トランザクション処理

データベースの整合性制約（1）

データベースが現実世界を正しく反映するために課せられる制約。

ドメイン制約：各属性の属性値は、属性のドメインに含まれる
いずれかの値でなければならない。

キー制約：主キーとして選択された属性において、同じ属性値が
リレーションの中で重複して現れることはない。

実体整合性制約：主キーとして選択された属性は空値をとらない。

（注）主キー：リレーションにおいて、タプルを一意的に特定する
候補キーのうち、空値をとらないもの（任意に1つ選択する）。

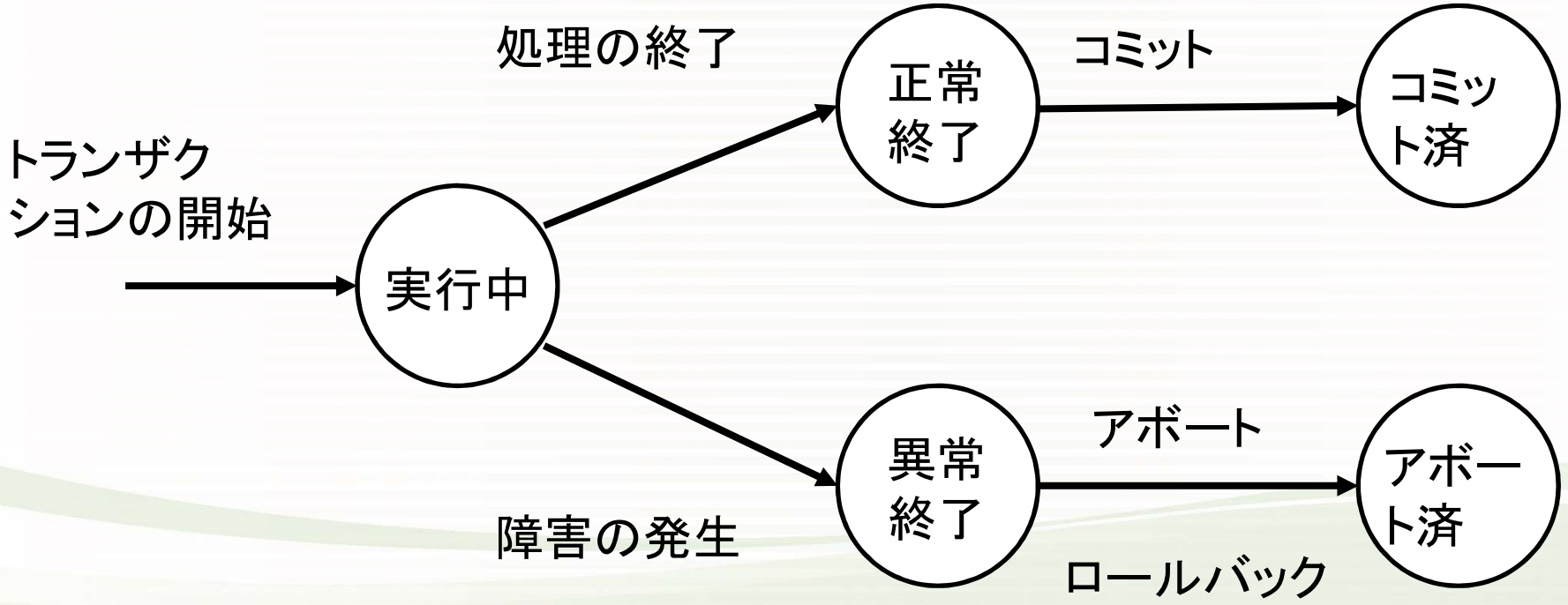
データベースの整合性制約 (2)

参照整合性制約：外部キー（他リレーションの主キーにある属性値をとる属性）の属性値は、空値の場合を除き、参照先の主キーの値でなければならない。

ユーザ定義制約：その他の制約

トランザクション

データベースの状態を、整合性のある状態から別の整合性のある状態に変化させるデータ操作の集合。



トランザクションの例

学生への仕送り:
父親の銀行口座Aから学生の銀行口座Bに10万円送金する。

データ操作1:
銀行口座Aの残高を更新する(10万円減らす)。

データ操作2:
銀行口座Bの残高を更新する(10万円増やす)。

トランザクションのACID特性

原子性 (atomicity)

一貫性 (consistency)

分離性 (isolation)

持続性 (durability)

トランザクションの同時実行

不整合の例:

(a) 更新喪失:トランザクションT2による更新が
トランザクションT1による更新で上書き

(b) ダーティリード:T1のコミット/アボートが
決まる前に、T2がT1の更新結果を読み出す

トランザクションの同時実行

不整合の例:

