

1. Vytvořte diagram ER konceptuálního schématu pro databázi na správu účtů v malé bance: [10]

Každý účet je identifikován dvojicí obsahující vlastní číslo účtu a kód banky. Kromě toho u každého účtu uchováváme jeho aktuální zůstatek a měnu, ve které je veden (např. CZK, EUR atp.). Rozlišujeme dva konkrétní druhy účtů, a to běžné a spořicí. Běžný účet má informaci o frekvenci zasílání výpisů, naopak spořicí účet má nepovinnou roční úrokovou sazbu. Každý účet je vlastněn výhradně jedním klientem. Klient je jednoznačně určen rodným číslem, dále u něj uchováváme nepovinnou emailovou adresu a libovolný počet telefonních čísel. V systému rovněž potřebujeme ukládat informaci, že nový klient mohl být doporučen jiným stávajícím klientem (nejvýše jedním). Mezi účty mohou být realizovány převody peněz. Každý převod je navázán na zdrojový a cílový účet. Obsahuje datum, čas, částku a pořadové číslo odchozí transakce, které je unikátní v rámci daného zdrojového účtu.

Nepřidávejte uměle generované identifikátory, alespoň jednou použijte ternární vztahový typ a specifikujte kritéria pokrytí a překrytí u ISA hierarchie. Namodelujte jen to, co je požadováno.

2. Transformujte ER schéma z předcházejícího příkladu do schématu v relačním modelu. Schémata jednotlivých tabulek popište textovou notací. Uvedte všechny klíče a cizí klíče, nepřidávejte uměle generované identifikátory. Předpokládejte relační model bez NULL hodnot. [10]

Užitečný symbol: \subseteq

3. Předpokládejme následující relační schéma databáze knihovny:

Čtenář (rodné číslo, jméno, příjmení, věk, město)

Knih (signatura, název, autor, žánr)

Výpůjčka (id, datum půjčení, knih, rodné číslo, datum vrácení)

(knih) \subseteq Knih (signatura)

(rodné číslo) \subseteq Čtenář (ročné číslo)

Vytvořte SQL výrazy pro následující dotazy: [15]

- a) Najděte unikátní příjmení čtenářů starších 30 let, kteří si nikdy nepůjčili žádnou knihu patřící do žánru *detektivka* nebo *kuchařka*.
- b) Najděte rodná čísla, jména a příjmení čtenářů, kteří alespoň jednou měli ve stejný okamžik vypůjčeny alespoň dvě knihy najednou.
- c) Pro každé město zjistěte celkový počet realizovaných výpůjček libovolných knih od autora *Karel Čapek*. Omezte se však jen na výpůjčky uskutečněné čtenáři s křestním jménem jiným než *Karel*. Ve výsledku uveďte jen ta města, kde takový počet je roven alespoň 10.
4. Předpokládejme relační schéma z předcházejícího příkladu. Vytvořte výraz pro SQL/XML dotaz, kde pro každou realizovanou výpůjčku (na samostatném řádku výsledné tabulky) uvedete v prvním sloupci její id a ve druhém zkonstruujete XML hodnotu podle následujícího vzoru: [5]

```
<vypujcka pujceno=" datum půjčení " vraceno=" datum vrácení ">
  <knih> název knihy </knih>
  <autor> autor knihy </autor>
</vypujcka>
```

5. Opět předpokládáme stejné relační schéma. Formulujte výrazy v relační algebře pro tyto dotazy: [10]

Čtenář (rodné číslo, jméno, příjmení, věk, město)
Kniha (signatura, název, autor, žánr)
Vypůjčka (id, datum půjčení, kniha, rodné číslo, datum vrácení)
(kniha) \subseteq Kniha (signatura)
(rodné číslo) \subseteq Čtenář (ročné číslo)

a) Signatury a názvy knih od *Karla Čapka*, které nikdy nebyly vypůjčeny ani jedním čtenářem z *Liberce* starším než 30 let.

b) Jména a příjmení čtenářů z jiného města než *Liberce*, kteří si alespoň jednou půjčili každou knihu od *Karla Čapka*.

