

Chöông 5

GIÔÙÌ THIEÄU VEÀ GIAO TAÙC

MUÏC TIEÄU

Chöông naøy giôùì thieäù khaùì nieäm giao taùc vaø ñònh nghóa hình thöùc cuûa giao taùc. Chöông naøy chia laøm ba phaàn:

1. Phaàn thöù nhaát: Khaùì nieäm giao taùc, ñònh nghóa hình thöùc cuûa giao taùc.
2. Phaàn thöù hai: Caùc tính chaát cuûa giao taùc
3. Phaàn thöù ba: Phaân loaïi giao taùc

MÔÛ ÑÀÙ

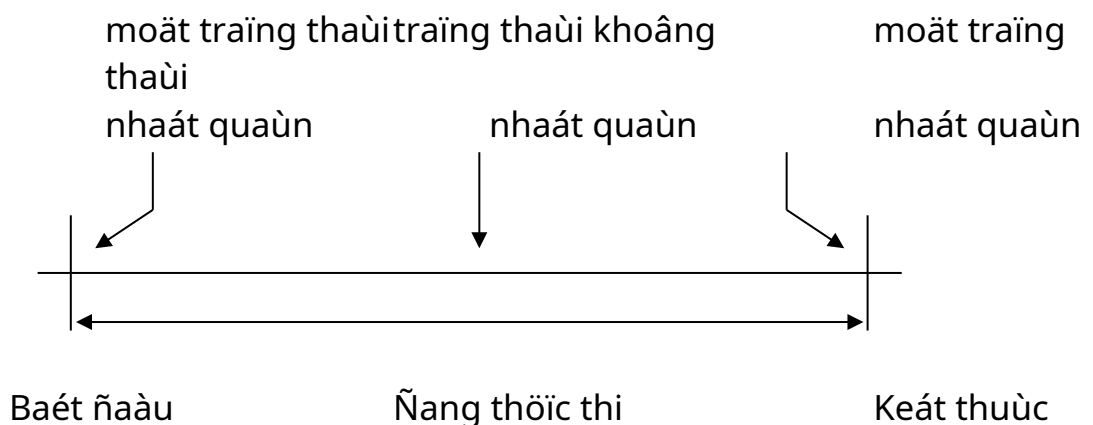
Cho ñéán luùc naøy, ñôn vò truy xuaát cô baûn laø caâu truy vaán. Trong caùc chöông 4, chuùng ta ñaõ thaûo luaän veà caùch xöù lyù vaø toái öu hoaù caùc truy vaán. Tuy nhieân chuùng ta chöa bao giôø xeùt ñéán caùc tình huaóng xaûy ra, chaúng haïn khi hai caâu truy vaán cuøng caäp nhaät moät muïc döõ lieäu, hoaëc tình huaóng heä thoáng bò söï coá phaûi ngöøng hoaït ñoäng trong khi ñang thöïc hieän caâu caäp nhaät. Ñoái vôùi nhöõng caâu truy vaán chæ truy xuaát, khoâng coù tình huaóng naøo ôû treân gaây raéc roái. Ngöôøi ta coù theå cho hai caâu truy vaán ñoïc döõ lieäu cuøng moät luùc. Tôöng töï, sau khi ñaõ xöù lyù xong söï coá, caùc truy vaán chæ ñoïc chæ caàn khôûi ñoäng laïi. Nhöng ngöôøi laïi coù theå nhaän ra raèng ñoái vôùi nhöõng caâu truy vaán caäp nhaät, nhöõng tình huaóng naøy coù theå gaây ra nhöõng toán haïi nghiêâm troïng cho cô sôû döõ lieäu. Nhö chuùng ta khoâng theå chæ khôûi ñoäng laïi cho caâu truy vaán caäp nhaät sau moät söï coá heä thoáng vì moät soá giaù trò cuûa caùc muïc döõ lieäu

coù theå ñaõ ñöôïc caäp nhaät tröôùc khi coù söï coá xaây ra vaø không cho pheùp caäp nhaät laïi khi caâu truy vaán ñöôïc khôù ñöông laïi, neáu không thì cô sôû döõ lieäu seõ chöùa nhöõng döõ lieäu sai leäch.

Ñieäm maáu choát ôû ñây laø không coù khai nieäm “thöïc thi nhaät quaùn” hoaëc “tính toaùn ñaùng tin caäy” ñi keøm vöù khai nieäm truy vaán. Khai nieäm *giao taùc* (transaction) ñöôïc söû düng trong laõnh vöïc cô sôû döõ lieäu nhö möät ñôn vò tính toaùn nhaät quaùn vaø tin caäy ñöôïc. Vì theå caùc caâu truy vaán seõ ñöôïc thöïc thi nhö caùc giao taùc möät khi caùc chieán löôïc thöïc thi ñöôïc xaùc ñònh vaø ñöôïc dòch thaønh caùc thao taùc cô sôû döõ lieäu nguyêân thuûy.

Trong thaùo luaän ôû treân chuùng ta ñaõ döng thuaät ngöõ “*nhaät quaùn*” (consistent) vaø “*ñaùng tin caäy*” (reliable) möät caùch hoø toaøn không hình thöïc. Theå nhöng do taàm quan troïng cuûa chuùng maø chuùng ta caàn phaûi ñònh nghóa chuùng möät caùch chuaån xaùc. Tröôùc tieân phaûi chæ ra raèng caàn phaân bieät giöõa *nhaät quaùn* cô sôû döõ lieäu (database consistency) vaø *nhaät quaùn giao taùc* (transaction consistency).

Möät cô sôû döõ lieäu ôû trong möät *traïng thaùi nhaät quaùn* (consistent state) neáu noù tuaân theo taát caù caùc raøng buoäc toaøn veïn (nhaät quaùn) ñöôïc ñònh nghóa treân noù. Đó nhieân chuùng ta caàn baõ ñaõm raèng cô sôû döõ lieäu không bao giôø chuyeån sang möät traïng thaùi không nhaät quaùn. Cô sôû döõ lieäu coù theå taïm thôøi không nhaät quaùn trong khi thöïc hieän giao taùc. Ñieàu quan troïng laø cô sôû döõ lieäu phaûi trôû veà traïng thaùi nhaät quaùn khi quan heä giao taùc chaám döùt.



