Bazy Danych 2022
Lista zadań 6

Z29. (2 pkt.) Nadleśnictwo posiada bazę danych umożliwiającą sporządzanie oraz przechowywanie tekstowych opisów poszczególnych sektorów lasu. Każdy pracownik leśny podczas obchodu lasu dysponuje aplikacją mobilną działającą w następujący sposób: w razie potrzeby edycji opisu bieżacego sektora aplikacja łączy się z bazą danych, przesyła lokalizację i pobiera krotkę dotyczącą tego sektora (w ramach pojedynczej transakcji typu READ ONLY). Pracownik może edytować pobrany opis, a po zakończeniu edycji uruchamiana jest nowa transakcja, która zapisuje wprowadzone zmiany w bazie.

Od pewnego czasu wprowadzono obowiązek aby każdy obszar leśny był regularnie wizytowany przez komisję składającą się ze specjalistów ds. ochrony przyrody, ds. gospodarki leśnej oraz innych. Specjaliści często zgłaszają, że wprowadzane przez nich modyfikacje opisów sektorów znikają mimo, że nikt nie przyznaje się do usuwania czegokolwiek. Prowadzi to do częstych nieporozumień i powstawania teorii spiskowych (np. notatka ekologa o gnieździe rzadkiego ptaka znika, a w jej miejsce pojawia się wycena drewna możliwego do pozyskania w przypadku wycinki).

Wiadomo, że system bazy danych stosuje protokół gwarantujący szeregowalność. Co w takim razie może być przyczyną problemu? W jaki sposób można się go pozbyć? Zaproponuj rozwiązanie, które pozwoli na maksymalną współbieżność.

Z30. (1 pkt) Stosujemy protokół z blokadami dzielonymi `S` i wyłącznymi `X`. Jeśli nie podano inaczej to blokady zakładane są na poszczególnych krotkach z bieżącego stanu bazy danych. Jakiego rodzaju anomalie ze zbioru *dirty read, nonrepeatable read, phantom read* mogą wystąpić przy poniższych protokołach zarządzania transakcjami?

1. Wszystkie blokady dzielone i wyłączne zwalniane są natychmiast po wykonaniu danej operacji odczytu lub zapisu na zablokowanym obiekcie.
2. Wszystkie blokady dzielone są zwalniane natychmiast po odczytaniu danego obiektu, blokady wyłączne utrzymywane są do zakończenia transakcji.
3. Wszystkie blokady dzielone oraz wyłączne są zwalniane dopiero po zakończeniu transakcji.
4. Wszystkie blokady dzielone oraz wyłączne są zwalniane dopiero po zakończeniu transakcji. Dodatkowo, jeśli zapytanie używa klauzuli WHERE z warunkiem z zakresem (*range query*) to blokowane są operacje dodawania, usuwania i modyfikacji wszystkich krotek spełniających ten warunek.

Z31. (1 pkt + po 0.5 pkt za podpunkt) Załóżmy, że na uczelni używana jest baza PostgreSQL. Przypomnij w jaki sposób każdy z poziomów izolacji jest implementowany w PostgreSQL. (Z31 a) 1 pkt.) Zastanów się, który poziom izolacji

najlepiej zastosować w następujących przypadkach. Co może pójść nie tak jeśli zostanie wybrany niższy poziom niż zalecany przez Ciebie? Czy Twoja decyzja będzie inna w przypadku systemu, który implementuje poziomy izolacji stosując blokady takie jak w zadaniu 2?

b) Zapytanie wylicza sumaryczną liczbę przepracowanych godzin dla wszystkich pracowników UWr, a następnie sumaryczną liczbę przepracowanych godzin przez każdy wydział. Wyniki następnie są porównywane w celu sprawdzenia poprawności rozliczeń.

```
BEGIN;
    SELECT SUM(hours) FROM employees;
    SELECT SUM(hours) FROM departments;
COMMIT;
```

c) Chcemy zapewnić, że podczas przydzielania zajęć pracownikowi nie zostanie zlecone więcej godzin niż wynosi jego pensum. Zakładamy, że każdy pracownik prowadzi zajęcia na wielu kierunkach, a każdy kierunek ma swojego dyrektora dydaktycznego i wszyscy ci dyrektorzy układają plan w nocy przed deadlinem.

```
BEGIN;
  UPDATE sheet SET hours = hours + $newhours
  WHERE teacher = $emp;
  SELECT SUM(hours) INTO asum FROM sheet;
  IF (asum <= 210)
    COMMIT;
  ELSE
    ROLLBACK;
```

