CIS 3400: A Database for Smiles Inc

Group 9 Team Members:

Alexander Batch - alexander.batch@baruchmail.cuny.edu

Sharia Hoque - sharia.hoque@baruchmail.cuny.edu

Guanqing Lin - guanqing.lin@baruchmail.cuny.edu

Shantoye Reid - shantoye.reid@baruchmail.cuny.edu

Shuai Yang - shuai.yang@baruchmail.cuny.edu

A Database for Smiles Inc

Smiles Inc needs a database that will effectively organize dental office employees, customers, procedures, equipment, supplies, and other services provided as well as specific details aligned with each listed entity. Procedures will be connected to the employees responsible for completing them along with other necessary details.

This database will primarily minimize the need for paper charts and other physical documents as they will be easily accessible and organized reducing customer wait times and increasing employee productivity. By connecting procedures, equipment, customers and employees, office errors will be reduced, and employee obligations will be clearer.

Entities

Employees – ID, Name, Contact Info, Job Title, Salary, Years Employed

Customers – ID, Name, Contact Info, Insurance, Medical History, Procedure History

Equipment – ID, Info (type/name/age/cost), Supplier, Procedure

Procedures – ID, Customer ID, Employee ID, Procedure Type, Date (info), Cost, Equipment Used

Member Roles

Alexander Batch – Systems Analyst

Sharia Hoque – Application Developer

Guanqing Lin – Documentation Writer

Shantoye Reid – Scrum Master

Shuai Yang – Product Owner

Relational Model

Insurance (Insurance_ID (PK), Company_Name, Company_Phone_Num, Plan_Type, Expiration_Date)

Patients (Patients_ID (PK), Name_First, Name_Last, Street_Address, Apt_Sutie_Num, City, State, Zip_Code, Phone_Num, Email_Address, Gender, DOB, Race, Insurance_ID (FK))

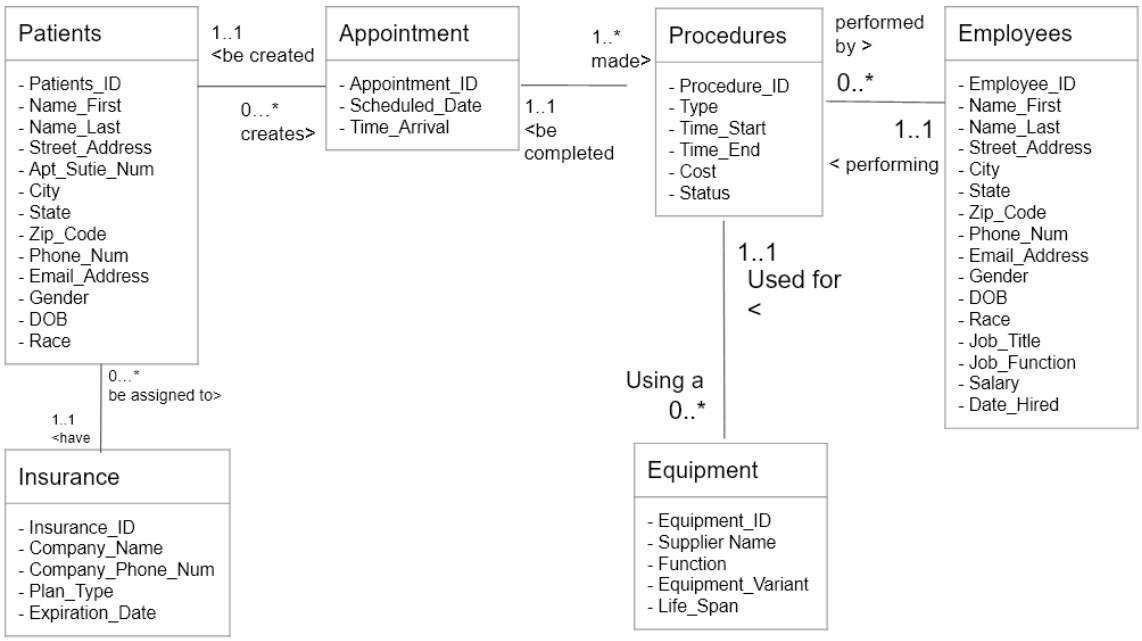
Appointment (Appointment_ID (PK), Scheduled_Date, Time_Arrival , Patients_ID (FK))

Equipment (Equipment_ID (PK), Supplier Name, Function, Equipment_Variant, Life_Span, Procedure_ID (FK))

Procedures (Procedure_ID (PK), Type,Time_Start, Time_End, Cost, Status, Employee_ID (FK))

Employees (Employee_ID (PK), Name_First, Name_Last, Street_Address, Apt_Sutie_Num, City, State, Zip_Code, Phone_Num, Email_Address, Gender, DOB, Race, Job_Title, Job_Function, Salary, Date_Hired)

Entity Relationship Model



Smile Inc Dental Clinic

Normalization

Insurance Relation

insurance(Insurance_ID (key), Company_Name, Company_Phone_Num, Plan_Type, Expiration_Date)

Key: Insurance_ID

FD1: Insurance_ID → Company_Name, Company_Phone_Num, Plan_Type, Expiration_Date

FD2 : Company_Name→ Company_Phone_Num, Plan_Type

1NF: Yes. Meets the definition of a relation | 2NF: Yes. No partial Key dependencies | 3NF: No. There is transitive dependency

Split insurance relation into two new relations.

insurance_Company(Company_Name, Company_Phone_Num, Plan_Type)

Key: Company_Name

FD1: Company_Name→ Company_Phone_Num, Plan_Type

1NF: Yes. Meets the definition of a relation | 2NF: Yes. No partial Key dependencies | 3NF: Yes. There is no transitive dependency

insurance_final(Insurance_ID(key), Company_Name (FK), Expiration_Date)

Key: Insurance_ID

FD1: Insurance_ID → Company_Name, Expiration_Date

1NF: Yes. Meets the definition of a relation | 2NF: Yes. No partial Key dependencies | 3NF: Yes. There is no transitive dependency

Patients Relation

Patients (Patients_ID (PK), Name_First, Name_Last, Street_Address, Apt_Sutie_Num, City, State, Zip_Code, Phone_Num, Email_Address, Gender, DOB, Race, Insurance_ID (FK))

Key: Patients_ID

FD1: Patients_ID(key) ⟶ Name_First, Name_Last, Street_Address, Apt_Sutie_Num, City, State, Zip_Code, Phone_Num, Email_Address, Gender, DOB, Race, Insurance_ID

FD2: Zip_Code -> City, State

1NF: Yes. Meets the definition of a relation| 2NF: 2NF is done because all of the non-key attributes are dependent on all of the key | 3NF: transitive dependency exists

Split patients relation into two new relations

zip_codes(Zip_Code, City, State)

Key: Zip_code

FD1: Zip_Code ⟶ City, State

1NF: Yes, It was split from a relation.| 2NF: yes. There's no Partial Dependency. |3NF: yes.. There are no transitive dependencies

Patients_final(Patients_ID (PK), Name_First, Name_Last, Street_Address, Apt_Sutie_Num, Zip_Code(FK), Phone_Num, Email_Address, Gender, DOB, Race, Insurance_ID (FK))

1NF: Yes. Meets the definition of a relation | 2NF: Yes. No partial Key dependencies | 3NF: Yes. There is no transitive dependency

Appointment Relation

Appointment(Appointment_ID (PK), Scheduled_Date, Time_Arrival , Patients_ID (FK))

Key: Appointment_ID

FD1: Appointment_ID ⟶ Scheduled_Date, Time_Arrival , Patients_ID (FK)

1NF: Yes, It was split from a relation.

2NF: Yes, All of the non-key attributes are dependent upon all of the key.

3NF: Yes, No Transitive dependencies.

Equipment Relation

Equipment (Equipment_ID (PK), Supplier Name, Function, Equipment_Variant, Life_Span, Procedure_ID (FK))

Key: Equipment_ID, Supplier_Name

FD1: Equipment_ID, Supplier_Name → Function, Equipment_Variant, Life_Span, Procedure_ID

1NF: Yes, It was split from a relation.

2NF: Yes, All of the non-key attributes are dependent upon all of the key.

3NF: Yes, No Transitive dependencies.

Procedures Relation

Procedures(Procedure_ID (PK), Type,Time_Start, Time_End, Cost, Status, Employee_ID (FK))

Key: Procedure_ID

FD1: Procedure_ID → Type, Time_Start, Time_End, Cost, Status, Employee_ID

FD2: Type → Cost

1NF: Yes, It was split from a relation. |2NF: Yes, All of the non-key attributes are dependent upon all of the key. | 3NF: Yes, Transitive dependencies.

Procedures_type(Type , Cost)

Key: Type

FD1: Type → Cost

1NF: Yes, It was split from a relation.| 2NF: yes. , there's no Partial Dependency. |3NF: yes.. There are no transitive dependencies

Procedures_final(Procedure_ID (PK), Type (FK) ,Time_Start, Time_End, Status, Employee_ID (FK))

Key: Procedure_ID

FD1: Procedure_ID → Type, Time_Start, Time_End, Status, Employee_ID

1NF: Yes, It was split from a relation.| 2NF: yes. There's no Partial Dependency. |3NF: yes There are no transitive dependencies

Employees Relation

Employees (Employee_ID (PK), Name_First, Name_Last, Street_Address, Apt_Sutie_Num, City, State, Zip_Code, Phone_Num, Email_Address, Gender, DOB, Race, Job_Title, Job_Function, Salary, Date_Hired)

Key: Employee_ID

FD1: Employee_ID → Name_First, Name_Last, Street_Address, Apt_Sutie_Num, City, State, Zip_Code, Phone_Num, Email_Address, Gender, DOB, Race, Job_Title, Job_Function, Salary,

FD2: Zip_Code → City, State

1NF: Yes, It was split from a relation.| 2NF: yes. , there's no Partial Dependency. |3NF: No. There are transitive dependencies

Split employees relation into two new relations.

zip_codes(Zip_Code, City, State)

Key: Zip_code

FD1: Zip_Code → City, State

1NF: Yes, It was split from a relation.| 2NF: yes. , there's no Partial Dependency. |3NF: yes.. There are no transitive dependencies

Employees_final (Employee_ID (PK), Name_First, Name_Last, Street_Address, Apt_Sutie_Num, Zip_Code (FK), Phone_Num, Email_Address, Gender, DOB, Race, Job_Title, Job_Function, Salary, Date_Hired)

1NF: Yes, It was split from a relation.| 2NF: yes. There's no Partial Dependency. |3NF: yes.. There are no transitive dependencies.

