a. Table "**tbRoom**"

Field name	Data type	Constraint description
RoomNo	int	Primary key
Type	Varchar(20)	Only accept values : "VIP", "Double" or "Single"
UnitPrice	Money	0 <= price < 1000

Define Check constraints on field Type, UnitPrice as column description

b. Table "**tbBooking**"

Field name	Data type	Constraint description
BookingNo	int	
RoomNo	int	Foreign key
TouristName	Varchar(20)	Not null
DateFrom	DateTime	
DateTo	DateTime	dateTo must be greater than dateFrom

*Define a Composite Primary key (**BookingNo, RoomNo**) on Booking table.*

c. Insert some records into tables "**tbRoom**" and "**tbBooking**" :

RoomNo	Type	Unitprice
101	Single	100
102	Single	100
103	Double	250
201	Double	250
202	Double	300
203	Single	150
301	VIP	900

BookingNo	RoomNo	TouristName	DateFrom	DateTo
1	101	Julia	12/11/2014	14/11/2014
1	103	Julia	12/12/2014	13/12/2014
2	301	Bill	10/01/2015	14/01/2015
3	201	Ana	12/01/2015	14/01/2015
3	202	Ana	12/01/2015	14/01/2015

3. Create a clustered index **ixName** on column *TouristName* of table **tbBooking**
4. Create a non-clustered index **ixType** on column *Type* of table **tbRoom**
5. Create a view **vwBooking** to see the information about bookings in year 2014 which contain the following columns:
BookingNo, TouristName, RoomNo, Type, UnitPrice, DateFrom, DateTo.
The definition of view must be encrypted.
6. Create a stored procedure name **uspPriceDecrease** will down the unit price of double rooms a given amount (input parameter). If non value given, display a list of rooms, sorted by price .
7. Create a stored procedure name **uspSpecificPriceIncrease** will increment the unit price of a given room (input parameter) by a given amount (input parameter) and return the number of rooms (output parameter) which have room rate above 250.

8. Create a trigger named **tgBookingRoom** that allows one booking order having 3 rooms maximum.
9. Create a trigger named **tgRoomUpdate** that doing the following (using try-catch structure) :
If new price is equal to 0 and this room has not existed in **tbBooking**, then remove it from **tbRoom** table
else display an error message and roll back transaction.