

Univerzita Pardubice – Fakulta elektrotechniky a informatiky

Semestrální práce z DAS2 a WWW

Databázová část

Matěj Trakal
8.12.2009

KAPITOLA 1: ZÁKLADNÍ CHARAKTERISTIKA	3
PODKAPITOLA 1: ZVOLENÉ TÉMA	3
PODKAPITOLA 2: URČENÍ FUNKCE DATABÁZE	3
KAPITOLA 2: ZÁKLADNÍ VLASTNOSTI DATABÁZE	4
PODKAPITOLA 1: OŠETŘENÍ VSTUPNÍCH DAT – SQL INJECTION	4
PODKAPITOLA 2: INTEGRITNÍ OMEZENÍ	4
PODKAPITOLA 3: PROPOJENÍ DATABÁZE S WEBOVOU ČÁSTÍ	4
NAVÁZÁNÍ SPOJENÍ S DATABÁZÍ	4
VYKONÁNÍ VYBRANÝCH SQL DOTAZŮ	4
ODPOJENÍ OD DATABÁZE	5
KAPITOLA 3: E-R DIAGRAM	6
KAPITOLA 4: POPIS TABULEK A JEJICH ATRIBUTŮ	8
TZANR	8
TTITULZANR	8
TFORMAT_FILMU	8
TFORMAT_TITULKU	8
TJAZYK	8
TTITULKY	8
TJAZYK_FILMU	8
TROLE	8
TUZIVATEL	8
TTITUL	9
TADRESA	9
TPUJCUJICI	9
TVYPUJCKA	9
TFILMOTEKA	9
KAPITOLA 5: POPIS A SYNTAXE POUŽITÝCH DATABÁZOVÝCH OBJEKTŮ	10
PODKAPITOLA 1: FUNKCE	10
PODKAPITOLA 2: TRIGERY	10
PODKAPITOLA 3: PROCEDURY	11
PODKAPITOLA 4: SEKVENCE	11
PODKAPITOLA 5: POHLEDY	11
PODKAPITOLA 6: DEFAULTNÍ PŘEDPŘIPRAVENÁ DATA PRO FUNGOVÁNÍ WEBOVÉ ČÁSTI	11

Kapitola 1: Základní charakteristika

Podkapitola 1: Zvolené téma

Pro semestrální práci jsem si zvolil tvorbu filmové databáze v plánovaném rozsahu asi 5 tabulek. Bohužel rozsah se výrazně zvětšil postupným zjišťováním návazností a potřeb dělení a zachování základních norem.

Podkapitola 2: Určení funkce databáze

Databáze má sloužit pro jednoho uživatele vlastního domácí filmotéku. Vlastník je hodný a tedy umožňuje svým známým výpůjčku jeho filmů, Je tedy do jisté míry implementována i půjčovna.

Jelikož vlastník využívá více fyzických úložišť filmů a filmy má často i vícekrát v různém formátu a jazycích, musí databáze pojmout i tyto požadavky. Z tohoto hlediska byla ve výsledku databáze navržena jako robustnější a odděluje od sebe jednotlivé tituly a nadále pak jednotlivé filmy umístěné na nějakém médiu.

Kapitola 2: Základní vlastnosti databáze

Podkapitola 1: Ošetření vstupních dat – SQL Injection

Díky využití kvalitního databázového layeru (frameworku) Dibi (<http://dibiphp.com/cs/>) je zajištěno, že provádění sql příkazů nemůže už z principu fungování Dibi být napadeno pomocí SQL Injection. Jediná nevýhoda z jejího nasazení je nutnost mít na serveru PHP ve verzi novější, než 5.2.0 (optimálně 5.3 a vyšší), jelikož využívá pokročilých funkcí PHP pro svůj chod.

Podkapitola 2: Integritní omezení

Podkapitola 3: Propojení databáze s webovou částí

Díky tomu, že jsem využil do jisté míry výhod OOP a databázového layeru Dibi se mi práce s databází značně ztížila, ač přinesla i spoustu výhod. Hlavní nevýhodou je nutnost zapouzdření, dále striktní oddělení datové, aplikační a prezentační vrstvy. Jako výhodou jsem naproti tomu získal bezpečnost a modifikovatelnost aplikace.

