

1) a) This is the entities I have chosen for designing relational database for DentistX:

**Patient (**

PatientID, FirstName, LastName, Gender, PhoneNumber, Email, Address, ZipCode,  
DateOfBirth, Age

)

**Situation (**

SituationID, PatientID, TheIssue

)

**Employee (**

EmployeeID, FirstName, LastName, WorkingPosition, Email, PhoneNumber, Address,  
ZipCode

)

**Appointment (**

AppointmentID, PatientID, TreatmentID, EmployeeID, BookingID, HygienistAppointment,  
Hygienist, Trainee, AppointmentDate, AppointmentTime, TotalHours

)

**Treatment (**

TreatmentID, PatientID, SituationID, TreatmentName, TreatmentFee

)

**Room (**

BookingID, TreatmentID, RoomName, TreatmentRoom, BookingTimeStartToEnd, Date

)

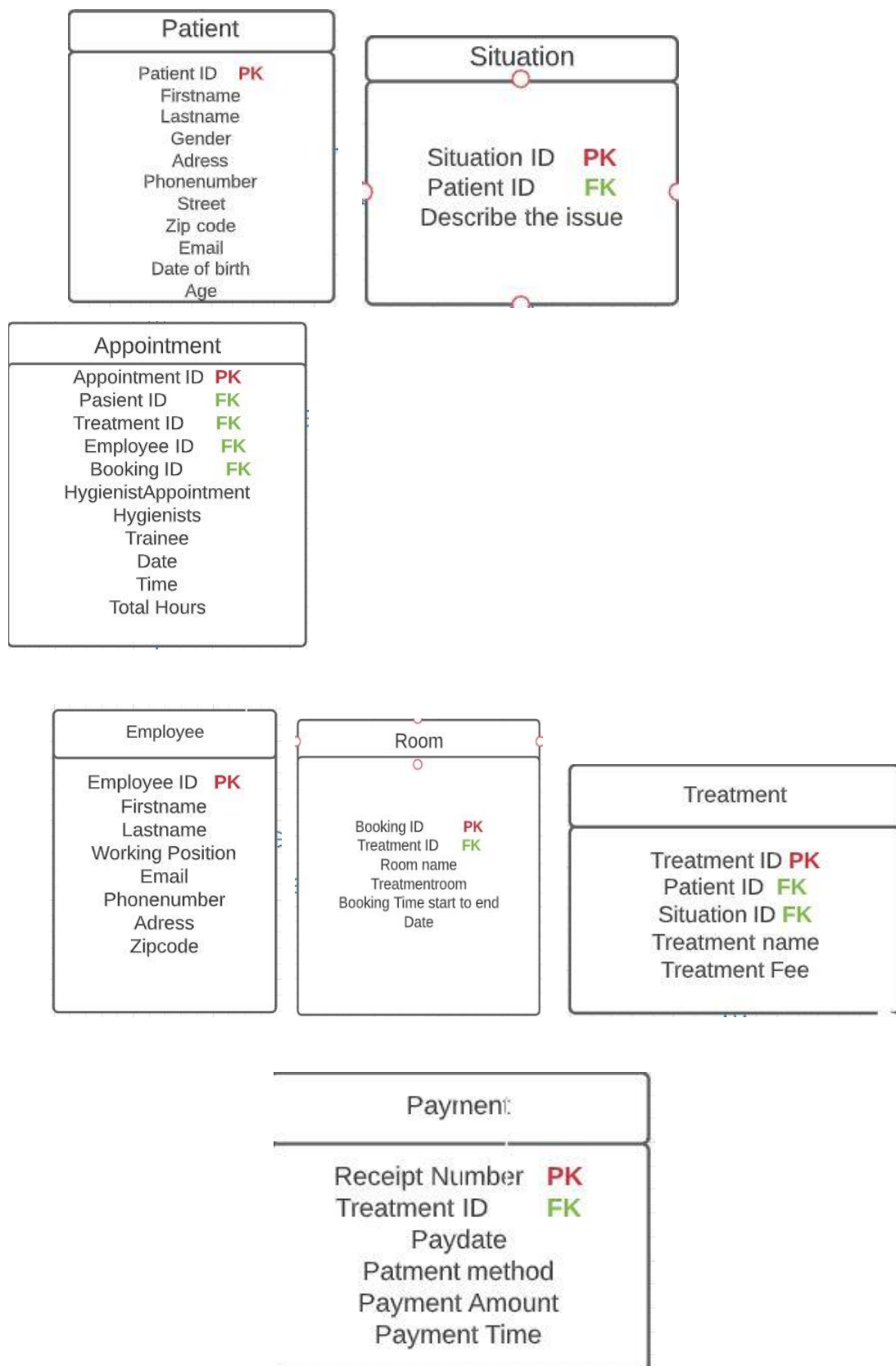
**Payment (**

ReceiptNumber, TreatmentID, Paydate, PaymentMethod, PaymentAmount, PaymentTime

)

b) I identify "Patient", "Situation", "Employee", "Appointment", "Treatment", "Room", and "Payment" as my attributes associated with the relations identified in task 1a.

c) These are the tables I have designed to identify primary key and foreign key. The red "PK" stands for primary key, and the green "FK" stands for foreign key. Primary key is used for to identify each row of data in a table. Foreign key is used for to define the relationship with another table. Candidate keys are the minimal subset of super key (Silberschatz, Korth, and Sudarshan, 2019). Which means for example the patients email or phone number.

