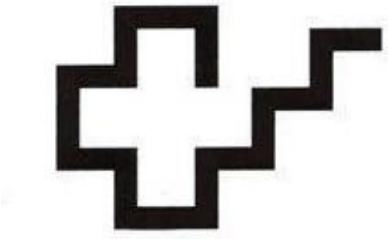
CSE 3055 DATABASE SYSTEMS TERM PROJECT

REHABILITATION CENTER



MUHAMMET KÜRŞAT AÇIKGÖZ	150116020
AHMET ELBURUZ GÜRBÜZ	150116024
AHMET ÖNKOL	150117018

PROJECT DESCRIPTION

In this project, our aim is to provide an admin panel for managing a rehabilitation center. By using the admin panel we created, all data related to tables can be accessed and managed by authorities. So that a user who has access to the panel can easily view,add,delete or update the data that is needed.

There are Employees, Disabled Students, Parents, Reports and related tables to these tables for sustaining data. Each student has a Ram and Health report. Also they have therapy sessions, Parents and Instructors. On the other hand, there are three types of employees in the rehab center such as Staff, Instructor and Psychologist. All of these information will be kept in the Rehab Center and can be accessed whenever it is required.

Overall, our main goal is to provide a panel so that whoever has access can quickly apply some changes and view them in case of need.

SCOPE OF THE PROJECT

Scope of our project is providing processable and organized data for our client. For that purpose we add additional tables for listing purposes.

What is included ?

For instance Student_list, Report_list, Instructor_list,Employee_list and Therapy_sessions are new tables for listing the relationships. With that way we separated the unrelated information kept in the tables and divide them into new tables.

Additionally we created a new table for address information for Employee and Parent, so that we don't have to rewrite the same information for different tables multiple times. Also we can view all address information from the Address table.

Furthermore, we assigned ID attribute to Parent, Employee, Therapy_sessions and Rehab_center for accessing these tables more quickly and establishing relationships between them.

What is excluded ?

On the other hand, we exclude attributes that are not directly related with the tables and separate them into new tables. So we ensure data integrity better.

We also exclude some unnecessary information from tables. For instance, we exclude identityNumber from Parent since it is not required by the Rehab Center.

Lastly we removed unrelated foreign keys from tables and stored them in lists(student_list,report_list,instructor_list,therapy_sessions,employee_list).

What is covered ?

We covered the recruitment and dismissal process of each employee with establishing related relationships. We also covered the student's registration process along with needed information such as reports which are mandatory and parent information. After student registration is done, students can adjust their instructors and therapy sessions accordingly.

What is not covered ?

We are not covered therapy descriptions since it's content may change with time depending on the student's condition. So we declare it as routine control.

Also the schedule of the lectures of each student or instructor is not covered since it is hard to handle without knowing actual information about them.

Addition to that content of the lectures are also not covered since it may vary depending on students' needs by time.

Finally, we have not covered the relationship between staff and student since it is not related to the rehab center and unnecessary.

REQUIREMENT ANALYSIS & CONCEPTUAL DATABASE DESIGN

Entities and Their Definitions :

Employee: Employee is the general table for a person who works for the company.

Staff: Staff is the subtable of the Employee which contains a set of employees who have a job that isn't directly related with the disabled students such as Chef, Servant, School bus driver etc.

Psychologist: Psychologist is the subtable of the Employee who has a direct connection with Disabled students.

Instructor: Instructor is another subtable of the Employee who also has a direct connection with Disabled students.

Rehabilitation Center: Rehab center is the company that is in a relation with both Employees and Disabled students. In case of need it can access both of the tables.

Disabled Student: Disabled Student is a person who has a physical or mental disability, gets education and psychological help from the rehabilitation center.

Parent: Parent is a person who is responsible for the Disabled Student and keeps in touch with employees that are in a relationship with disabled student.

RAM Report: Ram report holds the information of diagnosis of the disease for disabled students. It is mandatory for every student and is required document beforehand for application submission and related processes.

Health Report: Health report keeps the information of the specific diagnosis of biological system diseases. It determines which condition the student is currently in and it's percentage. It is also mandatory for application submission.

Business Processes and Their Definitions :

We meet with our customer for the details of the database we are going to create. Then we determine what the customer requirements are, and decide what rules and restrictions should be applied. Finally, we determine the customer's expectation from our database and user interface.

In the light of the information we get from our customer, we initially create the enhanced ER diagram. Basically, information about disabled students and employees will be kept in the diagram. We demonstrate a database structure that helps for managing new student registration and recruitment processes by using the user interface that the company wants from us.

We will create a database in MSSQL using the tables and relationships on the ER diagram. When creating the database, we will add primary keys and foreign keys. Before creating the user interface we will test our relationships and other functionalities by using necessary queries in MSSQL server. So that we can try what queries to use and how to demonstrate those functionalities. If the correct results are obtained from the tests, interface construction will be started. If the tests fail, we will make the necessary arrangements in the database.

We will create an interface that our customer can easily understand and use. So, while making this interface, we will put an username and password on the login screen in order to separate general manager-secretary access and increase security. Passwords will be stored in the database in encrypted form rather than direct form for security reasons. When admin logs in to the system with the user name and password , access to the entire database will be available. We will design a restricted user interface for the secretary that can only have the functionality of registering new students and accessing information about registered students. Finally, after all the user interface design is completed, we will do the necessary tests. We are going to test whether correct information is taken from the tables, the user interface does not have bugs, and adding-updating-deleting functionalities work dynamically . If the results of the tests are correct, it can be delivered to the customer.On the other hand, if an error is detected, we will repeat the tests after making the necessary checks on the interface.

Business rules, Constraints, etc:

There are some points we need to be careful about while creating this database. Apart from the company's desire for user interface**,** there are some other details and restrictions. We can list them as follows:

1. Employees cannot have an agreement with another company when they are currently working with the rehabilitation center.
2. There cannot be another rehabilitation center where disabled students attend.
3. Disabled students without a ram report cannot enroll in the rehabilitation center.
4. Disabled students who do not have a health report cannot register in the rehabilitation center.
5. In the user interface, only the rehabilitation center manager can recruit new employees, view employees' information, and update employees' information.
6. The founder of the rehabilitation center can add a secretary to the employee list as staff. People hired as secretaries can enroll new students. The secretary can list the students registered to the institution and modify the information of disabled students.
7. The secretary cannot see or edit information about other employees. Secretary doesn't have authority to access the information about other workers from the user interface.

Other functional & non-functional business requirements:

1. Functional Requirements

- Program allows users to create, delete or modify people in the system.
- Program can list and show every information in the database.
- The program has a login system that can be entered with a password.
- Admin can add information for disabled students.
- Admin can check and update information of disabled students.
- Admin can check and modify information of employees.
- Admin can access information from parents of disabled students.
- The secretary has limited access to information related to disabled students in the database since he or she doesn't have authority for employees.
- General manager can access all information in the database.

2. Non-functional Requirements

Usability

- User interface must be easy and understandable to use for users.
- Users need to use an interface to interact with the database.

Reliability

- Users can only enter the interface with password and username.
- Users can access the UI at any time.

Performance

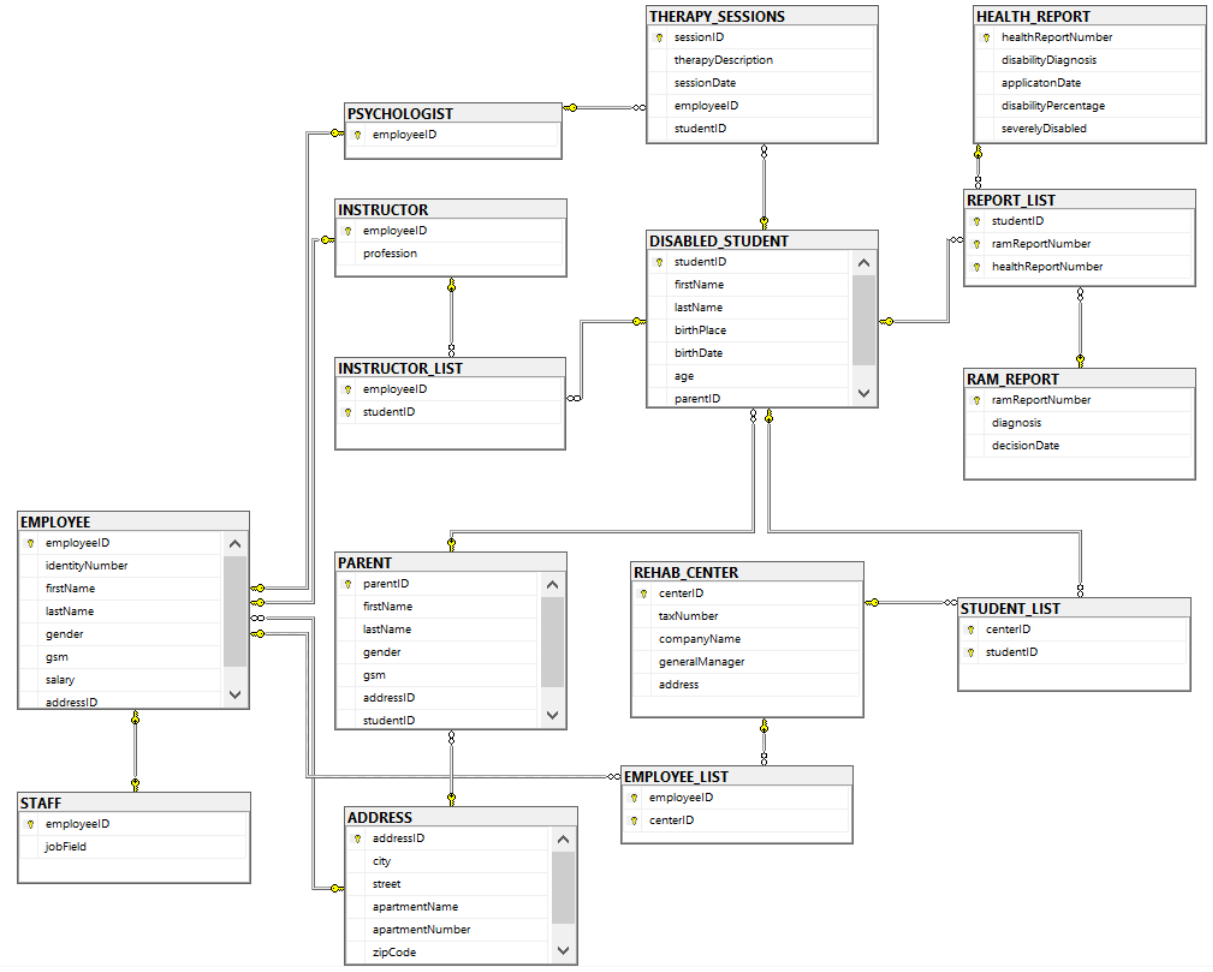
- New features should be easily added to the system.
- By minimizing the size of the data types, users can save memory space.

Supportability

- User interface must be platform independent.

Implementation

- Database will be implemented in MSSQL.
- Data will be provided to the database.



Information About Tables and Attributes

Table Name : ADDRESS

Table Definition : This table holds all address information

	Column Name	Data Type	Allow Nulls
▶🔑	addressID	int	☐
	city	nvarchar(25)	☑
	street	nvarchar(25)	☑
	apartmentName	nvarchar(25)	☑
	apartmentNumber	smallint	☑
	zipCode	int	☑
			☐

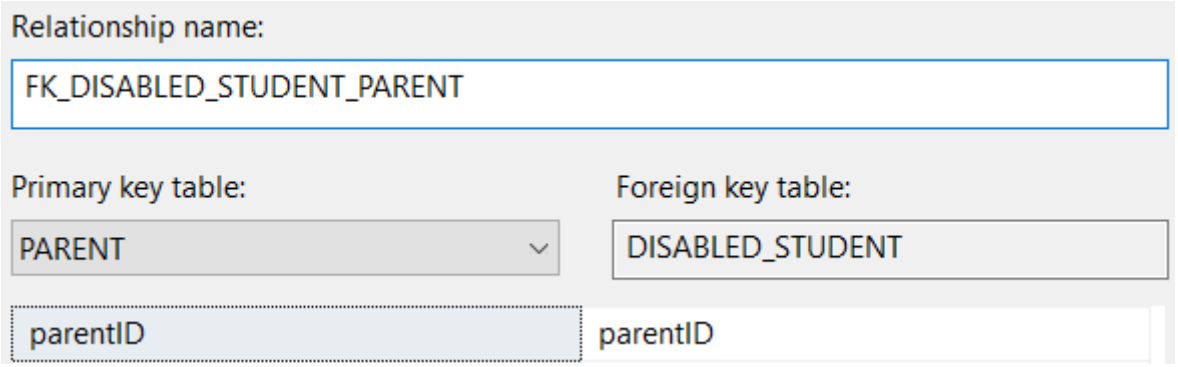
Identity : When each address is added, addressID is increased by 1 each, starting from 200.