Navázání spojení s databází

```
public function __construct() {
    try {
        dibi::connect(array(
            'driver' => 'oracle',
            'database' => 'sql101.upceucebny.cz:1521/oracle10',
            'username' => 'st22312',
            'password' => HESLO, /* globální proměnná načítaná z externího
souboru ve složce nepřístupné z webového prohlížeče */
            'charset' => 'UTF8'
        ));
    } catch (DibiException $e) {
        echo get_class($e), ': ', $e->getMessage(), "\n";
    }
}
```

Vykonání vybraných SQL dotazů

```
/**
 * Ověření uživatele skrz databázi
 * @param <string> $username
 * @param <string> $password
 * @return <string> Array[][]
 */
public function UserAuth($username, $password) {
    $result = null;
    if(dibi::isConnected()) {
        $result= dibi::query("select role as ROLE, nick as USERNAME,
password as PASSWORD from TUZIVATEL left join TROLE on
TROLE.ROLE_ID=TUZIVATEL.ROLE_ID where NICK='".$username.'" and
PASSWORD='".$password.'";");
        return $result->fetchAll();
    }
}

/**
 * Vrací informace o titulech - parsované jako XML
 * @return <Array['XML']> parsované XML
 */
public function MovieParseXML() {
    $result = null;
    $row = array();
    if(dibi::isConnected()) {
```

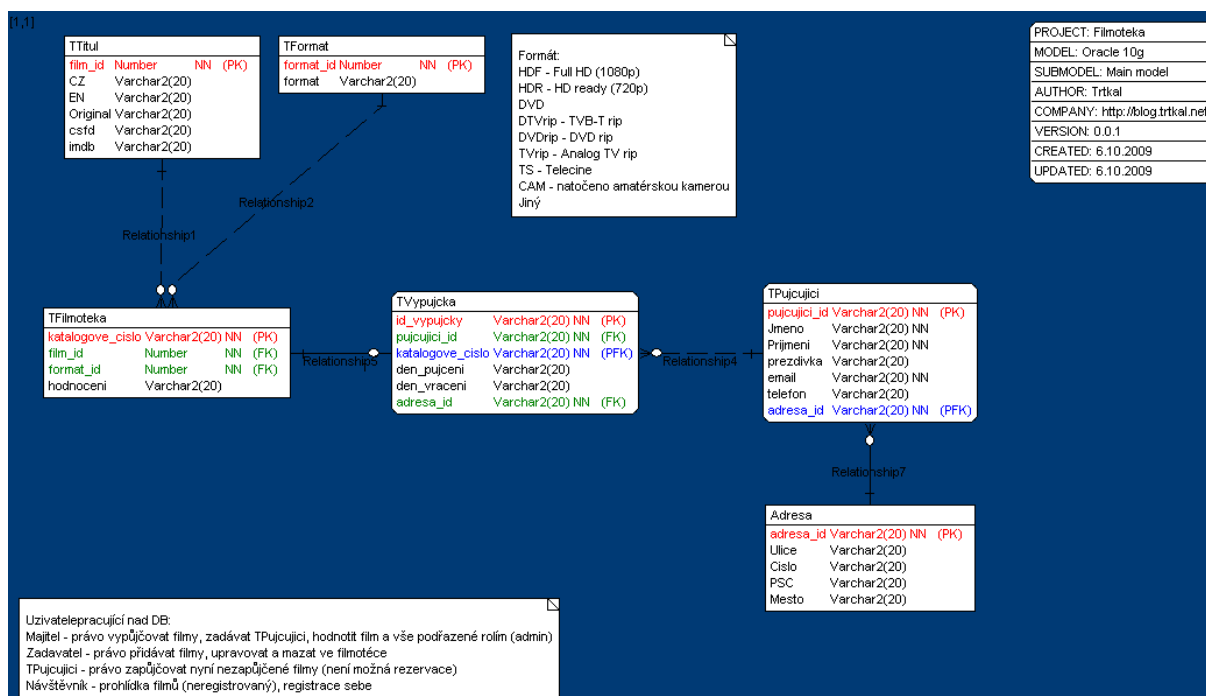
```
        $result= dibi::query("select xmlelement(`titul`,
xmlforest(film_id as `id-filmu`,cz as `nazev-cesky`,en as `nazev-
anglicky`,original as `nazev-originalni`,delka as `delka`,rok_vydani as
`datum-vydani`,csfd as `odkaz-csfd`,imdb as `odkaz-imdb`,popis as `popis-
filmu`) as xml from ttitul order by cz");
        $row = $result->fetchAll();
        return $row;
    }
}
/**
 * Doplnování názvu filmů
 * @param <string> $var
 * @return <Array[][]> vícerozměrné pole
 */
public function Autocomplete($var) {
    $result = null;
    $row = array();
    if(dibi::isConnected()) {
        $result= dibi::query("select film_id as IDFILMU, cz as CZ, en
as EN, original as ORIGINAL, to_char(substr(popis,0,50)) as POPIS from
ttitul where lower(ttitul.cz) like lower('%".$var."%') or lower(ttitul.en)
like lower('%".$var."%') or lower(ttitul.original) like
lower('%".$var."%')");
        $row = $result->fetchAll();
        return $row;
    }
}
```