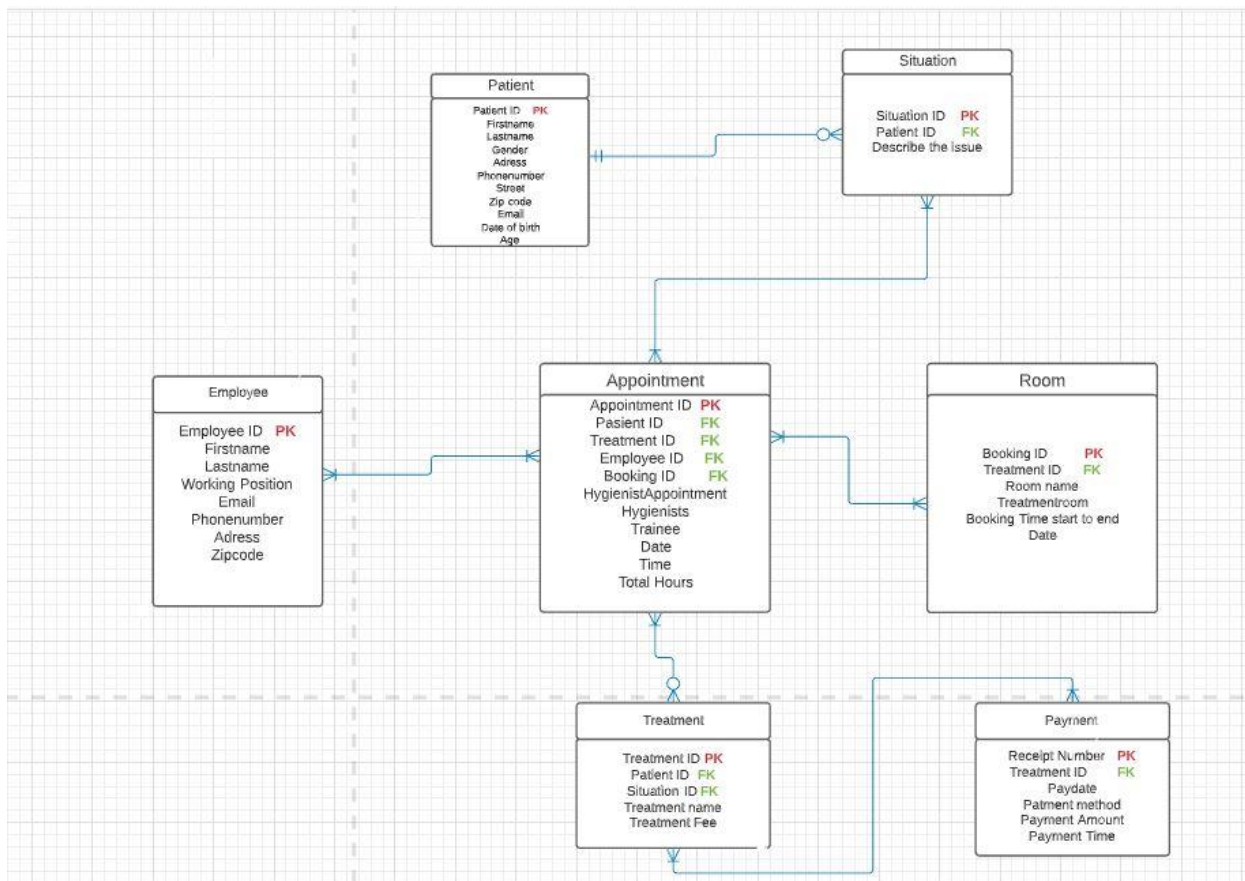


**d)** This is an ER-model for Dentistx. I created this ER-model to show the receptionist how this database works. As you can see it is only one and only one patient who can have 0 or multiple situations. The reason is because, this clinic has all their patients in their system from before, and its only one patient who can order an appointment for their own. Therefore, it leads us to situation where the patient can have zero or multiple situations. A situation can lead us to an appointment, and a patient can have one or multiple appointments for different situations.

To have an appointment it should consist of one or multiple situations.