Table Name : DISABLED_STUDENT

Table Definition : This table holds all students information

Column Name	Data Type	Allow Nulls
studentID	int	☐
firstName	nvarchar(50)	☐
lastName	nvarchar(50)	☐
birthPlace	nvarchar(50)	☑
birthDate	smalldatetime	☐
age	tinyint	☑
parentID	int	☐
		☐

Foreing Key for DISABLED_STUDENT Table



Computed Columns : Age is calculated based on birthDate

Identity : When each student is added, studentID is increased by 1 each, starting from 200.

Index: We have a primary key for each table, so that means we directly have one clustered index for each table. Except for primary key we added this index too:

CREATE INDEX [nameASC] ON [dbo].[DISABLED_STUDENT] ([firstName])

Triggers : We have 3 triggers.

- TrgAddStudentList adds a new student to the student list when a new student is registered.
- TrgUpdateAge calculates age attribute based on birthdate when a new student registered.

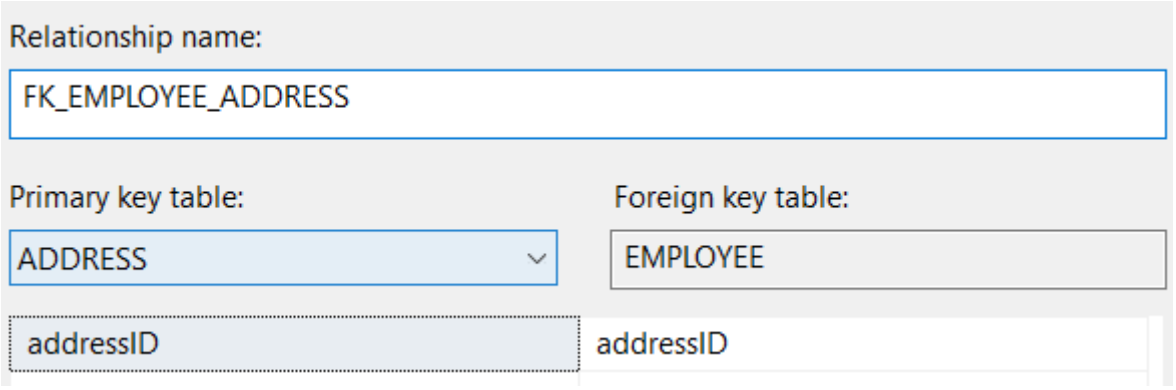
- TrgUpdateFullNameOfStudent enlarges the first character of students' first name if it is entered with a lower case when enrolling, and it also enlarges all characters of the surname if it isn't provided with that way.

Table Name : EMPLOYEE

Table Definition : This table holds employees information

Column Name	Data Type	Allow Nulls
employeeID	int	☐
identityNumber	char(11)	☐
firstName	nvarchar(50)	☐
lastName	nvarchar(50)	☐
gender	char(1)	☑
gsm	char(11)	☐
salary	smallint	☑
addressID	int	☑
		☐

Foreing Keys for EMPLOYEE Table



Uniques : identityNumber is unique for every employee

Identity : When each employee is added, employeeID is increased by 1 each, starting from 200.

Check Constraint : There are two check constraints

- Employees' gsm must be entered in 10 digits.
- Employees' identityNumbers must be entered in 11 digits.

Triggers : We have 3 triggers.

- TrgInsertEmployeeList adds a new employee to the employee list when a new employee is registered.
- TrgUpdateToMinSalary makes 2300 directly when a employees salary is entered less than 2300
- TrgUpdateFullNameOfEmployee enlarges the first character of employee' first name if it is entered with a lower case when enrolling, and it also enlarges all characters of the surname if it isn't provided with that way.

Table Name : EMPLOYEE_LIST

Table Definition : This table keeps the employees' together. Thus, we keep the list of employees in the rehabilitation center

Column Name	Data Type	Allow Nulls
employeeID	int	☐
centerID	int	☐

Foreing Keys for EMPLOYEE_LIST Table

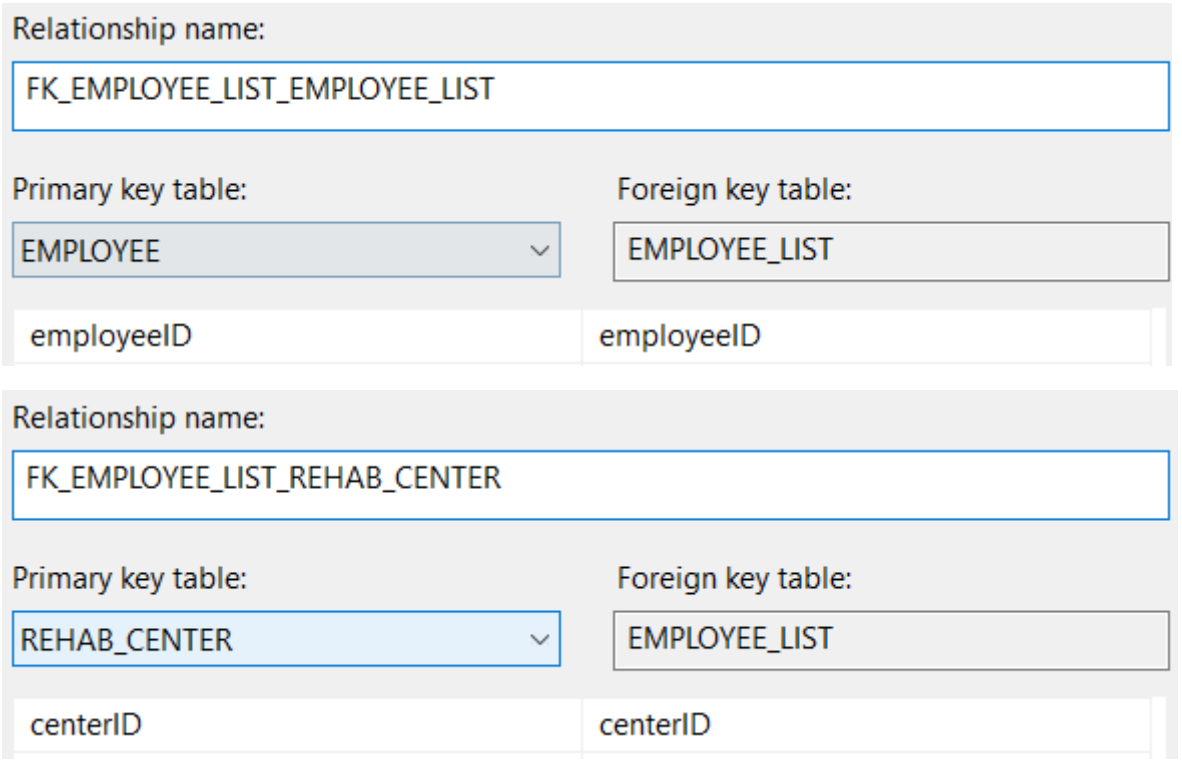


Table Name : HEALTH_REPORT

Table Definition : This table holds the information contained in the health report

Column Name	Data Type	Allow Nulls
healthReportNumber	nvarchar(9)	☐
disabilityDiagnosis	nvarchar(50)	☑
applicatonDate	smalldatetime	☑
disabilityPercentage	tinyint	☑
severelyDisabled	bit	☑

Default Values : The default value of the severelyDisabled is 0.

Index: We have a primary key for each table, so that means we directly have one clustered index for each table. Except for primary key we added this index too:

CREATE NONCLUSTERED INDEX [healthReportNumber] ON [dbo].[HEALTH_REPORT]([disabilityDiagnosis] ASC)

Computed Columns : severelyDisabled is calculated based on disabilityPercentage

Trigger : We have 1 trigger.

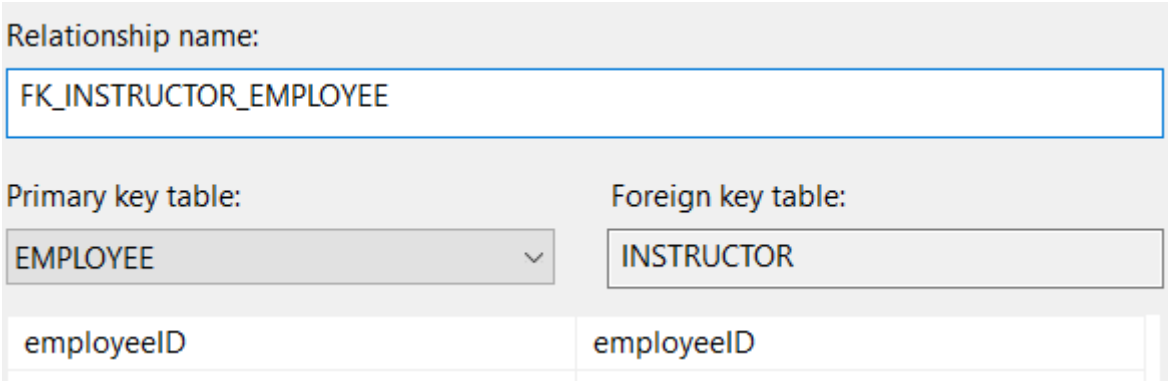
- DisabledUpdate makes 1 directly when disability percentage greater than or equal to 70

Table Name : INSTRUCTOR

Table Definition : This table holds instructor information

Column Name	Data Type	Allow Nulls
employeeID	int	☐
profession	nvarchar(50)	☑

Foreing Keys for INSTRUCTOR Table

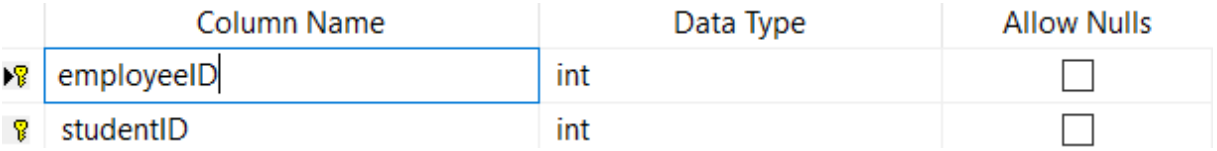


Identity : When an add a new instructor , employeeID increases one by one because employeeID is the foreign key in this table

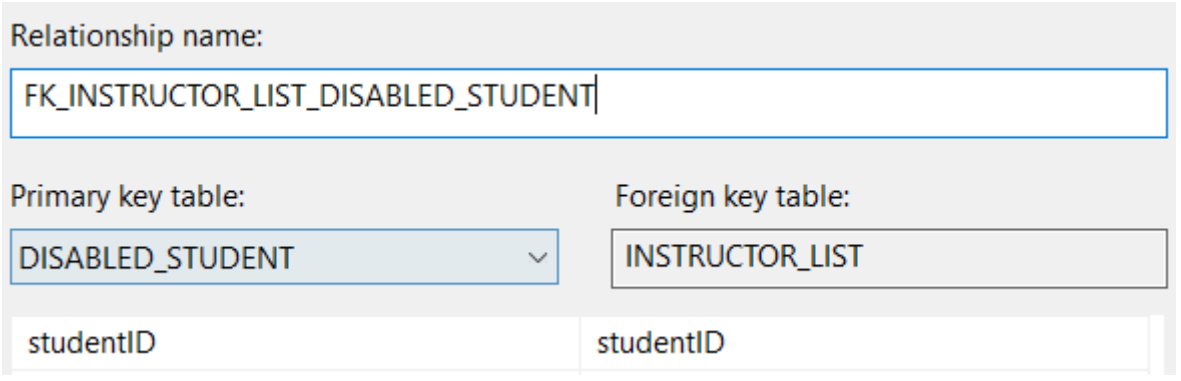
---

Table Name : INSTRUCTOR_LIST

Table Definition : This table keeps the instructors who have students together. Thus, we keep the list of instructor in the rehabilitation center



