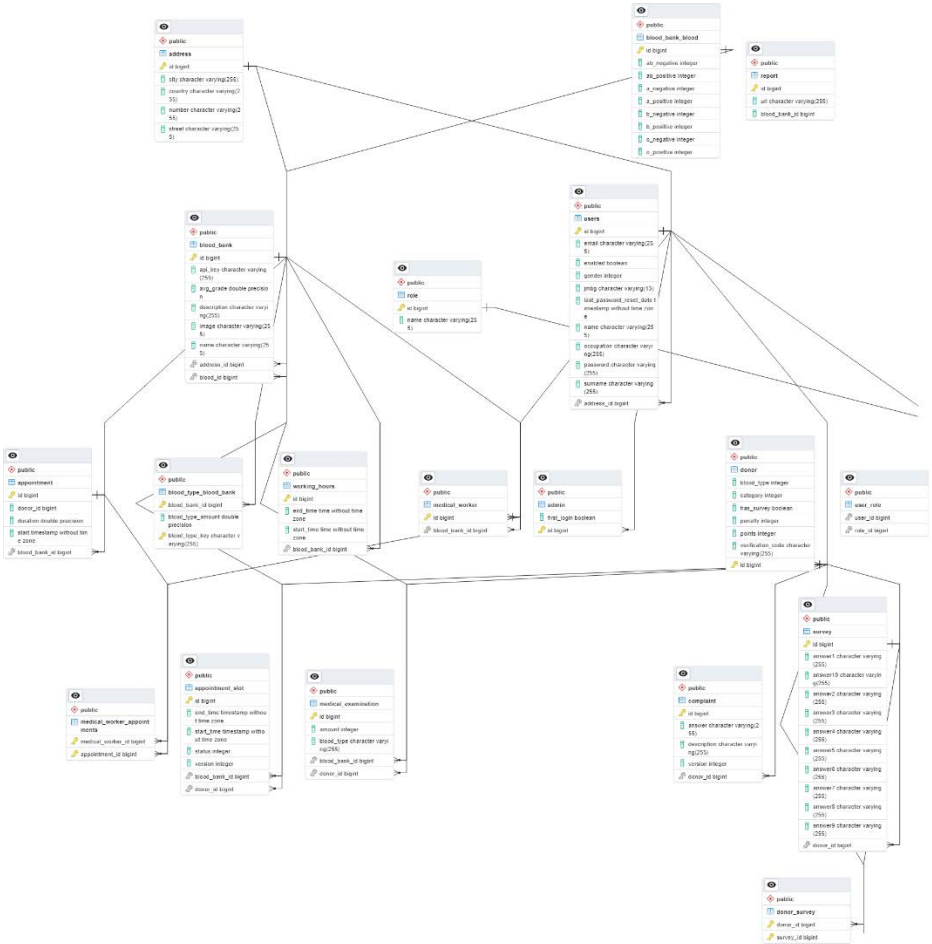
Nikola Kolarov ISA 2022/2023

RA 132/2019

PREDLOG SKALABILNOSTI APLIKACIJE

SKALABILNOST BAZE PODATAKA



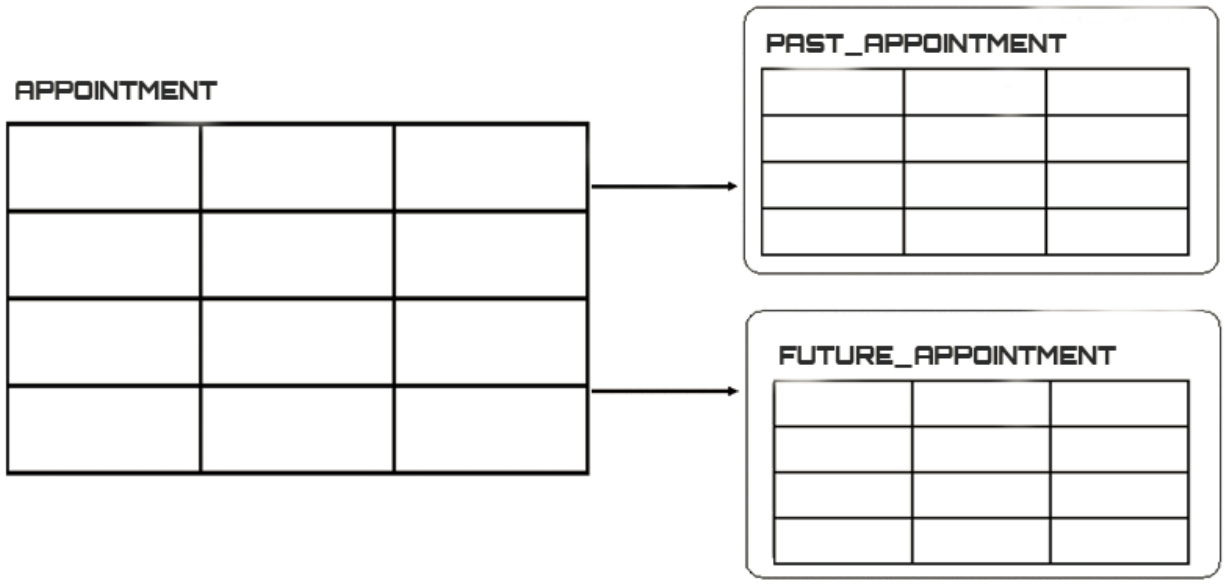
Što se tiče predloga skalabilnosti same baze podataka opredelio sam se za vertikalno skaliranje tj.

