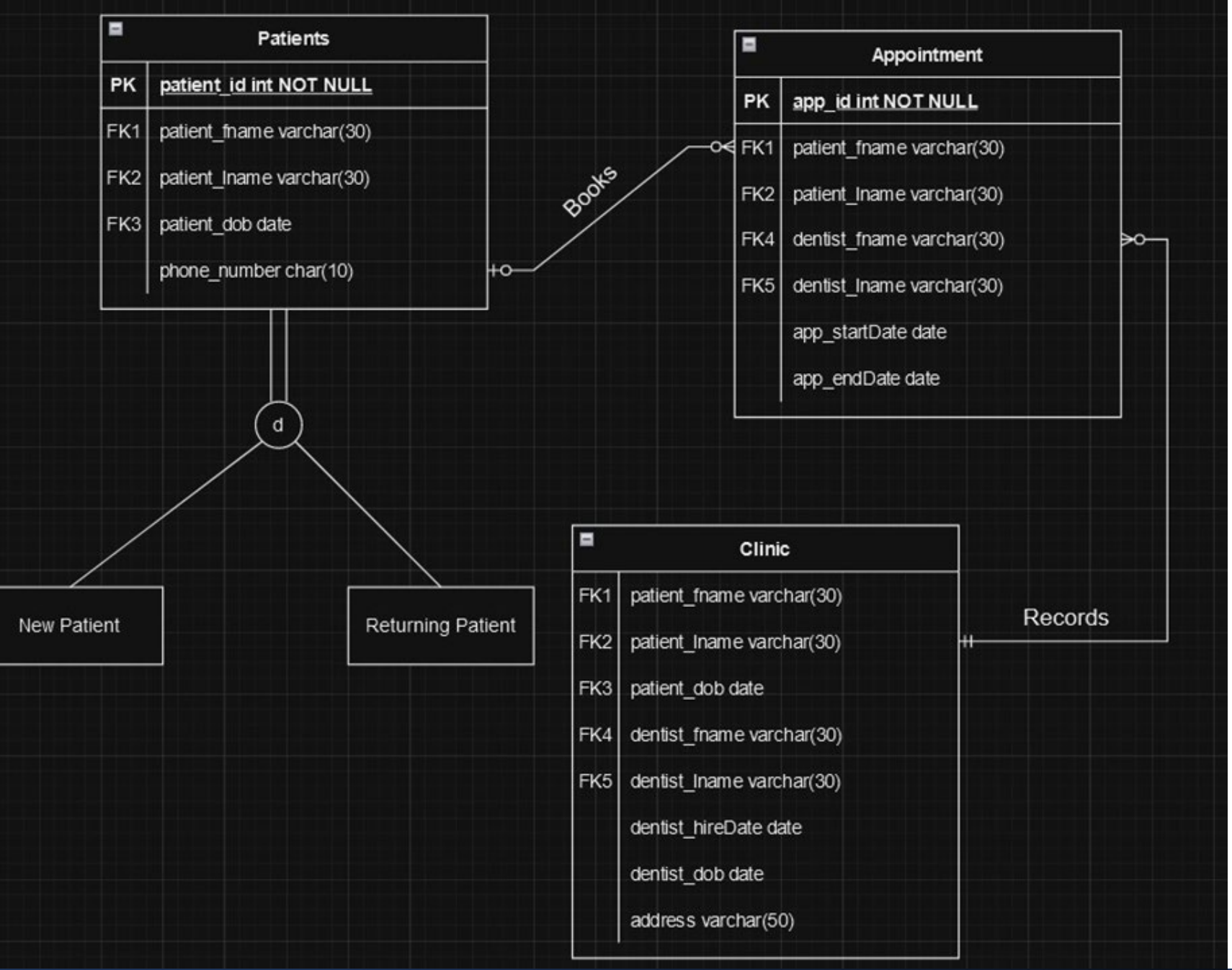
Dental Clinic Database

Members:

Nathan Beimnet Gebrezghi
Wendell Destang



EER Diagram

- Our project will be based on a dental clinic
- This diagram helps to establish relationships between patients and appointments, with clinic

SQL Database Creation & Queries

```sql
create table Patients(
    patient_id int primary key not null,
    patient_fname varchar(30),
    patient_lname varchar(30),
    patient_dob date,
    phone_number char(10)
);

insert into Patients
values(001, 'Mike', 'Butowski', '2000-04-21', '905-829-4791'),
      (002, 'Kimberly', 'Perkins', '2004-01-19', '905-814-6833'),
      (003, 'Brad', 'Magnuson', '1993-10-01', '905-855-0432'),
      (004, 'Mike', 'Perkins', '1999-06-12', '905-834-1945');

create table NewPatient(
    patient_id int,
    foreign key (patient_id) references Patients(patient_id)
);

create table ReturningPatient(
    patient_id int,
    foreign key (patient_id) references Patients(patient_id)
);

create table Appointment(
    app_id int primary key not null,
    patient_fname varchar(30),
    patient_lname varchar(30),
    app_startDate date,
    app_endDate date,
    dentist_id int,
    foreign key (dentist_id) references Clinic(dentist_id)
);

insert into Appointment
values(123, 'Mike', 'Butowski', '2022-03-11', '2022-03-11', 200),
      (124, 'Kimberly', 'Perkins', '2023-10-05', '2023-10-06', 201),
      (125, 'Brad', 'Magnuson', '2019-07-26', '2019-07-27', 202),
      (126, 'Mike', 'Perkins', '2020-01-04', '2020-01-07', 203);
```

```sql
create table Clinic(
    dentist_id int primary key not null,
    patient_id int,
    patient_fname varchar(30),
    patient_lname varchar(30),
    patient_dob date,
    dentist_fname varchar(30),
    dentist_lname varchar(30),
    dentist_startDate date,
    dentist_dob date,
    address varchar(50)
    foreign key (patient_id) references Patients(patient_id)
);

insert into Clinic
values(200, 123, 'Mike', 'Butowski', '2000-04-21', 'Bob', 'Diesel', '1994-12-29', '1970-04-22', '3559 Smite Ave.'
```