Foreing Keys for INSTRUCTOR_LIST Table



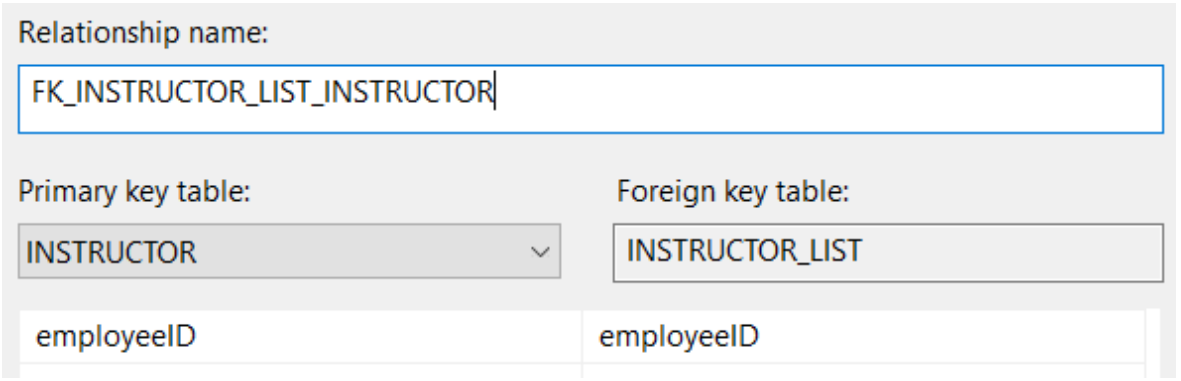
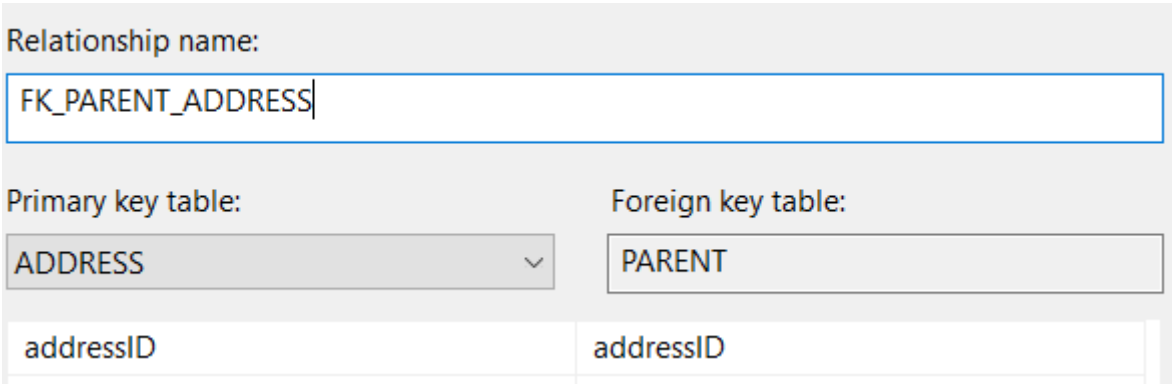


Table Name : PARENT

Table Definition : This table keeps information of parents of disabled students

Column Name	Data Type	Allow Nulls
parentID	int	☐
firstName	nvarchar(50)	☑
lastName	nvarchar(50)	☑
gender	char(1)	☑
gsm	char(11)	☑
addressID	int	☐
studentID	int	☑

Foreing Key for PARENT Table



Identity : When each student is added, studentID is increased by 1 each, starting from 200.

Check Constraint : There is a one check constraints

- Parents gsm must be entered in 10 digits.

Triggers : We have 1 trigger.

- TrgUpdateFullNameOfParent enlarges the first character of parents' first name if it is entered with a lower case when enrolling, and it also enlarges all characters of the surname if it isn't provided with that way.

Table Name : PSYCHOLOGIST

Table Definition : This table keeps information of psychologist

Column Name	Data Type	Allow Nulls
employeeID	int	☐

Foreing Key for PSYCHOLOGIST Table

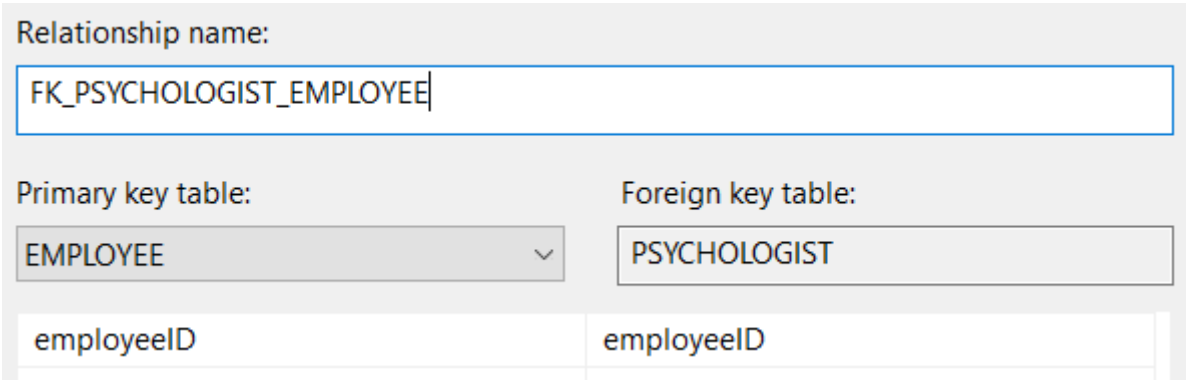


Table Name : RAM_REPORT

Table Definition : This table holds the information contained in the ram report

Column Name	Data Type	Allow Nulls
ramReportNumber	nvarchar(8)	☐
diagnosis	nvarchar(50)	☑
decisionDate	smalldatetime	☑

Table Name : REHAB_CENTER

Table Definition : This table keeps information about the rehabilitation center

Column Name	Data Type	Allow Nulls
centerID	int	☐
taxNumber	nvarchar(50)	☐
companyName	nvarchar(150)	☑
generalManager	nvarchar(100)	☑
address	nvarchar(50)	☑

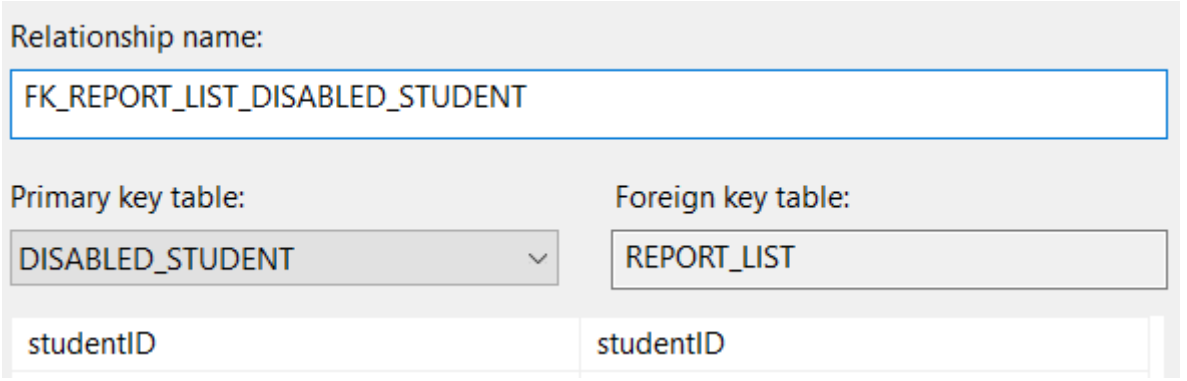
Unique : taxNumber is unique for the rehabilitation center.

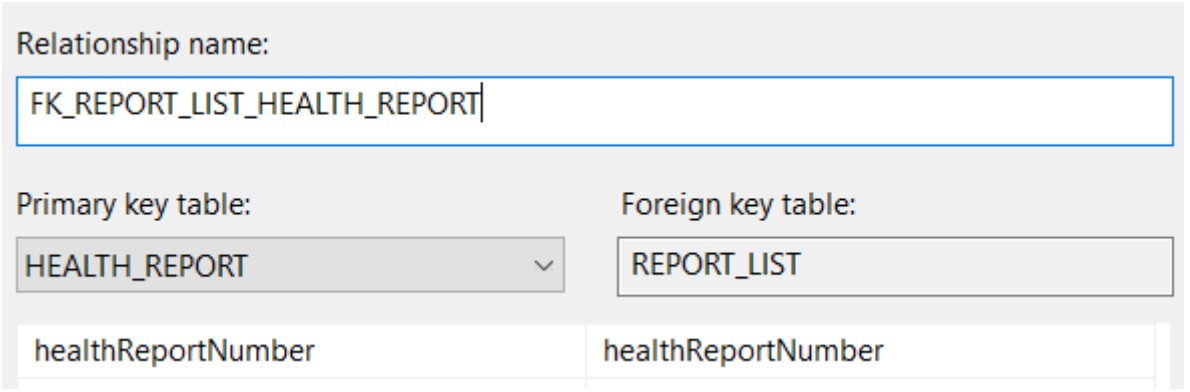
Table Name : REPORT_LIST

Table Definition : This table keeps all the reports of disabled students together. Thus, we can list all the reports of disabled students at the same time.

	Column Name	Data Type	Allow Nulls
🔑	studentID	int	☐
🔑	ramReportNumber	nvarchar(8)	☐
🔑	healthReportNumber	nvarchar(9)	☐

Foreign Keys for REPORT_LIST Table





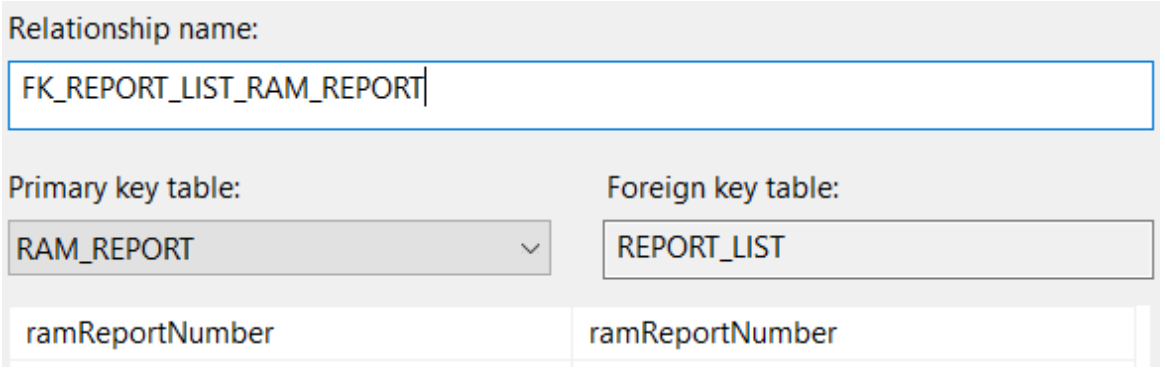


Table Name : STAFF

Table Definition : This table keeps information about staff.

Column Name	Data Type	Allow Nulls
employeeID	int	☐
jobField	nvarchar(50)	☑

Foreing Key for STAFF Table

Relationship name: FK_STAFF_EMPLOYEE

Primary key table:	Foreign key table:
EMPLOYEE	STAFF

employeeID	employeeID

Table Name : STUDENT_LIST

Table Definition : This table keeps information of all disabled students at the rehabilitation center.

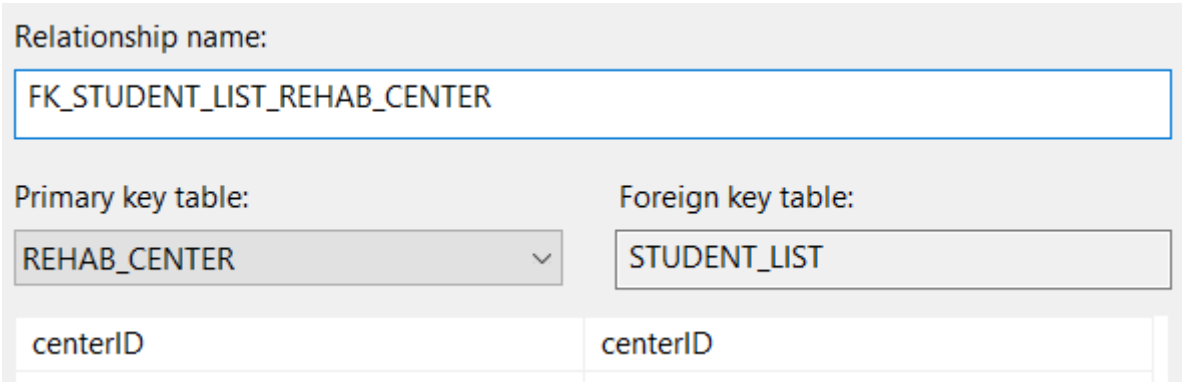
Column Name	Data Type	Allow Nulls
centerID	int	☐
studentID	int	☐

Foreigns Key for STUDENT_LIST Table

Relationship name: FK_STUDENT_LIST_DISABLED_STUDENT

Primary key table:	Foreign key table:
DISABLED_STUDENT	STUDENT_LIST

studentID	studentID



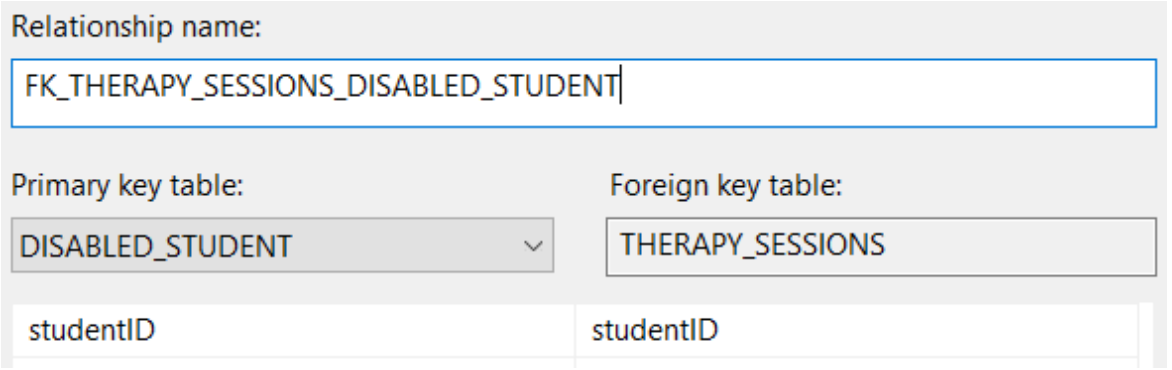
Default Values : The default value of the centerID is 101.