Final Normalization

insurance_final (Insurance_ID(key), Company_Name (FK), Expiration_Date)

insurance_Company (Company_Name, Company_Phone_Num, Plan_Type)

zip_codes (Zip_Code, City, State)

Patients_final (Patients_ID (PK), Name_First, Name_Last, Street_Address, Apt_Sutie_Num, Zip_Code(FK), Phone_Num, Email_Address, Gender, DOB, Race, Insurance_ID (FK))

Appointment (Appointment_ID (PK), Scheduled_Date, Time_Arrival , Patients_ID (FK))

Equipment (Equipment_ID (PK), Supplier Name, Function, Equipment_Variant, Life_Span, Procedure_ID (FK))

Procedures_type (Type , Cost)

Procedures_final (Procedure_ID (PK), Type (FK) ,Time_Start, Time_End, Status, Employee_ID (FK), Appointment_ID (FK), Employee_ID (FK))

Employees_final (Employee_ID (PK), Name_First, Name_Last, Street_Address, Apt_Sutie_Num, Zip_Code (FK), Phone_Num, Email_Address, Gender, DOB, Race, Job_Title, Job_Function, Salary, Date_Hired)

SQL / Creating Tables

Insurance:

```
CREATE TABLE Insurance
(    Insurance_ID          VARCHAR(40) NOT NULL,
   Company_Name      VARCHAR(40),
   Company_Phone_Num       VARCHAR(40),
   Plan_Type              VARCHAR(40),
   Expiration_Date       DATE,
   CONSTRAINT pk_insurance
          PRIMARY KEY (Insurance_ID)
)
```

Procedure:

```
CREATE TABLE Procedures
(   Procedure_ID          VARCHAR(40) NOT NULL,
  Type  VARCHAR(40),
 Date_Time_Start       DATE,
 Date_Time_End        DATE,
  Cost              NUMBER,
   Status          VARCHAR(40),
Employee_ID VARCHAR(20),
Appointment_ID VARCHAR(40) NOT NULL,
   CONSTRAINT pk_procedure
          PRIMARY KEY (Procedure_ID)
)
```

Equipment:

```
CREATE TABLE Equipment
(   Equipment_ID        VARCHAR(40) NOT NULL,
  Supplier_Name        VARCHAR(40),
  Function       VARCHAR(40),
  Equipment_Variant             VARCHAR(40),
  Life_Span      VARCHAR(40),
Procedure_ID  VARCHAR(40),
   CONSTRAINT pk_equipment
         PRIMARY KEY (Equipment_ID)
)
```

Zip Code:

Appointment:

```
CREATE TABLE Appointment
(    Appointment_ID    VARCHAR(40) NOT NULL,
   Scheduled_Date      DATE,
   Time_Arrival          VARCHAR(40),
   Patients_ID   VARCHAR(40)  NOT NULL,
   CONSTRAINT          pk_appointment
         PRIMARY KEY (Appointment_ID)
)
```

Patients_final:

```
CREATE TABLE Patients_final
(    Patients_ID VARCHAR(40) NOT NULL,
 Name_First     VARCHAR(40),
Name_Last      VARCHAR(40),
Street_Address          VARCHAR(40),
Zip_Code        VARCHAR(12) NOT NULL,
 Phone_Num    VARCHAR(40),
 Email_Address          VARCHAR(40),
 Gender                 VARCHAR(40),
DOB             DATE,
 Race            VARCHAR(40),
Insurance_ID   VARCHAR(40),
CONSTRAINT pk_patient
          PRIMARY KEY (Patients_ID)
)
```

Employee:

```
CREATE TABLE Employee
(   Employee_ID     VARCHAR(20),
Name_First     VARCHAR(35),
Name_Last     VARCHAR(25),
Street_Address       VARCHAR(45),
Zip_Code        VARCHAR(12),
City            VARCHAR(25),
State           VARCHAR(25),
Phone_num    VARCHAR(25),
```

```
Email_Address   VARCHAR(25),
Gender          VARCHAR(25),
DOB             DATE,
Race            VARCHAR(25),
Job_Title       VARCHAR(25),
Job_Function VARCHAR(25),
Date_Hired     DATE,
Salary          NUMBER,
    CONSTRAINT pk_employee
     PRIMARY KEY (Employee_ID)
)
(
 Zip_Code   VARCHAR(12) NOT NULL,
   city       VARCHAR(36),
   state      VARCHAR(20),
   CONSTRAINT pk_zipcode
        PRIMARY KEY (Zip_Code)
)
```

SQL / Adding Foreign Keys

```
ALTER TABLE Patients_final
ADD CONSTRAINT fk_patients_zipcode
    FOREIGN KEY (Zip_Code)
      REFERENCES zip_codes (Zip_Code)

ALTER TABLE Patients_final
ADD CONSTRAINT fk_patients_insurance
    FOREIGN KEY (Insurance_ID)
      REFERENCES Insurance (Insurance_ID)

ALTER TABLE Appointment
ADD CONSTRAINT fk_appointment_patient
   FOREIGN KEY (Patients_ID)
      REFERENCES Patients_final (Patients_ID)

ALTER TABLE Procedures
ADD CONSTRAINT fk_procedures_appointment
   FOREIGN KEY (Appointment_ID)
      REFERENCES Appointment (Appointment_ID)

ALTER TABLE Procedures
ADD CONSTRAINT fk_employee_procedures
   FOREIGN KEY (Employee_ID)
```

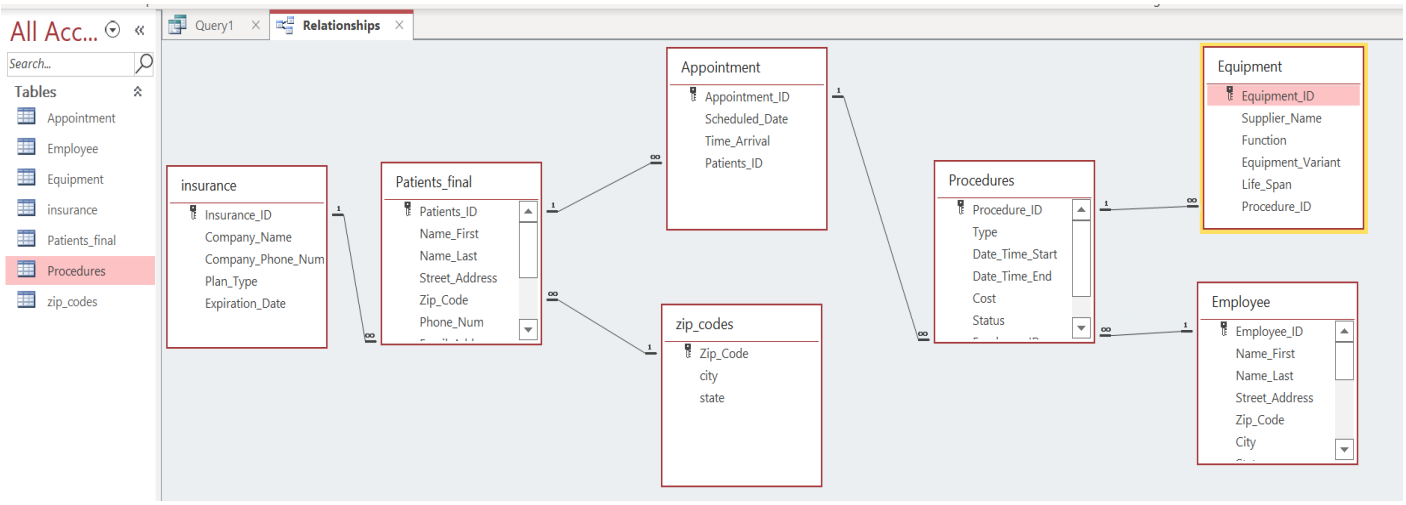
REFERENCES Employee (Employee_ID)

ALTER TABLE Equipment

ADD CONSTRAINT fk_procedure_equipment

FOREIGN KEY (Procedure_ID)

REFERENCES Procedures(Procedure_ID)



SQL / Inserting value into Tables

dtformat: yyyy-mm-dd hh:mm:ss

TABLE zip_codes

INSERT INTO zip_codes VALUES ('33400', 'Manalapan', 'NJ') ;

INSERT INTO zip_codes VALUES ('11430', 'Jamaica', 'NY') ;

INSERT INTO zip_codes VALUES ('11011', Manhattan', 'NY') ;

INSERT INTO zip_codes VALUES ('11237', 'Brooklyn', 'NY') ;

INSERT INTO zip_codes VALUES ('07097', 'Jersey City', 'NJ') ;

TABLE insurance

INSERT INTO insurance VALUES ('ABC01231', 'Health First', '844-488-1486', 'Medicaid', #2026/03/24#) ;

INSERT INTO insurance VALUES ('c54003', 'MetroPlus', '845-732-2096', 'Medicaid', #2024/3/20#) ;

INSERT INTO insurance VALUES ('ZXy5094', 'Health First', '095-732-5687', 'Health Plan', #2023/07/11#) ;

INSERT INTO insurance VALUES ('DCB90876', 'Metro-Plus', '567-850-3332', 'Health Plan', #2023/05/17#) ;

INSERT INTO insurance VALUES ('VDCB9876', 'Metro-Plus', '986-670-3323', 'Health Plan', #2023/05/20#) ;

TABLE Appointment

INSERT INTO appointment VALUES ('ik02kuus', #2019/04/16#, #08:45:00#, 'MRS00001') ;

INSERT INTO appointment VALUES ('t3bewi00', #2019/04/06#, #09:30:00#, 'MRS00002') ;

INSERT INTO appointment VALUES ('ik12kuns', #2019/04/16#, #10:45:00#, 'MRS00003') ;

INSERT INTO appointment VALUES ('ik22kubs', #2019/04/16#, #11:45:00#, 'MRS00004') ;

TABLE Procedure

INSERT INTO procedures VALUES ('BKK329FJ', 'Cleaning', #2019/03/22 09:15:01#, #2019/03/22 09:35:00#, 180, 'COMPLETE', 'ABJ00101' , 'ik02kuus');

