Proiect Baze de Date

Meva Hakan 331AA

Sistem de alocare a sarcinilor intr-o firma de prestare servicii la domiciliu

1. Descrierea cerintelor

Pentru a putea executa tema, m-am gandit sa aplic o utilitate practica si anume: o aplicatie de gestiune a unei firme de contructii/design interior.Aceasta se poate ocupa de reorganizarea completa a unei locuinte: de la spart pereti pana la remodelat si apoi curatat.

1. Etapa de proiectare

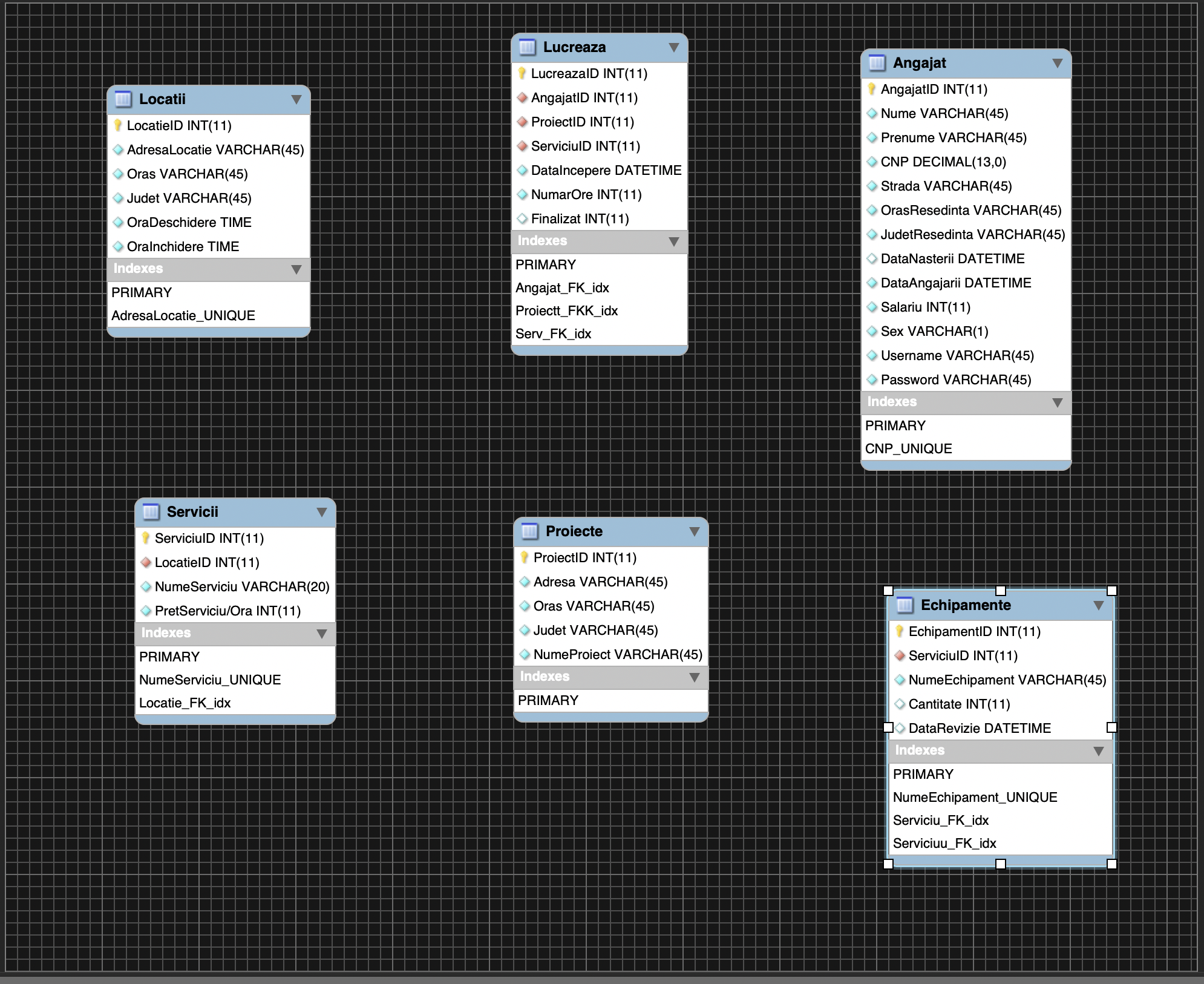
Pentru a proiecta aplicatia, a fost nevoie sa aleg 2 tool-uri absolut necesare: un server SQL si un IDE asociat unui limbaj de programare in care sa fac interfata grafica.Pentru serverul SQL, am ales varianta MySql fiind cea mai potrivita pentru sistemul meu de operare iar ca limbaj de programare am ales Python, prin IDE PyCharm.