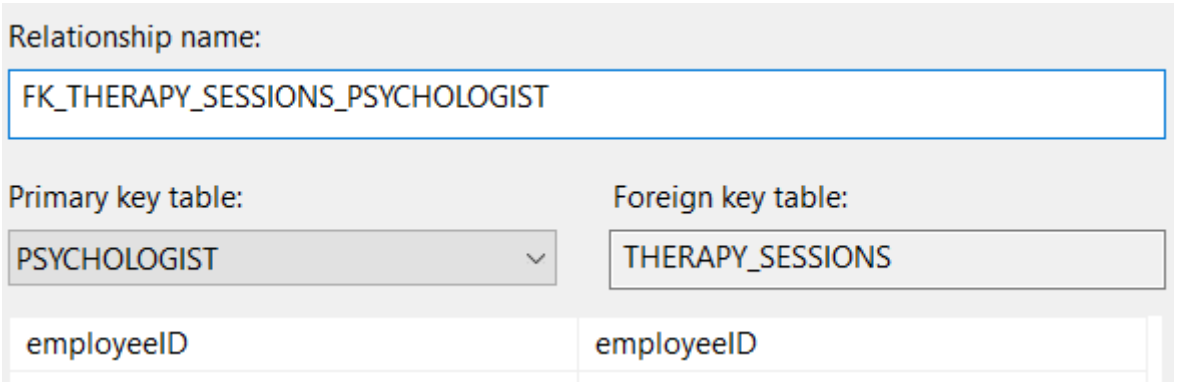
Table Name : THERAPY_SESSION

Table Definition : This table keeps information about the therapy session.

Column Name	Data Type	Allow Nulls
sessionID	int	☐
therapyDescription	nvarchar(250)	☑
sessionDate	smalldatetime	☑
employeeID	int	☐
studentID	int	☐

Foreigns Key for THERAPY_SESSION Table





Identity : When each therapy session is added, sessionID is increased by 1 each, starting from 200.

VIEWS

Name of the view: EMPLOYEE_BELOW_AVG_SALARY

Definition: This view will show employees which are under the average salary.

```
CREATE VIEW EMPLOYEE_BELOW_AVG_SALARY
AS
SELECT employeeID, firstName + ' ' + lastName AS fullName, salary
FROM dbo.EMPLOYEE AS e
WHERE ((SELECT AVG(DISTINCT salary) AS Expr1
      FROM dbo.EMPLOYEE) > salary)
```

	employeeID	fullName	salary
1	102	Melek ASLAN	2300
2	103	Sıddık SALİH	2450
3	104	Semiha ŞENGÜL	3000
4	120	Gülbeyaz ÖZALP	2750
5	122	Candan KİBAR	3000

Name of the view: LISTING_STUDENT_ASSIGNED_INST

Definition: This view will show which instructors students assigned.

```
create view LISTING_STUDENT_ASSIGNED_INST
as
SELECT ds.studentID, ds.firstName + ' ' + ds.lastName AS fullName
,emp.employeeID as instrucID,  emp.firstName as instrucName
, emp.lastName as instrucLastName
    FROM DISABLED_STUDENT ds,
          INSTRUCTOR ins , INSTRUCTOR_LIST il, EMPLOYEE emp
    WHERE ds.studentID = il.studentID
          and ins.employeeID = il.employeeID
          and emp.employeeID = il.employeeID
```

	studentID	fullName	instrucID	instrucName	instrucLastName
1	101	Sarp PARLAK	107	Haluk	DEMİR
2	102	Sevil ÜNEY	107	Haluk	DEMİR
3	103	Kader ALBOĞA	107	Haluk	DEMİR
4	104	Selin EMİN	107	Haluk	DEMİR
5	105	Cansen TANTEKİN	108	Mert	AY
6	106	Ecenur DAMAR	108	Mert	AY
7	107	Umut SAVAŞ	109	Hazel	KAYHAN
8	108	Seval FUAT	110	Cemalettin	SARIKAYA
9	109	Özer GÖRMÜŞ	110	Cemalettin	SARIKAYA
10	110	Erdem BÜYÜKTAŞ	111	Safiye	BEKRET
11	111	Zekiye KİREMİT	112	Yağmur	KAMİL
12	112	Ceyda ARKAN	112	Yağmur	KAMİL
13	113	Reha GÖKYAR	113	Dilek	EMEKLİ
14	114	Kağan İNAN	113	Dilek	EMEKLİ
15	115	Oğuz ORUÇ	113	Dilek	EMEKLİ
16	116	Gürhan ERYAVUZ	114	Mehmet	ULU
17	117	Öznur YERLİKAYA	115	Hüseyin	ÖZKAN
18	118	Sevim UZUNCAN	116	Hamit	CEVİZ
19	119	Cumhur ÇÖLBAY	116	Hamit	CEVİZ
20	120	Sinan ALKAN	117	Hacı	ÇİFTÇİ
21	121	Mervenur SERTD...	121	Ahmet	ERCAN
22	122	Umut KARADAŞ	121	Ahmet	ERCAN
23	123	Havva MAZLUM	118	Kadir	ERGAN
24	124	Handan ALAY	119	Zehra	ÖZKÇK
25	125	Şafak BARÇAK	121	Ahmet	ERCAN
26	126	Yusuf ERŞEKERCİ	120	Gülbeyaz	ÖZALP

Name of the view: LISTING_STUDENTS_ASSIGNED_TO_INSTRUCTOR

Definition: This view will show how many students each instructor has.

```
create view LISTING_STUDENTS_ASSIGNED_TO_INSTRUCTOR
as
SELECT i.employeeID, i.profession, emp.firstName,
emp.lastName, COUNT(*) AS numOfStudents
FROM dbo.DISABLED_STUDENT AS ds INNER JOIN
   dbo.INSTRUCTOR_LIST AS il ON ds.studentID = il.studentID INNER JOIN
   dbo.INSTRUCTOR AS i ON il.employeeID = i.employeeID INNER JOIN
   dbo.EMPLOYEE AS emp ON i.employeeID = emp.employeeID
GROUP BY i.employeeID, i.profession, emp.firstName, emp.lastName
```

	employeeID	profession	firstName	lastName	numOfStudents
1	107	Autism	Haluk	DEMİR	4
2	108	Autism	Mert	AY	2
3	109	Hearing impaired	Hazel	KAYHAN	1
4	110	Hearing impaired	Cemalettin	SARIKAYA	2
5	111	Developmental language disorder	Safiye	BEKRET	1
6	112	Developmental language disorder	Yağmur	KAMİL	2
7	113	Mental retardation	Dilek	EMEKLİ	3
8	114	Mental retardation	Mehmet	ULU	1
9	115	Cerebral palsy	Hüseyin	ÖZKAN	1
10	116	Cerebral palsy	Hamit	CEVİZ	2
11	117	Visually impaired	Hacı	ÇİFTÇİ	1
12	118	Visually impaired	Kadir	ERGAN	1
13	119	Physically disabled	Zehra	ÖZKÇK	1
14	120	Physically disabled	Gülbeyaz	ÖZALP	1
15	121	Kindergarten	Ahmet	ERCAN	3

Name of the view: THERAPY_SESSION_IN_ONE_WEEK

Definition: This view will show therapy sessions in the upcoming week.

```
create view THERAPY_SESSION_IN_ONE_WEEK
as
SELECT valids.studentID, valids.dsFullName AS studentFullName,
valids.fullNameOfPsychlogist, ts.therapyDescription, ts.sessionDate
FROM dbo.THERAPY_SESSIONS AS ts INNER JOIN
(SELECT ts.sessionID, ts.sessionDate, ds.studentID,
ds.firstName + ' ' + ds.lastName AS dsFullName,
e.firstName + ' ' + e.lastName AS fullNameOfPsychlogist, ts.therapyDescription
FROM dbo.DISABLED_STUDENT AS ds INNER JOIN
  dbo.THERAPY_SESSIONS AS ts ON ds.studentID = ts.studentID INNER JOIN
  dbo.EMPLOYEE AS e ON ts.employeeID = e.employeeID)
  AS valids ON ts.sessionID = valids.sessionID
WHERE (DATEDIFF(day, GETDATE(), ts.sessionDate) <= 7)
AND (GETDATE() < ts.sessionDate)
```

	studentID	studentFullName	fullNameOfPsychlogist	therapyDescription	sessionDate
1	104	Selin EMİN	Rümeysa SEVER	routine control	2020-12-29 15:55:00
2	105	Cansen TANTEKİN	Berkay ASLANTAŞ	routine control	2020-12-31 15:55:00

Name of the view: TOP_10_MOST_DISABLE

Definition: This view will show ten students' with the most disability percentage.

```
create view TOP_10_MOST_DISABLE
as
SELECT  TOP (10) ds.studentID, ds.firstName + ' ' + ds.lastName AS fullName
, matchedSt.per, p.firstName + ' ' + p.lastName AS parentName, p.gsm
FROM dbo.DISABLED_STUDENT AS ds INNER JOIN
     (SELECT ds.studentID AS stId, hr.disabilityPercentage AS per
        FROM dbo.DISABLED_STUDENT AS ds INNER JOIN
     dbo.REPORT_LIST AS rl ON ds.studentID = rl.studentID
     INNER JOIN dbo.HEALTH_REPORT AS hr
     ON rl.healthReportNumber = hr.healthReportNumber
     GROUP BY ds.studentID, hr.disabilityPercentage)
     AS matchedSt ON matchedSt.stId = ds.studentID INNER JOIN
     dbo.PARENT AS p ON ds.parentID = p.parentID
ORDER BY matchedSt.per DESC
```

	studentID	fullName	per	parentName	gsm
1	123	Havva MAZLUM	84	Mustafa MAZLUM	5356884328
2	108	Seval FUAT	82	Abdurrahman FUAT	5074445231
3	109	Özer GÖRMÜŞ	81	Bengühan GÖRMÜŞ	5075899658
4	116	Gürhan ERYAVUZ	77	Alper ERYAVUZ	5062886479
5	120	Sinan ALKAN	77	Onur ALKAN	5306895421
6	124	Handan ALAY	76	Hasret ALAY	5376445269
7	115	Oğuz ORUÇ	74	Uğurkan ORUÇ	5068661147
8	102	Sevil ÜNEY	73	Sevinç ÜNEY	5552990541
9	126	Yusuf ERŞEKERCİ	72	Seçil ERŞEKERCİ	5441889234
10	119	Cumhur ÇÖLBAY	71	Hasan ÇÖLBAY	5308254575

Name of the view: VALID_REPORT_FOR_STUDENT

Definition: This view will show the student's full name,ID and related report numbers along with diagnosis he or she has. With that usage, the user can confirm correctness between data by checking it.

```
create view VALID_REPORT_FOR_STUDENT
as
SELECT ds.studentID, ds.firstName + ' ' + ds.lastName as fullName,
valid.ramReportNumber,valid.healthReportNumber, valid.diagnosis
   FROM dbo.DISABLED_STUDENT AS ds
   INNER JOIN (SELECT ds.studentID, rl.healthReportNumber,
               rl.ramReportNumber, rr.diagnosis
               FROM dbo.DISABLED_STUDENT AS ds
               INNER JOIN dbo.REPORT_LIST AS rl
               ON ds.studentID = rl.studentID
               INNER JOIN dbo.HEALTH_REPORT AS hr
               ON rl.healthReportNumber = hr.healthReportNumber
               INNER JOIN dbo.RAM_REPORT AS rr
               ON rl.ramReportNumber = rr.ramReportNumber
               AND hr.disabilityDiagnosis = rr.diagnosis) AS valid
               ON ds.studentID = valid.studentID
```

	studentID	fullName	ramReportNumber	healthReportNumber	diagnosis
1	101	Sarp PARLAK	1323569	162415841	Autism
2	102	Sevil ÜNEY	1425793	163230397	Autism
3	103	Kader ALBOĞA	1574697	164396511	Autism
4	104	Selin EMİN	1657236	169495763	Autism
5	105	Cansen TANTEKİN	1727856	175382324	Autism
6	106	Ecenur DAMAR	1737817	177338829	Autism
7	107	Umut SAVAŞ	1724368	160144874	Hearing impaired
8	108	Seval FUAT	1744423	162395431	Hearing impaired
9	109	Özer GÖRMÜŞ	1746324	173684985	Hearing impaired
10	110	Erdem BÜYÜKTAŞ	1410236	161335385	Developmental language disorder
11	111	Zekiye KİREMİT	1502367	165743075	Developmental language disorder
12	112	Ceyda ARKAN	1782965	171078235	Developmental language disorder
13	113	Reha GÖKYAR	1453697	161675711	Mental retardation
14	114	Kağan İNAN	1525789	165981184	Mental retardation
15	115	Oğuz ORUÇ	1558964	169758382	Mental retardation
16	116	Gürhan ERYAVUZ	1737822	170979259	Mental retardation
17	117	Öznur YERLİKAYA	1711457	163413125	Cerebral palsy
18	118	Sevim UZUNCAN	1745678	168647576	Cerebral palsy
19	119	Cumhur ÇÖLBAY	1789567	179886953	Cerebral palsy
20	120	Sinan ALKAN	1302368	164828212	Visually impaired
21	121	Mervenur SERTD...	1703597	168837557	Visually impaired
22	122	Umut KARADAŞ	1717356	175904506	Visually impaired
23	123	Havva MAZLUM	1737456	176714168	Visually impaired
24	124	Handan ALAY	1224569	169963588	Physically disabled
25	125	Şafak BARÇAK	1628985	177388641	Physically disabled
26	126	Yusuf ERŞEKERCİ	1724698	178733654	Physically disabled

TRIGGERS

Name of the trigger: TrgAddStudentList (DISABLED_STUDENT table)

Definition: This trigger will add studentID and centerID into student_list whenever a new disabled student is registered since centerID is fixed.

