

Databáze pro půjčovnu lodí

KIV/DB1 – Semestrální práce

Student: Jan Pelikán

Osobní číslo: A19B0157P

Email: pelikanj@students.zcu.cz

Datum: 22.12.2020

Jméno a příjmení: Jan Pelikán

Osobní číslo: A19B0157P

Kontaktní e-mail: pelikanj@students.zcu.cz

Orion login: pelikanj

Obor/kombinace: Informatika a výpočetní technika **Rok studia:** 2.

Název práce: Půjčovna lodí a příslušenství pro vodáky

Zákazník si při registraci bude moci vybrat jakou řeku bude chtít sjíždět, typ lodi, kterou si vypůjčí a také si vybere dodatečné příslušenství. Důležité je, aby všechny vybrané věci byly správně předané do tabulek a každý zákazník měl u sebe správné data.

Při řešení mého zadání pravděpodobně použiji (ponechte tu variantu, kterou chcete použít):

- SŘBD MySQL na svém hardware (PC, notebook či tablet)

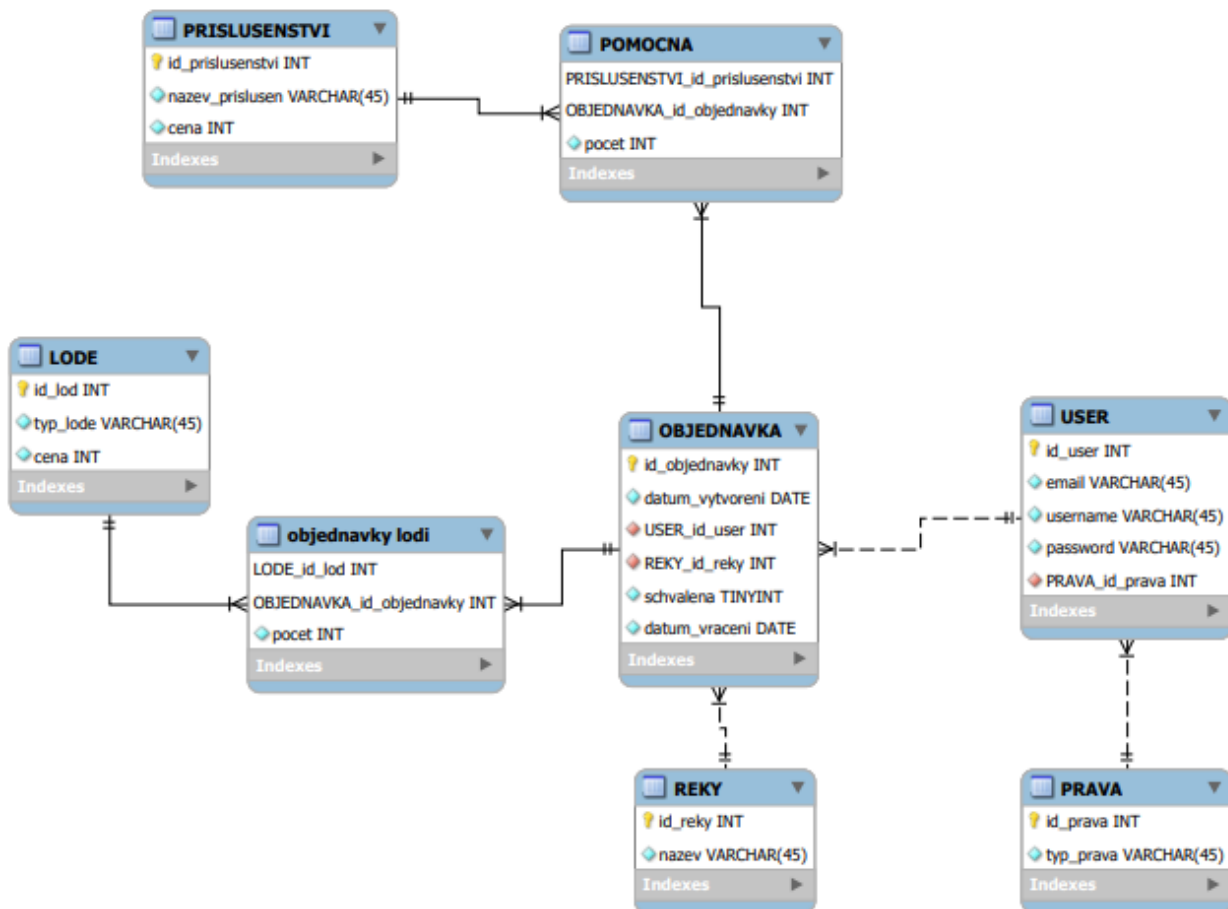
Při řešení mého zadání pravděpodobně budu řešit tyto dva dotazy:

- Dotaz se bude týkat vypsání informací o zákazníkovo objednávce, pokud jich bude mít více, tak vypsání všech objednávek.
- Jako druhý dotaz bych chtěl vypsát jméno a cenu každé lodě a každého příslušenství.

Vyplněný formulář zadání semestrální práce ve formátu PDF a odpovídající návrh relačního modelu exportovaného do formátu PDF odevzdejte na Portál ZČU ve formě archivu ideálně do **29. listopadu 2020**.

