

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Název projektu

Seminární práce z předmětu Databázové systémy II

Pracovní tým:

Michael Tichý, Vojtěch Martinec

Vdne.....

Obsah

Table of Contents

Zadání.....	10
Základní popis používané aplikace.....	12
Instalace.....	12
Přístupová oprávnění.....	12
Použití aplikace.....	12
Datová část.....	14
Analýza.....	14
Fyzický model dat.....	14
Číselníky.....	15
Pohledy.....	15
Funkce.....	16
Uložené procedury.....	17
Spouště.....	17
Sekvence.....	17
Aplikace.....	18
Použité prostředí.....	18
Řízení uživatelských účtů.....	18
Moduly.....	18
Formuláře.....	18
Orientace ve zdrojovém kódu.....	18

Úvod

Toto je dokumentace databázové aplikace pro soukromou vysokou školu. V této databázi bude možné prohlížet, editovat a vkládat nové data do tabulek.

1 Zadání

Cílem práce je vytvořit jednoduchou databázovou aplikaci pro soukromou vysokou školu, která umožňuje přidávat, modifikovat a mazat záznamy v tabulkách. Tato aplikace bude sloužit k tvorbě závazků a rozvrhů vyučujících, evidenci vyučujících, jejich rolí na předmětech. Aplikace musí rozlišovat role přihlášených uživatelů (Administrátor, vyučující, anonym).

Aplikace musí obsahovat následující záložky: Evidence vyučujících, pracovišť, předmětů, studijních oborů, studijních plánů a evidenci a tvoření rozvrhů.

Uživatelská dokumentace

1.1 Základní popis používané aplikace

Tato aplikace představuje databázový program pro vysokou školu. Je zde možné zobrazit a editovat data o dané universitě. Nalezneme zde seznamy Fakult, kateder, studijních oborů a plánů, rozvrhových akcí, vyučujících.

1.2 Instalace

Není potřeba žádná instalace. Stačí spustit soubor DB_semPrace_TichyMartinec.jar v projektu pod složkou dist.

1.3 Přístupová oprávnění

Aplikace umožňuje přisupovat pod třemi uživatelskými rolemi. Admin, který může spravovat všechny záznamy. Uživatel, který může spravovat pouze svůj rozvrh. Normal, který může pouze prohlížet.

1.4 Použití aplikace

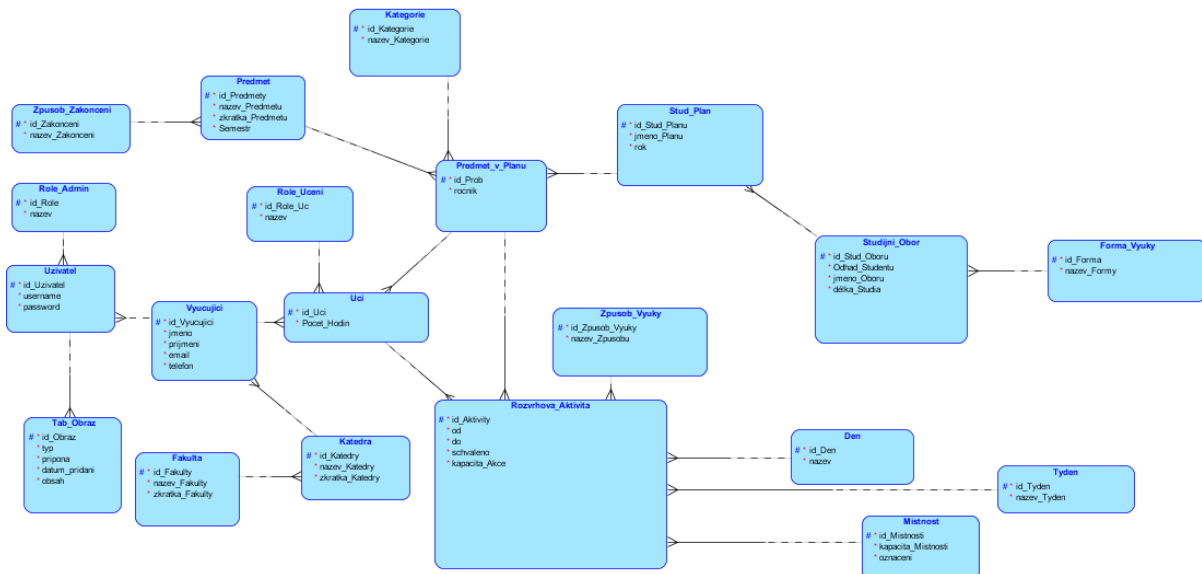
Aplikace obsahuje 9 záložek s různými záznamy. Od vyučujících až po stavbu rozvrhu. Uživatel si pomocí fomulářů může přidávat, upravovat a mazat záznamy, ke kterým má přístup.