Odpojení od databáze

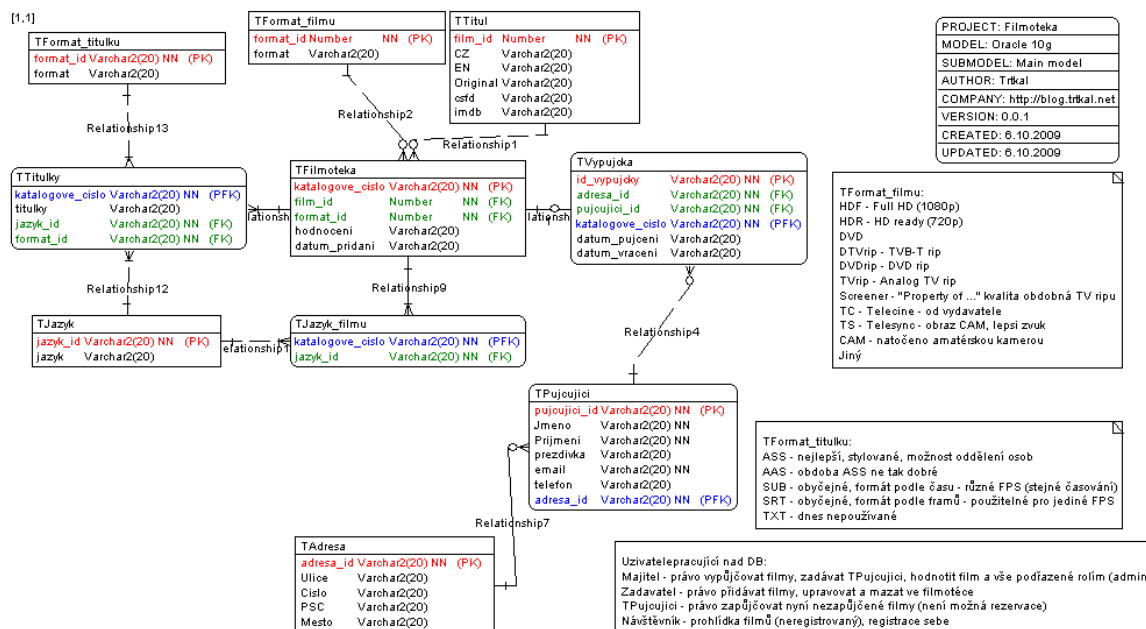
```
public function __destruct() {
    try {
        dibi::disconnect();
    } catch (DibiException $e) {
        echo get_class($e), ': ', $e->getMessage(), "\n";
    }
}
```