Mezní termín odevzdání a předvedení semestrální práce je **29. leden 2021 do 12:00**. Tento termín platí též pro získání zápočtu.

Relační databáze – model



Popis a charakteristika zadání

Cíle semestrální práce bylo vytvořit databázi pro půjčovnu lodí. Zároveň jsem této příležitosti využil a použil semestrální práci k další semestrální práci z KIV/WEB. Jedná se o objednávkový systém, tudíž nejdůležitější tabulkou je objednávka. Každá objednávka má svoje unikátní číslo, datum vypůjčení a vrácení objednaného zboží a ID uživatele, který provedl objednávku. Tabulky prislusenstvi a lode jsou moje zboží, které půjčuji. Mám zde i pomocné tabulky pro realizaci vztahu M:N, protože každá objednávka může mít více lodí/příslušenství. Byla potřeba si u každého uživatele zapamatovat jeho hlavní údaje a také jaké má na stránce pravomoce.

Reprezentativní dotazy

První dotaz vypisující informace o zákaznických objednávkách.

```
SELECT objednavka.id_objednavky as 'ID objednávky',objednavka.datum_vytvoreni as 'Datum  
vyzvednutí',objednavka.USER_id_user as 'ID zákazníka', objednavka.REKY_id_reky as 'ID  
řeky',objednavka.datum_vraceni as 'Datum vrácení' FROM objednavka INNER JOIN user ON objednavka.USER_id_user =  
user.id_user AND user.id_user = 4 ORDER BY objednavka.id_objednavky
```

ID objednávky	Datum vyzvednutí	ID zákazníka	ID řeky	Datum vrácení
2	2022-12-20	4	6	2024-12-20
3	2030-12-20	4	8	2031-12-20

Tento dotaz vypisuje hlavní informace o objednávkách uživatele s ID 4. Podle ID objednávky by se dalo dále zjistit loď a příslušenství u dané objednávky. Dotaz je nad dvěma tabulkami a jedná se o spojení INNER JOIN.

Druhý dotaz měl vypsát jméno a cenu všech lodí a všeho příslušenství.

Lodě:

```
SELECT `typ_lode` as 'Jméno loď','cena`as 'Cena loď' FROM `lode`
```

Jméno loď	Cena loď
Samba 2-m	900
Samba 3-m	1100
Colorado 4-m	1800
Colorado 6-m	2200
Kajak	800

Příslušenství:

```
SELECT `nazev_prislusen` as 'Název příslušenství','cena` as 'Cena' FROM `prislusenstvi`
```

Název příslušenství	Cena
Pumpa k raftu	10
Pádlo	50
Vesta - dospělý	75
Vesta - dítě	50
Barel	100

Tyto dva příkazy jsem využil pro vypsání ceníku půjčovny lodí. Oba příkazy jsou lehké modifikace výběru SELECT.

Komentované scénáře

Odebrání uživatele

Pokud se pokusíme odebrat uživatele (třeba ID = 5), smažou se všechny jeho objednávky a od objednávek se smažou i lodě a příslušenství u dané objednávky.

```
DELETE FROM user WHERE `id_user` = 5
```

Změna ID uživatele

Pokud změníme ID uživatele, kaskádově se změní ID na nové také u všech ostatních výskytů tohoto ID.

```
UPDATE `user` SET `id_user` = '5' WHERE `user`.`id_user` = 4;
```

Přidání nové objednávky

K přidání nové objednávky zde potřebuji znát ID uživatele, který objednávku „založil“, a také správné ID řeky.

```
INSERT INTO `objednavka` (`id_objednavky`, `datum_vytvoreni`, `USER_id_user`, `REKY_id_reky`, `schvalena`, `datum_vraceni`) VALUES ('10', '2020-12-23', '3', '1', '0', '2020-12-25');
```

Zde například objednávku „založil“ uživatel s ID 3 a řeku s ID 1. Dále byla nastavena hodnota „schvalena“ na 0. To znamená ne, neschválena.

Pokud například neznáme ID uživatele nebo ID řeky, můžeme využít vnořený select s podmínkou jména.

```
INSERT INTO objednavka
(objednavka.id_objednavky,objednavka.datum_vytvoreni,objednavka.USER_id_user,objednavka.REKY_id_reky,objednavka.schvalena,objednavka.datum_vraceni)
VALUES ('15','2020-10-25',(SELECT user.id_user FROM user WHERE user.email = 'spravce@email.cz'),'2','0','2020-11-02')
```

Závěr

Datový model jsem vymýšlel pro potřeby mé webové aplikace, tak aby vše mohlo fungovat tak jak jsem chtěl. Pro uvedení do praxe by byla potřeba asi ještě drobná změna, minimálně s přidáním hodnot, které mám k dispozici. (Vlastním jen 2 kajaky, nemůžu jich půjčit více) Celou databázi jsem vytvořil v MySQL Workbench, který mi osobně přišel nejvíce přehledný a používání byla intuitivní. Celkově jsem se semestrální prací spokojen, vyzkoušel jsem si práci s „větší“ databází a byla i užitečná i pro něco jiného než semestrální práce pro KIV/DB1.