Programová dokumentace

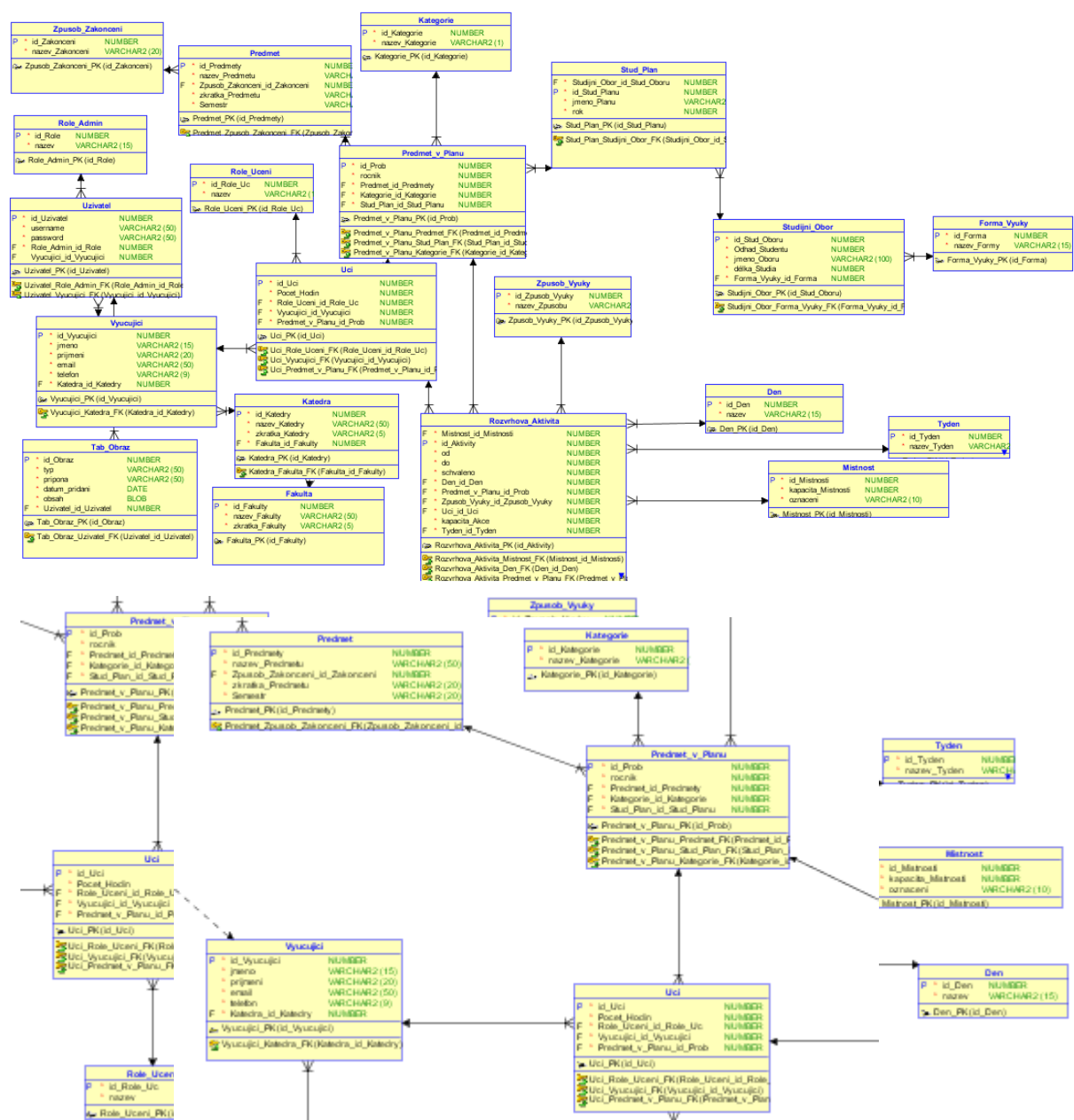
2 Datová část

Zde uveďte řešení zadavatelské dokumentace z pohledu datové části, postupujte dle zadání projektu.

2.1 Analýza



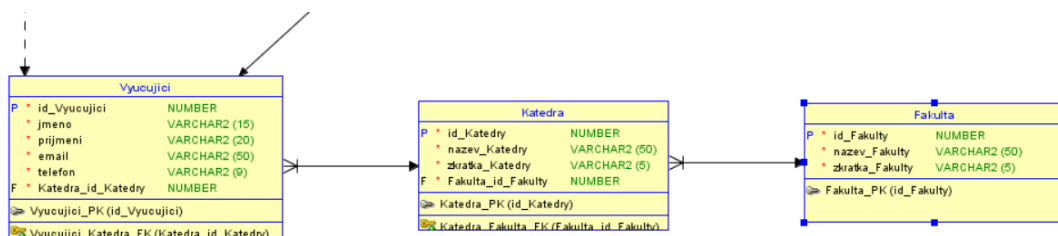