It is the appointment table who binds everything here in this database. An employee should at least have one or several appointments in the clinic, and to have an appointment should consist of one or several employees. To have an appointment the dentist/hygienist/trainees need a treatment room to treat the patients. Which leads us to have a treatment room. The patient can have multiple appointments and can be treated in different room on the same appointment; therefore, one appointment can have one or several rooms. To have a treatment room it should consist of one or multiple appointments. From the appointment table it leads us to treatment. It is not always that the patient needs to be treated, therefore an appointment can have zero or several treatments. Treatment should consist of one or several appointments. This leads us to the last table, which is the payment. After the treatment, the patient needs to pay. The patient only must pay if they have a treatment, therefore a patient pays when they have been treated. To be treated the patient must pay one or several times considering number of treatments. This ER-model describes to the receptionist that appointments bind everything together. They will have an overview over what is happening at the clinic. Everything from which employee who is busy to which room is fully booked.

Every table has relatable entities. The receptionist has all info on every patient. All about where they live to when they were born. This defines every individual patient. Situation table defines which patient has the different situations and their issue. Appointment table describes everything from which dentist is having their patient, to when the appointment is and their treatment. This table gives a lot of information. Treatment table describes which patients is getting treated. Employee table gives us the information about the hygienist, trainees, and the senior dentists. Room table describes which room the different treatments is having place. The payment table show us which treatment the patient pays for. The 'TreatmentID' is a foreign key, so we have access to see which patient it belongs to.



e) This is the database that has been typed in Microsoft SQL Server Management Studio 2018.

```

Create Table Employee(
EmployeeID INT,
FirstName VARCHAR(255),
LastName VARCHAR(255),
WorkingPosition VARCHAR(255),
Email VARCHAR(255),
PhoneNumber VARCHAR(255),
Address VARCHAR(255),
ZipCode INT,
PRIMARY KEY (EmployeeID)
)
  
```

```

Create Table Patient(
PatientID INT,
FirstName VARCHAR(255),
LastName VARCHAR(255),
Gender VARCHAR(255),
PhoneNumber VARCHAR(255),
Email VARCHAR(255),
Address VARCHAR(255),
ZipCode INT,
DateOfBirth DATE,
Age INT,
PRIMARY KEY (PatientID)
)
  
```

```

Create Table Situation(
  SituationID INT,
  PatientID INT,
  TheIssue VARCHAR(255)
  PRIMARY KEY (SituationID)
  FOREIGN KEY (PatientID) REFERENCES Patient
)

```

```

Create Table Treatment(
  TreatmentID INT,
  PatientID INT,
  SituationID INT,
  TreatmentName VARCHAR(255),
  TreatmentFee VARCHAR(255),
  PRIMARY KEY (TreatmentID),
  FOREIGN KEY (PatientID) REFERENCES Patient,
  FOREIGN KEY (SituationID) REFERENCES Situation
)

```

```

Create Table Room(
  BookingID INT,
  TreatmentID INT,
  RoomName VARCHAR(200),
  TreatmentRoom VARCHAR(200),
  BookingTimeStartToEnd VARCHAR(200),
  Date DATE,
  PRIMARY KEY (BookingID),
  FOREIGN KEY (TreatmentID) REFERENCES Treatment
)

```

```

Create Table Appointment(
  AppointmentID INT,
  PatientID INT,
  TreatmentID INT,
  EmployeeID INT,
  BookingID INT,
  HygienistAppointment VARCHAR(25),
  Hygienists VARCHAR(255),
  Trainee VARCHAR(255),
  AppointmentDate DATE,
  AppointmentTime TIME,
  TotalHours VARCHAR(255),
  PRIMARY KEY (AppointmentID),
  FOREIGN KEY (PatientID) REFERENCES Patient,
  FOREIGN KEY (TreatmentID) REFERENCES Treatment,
  FOREIGN KEY (EmployeeID) REFERENCES Employee,
  FOREIGN KEY (BookingID) REFERENCES Room
)

```

```

Create Table Payment(
  ReceiptNumber INT,
  TreatmentID INT,
  Paydate DATE,
  PaymentMethod VARCHAR(255),
  PaymentAmount VARCHAR(255),
  PaymentTime VARCHAR(255),
  PRIMARY KEY (ReceiptNumber),
  FOREIGN KEY (TreatmentID) REFERENCES Treatment
)

```

- 2) I have more than ten records in each Primary Table and more than twenty dummy records in each detailed/child table:

### Patient table

LAPTOP-AEH23G3E\...ntistX - dbo.Room				LAPTOP-AEH23G3E\...stX - dbo.Payment		LAPTOP-AEH23G3E\S...istX - dbo.Patient				
	PatientID	FirstName	LastName	Gender	PhoneNumber	Email	Adress	ZipCode	DateOfBirth	Age
	1001	Harvey	Specter	Male	+45363928	harvey.specter@pearsonspecter.com	Fifth Avenue 5B	7051	1975-02-03	46
	1002	Jessica	Pearson	Female	+45362829	jessica.pearson@pearsonspecter.com	Bond Street 10B	7032	1970-11-21	51
	1003	Donna	Paulsen	Female	+45273829	donna.paulsen@pearsonspecter.com	Sixth Avenue 4D	7053	1978-10-21	43
	1004	Louis	Litt	Male	+45328373	louis.litt@pearsonspecter.com	Gotham Street 6B	7056	1964-04-18	57
	1005	Rachael	Zane	Female	+45372293	rachael.zane@pearsonspecter.com	Hill Street 7B	7045	1991-02-16	30
	1006	Michael	Ross	Male	+45372983	michael.ross@pearsonspecter.com	Hill Street 7B	7032	1989-11-21	32
	1007	Dana	Scott	Female	+45362832	dana.scott@pearsonspecter.com	Liberty Avenue 3B	7045	1977-02-02	44
	1008	Katrina	Bennett	Female	+45738292	katrina.bennett@pearsonspecter.com	Kingston Street 2A	7061	1968-02-24	53
	1009	Robert	Zane	Male	+45382928	robert.zane@pearsonspecter.com	Sax Avenue 9F	7073	1957-08-28	64
	1010	Logan	Sanders	Male	+45372635	logan.sanders@pearsonspecter.com	Sars Street 1A	7058	1988-08-26	33
	1011	Daniel	Hardman	Male	+45362838	daniel.hardman@pearsonspecter.com	Chicago Avenue 8F	7098	1955-09-04	66
	1012	Samantha	Wheeler	Female	+45373922	samantha.wheeler@pearsonspecter.com	York Avenue 4B	7068	1963-05-23	58
	1013	Charles	Forstman	Male	+45373734	charles.forstman@pearsonspecter.com	Oak Avnue 6A	7093	1998-11-11	23
	1014	Ava	Hassingt...	Female	+45362819	ava.hassington@hassingtonoil.com	Mary Avenue 4C	7054	1956-10-12	65
	1015	Eric	Kaldor	Male	+45373827	eric.kaldor@randkaldorzane.com	Olav Street 12B	7081	1994-07-30	27
»	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### Situation table

LAPTOP-AEH23G3E\...stX - dbo.Payment			LAPTOP-AEH23G3E\S
	SituationID	PatientID	Thelssue
	5670	1001	Pain in wisdom teeth
	5671	1002	Cleaning
	5672	1003	Yellow teeth
	5673	1004	Cleaning
	5674	1005	Lost teeth after an accident
	5675	1006	Hurting while cheewing
	5676	1007	Yellow teeth
	5677	1008	Cleaning
	5678	1009	Cleaning
	5679	1010	Bad Breath
	5680	1011	Pain in wisdom teeth
	5681	1012	Yellow teeth
	5682	1013	Bad breath
	5683	1014	Yearly check
	5684	1015	Broken tooth
✎	5685	1007	Yearly check
	5686	1005	New tooth replacement
	5687	1015	New tooth replacement
	5688	1002	Pain in wisdom teeth
	5689	1006	Cleaning
	5690	1010	Yellow teeth
*	NULL	NULL	NULL



## Appointment table

LAPTOP-AEH23G3E\S...istX - dbo.Patient			LAPTOP-AEH23G3E\S...X - dbo.Situation			LAPTOP-AEH23G3E\S...dbo.Appointment		
	AppointmentID	PatientID	TreatmentID	EmployeeID	BookingID	HygienistAppointment	Hygienists	Trainee
	72606832	1001	97194052	2811	10004	NULL	NULL	Jonathan Sidwell
	72606833	1002	97194053	2827	10005	Yes	Faye Richardson	Edward Darby
	72606834	1003	97194054	2815	10006	Yes	Frank Gallo	Harold Gunderson
	72606835	1004	97194055	2819	10007	Yes	Gretchen Bodinski	NULL
	72606836	1005	97194056	2820	10008	NULL	NULL	Sean Cahill and Henry ...
	72606837	1006	97194057	2822	10009	NULL	NULL	Jack Soloff
	72606838	1007	97194058	2817	10010	Yes	Monica Eton	Jenny Griffith
	72606839	1008	97194059	2827	10011	Yes	Faye Richardson	NULL
	72606840	1009	97194060	2828	10012	Yes	Frank Gallo	NULL
	72606841	1010	97194061	2831	10013	Yes	Greychen Bodinski	NULL
	72606842	1011	97194062	2811	10014	NULL	NULL	Nikki Sordel and Moni...
	72606843	1012	97194063	2817	10015	Yes	Frank Gallo	Nicholas Palermo
	72606844	1013	97194064	2819	10016	Yes	Monica Eton	Edward Darby
	72606845	1014	97194065	2822	10017	NULL	NULL	Harold Gunderson
	72606846	1015	97194066	2831	10018	NULL	NULL	Jenny Griffith
	72606847	1007	97194067	2826	10019	NULL	NULL	Sean Cahill
	72606848	1005	97194068	2820	10020	NULL	NULL	Nicholas Palermo and ...
	72606849	1015	97194069	2817	10021	NULL	Gretchen Bodinski	Jonathan Sidwell
	72606850	1002	97194070	2827	10022	NULL	NULL	Henry Gerard
	72606851	1006	97194071	2822	10023	Yes	NULL	Monica Eton
	72606852	1010	97194072	2817	NULL	Yes	Faye Richardson	Jack Soloff
»	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