Reports created using Report Builder

Appointment

app id	patient fname	dentist lname	app Start Date	app End Date	patient_id	dentist_id
123	Mike	Butowski	2022-03-11	2022-03-11	001	200
124	Kimberly	Perkins	2023-10-05	2023-10-06	002	201
125	Brad	Magnuson	2019-07-26	2019-07-27	003	202
126	Mike	Perkins	2020-01-04	2020-01-07	004	203

Clinic

dentist id	patient id	patient fname	patient lname	patient dob	dentist fname	dentist lname	dentist Start Date
200	123	Mike	Butowski	2000-04-21	Bob	Diesel	1970-04-22

Patients

patient id	patient fname	patient lname	patient dob	phone number
001	Mike	Butowski	2000-04-21	905-829-4791
002	Kimberly	Perkins	2004-01-19	905-814-6833
003	Brad	Magnuson	1993-10-01	905-855-0432
004	Mike	Perkins	1999-06-12	905-834-1945

[&ExecutionTime]

Normalized Database Schema

Normalization (Appointment Table)

Patient

patient_id	patient_fname	patient_lname
001	Mike	Butowski

Appointment

App_id	App_startDate	App_endDate	Patient_id	Dentist_id
123	2022-03-11	2022-03-11	001	200

Clinic

Patient

Patient_id	patient_fname	patient_lname	patient_dob
001	Mike	Butowski	2000-04-21

Dentist

dentist_id	dentist_fname	dentist_lname	Dentist_startDate
200	Bob	Diesel	1994-12-29

Mongo DB Collections

```
db.Patients.insertMany([{patient_id: 001, patient_fname: 'Mike', patient_lname: 'Butowski', patient_dob: '2000-04-21', phone_number: '905-829-4791'},
                        {patient_id: 002, patient_fname: 'Kimberly', patient_lname: 'Perkins', patient_dob: '2004-01-19', phone_number: '905-814-6833'},
                        {patient_id: 003, patient_fname: 'Brad', patient_lname: 'Magnuson', patient_dob: '1993-10-01', phone_number: '905-855-0432'},
                        {patient_id: 004, patient_fname: 'Mike', patient_lname: 'Perkins', patient_dob: '1999-06-12', phone_number: '905-834-1945'}
                       ])



db.Appointment.insertMany([{app_id: 123, patient_fname: 'Mike', patient_lname: 'Butowski', app_startDate: '2022-03-11', app_endDate: '2022-03-11', dentist_id: 200},
                           {app_id: 124, patient_fname: 'Kimberly', patient_lname: 'Perkins', app_startDate: '2023-10-05', app_endDate: '2023-10-06', dentist_id: 201},
                           {app_id: 125, patient_fname: 'Brad', patient_lname: 'Magnuson', app_startDate: '2019-07-26', app_endDate: '2019-07-27', dentist_id: 202},
                           {app_id: 126, patient_fname: 'Mike', patient_lname: 'Perkins', app_startDate: '2020-01-04', app_endDate: '2020-01-07', dentist_id: 203}
                          ])
```

NoSQL Queries

```
db.Patients.deleteOne({patient_fname: 'Mike'})
db.Patients.deleteOne({patient_lname: 'Perkins'})
db.Patients.deleteOne({patient_id: 001})
db.Patients.deleteMany({patient_lname: 'Perkins'})
db.Patients.deleteMany({patient_fname: 'Mike'})
db.Patients.deleteMany({})
db.getCollection('Patients').find({})
db.Patients.find({patient_fname: 'Mike'})
db.Patients.find({patient_lname: 'Perkins'})
db.Patients.find({patient_id: {$gt: 1}})
db.Patients.find({patient_fname: {$not: {$eq: 'Mike'}}})
db.Patients.find({patient_id: 001})
db.Patients.find({patient_id: 001}, {patient_fname: 1, patient_lname: 1})
db.Patients.updateMany({}, {$set: {Age: 20}})
db.Pateints.updateMany({$or: [{Age: {$eq: 20}}, {patient_fname: {$eq: 'Mike'}}]},
                       {$set: {patient_id: 005}})
db.Patients.updateOne({patient_fname: 'Mike'},
                      {$set: {phone_number: '905-555-5555'}})
```

Thanks for watching!