2.2 Fyzický model dat



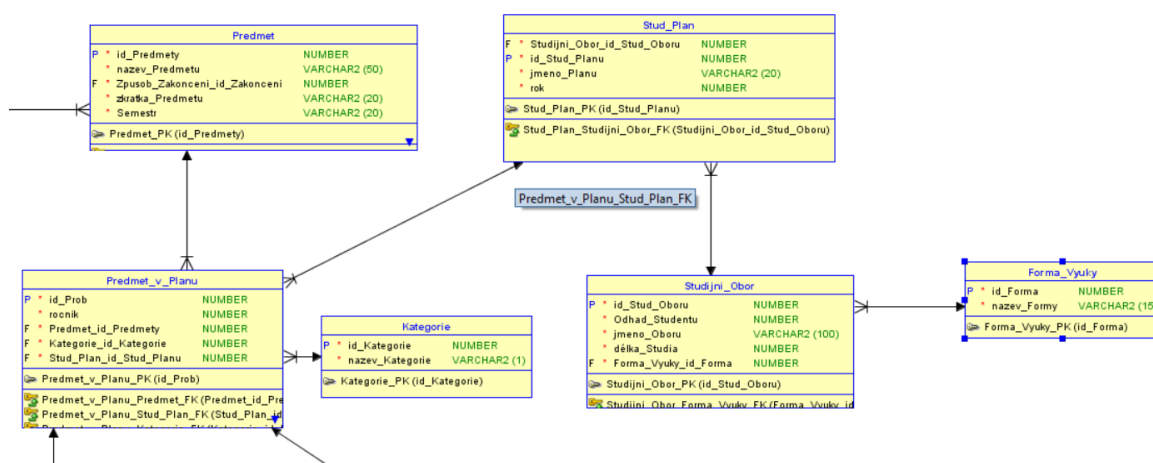
Segment rozvrhu. Zpusob_vyuky, Tyden a Den jsou číselníky dodávající data Rozvrhové aktivitě. Tabulka učí svazuje akci s učitelem a předmětem v konkrétním plánu.

