

Zápočtový test - Databázové systémy (NDBI025)

2019/2020

Jméno a příjmení (čitelně): Vojtěch Lengál

Login: 54625903

(nečitelný zápis bude automaticky chápán jako chybný)

Úlohy řešte samostatně a nezávisle. Úloha, nápadně podobná jiné, již odevzdané, nemusí být akceptována. Vyhražuji si právo vyžádat si k odevzdané úloze doplnění či vysvětlení, ať už emailem, případně prostřednictvím online komunikace.

1. KONCEPTUÁLNÍ MODELOVÁNÍ

(celkem 30 bodů, min. 15)

1. Navrhněte a slovy popište *jednoduché* vlastní zadání informačního systému s následujícími omezeními: Alespoň 3 třídy, alespoň jeden vztah M:N, alespoň jeden vztah 1:N. Pro popsání informačního modelu vytvořte konceptuální UML model. (10 bodů)
2. Převeďte Vámi vytvořený model do logického relačního modelu včetně referenční integrity. Vypište schémata relací ve formě $R_1(...)$, $R_2(...)$, ... (10 bodů)
3. Převeďte Vámi vytvořený logický relační model na fyzický pomocí CREATE TABLE příkazů. (10 bodů)

2. DOTAZOVÁNÍ V SQL

(celkem 40 bodů, min. 20)

Nad Vámi vytvořenými tabulkami zformulujte 4 rozumné dotazy v přirozeném jazyce s následujícími omezeními, a napište je pomocí SQL dotazu SELECT. (pro každý dotaz jeden SELECT). Samozřejmě je možné použít vnořené dotazy. Pokud uznáte za vhodné, můžete pro zjednodušení dotazů definovat max. 2 pohledy nad daty pomocí příkazu CREATE VIEW.:

- Dotaz 1.** S využitím spojení dvou nebo více tabulek s neprázdnou WHERE podmínkou a tříděním přes alespoň dva sloupce (10 bodů)
- Dotaz 2.** S využitím operátoru NOT EXISTS nebo obdobného (10 bodů)
- Dotaz 3.** Vyhledání objektů z nějaké skupiny s extrémní hodnotou v některém atributu (např. zaměstnanci skladu s nejvyšším platem, rychlíkové spoje do Ostravy s nejnižší cenou, ...) (10 bodů)
- Dotaz 4.** Vyhledání skupin objektů s určitou vlastností (např. oddělení s průměrným platem zaměstnanců přesahujícím 25000 Kč, zázazníci s celkovým objemem nákupů nepřevyšujícím 1000 Kč, ...) (10 bodů)

3. DDL SQL & EMBEDDED SQL

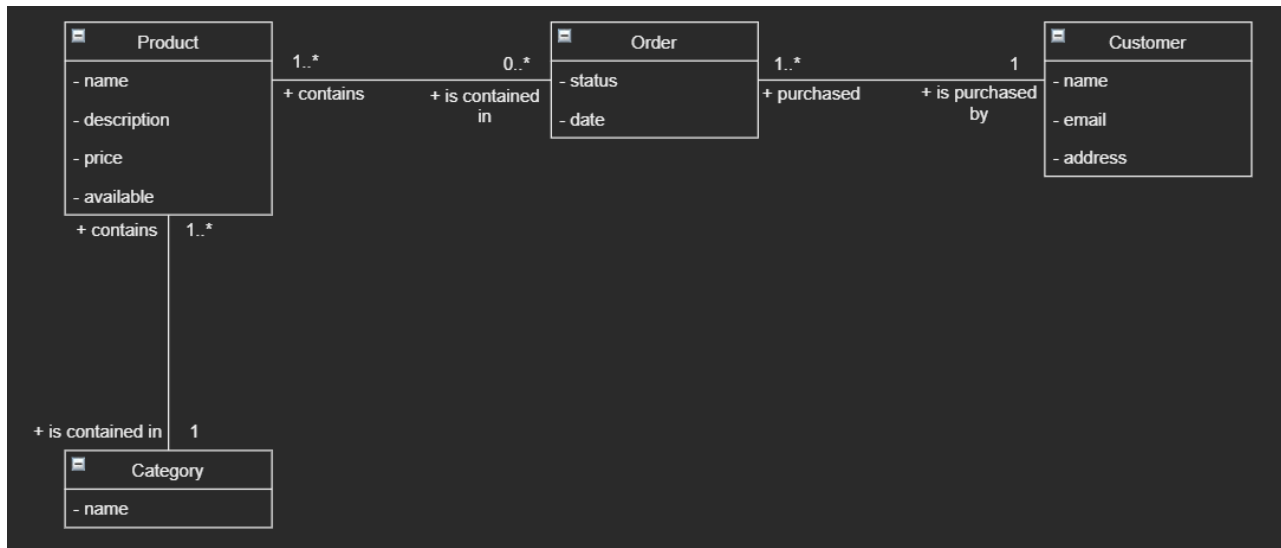
(celkem 30 bodů, min. 15)