Hình 5.1. Mô hình giao taùc.

Ngöôïc laïi, tính nhaát quaùn giao taùc muoán noùi ñeán haønh ñoäng cuûa caùc giao taùc ñoàng thôøi. Chuùng ta mong raêng cô sôû döõ lieäu vaãn nhaát quaùn ngay caù khi coù moät soá yeâu caàu cuûa ngöôøi söû döïng ñoàng thôøi truy xuaát ñeán cô sôû döõ lieäu (ñoïc hoaëc caäp nhaät). Tính chaát phöùc taïp naûy sinh khi xeùt ñeán caùc cô sôû döõ lieäu coù nhaân baân. Moät cô sôû döõ lieäu ñöôïc nhaân baân ôû trong moät *traïng thaùi nhaát quaùn laãn nhau* (mutually consistent state) neáu taát caù caùc baân sao cuûa moãi muïc döõ lieäu ôû trong ñoù ñeàu coù giao trò gioáng nhau. Ñieàu naøy thöôøng ñöôïc goïi laø *söï töông ñöông moät baân* (one copy equivalence) vì taát caù caùc baân ñeàu bò buoäc phaûi nhaän cuøng moät traïng thaùi vaøo cuoái luùc thöïc thi giao taùc. Moät soá khaùì nieäm veà tính nhaát quaùn baân sao cho pheùp giao trò caùc baân sao coù theå khaùc nhau. Nhöõng vaán ñeà naøy seõ ñöôïc thaùo luaän sau.

Noä *tin caäy* hay *khaù tin* (reliability) muoán noùi ñeán khaù naêng *töï thöïc öùng* (resistency) cuûa moät heä thoáng ñoái vôùi caùc loaïi söï coá vaø khaù naêng khoâi phuïc laïi töø nhöõng söï coá naøy. Moät heä thoáng khaù tin seõ töï thöïc öùng vôùi caùc söï coá heä thoáng vaø coù theå tieáp tuïc cung caáp caùc dòch vuï ngay caù khi xaûy ra söï coá.

Moät heä quaùn trò cô sôû döõ lieäu khaù hoài phuïc laø heä quaùn trò cô sôû döõ lieäu coù theå chuyeån sang traïng thaùi nhaát quaùn (baèng caùch quay trôû laïi traïng thaùi nhaát quaùn tröôùc ñoù hoaëc chuyeån sang moät traïng thaùi nhaát quaùn môùi) sau khi gaëp moät söï coá.

Quaùn lý giao taùc (transaction management) laø **giaùì quyeát caùc baøi toaùn duy trì ñöôïc cô sôû döõ lieäu ôû trong tình traïng nhaát quaùn ngay caù khi coù nhieàu truy xuaát ñoàng thôøi vaø khi coù söï coá.**

Muïc ñích cuûa chöông naøy laø ñònh nghóa nhöõng thuaät ngữ cô baân vaø ñieåm ra moät boä khung treân cô sôû ñoù ñeå thaùo luaän caùc vaán ñeà naøy. Ñaây cuøng laø phaàn giôùì thieäu ngaén goïn veà baøi toaùn caàn giaùì quyeát vaø caùc vaán ñeà coù lieân quan. Vì theå chuùng ta seõ trình baøy caùc khaùì nieäm ôû moät möùc tröøu töôïng khaù cao vaø khoâng trình baøy nhöõng kyõ thuaät quaùn lý. Trong phaàn tieáp theo chuùng ta seõ ñònh nghóa moät caùch hình thöïc vaø moät caùch tröïc quan veà khaùì nieäm giao taùc.

5.1 ÑÒNH NGHÓA GIAO TAÙC:

Giao tác được xem như một dãy các thao tác đọc và ghi trên cơ sở dữ liệu cùng với các bước tính toán cần thiết (Begin Trans, Commit, Rollback, Begin Distributed Trans) để đảm bảo tập lệnh như 1 đơn vị lệnh.

Thí dụ 5.1

Xét câu truy vấn SQL làm tăng ngân sách của dự án CAD/CAM lên 10%

```
UPDATE PROJ
SET      BUTGET = BUTGET * 1.1
WHERE    PNAME = "CAD/CAM"
```

Câu truy vấn này có thể hiểu như sau, qua cú pháp SQL ngắn gọn, nhờ một giao tác bằng cách cho nó một tên (thí dụ BUDGET_UPDATE) và khai báo như sau:

```
Begin transaction BUDGET_UPDATE
begin
    UPDATE PROJ
        SET BUTGET = BUTGET * 1.1
        WHERE PNAME = "CAD/CAM"
end
```

Các câu lệnh **Begin transaction** và **end** xác định ranh giới một giao tác. Chúng ta sẽ sử dụng thí dụ về một hệ thống phân cách này hoàn toàn không bắt buộc trong môi trường cơ sở dữ liệu. Chúng ta sẽ thấy rằng các câu phân cách không hiểu như sau, DB2 sẽ xử lý toàn bộ chương trình thời gian truy xuất cơ sở dữ liệu nhờ một giao tác.

