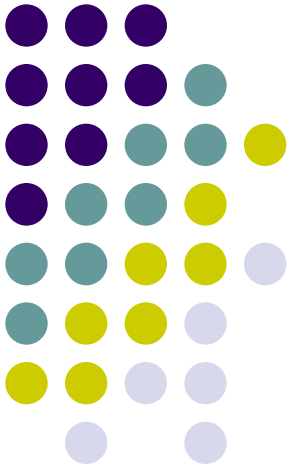
Applied Faculty
Hama University

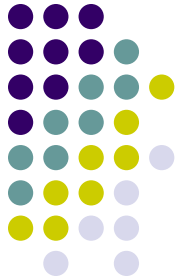
Database 2

معالجة المناقلات (١)

Transaction Processing



Dr. Ramez Alkhatib

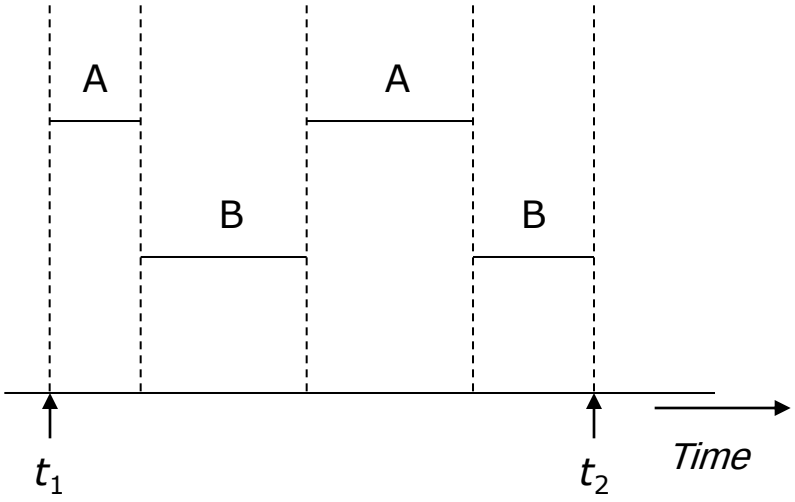


مقدمة

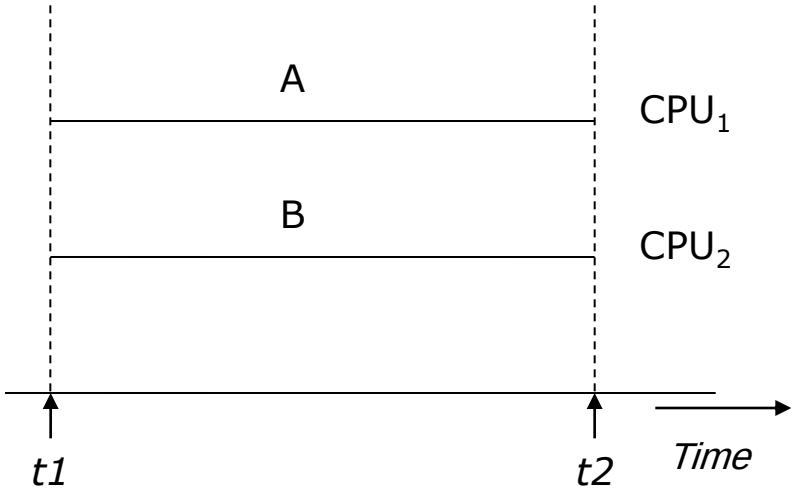
- **النظام وحيد المستخدم:** على الأكثر مستخدم واحد يمكنه استخدام النظام في ذات الوقت
- **النظام متعدد المستخدمين:** عدة مستخدمين يمكنهم استخدام النظام بشكل متزامن

- التزامن:
  - **المعالجة المتداخلة:** التنفيذ المتزامن للعمليات يتداخل في معالج واحد
  - **المعالجة التفرعية:** العمليات تنفذ بشكل متزامن في عدة معالجات

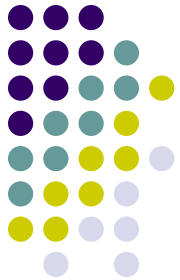
مقدمة



المعالجة المتداخلة
(معالج واحد)

