

Vytvořte diagram ER konceptuálního schématu pro databázi správy hradů a zámků: [10]

Rozlišujeme dva druhy památek, konkrétně hradů a zámků. Každá památka má svůj název, informaci o století vzniku a GPS souřadnicích (jde o strukturovaný údaj obsahující zeměpisnou šířku a délku). Název památky je unikátní, ale jen v rámci daného kraje. Ten má svůj název, identifikujeme jej pomocí kódu. U hradů dále evidujeme jejich stav (zachovalý, zřícenina, ...) a přístupnost, u zámků naopak styl. Průvodci mohou realizovat prohlídky v různých jazycích, a to vždy na konkrétních památkách. Průvodce samotný je identifikován dvojicí jméno a příjmení, současně ale i rodným číslem. Kromě toho má každý průvodce nepovinné číslo mobilního telefonu a libovolné množství emailových adres. Každý z nabízených jazyků je jednoznačně určen svým dvoupísmenným kódem, uchovávané dále jeho název v češtině a také přímo v daném jazyce. V systému potřebujeme uchovávat i informace o partnerských dvojicích hradů, a to včetně roku, kdy takové partnerství vzniklo.

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. Uveďte všechny klíče a cizí klíče, nepřidávejte uměle generované identifikátory. Předpokládejte relační model bez NULL hodnot. [10]

3. Předpokládejme následující relační schéma databáze jízdenek:

Zákazník (email, jméno, příjmení, kategorie)  
Linka (číslo, odkud, kam, délka jízdy)  
Jízdenka (id, datum a čas odjezdu, linka, email, číslo sedadla, cena)  
(linka)  $\subseteq$  Linka (číslo)  
(email)  $\subseteq$  Zákazník (email)

Vytvořte SQL výrazy pro následující dotazy: [15]

a) Najděte unikátní příjmení zákazníků patřících do kategorie *student*, kteří si nikdy nezakoupili žádnou jízdenku z *Liberce* nebo do *Liberce*.

b) Najděte emailové adresy, jména a příjmení zákazníků, kteří měli alespoň jednou zakoupeny časově se (byť jen částečně) překrývající jízdenky na různých linkách najednou.

c) Pro každou kategorii zákazníků zjistěte celkový počet prodaných jízdenek na libovolné linky jedoucí z *Liberce*. Omezte se však jen na jízdenky zakoupené zákazníky s křestním jménem jiným než *Petr*. Ve výsledku uveďte jen ty kategorie, kde takový počet je roven alespoň *1000*.

4. Předpokládejme relační schéma z předcházejícího příkladu. Vytvořte výraz pro SQL/XML dotaz, kde pro každou prodanou jízdenku (na samostatném řádku výsledné tabulky) uvedete v prvním sloupci její id a ve druhém zkonstruujete XML hodnotu podle následujícího vzoru: [5]

```
<jizdenka odjezd=" datum a čas odjezdu " >  
  <sedadlo> číslo sedadla </sedadlo>  
  <jmeno> jméno zákazníka </jmeno>  
  <prijmeni> příjmení zákazníka </prijmeni>  
</jizdenka>
```

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

Zákazník (email, jméno, příjmení, kategorie)  
Linka (číslo, odkud, kam, délka jízdy)  
Jízdenka (id, datum a čas odjezdu, linka, email, číslo sedadla, cena)  
(linka)  $\subseteq$  Linka (číslo)  
(email)  $\subseteq$  Zákazník (email)

a) Číslo a cílová města linek jedoucích z *Liberce*, na které nikdy nebyla prodána ani jedna jízdenka pro zákazníka s křestním jménem *Petr* v kategorii *student*.

b) Jména a příjmení zákazníků z jiné kategorie než *student*, kteří si alespoň jednou zakoupili jízdenku na každou linku jedoucí z *Liberce* nebo do *Liberce*.