2.3

Segment vyučujícího. Tabulka vyučující je, mimo jiné, navázána na předmět skrze tabulku učí. Ta určuje roli a počet hodin, které učitel v daném předmětu plánu má.



Segment pracoviště. Fakulta má katedry, katedry mají vyučující. Fakulty a katedry dohromady tvoří pracoviště.



Segment předmětu. Předmět je spojen s plánem a oborem pomocí tabulky Predmet_v_Planu. Předmět tak může být v jiné kategorii pro každý plán a mít pro něj jiný rozvrh.

Součástí schématu je pak i tabulka uživatelů navázaná na učitele tak, aby se mohli přihlásit i uživatelé, kteří nejsou učitelé. Každý uživatel také může vložit obrázek.

2.4 Číselníky

Číselníků je v databázi několik, např. tabulka Den, Týden, nebo Kategorie. Data jsou v nich vložena manuálně a nepočítá se s jejich další úpravou.

2.5 Pohledy

view_ucitel_uci - Jednoduchý testovací pohled pro zobrazení učitelů, předmětů, které učí, a

uzivatel_ucitel2 - Pohled pro získání informací o uživateli. Užíván především aplikačně.

view_ucitel_uci2 - Pohled pro získání informací o vyučujícím, co všechno učí, pod jakou rolí, které patří fakultě. Užíván především aplikačně.

uzivatel_ucitel1 - Jednoduchý pohled pro kontrolu dat o uživateli.

uzivatel_role - Jednoduchý pohled pro kontrolu dat o uživateli. Používán pro selekty v aplikaci.

view_obor - Jednoduchý pohled pro výběr dat oboru. Používán aplikačně.

view_rozvrhova_akce2 - Pohled využívaný funkcemi i aplikací, zobrazuje kompletní data rozvrhové akce pro jejich zobrazení.

view_uc_kat_fak - Jednoduchý pohled pro výběr dat o učiteli spolu s jeho pracovištěm. Používán pro selekty v aplikaci.

view_predmety_uplny - Jednoduchý pohled pro výběr dat o předmětu jeho plánem. Používán v databázi.

view_predmety_uplny2 - Komprehensivní pohled pro výběr dat o předmětu jeho plánem. Používán v databázi i v aplikaci.

view_obor_plan - Jednoduchý view zobrazující plány v oboru.

db_pracoviste - Pohled pro výběr dat o pracovišti. Používán v aplikaci.

ucitel_katedra - Jednoduchý pohled pro kontrolu učitele v jeho katedře.

view_predmety - Jednoduchý pohled pro kontrolu předmětu.

2.6 Funkce

func_delete_vlozi(ucitel_id NUMBER, pred_plan NUMBER) - Funkce maže záznamy z tabulky UCI.

func_update_plan(id_planu Number, studo Number, jmeno Varchar2, rokk Number) - Funkce edituje záznamy z tabulky plán tak, aby nevznikaly duplicity v rozvrhu.

func_update_pvp(id_pvp NUMBER, zkratka Varchar2, zpusob Varchar2, kateg Varchar2, plan_id Number, roc Number) - Funkce edituje záznamy z tabulky předmětu v plánu tak, aby nevznikaly duplicity v rozvrhu.

func_update_rozvrh(id_ak NUMBER, kapac NUMBER, misto NUMBER, tydenn VARCHAR2, od_cas NUMBER, do_cas NUMBER, zpusob VARCHAR2, denn VARCHAR2, ucitel_id NUMBER, predmet_id NUMBER, rokk NUMBER, schval NUMBER) - Funkce edituje záznamy z tabulky rozvrh tak aby nevznikali duplicity.

func_update_user(id_usr NUMBER, jmeno varchar2, pass Varchar2, rol Varchar2, ucitel_id NUMBER) - Funkce edituje tabulku uživatel.

