Attendance application

Proiect la disciplina
Baze de Date

Realizat de: Galan Ionut Andrei
Grupa: 1310A

Introducere

Proiectul intitulat Attendance application este creat pentru a facilita realizarea prezentei in cadrul laboratoarele si cursurilor. Acesta poate fi extins la nivel de aplicaţie care sa ruleze la un server.

Principalele puncte pe care le-am luat in calcul in realizarea acestui proiect sunt legate de lucru cu baze de date si nu axat foarte mult pe interfaţa:

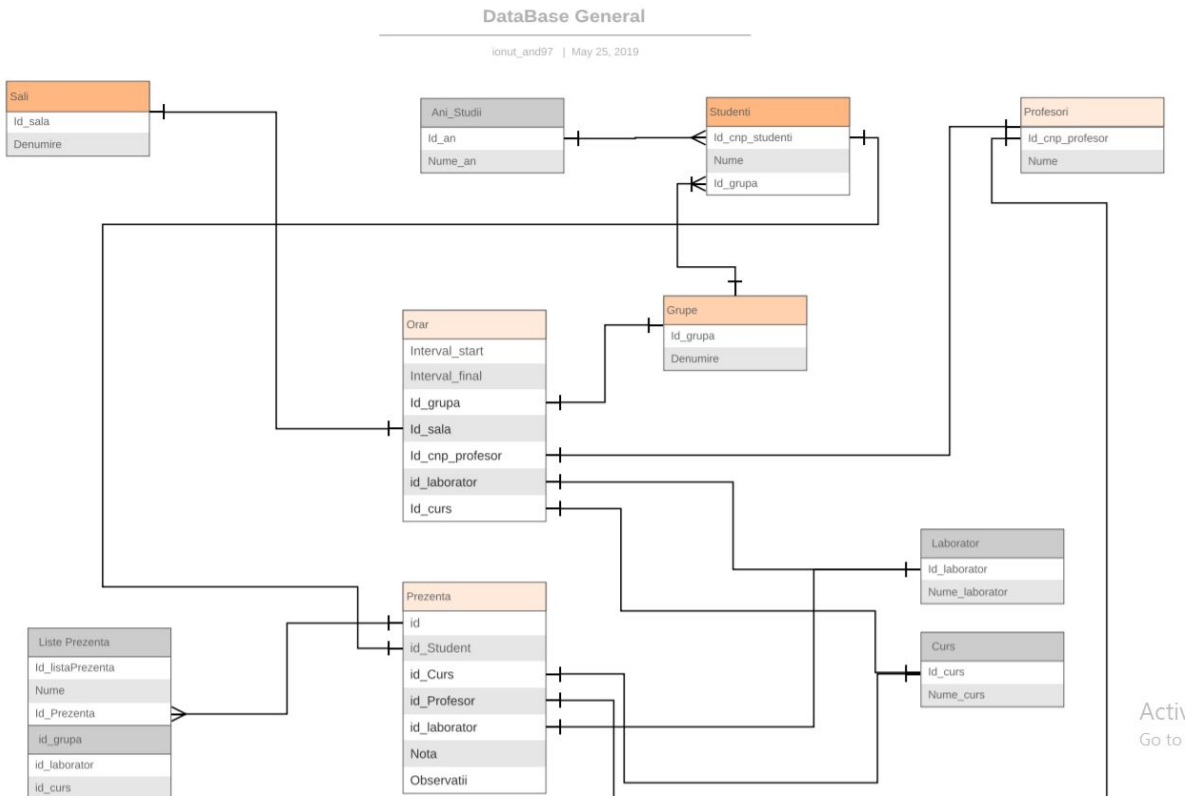
1. Crearea relaţiilor intre tabele
2. Insearea unor valori initiale
3. Interfaţa cu utilizator

Pentru realizare acestor puncte menţionate mai sus am folosit ca si tehnologi SQL Oracle si Java. Ca si IDE am folosit SQLDeveloper pentru interacţiunea cu baza de date in timp ce la interfaţa Eclipse , unde am creat totul folosind clasele din Window Builder.

Relatiile intre tabele (Schema ER)

Conexiunea intre tabele am reprezentat-o pe baza diagramelor ER. Se pot adaugă unele observaţi cu privire la constrângerile de foreign Key pe care nu le-am adaugat la toate tabelele deoarece se încarcă foarte mult desenul si nu era atat de sugestiv. In fişierul de creare al bazei de date se găseşte toate relaţiile intre tabele si câmpurile acestora având constrângerile aferente.

Suplimentar a fost necesara crearea unor triggere care imi permit inserarea automata a id-ului (PK) pentru tabele Liste_prezenta si Prezenta.



