

Bazy danych 2019/2020. Egzamin poprawkowy

1. Sformułuj poniższe zapytania w SQLu, zakładając że w bazie danych jest pojedyncza tabela `Uczony(id, imię, nazwisko, id_promotora)`, w której `id` jest kluczem głównym, a `id_promotora` jest kluczem obcym do siebie samej.
 - (a) (8 punktów) Wypiszuczonych, którzy wypromowali więcejuczonych niż ich promotorzy.
 - (b) (8 punktów) Wypiszuczonych, którzy wypromowali więcejuczonych niż każdy z wypromowanych przez nichuczonych.
 - (c) (4 punkty) Wypiszuczonych malejąco według liczby wszystkich ich „potomków”.
2.
 - (a) (10 punktów) Zaproponuj pojęciowy model danych (wykorzystując diagram EER lub UML) dla platformy crowdfundingowej, tj. serwisu pozwalającego użytkownikom na zgłaszanie projektów do finansowania i finansowanie cudzych projektów (częściowe lub całkowite) poprzez dokonanie płatności kartą. Po zebraniu całej potrzebnej kwoty, platforma ma wysyłać autorowi e-mailem powiadomienie oraz dokonywać przelewu środków na wskazane dla tego projektu konto bankowe. Przy dokonywaniu płatności kartą powinna być możliwość szybkiego wyboru każdej wcześniej użytej karty bez powtórnego wpisywania jej danych. Podobnie, przy tworzeniu projektu powinna być możliwość ponownego wskazania wcześniej użytego konta.
 - (b) (10 punktów) Na podstawie uzyskanego wyżej modelu pojęciowego zaproponuj model logiczny (tj. odpowiedni zestaw tabel). Tabele nie muszą dokładnie odpowiadać diagramowi, ale nie mogą gubić informacji; więzy (UNIQUE, REFERENCES, itp.) pełnią ważną rolę. Jeśli to możliwe bez utraty zależności funkcyjnych, to należy zapewnić, żeby tabele były w postaci BCNF (uzasadnić, że są). Jeśli to niemożliwe (uzasadnić), to powinny być w 3NF (uzasadnić, że są). W szczególności, należy opisać zależności funkcyjne wynikające z modelu i klucze główne.
3.
 - (a) (10 punktów) Rozważmy relację $R(A, B, C)$. Napisz wyrażenie algebry relacji (wariant z numerami kolumn i operacjami $\times, \sigma, \pi, \cup, -$ albo z nazwami kolumn i operacjami $\rho, \bowtie, \sigma, \pi, \cup, -$), które jest równoważne następującemu zapytaniu wyrażonemu w logice pierwszego rzędu:

$$\{a \mid \forall b \forall b' \forall c \forall c' R(a, b, c) \wedge R(a, b', c') \implies R(a, b, c') \wedge R(a, b', c)\}.$$

- (b) (10 punktów) Ile wierszy może zwrócić to zapytanie jeśli wejściowa tabela ma n wierszy i spełnia zależność funkcyjną $A \rightarrow BC$? A jeśli ma n wierszy i spełnia zależność funkcyjną $AB \rightarrow C$?

4. (10 punktów) W bazie danych są następujące tabele: `Województwo(nr, nazwa, stolica)`, `Gmina(nr, nazwa, liczba_mieszkańców, rodzaj, nr_województwa)`.

W takiej bazie danych ma zostać wykonane następujące zapytanie:

```
SELECT G.nazwa,  
       (SELECT nazwa FROM Województwo WHERE nr = G.nr_województwa) nazwa_woj  
FROM Gmina G  
WHERE G.rodzaj = 'wiejska'  
      AND G.liczba_mieszkańców > (SELECT AVG(liczba_mieszkańców)  
                                  FROM Gmina G1  
                                  WHERE G1.nr_województwa = G.nr_województwa)  
ORDER BY nazwa_woj, nazwa;
```

Wskaż jak najlepszy plan realizacji tego zapytania i oszacuj jego koszt mierzony liczbą operacji wejścia/wyjścia (na blokach), biorąc pod uwagę następujące założenia:

- w pamięci operacyjnej jest miejsce na 51 bloków;
 - w jednym bloku mieści się 51 rekordów tabeli `Województwo` albo 35 rekordów tabeli `Gmina`;
 - w tabeli `Województwo` jest 16 rekordów, a w tabeli `Gmina` 2477 rekordów;
 - nie są dostępne żadne indeksy ani inne struktury pomocnicze.
5. (10 punktów) Rozważmy następujące dwie transakcje, A i B:

```
A.1: INSERT into T(name, sal) values ('Alice', 1000);  
A.2: SELECT SUM(sal) FROM T;  
A.3: COMMIT;
```

```
B.1: INSERT into T(name, sal) values ('Bob', 1500);  
B.2: SELECT SUM(sal) INTO $total FROM T;  
B.3: IF $total < 100000 THEN COMMIT ELSE ROLLBACK;
```

Przyjmij, że każda z trzech instrukcji każdej z transakcji wykonywana jest atomowo. Wypisz przeploty tych dwóch transakcji, które są szeregowe niezależnie od pierwotnej zawartości bazy. Wskaż, które z nich są *recoverable* i które są *cascadeless*. Odpowiedzi uzasadnij.

Databases 2019/2020. Second-Try Exam

1. Formulate the following queries in SQL, assuming that the database contains a single table `Scholar(id, first_name, last_name, supervisor_id)`, where `id` is a primary key, and `supervisor_id` is a foreign key referencing the same table.
 - (a) (8 points) Select each scholar that supervised more scholars than his/her supervisor.
 - (b) (8 points) Select each scholar that supervised more scholars than each scholar he/she supervised.
 - (c) (4 points) Order scholars by the total number of their “descendants”.
2.
 - (a) (10 points) Design a conceptual data model (using an EER diagram or a UML diagram) for a crowdfunding platform, that is, a webservice allowing users to propose their own projects for funding and fund (fully or partly) other users’ projects by means of an online card payment. When the total required sum of money is collected, the platform should send an e-mail notification to the author of the project and transfer the money to the bank account indicated for this project. Users should be able to reuse any previously used card without reentering its details. Similarly, when creating a new project, users should be able to reuse any previously used bank account.
 - (b) (10 points) Based on the conceptual model obtained above, propose a logical model (a set of table definitions). The tables need not reflect the conceptual model, but they should not lose information; constraints like UNIQUE, REFERENCES, etc. play an important role. If it is possible without losing functional dependencies, ensure that the tables are in BCNF (justify that they are). If it is impossible (justify), ensure they are in 3NF (justify that they are). In particular, list all functional dependencies describing your logical model and all keys.
3.
 - (a) (10 points) Consider a relation $R(A, B, C)$. Give an expression of relational algebra (using either column numbers and operations $\times, \sigma, \pi, \cup, -$ or column names and operations $\rho, \bowtie, \sigma, \pi, \cup, -$) that is equivalent to the following query expressed in first order logic:

$$\{a \mid \forall b \forall b' \forall c \forall c' R(a, b, c) \wedge R(a, b', c') \implies R(a, b, c') \wedge R(a, b', c)\}.$$

- (b) (10 points) How many tuples may this query return if the input relation has n tuples and satisfies the functional dependency $A \rightarrow BC$? And if the input relation has n tuples and satisfies the functional dependency $AB \rightarrow C$?

4. (10 points) The database stores the following tables: `Voivodeship(nr, name, capital)`, `County(nr, name, population, type, voivodeship_nr)`.

The following query was issued:

```
SELECT C.name,  
       (SELECT name FROM Voivodeship WHERE nr = C.voivodeship_nr) voi_name  
FROM County C  
WHERE C.type = 'rural'  
      AND C.population > (SELECT AVG(population)  
                          FROM County C1  
                          WHERE C1.voivodeship_nr = C.voivodeship_nr)  
ORDER BY voi_name, name;
```

Design an execution plan for this query involving as few input/output block operations as possible (estimate their number), based on the following assumptions:

- RAM can store 51 blocks;
 - one block stores 51 records of table `Voivodeship` or 35 records of table `County`;
 - `Voivodeship` has 16 records, and `County` has 2477 records;
 - no indexes or other auxiliary data structures are available.
5. (10 points) Consider the following two transactions A and B:

```
A.1: INSERT into T(name, sal) values ('Alice', 1000);  
A.2: SELECT SUM(sal) FROM T;  
A.3: COMMIT;
```

```
B.1: INSERT into T(name, sal) values ('Bob', 1500);  
B.2: SELECT SUM(sal) INTO $total FROM T;  
B.3: IF $total < 100000 THEN COMMIT ELSE ROLLBACK;
```

Assume that all three instructions of each transaction are executed atomically. List all serializable schedules of these two transactions. Mark the *recoverable* ones and the *cascadeless* ones.