Code:

```
-- ============================================
-- Author:      <Kürşat Açıkgöz>
-- Create date: <26.12.2020>
-- Description: <Trigger Add new student to Student List>
-- ============================================
ALTER TRIGGER [dbo].[TrgAddStudentList]
   ON  [dbo].[DISABLED_STUDENT]
   AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO  STUDENT_LIST(studentID,centerID)
    VALUES (IDENT_CURRENT('DISABLED_STUDENT'),101)
END
```

Disabled Student table before execution:

118	Sevim	UZUNCAN	Kayseri	2010-11-08 00:0...	10	118
119	Cumhur	ÇÖLBAY	Ardahan	2012-03-30 00:0...	8	119
120	Sinan	ALKAN	Çanakkale	2012-11-21 00:0...	8	120
121	Mervenur	SERTDEMİR	Adıyaman	2013-04-11 00:0...	7	121
122	Umut	KARADAŞ	Tokat	2013-05-01 00:0...	7	122
123	Havva	MAZLUM	Iğdır	2014-03-11 00:0...	6	123
124	Handan	ALAY	Diyarbakır	2014-06-23 00:0...	6	124
125	Şafak	BARÇAK	Hakkari	2014-08-24 00:0...	6	125
126	Yusuf	ERŞEKERCİ	Konya	2014-11-03 00:0...	6	126
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Student_list table before execution:

101	117
101	118
101	119
101	120
101	121
101	122
101	123
101	124
101	125
101	126
NULL	*NULL*

Student table with new student(with student id 229):

117	Öznur	YERLİKAYA	Sınop	2010-10-23 00:0...	10	117
118	Sevim	UZUNCAN	Kayseri	2010-11-08 00:0...	10	118
119	Cumhur	ÇÖLBAY	Ardahan	2012-03-30 00:0...	8	119
120	Sinan	ALKAN	Çanakkale	2012-11-21 00:0...	8	120
121	Mervenur	SERTDEMİR	Adıyaman	2013-04-11 00:0...	7	121
122	Umut	KARADAŞ	Tokat	2013-05-01 00:0...	7	122
123	Havva	MAZLUM	Iğdır	2014-03-11 00:0...	6	123
124	Handan	ALAY	Diyarbakır	2014-06-23 00:0...	6	124
125	Şafak	BARÇAK	Hakkari	2014-08-24 00:0...	6	125
126	Yusuf	ERŞEKERCİ	Konya	2014-11-03 00:0...	6	126
229	İsim	SOYİSİM	sehir	2020-12-12 00:0...	0	130
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Student_list table after execution:

101	117
101	118
101	119
101	120
101	121
101	122
101	123
101	124
101	125
101	126
101	229
NULL	*NULL*

Name of the trigger: TrgUpdateAge (DISABLED_STUDENT table)

Definition: This trigger will update the age column of disabled student using its birth date column when there is an insert or update.

Code

```
-- ============================================
-- Author:  <Kursat Acikgoz>
-- Create date: <26.12.2020>
-- Description: <Update Age>
-- ============================================
ALTER TRIGGER [dbo].[TrgUpdateAge]
   ON [dbo].[DISABLED_STUDENT]
   AFTER insert,update
AS
BEGIN
     SET NOCOUNT ON;
     IF TRIGGER_NESTLEVEL() > 1
          RETURN

     update ds
     set ds.age=datediff(year,ds.birthDate,getdate())
     from DISABLED_STUDENT ds
END
```

Disabled Student table before insertion:

118	Sevim	UZUNCAN	Kayseri	2010-11-08 00:0...	10	118
119	Cumhur	ÇÖLBAY	Ardahan	2012-03-30 00:0...	8	119
120	Sinan	ALKAN	Çanakkale	2012-11-21 00:0...	8	120
121	Mervenur	SERTDEMİR	Adıyaman	2013-04-11 00:0...	7	121
122	Umut	KARADAŞ	Tokat	2013-05-01 00:0...	7	122
123	Havva	MAZLUM	Iğdır	2014-03-11 00:0...	6	123
124	Handan	ALAY	Diyarbakır	2014-06-23 00:0...	6	124
125	Şafak	BARÇAK	Hakkari	2014-08-24 00:0...	6	125
126	Yusuf	ERŞEKERCİ	Konya	2014-11-03 00:0...	6	126
NULL	isim	soyisim	sehir	2010-10-10 00:0	*NULL*	150
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Disabled Student table after insertion:

118	Sevim	UZUNCAN	Kayseri	2010-11-08 00:0...	10	118
119	Cumhur	ÇÖLBAY	Ardahan	2012-03-30 00:0...	8	119
120	Sinan	ALKAN	Çanakkale	2012-11-21 00:0...	8	120
121	Mervenur	SERTDEMİR	Adıyaman	2013-04-11 00:0...	7	121
122	Umut	KARADAŞ	Tokat	2013-05-01 00:0...	7	122
123	Havva	MAZLUM	Iğdır	2014-03-11 00:0...	6	123
124	Handan	ALAY	Diyarbakır	2014-06-23 00:0...	6	124
125	Şafak	BARÇAK	Hakkari	2014-08-24 00:0...	6	125
126	Yusuf	ERŞEKERCİ	Konya	2014-11-03 00:0...	6	126
230	İsim	SOYİSİM	sehir	2010-10-10 00:0...	10	150
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Name of the trigger: TrgUpdateFullNameOfStudent (DISABLED_STUDENT)

Definition: This trigger will update name and surname of a student as first name's first letter with upper case and last name as full of upper case characters when there is an insert or update.

Code:

```
-- ============================================
-- Author:  <Kursat Acikgoz>
-- Create date: <26.12.2020>
-- Description: <Update FullName with capital letters>
-- ============================================
ALTER TRIGGER [dbo].[TrgUpdateFullNameOfStudent]
   ON  [dbo].[DISABLED_STUDENT]
   AFTER INSERT,update
AS
BEGIN
    SET NOCOUNT ON;
    update ds
    set ds.lastName=UPPER(ds.lastName),
    ds.firstName=UPPER(LEFT(ds.firstName,1))
    +LOWER(SUBSTRING(ds.firstName,2,LEN(ds.firstName)))
    from DISABLED_STUDENT ds

END
```

Disabled Student table before insertion:

122	Umut	KARADAŞ	Tokat	2013-05-01 00:0...	7	122
123	Havva	MAZLUM	Iğdır	2014-03-11 00:0...	6	123
124	Handan	ALAY	Diyarbakır	2014-06-23 00:0...	6	124
125	Şafak	BARÇAK	Hakkari	2014-08-24 00:0...	6	125
126	Yusuf	ERŞEKERCİ	Konya	2014-11-03 00:0...	6	126
NULL	ali	yılmaz	şehir	2000-10-10 00:0	*NULL*	160
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Disabled Student table after insertion:

121	Mervenur	SERTDEMİR	Adıyaman	2013-04-11 00:0...	7	121
122	Umut	KARADAŞ	Tokat	2013-05-01 00:0...	7	122
123	Havva	MAZLUM	Iğdır	2014-03-11 00:0...	6	123
124	Handan	ALAY	Diyarbakır	2014-06-23 00:0...	6	124
125	Şafak	BARÇAK	Hakkari	2014-08-24 00:0...	6	125
126	Yusuf	ERŞEKERCİ	Konya	2014-11-03 00:0...	6	126
232	Ali	YILMAZ	şehir	2000-10-10 00:0...	20	160
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Name of the trigger: TrgInsertEmployeeList (EMPLOYEE table)

Definition: This trigger will add employeeID and centerID into employee_list whenever a new employee is registered since centerID is fixed.

Code:

```
-- =============================================
-- Author:      <Kürşat Açıkgöz>
-- Create date: <26.12.2020>
-- Description: <Add Employees to Employee List Table>
-- =============================================
ALTER TRIGGER [dbo].[TrgInsertEmployeeList]
   ON  [dbo].[EMPLOYEE]
   AFTER insert
AS
BEGIN
IF TRIGGER_NESTLEVEL() > 1

RETURN
    SET NOCOUNT ON;
    ALTER TABLE EMPLOYEE_LIST NOCHECK CONSTRAINT ALL

     INSERT INTO  EMPLOYEE_LIST(employeeID,centerID)
     VALUES (IDENT_CURRENT('EMPLOYEE'),101)

END
```

Employee table before execution:

121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
NULL	12213232232	isim	soyisim	f	5555050505	3000	239
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Employee table after execution:

119	12345678891	Zehra	ÖZKÇK	F	5361234567	3675	119
120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
213	12213232232	İsim	SOYİSİM	f	5555050505	3000	239
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

EmployeeID will be automatically assigned since we used identity for it.

Employee list table after insertion:

116	101
117	101
118	101
119	101
120	101
121	101
122	101
201	101
213	101
NULL	*NULL*

Name of the trigger: TrgUpdateFullNameOfEmployee (EMPLOYEE table)

Definition: This trigger will update the name and surname of a employee as first name's first letter with upper case and last name as full of upper case characters when there is an insert or update.

Code:

```
-- =============================================
-- Author:  <Kursat Acikgoz>
-- Create date: <26.12.2020>
-- Description: <Update FullName with capital letters>
-- =============================================
ALTER TRIGGER [dbo].[TrgUpdateFullNameOfEmployee]
   ON  [dbo].[EMPLOYEE]
   AFTER INSERT,UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    update e
    set e.lastName=UPPER(e.lastName)
    , e.firstName=UPPER(LEFT(e.firstName,1))
    +LOWER(SUBSTRING(e.firstName,2,LEN(e.firstName)))
    from EMPLOYEE e

END
```

Employee table before execution:

121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
213	12213232232	meltem	çelik	f	5555050505	3000	239
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Employee table after execution:

121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
213	12213232232	Meltem	ÇELİK	f	5555050505	3000	239
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Name of the trigger: TrgUpdateToMinSalary (EMPLOYEE table)

Definition: This trigger will update the salary of an employee to 2300 if salary is smaller than that when there is an insert or update.

Code:

```
-- =============================================
-- Author:  <Ahmet Elburuz Gürbüz>
-- Create date: <26.12.2020>
-- Description: <Determine minimum salary>
-- =============================================
ALTER TRIGGER [dbo].[TrgUpdateToMinSalary]
   ON  [dbo].[EMPLOYEE]
   AFTER INSERT,UPDATE
AS
BEGIN
IF TRIGGER_NESTLEVEL() > 1

RETURN
    SET NOCOUNT ON;

    update e
    set e.salary = 2300
    from EMPLOYEE e
    Where e.salary<2300
END
```

Employee table before execution:

119	12345678891	Zehra	ÖZKÇK	F	5361234567	3675	119
120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
213	12213232232	Meltem	ÇELİK	f	5555050505	2300	239
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Employee table after execution:

119	12345678891	Zehra	ÖZKÇK	F	5361234567	3675	119
120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
213	12213232232	Meltem	ÇELİK	f	5555050505	2300	239
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Name of the trigger: DisabledUpdate (HEALTH_REPORT table)

Definition: This trigger will set severely disabled attribute to 1 if newly added student has disability percentage of 70 or above. It will happen when there is an insert or update of a student.

Code:

```
-- ================================================
-- Author:		Ahmet Önkol
-- Create date: 26.12.2020
-- Description: SeverelyDisabledUpdate
-- ================================================
ALTER TRIGGER [dbo].[DisabledUpdate]
   ON [dbo].[HEALTH_REPORT]
   AFTER INSERT,UPDATE
AS
BEGIN
    SET NOCOUNT ON;
            UPDATE hr
            SET hr.severelyDisabled=1
            FROM HEALTH_REPORT hr
            WHERE hr.disabilityPercentage >= 70


END
```

Health Report table before execution:

175904506	Visually impaired	2020-03-01 00:0...	67	False
176714168	Visually impaired	2020-09-02 00:0...	84	True
177338829	Autism	2020-04-03 00:0...	40	False
177388641	Physically disab...	2020-04-23 00:0...	32	False
178733654	Physically disab...	2020-01-03 00:0...	72	True
179886953	Cerebral palsy	2020-10-02 00:0...	71	True
211334344	hastalık	2010-10-10 00:0	82	NULL
NULL	NULL	NULL	NULL	NULL

Health Report table after execution:

175382324	Autism	2020-03-05 00:0...	57	False
175904506	Visually impaired	2020-03-01 00:0...	67	False
176714168	Visually impaired	2020-09-02 00:0...	84	True
177338829	Autism	2020-04-03 00:0...	40	False
177388641	Physically disab...	2020-04-23 00:0...	32	False
178733654	Physically disab...	2020-01-03 00:0...	72	True
179886953	Cerebral palsy	2020-10-02 00:0...	71	True
211334344	hastalık	2010-10-10 00:0...	82	True
NULL	*NULL*	*NULL*	*NULL*	*NULL*

Name of the trigger: TrgUpdateFullNameOfParent (PARENT table)

Definition: This trigger will update the name and surname of a parent as the first name's first letter with upper case and last name as full of upper case characters when there is an insert or update.

