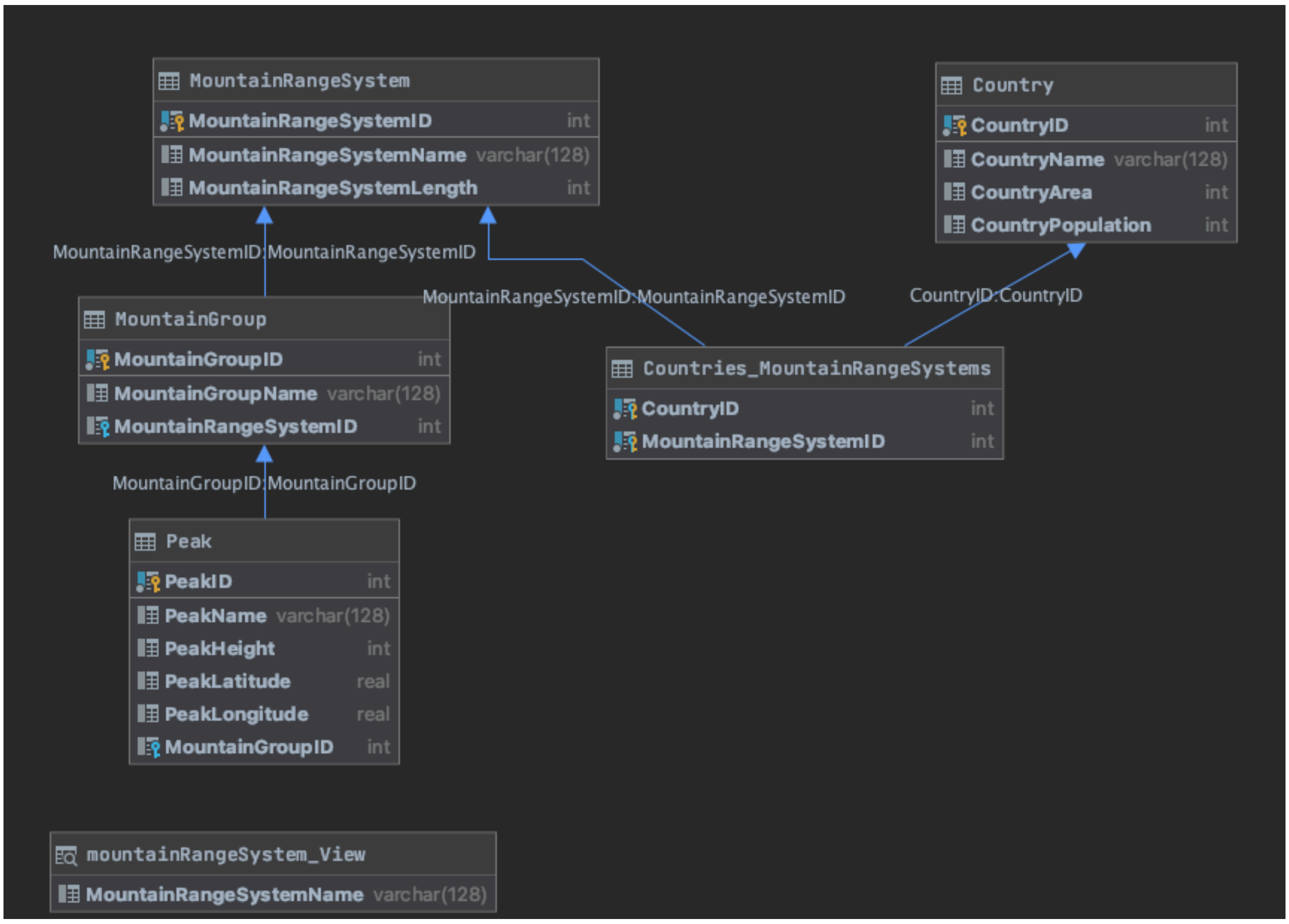
```
/*
 countries - name, area, population, more than one mountain range system
 mountain range systems - name, length, can spread along multiple countries, can include
multiple mountain groups
 mountain groups name, belongs to a mountain range system
 peaks - name, height(meters), latitude, longitude (real), belongs to a mountain group
=> countries, mountainRangeSystem = m:n
=> mountainRangeSystems, mountainGroups -> 1:n
=> mountainGroup, peaks -> 1:n
 */

--a) table creation
Create table Country
   (CountryID int primary key IDENTITY(1,1),
   CountryName varchar(128),
   CountryArea int,
   CountryPopulation int)

Create table MountainRangeSystem
   (MountainRangeSystemID int primary key identity (1,1),
   MountainRangeSystemName varchar(128),
   MountainRangeSystemLength int)

Create table Countries_MountainRangeSystems
   (CountryID int foreign key references Country(CountryID),
   MountainRangeSystemID int foreign key references
MountainRangeSystem(MountainRangeSystemID),
   primary key (CountryID, MountainRangeSystemID))

Create table MountainGroup
   (MountainGroupID int primary key identity (1,1),
/*we set MountainGroupName unique, because at subpoint b) we want to have only one mountain
group with given name. From a real life point of view, there can not be two mountain groups with
the exact same name.*/
   MountainGroupName varchar(128) unique,
   MountainRangeSystemID int foreign key references
MountainRangeSystem(MountainRangeSystemID))

Create table Peak
   (PeakID int primary key identity (1,1),
   PeakName varchar(128),
   PeakHeight int,
   PeakLatitude real,
   PeakLongitude real,
   --we set on delete cascade, so our stored procedure from b) automatically deletes the peaks
   --when the mountain group is deleted
   MountainGroupID int foreign key references MountainGroup(MountainGroupID) on delete
cascade)

drop table Peak
```

```
--b)Implement a stored procedure that receives the name of a mountain group as parameter,
-- and deletes the mountain group with that name and all its peaks.

create procedure sp_Delete_MountainGroup_withPeaks @MountainGroupName varchar(128)
as
   declare @mountainGroupID int;
   select @mountainGroupID = MountainGroupID from MountainGroup where
MountainGroupName = @MountainGroupName