1. Ve Vámi vytvořeném relačním schématu vytvořte proceduru, která vloží novou instanci zvolené třídy včetně vazby na konkrétní již existující instanci, se kterou existuje asociace M:N. Všechny potřebné údaje procedura dostane předané prostřednictvím svých parametrů. (15 bodů)
2. Zformulujte slovní zadání buďto vhodné funkce nebo integritního omezení nad Vaším schématem, a vytvořte odpovídající funkci nebo trigger v jazyce T-SQL. (15 bodů)

ŘEŠENÍ

1. Téma: E-Shop

1. E-Shop bude obsahovat 4 entity. Produkt bude reprezentovat nějaký produkt nabízený v obchodě (tzn. v databázi nebude záznam pro každou instanci daného produktu). U každého produktu budu uchovávat jeho název, cenu, popis a dostupnost (dostupný/nedostupný). Produkty budou rozděleny do kategorií, ty budou mít jediný atribut - název (např. elektronika, domácí potřeby, apod.), s tím, že každý produkt bude patřit do právě jedné kategorie. Každá kategorie pak bude obsahovat aspoň 1 produkt. Další entitou je zákazník, u kterého budu uchovávat jméno, email a adresu. Zákazník bude mít v obchodě aspoň 1 objednávku. Objekt objednávky bude obsahovat její stav (objednáno, odesláno, dokončeno, apod.) a datum objednání. V každé objednávce bude aspoň 1 produkt, a každý produkt může být obsažen ve více objednávkách. Tedy mezi entitami *Order* a *Product* bude vazba many-to-many.



UML Model eshopu (pro náčrtek jsem použil aplikaci *draw.io*).

2. Logický relační model: (primární klíče jsou tučně a podtržené, cizí klíče jsou pouze tučně). ProductOrder reprezentuje vazební tabulku mezi produktem a objednávkou.

- Category(**Id**, Name)
- Product(**Id**, Name, Description, Price, Available, **IdCategory**)
IdCategory ⊆ Category.**Id**
- ProductOrder(**IdProduct**, **IdOrder**)
IdProduct ⊆ Product.**Id**
IdOrder ⊆ Order.**Id**
- Order(**Id**, Status, Date, **IdCustomer**)
IdCustomer ⊆ Customer.**Id**
- Customer(**Id**, Name, Email, Address)

3. SQL příkazy na vytvoření tabulek jsou v souboru *create_tables.sql*, v souboru *insert_values.sql* jsou pak příkazy pro vložení testovacích dat.

2. Dotazy (i s vysvětlením) jsou ve složce *queries*.

3.

1. Procedura pro přidání nové objednávky s 1 produktem je v souboru *procedures/addNewOrderWithOneItem.sql*
2. Funkce pro spočtení ceny objednávky je v souboru *functions/getTotalPrice.sql*