Dupa cum se observa in diagrama de mai sus principalele tabele care necesita o explicatie suplimentara sunt Liste_prezenta si Prezenta. In jurul acestor 2 tabele se bazează aproximativ toata logica de funcţionarea aplicaţiei.

Tabela Liste_prezenta are rolul de a stoca o noua lista de prezenta creata de profesor pe care urmează sa se treacă fiecare student in parte. Câmpurile necesare pe care un profesor trebuie sa le completeze pentru ca o lista de prezenta sa fie inserata sunt: Grupa , numele Cursului sau Laboratorului. Daca se creează o lista de prezenta pentru un Curs nu mai este necesara trecerea grupei.

Tabele Prezenta conţine toti studenţii care s-au adăugat pe listele existente de prezenta. Un student se poate trece pe lista de prezenta chiar dacă nu face parte din grupa respectiva. Tabela studenţi identifica acest lucru.

Inserarea Valorilor

Validarea constrângerilor şi a relaţiilor intre tabele s-a realizata prin inserarea unor seturi de valori iniţiale care pe parcurs au fost completate cu alte valori introduse din interfaţa. Valorile introduse nu acoperă toate cazurile posibile astfel încât sa se realizeze o validare perfecta.

Fiecare tabela conţine cate o constrangere PK sub forma de id_*****(nume tabel) care relizează unicitatea fiecărei înregistrări.

Tabela Liste_prezenta conţine urmatoarele campuri:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_LISTE_PREZENTA	NUMBER(5,0)	No	(null)	1	(null)
2	NUME_LISTE_PREZENTA	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	ID_GRUPA	NUMBER(38,0)	Yes	(null)	3	(null)
4	SAPTAMANA	NUMBER	Yes	(null)	4	(null)
5	ID_CURS	NUMBER(38,0)	Yes	(null)	5	(null)
6	ID_LABORATOR	NUMBER(38,0)	Yes	(null)	6	(null)

Am utilizat constrângerile foreing key pentru a face legăturile cu tabele Grupe (id_grupa), Cursuri(id_curs), Laboratoare(id_laboratoare) si o constrângere de tip Check pentru verificarea datelor introduse. Alte tipuri de validare se fac si in cadrul interfeţei in backend-ul aplicaţiei.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1	ID_CURS_LISTE_PREZENTA_FK	Foreign_Key	(null)	SYSTEM	CURSURI	ID_CURS_PK
2	ID_GRUP_LISTE_PREZENTA_FK	Foreign_Key	(null)	SYSTEM	GRUPE	ID_GRUPA_PK
3	ID_LABORATOR_LISTE_PREZENTA_FK	Foreign_Key	(null)	SYSTEM	LABORATOARE	ID_LABORATOR_PK
4	LISTE_PREZENTA_PK	Primary_Key	(null)	(null)	(null)	(null)
5	SYS_C007255	Check	"ID_LISTE_PREZENTA" IS NOT NULL	(null)	(null)	(null)
6	SYS_C007264	Check	saptamana>0 and saptamana <15	(null)	(null)	(null)