Code:

```
-- ============================================
-- Author:  <Kursat Acikgoz>
-- Create date: <26.12.2020>
-- Description: <Update FullName with capital letters>
-- ============================================
ALTER TRIGGER [dbo].[TrgUpdateFullNameOfParent]
   ON  [dbo].[PARENT]
   AFTER INSERT,UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    update p
    set p.lastName=UPPER(p.lastName),
    p.firstName=UPPER(LEFT(p.firstName,1))
    +LOWER(SUBSTRING(p.firstName,2,LEN(p.firstName)))
    from PARENT p

END
```

Parent table before execution:

121	Pınar	SERTDEMİR	F	5390281475	152	121
122	Erem	KARADAŞ	M	5521235874	153	122
123	Mustafa	MAZLUM	M	5356884328	154	123
124	Hasret	ALAY	F	5376445269	155	124
125	Şerife	BARÇAK	F	5492998919	156	125
126	Seçil	ERŞEKERCİ	F	5441889234	157	126
NULL	efe	can	M	5050505055	240	128
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Parent table after execution:

122	Erem	KARADAŞ	M	5521235874	153	122
123	Mustafa	MAZLUM	M	5356884328	154	123
124	Hasret	ALAY	F	5376445269	155	124
125	Şerife	BARÇAK	F	5492998919	156	125
126	Seçil	ERŞEKERCİ	F	5441889234	157	126
231	Efe	CAN	M	5050505055	240	128
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

STORED PROCEDURES

Name of stored procedure: sp_AddAddress

Definition: This stored procedure is written to add a new address information to the address table.

Below are the screenshots respectively:

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddAddress]    Script Date: 27.12.2020 17:10:56 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Kürşat Açıkgöz>
-- Create date: <26.12.2020>
-- Description: <Add Address>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddAddress]
    @city nvarchar(25),
    @street nvarchar(25),
    @apartmentName nvarchar(25),
    @apartmentNumber smallint,
    @zipCode int
AS
BEGIN
 -- SET NOCOUNT ON added to prevent extra result sets from
 -- interfering with SELECT statements.
  SET NOCOUNT ON;

    INSERT INTO ADDRESS(city, street,apartmentName,apartmentNumber,zipCode)
    VALUES (@city, @street,@apartmentName,@apartmentNumber,@zipCode);
END
```

```
exec sp_AddAddress 'Eskişehir','Şair', 'Komando', 27 , 26000
```

48	157	Eskişehir	Uçaklar Sk.	Alperen	33	26000
49	200	Eskişehir	Yalı Sk.	Saman	35	26000
50	201	Eskişehir	Altay Cd.	Barkın	37	26000
51	203	Eskişehir	Altay Cd.	Kaynar	28	26000

48	157	Eskişehir	Uçaklar Sk.	Alperen	33	26000
49	200	Eskişehir	Yalı Sk.	Saman	35	26000
50	201	Eskişehir	Altay Cd.	Barkın	37	26000
51	203	Eskişehir	Altay Cd.	Kaynar	28	26000
52	207	Eskişehir	Şair	Komando	27	26000

Name of stored procedure: sp_AddDisabledStudent

Definition: This stored procedure takes address, parent, disabled student, health report, ram report information as arguments. After adding to tables with this information, additions are made to the report list and student list. The main purpose of this stored procedure, which is very frequently associated with foreign keys, is to add students.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddDisabledStudent]    Script Date: 28.12.2020 01:08:31 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Kürşat Açıkgöz>
-- Create date: <26.12.2020>
-- Description: <Add Disabled Student>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddDisabledStudent]
    @firstName nvarchar(50),
    @lastName nvarchar(50),
    @birthPlace nvarchar(50),
    @birthDate smalldatetime,
    @parentFirstName nvarchar(50),
    @parentLastName nvarchar(50),
    @parentGender char(1),
    @parentGsm char(11),
    @city nvarchar(25),
    @street nvarchar(25),
    @apartmentName nvarchar(25),
    @apartmentNumber smallint,
    @zipCode int,
    @ramReportNumber nvarchar(8),
    @diagnosis nvarchar(50),
    @decisionDate smalldatetime,
    @healthReportNumber nvarchar(9),
    @disabilityDiagnosis nvarchar(50),
    @applicatonDate smalldatetime,
    @disabilityPercentage tinyint
AS
BEGIN
```

```
    SET NOCOUNT ON;
    ALTER TABLE DISABLED_STUDENT NOCHECK CONSTRAINT ALL

    declare @stdID int, @parentID int, @addressID int
    set @stdID = IDENT_CURRENT('DISABLED_STUDENT')
    set @parentID = IDENT_CURRENT('PARENT')+1
    set @addressID = IDENT_CURRENT('ADDRESS')

    set @decisionDate = CONVERT(smalldatetime, @decisionDate, 104)

    exec sp_AddReports @ramReportNumber,@diagnosis,@decisionDate,@healthReportNumber,@disabilityDiagnosis,@applicatonDate,@disabilityPercentage

    exec sp_AddParent @city, @street,@apartmentName,@apartmentNumber,@zipCode,@parentFirstName,@parentLastName,@parentGender,@parentGsm,@stdID

    exec sp_AddReportList @stdID, @ramReportNumber, @healthReportNumber

    INSERT INTO DISABLED_STUDENT(firstName, lastName,birthPlace,birthDate, parentID)
    VALUES (@firstName, @lastName,@birthPlace,CONVERT(smalldatetime, @birthDate, 104),@parentID);

END
```

Address Table

157	Eskişehir	Uçaklar Sk.	Alperen	33	26000
200	Eskişehir	Yalı Sk.	Saman	35	26000
201	Eskişehir	Altay Cd.	Barkın	37	26000
203	Eskişehir	Altay Cd.	Kaynar	28	26000

Disabled Student Table

124	Handan	ALAY	Diyarbakır	2014-06-23 ...	6	124
125	Şafak	BARÇAK	Hakkari	2014-08-24 ...	6	125
126	Yusuf	ERŞEKERCİ	Konya	2014-11-03 ...	6	126
227	İsim	SOYAD	Malatya	2018-10-10 ...	2	229

Parent Table

124	Hasret	ALAY	F	5376445269	155	124
125	Şerife	BARÇAK	F	5492998919	156	125
126	Seçil	ERŞEKERCİ	F	5441889234	157	126
229	Parentname	PARENTLAST	F	1201201232	236	227

Ram Report Table

1745678	Cerebral pal...	2020-09-18 ...
1746324	Hearing im...	2020-09-05 ...
1782965	Developme...	2020-03-29 ...
1789567	Cerebral pal...	2019-08-07 ...

Health Report Table

177338829	Autism	2020-04-03 ...	40	False
177388641	Physically di...	2020-04-23 ...	32	False
178733654	Physically di...	2020-01-03 ...	72	True
179886953	Cerebral pal...	2020-10-02 ...	71	True

Report List Table

124	1224569	169963588
125	1628985	177388641
126	1724698	178733654
227	1475841	5614814

```
sp_AddDisabledStudent 'Halit','Poyraz','Denizli','2011-08-19','Mert','Poyraz','M',
'5551425698','Eskişehir','Sakarya','29 Ekim',30,26000,'15847569','Autism','2019.12.29','198562458','Autism','2020.05.06',40
```

200	Eskişehir	Yalı Sk.	Saman	35	26000
201	Eskişehir	Altay Cd.	Barkın	37	26000
203	Eskişehir	Altay Cd.	Kaynar	28	26000
237	Eskişehir	Sakarya	29 Ekim	30	26000

125	Şafak	BARÇAK	Hakkari	2014-08-24 ...	6	125
126	Yusuf	ERŞEKERCİ	Konya	2014-11-03 ...	6	126
227	İsim	SOYAD	Malatya	2018-10-10 ...	2	229
228	Halit	POYRAZ	Denizli	2011-08-19 ...	9	230

125	Şerife	BARÇAK	F	5492998919	156	125
126	Seçil	ERŞEKERCİ	F	5441889234	157	126
229	Parentname	PARENTLAST	F	1201201232	236	227
230	Mert	POYRAZ	M	5551425698	237	228

1558964	Mental retar...	2020-09-13 ...
1574697	Autism	2020-01-13 ...
15847569	Autism	2019-12-29 ...
1628985	Physically di...	2020-07-20 ...

177388641	Physically di...	2020-04-23 ...	32	False
178733654	Physically di...	2020-01-03 ...	72	True
179886953	Cerebral pal...	2020-10-02 ...	71	True
198562458	Autism	2020-05-06 ...	40	False

125	1628985	177388641
126	1724698	178733654
227	1475841	5614814
228	15847569	198562458

Name of stored procedure: sp_ListAppropriateInstructors

Definition: This stored procedure lists instructors with expertise suitable for the patient's disease.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_ListAppropriateInstructors]    Script Date: 27.12.2020 19:58:07 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Kursat Acikgoz>
-- Create date: <27.12.2020>
-- Description: <List Appropriate Instructors for Student diagnosis>
-- =============================================
ALTER PROCEDURE [dbo].[sp_ListAppropriateInstructors]
    @stdID int
AS
BEGIN
    SET NOCOUNT ON;

    declare @diagnosis nvarchar(50)

    select @diagnosis=hr.disabilityDiagnosis
    from REPORT_LIST rl, HEALTH_REPORT hr, DISABLED_STUDENT ds
    where rl.healthReportNumber=hr.healthReportNumber
        and rl.studentID=ds.studentID
        and ds.studentID=@stdID

    select e.employeeID, e.firstName, e.lastName,i.profession, e.gsm
    from INSTRUCTOR i, EMPLOYEE e
    where i.profession like '%' + @diagnosis +'%'
        and e.employeeID=i.employeeID

END
```

23	123	1737456	176714168
24	124	1224569	169963588
25	125	1628985	177388641
26	126	1724698	178733654

20	175382324	Autism	2020-03-05 00:00:00	57	0
21	175904506	Visually impaired	2020-03-01 00:00:00	67	0
22	176714168	Visually impaired	2020-09-02 00:00:00	84	1
23	177338829	Autism	2020-04-03 00:00:00	40	0

	employeeID	firstName	lastName	profession	gsm
1	117	Hacı	ÇİFTÇİ	Visually impaired	5381234567
2	118	Kadir	ERGAN	Visually impaired	5371234567

Name of stored procedure: sp_EditRamReport

Definition: This stored procedure used to edit ram report's content.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_EditRamReport]    Script Date: 27.12.2020 20:04:20 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Kursat Acikgoz>
-- Create date: <27.12.2020>
-- Description: <Edit Ram Report>
-- =============================================
ALTER PROCEDURE [dbo].[sp_EditRamReport]
    @ramReportNumber    nvarchar(8),
    @diagnosis nvarchar(50),
    @decisionDate smalldatetime
AS
BEGIN
    SET NOCOUNT ON;
    update rr
    set rr.decisionDate=CONVERT(smalldatetime, @decisionDate, 104), rr.diagnosis=@diagnosis
    from RAM_REPORT rr
    where rr.ramReportNumber=@ramReportNumber
END

```

	ramReportNumber	diagnosis	decisionDate
1	1224569	Physically disabled	2020-02-06 00:00:00
2	1227456	diagnosis	2020-01-01 00:00:00
3	1302368	Visually impaired	2020-11-22 00:00:00

```
sp_EditRamReport 1227456, 'Diagnosis2', '2018.02.04'
```

	ramReportNumber	diagnosis	decisionDate
1	1224569	Physically disabled	2020-02-06 00:00:00
2	1227456	Diagnosis2	2018-02-04 00:00:00
3	1302368	Visually impaired	2020-11-22 00:00:00

Name of stored procedure: sp_EditHealthReport

Definition: This stored procedure used to edit health report's content.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_EditHealthReport]    Script Date: 27.12.2020 20:17:52 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Kursat Acikgoz>
-- Create date: <27.12.2020>
-- Description: <Edit Health Report>
-- =============================================
ALTER PROCEDURE [dbo].[sp_EditHealthReport]
    @healthReportNumber nvarchar(9),
    @disabilityDiagnosis nvarchar(50),
    @applicatonDate smalldatetime,
    @disabilityPercentage tinyint
AS
BEGIN
    SET NOCOUNT ON;

    update hr
    set hr.applicatonDate=CONVERT(smalldatetime, @applicatonDate, 104),
        hr.disabilityDiagnosis=@disabilityDiagnosis,
        hr.disabilityPercentage=@disabilityPercentage
    from HEALTH_REPORT hr
    where hr.healthReportNumber=@healthReportNumber