Thí dụ 5.2

Trong phần thảo luận về các khái niệm quản lý giao tác, chúng ta sẽ sử dụng thí dụ về một hệ thống phân cách này hoàn toàn không bắt buộc trong môi trường cơ sở dữ liệu. Chúng ta sẽ thấy rằng các câu phân cách không hiểu như sau, DB2 sẽ xử lý toàn bộ chương trình thời gian truy xuất cơ sở dữ liệu nhờ một giao tác.

```
FLIGHT (FNO, DATE, SRC, DEST, STSOLD, CAP)
CUST (CNAME, ADDR, BAL)
FC (FNO, CNAME, SPECIAL)
```

Những thuộc tính trong lược đồ dữ liệu sau: **FNO** là mã số chuyến bay, **DATE** biểu ngày tháng chuyến bay, **STSOLD** là số lượng ghế (seat) đã được bán trên chuyến bay đó, **CAP** là số lượng ghế (so số lượng hành khách có thể chỗ ngồi, capacity) trên chuyến bay, **CNAME** là tên khách hàng với họ và họ tên trong **ADDR** và số điện thoại trong **BAL**, còn **SPECIAL** tổng cộng với các yêu cầu khác biệt mà khách hàng đưa ra khi đặt chỗ.

Chúng ta xem một phiên bản đơn giản hóa của một ứng dụng đặt chỗ, trong đó một nhân viên bán vé nhập mã số chuyến bay, ngày tháng, tên khách hàng và thời gian đặt chỗ. Giao tác thời gian công việc này có thể được cài đặt như sau, trong đó các truy vấn cơ sở dữ liệu được thực hiện bằng SQL:

```
Begin transaction Reservation -- đặt chỗ
begin
    input (flight_no, date, customer_name);
        UPDATE FLIGHT
            SET STSOLD = STSOLD +1
            WHERE FNO = flight_no
        INSERT
            INTO FC (FNO, CNAME, SPECIAL)
            VALUES (flight_no, customer_name, null);
    output ("reservation completed")
end
```

5.1.1 Tình huống kết thúc giao tác

Một giao tác luôn luôn phải kết thúc ngay cả khi có xảy ra lỗi. Nếu giao tác có thể hoàn tất thành công tác vụ của nó, chúng ta gọi rằng giao tác có *ủy thác* (commit). Ngược lại nếu một giao tác phải ngừng lại khi chờ hoàn tất công việc, chúng ta gọi rằng nó *bỏ hủy bỏ* (abort). Một giao tác phải tới hủy bỏ vì có một hoặc nhiều kiến thức làm cho nó không hoàn tất được công việc. Ngoài ra hệ quản trị cơ sở dữ liệu có thể xảy ra một giao tác, chúng ta gọi nó là *khóa chết* (deadlock). Khi một giao tác bỏ hủy bỏ, quá trình thời gian sẽ ngừng và tất cả các hành động đang chờ đợi thời gian sẽ

phải có "undo", nó có thể khôi phục lại trạng thái trước khi thực hiện giao tác. Quá trình này gọi là "rollback".

Vai trò quan trọng của ủy thác biểu hiện ở hai mặt. **Thứ nhất** lệnh ủy thác bảo cho hệ quản trị cơ sở dữ liệu biết rằng tác động của giao tác đó bây giờ cần phải phân ánh vào cơ sở dữ liệu, qua đó làm cho *các giao tác đang truy xuất các mục dữ liệu đó có thể thấy được đúng*. **Thứ hai**, đảm bảo giao tác ủy thác là một *nhãn "không ô nhiễm về"*. Kết quả của một giao tác nào ủy thác bây giờ phải luôn có thể khôi phục lại cơ sở dữ liệu về trạng thái trước khi thực hiện giao tác.

Thí dụ 5.3

Một ví dụ chúng ta có thể thấy là tình huống không còn choãi trống trên chuyến bay. Nếu bao quát khu vực này, giao tác cần phải viết lại như sau:

Begin_transaction Reservation

begin

input(flight_no, date, customer_name);

SELECT STSOLD, CAP

INTO temp1, temp2

FROM FLIGHT

WHERE FNO = flight_no

if temp1 = temp2 **then**

begin

Output("no free seat");

Abort

end

else begin

UPDATE FLIGHT

SET STSOLD = STSOLD +1

WHERE FNO = flight_no

INSERT

INTO FC(FNO, CNAME, SPECIAL)

VALUES(flight_no, customer_name, null);

Commit;

Output("reservation completed")

end

end.

Qua thí dui nàøy chuùng ta thaáy ñöôïc nhieàu ñieåm quan troïng. Roõ raøng neáu khoâng coøn choã troáng, giao taùc phaûi huûy boû. Thöù hai laø vieäc saép thöù töï caùc keát quaû ñeà trình baøy ra cho ngôøi söû duïng tuøy theo caùc leänh **abort** vaø **commit**. Chuù yù raèng neáu giao taùc bò huûy boû, ngôøi söû duïng seõ ñöôïc thoâng baùo troöùc khi heà quaûn trò cô söû döõ lieäu ñöôïc höôùng daãn ñeà huûy boû noù. Theá nhöng vöùi troöøng hôïp uûy thaùc, thoâng baùo cho ngôøi söû duïng phaûi xaûy ra sau khi heà quaûn trò cô söû döõ lieäu ñaõ thöïc hieän xong leänh uûy thaùc ñeà baùo ñaâm ñoà khaù tïn.

5.1.2 Ñăéc tröng hoàu caùc giao taùc

Chuùng ta nhaän xeùt raèng caùc giao taùc ñeàu ñoïc vaø ghi moät soá döõ lieäu. Ñieàu naøy ñöôïc duøng laøm cô sôû nhaän bieát moät giao taùc. Caùc muïc döõ lieäu ñöôïc giao taùc ñoïc caáu taïo nên *taäp ñoïc* RS (read set) cuûa noù. Töông töï, caùc muïc döõ lieäu ñöôïc moät giao taùc ghi ñöôïc goïi laø *taäp ghi* WS (write set). Chuù yù raèng taäp ñoïc vaø taäp ghi cuûa moät giao taùc khoâng nhaát thieát phaûi taùch bieät. Cuái cuøng hôïp cuûa taäp ñoïc vaø taäp ghi cuûa moät giao taùc taïo ra *taäp cô sôû* BS (base set), nghóa laø $BS = RS \cup WS$.