AppointmentDate	AppointmentTime	TotalHours
2021-11-27	09:00:00	1.5
2021-11-21	09:00:00	1.5
2021-12-25	09:00:00	1.5
2021-11-25	13:00:00	1.5
2021-11-26	13:00:00	1.5
2021-12-23	10:45:00	1.5
2021-11-27	10:45:00	1.5
2021-11-18	09:00:00	1.5
2021-11-18	14:45:00	1.5
2021-11-26	10:45:00	1.5
2021-11-28	13:00:00	1.5
2021-11-29	09:00:00	1.5
2021-11-30	10:45:00	1.5
2021-12-15	14:45:00	1.5
2021-11-22	09:00:00	1.5
2021-12-22	10:45:00	1.5
2021-11-28	14:45:00	1.5
2021-11-27	09:00:00	1.5
2021-11-24	13:00:00	1.5
2021-11-24	13:00:00	1.5
2021-12-21	13:00:00	1.5
NULL	NULL	NULL

## Employee table

LAPTOP-AEH23G3E\S...X - dbo.Situation		LAPTOP-AEH23G3E\S...X - dbo.Appointment		LAPTOP-AEH23G3E\S...X - dbo.Employee				
	EmployeeID	FirstName	LastName	WorkingPosition	Email	PhoneNumber	Address	ZipCode
	2811	Rosalie	Williams	Senior Dentist	rosalie.williams@den...	+43648403	Sars Street 4A	5053
	2812	Jonathan	Sidwell	Trainee	jonathan.sidwell@den...	+43749494	Ballock Avenue ...	5044
	2813	Edward	Darby	Trainee	edward.darby@den...	+43648493	Kenwood Street ...	5098
	2814	Faye	Richardson	Hygienist	faye.richardson@den...	+43648392	Mars Street 5B	5064
	2815	Cameron	Dennis	Senior Dentist	cameron.dennis@den...	+43749404	Saturn Avenue 8F	5067
	2816	Harold	Gunderson	Trainee	harold.gunderson@d...	+43629202	Jupiter Street 4C	7089
	2817	Thomas	Kessler	Senior Dentist	thomas.kessler@den...	+43940484	Uranus Avenue ...	5064
	2818	Frank	Gallo	Hygienist	frank.gallo@dentistx.c...	+43648494	Alley Street 44F	7043
	2819	Alex	Williams	Senior Dentist	alex.williams.dentistx.c...	+43647444	Dahls Street 7A	5065
	2820	Sheila	Sazs	Senior Dentist	sheila.sazs@dentistx.c...	+43648494	Tors Veg 3A	5066
	2821	Sean	Cahill	Trainee	sean.cahill@dentistx.c...	+43637222	New Avenue 8B	5079
	2822	Jeff	Malone	Senior Dentist	jeff.malone@dentistx.c...	+43647828	Park Avenue 8B	7045
	2823	Gretchen	Bodinski	Hygienist	gretchen.bodinski@d...	+43648393	Kajana Street 1B	5068
	2824	Henry	Gerard	Trainee	henry.gerard@dentist...	+43839333	Main Avenue 11F	5097
	2825	Jack	Soloff	Trainee	jack.soloff@dentistx...	+43623637	2nd Avenue 8B	5084
	2826	Jenny	Griffith	Trainee	jenny.griffith@dentist...	+43694445	Steinway Street ...	7095
	2827	Lily	Specter	Senior Dentist	lily.specter@dentistx.c...	+43729273	Wood Street 9B	7054
	2828	Nick	Zegan	Senior Dentist	nick.zegan@dentistx.c...	+43749505	Broadway Aven...	7032
	2829	Nikki	Sordel	Trainee	nikki.sordel@dentistx...	+43788244	Krueger Street 7B	5015
	2830	Paul	Porter	Trainee	paul.porter@dentistx...	+43859953	Jackson Avenue ...	7055
	2831	Nigel	Nesbitt	Senior Dentist	nigel.nesbitt@dentistx...	+43749993	Skillman Avenue...	7096
	2832	Nicholas	Palermo	Trainee	nicholas.palermo@de...	+43647595	SA Avenue 4A	7065
	2833	Monica	Eton	Hygienist	monica.eton@dentist...	+43527393	Zen Avenue 7B	7054