6. Najděte všechny klíče v relačním schématu s množinou atributů $\{K, L, M, N, O\}$ a množinou funkčních závislostí $\{K \rightarrow MO, L \rightarrow O, KL \rightarrow N, N \rightarrow L, MNO \rightarrow K\}$. Postup komentujte, použijte algoritmus Lucchesi-Osborn. Následně určete normální formu každé jednotlivé závislosti a také celého uvedeného schématu. [10]

7. Cizí klíče a referenční integrita [5]

- a) Cizí klíč musí být v referující a stejně tak i v referenční tabulce klíčem (PRIMARY KEY nebo UNIQUE) a) true false
- b) U daného cizího klíče může být definována nejvýše jedna referenční akce obsluhující událost DELETE, UPDATE nebo INSERT b) true false
- c) Referenční akce popisuje dopad na řádky v referující tabulce, pokud by příslušná aktualizací operace v referenční tabulce způsobila narušení referenční integrity c) true false
- d) Pokud není definována žádná akce, chování je identické jako u varianty NO ACTION d) true false
- e) Výsledek referenčních akcí NO ACTION a RESTRICT může být v případě použití vhodně implementovaných triggerů rozdílný e) true false

8. Databázové pohledy [5]

- a) Pohled je definovaný SELECT dotazem a obsahuje data, která odpovídají výsledku tohoto dotazu v okamžiku vytvoření daného pohledu pomocí CREATE VIEW a) true false
- b) Existující pohledy je možné používat ve FROM klauzulích stejným způsobem jako obyčejné tabulky b) true false
- c) Pohled může být závislý na jiném pohledu, dokonce i tranzitivně, ale nikdy nemůže vzniknout jako spojení (join) více tabulek a/nebo pohledů c) true false
- d) Je-li uvedena klauzule WITH LOCAL CHECK OPTION, provedení aktualizací operace bude umožněno jen tehdy, pokud změna bude v pohledu viditelná (např. u operace INSERT bude vkládaný řádek splňovat případné vyhledávací podmínky v definici pohledu) d) true false
- e) Klauzule WITH CHECK OPTION je nepovinná; je-li však uvedena a není-li vybrána její varianta, za výchozí je považována varianta CASCADED e) true false

9. B⁺-stromový index [5]

- a) B⁺-strom je výškově vyvážený strom, časová složitost základních operací je logaritmická a) true false
- b) Vnitřní uzly (včetně kořenového) mají jinou strukturu než uzly listové, ty totiž obsahují řídí ukazatele na záznamy do primárního souboru b) true false
- c) Vnitřní uzly obsahují řádově i stovky dělicích hodnot, každý uzel je fyzicky uložen jako jeden blok (stránka) indexového souboru c) true false
- d) B⁺-stromový index může být konstruován jen pro vyhledávací klíče, které jsou unikátní d) true false
- e) Uzly v listové úrovni B⁺-stromu jsou navzájem provázány pomocí ukazatelů e) true false

10. **Zotavitelnost (recoverability) rozvrhu (historie) v transakčním zpracování** [5]
- a) Rozvrh (historie) je nezotavitelný (unrecoverable) právě tehdy, když pro každou úspěšně dokončenou transakci T platí, že operaci COMMIT provedla nejdříve tehdy, když skončily všechny další transakce ovlivňující T a) true false
 - b) Pokud každá transakce může číst jen úspěšně potvrzená (commitovaná) data, garantujeme kaskádové rušení transakcí (cascade aborts) b) true false
 - c) Pokud je precedenční graf (precedence graph) acyklický, daný rozvrh (historie) už nezbytně nutně musí být zotavitelný (recoverable) c) true false
 - d) Dvofázový zamykací protokol není schopen garantovat zotavitelnost (recoverability) rozvrhu (historie) ani předcházet kaskádovému rušení transakcí d) true false
 - e) Striktní dvofázový zamykací protokol je schopen garantovat zotavitelnost (recoverability) rozvrhu (historie) stejně jako je schopen předcházet kaskádovému rušení transakcí e) true false

11. **Striktní dvofázový zamykací protokol** [5]
- a) Pokud chce transakce provést operaci `write(A)`, musí nejprve úspěšně získat (nebo už vlastnit) exkluzivní nebo alespoň sdílený zámek pro entitu A a) true false
 - b) Všechny požadované zámky musí být u striktního dvofázového zamykacího protokolu získány najednou hned na úplném začátku transakce b) true false
 - c) Všechny získané zámky jsou uvolněny až na úplném konci transakce, tedy jako inherentní součást provádění operací COMMIT nebo ABORT c) true false
 - d) Striktní dvofázový zamykací protokol garantuje, že precedenční graf je acyklický, a tedy že rozvrh (historie) je konfliktově uspořádatelný (serializovatelný) d) true false
 - e) Striktní dvofázový zamykací protokol je schopen garantovat zotavitelnost rozvrhu (historie), ale už není schopen předcházet kaskádovému rušení transakcí e) true false
- ..

12. **Pojmenujte a vysvětlete 4V charakteristiky pojmu Big Data.** [5]
13. **Vyjmenujte a vysvětlete jednotlivé Coffmanovy podmínky.** S jakou problematikou souvisí a jakým způsobem je můžeme použít? [5]
14. **Vysvětlete použití prioritní fronty v rámci algoritmu vnějšího třídění (N-way External Merge Sort), využití stránek systémové paměti a dopad na velikost běhů.** [5]