

**ĐỀ KIỂM TRA THỰC HÀNH**

**HỌC KỲ 1A NĂM HỌC 2022 - 2023**

Tên môn thi **: CƠ SỞ DỮ LIỆU NÂNG CAO**

MSMH **: IT301DV01**

Thời lượng thi **: 60 phút** (không kể thời gian phát đề)

**Ghi chú**: Sinh viên làm bài trực tiếp trên tập tin này.

**MSS**V:

**Họ tên**:

[Đổi tên tập tin: MSSV\_HoTen.docx]

**NỘI DUNG ĐỀ THI**

**Câu 1:** (4 điểm) Các giao dịch Ti và lịch S:

T1: R(Z), R(Y), C1 T2: R(Y), W(Y), C2

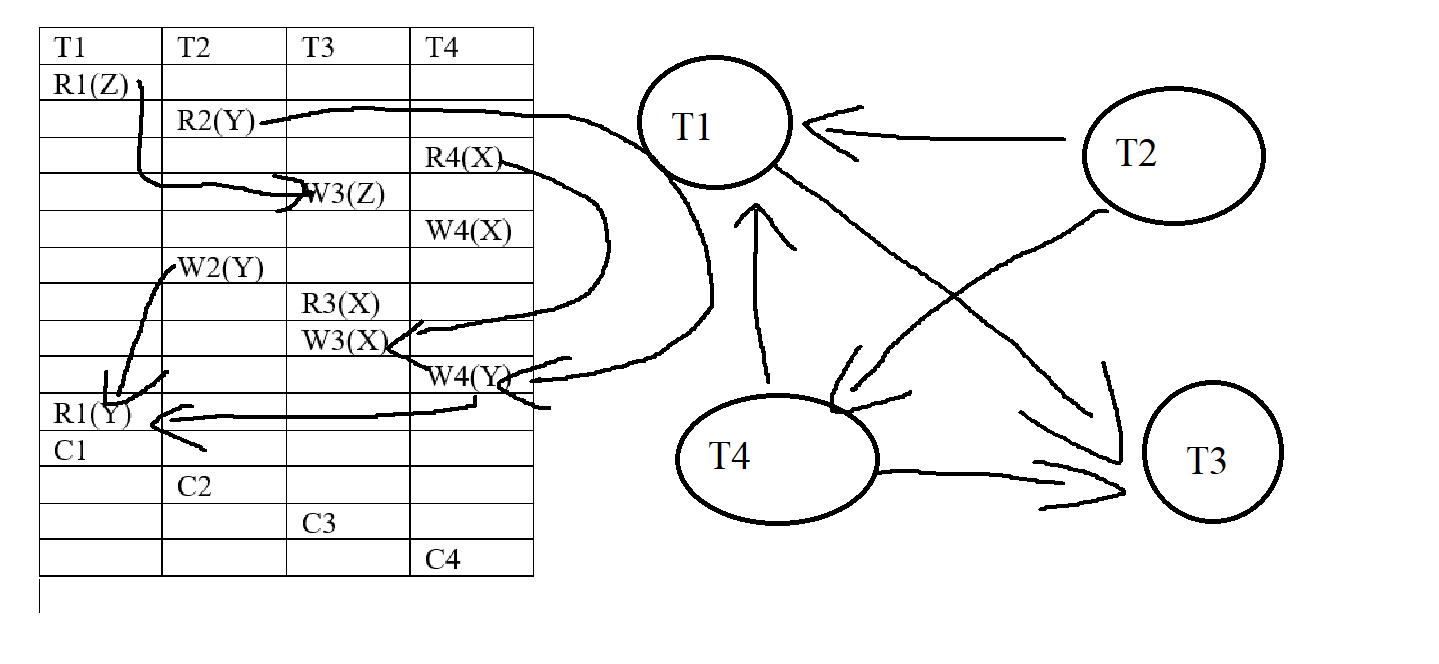
T3: W(Z), R(X), W(X), C3 T4: R(X), W(X), W(Y), C4

S: R1(Z), R2(Y), R4(X), W3(Z), W4(X), W2(Y), R3(X), W3(X), W4(Y), R1(Y), C1, C2, C3, C4

1. Lịch S có khả tuần tự xung đột (conflict serializable) không? Giải thích và cho biết có tương đương với các lịch tuần tự (serial schedule) nào không? (2 điểm)

**Trả lời:**

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | T2 | T3 | T4 |
| R1(Z) |  |  |  |
|  | R2(Y) |  |  |
|  |  |  | R4(X) |
|  |  | W3(Z) |  |
|  |  |  | W4(X) |
|  | W2(Y) |  |  |
|  |  | R3(X) |  |
|  |  | W3(X) |  |
|  |  |  | W4(Y) |
| R1(Y) |  |  |  |
| C1 |  |  |  |
|  | C2 |  |  |
|  |  | C3 |  |
|  |  |  | C4 |



**Lịch S có khả tuần tự xung đột (conflict serializable) vì không có cycle.**

1. Hãy viết lại lịch này sao cho

* Khả phục hồi (recoverable) (1 điểm)
* Nghiêm ngặt (strick). (1 điểm)