Thí dụ 5.4

Chuồng ta xeùt laii giao taùc ñaët choã cuûa thí dui 5.3 vaø thao taùc cheøn chöùa moät soá thao taùc ghi. Caùc taäp neâu treân ñöôïc ñònh nghóa nhö sau:

$$RS [Revervation] = \{FLIGHT.STSOLD, FLIGHT.CAP\}$$
$$WS[Revervation] = \{FLIGHT.STSOLD, FC.FNO, FC.CNAME, FC.SPECIAL\}$$
$$BS[Revervation] = \{FLIGHT.STSOLD, FLIGHT.CAP, FC.FNO, FC.DATE, FC.CNAME, FC.SPECIAL\}$$

Chuùng ta ñã ñãẽ trõng caùc giao taùc chæ treân cô sôu caùc thao taùc ñoïc vaø ghi maø khoâng xem xeùt caùc thao taùc cheøn, xoaù. Nhõ theá chuùng ta ñãõ thaõ luaän veà khaùì nieãm caùc giao taùc döïa treân caùc cô sôu döõ lieäu tónh, khoâng nôùì roãng hoaëc thu laii. Giaûn löôïc naøy ñöôïc ñöa ra ñeå coù ñöôïc tính ñôn giaûn. Caùc cô sôu döõ lieäu ñoäng phaùì giaùì quyeát baøi toaùn *aûnh aùo* (phantom), ñöôïc giaùì thíc nhõ ví dụ sau.

Xeùt giao taùc T_1 , khi thöïc hieän caàn tìm trong baùng **FC** teân nhöõng khaùch haøng ñaõ yeâu caàu duøng moät böõa aên ñaëc bieät. Noù nhaän ñöôïc moät taäp **CNAME** goàm nhöõng khaùch haøng thoûa thuaän ñieàu kieän tìm kieám. Khi T_1 ñang thöïc hieän, moät giao taùc T_2 cheøn caùc böõa môùi vaøo **FC** coù yeâu caàu böõa aên ñaëc bieät roài uý thaùc. Neáu sau ñoù T_1 laïi ñöa ra caâu truy vaán tìm kieám nhö cuõ, noù seõ nhaän ñöôïc moät taäp **CNAME** khaùc vôùi taäp ban ñaàu maø noù ñaõ nhaän. Vì theá caùc böõa “aûnh aûo” ñaõ xuaát hieän trong cô sôù döõ lieäu.

5.1.3. Hình thöùc hoàu khaùì nieäm giao taùc

Cho ñeán luùc naøy, yù nghóa tröïc quan cuûa giao taùc ñaõ hoæn toaøn roõ raøng. Ñeå baøn luaän veà caùc giao taùc vaø suy dieãn tính ñuùng ñaén cuûa caùc thuaät toaùn quaûn lý giao taùc, chuùng ta caàn ñònh nghóa khaùì nieäm naøy moät caùch hình thöùc. Chuùng ta bieâu thò *pheùp toaùn* O_j cuûa giao taùc T_i khi hoaït taùc treân thöïc theá cô sôù döõ lieäu x laø $O_{ij}(x)$. Theo qui öôùc ñöôïc thoøa nhaän ôû phaàn tröôùc, $O_{ij} \in \{\text{read}, \text{write}\}$. Caùc pheùp toaùn ñöôïc giaù thieát laø *nguyeân töû* (nghóa laø moãi pheùp toaùn ñöôïc thöïc thi nhö moät ñôn vò khoâng theá chia nhoû ñöôïc nöõa). Chuùng ta haõy kyù hieäu OS_i laø taäp taát caù caùc pheùp toaùn trong T_i (nghóa laø $OS_i = \bigcup_j O_{ij}$). N_i bieâu thò cho tình huoáng cuûa T_i , trong ñoù $N_i \in \{\text{abort}, \text{commit}\}$.

Vôùi caùc thuaät ngữõ naøy, chuùng ta coù theá ñònh nghóa T_i laø moät *thòu töï böõa phaän* treân caùc pheùp toaùn vaø tình huoáng keát thuùc cuûa noù. Thòu töï böõa phaän $P = \{\Sigma, <\}$ ñònh nghóa moät traät töï giöõa caùc phaàn töû cuûa Σ (ñöôïc goïi laø *mieàn*) qua moät quan heä hai ngoâi baéc caàu vaø khoâng phaân xaï $<$ ñöôïc ñònh nghóa treân Σ . Trong tröôøng hôïp ñang xeùt, Σ bao goàm caùc pheùp toaùn vaø tình huoáng keát thuùc cuûa moät giao taùc, trong ñoù $<$ chæ thòu töï thöïc hieän cuûa caùc pheùp toaùn naøy (ñöôïc chuùng ta ñöïc laø “ñònh tröôùc theo thòu töï thöïc thi”).

Moät caùch hình thöùc, moät giao taùc T_i laø moät thòu töï böõa phaän $T_i = \{\Sigma_i, <_i\}$, trong ñoù

1. $\Sigma_i = OS_i \cup \{N_i\}$.
2. Vôùi hai pheùp toaùn baát kyø $O_{ij}, O_{ik} \in OS_i$, neáu $O_{ij} = \{R(x) \text{ or } W(x)\}$ vaø $O_{ik} = W(x)$ vôùi moät muïc döõ lieäu x naøo ñoù, theá thì $O_{ij} <_i O_{ik}$ hoaëc $O_{ik} <_i O_{ij}$.
3. $\forall O_{ij} \in OS_i, O_{ij} <_i N_i$.

Ñieàu kieän thöù nhaát veà hình thöùc ñönh nghóa *mieän* nhö moät taäp caùc thao taùc ñöïc vaø ghi caáu taïo neân giao taùc coäng vôùi tình huoáng keát thuùc, coù theå laø *commit* hoaëc *abort*. Ñieàu kieän thöù hai xaùc ñönh quan heä thöù töï giöõa caùc thao taùc ñöïc vaø ghi coù töông tranh cuûa giao taùc, coøn ñieàu kieän cuoái cuøng chæ ra raèng tình huoáng keát thuùc luôn ñi sau taát caù nhöõng thao taùc khaùc.