6. Najděte všechny klíče v relačním schématu s množinou atributů  $\{D, E, F, G, H\}$  a množinou funkčních závislostí  $\{FH \rightarrow G, F \rightarrow DE, DE \rightarrow F, H \rightarrow E, G \rightarrow H\}$ . 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 vhodné implementovaných triggerů rozdílný e) true false

8. Tradiční (formální) relační model [5]

- a) Relační databáze obsahuje množinu relací, každá má své relační schéma a vlastní data, tedy množinu záznamů (řádků, n-tic) a) true false
- b) Záznamy (řádky) v relacích nemají definované vzájemné pořadí, relace mohou obsahovat duplicitní záznamy (řádky) b) true false
- c) Atributy mohou nabývat pouze a jenom atomických hodnot, mluvíme proto o tzv. první normální formě c) true false
- d) Klíč je libovolná množina atributů dané relace, která má identifikační schopnost, tj. dokáže unikátně určit (rozlišit) každý záznam (řádek) dané relace d) true false
- e) Libovolná relace může obsahovat i více klíčů, nejvýše však jeden jediný nadklíč e) true false

MCQ-8	a	b	c	d	e
true	true	false	true	false	false

9. Buffer management [5]

- a) Jeden diskový blok (stránka) se namapuje do jedné stránky v systémové paměti, a to i pokud diskový blok a paměťová stránka mají různé velikosti a) true false
- b) Sekvenční přístup k blokům primárního souboru na tradičním magnetickém pevném disku je výrazně rychlejší než přístup náhodný b) true false
- c) U každé stránky v systémové paměti si (mimo jiné) pamatujeme počet transakcí, které k dané stránce aktuálně přistupují c) true false
- d) Pokud už danou stránku v systémové paměti nikdo nevyužívá, je vždy nutné její aktuální obsah zapsat na pevný disk d) true false
- e) Pokud už v bufferu není volné místo, vhodná stránka k uvolnění se najde např. pomocí politiky LRU (least recently used) e) true false

10. Dvofázový zamykací protokol [5]

- a) Dvofázový zamykací protokol pracuje ve dvou fázích, zamykací a odemykací, tyto fáze se mohou v případě potřeby překrývat a) true false
- b) Pokud chce transakce provést operaci `read(A)`, musí nejprve úspěšně získat (nebo už vlastnit) exkluzivní nebo alespoň sdílený zámek pro entitu `A` b) true false
- c) Všechny požadované zámky musí být u dvofázového zamykacího protokolu získány najednou hned na úplném začátku transakce c) true false
- d) Dvofázový zamykací protokol garantuje, že graf čekání (wait-for graph) je acyklický, a tedy že rozvrh (historie) je konfliktově uspořádatelný (serializovatelný) d) true false
- e) Dvofázový zamykací protokol je schopen garantovat zotavitelnost rozvrhu (historie), ale už není schopen předcházet kaskádovému rušení transakcí e) true false

11. Coffmanovy podmínky

[5]

- |  |               |
|--|---------------|
| a) Pokud je splněna alespoň jedna Coffmanova podmínka, zabráníme možnosti vzniku deadlocků (uváznutí)                                | a) true false |
| b) Jedna z Coffmanových podmínek vyžaduje existenci prostředků (např. zámků), které je možné držet v nesdíleném (exkluzivním) režimu | b) true false |
| c) Podmínka resource holding (hold and wait) říká, že je možné žádat o další prostředky i v situaci, kdy nějaké už v držení jsou     | c) true false |
| d) V praktické rovině má smysl v databázových systémech narušit jen podmínku vzájemného vyloučení (mutual exclusion)                 | d) true false |
| e) Podmínku čekání do kruhu (circular wait) lze úspěšně narušit např. pomocí strategií jako wound-wait nebo wait-die                 | e) true false |

12. Co jsou **Armstrongovy axiomy** a k čemu se používají? Vymenujte tyto axiomy, uveďte jejich názvy a určete, které z nich jsou nezávislé a co to znamená. [5]

13. Definujte pojem **nezotavitelný rozvrh** (unrecoverable schedule). Co musí rozvrh splňovat, abychom zcela vyloučili **kaskádové rušení transakcí**? [5]

14. Na jakých aspektech, informacích nebo strukturách závisí výběr vhodných algoritmů pro jednotlivé operace v rámci **vytváření plánů vyhodnocení výrazu SQL dotazu**? [5]