END
```

24	177388641	Physically disabled	2020-04-23 00:00:00	32	0
25	178733654	Physically disabled	2020-01-03 00:00:00	72	1
26	179886953	Cerebral palsy	2020-10-02 00:00:00	71	1
27	5614814	diagnosis	2020-01-01 00:00:00	50	0

```
exec sp_EditHealthReport '5614814', 'Diagnosis', '2018.10.10',25
```

25	178733654	Physically disabled	2020-01-03 00:00:00	72	1
26	179886953	Cerebral palsy	2020-10-02 00:00:00	71	1
27	5614814	Diagnosis	2018-10-10 00:00:00	25	0

Name of stored procedure: sp_EditCompanyInformation

Definition: This stored procedure used to edit company information.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_EditCompanyInformation]    Script Date: 27.12.2020 20:25:12 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Kursat Acikgoz>
-- Create date: <26.12.2020>
-- Description: <Edit Company Information>
-- =============================================
ALTER PROCEDURE [dbo].[sp_EditCompanyInformation]
    @taxNumber nvarchar(50),
    @companyName nvarchar(150),
    @generalManager nvarchar(100),
    @address nvarchar(50)
AS
BEGIN
 SET NOCOUNT ON;

    update rc
    set rc.address=@address, rc.companyName=@companyName, rc.generalManager=@generalManager,rc.taxNumber=@taxNumber
    from REHAB_CENTER rc

END

```

	centerID	taxNumber	companyName	generalManager	address
1	101	12345678910	Eskişehir Kardelen Rehabilitasyon ve Özel Eğitim ...	Ayşegül Kedek	Eskişehir

```
sp_EditCompanyInformation '12345678910','Eskişehir Kardelen Rhb. ve Özel Eğitim Merkezi', 'Ayşegül Kedek','Eskişehir'
```

	centerID	taxNumber	companyName	generalManager	address
1	101	12345678910	Eskişehir Kardelen Rhb. ve Özel Eğitim Merkezi	Ayşegül Kedek	Eskişehir

---

Name of stored procedure: sp_AddHealthReport

Definition: This stored procedure used to add health reports.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddHealthReport]    Script Date: 27.12.2020 20:30:29 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Kursat Acikgoz>
-- Create date: <26.12.2020>
-- Description: <Add Health Report>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddHealthReport]
    @healthReportNumber nvarchar(9),
    @disabilityDiagnosis nvarchar(50),
    @applicatonDate smalldatetime,
    @disabilityPercentage tinyint
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO HEALTH_REPORT(healthReportNumber, disabilityDiagnosis , applicatonDate, disabilityPercentage)
    VALUES (@healthReportNumber, @disabilityDiagnosis , CONVERT(smalldatetime, @applicatonDate, 104),@disabilityPercentage);
END

```

24	177388641	Physically disabled	2020-04-23 00:00:00	32	0
25	178733654	Physically disabled	2020-01-03 00:00:00	72	1
26	179886953	Cerebral palsy	2020-10-02 00:00:00	71	1
27	5614814	Diagnosis	2018-10-10 00:00:00	25	0

```
exec sp_AddHealthReport '15146','DiagTest','2017.10.10',75
```

	healthReportNumber	disabilityDiagnosis	applicatonDate	disabilityPercentage	severelyDisabled
1	15146	DiagTest	2017-10-10 00:00:00	75	1
2	160144874	Hearing impaired	2020-10-01 00:00:00	39	0

Name of stored procedure: sp_AddInstructor

Definition: This stored procedure used to add instructors.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddInstructor]    Script Date: 27.12.2020 20:34:06 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Ahmet Elburuz Gürbüz>
-- Create date: <26.12.2020>
-- Description: <Add Instructor>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddInstructor]
    @city nvarchar(25),
    @street nvarchar(25),
    @apartmentName nvarchar(25),
    @apartmentNumber smallint,
    @zipCode int,
    @profession nvarchar(50),
    @identityNumber char(11),
    @firstName nvarchar(50),
    @lastName nvarchar(50),
    @gender nvarchar(1),
    @gsm nvarchar(11),
    @salary smallint
AS
BEGIN
 -- SET NOCOUNT ON added to prevent extra result sets from
 -- interfering with SELECT statements.
    SET IDENTITY_INSERT INSTRUCTOR ON
    SET NOCOUNT ON;

    exec sp_AddAddress @city=@city,@street=@street,
                       @apartmentName=@apartmentName,
                       @apartmentNumber=@apartmentNumber, @zipCode=@zipCode

    INSERT INTO EMPLOYEE(identityNumber,firstName,lastName,gender,gsm,salary,addressID)
    VALUES(@identityNumber,@firstName,@lastName,@gender,@gsm,@salary,IDENT_CURRENT('ADDRESS'))

    INSERT INTO INSTRUCTOR(employeeID,profession)
    VALUES (IDENT_CURRENT('EMPLOYEE'),@profession)

    SET IDENTITY_INSERT INSTRUCTOR OFF
END
```

12	118	Visually impaired
13	119	Physically disabled
14	120	Physically disabled
15	121	Kindergarten

20	120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
21	121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
22	122	18384891830	Candan	KİBAR	F	5333168800	3000	122
23	201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203

48	157	Eskişehir	Uçaklar Sk.	Alperen	33	26000
49	200	Eskişehir	Yalı Sk.	Saman	35	26000
50	201	Eskişehir	Altay Cd.	Barkın	37	26000
51	203	Eskişehir	Altay Cd.	Kaynar	28	26000
52	232	Ankara		apname	25	26000

```
45
46  sp_AddInstructor 'Eskişehir','Şair','Göker',16,26000,'ZİHİN',15145325741,'Veli','Çakı','M','1513585216',1528
```

13	119	Physically disabled
14	120	Physically disabled
15	121	Kindergarten
16	208	ZİHİN

Name of stored procedure: sp_AddInstructorList

Definition: This stored procedure used to add instructors.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_addInstructorList]    Script Date: 27.12.2020 20:44:07 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Kursat Acikgoz>
-- Create date: <27.12.2020>
-- Description: <Add To Instructor List>
-- =============================================
ALTER PROCEDURE [dbo].[sp_addInstructorList]
    @employeeID int,
    @studentID int
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO INSTRUCTOR_LIST(employeeID,studentID)
    VALUES (@employeeID,@studentID)

```

23	118	123
24	119	124
25	121	125
26	120	126

```
sp_addInstructorList 121, 123
```

22	121	122
23	118	123
24	121	123
25	119	124

Name of stored procedure: sp_AddParent

Definition: This stored procedure used to add parent.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddParent]    Script Date: 27.12.2020 20:57:36 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Ahmet Önkol
-- Create date: 26.12.2020
-- Description: addParent
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddParent]
    @city nvarchar(25),
    @street nvarchar(25),
    @apartmentName nvarchar(25),
    @apartmentNumber smallint,
    @zipCode int,
    @firstName nvarchar(50),
    @lastName nvarchar(50),
    @gender char(1),
    @gsm char(11),
    @studentID int
AS
BEGIN
    SET NOCOUNT ON;
    ALTER TABLE PARENT NOCHECK CONSTRAINT ALL

    declare @adrsID int
    set @adrsID = IDENT_CURRENT('ADDRESS')

    exec sp_AddAddress @city,@street,@apartmentName,@apartmentNumber, @zipCode

    INSERT INTO PARENT(addressID,firstName,lastName,gender,gsm,studentID)
    VALUES (@adrsID,@firstName,@lastName,@gender,@gsm,@studentID);

END
```

124	Hasret	ALAY	F	5376445269	155	124
125	Şerife	BARÇAK	F	5492998919	156	125
126	Seçil	ERŞEKERCİ	F	5441889234	157	126
227	Parentname	PARENTLAST	F	5557651111	231	225

```
sp_AddParent 'Eskişehir','Yunus Emre','Gül',26,26000,'Fatoş','Kaya','F','05552991214',226
```

124	Hasret	ALAY	F	5376445269	155	124
125	Şerife	BARÇAK	F	5492998919	156	125
126	Seçil	ERŞEKERCİ	F	5441889234	157	126
227	Parentname	PARENTLAST	F	5557651111	231	225
228	Fatoş	KAYA	F	05552991214	234	226

Name of stored procedure: sp_AddPsyhchologist

Definition: This stored procedure used to add psychologist.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddPsyhchologist]    Script Date: 27.12.2020 21:06:51 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Ahmet Elburuz Gürbüz>
-- Create date: <26.12.2020>
-- Description: <Add Psyhchologist>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddPsyhchologist]
    @city nvarchar(25),
    @street nvarchar(25),
    @apartmentName nvarchar(25),
    @apartmentNumber smallint,
    @zipCode int,
    @identityNumber char(11),
    @firstName nvarchar(50),
    @lastName nvarchar(50),
    @gender nvarchar(1),
    @gsm nvarchar(11),
    @salary smallint
AS
BEGIN
 -- SET NOCOUNT ON added to prevent extra result sets from
 -- interfering with SELECT statements.
 SET NOCOUNT ON;

    exec sp_AddAddress @city=@city,@street=@street,
                           @apartmentName=@apartmentName,
                           @apartmentNumber=@apartmentNumber, @zipCode=@zipCode

    INSERT INTO EMPLOYEE(identityNumber,firstName,lastName,gender,gsm,salary,addressID)
    VALUES(@identityNumber,@firstName,@lastName,@gender,@gsm,@salary,IDENT_CURRENT('ADDRESS'))

    INSERT INTO PSYCHOLOGIST(employeeID)
    VALUES (IDENT_CURRENT('EMPLOYEE'))
END
```

	employeeID
▶	105
	201

121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
208	15145325741	Veli	ÇAKI	M	1513585216	2300	234

```
sp_AddPsyhchologist 'Eskişehir','Atatürk','Kurtuluş Savaşı',34,26000,'29778841452','Hilmi','Deniz','M','5075852781',5000
```

	employeeID
▶	105
	201
	211

121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
208	15145325741	Veli	ÇAKI	M	1513585216	2300	234
211	29778841452	Hilmi	DENİZ	M	5075852781	5000	238

Name of stored procedure: sp_AddRamReport

Definition: This stored procedure used to add ram report.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddRamReport]    Script Date: 27.12.2020 22:46:43 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Kursat Acikgoz>
-- Create date: <26.12.2020>
-- Description: <Add Ram Report>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddRamReport]
    @ramReportNumber nvarchar(8),
    @diagnosis nvarchar(50),
    @decisionDate smalldatetime
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO RAM_REPORT(ramReportNumber, diagnosis , decisionDate)
    VALUES (@ramReportNumber, @diagnosis , CONVERT(smalldatetime, @decisionDate, 104));
END
```

1745678	Cerebral palsy	2020-09-18 00:00:...
1746324	Hearing impaired	2020-09-05 00:00:...
1782965	Developmental language diso...	2020-03-29 00:00:...
1789567	Cerebral palsy	2019-08-07 00:00:...

```
sp_AddRamReport '14658731','Cerebral Palsy','2020.12.06'
```

1745678	Cerebral palsy	2020-09-18 0...
1746324	Hearing impaired	2020-09-05 0...
1782965	Developmental langua...	2020-03-29 0...
1789567	Cerebral palsy	2019-08-07 0...
18658731	Autism	2020-11-06 0...

Name of stored procedure: sp_AddReportList

Definition: This stored procedure used to add reports to report list

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddReportList]    Script Date: 27.12.2020 22:53:55 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Kursat Acikgoz>
-- Create date: <27.12.2020>
-- Description: <add report list>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddReportList]
    @studentID int,
    @ramReportNumber nvarchar(8),
    @healthReportNumber nvarchar(9)
AS
BEGIN
    SET NOCOUNT ON;
    ALTER TABLE REPORT_LIST NOCHECK CONSTRAINT ALL

   INSERT INTO REPORT_LIST (healthReportNumber,ramReportNumber,studentID)
   VALUES(@healthReportNumber,@ramReportNumber,@studentID+1)
END

```

124	1224569	169963588
125	1628985	177388641
126	1724698	178733654
227	1475841	5614814

```
sp_AddReportList 123,'14567216','146982354'
```

124	1224569	169963588
125	1628985	177388641
126	1724698	178733654
227	1475841	5614814
228	14567216	146982354

Name of stored procedure: sp_AddReports

Definition: This stored procedure used to add report

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddReports]    Script Date: 27.12.2020 22:59:44 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Kürşat Açıkgöz>
-- Create date: <26.12.2020>
-- Description: <Add Reports>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddReports]
@ramReportNumber nvarchar(8),
@diagnosis nvarchar(50),
@decisionDate smalldatetime,
@healthReportNumber nvarchar(9),
@disabilityDiagnosis nvarchar(50),
@applicatonDate smalldatetime,
@disabilityPercentage tinyint
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO RAM_REPORT(ramReportNumber,diagnosis, decisionDate)
    VALUES (@ramReportNumber,@diagnosis, @decisionDate);

    INSERT INTO HEALTH_REPORT(healthReportNumber, disabilityDiagnosis,applicatonDate,disabilityPercentage)
    VALUES (@healthReportNumber, @disabilityDiagnosis,@applicatonDate, @disabilityPercentage);

END
```

Ram_Report

1746324	Hearing im...	2020-09-05 ...
1782965	Developme...	2020-03-29 ...
1789567	Cerebral pal...	2019-08-07 ...
18658731	Autism	2020-11-06 ...