**Trả lời:**

**Recoverable: R1(Z), R2(Y), R4(X), W3(Z), W4(X), W2(Y), R3(X), W3(X), W4(Y), R1(Y), C4,C3,C2,C1**

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | T2 | T3 | T4 |
| R1(Z) |  |  |  |
|  | R2(Y) |  |  |
|  |  |  | R4(X) |
|  |  | W3(Z) |  |
|  |  |  | W4(X) |
|  | W2(Y) |  |  |
|  |  | R3(X) |  |
|  |  | W3(X) |  |
|  |  |  | W4(Y) |
| R1(Y) |  |  |  |
|  |  |  | C4 |
|  |  | C3 |  |
|  | C2 |  |  |
| C1 |  |  |  |

**Strick:**

**R1(Z),R1(Y),C1, R2(Y), W2(Y),C2,W3(Z),R3(X), W3(X),C3, W4(Y),R4(X), W4(X), C4**

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | T2 | T3 | T4 |
| R1(Z) |  |  |  |
| R1(Y) |  |  |  |
| C1 |  |  |  |
|  | R2(Y) |  |  |
|  | W2(Y) |  |  |
|  | C2 |  |  |
|  |  | W3(Z) |  |
|  |  | R3(X) |  |
|  |  | W3(X) |  |
|  |  | C3 |  |
|  |  |  | W4(X) |
|  |  |  | R4(X) |
|  |  |  | W4(X) |
|  |  |  | C4 |

**Câu 2:** (2 điểm)

- Trình bày thuật toán timestamp ordering. (1 điểm)

**Trả lời:**

Giao dịch T muốn thực hiện Write\_item(X):

1). Nếu read\_Ts(X) >TS(T) hoặc write\_TS(X) > TS(T) thì bỏ qua và rollback.

2). Nếu điều kiện 1 không thõa mãn, cho phép T write\_item(X) và gán lại:

Write\_TS(X)=TS(T).

Giao dịch T muốn thực hiện read\_item(X):

1). Nếu Write\_TS(X) > TS(T) thì bỏ qua rolback.

2). Nếu điều kiện 1 không thõa mãn, cho phép T read\_item(X) và gán lại:

Read\_TS(X) = MAX {READ\_TS(X),TS(T)}.

- Cho lịch S: R2(x), R3(x), R2(y), W3(x), R1(y), R4(z), R1(x), W4(x), W1(z)

và các nhãn thời gian T1 = 400, T2 = 200, T3 = 300, T4 = 500

Thực hiện lịch S với thuật toán timestamp ordering (1 điểm)

**Trả lời:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | T4 | R\_TS(X) | W\_TS(X) | R\_TS(Y) | W\_TS(Y) | R\_TS(Z) | W\_TS(Z) |
| 400 | 200 | 300 | 500 |  |  |  |  |  |  |
|  | R2(X) |  |  | 200 | 0 | 0 | 0 | 0 | 0 |
|  |  | R3(X) |  | 300 | 0 | 0 | 0 | 0 | 0 |
|  | R2(Y) |  |  | 400 | 0 | 0 | 0 | 0 | 0 |
|  |  | W3(X) |  | 400 | 300 | 0 | 0 | 0 | 0 |
| R1(Y) |  |  |  | 400 | 300 | 400 | 0 | 0 | 0 |
|  |  |  | R4(Z) | 400 | 300 | 400 | 0 | 500 | 0 |
| R1(X) |  |  |  | 400 | 300 | 400 | 0 | 500 | 0 |
|  |  |  | W4(X) | 400 | 500 | 400 | 0 | 500 | 0 |
| W1(Z) |  |  |  |  | **Abort and Rollback** | | | | |

**Câu 3:** (4 điểm)

Cho lược đồ quan hệ

KháchSạn(**MãKS**, Tên KS, ĐịaChỉKS)

Phòng(**SốPhòng, MãKS**, LoạiPhòng, Giá)

ĐặtPhòng(MãKS, MãKH, TừNgày, ĐếnNgày, SốPhòng)

KháchHàng(**MãKH**, TênKH, ĐịaChỉKH)

1. Giả sử các quan hệ được tạo bởi User **admin** là người muốn cấp các quyền cho tài khoản User quanly. User **quanly** có thể đọc(read) hay ghi(write) trên các quan hệ, và có thể cấp bất kỳ những quyền này cho user khác.

**GRANT READ, WRITE ON admin TO quanly WITH GRANT OPTION**

1. User **quanly** cấp tất cả các quyền trên quan hệ DatPhong cho user **Hoa** (tiếp tân), và các quan hệ còn lại chỉ được phép đọc.

**GRANT ALL ON DatPhong TO Hoa WITH GRANT READ**

1. User **Hoa** cấp quyền đọc dữ liệu trên quan hệ bất kỳ, ngoại trừ quan hệ KhachSan cho user **Huong** (quản lý buồng)

**GRANT READ ON Hoa EXCEPT DatPhong TO Huong**

1. User **admin** cấp quyền đọc các thông tin: tên khách hàng, tên khách sạn, loại phòng, ngày đến, số ngày lưu trú, giá cho user **Lan** (thống kê).

**GRANT ALL ON KhanhSan, Phong, DatPhong, KhangHang TO Lan**

1. Khi user **admin** thu hồi quyền đọc trên tất cả các quan hệ đã cấp cho user **quanly** và user **Lan** thì có vấn đề gì xảy ra? Viết lệnh và giải thích.

**REVOKE ALL ON quanly, Lan FROM admin**

Viết các câu lệnh SQL để gán các quyền này cho các câu a, b, c, d. Bên cạnh đó hãy sử dụng các bảng View một cách thích hợp.

**Trả lời:**

**HẾT**