Coù hai ñieäm quan troïng caàn löu yù trong ñönh nghóa naøy. Tröôùc tieân, quan heä thöù töï $<$ ñöôïc cho tröôùc vaø ñönh nghóa naøy khoâng heä xaây döïng noù. Quan heä thöù töï thöïc söï phui thuoäc vaøo öùng duïng. Keá ñeán, ñieàu kieän thöù hai chæ ra raèng thöù töï giöõa caùc thao taùc coù töông tranh phaûi toàøn taïi beân trong $<$. Hai thao taùc $O_i(x)$ vaø $O_j(x)$ ñöôïc goïi laø coù töông tranh neáu $O_i = \text{Write}$ hoaëc $O_j = \text{Write}$ (coù nghóa ít nhaát moät trong chuùng laø *Write* vaø chuùng truy xuaát cuøng moät muïc döõ lieäu).

Thí duï 5.5

Xeùt moät giao taùc ñôn giaûn T coù caùc böôùc sau:

Read(x)
Read(y)
 $x \leftarrow x+y$
Write(x)
Commit

Ñeéc taù cuûa giao taùc naøy theo kyù phaùp hình thöùc ñeä ñöôïc giöùì thieäü ôû treân laø:

$$\Sigma = \{R(x), R(y), W(x), C\}$$

$$< = \{(R(x), W(x)), (R(y), W(x)), (W(x), C), (R(x), C), (R(y), C)\}$$

trong ñoù (O_i, O_j) , laø moät phaàn töù cuûa quan heä $<$, bieäu thò raèng $O_i < O_j$.

Chuù yù raèng quan heä thöù töï töông ñoái cuûa taát caù caùc thao taùc öùng vôùi tình huoáng keát thuùc. Ñieàu naøy laø do ñieàu kieän thöù ba cuûa ñönh nghóa giao taùc. Cuõng caàn chuù yù raèng chuùng ta khoâng mô taù thöù töï giöõa moãi caëp thao taùc. Ñieàu ñoù giaûi thích vì sao ñeäy laø moät thöù töï boä phaën.

Thí duï 5.6

Giao taùc ñeä choã ñöôïc xaây döïng trong thí duï 5.3 phöùc taïp hôn. Chuù yù cho raèng coù hai tình huoáng keát thuùc, tuøy vaøo tình traïng coù coøn choã troáng hay khoâng. Tröôùc tieân, ñieàu naøy

dùng nhớ mẫu thừa vôi nòng nghóa của giao tác, nó là chæ toàn tã một tình huống kết thúc. Tuy nhiên cần nhò ràng giao tác là một thõc thì của một chương trình. Rõ ràng là trong bất kỳ lần thõc thì nào, chæ một trong hai tình huống kết thúc xảy ra. Vì thế nều có thể xảy ra là một giao tác huỷ bỏ và một giao tác khác uý thác. Số dùng kỳ phò hình thõc này, giao tác nào có thể nõic nãc tũ nhò sau:

$$\Sigma = \{R(STSOLD), R(CAP), A\}$$

$$< = \{(O_1, A), (O_2, A)\}$$

và giao tác sau nõic nãc tũ nhò sau

$$\Sigma = \{R(STSOLD), R(CAP), W(STSOLD), W(FNO), W(ĐATE), W(CNAME), W(SPECIAL), C\}$$

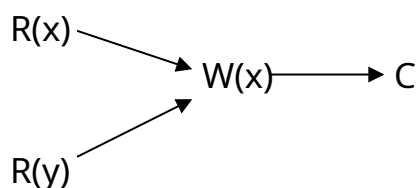
$$< = \{(O_1, O_3), (O_2, O_3), (O_1, O_4), (O_1, O_5), (O_1, O_6), (O_1, O_7), (O_2, O_4), (O_2, O_5), (O_2, O_6), (O_2, O_7), (O_1, C), (O_2, C), (O_3, C), (O_4, C), (O_5, C), (O_6, C), (O_7, C)\}$$

trong nó $O_1 = R(STSOLD)$, $O_2 = R(CAP)$, $O_3 = W(STSOLD)$, $O_4 = W(FNO)$, $O_5 = W(ĐATE)$, $O_6 = W(CNAME)$, $O_7 = W(SPECIAL)$.

Một ầu nểm của việc nòng nghóa giao tác nhò một thõc tũ bỏ phàn là sũ tòng ờng của nó vôi *nóà thò có hõùng không vọng* DAG (directed acyclic graph). Nhò thế một giao tác có thể nõic nãc tũ nhò một DAG vôi nẽnh là các pheùp toàn giao tác và cùng chæ ra mõi liên hệ thõc tũ giõa các caù pheùp toàn nãc cho.

Thí dụ 5.7

Giao tác nõic thảo luận trong thí dụ 5 có thể nõic vẽ nhò một DAG của hình 5.2. Chuỳ ý rằng chúng ta không vẽ các cùng nõic suy ra nhò tính chæt bácc cầu đợ ràng chúng ta xem chúng nhò nhõng phàn tũ của <



Hình 5.2. Biểu diễn dạng DAG cho một giao tác.

Trong phaàn lôùn caùc trööông hôïp chuùng ta khoâng caàn phaûi ñeà caäp ñeán nhieàu mieàn cuûa thòu töï boä phaän moät caùch rieâng reõ vôùi quan heä thòu töï. Vì theá thoâng thöông chuùng ta boû Σ ra khoûi ñònh nghóa giao taùc vaø söû duïng teân cuûa thòu töï boä phaän ñeà chæ ñeán caù mieàn laãn teân cuûa thòu töï boä phaän. Ñieàu ñoù seõ tieän lôïi böôù vì noù cho pheùp chuùng ta ñeà taù thòu töï cuûa caùc pheùp toaùn trong moät giao taùc nhö moät phöông thöùc khaù ñôn giaûn baèng caùch duøng thòu töï töông ñoái cuûa ñònh nghóa giao taùc. Chaúng haïn chuùng ta coù theá ñònh nghóa giao taùc cuûa thí dụ 5 nhö sau:

$$T = \{R(x), R(y), W(x), C\}$$

thay vì ñeà taù daøi doøng nhö trööùc.