   --if the mountain group exists, then delete it
   if(@mountainGroupID <> 0) begin
   delete from MountainGroup
   where MountainGroupID = @mountainGroupID
   end
   --otherwise raise an error that the mountaing group doesn't exist
   else begin
      raiserror('There is no mountain group with given name!', 10, 1)
   end

--calling the procedure
exec sp_Delete_MountainGroup_withPeaks 'Munte1'

/*c)Create a view that shows the name of every mountain range system M that satisfies the
following conditions:
- M has at least 10 mountain groups AND
- M has at least 5 peaks over 2000 meters high.
 */

create or alter view mountainRangeSystem_View
as
   select MRS.MountainRangeSystemName
   from MountainRangeSystem MRS
   where
   (select count(*)
   from MountainGroup MG
   where MG.MountainRangeSystemID = MRS.MountainRangeSystemID) >= 10 AND
   (select count(*) as NumberOfPeaks
   from MountainRangeSystem MRS2
   inner join MountainGroup MG2 on MRS2.MountainRangeSystemID =
MG2.MountainRangeSystemID
   inner join Peak P on MG2.MountainGroupID = P.MountainGroupID
   where P.peakID > 2000) >= 5
go

--executing the view
select * from mountainRangeSystem_View

/*
 d. Implement a function that lists the names of the mountain groups with at least P peaks over M
meters high,
 where P and M are the function's parameters.
 */
create function uf_ListMountainGroups(@P int, @M int)
returns table
as
return
   select MG.MountainGroupName
   from MountainGroup MG
```

```
    where
    (select count(*)
    from Peak P
    where P.MountainGroupID = MG.MountainGroupID and P.PeakHeight >= @M) >= @P
go
```


MountainRangeSystem
MountainRangeSystemID int
MountainRangeSystemName varchar(128)
MountainRangeSystemLength int
Country
CountryID int
CountryName varchar(128)
CountryArea int
CountryPopulation int
MountainRangeSystemID:MountainRangeSystemID
MountainRangeSystemID:MountainRangeSystemID
CountryID:CountryID
MountainGroup
MountainGroupID int
MountainGroupName varchar(128)
MountainRangeSystemID int
Countries_MountainRangeSystems
CountryID int
MountainRangeSystemID int
MountainGroupID:MountainGroupID
Peak
PeakID int
PeakName varchar(128)
PeakHeight int
PeakLatitude real
PeakLongitude real
MountainGroupID int
mountainRangeSystem_View
MountainRangeSystemName varchar(128)