INSERT INTO procedures VALUES('BKK328FJ', 'Braces_Treatment', #2019/05/12 10:17:00#, #2019/05/12 10:45:00#, 180, 'COMPLETE', 'ABJ00102' , 't3bewi00');

INSERT INTO procedures VALUES('BKK327FJ', 'Invisalign_Treatment', #2019/08/09 14:17:00#, #2019/08/09 14:35:00#, 180, 'COMPLETE', 'ABJ00103' , 't3bewi00');

INSERT INTO procedures VALUES('BKK326FJ', 'Phase_1_Interceptive_Treatment', #2019/11/12 15:17:00#, #2019/11/12 15:35:00#, 180, 'COMPLETE', 'ABJ00104' , 'ik12kuns');

INSERT INTO procedures VALUES('BKK325FJ', 'Orthodontic_Retainers', #2019/12/12 16:17:00#, #2019/12/12 16:35:00#, 180, 'COMPLETE', 'ABJ00103' , 'ik12kuns');

TABLE Equipment

INSERT INTO equipment VALUES ('CT393J91','oral-b', 'cleaner toothbrush', 'Sonic Plus Two', '2800', 'BKK325FJ') ;

INSERT INTO equipment VALUES ('CT393J92','invisalign braces', 'braces','Invisa_This', '2800','BKK327FJ') ;

INSERT INTO equipment VALUES ('CT393J93', 'metal braces', 'braces', 'ugly_braces', '2500', 'BKK328FJ') ;

INSERT INTO equipment VALUES ('CT393J94', 'X-Ray Dental Manikin', 'xray', 'hes_not_alive', '2648','BKK326FJ') ;

INSERT INTO equipment VALUES ('CT393J95', 'Clear Retainers', 'retainer', 'see_what', '400', 'BKK327FJ') ;

INSERT INTO equipment VALUES ('CT393J96', 'Vivera Retainers', 'retainer', 'its_not_real_cheese', '420, 'BKK325FJ') ;

INSERT INTO equipment VALUES ('CT393J96', 'Hawley Retainers', 'retainer', 'harley_road_trip', '390, 'BKK325FJ') ;

TABLE Employee

INSERT INTO employee VALUES ('ABJ00101', 'Andrew', 'Jeremy', '833 West Treehaw Avenue', '07097', 'Elsewhere', 'New Jersey', '228-8002-2875', 'abj01_smiles@gmail.com', 'male', #12/04/1992#, 'white', 'dentist', 'cleaner', #06/22/2018#, 128000);

INSERT INTO employee VALUES ('ABJ00102', 'Jenny', 'Li', '4523 6th Ave', '11220', 'Brooklyn', 'New York', '228-8002-2976', 'abj02_smiles@gmail.com', 'Female', #12/04/1990#, 'Asian', 'dentist', 'cleaner', #09/22/2018#, 128001);

INSERT INTO employee VALUES ('ABJ00103', 'Victor', 'LUI', '255A 19th St', '11215', 'Brooklyn', 'New York', '228-8002-3758', 'abj03_smiles@gmail.com', 'male', #01/04/1990#, 'Black', 'dentist', 'cleaner', #08/22/2018#, 128002);

INSERT INTO employee VALUES ('ABJ00104', 'Kimbery', 'Bui', '305 6th Ave', '11215', 'Brooklyn', 'New York', '228-8002-0486', 'abj04_smiles@gmail.com', 'male', #01/23/1990#, 'Black', 'dentist', 'cleaner', #03/22/2018#, 128003);

TABLE Patients_Final

INSERT INTO Patients_final VALUES ('MRS00001', 'Manu', 'Sultan', '170 st', '11430', '657-978-0499' , 'Sulatan.Manu@gmail.com', 'Male', #05/02/1998# , 'Arabian', 'ABC01231') ;

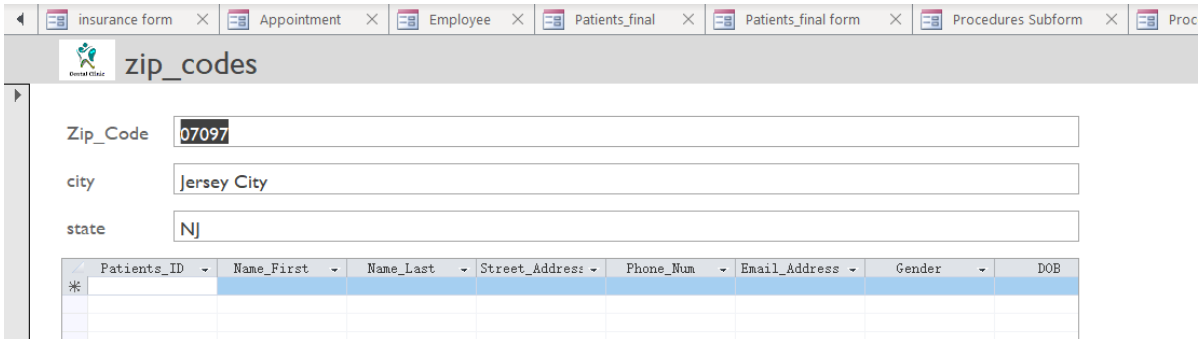
INSERT INTO Patients_final VALUES ('MRS00002', 'Leilani', 'Alaina', '180 st', '11430', '657-978-9575' , 'Leilani.Alaina@gmail.com', 'Female', #02/02/1988# , 'Hawaiian', 'c54003');

INSERT INTO Patients_final VALUES ('MRS00003', 'Williams', 'Johnson', '190 st', '11430', '657-978-0985' , 'Williams.Johnson@gmail.com', 'Male', #02/02/2000# , 'Black', 'ABC01233') ;

INSERT INTO Patients_final VALUES ('MRS00004', 'Ming', 'Li', '160 st', '11430', '657-978-2375' , 'Ming.Li@gmail.com', 'Female', #02/02/1973# , 'Asian', 'ZXy5094') ;

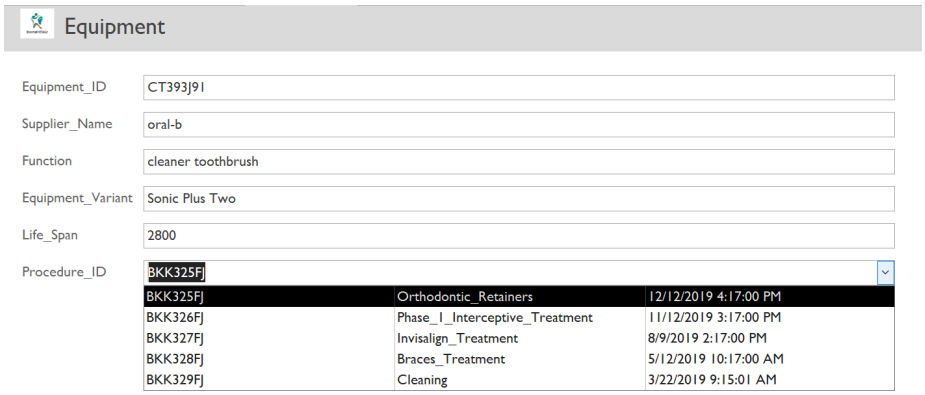
Database Application

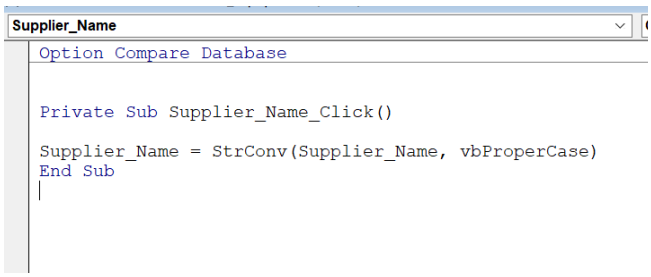
Forms



Holds info about all zip-codes and cities that are fed into multiple forms later on.

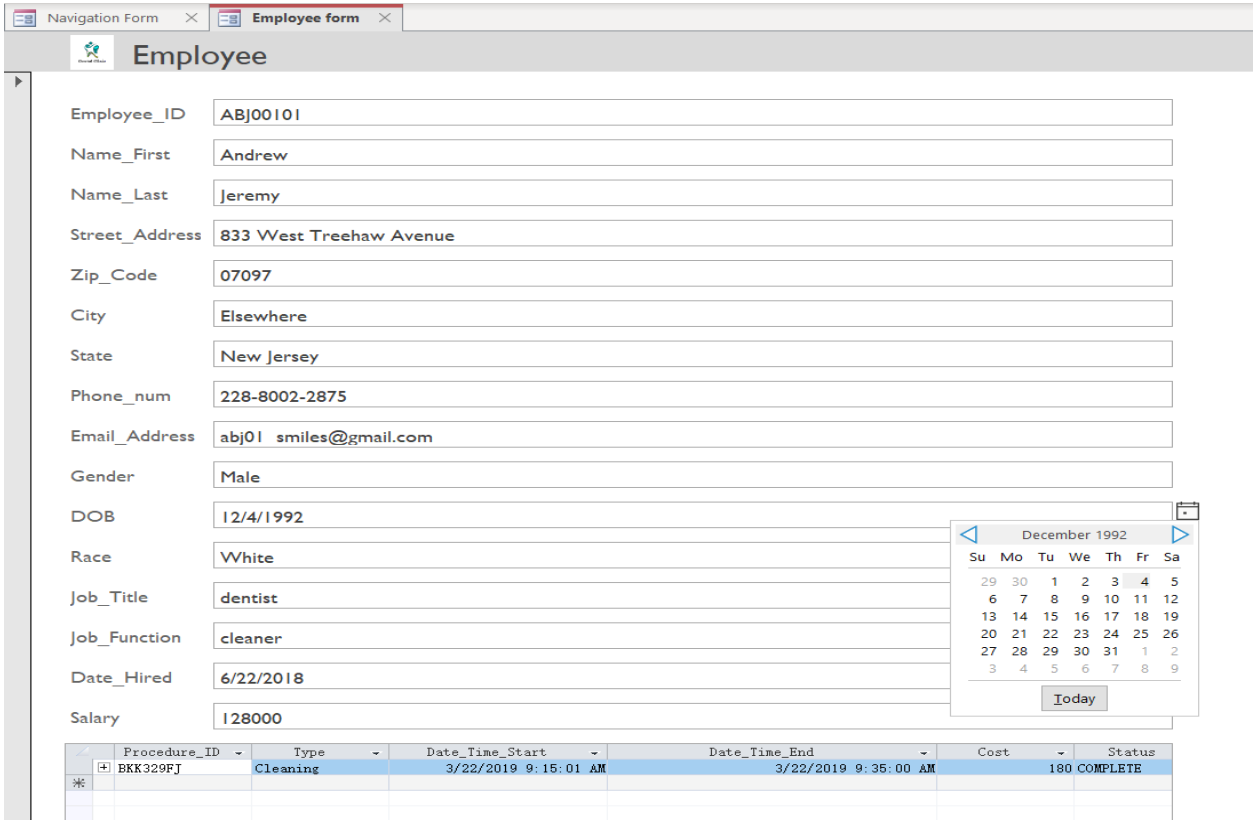
Equipment:

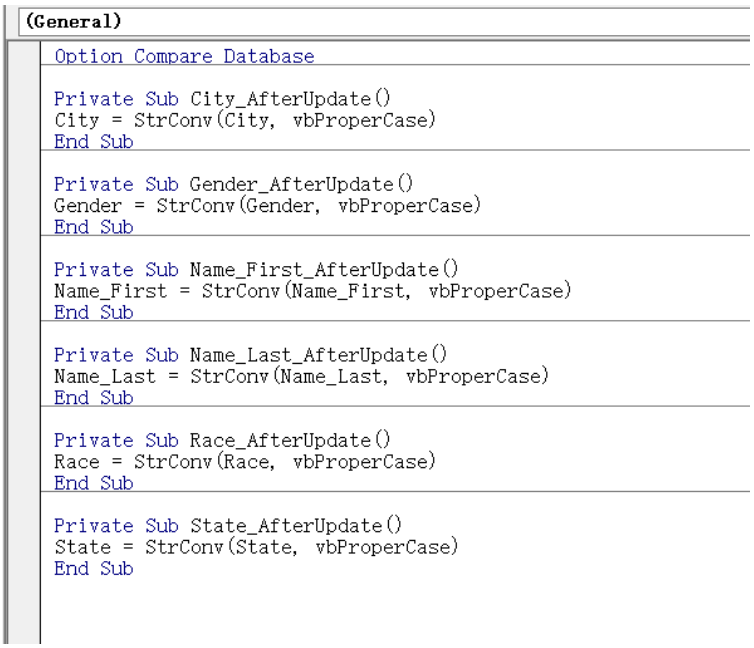




Equipment form is used to lookup various types of equipment, its life-span and the supplier for equipment. It also can identify for which procedure the equipment was used.

Smile Inc Dental Clinic

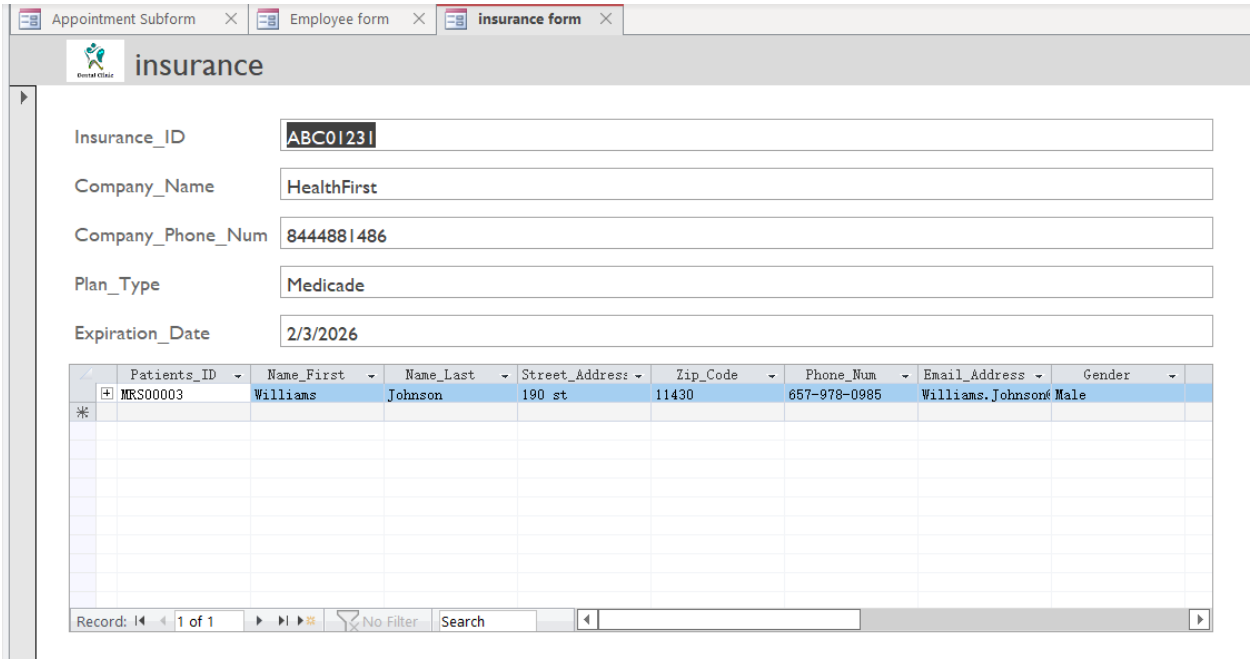




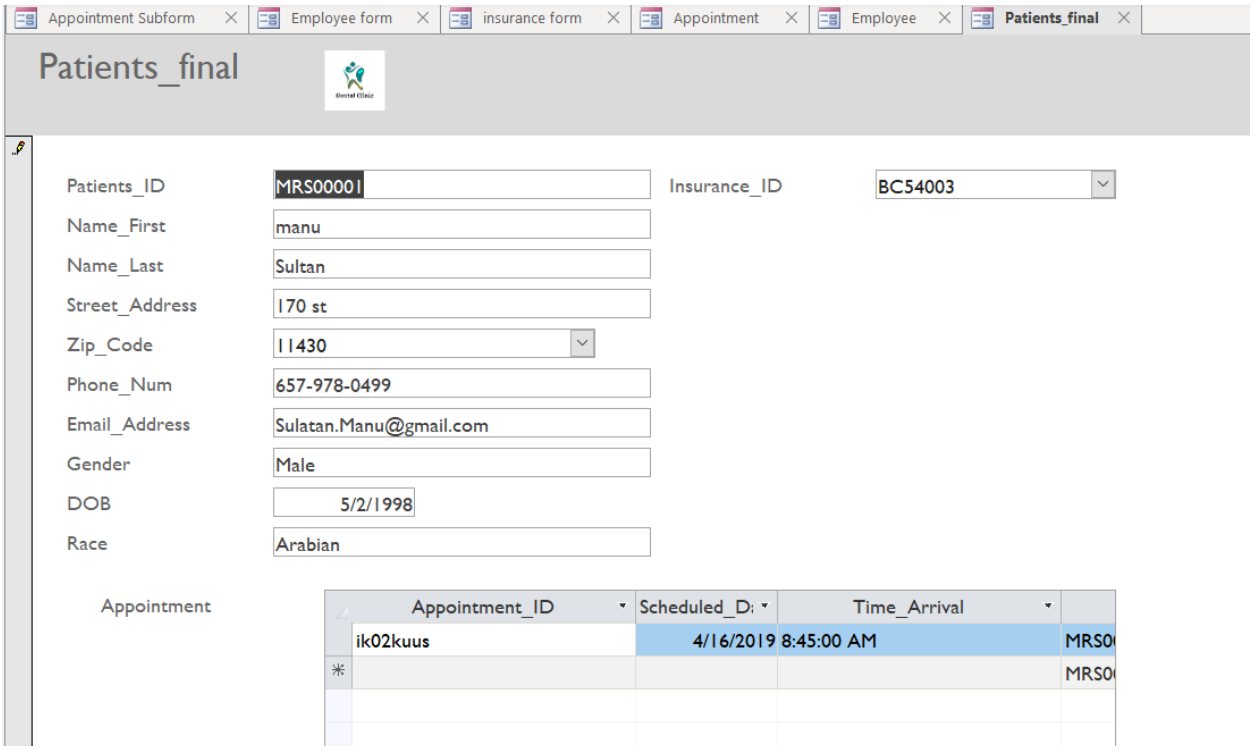
```
(General)
Option Compare Database

Private Sub City_AfterUpdate()
City = StrConv(City, vbProperCase)
End Sub

Private Sub Gender_AfterUpdate()
Gender = StrConv(Gender, vbProperCase)
End Sub

Private Sub Name_First_AfterUpdate()
Name_First = StrConv(Name_First, vbProperCase)
End Sub

Private Sub Name_Last_AfterUpdate()
Name_Last = StrConv(Name_Last, vbProperCase)
End Sub

Private Sub Race_AfterUpdate()
Race = StrConv(Race, vbProperCase)
End Sub

Private Sub State_AfterUpdate()
State = StrConv(State, vbProperCase)
End Sub
```

The employee form allows someone from the business to better keep track off all needed info about who works there, their contact information, and their position in the company, and other

work related information. VBA code makes sure the first letter of City, State, First name, Last name are capital.



