Bài Kiểm Tra Giữa Kì

**Bài 1: Trình bày các tính chất ACID của giao tác trong cơ sở dữ liệu**

Các tính chất ACID của giao tác trong cơ sở dữ liệu :

* **Tính nguyên tố (Atomicity):** toàn bộ các thao tác trong cùng một giao tác đều được thực hiện hoặc không một thao tác nào trong chúng được thực hiện. Một giao tác là một đơn vị công việc không thể phân chia
* **Tính nhất quán (Consistency):** một giao tác phải chuyển CSDL từ một trạng thái nhất quán này sang một trạng thái nhất quán khác
* **Tính cô lập (Isolation):** Các giao tác phải được thực hiện một cách độc lập với nhau. Nói một cách khác những tác động của giao tác này sẽ không thể thấy được đối với những giao tác khác, khi giao tác này chưa thực hiện xong hoàn toàn
* **Tính bền vững (Durability):** những thay đổi của CSDL do tác động của một giao tác thành công là bền vững, không bị mất đi

**Mở rộng :** chúng ta sẽ để cập đến kiến trúc của một hệ CSDL, ở trong kiến trúc này bộ quản lí giao tác (transaction manager) thay mặt các chương trình ứng dụng điều phối các giao tác. Nó liên hệ với bộ lập lịch, nơi có trách nhiệm thực hiện một chiến lược điều khiển tương tranh. Nếu có một sự cố trong quá trình thực hiện giao tác thì tính nhất quán của CSDL cần được bảo toàn, đây là trách nhiệm của bộ quản lí khôi phục dữ liệu. Bộ quản lí vùng nhớ đệm có nhiệm vụ chuyển giao dữ liệu qua lại giữ các thiết bị nhớ thứ cấp và bộ nhớ chính.

**Bài 2:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| **R1** | **X** | **X** | **X** | **O** | **O** |
| **R2** | **X** | **O** | **O** | **X** | **X** |

* A -> BC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| **R1** | **X** | **X** | **X** | **O** | **O** |
| **R2** | **X** | **X** | **X** | **X** | **X** |

* B -> D

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| **R1** | **X** | **X** | **X** | **X** | **O** |
| **R2** | **X** | **X** | **X** | **X** | **X** |

* CD -> E

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| **R1** | **X** | **X** | **X** | **X** | **X** |
| **R2** | **X** | **X** | **X** | **X** | **X** |

* E -> A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| **R1** | **X** | **X** | **X** | **X** | **X** |
| **R2** | **X** | **X** | **X** | **X** | **X** |

**Bài 3:**

**Tạo Bảng và lược đồ quan hệ**

Create database kthk;

use kthk;

CREATE TABLE Person (

pid int not null,

name varchar(255),

street varchar(255),

postcode varchar(255),

primary key (pid)

);

CREATE TABLE Car (

cid int not null,

year int,

model varchar(255),

primary key (cid)

);

CREATE TABLE OwnedBy (

pid int not null,

cid int not null,

primary key (pid, cid)

);

CREATE TABLE AccidentReport (

rid int not null,

damage int,

details varchar(255),

primary key (rid)

);

CREATE TABLE ParticipatedIn (

pid int not null,

rid int not null,

cid int not null,

primary key (pid, rid, cid)

);

ALTER TABLE OwnedBy ADD FOREIGN KEY(pid) REFERENCES Person(pid);

ALTER TABLE OwnedBy ADD FOREIGN KEY(cid) REFERENCES Car(cid);

ALTER TABLE ParticipatedIn ADD FOREIGN KEY(cid) REFERENCES OwnedBy(cid);

ALTER TABLE ParticipatedIn ADD FOREIGN KEY(rid) REFERENCES AccidentReport(rid);

-- 1. Viết một truy vấn SQL trả về mã pid của những người lái xe trong ít nhất một tai nạn, mà không có sự trùng lặp

Select distinct pid

From ParticipatedIn

-- 2. Viết một truy vấn SQL trả về tất cả các bộ (pid, c), trong đó c là số xe được sở hữu bởi người có mã pid (các bản ghi mà c=0 không cần phải tạo ra)

Select pid, count(cid) From OwnedBy

Where count(cid) not is 0

Group by pid

-- 3. Viết một lệnh SQL truy vấn tất cả các bộ (cid, c) trong đó c là số người sở hữu xe (các bản ghi c=0 không cần phải tạo ra)

Select count(pid), cid From OwnedBy

Where count(pid) not is 0

Group by cid

-- 4. Viết một truy vấn lồng nhau (nested) để trả về tất cả các bộ (pid,rid) trong đó pid lái xe trong vụ tai nạn được báo báo trong rid, nhưng xe được lái bởi người có mã pid không phải là xe được sở hữu bởi người có mã pid.

Select p.pid , p.rid from ParticipatedIn p

Join OwnedBy or on p.pid = o.pid

where p.cid not in ( Select cid from OwnedBy);

-- 5. Viết một truy vấn SQL trả về tất cả các bản ghi (rid,c) trong đó c là số những người lái xe có liên quan đến vụ tai nạn được báo cáo bởi rid (các bản ghi trong đó c=0 không cần phải sinh ra)

Select AccidentReport.rid,ParticipatedIn.c

from AccidentReport join ParticipatedIn on AccidentReport.rid = ParticipatedIn.rid

Where count(pid) not is 0;

-- 6. Viết một truy vấn SQL trả về tất cả các bản ghi (rid,c), trong đó c là số ô tô có liên quan đến vụ tai nạn được báo cáo bởi rid (các bản ghi trong đó c=0 không cần phải được tạo ra)

select rid,cid as c from ParticipatedIn

where cid != 0