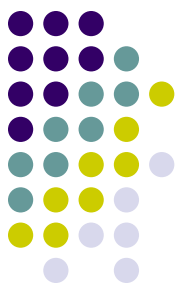


المعالجة التفرعية
(عدة معالجات)



المناقلة (Transaction)

- المناقلة هي وحدة منطقية من معالجة قاعدة البيانات تتضمن واحدة أو أكثر من عمليات الوصول:
  - قراءة: استرجاع
  - كتابة: إدراج، حذف، تعديل
- قد تكون المناقلة موصفة بلغة عالية المستوى مثل SQL و محالة بشكل تفاعلي
- أو قد تكون مزروعة في برنامج
- حدود المناقلة: Begin Transaction و End Transaction
- قد يحتوي برنامج تطبيقي على عدة مناقلات مفصولة بالحدود Begin Transaction و End Transaction



المناقلة (Transaction)

- المناقلة هي برنامج تنفيذي يشكل وحدة منطقية من معالجة قاعدة البيانات
- تتضمن المناقلة واحدة أو أكثر من عمليات الوصول إلى قاعدة البيانات
  - مثل الإدراج و الحذف و التعديل و الاسترجاع

- مثال:
  - سلسلة من تعليمات SQL مجمعة معاً
  - أو كتلة برمجية في PL/SQL

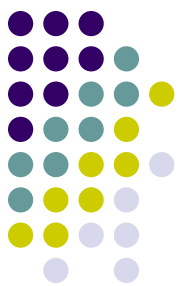
- إذا تم إنجاز جزء من المناقلة فقط ، قد تصبح قاعدة البيانات في حالة غير متجانسة

مثال عن برنامج PL/SQL

- البرنامج التالي ينقل مبلغ من المال من حساب إلى آخر
  - srcacct الحساب الذي سننقل المال منه
  - dstacct الحساب الذي سننقل المال إليه
  - amt مبلغ المال الذي سننقله

```
BEGIN
   UPDATE account
   SET balance = balance - amt
   WHERE account_number = srcacct;
   UPDATE account
   SET balance = balance + amt
   WHERE account_number = dstacct;
END;
/
```

لنفرض حصول انهيار عند هذه النقطة



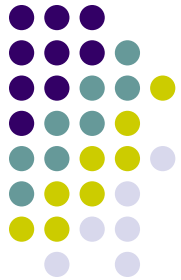
نموذج مبسط

- قاعدة البيانات: مجموعة من عناصر البيانات
- عناصر البيانات: حقل أو سجل أو كتلة قرص كاملة

- العمليات الأساسية هي القراءة و الكتابة

read_item(x)	تقرأ عنصر قاعدة بيانات اسمه X في متحول برنامج (اسمه X أيضاً)
write_item(x)	تكتب قيمة متحول برنامج X إلى عنصر قاعدة بيانات (اسمه X أيضاً)

نموذج مبسط



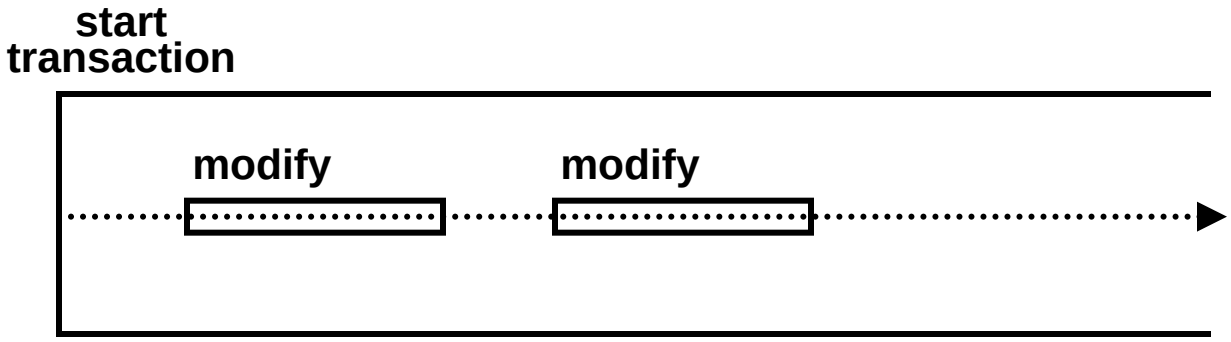
T_1	T_2
read_item(X);	read_item(X);
X := X - N;	X := X + M;
write_item(X);	write_item(X);
read_item(Y);	
Y := Y + N;	
write_item(Y);	