Przy okazji, jak w naturalny sposób powyższy warunek wymusić nie przejmując się poziomami izolacji?

d) Chcemy zapewnić aby żaden student nie zapisał się na więcej niż 2 terminy z egzaminu z podanego przedmiotu.

```
BEGIN;
  SELECT COUNT(*) INTO anumber FROM registration
     WHERE student = $student
       AND course = $course;
  IF (anumber <= 1)
    INSERT INTO registration VALUES ($student, $course, $newslot);
COMMIT
```

Z32. (1 pkt) Przeanalizuj co się stanie w PostgreSQL w przypadku następującego harmonogramu na poziomie REPEATABLE READS. Sprawdź czy dojdzie do jakichś anomalii. Kolejno dzieją się następujące zdarzenia:

1. Transakcja 1 rozpoczyna się i czyta przydział (tj. liczbę przypisanych godzin) z wszystkich przedmiotów pracownika A. Na tej podstawie zmienia liczbę godzin przypisaną temu pracownikowi z AiSD.

2. Transakcja 2 rozpoczyna się i zmienia liczbę godzin przypisaną pracownikowi A z Matematyki Dyskretnej. Następnie jest zatwierdzana.
3. Transakcja 3 rozpoczyna się i czyta jakieś dane dotyczące innych pracowników niż A. Potem zatwierdzana jest transakcja 1. Następnie transakcja 3 czyta wszystkie przydziały pracownika A.

Z33. (1 pkt) Rozważmy następujące zapytanie inkramentujące liczbę kliknięć w link na pewnej stronie.

```
UPDATE links SET clicked = clicked + 1 WHERE url = $param
```

Załóżmy, że strona jest bardzo popularna, a harmonogram jest następujący:

Transakcja 1	Transakcja 2
czyta `clicked` i widzi `clicked` = 44	
	czyta `clicked` i widzi `clicked` = 44
	oblicza `clicked` = 45
oblicza `clicked` = 45	
	próbuje `commit`
próbuje `commit`	

Jaki będzie efekt w PostgreSQL jeśli poziom izolacji będzie ustawiony jako READ COMMITTED, a jaki gdy SERIALIZABLE ? Czy możliwe jest, że obie transakcje zostaną zatwierdzone, a jedno z kliknięć zostanie pominięte?

Z34. (1 pkt.) Mamy bazę z dwiema tabelami:

```
Bombiarze(id INT PRIMARY KEY, stopien INT),
Agenci(id INT PRIMARY KEY, bombiarz INT).
```

Bombiarze są podejrzewani o podkładanie bomb w autobusach, agenci ich śledzą, monitorują stopień zagrożenia (atrybut `stopien`) i w przypadku zebrania odpowiednich dowodów zatrzymują do wyjaśnienia. Można założyć, że jesli bombiarz ma przydzielonego agenta to nie będzie w stanie podłożyć bomby. Agenci są dobrze zakonspirowani i czasem ich atrybut `bombiarz` ma wartość NULL - nie wiemy wtedy, którego bombiarza śledzi agent (a nawet czy w ogóle jakiegoś śledzi).

Czy poniższe zapytanie poprawnie wypisuje listę bombiarzy, którzy nie mają przydzielonego agenta?

```
SELECT id FROM Bombiarze
WHERE id NOT IN (SELECT bombiarz FROM Agenci)
```

Odpowiedź uzasadnij. Jeśli uważasz zapytanie za niepoprawne to je popraw. Przetestuj jak szybko działa Twoje zapytanie (dla 10k Bombiarzy i Agentów) i popraw je jeśli jego działanie trwa zbyt długo.