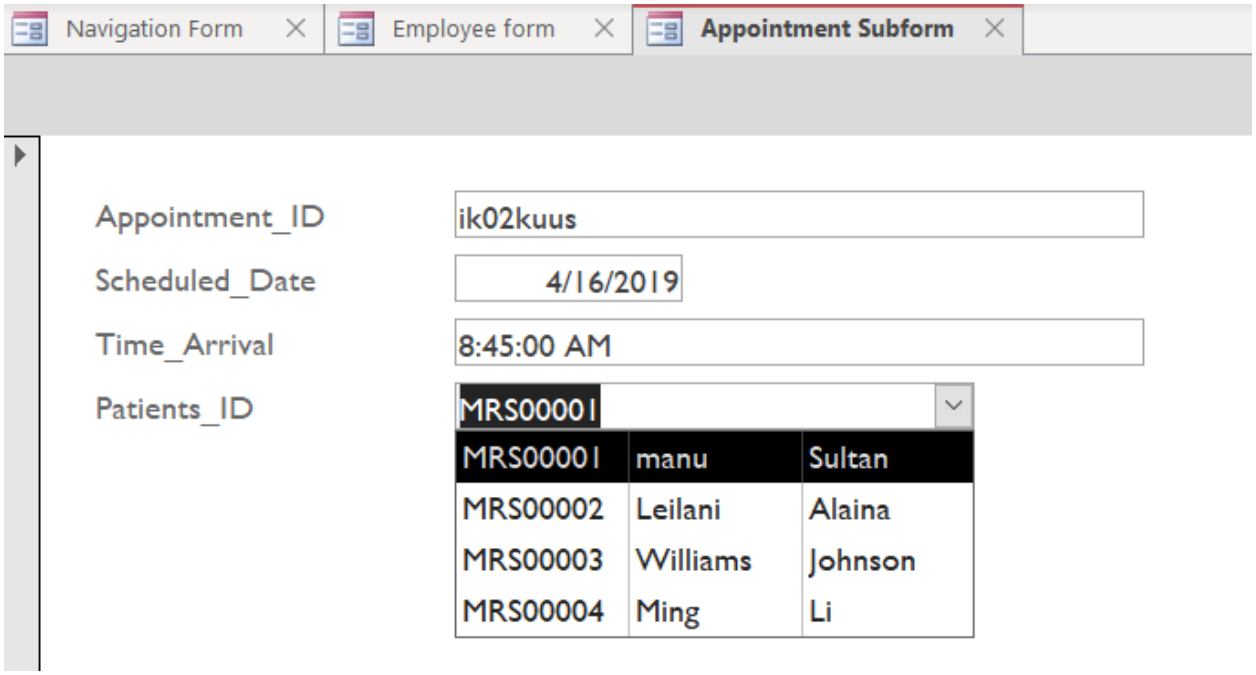
Keeping track of insurance is very important to know who to contact about possible billings.



Smile Inc Dental Clinic

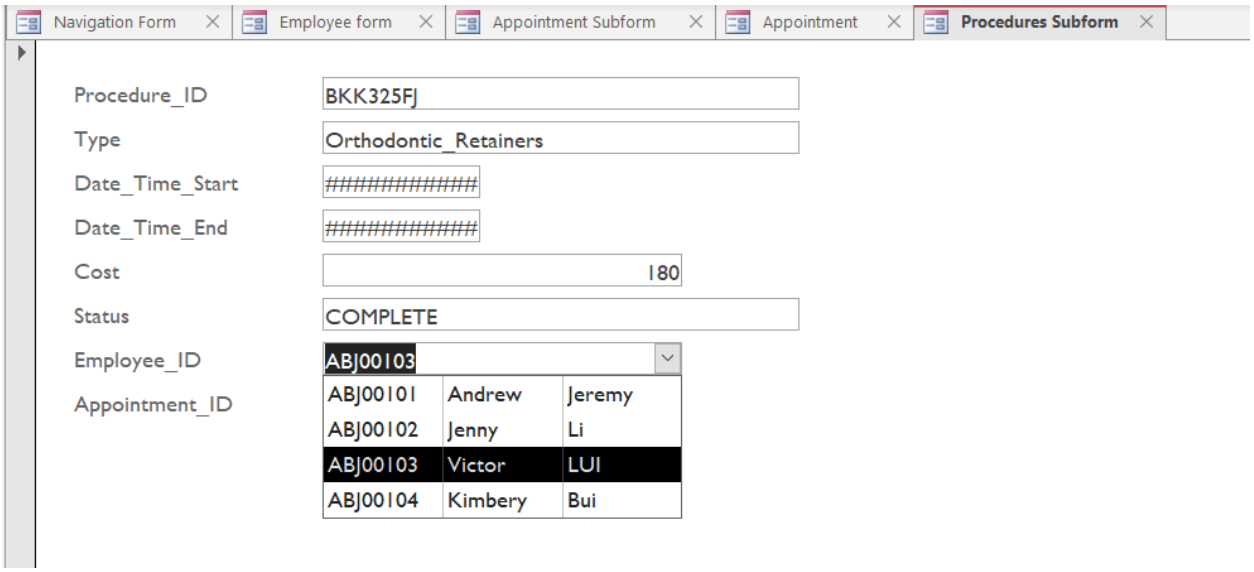
```
(General)
Option Compare Database

Private Sub City_AfterUpdate()
City = StrConv(City, vbProperCase)
End Sub

Private Sub Name_First_AfterUpdate()
Name_First = StrConv(Name_First, vbProperCase)
End Sub

Private Sub Name_Last_AfterUpdate()
Name_Last = StrConv(Name_Last, vbProperCase)
End Sub

Private Sub State_AfterUpdate()
State = StrConv(State, vbProperCase)
End Sub
```

A form listing all patients that have either had services done or have upcoming appointments set. Includes contact information and any insurance info that's needed. The VBA code is to make sure the customer's name and state info is capitalized properly.



```
(General)

Option Compare Database

Private Sub Appointment_ID_AfterUpdate()
Appointment_ID = StrConv(Appointment_ID, vbProperCase)
End Sub
```

This form helps setup when upcoming appointments are for scheduling purposes. It connects to the patients table so double entry isn't needed. The VBA code makes sure the appointment ID is all capitals.

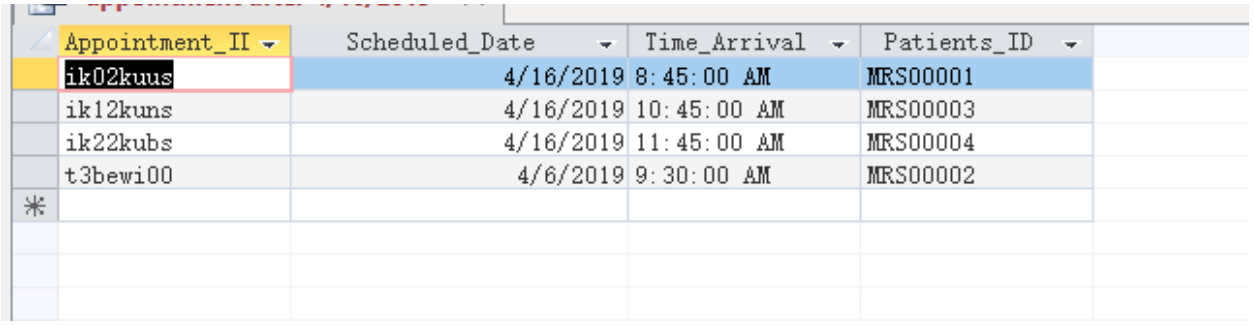


This last main form helps keep track of what procedures are needed to be done, what equipment used, when it's to be done, and by whom. Arguably the most important form because it helps with process management.

Queries

1. Appointment after 4/15/2019

```
SELECT*

FROM Appointment

WHERE Scheduled_Date >4/15/2019;
```



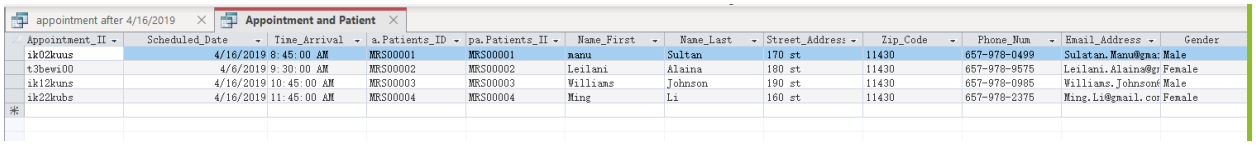
Appointment_ID	Scheduled_Date	Time_Arrival	Patients_ID
ik02kuus	4/16/2019	8:45:00 AM	MRS00001
ik12kuns	4/16/2019	10:45:00 AM	MRS00003
ik22kubs	4/16/2019	11:45:00 AM	MRS00004
t3bewi00	4/6/2019	9:30:00 AM	MRS00002

This query shows appointment dates which happened after 4-15-2019.

2. A combination of appointments and patients

SQL:

```
SELECT *

FROM Appointment AS a

INNER JOIN Patients_final AS pa

ON a.Patients_ID = pa.Patients_ID;
```



appointment after 4/16/2019 | Appointment and Patient

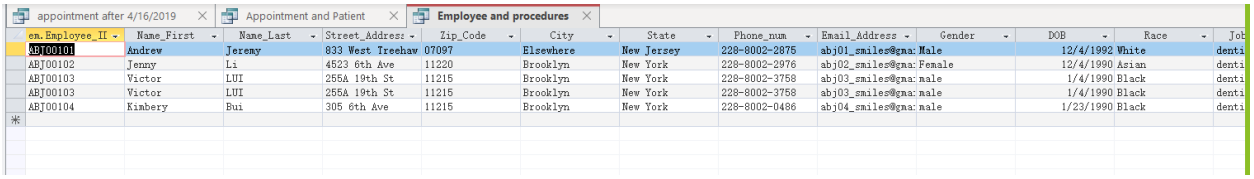
Appointment_ID	Scheduled_Date	Time_Arrival	a.Patients_ID	pa.Patients_ID	Name_First	Name_Last	Street_Address	Zip_Code	Phone_Num	Email_Address	Gender
ik02kuus	4/16/2019	8:45:00 AM	MRS00001	MRS00001	manu	Sultan	170 st	11430	657-978-0499	Sulatan.Manu@gma	Male
t3bewi00	4/6/2019	9:30:00 AM	MRS00002	MRS00002	Leilani	Alaina	180 st	11430	657-978-9575	Leilani.Alaina@g	Female
ik12kuns	4/16/2019	10:45:00 AM	MRS00003	MRS00003	Williams	Johnson	190 st	11430	657-978-0985	Williams.Johnson	Male
ik22kubs	4/16/2019	11:45:00 AM	MRS00004	MRS00004	Ming	Li	160 st	11430	657-978-2375	Ming.Li@gmail.co	Female

A query was made to show the combination of appointments and patients to see all the information about upcoming appointments along with detailed info about the patients.