Time ↓

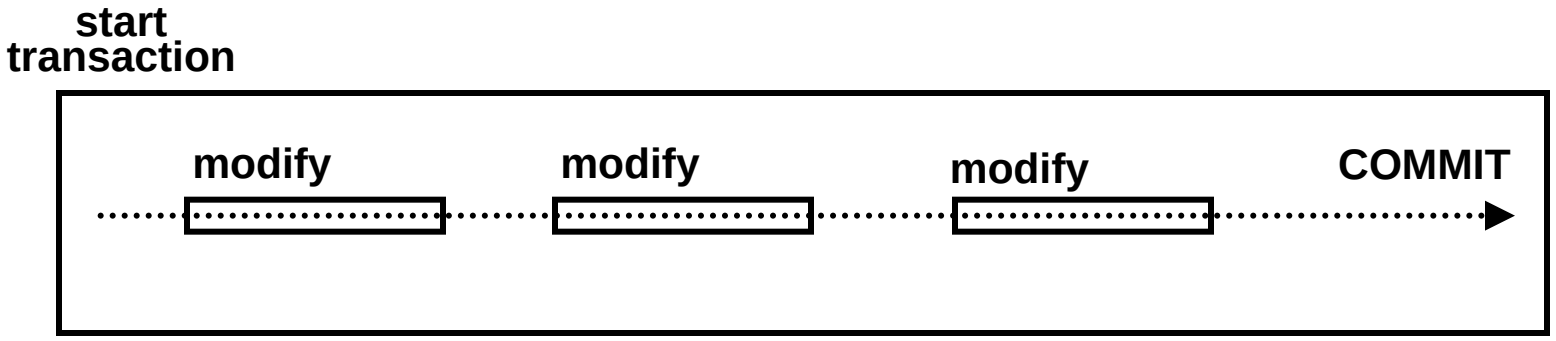
حالات المناقلات

- المناقلة هي وحدة ذرية من العمل، يجب أن تنجز بكاملها أو لا تنجز أبداً
- بهدف الاستعادة، يحتاج النظام إلى تتبع أثر متى تبدأ المناقلة، متى تنتهي، و متى تثبت أو تلغى.
- حالات المناقلة:
  - حالة النشاط
  - الحالة التثبيت الجزئي
  - حالة التثبيت
  - حالة الفشل
  - حالة الانتهاء