Kapitola 3: E-R Diagram

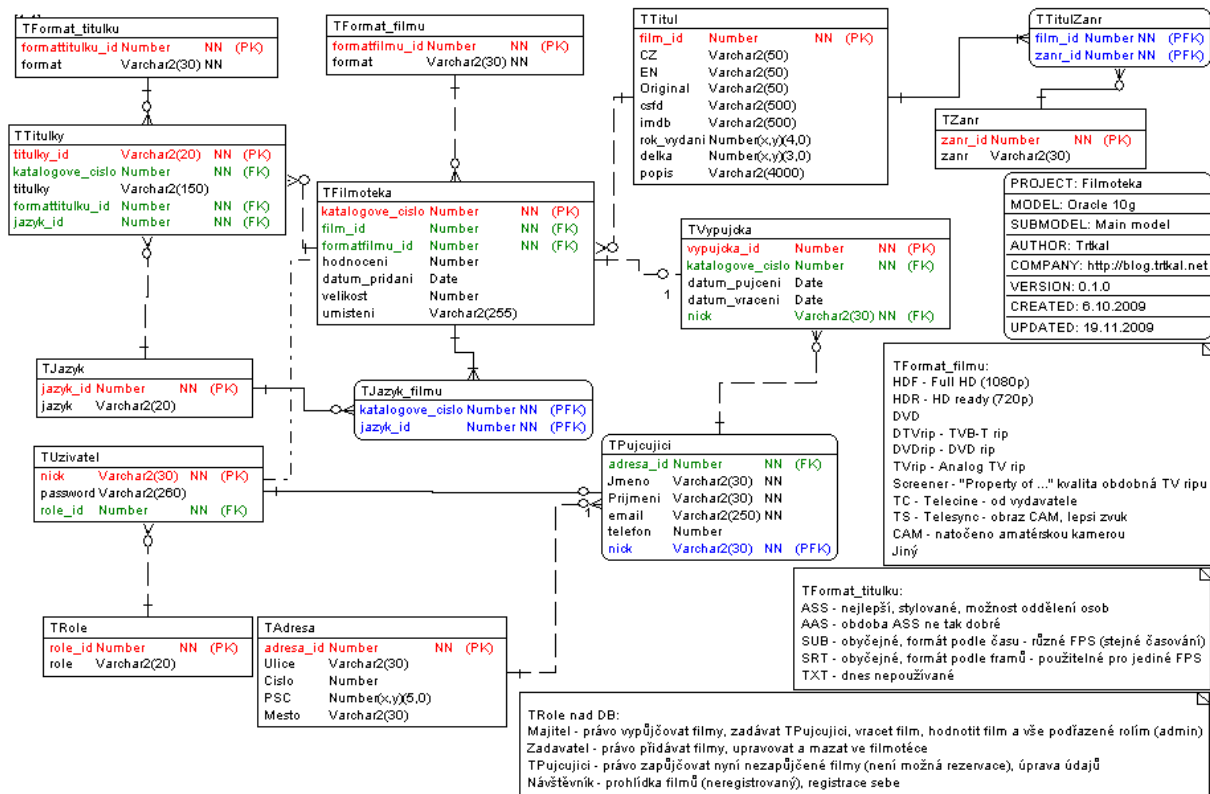
Základní návrh databáze Oracle se postupně vyvinul do značně většího modelu, než se kterým se podle zadání a návrhu počítalo.



Obrázek 1 Předpokládaný rozsah databáze



Obrázek 2 Postup vývoje databázové struktury



Obrázek 3 Finální podoba struktury tabulek

Kapitola 4: Popis tabulek a jejich atributů

TZanr

Název atributu	Vlastnosti a typ	Popis
zanr_id	Number [PK]	ID žánru filmu
Zanr	Number	Textový popis žánru

TTitulZanr

Název atributu	Vlastnosti a typ	Popis
film_id	Number [PFK]	ID žánru filmu
zanr_id	Number [PFK]	ID titulu filmu

TFormat_filmu

Název atributu	Vlastnosti a typ	Popis
formatfilmu_id	Number [PK]	ID formátu filmu
format	Varchar(30) [NN]	Textový popis formátu filmu

TFormat_titulku

Název atributu	Vlastnosti a typ	Popis
formattitulku_id	Number [PK]	ID formátu titulků
format	Varchar(30) [NN]	Textový popis formátu titulků

TJazyk

Název atributu	Vlastnosti a typ	Popis
jazyk_id	Number [PK]	ID jazyku filmu a titulků
jazyk	Varchar(20) [NN]	Textový popis jazyku