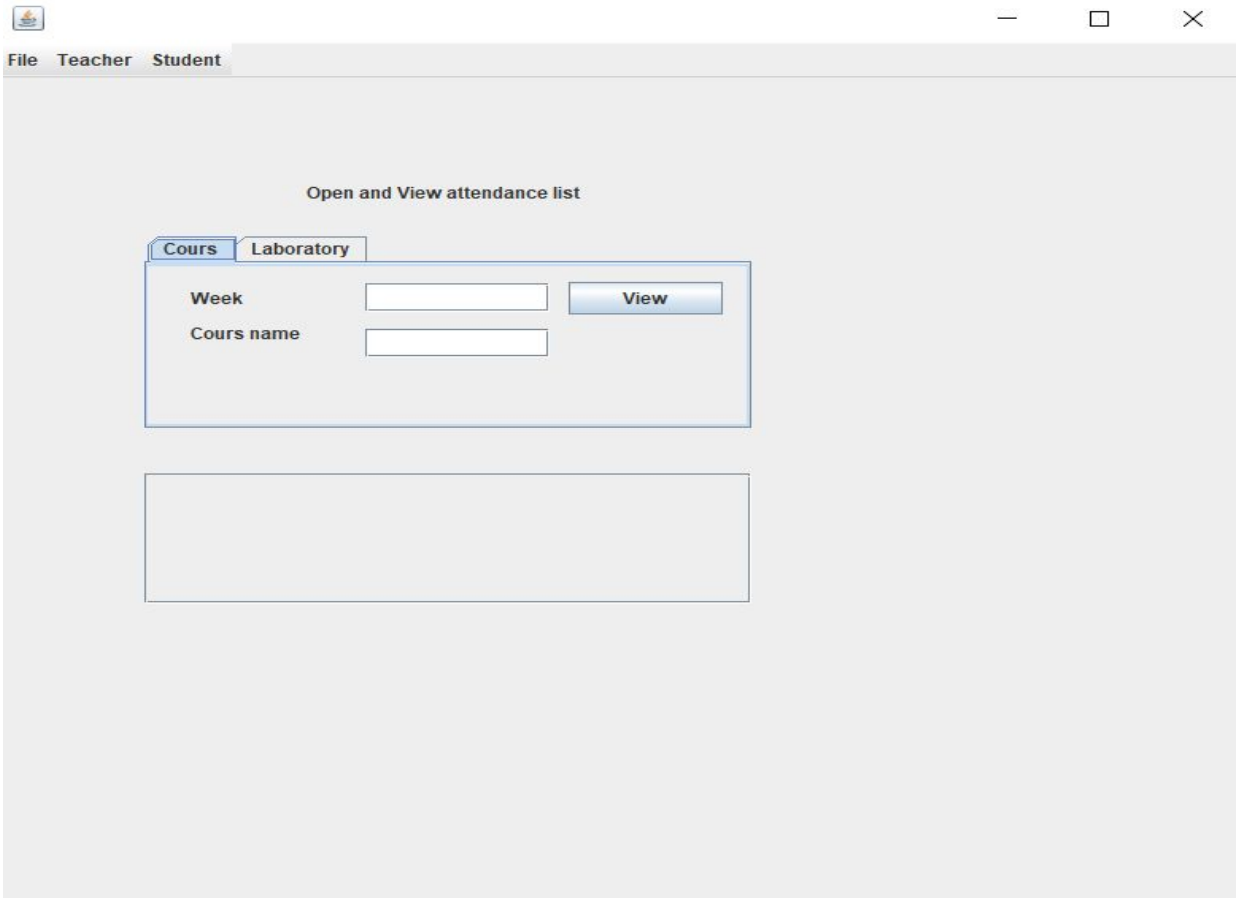
Funcţionalitatea aplicaţiei

Funcţionalitatea aplicaţie, pentru înţelegere mai uşoara a lucrului cu baza de date, am ales sa o descriu print-un fişier sql in care introduc toate interogările pe care le fac in mediul de dezvoltare eclipse. Aşadar

parcurgerea lor poate ajuta la înţelegerea intregii funcţionalităţi şi modului de realizare al acesteia din perspectiva relaţiile cu baza de date.

Interfaţa aplicatiei este sugestiva si are urmatoarea functionalitate:

1. In meniul exista doua entităţi Teacher cat si Student. Teacher conţine operaţiile care sunt realizate doar de către profesor. Student conţine operaţiile care pot fi realizate de studenţi.
2. Teacher poate efectua:
    a. Crearea unei noi liste prezenta
    b. Ştergerea unei liste prezenta
    c. Vizualizarea unei liste de prezenţă
    d. Adăugarea unor informaţii pentru studenţii prezenţi (Nota la laborator sau Observaţii cu privire la activitatea unuia)
3. Student poate efectua
    a. Adăugarea pe o lista de prezenta
4. File contine operaţii asupra aplicaţiei (Momentan doar exit)



Conectarea cu baza de date

Conexiunea cu baza de date se face prin intermediul pachetului jar care se afla in folderul SqlDeveloper. Principiul de conectare are la baza o clasa care implementează designul Singleton care permite crearea unei singure instante indiferent de chiar dacă se încercă crearea mai multor instante.

```
8  public class ConnectionClass {
9      public static Connection dbConnect() {
10         try {
11             Class.forName("oracle.jdbc.driver.OracleDriver");
12             Connection conn= DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "******");
13             return conn;
14         }
15         catch(Exception e) {
16             JOptionPane.showMessageDialog(null, e);
17             return null;
18         }
19     }
20 }
21
```

Concluzii

Proiectul realizat m-a ajutat sa imi consolidez cunoştinţele dobândite in cadrul laboratorului si sa înţeleg importanta creerii unei arhitecturi corecte si bune pentru o baza de date.

Îmbunatăţirile pe care urmează sa le aduc proiectului sunt:

1. La nivelul interfeţei sa se pună la dispoziţia utilizatorului pentru fiecare valoarea care urmează a fi introdusa o lista de elemente din care trebuie sa aleagă. Astfel evit erorile in care numele introdus nu exista in baza de date
2. Crearea mai multor tabele pentru fiecare lista de prezenta creata. Nu doar una singura cum este in cazul de fata.
3. Crearea si unui orar folosindu-ma de tabela care a fost deja creata.