func_uprav_rozvrh_admin(id_ak NUMBER, kapac NUMBER, misto NUMBER, tydenn VARCHAR2, od_cas NUMBER, do_cas NUMBER, zpusob VARCHAR2, denn VARCHAR2, ucitel_id NUMBER, predmet_id NUMBER, rokk NUMBER, schval NUMBER) - Funkce edituje rozvrh s minimem restrikcí.

func_vloz_pvp2(zkratka Varchar2, zpusob Varchar2, kateg Varchar2, plan_id Number, roc Number) - Funkce vkládá údaje do tabulky předmět_v_planu.

func_vloz_uci(ucitel_id NUMBER, pred_plan NUMBER, rolle Varchar2, hodiny NUMBER) - Funkce vkládá údaje do tabulky UCI.

func_vloz_user(jmeno varchar2, pass Varchar2, rol Varchar2, ucitel_id NUMBER) - Funkce vkládá záznamy do tabulky uživatele.

verify_integrity_rozvrh1(rokk NUMBER, plan_id NUMBER) - Funkce kontroluje integritu rozvrhu při změně plánu.

verify_integrity_uci(rokk NUMBER, plan_id NUMBER) - Funkce kontroluje integritu učitelů při změně plánu.

func_vloz_rozvrh(kapac NUMBER, misto NUMBER, tydenn VARCHAR2, od_cas NUMBER, do_cas NUMBER, zpusob VARCHAR2, denn VARCHAR2, ucitel_id NUMBER,

predmet_id NUMBER, rokk NUMBER, schval NUMBER) - Funkce vkládá záznam do rozvrhu.
func_vloz_rozvrh_admin(kapac NUMBER, misto NUMBER, tydenn VARCHAR2, od_cas NUMBER, do_cas NUMBER, zpusob VARCHAR2, denn VARCHAR2, ucitel_id NUMBER, predmet_id NUMBER, rokk NUMBER) - Funkce vkládá záznam do rozvrhu s minimálními restrikcemi.

2.7 Uložené procedury

proc_delete_predmet(id_pred NUMBER) - Procedura maže předmět_v_planu nebo předmět, pokud má jen jednoho potomka.
proc_delete_vlozi2(ucitel_id NUMBER, pred_plan NUMBER) - Procedura maže vazbu mezi učitelem a předmětem.
proc_update_obor(ob_id NUMBER, odhad NUMBER, jmeno VARCHAR2, fo VARCHAR2) - Procedura upravuje záznam v oboru.
create or replace procedure proc_update_predmet(id_pred NUMBER, nazev Varchar2, zkratka Varchar2, semester Varchar2, zpus Varchar2) - Procedura mění záznam předmětu tak aby u předmětu v plánu nevznikly duplicity.
proc_vloz_obor(odhad NUMBER, jmeno VARCHAR2, fo VARCHAR2) - Procedura vkládá záznam do oboru.
proc_vloz_predmet(nazev Varchar2, zkratka Varchar2, semester Varchar2, zpus Varchar2) - Procedura vkládá záznam do předmětu.

2.8 Spouště

Spouště se používají pro vkládání id do tabulek, u nichž se předpokládá jejich úprava. Také nahrazují cascade constraint při vymazání záznamu.

2.9 Sekvence

Slouží k vytváření identifikátoru pro tabulky

```
CREATE SEQUENCE katedra_seq START WITH 1;
CREATE SEQUENCE fakulta_seq START WITH 1;
CREATE SEQUENCE mistnost_seq START WITH 1;
CREATE SEQUENCE predmet_seq START WITH 1;
CREATE SEQUENCE predmet_plan_seq START WITH 1;
CREATE SEQUENCE rozvrh_seq START WITH 1;
CREATE SEQUENCE plan_seq START WITH 1;
CREATE SEQUENCE obor_seq START WITH 1;
CREATE SEQUENCE uci_seq START WITH 1;
CREATE OR REPLACE SEQUENCE uzivatel_seq START WITH 1;
CREATE OR REPLACE SEQUENCE vyucujici_seq START WITH 1;
```

3 Aplikace

3.1 Použité prostředí

Pro vývoj databáze byly využity programy Datamodeller a SQLdeveloper od společnosti Oracle. Pro vývoj aplikace bylo použito prostředí Netbeans.