3. A combination of Employee and Procedure

SQL:

```
SELECT *

FROM Employee AS em

INNER JOIN Procedures AS pr

ON em.Employee_ID = pr.Employee_ID;
```



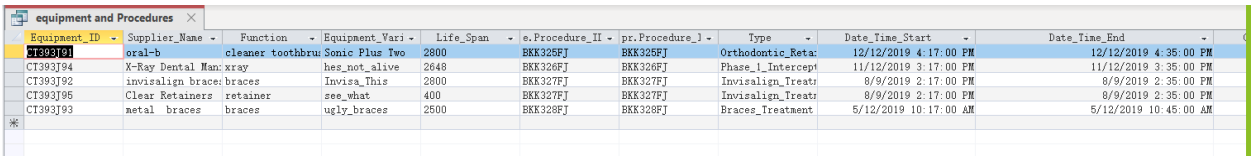
em.Employee_ID	Name_First	Name_Last	Street_Address	Zip_Code	City	State	Phone_num	Email_Address	Gender	DOB	Race	Job
ABJ00101	Andrew	Jeremy	833 West Treehaw	07097	Elsewhere	New Jersey	228-8002-2875	abj01_smiles@gma	Male	12/4/1992	White	denti
ABJ00102	Jenny	Li	4523 6th Ave	11220	Brooklyn	New York	228-8002-2976	abj02_smiles@gma	Female	12/4/1990	Asian	denti
ABJ00103	Victor	LUI	255A 19th St	11215	Brooklyn	New York	228-8002-3758	abj03_smiles@gma	male	1/4/1990	Black	denti
ABJ00103	Victor	LUI	255A 19th St	11215	Brooklyn	New York	228-8002-3758	abj03_smiles@gma	male	1/4/1990	Black	denti
ABJ00104	Kimbery	Bui	305 6th Ave	11215	Brooklyn	New York	228-8002-0486	abj04_smiles@gma	male	1/23/1990	Black	denti

Being shown is a query to know which employees are working on what procedures. This way if there's schedule conflict or a gap, it can be very quickly solved by looking at this newly combined table.

4. A combination of Equipment and Procedure

SQL:

```
SELECT *

FROM Equipment AS e

INNER JOIN Procedures AS pr

 ON e.Procedure_ID = pr.Procedure_ID;
```



Equipment_ID	Supplier_Name	Function	Equipment_Vari	Life_Span	e.Procedure_ID	pr.Procedure_I	Type	Date_Time_Start	Date_Time_End
CT393J91	oral-b	cleaner toothbru	Sonic Plus Two	2800	BKK325FJ	BKK325FJ	Orthodontic_Reta	12/12/2019 4:17:00 PM	12/12/2019 4:35:00 PM
CT393J94	X-Ray Dental Man	xray	hes_not_alive	2648	BKK326FJ	BKK326FJ	Phase_1_Intercept	11/12/2019 3:17:00 PM	11/12/2019 3:35:00 PM
CT393J92	invisalign brace	braces	Invisa_This	2800	BKK327FJ	BKK327FJ	Invisalign_Treat	8/9/2019 2:17:00 PM	8/9/2019 2:35:00 PM
CT393J95	Clear Retainers	retainer	see_what	400	BKK327FJ	BKK327FJ	Invisalign_Treat	8/9/2019 2:17:00 PM	8/9/2019 2:35:00 PM
CT393J93	metal braces	braces	ugly_braces	2500	BKK328FJ	BKK328FJ	Braces_Treatment	5/12/2019 10:17:00 AM	5/12/2019 10:45:00 AM

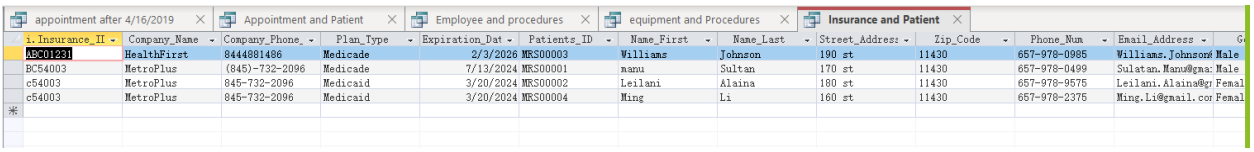
Filtered above joins the equipment table with procedures so an employee can know what tools are needed to get a job done and complete a procedure.

5. A combination of Insurance and Patient

SQL:

```
SELECT *
```

```
FROM insurance AS i

INNER JOIN Patients_final AS pa

ON i.Insurance_ID = pa.Insurance_ID;
```



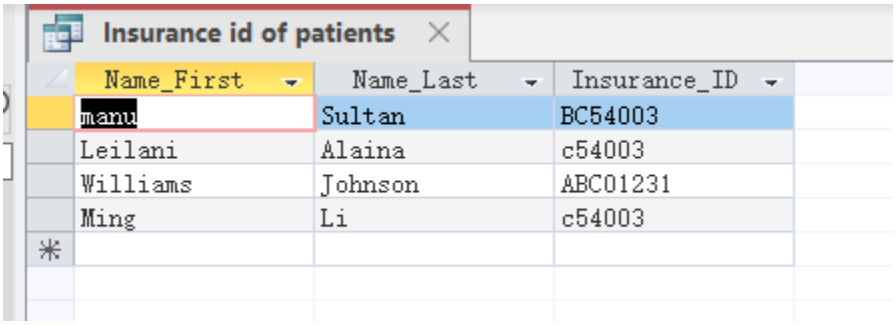
i.Insurance_ID	Company_Name	Company_Phone_	Plan_Type	Expiration_Dat	Patients_ID	Name_First	Name_Last	Street_Address	Zip_Code	Phone_Num	Email_Address	G
ABC01231	HealthFirst	9444881486	Medicade	2/3/2026	MRS00003	Williams	Johnson	190 st	11430	657-978-0985	Williams.Johnson@	Male
BC54003	MetroPlus	(845)-732-2096	Medicade	7/13/2024	MRS00001	manu	Sultan	170 st	11430	657-978-0499	Sulatan.Manu@gma	Male
c54003	MetroPlus	845-732-2096	Medicaid	3/20/2024	MRS00002	Leilani	Alaina	180 st	11430	657-978-9575	Leilani.Alaina@gr	Femal
c54003	MetroPlus	845-732-2096	Medicaid	3/20/2024	MRS00004	Ming	Li	160 st	11430	657-978-2375	Ming.Li@gmail.cor	Femal

The final table combination was to link patients with their insurance they have in case the insurance company needs contacting or charged.

6. Patient's Insurance ID

SQL:

```
SELECT Name_First, Name_Last, Insurance_ID

FROM Patients_final;
```



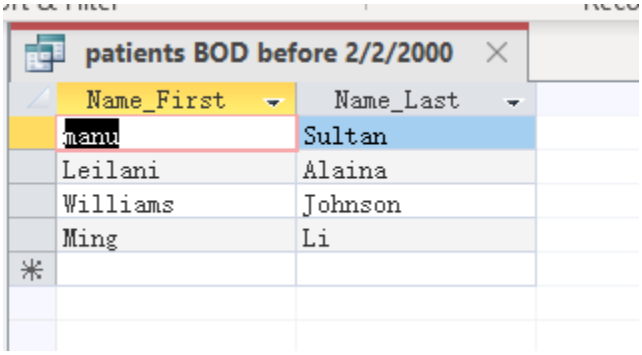
Name_First	Name_Last	Insurance_ID
manu	Sultan	BC54003
Leilani	Alaina	c54003
Williams	Johnson	ABC01231
Ming	Li	c54003

Instead of showing all information about a patients insurance data, this query gives the insurance lookup id for each patient.

7. older than 21 year Patients

SQL:

```
SELECT Name_First, Name_Last

FROM Patients_final

WHERE DOB > 2/2/2000;
```

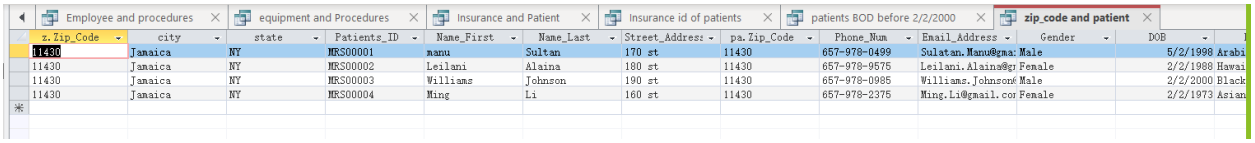


This query was made to find any patients that are older than 21 years old (pretend the dates match up please *wink)

8. A combination o Zipcode and Patients

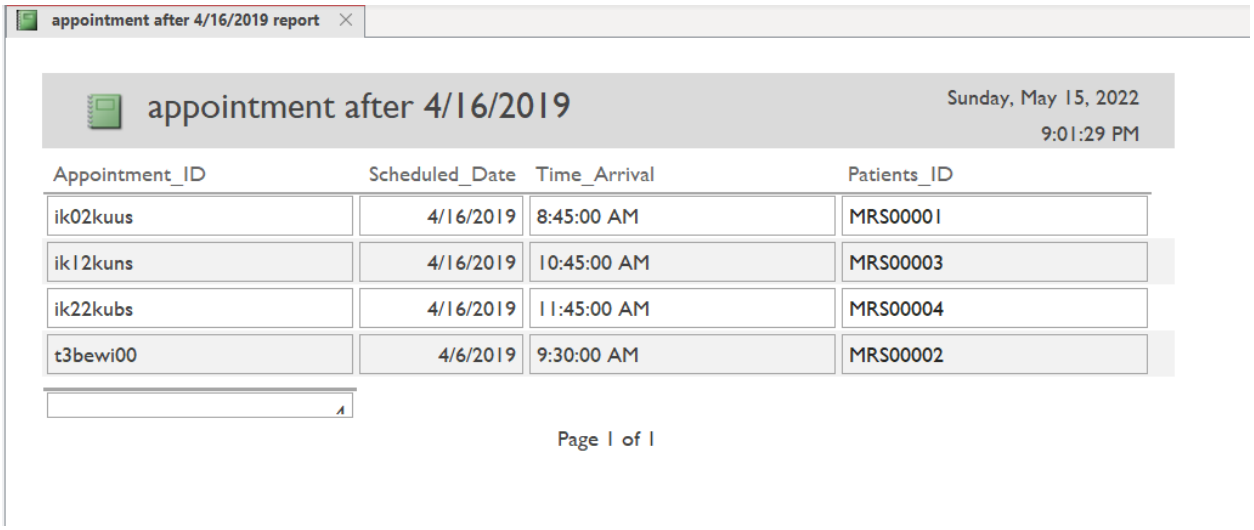
SQL:

```
SELECT *

FROM zip_codes AS z

INNER JOIN Patients_final AS pa ON

z.Zip_Code= pa.Zip_Code;
```



For the final query the zip codes and state info was shown along with all the data about a patient.