5.2 CAÙC TÍNH CHAÁT CUÛA GIAO TAÙC

Caùc khía caïnh nhaát quaùn vaø khaù tín cuûa giao taùc laø do boán tính chaát: (1) *tính nguyêân töù* (atomicity), (2) *tính nhaát quaùn* (consistency), (3) *tính bieät laäp* (isolation), (4) *tính beàn vôõng* (durability); vaø chuùng ta thöông ñöôïc goïi chung laø tính chaát ACID.

5.2.1 Tính nguyêân töù

Tính nguyêân töù lieân quan ñeán söï kieän laø moät giao taùc ñöôïc xöû lý nhö moät ñôn vò hoaït taùc. Chính vì theá maø caùc haønh ñoäng cuûa giao taùc, hoaëc taát caù ñeà hoøn taát hoaëc khoâng moät haønh ñoäng naøu hoøn taát. Ñieàu naøy cuõng ñöôïc goïi laø tính chaát “ñöôïc aên caù ngaõ veà khoâng” (all-or-nothing). Tính nguyêân töù ñoøi hoûi raèng neáu vieäc thöïc thi giao taùc bò caét ngang böôù moät loaïi söï coá naøu ñoù thì heä quaùn trò cô söû döõ lieäu seõ chòu traùch nhieäm xaùc ñònh coâng vieäc caàn thöïc hieän ñoái vôùi giao taùc ñeà khoâi phuïc laïi sau söï coá. Đó nhieân coù hai chieàu höôùng haønh ñoäng: hoaëc noù seõ ñöôïc keát thuùc baèng caùch hoøn taát caùc haønh ñoäng coøn laïi, hoaëc coù theá ñöôïc keát thuùc baèng caùch hoài laïi taát caù caùc haønh ñoäng ñaõ ñöôïc thöïc hieän.

5.2.2 Tính nhaát quaùn

Tính nhaát quaùn (consistency) cuûa moät giao taùc chæ ñôn giaûn laø tính ñuùng ñeán cuûa noù. Noù caùch khaùc, moät giao taùc laø moät chöông trình ñuùng ñeán, aùnh xaï cô söû döõ lieäu töø traïng thaùi nhaát quaùn naøy sang moät traïng thaùi nhaát quaùn khaùc.

Trong ñönh nghöa döùì ñây, döù lieäu raùc (dirty data) muón noùi ñeán nhöõng giao trò döù lieäu ñaõ ñöïc caäp nhaät böüi möät giao taùc tröôùc khi noù uý thaùc. Do ñoù döü treân khai nieäm veà döù lieäu raùc, ba möùc ñaõ ñöïc ñönh nghöa nhö sau:

Ñöä 3: Giao taùc T thoa **nhaät quaùn ñöä 3** neáu:

1. T khoâng ñeù leân döù lieäu raùc cuûa nhöõng giao taùc khaiùc.
2. T khoâng uý thaùc baát kyø thao taùc ghi naøo cho ñeán khi noù hoæn taát möüi thao taùc ghi [nghöa laø cho ñeán luùc cuái giao taùc (end-of-transaction, EOT)].
3. T khoâng ñöïc döù lieäu raùc cuûa nhöõng giao taùc khaiùc.
4. *Nhöõng giao taùc khaiùc khoâng laøm cho döù lieäu maø T ñaõ ñöïc - tröôùc khi T hoæn taát - trôù thaønh döù lieäu raùc.*

Ñöä 2: Giao taùc T thoa **nhaät quaùn ñöä 2** neáu:

1. T khoâng ñeù leân döù lieäu raùc cuûa nhöõng giao taùc khaiùc.
2. T khoâng uý thaùc baát kyø thao taùc ghi naøo tröôùc EOT.
3. T khoâng ñöïc döù lieäu raùc cuûa nhöõng giao taùc khaiùc.

Ñöä 1: Giao taùc T thoa **nhaät quaùn ñöä 1** neáu:

1. T khoâng ñeù leân döù lieäu raùc cuûa nhöõng giao taùc khaiùc.
2. T khoâng uý thaùc baát kyø thao taùc ghi naøo tröôùc EOT.

Ñöông nhieân ñaõ nhaät quaùn cao bao truøm taát cá ñaõ nhaät quaùn möùc thaáp hôn. YÙ töôùng trong vieäc ñönh nghöa nhieàu möùc nhaät quaùn laø cung caäp cho läp trình vieân öùng döïng möät khai naêng linh hoaït khi ñönh nghöa caùc giao taùc hoaït taùc öü nhöõng möùc khaiùc nhau. Heä quaù laø maëc duø möät soá giao taùc hoaït taùc öü möùc nhaät quaùn Ñöä 3, caùc giao taùc khaiùc coù theä hoaït taùc öü nhöõng möùc thaáp hôn, vaø raát coù theä seõ nhìn thaáy caùc döù lieäu raùc.

5.2.3 Tính bieät läp

Bieät läp laø tính chaát cuûa caùc giao taùc, ñöü hoüi möüi giao taùc phaüi luôn nhìn thaáy cô söù döù lieäu nhaät quaùn. Noùi caùch khaiùc, möät giao taùc ñang thöïc thi khoâng theä laøm loä ra caùc keát quaù cuûa noù cho nhöõng giao taùc khaiùc ñang cuøng hoaït ñöäng tröôùc khi noù uý thaùc.

Coù möät soá lý do caàn phaüi nhaän maïnh ñeán tính bieät läp. Möät laø duy trì tính nhaät quaùn qua läü giöüa caùc giao taùc. Neáu

hai giao tác nào thời truy xuất nên một mức độ liệu rằng
nó có một trong chúng cấp nhất thì không thể bảo đảm
ràng giao tác thời hai sẽ nó có nó có giao tác trở nên.