3.2 Řízení uživatelských účtů

Aplikace využívá tříúrovňový model rolí. Uživatel může se k aplikaci přihlásit jako Admin, Uživatel a Normal. Administrátor má možnost spravovat všechny záznamy v databázi. Uživatel si může upravovat pouze svůj rozvrh. Normal má možnost pouze prohlížet záznamy.

3.3 Moduly

V aplikaci se nachází celkově 9 modulů

- Vyučující – pro správu vyučujících. Administrátor má zde možnost přidávat, upravovat a mazat záznamy.
- Rozvrh – pro správu rozvrhových aktivit. Uživatelé si zde můžou upravovat svůj rozvrh. Administrátor může měnit rozvrh všem.
- Předměty – pro správu předmětů. Administrátor má zde možnost přidávat, upravovat a mazat záznamy a přidávat vyučující pod předmět.
- Fakulty – pro správu fakult. Administrátor má zde možnost přidávat, upravovat a mazat záznamy.
- Pracoviště – pro správu kateder. Administrátor má zde možnost přidávat, upravovat a mazat záznamy.
- Místnost – pro správu místností. Administrátor má zde možnost přidávat, upravovat a mazat záznamy. Je zde i možnost přidávat data ze souboru.
- Obory – pro správu oborů. Administrátor má zde možnost přidávat, upravovat a mazat záznamy.
- Plány – pro správu plánů. Administrátor má zde možnost přidávat, upravovat a mazat záznamy.
- Uživatelé – pro správu uživatelů. Administrátor má zde možnost přidávat, upravovat a mazat záznamy. Nikdo jiný kromě administrátora nemůže prohlížet tento modul.

3.4 Formuláře

Všechny formuláře pracují na stejném principu. Nabízí možnost upravování a přidávání záznamů do databáze. Při kliknutí na záznam z tabulky nebo seznamu se naplní formulář příslušnými hodnotami a připraví se tlačítko pro úpravu záznamu.

3.5 Orientace ve zdrojovém kódu

V balíčku entity jsou třídy, které kopírují strukturu tabulek.

Dále pomocí tříd s koncovkou Manager aplikace komunikuje přímo s databází za pomoci dříve zmiňovaných entit. Zde se volají všechny SQL dotazy, funkce a procedury. Každá entita má svůj vlastní manager. Za přímé připojení do databáze jsou zodpovědné třídy DatabaseManager a OracleConnector.

Ke grafické části se využívá JavaFX a FXML. Jsou zde dva FXML soubory (FXMLLogin a FXMLMain) společně se svými controllery, které propojují logiku s grafikou. Login slouží k přihlašování uživatele do aplikace a main je hlavní část aplikace, která se zobrazí po úspěšném přihlášení.

Jsou zde i pomocné třídy AlertWindow pro ulehčení zobrazování vyskakovacích oken a Rok pro správné zobrazování školního roku.

Závěr

Ve výsledné aplikaci se nám povedlo vkládat, editovat i odebírat jakékoliv záznamy, které jsou načtené z databáze, rozlišovat práva přihlášených uživatelů a další. Tento program by se dal v budoucnu využít jako šablona na které by šel vybudovat databázový systém vysoké školy.

Přílohy

1. Skripty SQL(soubory zabalené v zipu spolu s dokumentací)
2. Zdrojové kódy aplikace, grafika, apod. vše v jednom zabaleném ZIPU! (Poskytuje Michael Tichý)
3. Samostatně spustitelná aplikace.(Poskytuje Michael Tichý)