التثبيت

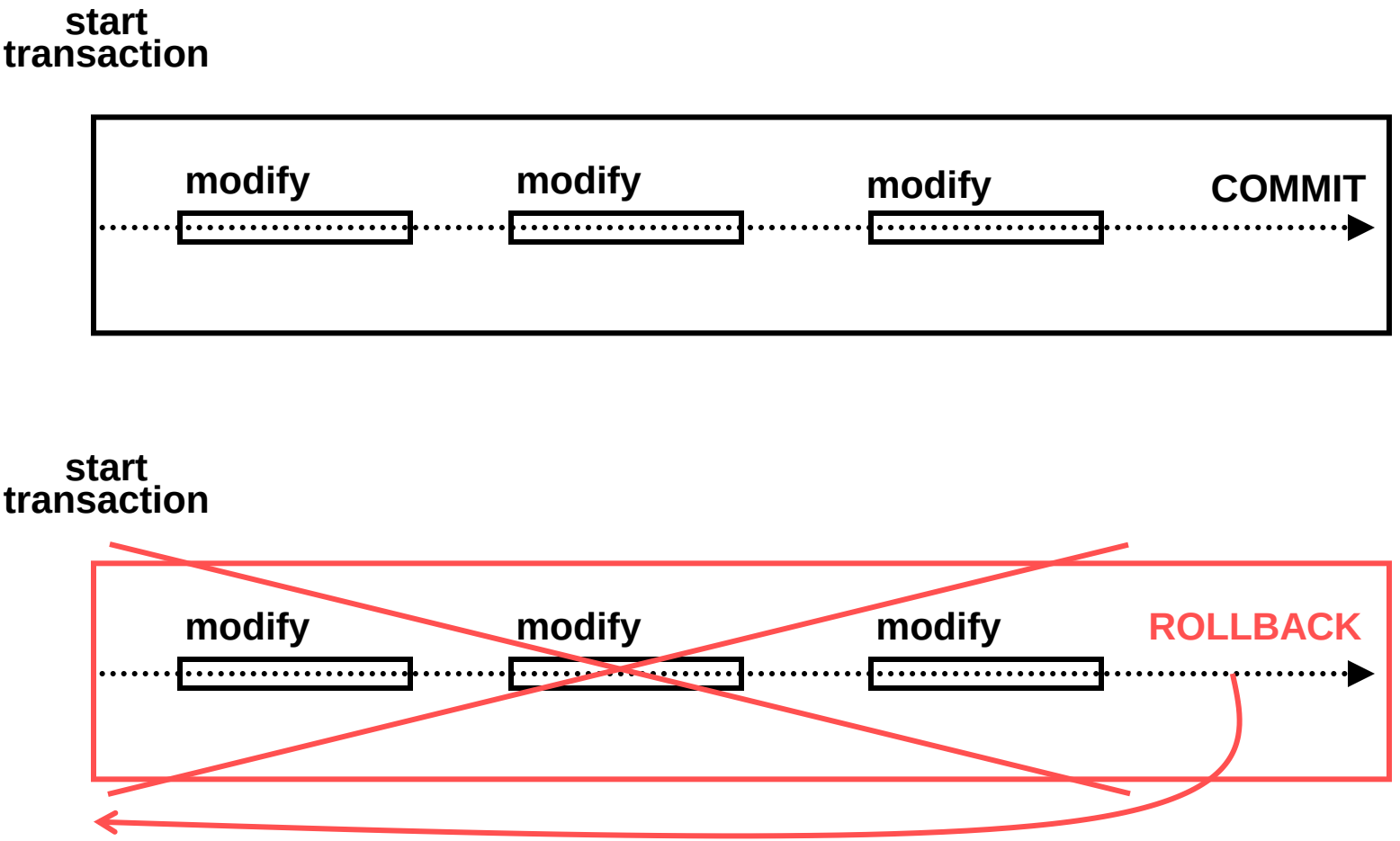


Modifications not yet persisted



Modifications persisted to DB

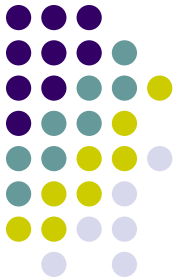
التراجع


start
transaction
modify
modify
modify
COMMIT
start
transaction
modify
modify
modify
ROLLBACK

حالات المناقلات

- يتتبع مدير الاستعادة (recovery manager) أثر العمليات التالية:

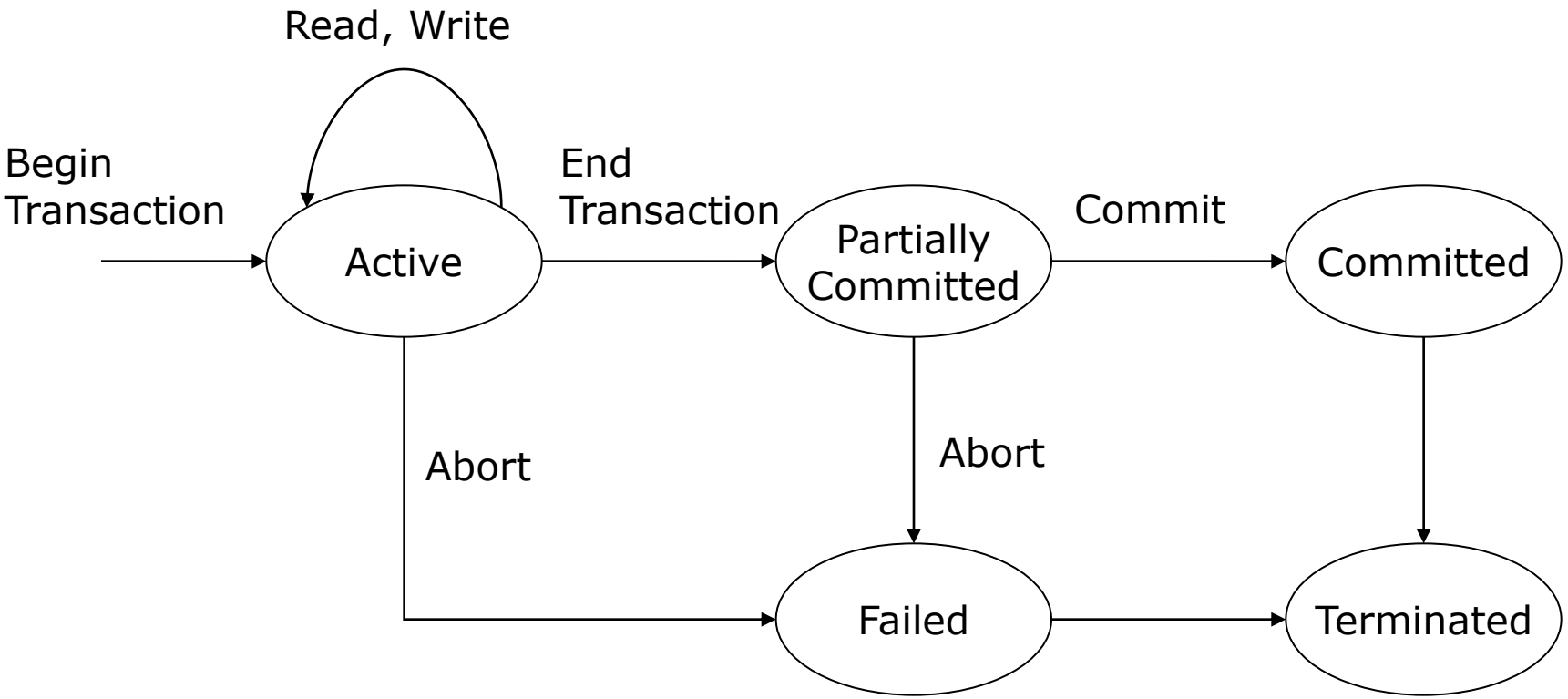
begin_transaction	تحدد بداية تنفيذ المناقلة
read or write	تحدد عمليات القراءة أو الكتابة على عناصر قاعدة البيانات، و التي تنفذ كجزء من المناقلة
end_transaction	تحدد أن عمليات القراءة و الكتابة قد انتهت و تحدد حد النهاية لتنفيذ المناقلة. في هذه النقطة من الضروري التحقق من كون التغييرات التي أجرتها المناقلة يمكن تطبيقها بشكل دائم على قاعدة البيانات أم يجب إلغاء المناقلة لأنها تنتهك التحكم بالتزامن أو لسبب آخر
commit_transaction	تشير إلى نهاية ناجحة للمناقلة و بالتالي أي تغييرات أجرتها المناقلة يمكن تثبيتها بأمان إلى قاعدة البيانات و لن يتراجع عنها
rollback (or abort)	تشير إلى أن نهاية غير ناجحة للمناقلة و بالتالي أي تغييرات قد تكون أجرتها المناقلة على قاعدة البيانات يجب التراجع عنها



حالات المناقلات

- تقنيات الاستعادة تستخدم العمليات التالية:
- UNDO
  - تشبه عملية rollback إلى أنها تطبق على عملية واحدة بدلاً من مناقلة كاملة
- REDO
  - تحدد أن عمليات معينة من المناقلة يجب إعادة إجراءها لضمان أن جميع عمليات مناقلة مثبتة قد طبقت بنجاح على قاعدة البيانات

حالات المناقلات



صحيفة النظام (System Log)

- تتتبع الصحيفة أثر كل عمليات المناقلات التي تؤثر على قيم عناصر قاعدة البيانات
- هذه المعلومات ضرورية من أجل السماح بالاستعادة من حالات فشل المناقلات.
- يتم حفظ الصحيفة على القرص الصلب، و بالتالي لا تتأثر بأي نوع من الفشل ما عدا فشل القرص
- يتم الحفظ الاحتياطي للصحيفة بشكل دوري إلى تخزين أرشيفي

صحيفة النظام (System Log)

- أنواع تسجيلات الصحيفة

[start_transaction, T]

تسجل أن المناقلة T قد بدأت التنفيذ

[write_item, T, X, old_value, new_value]

تسجل أن المناقلة T قد غيرت قيمة عنصر قاعدة البيانات X من قيمة قديمة old_value إلى قيمة جديدة new_value