## Room table

LAPTOP-AEH23G3E\...dbo.Appointment			LAPTOP-AEH23G3E\...X - dbo.Employee		LAPTOP-AEH23G3E\...ntistX - dbo.R	
	Booking...	TreatmentID	RoomName	TreatmentRoom	BookingTimeStartToEnd	Date
	10004	97194052	G	Surgery Room	09:00 - 10:30	2021-11-27
	10005	97194053	B	Hygiene Room	09:00 - 10:30	2021-11-21
	10006	97194054	C	Teeth Whitening Room	09:00 - 10:30	2021-12-25
	10007	97194055	B	Hygiene Room	13:00 - 14:30	2021-11-25
	10008	97194056	D	Surgery Room	13:00 - 14:30	2021-11-26
	10009	97194057	E	Dental Examination Room	10:45 - 12:15	2021-12-23
	10010	97194058	F	Teet Whitening Room	10:45 - 12:15	2021-11-27
	10011	97194059	B	Hygiene Room	09:00 - 10:30	2021-11-18
	10012	97194060	B	Hygiene Room	14:45 - 16:15	2021-11-18
	10013	97194061	B	Hygiene Room	10:45 - 12:15	2021-11-26
	10014	97194062	G	Surgery Room	13:00 - 14:30	2021-11-28
	10015	97194063	F	Teeth Whitening Room	09:00 - 10:30	2021-11-29
	10016	97194064	B	Hygiene Room	10:45 - 12:15	2021-11-30
	10017	97194065	E	Dental Examination	14:45 - 16:15	2021-12-15
	10018	97194066	A	Dental Examination	09:00 -10:30	2021-11-22
	10019	97194067	A	Dental Examination	10:45 - 12:15	2021-12-22
	10020	97194068	D	Surgery Room	14:45 - 16:15	2021-11-28
	10021	97194069	G	Surgery Room	09:00 - 10:30	2021-11-27
	10022	97194070	G	Surgery Room	13:00 - 14:30	2021-11-24
	10023	97194071	B	Hygiene Room	13:00 - 14:30	2021-11-24
	10024	97194072	C	Teeth Whitening Room	13:00 - 14:30	2021-12-21



## Treatment table

LAPTOP-AEH23G3E\...X - dbo.Employee		LAPTOP-AEH23G3E\...ntistX - dbo.Room		LAPTOP-AEH23G3E\...X - dbo	
	TreatmentID	PatientID	SituationID	TreatmentName	TreatmentFee
	97194052	1001	5670	Draw wisdom teeth	2500kr
	97194053	1002	5671	Hygiene measures	400kr
	97194054	1003	5672	Teeth whitening	1900kr
	97194055	1004	5673	Hygiene Measure	500kr
	97194056	1005	5674	Dental Examination	680kr
	97194057	1006	5675	Dental Examination	680kr
▶	97194058	1007	5676	Teeth Whitening	1900kr
	97194059	1008	5677	Hygiene Measures	550kr
	97194060	1009	5678	Hygiene Measures	550kr
	97194061	1010	5679	Hygiene Measures	700kr
	97194062	1011	5680	Draw wisdom teeth	2600kr
	97194063	1012	5681	Teeth Whitening	2000kr
	97194064	1013	5682	Hygiene Measure	450kr
	97194065	1014	5683	Dental Examination	800kr
	97194066	1015	5684	Dental Examination	670kr
	97194067	1007	5685	Dental examination	590kr
	97194068	1005	5686	Tooth replacement	2200kr
	97194069	1015	5687	Tooth replacement	580kr
	97194070	1002	5688	Draw wisdom teeth	1900kr
	97194071	1006	5689	Hygiene Measures	500kr
	97194072	1010	5690	Teeth Whitening	1900kr
*	NULL	NULL	NULL	NULL	NULL

