



FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Půjčovna motocyklů

KIV/DB1 – Semestrální práce

student:	<i>Lukáš Varga</i>
osobní číslo:	<i>A19B0219P</i>
email:	<i>lvarga@students.zcu.cz</i>
datum:	<i>21. 12. 2021</i>

Obsah

1. Zadání semestrální práce	3
2. Schéma relační databáze	4
3. Popis a charakteristika zadání.....	5
4. Databázové pohledy	6
1. Zákazníci v Praze	6
2. Průměrná cena dle značky	6
5. Testovací scénáře	7
1. Výpůjčka	7
2. Vracení	7
6. Závěr	8

1. Zadání semestrální práce



Databázové systémy I

**zimní semestr
2021/2022**

Lukáš Varga

Orion login: lvarga

Osobní číslo: A19B0219P

Studuji	Vyplňte název nebo zkratku
Fakulta	Fakulta aplikovaných věd
Studijní program	Informatika a výpočetní technika
Studijní obor	Informatika
Rok studia	3.

Název práce: Půjčovna motocyklů

Zaměstnanci půjčovny mají k dispozici několik kusů motocyklů, které mohou půjčit zákazníkům. Tyto stroje mají různý stav a také cenu, za kterou si je lze pronajmout. Půjčovna má více druhů motocyklů, aby si vybral každý zákazník. Každá výpůjčka je evidována, aby zaměstnanci měli přehled o svých motocyklech a lidech, kterým je vypůjčili. V půjčovně mají také detailnější informace o zákaznících jako například kontaktní údaje nebo adresu, aby mohli v budoucnu řešit případnou fakturaci, reklamaci apod.

Zadání společné s předmětem KIV/WEB: Ne

Vyplněný formulář zadání semestrální práce ve formátu PDF odevzdejte na Portál ZČU do **31. října 2021**.

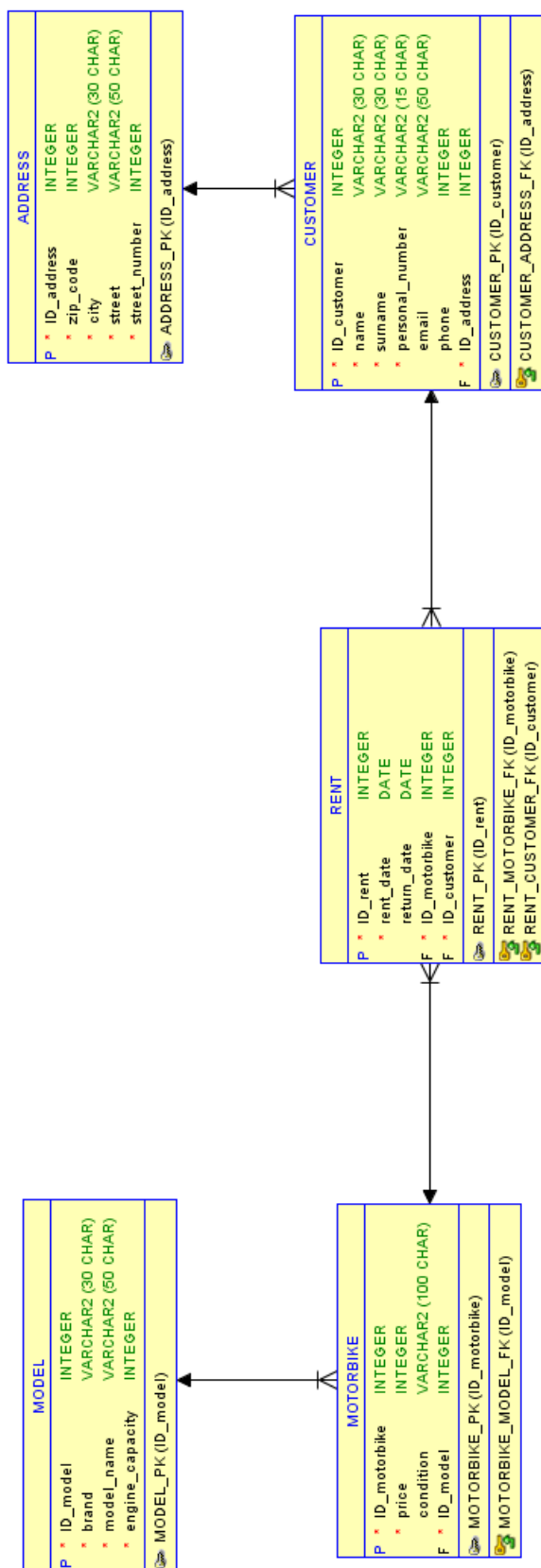
Nezapomeňte na další důležité termíny:

28. listopad 2021 - mezní termín odevzdání schématu relační databáze

23. leden 2022 - mezní termín odevzdání vypracované práce

31. leden 2022 - mezní termín předvedení práce, mezní termín získání zápočtu

2. Schéma relační databáze



3. Popis a charakteristika zadání

V rámci samostatné semestrální práce vytvoří student jednoduchou databázovou aplikaci, jejíž téma si sám zvolí. Rozsah úlohy je požadován minimálně pět tabulek ve schématu relační databáze, přičemž tabulka typu "číselník" není do tohoto počtu započítávána. K vlastní úloze je vypracován referát ve struktuře odpovídající konceptuálnímu modelování.