[read_item, T, X]

تسجل أن المناقلة T قد قرأت قيمة عنصر قاعدة البيانات X

[commit, T]

تسجل أن المناقلة T قد اكتملت بنجاح، و تؤكد أن تأثيرها يمكن تثبيته (يسجل بشكل دائم) إلى قاعدة البيانات

[abort, T]

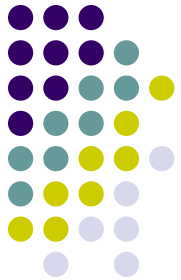
تسجل أن المناقلة T قد ألغيت

الاستعادة باستخدام تسجيلات الصحيفة

- إذا انهار النظام، يمكننا استعادة قاعدة البيانات إلى حالة متجانسة من خلال تفحص الصحيفة :
  - بما أن الصحيفة تحتوي تسجيلة من أجل كل عملية كتابة التي تغير من قيمة عنصر بيانات، من الممكن التراجع (undo) تأثير عمليات الكتابة هذه للمناقلة T من خلال التتبع الرجعي عبر الصحيفة و إعادة كل العناصر التي تغيرت بعملية كتابة من T إلى قيمها القديمة
  - يمكننا أيضاً إعادة إجراء (redo) تأثير عمليات الكتابة للمناقلة T من خلال التتبع التقدمي عبر الصحيفة و إعطاء كل العناصر التي تغيرت بعملية كتابة من T (و التي لم تتم بشكل دائم) إلى قيمها الجديدة.

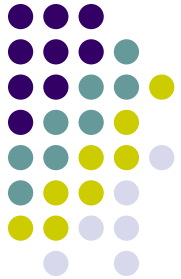
الخصائص المرغوبة في المناقلات – خصائص ACID

- الذرية (Atomicity)
- التجانس (Consistency)
- العزل (Isolation)
- الديمومة (Durability)



الذرية (Atomicity)

- التعديلات على قاعدة البيانات الموجودة في مناقلة يجب أن تنجز كلها أو لا ينجز أي منها
- المناقلة هي وحدة ذرية (لا يمكن تجزئتها) من العمل
- لا يمكنها أن تترك قاعدة البيانات في حالة وسطى:
  - حدوث فشل أو خطأ قبل التثبيت يؤدي إلى التراجع عن العمل المنجز
  - حدوث فشل أو خطأ بعد التثبيت قد يتطلب إعادة العمل المنجز، إذا لم يكن تأثيره على حالة قاعدة البيانات مضمونا



التجانس (Consistency)

- كل مناقلة يجب أن تترك قاعدة البيانات في حالة متجانسة
  - بالاعتماد على فرض قيود الحالة
- التجانس يتطلب أن لا تخرق المناقلة أي قيد كمال

- التحقق من قيود الكمال قد يكون:
  - فوري: أثناء المناقلة (نرفض العملية التي سببت الخرق)
  - مؤجل: في نهاية المناقلة (إذا تم خرق قيد كمال ما، نرفض المناقلة كاملة)

العزل (Isolation)

- كل مناقلة، عندما يتم تنفيذها بشكل متزامن مع مناقلات أخرى، يجب أن يكون لها نفس التأثير كما لو تم تنفيذها لوحدها
- يتطلب العزل أن تنفذ كل مناقلة بشكل مستقل عن تنفيذ كل المناقلات الأخرى المتزامنة
- يتطلب العزل أن يكون التنفيذ المتزامن لمجموعة من المناقلات يؤدي إلى نفس النتائج في حال التنفيذ التسلسلي لنفس العمليات

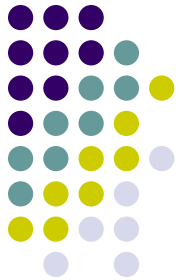
الديمومة (Durability)

- عند إتمام تنفيذ المناقلة بنجاح، التغييرات التي قامت بها على قاعدة البيانات يجب أن تصبح دائمة
- تتطلب الديمومة أن لا يفقد تأثير المناقلة التي تم تثبيتها بنجاح
  - (أن يدوم التأثير إلى الأبد)