## Payment table

LAPTOP-AEH23G3E\...ntistX - dbo.Room		LAPTOP-AEH23G3E\...X - dbo.Treatment			LAPTOP-AEH23G3E\...stX - dbo.Paym	
	ReceiptNumber	TreatmentID	Paydate	PaymentMethod	PaymentAmount	PaymentTime
	901	97194052	2021-11-27	Visa	2500kr	10:22:33
	902	97194053	2021-11-21	Mastercard	400kr	10:23:34
	903	97194054	2021-12-25	Mastercard	1900kr	10:29:30
	904	97194055	2021-11-25	Visa	500kr	14:27:27
	905	97194056	2021-11-26	Mastercard	680kr	14:30:00
	906	97194057	2021-12-23	Visa	680kr	12:10:02
	907	97194058	2021-11-27	Visa	1900kr	12:04:03
	908	97194059	2021-11-18	Mastercard	550kr	10:22:14
	909	97194060	2021-11-18	Mastercard	550kr	16:05:55
	910	97194061	2021-11-26	Visa	700kr	12:10:23
	911	97194062	2021-11-28	Visa	2600kr	14:25:26
	912	97194063	2021-11-29	Mastercard	2000kr	10:28:30
	913	97194064	2021-11-30	Visa	450kr	12:06:19
	914	97194065	2021-12-15	Mastercard	800kr	16:11:23
	915	97194066	2021-11-22	Mastercard	670kr	10:22:28
	916	97194067	2021-12-22	Mastercard	590kr	12:11:06
	917	97194068	2021-11-28	Visa	2200kr	16:10:23
	918	97194069	2021-11-27	Visa	580kr	10:25:26
	919	97194070	2021-11-24	Visa	1900kr	14:22:12
	920	97194071	2021-11-24	Mastercard	500kr	14:10:23
✍	921	97194072	2021-12-21	Visa	1900kr	14:25:34
*	NULL	NULL	NULL	NULL	NULL	NULL

3) a)

Input:

```
SELECT * FROM Appointment  
SELECT EmployeeID, count(*) as 'TotalPatients' FROM Appointment GROUP BY EmployeeID
```

Employee ID consist of all the Senior Dentists. Which means I have chosen that every Senior Dentist have an appointment in Dentistx clinic. The dentist treated the lowest number of patients is "2815" (Cameron Dennis) and "2828" (Nick Zegan).

Output:

	EmployeeID	TotalPatients
1	2811	2
2	2815	1
3	2817	4
4	2819	2
5	2820	2
6	2822	3
7	2827	4
8	2828	1
9	2831	2

b)

Input:

```
SELECT  
DATEPART(WEEK, AppointmentDate) as Week,  
COUNT(AppointmentID) as Appointment  
FROM Appointment  
WHERE '20210101' <= AppointmentDate  
AND AppointmentDate < '20220101'  
GROUP BY DATEPART(WEEK, AppointmentDate), AppointmentTime  
ORDER BY DATEPART(WEEK, AppointmentDate), AppointmentTime
```

Output:

Week	Appointment
47	1
47	1
48	4
48	2
48	4
49	1
49	1
49	1
49	1
51	1
52	1
52	2
52	1

c)

**Input:**

```
SELECT PatientID, COUNT(PatientID) as 'TotalTreatments'  
FROM Treatment  
GROUP BY PatientID
```

In this output patient '1002' (Jessica Pearson), '1006' (Michael Ross), '1007'(Dana Scott), '1010'(Logan Sanders) and '1015'(Eric Kaldor) have more than one treatment spanned over more than one appointment.

**Output:**

PatientID	TotalTreatments
1001	1
1002	2
1003	1
1004	1
1005	2
1006	2
1007	2
1008	1
1009	1
1010	2
1011	1
1012	1
1013	1
1014	1
1015	2

d)

**Input:**

```
SELECT  
AppointmentID  
FROM Appointment  
WHERE Trainee is NULL
```

It looks like on appointment '72606835', '72606839', '72606840' and '72606841' there where no student trainees assigned.

**Output:**

AppointmentID
72606835
72606839
72606840
72606841

e)

**Input:**

```
SELECT RoomName, Count(RoomName) as 'Total appointments in each room'
FROM Room
GROUP BY RoomName
ORDER BY 'RoomName' asc
```

The room with least number of appointments performed is room A, C, D, E, and F.

**Output:**

RoomName	Total appointments in each room
NULL	0
A	2
B	7
C	2
D	2
E	2
F	2
G	4

f)

**Input:**

```
SELECT *
FROM Patient
WHERE Age > '55';
```

These are the list of patients who are more than 55 years old.

**Output:**

PatientID	FirstName	LastName	Gender	PhoneNumber	Email	Adress	ZipCode	DateOfBirth	Age
1004	Louis	Litt	Male	+45328373	louis.litt@pearsonspecter.com	Gotham Street 6B	7056	1964-04-18	57
1009	Robert	Zane	Male	+45382928	robert.zane@pearsonspecter.com	Sax Avenue 9F	7073	1957-08-28	64
1011	Daniel	Hardman	Male	+45362838	daniel.hardman@pearsonspecter.com	Chicago Avenue 8F	7098	1955-09-04	66
1012	Samantha	Wheeler	Female	+45373922	samantha.wheeler@pearsonspecter.com	York Avenue 4B	7068	1963-05-23	58
1014	Ava	Hassington	Female	+45362819	ava.hassington@hassingtonoil.com	Mary Avenue 4C	7054	1956-10-12	65

g)