Thí dụ 5.8

Xét hai giao tác nào thời T_1 và T_2 cùng truy xuất nên
mức độ liệu x . Giả sử **giao tác của x trở về khi biết rằng
thời hiện là 50.**

T_1 :	Read(x)	T_2 :	Read(x)
	$x \leftarrow x + 1$		$x \leftarrow x + 1$
	Write(x)		Write(x)
	Commit		Commit

Đôi khi là một dãy thời thì cho các hành động này.

T_1 :	Read(x)
T_1 :	$x \leftarrow x + 1$
T_1 :	Write(x)
T_1 :	Commit
T_2 :	Read(x)
T_2 :	$x \leftarrow x + 1$
T_2 :	Write(x)
T_2 :	Commit

Ôu trở về hợp này không có vấn đề gì; các giao tác T_1
và T_2 nó có thời hiện là lần lượt và giao tác T_2 nó có nó có giao tác
của x là 51. Chuỗi này nếu T_2 thời thì trở về T_1 thì T_2 nó có nó có
giao tác 50. Vì thế nếu T_1 và T_2 nó có thời thì lần lượt giao tác
này rồi nên giao tác kia, giao tác thời hai sẽ nó có nó có giao tác
của x là 51 và sau khi kết thúc hai giao tác x có giao tác 52. Tuy
nhiên vì các giao tác này thời thì nào thời, dãy thời thì sau
này có thể sẽ xảy ra:

T_1 :	Read(x)
T_1 :	$x \leftarrow x + 1$
T_2 :	Read(x)
T_1 :	Write(x)
T_2 :	$x \leftarrow x + 1$
T_2 :	Write(x)
T_1 :	Commit
T_2 :	Commit

Trong trở về hợp này, giao tác T_2 nó có nó có giao tác của x
là 50. Giao tác này không nên bị bỏ vì T_2 nó có x trong khi giao tác
của nó này nó có thay nó từ 50 thành 51. Hết nó giao tác

cuûa x seõ laø 51 vaøo luùc keát thuùc caùc giao taùc T_1 vaø T_2 böüi vì haønh ñoäng ghi cuûa T_2 seõ ñeõ leân keát quaû ghi cuûa T_1 .

Baûu ñaûm tính bieät laäp baèng caùch khoâng cho pheùp caùc giao taùc khaùc nhìn thaáy caùc keát quaû chöa hoàøn taát nhö trong thí dụi trên seõ giaûi quyéát ñöôïc vaán ñeà *caáp nhaät thaát laïc* (lost update). Loaiï bieät laäp naøy ñaõ ñöôïc goïi laø *tính oãn ñoàng con cháiï* (cursor stability). Trong thí dụi ôû trên, daõy thöïc thi thöu hai ñaõ laøm cho taùc düng cuûa T_1 bò maát. Moät lyù do thöu hai cuûa tính bieät laäp laø caùc *huûy boû daây chuyeàn* (cascading abort). Neáu moät giao taùc cho pheùp nhöõng giao taùc khaùc nhìn thaáy nhöõng keát quaû chöa hoàøn taát cuûa noù tröôùc khi uûy thaùc roài noù quyéát ñoàng huûy boû, moïi giao taùc ñaõ ñöôïc nhöõng giaù trò chöa hoàøn taát ñoù cuõng seõ phaûi huûy boû. Xaâu maéc xích naøy ñeõ laøm taêng nhanh vaø gaây ra nhöõng phí toaøn ñaùng keå cho heä quaûn trò cô sôû döõ lieäu.

Cuõng coù theå xöu trí caùc möùc nhaät quaûn ñaõ thaûo luaän trong phaàn tröôùc töø quan ñieäm cuûa tính chaát bieät laäp (vì theå ñaõ minh hoã cho söi phuï thuoác giöõa tính nhaät quaûn vaø tính bieät laäp). Khi di chuyeån leân caây phaân caáp caùc möùc nhaät quaûn, caùc giao taùc ngaøy caøng bieät laäp hôn. Ñoã 0 cung caáp raát ít tính chaát “bieät laäp” ngoaøi vieäc ngaên caûn caùc caáp nhaät thaát laïc. Tuy nhieân vì caùc giao taùc seõ uûy thaùc tröôùc khi chuùng hoàøn taát taát caû moïi thao taùc ghi cuûa chuùng, neáu coù moät huûy boû xaûy ra sau ñoù, noù seõ ñoøi hoûi phaûi hoài laïi taát caû caùc caáp nhaät trên caùc müic döõ lieäu ñaõ ñöôïc uûy thaùc vaø hieän ñang ñöôïc truy xuaát böüi nhöõng giao taùc khaùc. Nhaät quaûn ñoã 2 traùnh ñöôïc caùc huûy boû daây chuyeàn. Ñoã 3 cung caáp toaøn boã khaû naêng bieät laäp, buoäc moät trong caùc giao taùc töõng tranh phaûi ñöü cho ñeán khi giao taùc kia keát thuùc. Nhöõng daõy thöïc thi nhö theå ñöôïc goïi laø nghieâm ngaët (strict) vaø seõ ñöôïc thaûo luaän nhieàu hôn trong chöông tieáp theo. Roõ raøng laø vaán ñeà bieät laäp coù lieân quan tröïc tieáp ñeán tính nhaät quaûn cô sôû döõ lieäu vaø vì theå laø ñeà taøi cuûa ñieäu khieän ñoàng thöu.

