Databases
Seminary 6

Practical Exam

You must create a database for train scheduling management. The database will store data about the routes of all the trains.

a) The entities of interest to the problem domain are: *Trains*, *Train Types*, *Stations* and *Routes*.
b) Each train has a name and belongs to a type. The train type has only a description.
c) Each station has a name.
d) Each route has a name, an associated train and a list of stations with arrival and departure times in each station. The arrival and departure times are represented as hour:minute pairs (e.g., train arrives at 5pm and leaves at 5:10pm).

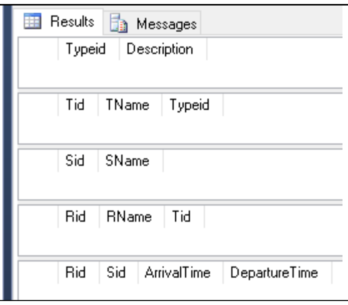
Your tasks are:
1) Write an SQL script that creates the corresponding relational data model. (4p)
2) Create a stored procedure that receives a route, a station, arrival and departure times and adds the station to the route. (1p)
3) Create a view that shows the names of the routes that contain all the stations. (2p)
4) Create a function that lists the names of the stations with more than ***R*** routes, where ***R>=1*** is a function parameter. (2p)
(1p of)

Solution:

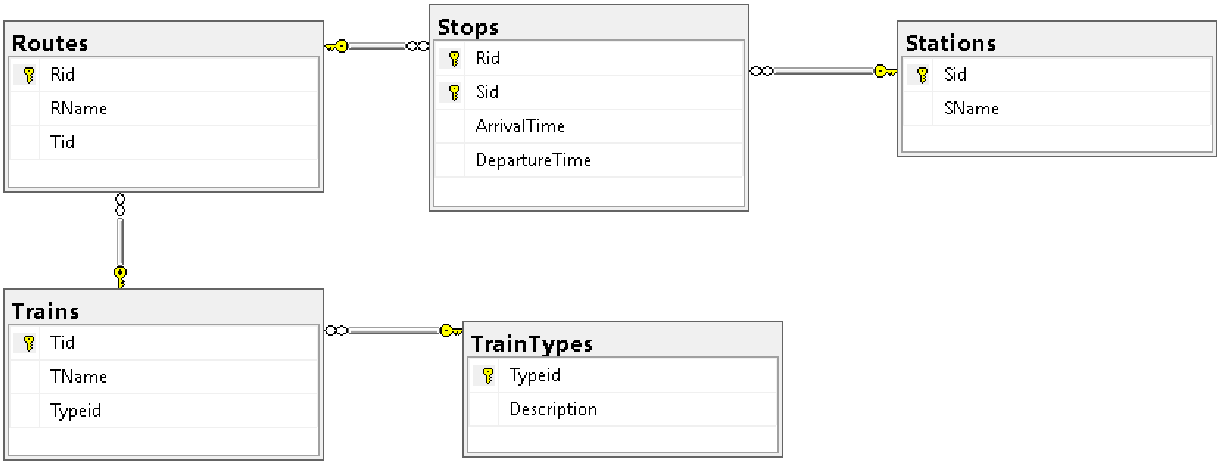
```
-- 1) Write an SQL script that creates the corresponding relational data model.      (4p)
create table Stations(
	Sid int primary key identity(1,1),
	SName VARCHAR(50)
)

create table TrainTypes(
	Typeid int primary key identity(1,1),
	Description VARCHAR(50)
)

create table Trains
(
	Tid int primary key identity(1,1),
	TName varchar(50),
	Typeid int foreign key references TrainTypes(Typeid)
)

create table Routes(
	Rid int primary key identity(1,1),
	RName varchar(50),
	Tid int foreign key references Trains(Tid)
)

create table Stops( -- RoutesStations
	Rid int foreign key references Routes(Rid),
	Sid int foreign key references Stations(Sid),
	ArrivalTime time,
	DepartureTime time
```

```
        CONSTRAINT pk_Stops PRIMARY KEY(Rid, Sid)
)

GO
```

```
select * from TrainTypes
select * from Trains
select * from Stations
select * from Routes
select * from Stops
```

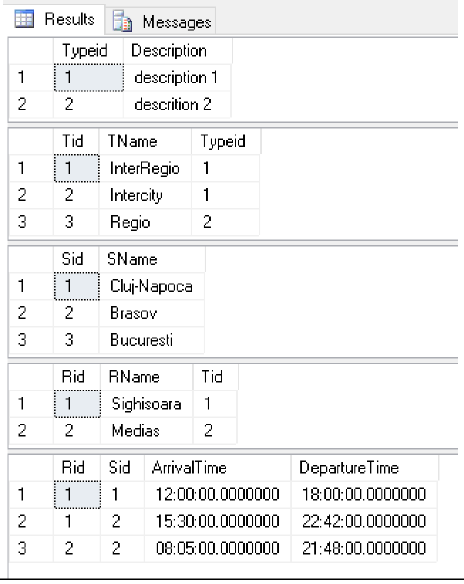


```
Results   Messages
    Typeid   Description

    Tid   TName   Typeid

    Sid   SName

    Rid   RName   Tid

    Rid   Sid   ArrivalTime   DepartureTime
```



```
INSERT INTO TrainTypes VALUES('description 1'), ('descrition 2')
INSERT INTO Trains values ('InterRegio', 1), ('Intercity', 1), ('Regio', 2)
INSERT INTO Stations values ('Cluj-Napoca'), ('Brasov'), ('Bucuresti')
Insert into Routes values ('Sighisoara', 1), ('Medias', 2)
INSERT Stops VALUES(1,1,'12:00:00', '18:00:00'), (1,2,'15:30:00', '22:42:00'),
(2,2,'08:05:00', '21:48:00')
GO
```

```
select * from TrainTypes
select * from Trains
select * from Stations
select * from Routes
select * from Stops
```