TTitulky

Název atributu	Vlastnosti a typ	Popis
titulky_id	Number [PK]	ID titulků filmu
katalogove_cislo	Number [FK]	Katalogové číslo filmu
titulky	Varchar2(150) [NN]	Název titulků
formattitulku_id	Number [FK]	ID formátu titulků
jazyk_id	Number [FK]	ID jazyku titulků

TJazyk_filmu

Název atributu	Vlastnosti a typ	Popis
katalogove_cislo	Number [PFK]	Katalogové číslo filmu
jazyk_id	Number [PFK]	ID jazyku filmu

TRole

Název atributu	Vlastnosti a typ	Popis
role_id	Number [PK]	ID role uživatelů
Role	Varchar(20) [NN]	Textový popis role uživatelů

TUzivatel

Název atributu	Vlastnosti a typ	Popis
Nick	Varchar(30) [PK]	Přezdívká uživatele
password	Varchar(260) [NN]	Heslo uživatele (šifrované pomocí SHA1 + MD5)
role_id	Number [FK]	ID role uživatelů

TTitul

Název atributu	Vlastnosti a typ	Popis
film_id	Number [PK]	ID titulu filmu
CZ	Varchar2(50)	Český název titulu
EN	Varchar2(50)	Anglický název titulu
Original	Varchar2(50) [NN]	Originální název titulu
Csfd	Varchar2(500)	Odkaz na filmovou databázi ČSFD.cz
Imdb	Varchar2(500)	Odkaz na filmovou databázi imdb.com
rok_vydani	Number(4,0) [NN]	Rok vydání titulu
Delka	Number(3,0) [NN]	Délka titulu
Popis	Varchar2(4000)	Krátký popis titulu

TAdresa

Název atributu	Vlastnosti a typ	Popis
adresa_id	Number [PK]	ID adresy půjčujícího
Ulice	Varchar2(3) [NN]	Název ulice půjčujícího
Cislo	Number [NN]	Číslo popisné
PSC	Number(5,0) [NN]	PSC
Město	Varchar2(30)	Město půjčujícího (<i>duplicitní údaj díky PSC</i>)

TPujcujici

Název atributu	Vlastnosti a typ	Popis
adresa_id	Number [FK]	ID adresy půjčujícího
Jmeno	Varchar2(30) [NN]	Jméno půjčujícího
Prijmeni	Varchar2(30) [NN]	Příjmení půjčujícího
email	Varchar2(250) [NN]	Email půjčujícího
telefon	Number	Telefon půjčujícího
Nick	Varchar2(30) [PFK]	Přezdívka uživatele, který patří k půjčujícímu

TVypujcka

Název atributu	Vlastnosti a typ	Popis
vypujcka_id	Number [PK]	ID výpůjčky
katalogove_cislo	Number [FK]	Katalogové číslo filmu
datum_pujceni	Date [NN]	Datum vypůjčení filmu
datum_vraceni	Date	Datum vrácení filmu
Nick	Varchar2(30) [FK]	Přezdívka uživatele, který patří k půjčujícímu

TFilmoteka

Název atributu	Vlastnosti a typ	Popis
katalogove_cislo	Number [PK]	Katalogové číslo filmu
film_id	Number [FK]	ID titulu filmu
formatfilmu_id	Number [FK]	ID formátu filmu
hodnoceni	Number	Hodnocení filmu majitelem (hvězdičky)
datum_pridani	Date [NN]	Datum přidání do filmotéky
velikost	Number	Velikost filmu na disku [MB]
umisteni	Varchar(255)	Volitelný údaj o tom, kde se film nachází

Kapitola 5: Popis a syntaxe použitých databázových objektů

Podkapitola 1: Funkce

Funkce vracející celkový počet nevrácených filmů.

create or replace

```
function f_pocetnevracenychfilmu return number as
  pocet_pujcenyh number;
begin
  select count(*) into pocet_pujcenyh from tpujcujici
  left join tvypujcka on tvypujcka.datum_vraceni is null
  where tvypujcka.nick = tpujcujici.nick
  order by tpujcujici.prijmeni;
  return pocet_pujcenyh;
end f_pocetnevracenychfilmu;
/
```