Ba hieän töõng ñöôïc ñaéc taù cho nhöõng tình huoáng coù theå xaûy ra neáu söi bieät laäp thích hôïp khoâng ñöôïc duy trì laø:

Ñöïc raùc (Dirty Read): döõ lieäu raùc muoán noù ñeán caùc müic döõ lieäu maø giaù trò cuûa chuùng ñaõ ñöôïc söûa ñoái böüi moät giao taùc chöa uûy thaùc. Xeùt tröôøng hôïp giao taùc T_1 söûa ñoái moät giaù trò döõ lieäu roài noù laïi ñöôïc böïc böüi

một giao tử T2 khác trước khi T1 thực hiện Commit hay Abort. Trong trường hợp Abort, T2 nào nào một giao tử chờ đợi toàn tài trong cơ sở dữ liệu.

Moät ñaëc taû chính xaùc trong hieän töôïng naøy nhö sau (vôùi caùc cöôùc soá chæ ra teân caùc giao taùc)

$$..., W_1(x), ..., R_2(x), ... C_1(\text{hoa} \ddot{e}c A_1), ..., C_2(\text{hoa} \ddot{e}c A_2)$$

hoaëc

$$..., W_1(x), ..., R_2(x), ..., C_2(\text{hoa} \ddot{c} A_2), ..., C_1(\text{hoa} \ddot{c} A_1)$$

Không thể nào lại (Non-repeatable Read): Giao tử T1 nào một lần rồi. Sau đó một giao tử T2 khác sẽ đọc lại và có thể khác với lần đầu. Nếu sau T1 nào lại một lần rồi, hoặc nó nào khác một lần rồi khác với lần đầu; vì thế hai lần nào trong cùng một giao tử T1 tra về cùng kết quả khác nhau.

Moät ñaéc taû chính xác của hiện tượng này như sau:

$$..., R_1(x), ..., W_2(x), ..., C_1(\text{hoa} \ddot{e}c A_1), ..., C_2(\text{hoa} \ddot{e}c A_2)$$

hoaëc

$$..., R_1(x), ..., W_2(x), ..., C_2(\text{hoa} \ddot{e}c A_2), ..., C_1(\text{hoa} \ddot{e}c A_1)$$

Aûnh aûo (phantom): Nïeàu kieän aûnh aûo trööùc kia ñã ñöôïc ñönh nghóa xâu ra khi T1 thöïc hieän tìm kieám theo möät vò töø vao T2 cheøn nhöõng böa möüi thoa vò töø ñoù. Ñaéc taû chính xâu cuûa hieän tööïng naøy laø (P laø vò töø tìm kieám)

..., $R_1(P)$, ..., $W_2(y \text{ thuộc } P)$, ..., $C_1(\text{hoaëc } A_1)$, ..., $C_2(\text{hoaëc } A_2)$

hoaëc

..., $R_1(P)$, ..., $R_2(y \text{ thuộc } P)$, ..., $C_2(\text{hoaëc } A_2)$, ..., $C_1(\text{hoaëc } A_1)$

Döia treân nhöông hieän tööng naøy, caùc möüc bieät laäp ñaõ ñöôïc ñònh nghóa nhö sau. Muïc tieâu cuûa vieäc ñònh nghóa nhieàu möüc bieät laäp cuõng gioáng nhö vieäc ñònh nghóa caùc möüc nhaát quaùn.

5.2.4 Tính bền vững

Tính bền vững (durability) muốn nói đến tính chất của giao thức, bảo đảm rằng một khi giao thức ứng dụng, kết quả của nó không duy trì có thể không rõ ràng ra khỏi cơ sở dữ liệu. Vì thế hệ qua đời từ cơ sở dữ liệu bảo đảm rằng kết quả của giao thức sẽ vẫn tồn tại dù có xảy ra sự cố hệ thống. Đây chính là lý do mà trong thí dụ 5.2 chúng ta đã nhận thấy rằng giao thức ứng dụng trở nên khi nó thông báo

cho người sử dụng biết rằng nó đã hoàn tất thành công. Tính bền vững nữa ra vấn đề *khôi phục* cơ sở dữ liệu (database recovery), nghĩa là cách khôi phục cơ sở dữ liệu về trạng thái nhất quán mà nó có mọi hành động đã ủy thác nên nó có phân ảnh.

5.3 CÁC LOẠI GIAO TÁC: giao tác tập trung, giao tác phân tán.

5.3.1. Giao tác tập trung:

a/ Giao tác phaúng

Giao tác phaúng (flat transaction) có một khối nghiệp vụ duy nhất (**Begin_transaction**) và một nghiệp vụ kết thúc duy nhất (**End_transaction**). Tất cả các thí dụ của chúng ta đã xem xét đều nằm trong nhóm này. Phần lớn các nghiên cứu về quản lý giao tác trong cơ sở dữ liệu đều tập trung vào các giao tác phaúng.

b/ Giao tác lồng

Nhảy là mô hình giao tác cho phép một giao tác gồm chứa giao tác khác với nghiệp vụ bắt đầu và ủy thác của riêng chúng. Những giao tác nhỏ thế nên gọi là *giao tác lồng* (nested transaction). Những giao tác nên đặt vào trong giao tác khác thông thường nên gọi là *giao tác con* (subtransaction)

Thí dụ 5.10

Chúng ta hãy mô tả rằng giao tác đặt cho cả thí dụ 2. Phần lớn các hàng du lịch nên lo cả việc đặt cho cả khách sạn và phòng ở tại ngoại dèch với đặt về mùa bay. Nếu người ta muốn mô tả tất cả những công việc này bằng một giao tác, thì giao tác đặt cho sẽ có cấu trúc như sau:

Begin_transaction Reservation

begin

Begin_transaction Airline with mark

...

Begin_transaction Hotel with mark

...

end. { Hotel }

.....

end. { Airline }

Begin_transaction Car with mark

...

end. {Car }

end.

Các giao tác luôn phải tuân thủ một khai niệm giao tác tổng quát hơn. Mục đích của chúng là để đảm bảo, cho phép các giao tác con cũng có thể có các giao tác con. Tính tổng quát này có ích trong các lĩnh vực ứng dụng mà ở đó các giao tác phức tạp hơn so với việc xử lý dữ liệu truyền thống.

5.3.2. Giao tác phân tán: chỉ có giao tác phẳng