Davanje više resursa samoj bazi za funkcionisanje. Dakle, dodela više RAM memorije, procesorske moći, prostora itd. Za ovakvo skaliranje baze opredelio sam se zato što je sa aspekta vremena ovo najefikasnije rešenje. Ono što bih takođe uradio, jeste dodavanje više servera u klaster tj. Podigao bih instancu

bekend servera na više različitih portova, što bi mi kasnije omogućilo da dodam Load Balancer i uradim replikaciju podataka.

PARTICIONISANJE PODATAKA

Što se tiče particionisanja podataka ovde bih se opredelio za horizontalno particionisanje, dakle odabraio bih tabele koje su pogodne za ovaj proces i izdelio na više manjih tabela.



Ovo ću pokazati na primeru tabele Appointment, gde bih particionisanjem ove tabele na dve manje, termini koji su prošli i termini u budućnosti, smanjio vreme obrade, ubacivanja novih podataka i izvlačenja podataka iz ovih tabela značajno.

Ovakav jedan primer već postoji implementiran u našoj aplikaciji (Appointment i AppointmentSlot) gde smo razdvojili termine koje korisnik sam zakazuje i unapred definisane termine u dve različite tabele.

REPLIKACIJA BAZE I OTPORNOST NA GREŠKE

Što se tiče replikacije opredelio bih se za Master-master replikaciju: Ova vrsta replikacije ima više master servera koji se međusobno repliciraju. Svaki master server može primati i izvršavati promene (npr. insert, update, delete) i replicirati ih na druge master servere. Ova metoda omogućava višestruko pisanje.

Otpornost na greške bi bila realizovana putem Transakcione sigurnosti: Koristila bi se transakciona sigurnost i ACID transakcije koje pomažu da se osigura integritet podataka i da se spreče greške u slučaju problema sa hardverom ili mrežom.

Validacija unosa: Validiranje unosa podataka koji se unosi u bazu podataka, koristeći metode kao što su provere formata, kontrole duplikata i provere integriteta referenci, pomoći će da se spreče greške u unosu podataka.

Korišćenje replikacije: Korišćenjem već gorepomenute replikacije osiguraćemo dostupnost podataka i sprečiti gubitak podataka u slučaju gubitka nekog od servera.

KEŠIRANJE PODATAKA

Za keširanje bih iskoristio distribuirano keširanje podataka, tj. na više računara bi se istovremeno čuvali keširani podaci što bi umnogome povećalo performance I skalabilnost jer bih ovim keširanjem smanjio protok saobraćaja samim tim I troškove održavanja, takođe sa distribuiranjem keša na mnogo računara bismo mogli da primimo ogroman broj korisnika.

PROCENA HARDVERSKIH RESURSA

Ako podelimo broj mesečnih rezervacija sa 30, dolazimo do cifre od oko 17000 rezervacija dnevno, što bi značilo da u isto vreme aplikaciju koristi oko 20 hiljada korisnika. Uzećemo po 5MB rama za svakog korisnika gde dolazimo do cifre od oko 100GB RAM memorije potrebne, zbog sigurnosti povećaćemo ovo na 128GB. Za skladištenje se treba opredeliti za SSD od oko 5TB (zbog sigurnosti) I brzinu konekcije od 10Gbit/s. Što se procesora tiče neki minimum bi bio Intel Xeon E1230.

LOAD BALANCER

Što se tiče implementacija load balancer-a, morali bismo prvo podeliti naš backend server na više backend servera. Kada smo to uradili, iskoristićemo Least Connections tehniku za load balancing. Dakle, ako podelimo naš server na 3 preko replikacije koju sam gore opisao, putem load balancer-a ćemo novi zahtev uvek slati onom serveru koji trenutno obrađuje najmanje zahteva. Ovime ćemo drastično povećati performanse i brzinu obrade samih zahteva.

NADGLEDANJE OPERACIJA KORISNIKA

Zakazivanje termina: Potrebno je nadgledati kako se korisnici kreću kroz proces zakazivanja termina i koliko je korisnika završilo proces zakazivanja uspešno.

Otkazivanje termina: Potrebno je nadgledati koliko korisnika otkazuje svoje termine i razlog za otkazivanje.

Pregled termina: Potrebno je nadgledati koliko korisnika pregleda svoje termine i koliko korisnika menja svoje termine.

Pretraga: Potrebno je nadgledati koliko korisnika koristi pretragu za pronalaženje raspoloživih termina i koliko korisnika pronalazi željeni termin.

Nadgledanjem ovih operacija korisnika, moguće je identifikovati probleme sa korisničkim iskustvom i poboljšati sistem da bi se poboljšalo korisničko iskustvo.

DIJAGRAM