Health_Report

177338829	Autism	2020-04-03 ...	40	False
177388641	Physically di...	2020-04-23 ...	32	False
178733654	Physically di...	2020-01-03 ...	72	True
179886953	Cerebral pal...	2020-10-02 ...	71	True

```
sp_AddReports '19638731','Autism','2020.11.06','199678731','Autism','2020.08.02',60
```

1782965	Developme...	2020-03-29 ...
1789567	Cerebral pal...	2019-08-07 ...
18658731	Autism	2020-11-06 ...
19638731	Autism	2020-11-06 ...

177388641	Physically di...	2020-04-23 ...	32	False
178733654	Physically di...	2020-01-03 ...	72	True
179886953	Cerebral pal...	2020-10-02 ...	71	True
199678731	Autism	2020-08-02 ...	60	False

Name of stored procedure: sp_AddStaff

Definition: This stored procedure used to add staff

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddStaff]    Script Date: 27.12.2020 23:17:04 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Ahmet Elburuz Gürbüz>
-- Create date: <26.12.2020>
-- Description: <Add Staff>
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddStaff]
    @city nvarchar(25),
    @street nvarchar(25),
    @apartmentName nvarchar(25),
    @apartmentNumber smallint,
    @zipCode int,
    @jobField nvarchar(50),
    @identityNumber char(11),
    @firstName nvarchar(50),
    @lastName nvarchar(50),
    @gender nvarchar(1),
    @gsm nvarchar(11),
    @salary smallint
AS
BEGIN
 -- SET NOCOUNT ON added to prevent extra result sets from
 -- interfering with SELECT statements.
 SET NOCOUNT ON;
    exec sp_AddAddress @city=@city,@street=@street,
                       @apartmentName=@apartmentName,
                       @apartmentNumber=@apartmentNumber, @zipCode=@zipCode

    INSERT INTO EMPLOYEE(identityNumber,firstName,lastName,gender,gsm,salary,addressID)
    VALUES(@identityNumber,@firstName,@lastName,@gender,@gsm,@salary,IDENT_CURRENT('ADDRESS'));

    INSERT INTO STAFF(employeeID,jobField)
    VALUES (IDENT_CURRENT('EMPLOYEE'),@jobField)

END
```

Address Table

157	Eskişehir	Uçaklar Sk.	Alperen	33	26000
200	Eskişehir	Yalı Sk.	Saman	35	26000
201	Eskişehir	Altay Cd.	Barkın	37	26000
203	Eskişehir	Altay Cd.	Kaynar	28	26000

Staff Table

103	Temizlik Gö...
104	Aşçı
106	Servis şoförü
122	Sekreter

Employee Table

120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203

```
sp_AddStaff 'Eskişehir','Turanbey','Muzaffer',46,26000,'Tekniker','29221567852','Hasan','Uslu','M','5075554411',2300
```

200	Eskişehir	Yalı Sk.	Saman	35	26000
201	Eskişehir	Altay Cd.	Barkın	37	26000
203	Eskişehir	Altay Cd.	Kaynar	28	26000
237	Eskişehir	Turanbey	Muzaffer	46	26000

104	Aşçı
106	Servis şoförü
122	Sekreter
210	Tekniker

121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
210	29221567852	Hasan	USLU	M	5075554411	2300	238

Name of stored procedure: sp_AddTherapy

Definition: This stored procedure used to add therapy

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_AddTherapy]    Script Date: 27.12.2020 23:26:50 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Ahmet Önkol
-- Create date: 26.12.2020
-- Description: addTherapySession
-- =============================================
ALTER PROCEDURE [dbo].[sp_AddTherapy]
    @therapyDescription nvarchar(250),
    @sessionDate smalldatetime,
    @employeeID int,
    @studentID int
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO THERAPY_SESSIONS(therapyDescription,sessionDate,employeeID,studentID)
    VALUES (@therapyDescription,CONVERT(smalldatetime, @sessionDate, 104),@employeeID,@studentID);
END

```

23	routine control	2020-01-05 ...	105	123
24	routine control	2020-12-27 ...	201	124
25	routine control	2020-01-12 ...	105	125
26	routine control	2020-12-27 ...	105	126

```
sp_AddTherapy 'Urgent session due to violent tendency','2020.12.27',105,121
```

24	routine control	2020-12-27 ...	201	124
25	routine control	2020-01-12 ...	105	125
26	routine control	2020-12-27 ...	105	126
201	Urgent session due to violent tendency	2020-12-27 ...	105	121

Name of stored procedure: sp_DeleteInstructor

Definition: This stored procedure used to delete instructor

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_DeleteInstructor]    Script Date: 27.12.2020 23:50:05 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Ahmet Elburuz Gürbüz>
-- Create date: <26.12.2020>
-- Description: <Delete Instructor>
-- =============================================
ALTER PROCEDURE [dbo].[sp_DeleteInstructor]
    @employeeID int
AS
BEGIN
 -- SET NOCOUNT ON added to prevent extra result sets from
 -- interfering with SELECT statements.
 SET NOCOUNT ON;
    DELETE
    FROM INSTRUCTOR_LIST
    WHERE INSTRUCTOR_LIST.employeeID = @employeeID

    DELETE
    FROM INSTRUCTOR
    WHERE INSTRUCTOR.employeeID = @employeeID

    DELETE
    FROM EMPLOYEE
    WHERE EMPLOYEE.employeeID = @employeeID
END
```

Instrutor Table

117	Visually imp...
118	Visually imp...
119	Physically di...
120	Physically di...
121	Kindergarten

Instructor_List Table

118	123
119	124
121	125
120	126

Employee

119	12345678891	Zehra	ÖZKÇK	F	5361234567	3675	119
120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
121	34567889101	Ahmet	ERCAN	M	5341234567	3600	121
122	18384891830	Candan	KİBAR	F	5333168800	3000	122

```
sp_DeleteInstructor 121
```

118	Visually imp...
119	Physically di...
120	Physically di...
NULL	*NULL*

118	123
119	124
120	126
NULL	*NULL*

119	12345678891	Zehra	ÖZKÇK	F	5361234567	3675	119
120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
122	18384891830	Candan	KİBAR	F	5333168800	3000	122

Name of stored procedure: sp_DeletePsyhchologist

Definition: This stored procedure used to delete psychologist.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_DeletePsyhchologist]    Script Date: 28.12.2020 00:02:48 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Ahmet Elburuz Gürbüz>
-- Create date: <26.12.2020>
-- Description: <Delete Psyhchologist>
-- =============================================
ALTER PROCEDURE [dbo].[sp_DeletePsyhchologist]
    @employeeID int
AS
BEGIN
 -- SET NOCOUNT ON added to prevent extra result sets from
 -- interfering with SELECT statements.
 SET NOCOUNT ON;
    DELETE
    FROM THERAPY_SESSIONS
    WHERE THERAPY_SESSIONS.employeeID = @employeeID

    DELETE
    FROM PSYCHOLOGIST
    WHERE PSYCHOLOGIST.employeeID = @employeeID

    DELETE
    FROM EMPLOYEE
    WHERE EMPLOYEE.employeeID = @employeeID
END
```

	employeeID
▶	105
	201

120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
201	18735017601	Rümeysa	SEVER	F	5558133201	5000	203
210	29221567852	Hasan	USLU	M	5075554411	2300	238

22	routine cont...	2020-01-06 ...	201	122
23	routine cont...	2020-01-05 ...	105	123
24	routine cont...	2020-12-27 ...	201	124
25	routine cont...	2020-01-12 ...	105	125
26	routine cont...	2020-12-27 ...	105	126

```
sp_DeletePsyhchologist 201
```

	employeeID
▶	105
*	*NULL*

120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
210	29221567852	Hasan	USLU	M	5075554411	2300	238
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

21	routine cont...	2020-01-03 ...	105	121
23	routine cont...	2020-01-05 ...	105	123
25	routine cont...	2020-01-12 ...	105	125
26	routine cont...	2020-12-27 ...	105	126

Name of stored procedure: sp_DeleteStaff

Definition: This stored procedure used to delete staff

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_DeleteStaff]    Script Date: 28.12.2020 00:26:03 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:  <Ahmet Elburuz Gürbüz>
-- Create date: <26.12.2020>
-- Description: <Delete Staff>
-- =============================================
ALTER PROCEDURE [dbo].[sp_DeleteStaff]
    @employeeID int
AS
BEGIN
 -- SET NOCOUNT ON added to prevent extra result sets from
 -- interfering with SELECT statements.
 SET NOCOUNT ON;

    DELETE
    FROM STAFF
    WHERE STAFF.employeeID = @employeeID

    DELETE
    FROM EMPLOYEE
    WHERE EMPLOYEE.employeeID = @employeeID
END

```

104	Aşçı
106	Servis şoförü
122	Sekreter
210	Tekniker

119	12345678891	Zehra	ÖZKÇK	F	5361234567	3675	119
120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
210	29221567852	Hasan	USLU	M	5075554411	2300	238

```
sp_DeleteStaff 210
```

104	Aşçı
106	Servis şoförü
122	Sekreter
NULL	*NULL*

119	12345678891	Zehra	ÖZKÇK	F	5361234567	3675	119
120	23456788910	Gülbeyaz	ÖZALP	F	5351234567	2750	120
122	18384891830	Candan	KİBAR	F	5333168800	3000	122
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Name of stored procedure: sp_DeleteStudent

Definition: This stored procedure used to delete disabled student.

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_DeleteStudent]    Script Date: 28.12.2020 00:25:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Kursat Acikgoz>
-- Create date: <27.12.2020>
-- Description: <Delete Student>
-- =============================================
ALTER PROCEDURE [dbo].[sp_DeleteStudent]
    @stdID int
AS
BEGIN
    SET NOCOUNT ON;

    delete
    from STUDENT_LIST
    where @stdID=studentID

    delete
    from THERAPY_SESSIONS
    where @stdID=studentID

    delete
    from REPORT_LIST
    where @stdID=studentID

    delete
    from DISABLED_STUDENT
    where @stdID=studentID

END
```

Disabled Student Table

119	Cumhur	ÇÖLBAY	Ardahan	2012-03-30 ...	8	119
120	Sinan	ALKAN	Çanakkale	2012-11-21 ...	8	120
121	Mervenur	SERTDEMİR	Adıyaman	2013-04-11 ...	7	121
122	Umut	KARADAŞ	Tokat	2013-05-01 ...	7	122

Student List Table

101	121
101	122
101	123
101	124

Therapy Session Table

23	routine cont...	2020-01-05 ...	105	123
25	routine cont...	2020-01-12 ...	105	125
26	routine cont...	2020-12-27 ...	105	126
201	Urgent sessi...	2020-12-27 ...	105	121

Report List Table

119	1789567	179886953
120	1302368	164828212
121	1703597	168837557
122	1717356	175904506

```
sp_DeleteStudent 121
```

120	Sinan	ALKAN	Çanakkale	2012-11-21 ...	8	120
122	Umut	KARADAŞ	Tokat	2013-05-01 ...	7	122
123	Havva	MAZLUM	Iğdır	2014-03-11 ...	6	123
124	Handan	ALAY	Diyarbakır	2014-06-23 ...	6	124

101	119
101	120
101	122
101	123

23	routine cont...	2020-01-05 ...	105	123
25	routine cont...	2020-01-12 ...	105	125
26	routine cont...	2020-12-27 ...	105	126
NULL	*NULL*	*NULL*	*NULL*	*NULL*

119	1789567	179886953
120	1302368	164828212
122	1717356	175904506

Name of stored procedure: sp_DeleteTherapySession

Definition: This stored procedure used to delete therapy session

- Stored procedure
- The table before the stored procedure is called
- Call stored procedure
- The table after the stored procedure call.

```
USE [REHABILITATION_CENTER]
GO
/****** Object:  StoredProcedure [dbo].[sp_DeleteTherapy]    Script Date: 28.12.2020 00:47:18 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Ahmet Önkol
-- Create date: 26.12.2020
-- Description: DeleteTherapy
-- =============================================
ALTER PROCEDURE [dbo].[sp_DeleteTherapy]
    @sessionID int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;


    -- Delete statements for procedure here
    DELETE FROM THERAPY_SESSIONS
    WHERE sessionID= @sessionID
END
```

19	routine cont...	2020-01-01 ...	105	119
23	routine cont...	2020-01-05 ...	105	123
25	routine cont...	2020-01-12 ...	105	125
26	routine cont...	2020-12-27 ...	105	126

```
sp_DeleteTherapy 25
```

19	routine cont...	2020-01-01 ...	105	119
23	routine cont...	2020-01-05 ...	105	123
26	routine cont...	2020-12-27 ...	105	126