	Typeid	Description
1	1	description 1
2	2	descrition 2

	Tid	TName	Typeid
1	1	InterRegio	1
2	2	Intercity	1
3	3	Regio	2

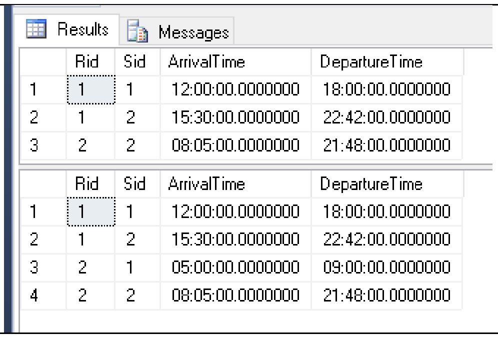
	Sid	SName
1	1	Cluj-Napoca
2	2	Brasov
3	3	Bucuresti

	Rid	RName	Tid
1	1	Sighisoara	1
2	2	Medias	2

	Rid	Sid	ArrivalTime	DepartureTime
1	1	1	12:00:00.0000000	18:00:00.0000000
2	1	2	15:30:00.0000000	22:42:00.0000000
3	2	2	08:05:00.0000000	21:48:00.0000000

```
-- 2) Create a stored procedure that receives a route, a station, arrival and departure
times and adds the station to the route. (1p)
go
create proc Add_Stops @Rid int, @Sid int, @at time, @dt time
AS
        DECLARE @nr int;
        SET @nr = 0;
        SELECT @nr = COUNT(*) FROM Stops WHERE Rid=@Rid and Sid=@Sid

        IF(@nr<>0) BEGIN
                PRINT 'This Stop already exists'
        END
        ELSE BEGIN
                INSERT INTO Stops VALUES (@Rid, @Sid, @at, @dt)
        END
Go
```

```
-- insert
select * from Stops
EXEC Add_Stops 2,1, '5:00:00', '9:00:00'
select * from Stops
```



	Rid	Sid	ArrivalTime	DepartureTime
1	1	1	12:00:00.0000000	18:00:00.0000000
2	1	2	15:30:00.0000000	22:42:00.0000000
3	2	2	08:05:00.0000000	21:48:00.0000000

	Rid	Sid	ArrivalTime	DepartureTime
1	1	1	12:00:00.0000000	18:00:00.0000000
2	1	2	15:30:00.0000000	22:42:00.0000000
3	2	1	05:00:00.0000000	09:00:00.0000000
4	2	2	08:05:00.0000000	21:48:00.0000000

```
-- already inserted
select * from Stops
EXEC Add_Stops 2,1, '15:00:00', '19:00:00'
select * from Stops
```

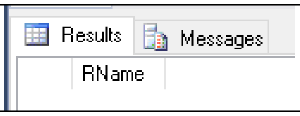
```
(4 row(s) affected)
This Stop already exists

(4 row(s) affected)
```

```
-- 3) Create a view that shows the names of the routes that contain all the stations.
(2p)
```

```
CREATE VIEW vRoutesStations
       AS
       SELECT RName
       FROM Routes r INNER JOIN Stops ss ON r.Rid = ss.Rid
       GROUP BY r.Rid, RName
       HAVING COUNT(*) = (SELECT COUNT(*) FROM Stations)
```

```
Command(s) completed successfully.
```

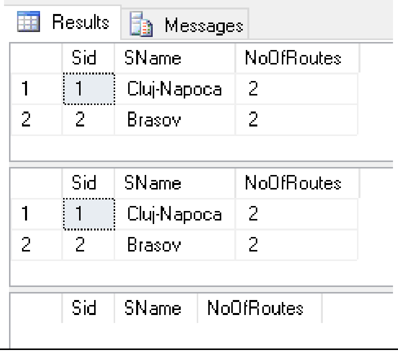
```
SELECT * FROM vRoutesStations
```



	RName

```
-- 4) Create a function that lists the names of the stations with more than R routes,
where R>=1 is a function parameter. (2p)

CREATE FUNCTION uf_StationsRoutes(@r int)
RETURNS TABLE
AS
RETURN
       SELECT DISTINCT s.Sid, SName, count(SName) as NoOfRoutes
       FROM Stations s INNER JOIN Stops ss ON  ss.Sid=s.Sid
       -- INNER JOIN Routes r ON  r.Rid=ss.Rid
       group by s.Sid, SName
       having count(SName)>=@r
go
```

```
SELECT * FROM uf_StationsRoutes(1)
SELECT * FROM uf_StationsRoutes(2)
SELECT * FROM uf_StationsRoutes(3)
```



	Sid	SName	NoOfRoutes
1	1	Cluj-Napoca	2
2	2	Brasov	2

	Sid	SName	NoOfRoutes
1	1	Cluj-Napoca	2
2	2	Brasov	2

	Sid	SName	NoOfRoutes

```
-- or
SELECT DISTINCT SName, count(SName) as NoOfRoutes
FROM Stations s INNER JOIN Stops ss ON  ss.Sid=s.Sid
group by SName
having count(SName)>=2
```