Initial, am conceput baza de date in MySql impreuna cu relatii si constangeri.Apoi am populat-o cu cateva elemente pentru ca apoi sa pot tesla functionalitatiile in aplicatie.

Modul in care am gandit aplicatia este unul foarte intuitiv: fereastra initiala reprezinta Login-ul.Aici se face distinctia clara intre un user normal si admin.Astfel in urma procesului de login, in functie de credentials, se va deschide ori o fereastra pentru user normali(angajati) ori o fereastra pentru administrator.In acest fel, angajatii nu pot accesa decat informatii ce ii privesc direct precum: salariu, proiectele la care lucreaza, tipul de proiect, adresa etc. In plus, isi pot actualiza proiectele in cazul in care acestea s-au finalizat sau pot actualiza timpul de lucru necesar la un proiect.

Administratorul are permsiuni depline si poate accesa informatii despre toti utilizatorii.Rolul sau este de a controla munca angajatilor astfel ca poate introduce angajati, locatii , sterge proiecte sau chiar sterge servicii.Toate aceste lucruri se fac intuitiv avand ori casute cu variante ori casute goale unde acesta sa poata introduce informatiile cerute pentru actualizarea bazei de date.Mai mult de atat, acesta poate accesa si informatii utile predefinite prin anumite functionalitati speciale definite mai jos in sectiunea Interogari complexe.

1. Descrierea tabelelor si a relatiilor dintre ele



In tabela „Angajat” sunt stocare informatii despre toti angajatii companiei.Tabela „Locatii” reprezinta toate locatiile unde firma are sediu.Tabela „Servicii” reprezinta toate serviciile oferite de firma.Tabela „Proiecte” pastreaza toate proiectele firmei atat trecute cat si curente si viitoare.Tabela „Echipamente” pastreaza toate materialele/ustensilele necesare angajilor pentru rezolvarea task-urilor.Tabela „Lucreaza” este o tabela de legatura intre angajati, proiectele lor si serviciul asociat proiectului.

Relatii:

Servicii-Echipamente 1:N intrucat mai multe echipamente sunt asociate unui serviciu si numai unuia( ex: matura e folosita doar pentru curatenie, nu si pentru varuit)

Locatii-Servicii 1:N intrucat la o locatie se pot programa/verifica doar unele din servicii iar un serviciu nu face parte din mai multe locatii intrucat s-ar creea haos pentru angajati si clienti

Proiecte-Angajat N:N intrucat un angajat poate avea mai multe proiecte planificate iar la un proiect pot lucra mai multi angajati . Relatia este modelata prin tabelul de legatura : Lucreaza.

Servicii-Proiecte N:N intrucat un proiect poate include mai multe servicii( spart pereti+ curatenie) si pot exista mai multe proiecte ce au nevoie de acelasi tip de serviciu.Relatia este modelata prin tabelul de legatura: Lucreaza .

1. Constrangeri impuse

Angajat: AngajatID reprezinta PrimaryKey. Toate coloanele in afara de DataNasterii sunt Not Null intrucat unii angajati pot sa nu isi cunoasca data nasterii. CNP-ul este Unique fiind unic pentru fiecare cetatean.CNP are si valoare definita de 13.

Echipamente: EchipamentID este PK, ServiciuID este Foreign Key facand referire la tabelul Servicii ,NumeEchipament este Unique iar Cantitate si DataReviziei nu au constrangere Not Null

Locatii: LocatieID este PK, AdresaLocatie este Unique iar toate coloanele sunt Not Null

Lucreaza: LucreazaID este PK, AngajatID FK, ProiectID FK, Serviciu FK.Numarul mare de Foreign Key-uri este dat de faptul ca acesta este un tabel de legatura. Finalizat are ca valoare implicita 0 reprezentand faptul ca un angajat nu si-a terminat treaba la acel proiect. Valoarea 1 inseamna task finalizat.

Proiecte: ProiectID PK iar restul de coloane sunt Not Null

Servicii: ServiciuID PK , LocatieID FK facand referire la tabelul Locatii, NumeServiciu este Unique

Toate Primary Keys sunt autoincrementate de la valoarea initiala 1.

1. Functionarea aplicatiei

Functionarea aplicatiei se bazeaza pe modulele Tkinter si MySql.connector.Initial se realizeaza conexiunea la baza de date dand ca parametrii: host,username,parola si baza de date de care avem nevoie. Avand conexiunea stabilita putem incepe sa lucram la interfata.Modulul Tkinter ne ofera o multitudine de tool-uri precum: Label, Textbox, Dropdown Menu, Button etc. Fiecare dintre acestea este pozitionat intr-o fereastra la o anumita pozitie folosind coordonate.