**Input:**

```
SELECT EmployeeID, Trainee, PatientID, AppointmentDate, TotalHours
FROM Appointment
ORDER BY EmployeeID, Trainee, PatientID, AppointmentDate, TotalHours
```

I had no possible to calculate this task. The reason for that is one of my entities under attribute 'Room' has an entity called "BookingTimeStartToEnd", I found out later in this task it should be separated in two entities called 'BookingTimeStart' and 'BookingTimeEnd' so that I could calculate. Instead, I have sorted out which of the Senior Dentist and Trainees have spent total hours on all patients across all periods:

**Output:**

EmployeeID	Trainee	PatientID	AppointmentDate	TotalHours
2811	Jonathan Sidwell	1001	2021-11-27	1.5
2811	Nikki Sordel and Monica Eton	1011	2021-11-28	1.5
2815	Harold Gunderson	1003	2021-12-25	1.5
2817	Jack Soloff	1010	2021-12-21	1.5
2817	Jenny Griffith	1007	2021-11-27	1.5
2817	Jonathan Sidwell	1015	2021-11-27	1.5
2817	Nicholas Palermo	1012	2021-11-29	1.5
2819	NULL	1004	2021-11-25	1.5
2819	Edward Darby	1013	2021-11-30	1.5
2820	Nicholas Palermo and Nikki ...	1005	2021-11-28	1.5
2820	Sean Cahill and Henry Ger...	1005	2021-11-26	1.5
2822	Harold Gunderson	1014	2021-12-15	1.5
2822	Jack Soloff	1006	2021-12-23	1.5
2822	Monica Eton	1006	2021-11-24	1.5
2827	NULL	1008	2021-11-18	1.5
2827	Edward Darby	1002	2021-11-21	1.5
2827	Henry Gerard	1002	2021-11-24	1.5
2827	Sean Cahill	1007	2021-12-22	1.5
2828	NULL	1009	2021-11-18	1.5
2831	NULL	1010	2021-11-26	1.5
2831	Jenny Griffith	1015	2021-11-22	1.5

4. The reason I think my database design choices are good in normalized forms is that every table is led to have a good overview of what is happening in a dentist clinic. I have a table for patients, what treatment the patient needs, employees and their working position, and the table which binds them all, which is the appointment table. The appointment table gives the most detail about everything from when the appointment starts, to which patients it is about and the treatment they are having. It also describes which employee and if they are having a hygienist control with a hygienist. The primary key and foreign key is the main reason this database works. I have 'ID' on primary key to define the tables who could not change. The foreign key is also with an 'ID' to find the relation between the tables. This database gives a lot of



information to the receptionist who can easily find the information they need about how the DentistX clinic works in practice.

5. I understand that when several users or applications have an attempt to make a conflict to change the data at the same time. Concurrency methods will minimize the potential conflicts while maintaining the integrity of the data (Oracle, 2021). Concurrency will come, when the activity when multiple users or applications invoke queries at the same time on the same database. Therefore, in databases who consist of a company helping customers, patients etc, it operates data isolation. Which means the database is using the 'views' method to isolate specific columns. For example, through online shopping. A customer has only possible to see their orders. Its not possible to a customer to see the other customers orders because of general data protection regulation. Only the company who owns the online shopping has the possibilities to have the detail about every customer order, name, email, etc.... They are using this method to see the customers orders goes through and to have a clean system to collect all the customers in their database.

6. SQL stands for Structured Query Language which is a relational database. SQL are used when it is **not** possible to define the patterns, and NoSQL is just the opposite – when it is possible to define. In this case, DentistX which is a dental clinic it is possible to solve this in both, but I personally prefer the SQL in this task. The reason is because you have the able to perform flexible queries, create relational queries, and can enforce field constraints (Silberschatz, Korth, and Sudarshan, 2019). NoSQL is anything that is non-relational, and it has many implementations. The queries are less flexible. Patterns are defined, and the primary key is known.

I think it is not a good idea to use NoSQL because this dentist clinic is a small project, it is a small business and its easy to find the relation between the employee and the patient. It is also more flexible comparing to NoSQL. Because of SQL I have the possible to create my own primary key and their relation to each table. That is the reason I think it is a good idea to solve DentistX database in SQL.

**Reference list:**

Oracle, (2021), "What is a relational database(RDBMS)?". Oracle. Unknown date.

<https://www.oracle.com/database/what-is-a-relational-database/?fbclid=IwAR2bheVTMLTfB3aKpsszqMt7fEbbq6pR2Ni6xVYjblv69QpppbqJGgfZhOM> (Accessed: 29. november 2021)

Silberschatz, A., Korth F.H., Sudarshan, S. *Database system concepts* 7<sup>th</sup>.edition (2019)