مشاكل التزامن التقليدية

- مشكلة التعديل المفقود (Lost Update Problem)
- مشكلة القراءة القذرة (Dirty Read Problem)
- مشكلة التلخيص غير الصحيح (Incorrect Summary Problem)
- مشكلة القراءة غير المكررة (Non-Repeatable Read Problem)
- مشكلة التعديل الشبح (Ghost Update Problem)

مشكلة التعديل المفقود (Lost Update Problem)



- التعديل الذي تنجزه مناقلة يغطي على تعديل مناقلة أخرى
- تحدث هذه المشكلة عندما تصل مناقلتان إلى نفس عناصر قاعدة البيانات و تتداخل العمليات بشكل يجعل قيمة بعض عناصر البيانات غير صحيحة

Time ↓

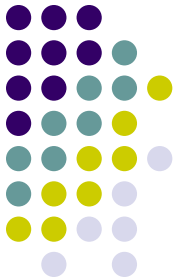
T_1	T_2
read_item(X);	
X := X - N;	
	read_item(X);
	X := X + M;
write_item(X);	
read_item(Y);	
	write_item(X);
Y := Y + N;	
write_item(Y);	

X = 80
N = 5
M = 4

⟹ X = 84 ✘

X = 79 ✓

مشكلة القراءة القذرة (Dirty Read Problem)



- مناقلة تقوم بقراءة بيانات غير مثبتة لمناقلة أخرى
- تحدث هذه المشكلة عندما تعدل مناقلة عنصر بيانات، ثم تفشل المناقلة لسبب ما، ولكن تم الوصول إلى العنصر المعدل من قبل مناقلة أخرى قبل أن يعاد العنصر المعدل إلى قيمته الأصلية

T_1	T_2
read_item(X);	
X := X - N;	
write_item(X);	
	read_item(X);
	X := X + M;
	write_item(X);
read_item(Y);	
rollback	

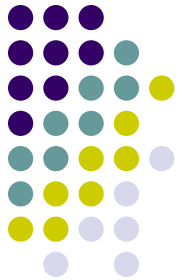
Time ↓

X = 80
N = 5
M = 4

⟹ X = 79 ✘

X = 84 ✓

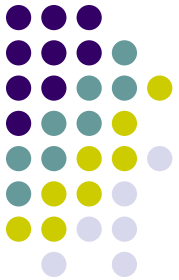
مشكلة التلخيص غير الصحيح (Incorrect Summary Problem)



- تحدث هذه المشكلة عندما تقوم مناقلة بحساب تابع تلخيص تجميعي على مجموعة من السجلات بينما تقوم مناقلات أخرى بتعديل بعض هذه السجلات.
- قد يقوم تابع التجميع بحساب بعض القيم قبل تعديلها و بعض القيم بعد تعديلها، و بالتالي تصبح نتيجة الحساب غير صحيحة

T_1	T_3
	sum := 0;
	read_item(A);
	sum := sum + A;
	.
read_item(X);	.
X := X - N;	.
write_item(X);	
	read_item(X);
	sum := sum + X;
	read_item(Y);
	sum := sum + Y;
read_item(Y);	
Y := Y + N;	
write_item(Y);	

مشكلة القراءة غير المكررة
(Unrepeatable Read Problem)

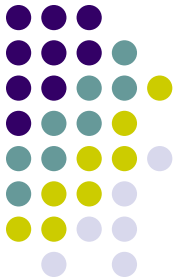


- تحدث هذه المشكلة عندما تقرأ مناقلة عنصر بيانات مرتين، و يكون العنصر قد تغيرت قيمته من قبل مناقلة أخرى بين القراءتين

T_1	T_2
read_item(X);	
	read_item(X);
	X := X + M;
	write_item(X);
read_item(X);	

Time ↓

مشكلة التعديل الشبح (Ghost Update Problem)



- تحدث هذه المشكلة عندما تلحظ مناقلة فقط بعض تأثيرات مناقلة أخرى، و بالتالي تلحظ حالة لا تحقق قيود الكمال

X + Y + Z = 1000

T_1	T_2
read_item(X);	
	read_item(Y);
read_item(Y);	
	Y := Y - 100;
	read_item(z);
	Z := Z + 100;
	write_item(Y);
	write_item(Z);
read_item(Z);	
S = X + Y + Z	

Time

المناقلة T1 تلحظ تعديل شبح

X + Y + Z = 1100 ✘