Vypíše celkový počet filmů v databázi.

```
create or replace function f_pocetTitulu
return number as v_pocet number;
begin
  select count(*) into v_pocet from ttitul;
  return v_pocet;
end;
/
```

Podkapitola 2: Trigery

Spouštěče pro automatické číslování primárních klíčů v tabulkách.

```
create or replace trigger tformattitulku_id before insert on tformat_titulku for each row begin
select sformattitulku_id.nextval into :new.formattitulku_id from dual; end;
create or replace trigger ttitulky_id before insert on ttitulky for each row begin select
stitulky_id.nextval into :new.titulky_id from dual; end;
create or replace trigger tjazyk_id before insert on tjazyk for each row begin select
sjazyk_id.nextval into :new.jazyk_id from dual; end;
create or replace trigger trole_id before insert on trole for each row begin select
srole_id.nextval into :new.role_id from dual; end;
create or replace trigger tadresa_id before insert on tadresa for each row begin select
sadresa_id.nextval into :new.adresa_id from dual; end;
create or replace trigger tvypujcka_id before insert on tvypujcka for each row begin select
svypujcka_id.nextval into :new.vypujcka_id from dual; end;
create or replace trigger tfilm_id before insert on ttitul for each row begin select
sfilm_id.nextval into :new.film_id from dual; end;
create or replace trigger tzanr_id before insert on tzanr for each row begin select
szanr_id.nextval into :new.zanr_id from dual; end;
create or replace trigger tkatalogove_cislo before insert on tfilmoteka for each row begin
select skatalogove_cislo.nextval into :new.katalogove_cislo from dual; end;
```

```
create or replace trigger tformatfilmu_id before insert on tformat_filmu for each row begin
select sformatfilmu_id.nextval into :new.formatfilmu_id from dual; end;
/
```

Podkapitola 3: Procedury

Podkapitola 4: Sekvence

Sekvence pro indexování tabulek – autoincrement.

```
CREATE SEQUENCE sformattitulku_id INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE stitulky_id INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE sjazyk_id INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE srole_id INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE sadresa_id INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE svypujcka_id INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE sfilm_id INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE szanr_id INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE skatalogove_cislo INCREMENT BY 1 START WITH 1 nomaxvalue;
CREATE SEQUENCE sformatfilmu_id INCREMENT BY 1 START WITH 1 nomaxvalue;
/
```

Podkapitola 5: Pohledy

Pohled pro zobrazení počtu titulů v databázi. Vhodnější by bylo použití funkce.

```
CREATE OR REPLACE FORCE VIEW movie_count ("POCET_FILMU") AS SELECT count(film_id) as
pocet_filmu FROM ttitul;
```

Pohled pro zobrazení počtu uživatelů v databázi. Vhodnější by bylo použití funkce,

```
CREATE OR REPLACE FORCE VIEW users_count ("POCET_UZIVATELU") AS SELECT count(nick) as
pocet_uzivatelu FROM tuzivatel;
```

Podkapitola 6: Defaultní předpřipravená data pro fungování webové části

/ Přístupové role */*

```
INSERT INTO "ST22312"."TROLE" (ROLE) VALUES ('Visitor');
INSERT INTO "ST22312"."TROLE" (ROLE) VALUES ('Borrower');
INSERT INTO "ST22312"."TROLE" (ROLE) VALUES ('Administrator');
```

Pro získání hesla je třeba použít skriptu:

```
<?php
$password = "nekriptované heslo";
echo sha1($password) . md5(strlen($password));
?>
```

/ Vytvoření základního administrátora s přístupem do administrace */*

```
INSERT INTO TUZIVATEL (NICK,PASSWORD,ROLE_ID) VALUES ('Admin','kriptované heslo',3);
```

OBRÁZEK 1 PŘEDPOKLÁDANÝ ROZSAH DATABÁZE.....	6
OBRÁZEK 2 POSTUP VÝVOJE DATABÁZOVÉ STRUKTURY	6
OBRÁZEK 3 FINÁLNÍ PODOBA STRUKTURY TABULEK	7