Aplicatia are 3 window-uri principale si 2 secundare.Cele principale sunt : Login, WindowAngajat si WindowAdmin.Un window secundar este folosit pentru colectarea datelor la introducerea unui angajat nou iar al doilea este un window fara interactiune ce arata ca o instructiune s-a executat cu succes.

Toate functionalitatile merg dupa acelasi principiu.Un buton are asociat unul sau mai multe entrybox-uri in functie de query.La apasarea butonului se acceseaza o functie.Functia respectiva reprezinta executia query-ul efectiv. In cadrul functiei, se acceseaza datele din entrybox-urile necesare, se formateaza query-ul cu datele colectate si se executa.In cazul in care este vorba de un SELECT, in urma executarii, rezultatul se salveaza intr-o variabila ce apoi este afisata intr-un label ce primeste ca parametru variabila rezultat.

Foarte important este de retinut ca sql.connection foloseste o schema proprie obtinuta din baza de date initiala.In cazul in care vrem ca modificarile facute de noi prin INSERT, UPDATE, DELETE sa fie observate in baza de date, este nevoie de commit.Aceasta operatie actualizeaza baza de date in urma operatiunilor facute in programul nostru.

1. Intrerogari

Interogari simple

1. select Salariu from Angajat where Nume= %s AND Prenume= %s

2. SELECT S.NumeServiciu, E.DataRevizie from ProiectBD.Servicii S INNER JOIN ProiectBD.Echipamente E on S.ServiciuID=E.ServiciuID WHERE E.DataRevizie > 2020-10-10

3. SELECT L.DataIncepere from ProiectBD.Lucreaza L INNER JOIN ProiectBD.Angajat A on L.AngajatID=A.AngajatID WHERE A.Nume= %s AND A.Prenume= %s

4. select A.Adresa, A.Oras, A.Judet from ProiectBD.Proiecte A INNER JOIN ProiectBD.Lucreaza B ON A.ProiectID=B.ProiectID INNER JOIN ProiectBD.Angajat C ON B.AngajatID=C.AngajatID WHERE C.Nume= %s AND C.Prenume= %s

5. SELECT L.NumarOre from ProiectBD.Lucreaza L INNER JOIN ProiectBD.Angajat A on L.AngajatID=A.AngajatID WHERE A.Nume= %s AND A.Prenume= %s

6. SELECT S.NumeServiciu FROM ProiectBD.Servicii S INNER JOIN ProiectBD.Lucreaza L ON S.ServiciuID=L.ServiciuID INNER JOIN ProiectBD.Angajat A on L.AngajatID=A.AngajatID WHERE A.Nume= %s AND A.Prenume= %s

7. SELECT L.Finalizat FROM ProiectBD.Lucreaza L INNER JOIN ProiectBD.Angajat A on L.AngajatID=A.AngajatID where Nume= %s AND Prenume= %s

8. SELECT NumeProiect FROM ProiectBD.Proiecte

9. SELECT NumeServiciu FROM ProiectBD.Servicii

10. SELECT AngajatID from ProiectBD.Angajat Where Username= %s and Password= %s

Interogari complexe

1. Sa se afiseze toti angajatii care nu lucreaza la niciun proiect si au Salariu > x.

SELECT Nume, Prenume from ProiectBD.Angajat where Salariu > x AND AngajatID not in ( SELECT AngajatID from ProiectBD.Lucreaza)

2.Sa se afiseze numele si prenumele angajatului cu cel mai mare salariu.

SELECT Nume, Prenume from ProiectBD.Angajat where Salariu = (SELECT MAX(Salariu) from ProiectBD.Angajat)

3. Sa se afiseze numele serviciului si adresa locatiei asociata ce detine echipament mai mult decat media echipamentelor insumate

select S.NumeServiciu , L.AdresaLocatie, SUM(E.Cantitate) from ProiectBD.Servicii S INNER JOIN ProiectBD.Locatii L on S.LocatieID=L.LocatieID INNER JOIN ProiectBD.Echipamente E on S.ServiciuID=E.ServiciuID GROUP BY S.NumeServiciu, L.AdresaLocatie HAVING SUM(E.Cantitate) > ( SELECT AVG(Cantitate) from ProiectBD.Echipamente)

4. Sa se afiseze toti angajatii care vor incepe sa lucreze la un proiect inaintea zilei lor de nastere.

select A.Nume, A.Prenume , DATE\_FORMAT(DataNasterii, '%m-%d') as data, P.NumeProiect from ProiectBD.Angajat A INNER JOIN ProiectBD.Lucreaza L on A.AngajatID=L.AngajatID INNER JOIN ProiectBD.Proiecte P on L.ProiectID=P.ProiectID HAVING data < ( SELECT DATE\_FORMAT(DataIncepere, '%m-%d') FROM ProiectBD.Lucreaza LL where A.AngajatID=LL.AngajatID)