Reports


appointment after 4/16/2019 report

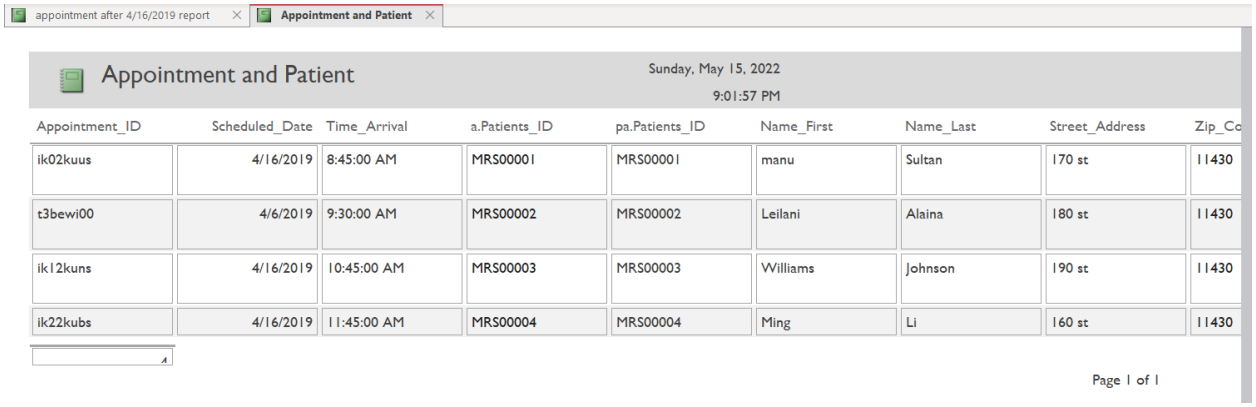
appointment after 4/16/2019

Sunday, May 15, 2022
9:01:29 PM

Appointment_ID	Scheduled_Date	Time_Arrival	Patients_ID
ik02kuus	4/16/2019	8:45:00 AM	MRS00001
ik12kuns	4/16/2019	10:45:00 AM	MRS00003
ik22kubs	4/16/2019	11:45:00 AM	MRS00004
t3bewi00	4/6/2019	9:30:00 AM	MRS00002

Page 1 of 1

This report shows appointments set after 4/16/2019. At the time of running this report it was to show appointments a month or more out.


appointment after 4/16/2019 report | Appointment and Patient

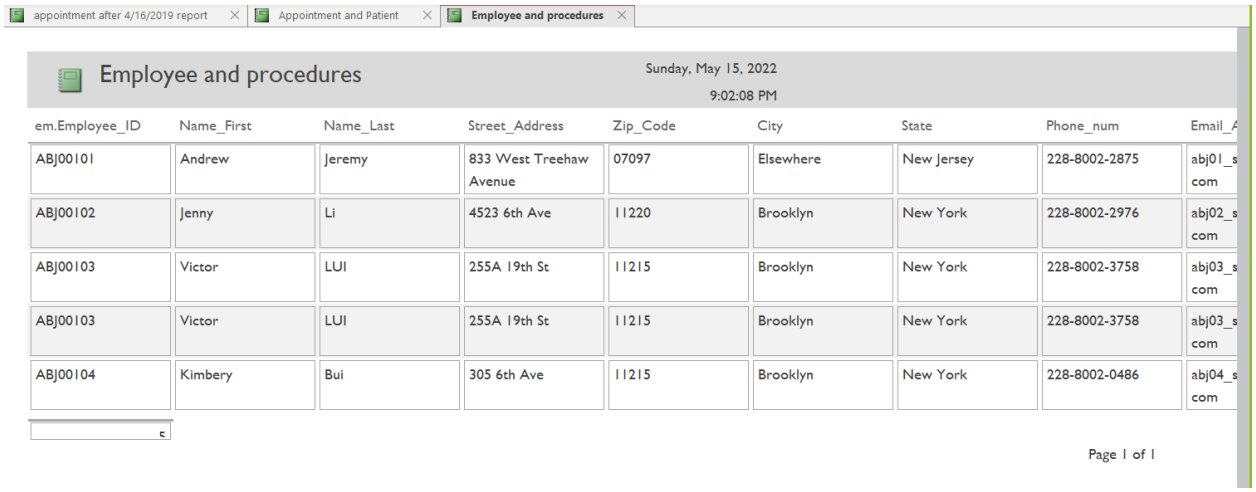
Appointment and Patient

Sunday, May 15, 2022
9:01:57 PM

Appointment_ID	Scheduled_Date	Time_Arrival	a.Patients_ID	pa.Patients_ID	Name_First	Name_Last	Street_Address	Zip_Co
ik02kuus	4/16/2019	8:45:00 AM	MRS00001	MRS00001	manu	Sultan	170 st	11430
t3bewi00	4/6/2019	9:30:00 AM	MRS00002	MRS00002	Leilani	Alaina	180 st	11430
ik12kuns	4/16/2019	10:45:00 AM	MRS00003	MRS00003	Williams	Johnson	190 st	11430
ik22kubs	4/16/2019	11:45:00 AM	MRS00004	MRS00004	Ming	Li	160 st	11430

Page 1 of 1

Presented are all appointments and the patient info.



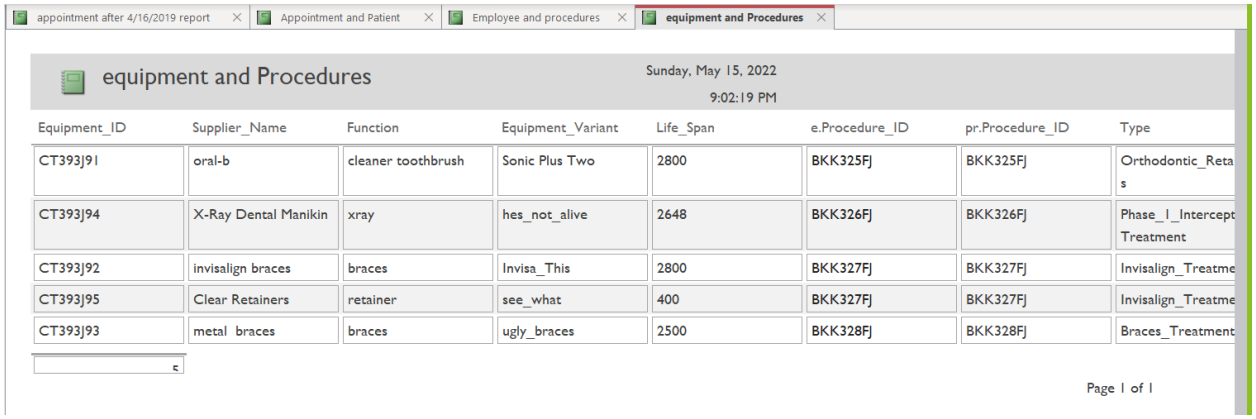
Employee and procedures

Sunday, May 15, 2022 9:02:08 PM

em.Employee_ID	Name_First	Name_Last	Street_Address	Zip_Code	City	State	Phone_num	Email_A
ABJ00101	Andrew	Jeremy	833 West Treehaw Avenue	07097	Elsewhere	New Jersey	228-8002-2875	abj01_s com
ABJ00102	Jenny	Li	4523 6th Ave	11220	Brooklyn	New York	228-8002-2976	abj02_s com
ABJ00103	Victor	LUI	255A 19th St	11215	Brooklyn	New York	228-8002-3758	abj03_s com
ABJ00103	Victor	LUI	255A 19th St	11215	Brooklyn	New York	228-8002-3758	abj03_s com
ABJ00104	Kimbery	Bui	305 6th Ave	11215	Brooklyn	New York	228-8002-0486	abj04_s com

Page 1 of 1

An easily readable report of what employees are working on which procedures.



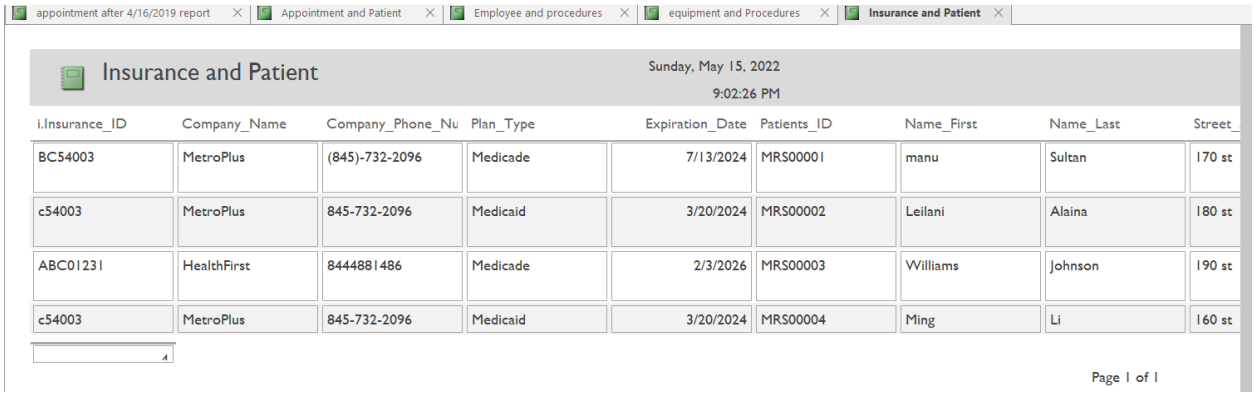
equipment and Procedures

Sunday, May 15, 2022 9:02:19 PM

Equipment_ID	Supplier_Name	Function	Equipment_Variant	Life_Span	e.Procedure_ID	pr.Procedure_ID	Type
CT393J91	oral-b	cleaner toothbrush	Sonic Plus Two	2800	BKK325FJ	BKK325FJ	Orthodontic_Reta s
CT393J94	X-Ray Dental Manikin	xray	hes_not_alive	2648	BKK326FJ	BKK326FJ	Phase_1_Intercept Treatment
CT393J92	invisalign braces	braces	Invisa_This	2800	BKK327FJ	BKK327FJ	Invisalign_Treatme
CT393J95	Clear Retainers	retainer	see_what	400	BKK327FJ	BKK327FJ	Invisalign_Treatme
CT393J93	metal braces	braces	ugly_braces	2500	BKK328FJ	BKK328FJ	Braces_Treatment

Page 1 of 1

For procedures this report shows what equipment is needed for each one.