Při tvorbě své semestrální práce jsem si zvolil téma „Půjčovna motocyklů“. Zaměstnanci půjčovny mají k dispozici několik kusů motocyklů, které mohou půjčit zákazníkům. Různé typy motocyklů jsou v databázovém modelu reprezentované entitou **model**. Půjčovna má více druhů motocyklů, aby si vybral každý zákazník. Každý typ motocyklu existuje v půjčovně formou konkrétního kusu stroje. Tyto stroje mají různý stav a také cenu, za kterou si je lze pronajmout. Fyzickým strojům odpovídá entita **motorbike**. Každá výpůjčka je evidována, aby zaměstnanci měli přehled o svých motocyklech a lidech, kterým je vypůjčili. Toho je docíleno entitou **rent**. V půjčovně mají také detailnější informace o zákaznících díky entitě **customer** jako například kontaktní údaje nebo adresu, aby mohli v budoucnu řešit případnou fakturaci, reklamaci apod. Samotná adresa je uchována v samostatné entitě **address**.

V modelu má každá tabulka vlastní unikátní primární klíč, který je automaticky inkrementován. Jednotlivé tabulky jsou mezi sebou provázané cizími klíči, které zajišťují integritu databáze.

4. Databázové pohledy

1. Zákazníci v Praze

U tohoto pohledu zjišťuji informaci o jménu a příjmení těch zákazníků, kteří mají v adrese uvedené hlavní město Praha.

```
CREATE VIEW prague_customers AS
SELECT name, surname, street, street_number
FROM customer c LEFT JOIN address a
ON c.id_address = a.id_address
WHERE a.city LIKE '%Praha%'
OR a.city LIKE '%Prague';
```

V pohledu zobrazíme jméno, příjmení, ulici a číslo popisné zákazníků. Tabulku zákazníků spojíme s tabulkou adres přes klíč **id_address**. V podmínce zobrazení uvedeme, že město musí být textový řetězec obsahující slovo **Praha** nebo **Prague**. Níže na Obr. 1 můžete vidět výsledek prvního databázového pohledu and testovacími daty.

	name	surname	street	street_number
►	Daniel	Novotný	Václavská	316
	Petr	Novotný	Václavská	316

Obr. 1: VIEW prague_customer

2. Průměrná cena dle značky

Tady zjišťujeme průměrnou cenu za výpůjčku u jednotlivých značek motocyklů.

```
CREATE VIEW avg_price_by_brand (brand, avg_price) AS
SELECT mo.brand, AVG(mtr.price)
FROM model mo LEFT JOIN motorbike mtr
ON mo.id_model = mtr.id_model
GROUP BY mo.brand;
```

Nejprve opět spojíme tabulky modelů a motocyklů přes **JOIN**, kde jako klíč použijeme **id_model**. Následně seřadíme motocykly do skupin podle značek a zobrazíme jednotlivé značky a k nim dané hodnoty průměrné ceny skrze agregovanou funkci **AVG**. Výsledek pohledu můžete vidět na Obr. 2.

	brand	avg_price
►	BMW	5250.0000
	Kawasaki	8750.0000

Obr. 2: VIEW avg_price_by_brand

5. Testovací scénáře

1. Výpůjčka

- Nový uživatel přijde do půjčovny a zadá svou adresu bydliště při registraci

```
INSERT INTO address (zip_code, city, street, street_number)
VALUES ('10800', 'Praha', '17. listopadu', '28');
```
- Zápis detailů nového uživatele s danou adresou do databázového systému

```
INSERT INTO customer (id_customer, name, surname, personal_number, email,
phone, id_address)
VALUES ('5', 'Lukáš', 'Varga', '5856581254', 'lukas.varga128@gmail.com',
'604543693',
(SELECT id_address FROM address
WHERE street = '17. listopadu' AND street_number = '28' LIMIT 1)
);
SELECT * FROM customer;
```
- Uživatel si chce pronajmout motocykl s cenou nižší než 5000

```
SELECT * FROM motorbike WHERE price < '5000';
INSERT INTO rent (rent_date, return_date, id_motorbike, id_customer) VALUES
('2021-01-03', '2021-03-25', 1, 5);
```
- Nalezení všech aktivních výpůjček ke dni 2021-03-10

```
SELECT r.id_rent, c.name, c.surname, c.phone, r.rent_date, r.return_date
FROM customer c INNER JOIN rent r
ON c.id_customer = r.id_customer
AND '2021-03-10' BETWEEN r.rent_date AND r.return_date;
```

2. Vrácení

- Uživatel si nejprve chce prodloužit datum výpůjčky

```
SELECT id_rent, return_date, id_customer FROM rent WHERE id_customer = '5';
UPDATE rent SET return_date = '2021-05-20' WHERE id_rent = '9';
```
- Demonstrace integrity databáze při pokusu o smazání uživatele s aktivní výpůjčkou, která skončí chybou (Error Code: 1451)

```
DELETE FROM customer WHERE id_customer = '5';
```
- Vrácení motocyklu u daného uživatele

```
SELECT COUNT(*) AS rents_of_cust_5 FROM rent WHERE id_customer = '5';
SELECT id_rent, id_customer FROM rent WHERE id_customer = '5';
DELETE FROM rent WHERE id_rent = '9';
```
- Smazání uživatele z DB, který nemá aktuálně žádné aktivní výpůjčky

```
SELECT * FROM rent WHERE id_customer = '5';
DELETE FROM customer WHERE id_customer = '5';
```

6. Závěr

Při vypracování této semestrální práce jsem vytvořil zjednodušený databázový model, který umožňuje administraci půjčovny motocyklů. Ačkoliv je model plně funkční, nehodí se plně do praxe a to kvůli několika důvodům.

Neřeší například půjčení toho samého motocyklů ve stejný čas dvěma různými lidmi. Tuto situaci si musí řešit sám administrátor databáze. Dále by se v praxi měly všechny záznamy v databázi zachovávat, i když už nejsou platné například při navrácení motocyklu. V mém modelu je při navrácení motocyklu daný záznam v tabulce výpůjček odstraněn z tabulky.