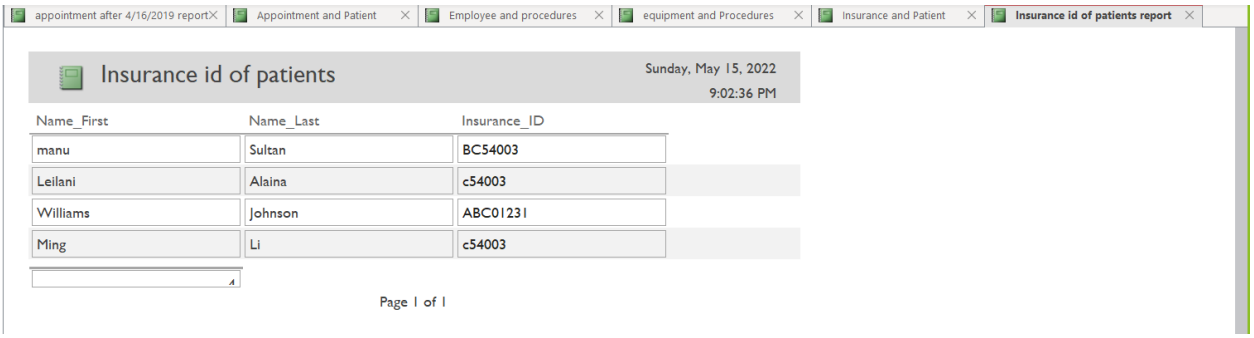
Insurance and Patient

Sunday, May 15, 2022 9:02:26 PM

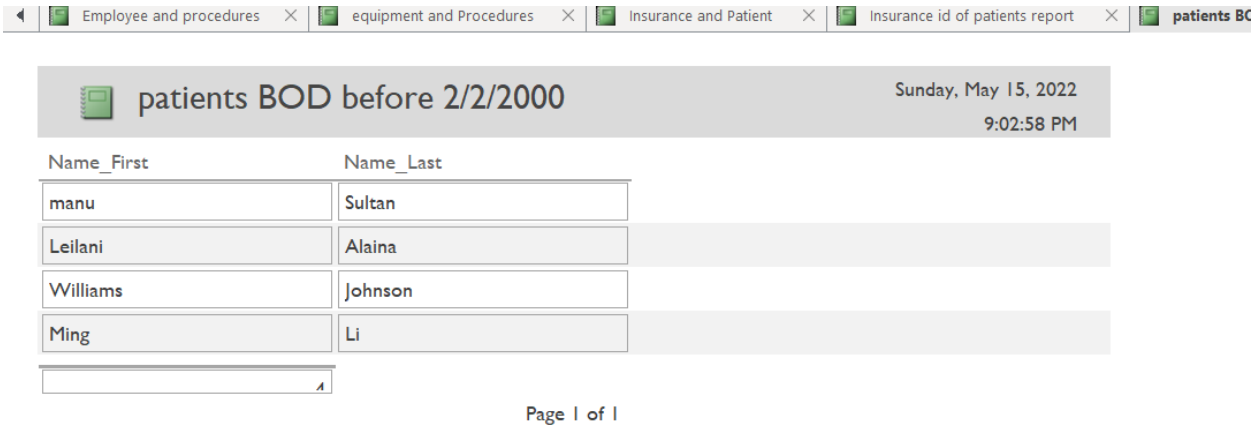
i.Insurance_ID	Company_Name	Company_Phone_Nu	Plan_Type	Expiration_Date	Patients_ID	Name_First	Name_Last	Street_
BC54003	MetroPlus	(845)-732-2096	Medicade	7/13/2024	MRS00001	manu	Sultan	170 st
c54003	MetroPlus	845-732-2096	Medicaid	3/20/2024	MRS00002	Leilani	Alaina	180 st
ABC01231	HealthFirst	8444881486	Medicade	2/3/2026	MRS00003	Williams	Johnson	190 st
c54003	MetroPlus	845-732-2096	Medicaid	3/20/2024	MRS00004	Ming	Li	160 st

Page 1 of 1

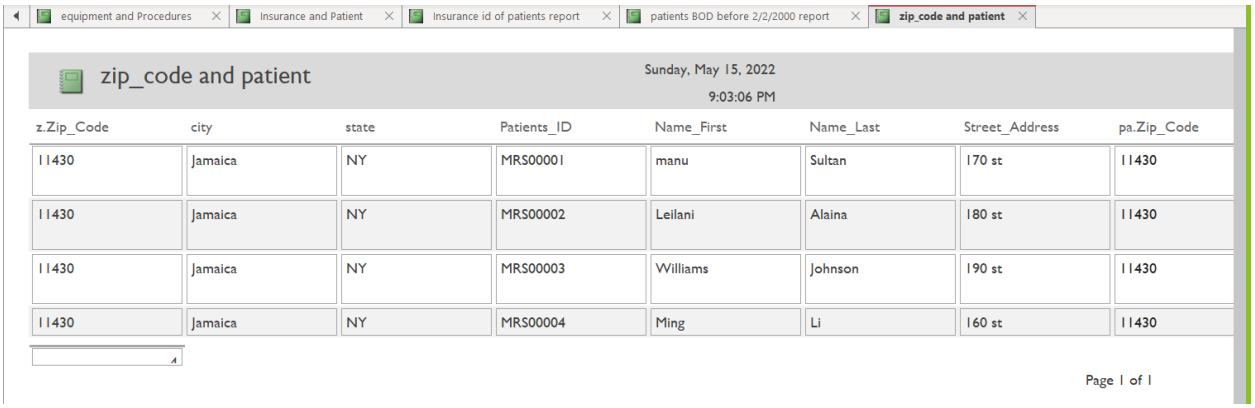
In case an employee needs to check what insurance is associated with a customer they can check here.



A simple report showing the insurance lookup id for a patient.

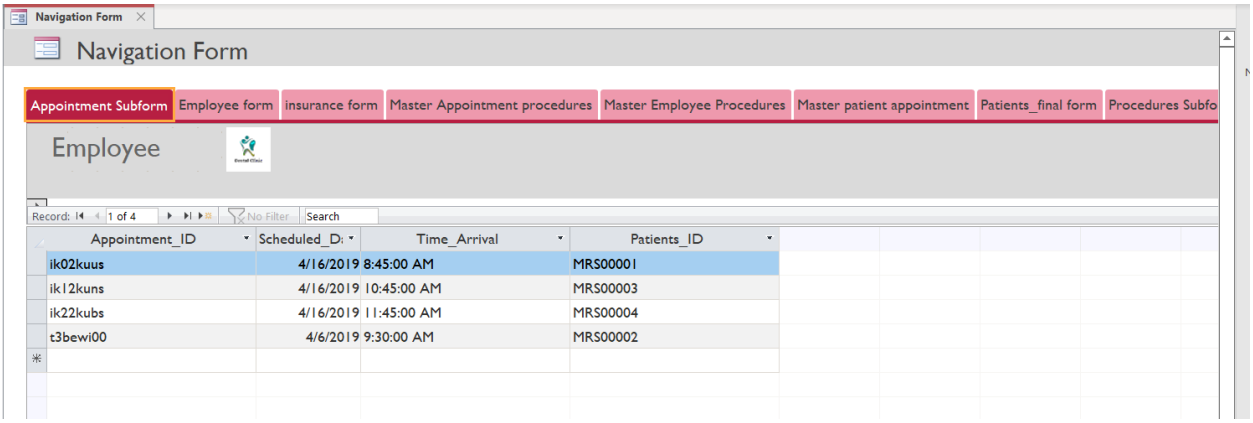


This report presents what patients are older than the age of 21 (at the time of search).



This last report shows all the zip codes in the database and who they are associated to.

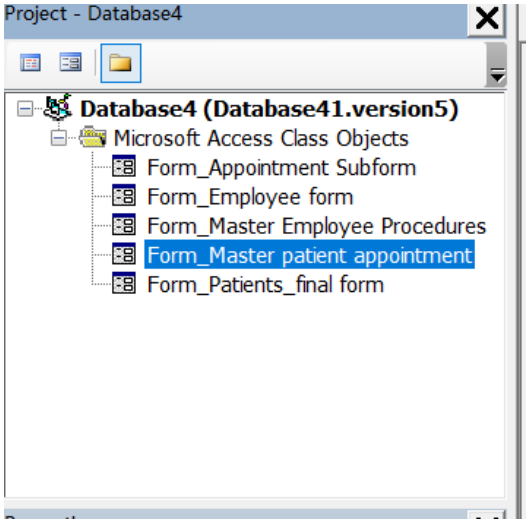
Navigation Form



Navigation form puts all the forms in one table, which makes it easier to retrieve, update, and view info.

Misc VBA Screenshot:

All the tables that use some form of VBA code.



Smile Inc Dental Clinic

Narrative Conclusion

Our group utilized several applications to collaborate and design The Smiles Inc Database. The first and one of the most critical software we used to collaborate was Zoom for the video capabilities, we also used Whatsapp to arrange meetings and discuss times available. For compilation and sharing capabilities we utilized Google Suite, we used Google Drive to save our documents, We used slides for our initial UML notations and ER modeling and Google Docs to document other areas of our project. Finally we used Microsoft Access to bring our database together. We used Powerpoint to help us with initial designing as well and Microsoft Outlook to communicate with our professor and provide updates or request help.

Our experience initially was pretty seamless, we were able to arrange meetings and discuss ideas, our biggest issue initially was deciding what type of organization we wanted to create a database for but upon receiving the greenlight for a dental office we proceeded and worked toward this database. After working on it for a few weeks, we sometimes found ourselves to have differences in opinions but eventually arrived at mutual agreements.

Though the proposal portion was a bit difficult it was the easiest portion since we only had to create an idea of what we would be working on. The application implementation was one of the most difficult portions because we faced not only errors in the code but also issues with scheduling meetings because of other class obligations and personal issues. We learned that database management requires a lot of attention to detail. If we could start all over again we would probably try to be more organized and handle feedback immediately or perhaps have someone who was able to at the time address issues that needed to be fixed.