(c) 非再現リード:トランザクションが同じデータを
繰り返し読み出すときに、値が同じにならない

(d) ファントムリード:最初の検索で存在しなかった
データが次の検索で読み出される

解決法としては、直列スケジュールや
直列化可能スケジュール

ロックによる同時実行制御

共有ロック：データの読み出しを行うためのロック。

排他ロック：データの更新を行うためのロック。

すでにロックされたデータに対して、他のトランザクションがロックを要求した場合、両方とも共有ロックであればロックが許可される。

データベース障害

トランザクション障害：ACID特性が保証できなかった場合。

システム障害：システム全体が停止し、その時点での処理が強制的に終了される場合。

メディア障害：データを格納している記録メディアのトラブルで、データベースの全体もしくは一部が消失する場合。

データベースの障害回復

ログファイルを用いたロールバックを行う。

チェックポイント：その時点でキャッシュされている
すべてのデータを外部記憶装置に書き出す処理を実行する
時刻。

障害が発生する直前のチェックポイントを見つけ、ロール
バックして、チェックポイント時点のデータを再現。
さらに、チェックポイントから障害発生時までにコミット
されたトランザクションは、ログに書かれた処理をデータに
反映させて処理を再現する（ロールフォワード）。

バックアップデータ

フルバックアップ

差分バックアップ

増分バックアップ

RAIDによる冗長化

関係データベース設計と操作言語

データベースの必要性

ファイルシステムの管理の問題点：

・目的のデータがどこにあるかを発見するのが難しい。
・異なるアプリケーション間でのデータ共有が難しい。
・複数利用者に対するアクセス制限管理を行いにくい。
・共有利用されるデータへの同時アクセス制御を行いにくい。
・障害が発生し、データの一部が失われると、回復が困難。

→ 複数の利用者や異なるアプリケーションが大量のデータを
安全に共有し、用途に応じて必要なデータを管理できるよう、
データを一元管理したい

→ 設計したデータベースをコンピュータ上に構築・管理・運用
するためのシステムをデータベース管理システム(DBMS)という。

ANSI/X3/SPARCの3層スキーマ構造

データベースの構造：

内部レベル
　物理的な記憶装置（ストレージシステム）にデータベースを格納するためのデータ構造を規定

概念レベル
　データベースで管理しようとする実世界の情報を表現するためのデータ構造を規定

外部レベル
　個別のアプリケーションの利用目的に応じた見え方（ビュー）を規定

データベースの種類

ネットワーク型データベース

階層型データベース

リレーショナルデータベース（関係データベース）

オブジェクト指向データベース

半構造データベース

NOSQLデータベース

リレーショナルデータモデル

E. F. Codd が1970年に提案

タプル(tuple, 組):データやデータ同士の関係
リレーション:タプルの集まり。表として表せる

書籍シリーズ

ISBN	書名	判型	ページ数
978-4-123-45678-0	データ科学概論	B5	176
978-4-123-31233-1	機械学習入門	B5	223
978-4-123-78356-3	データベース詳解	A4	250

リレーションの数学的定義

リレーション名：表の名前に相当
タプル：行（横の並び）に相当
属性（attribute）：列（縦の並び）に相当
ドメイン：属性がとりうる値の範囲

リレーション名がRであるリレーションに対して、
n個の属性 A1, ..., An があり、それぞれのドメインが
D1, ..., Dn であるとする。

タプルとは、各ドメインから1つずつ選んだ属性値
の組である。すなわち、直積集合 D1 × … × Dn の
元である。リレーションはタプルの集まりなので、
この直積集合の部分集合である。

リレーションのキー

スーパーキー（超キー）：リレーションにおいて、
タプルを一意的に特定できる属性の集合

キー（候補キー）：スーパーキーのうち、真部分集合が
スーパーキーにならないもの

空値：タプルにおいて、ある属性値が決まっていない場合
に入れる。

主キー：タプルを一意的に特定するキーのうち、空値を
とらないものを任意に1つ選んだもの。

リレーションの正規化

表の各行が共通の構造をしたタプルに対応する。
各行のデータは単純であり、分解できない値である。
（第1正規形）

第1正規形であって、さらにすべての非キー属性が
各候補キーに完全関数従属しているとき、第2正規形
という。

第2正規形であって、さらにすべての非キー属性が
どの候補キーにも推移的に関数従属しないとき、
第3正規形という。

例題：ある学校に通う1年生のデータベース

学生

学籍番号	クラス	氏名	電話番号	住所
210101	1組	鈴木一郎	03-3456-7890	東京都
210102				

クラス

クラス	担任	副担任
A	高橋花子	伊藤太郎
B		

リレーショナル代数

単一または複数のリレーションに対して定義された
演算の体系

集合演算
　和集合、差集合、共通集合、直積

関係演算
　射影、選択、結合、商

SQL

リレーショナルデータベースを操作するための共通言語。
リレーショナルDBMSはSQL文の実行機能をもつ。

リレーショナルデータモデル vs SQL （用語の違い）

リレーション: テーブル
タプル: 行
属性: 列

SQLでは、テーブルの行と列に順序があるほか、
同じ内容をもつ行を重複してテーブルに入れることも
可能である。

SQL文

データ定義言語 (DDL): テーブルの枠組みを定義

create table テーブル名

drop table テーブル名

SQL文

データ操作言語 (DML): テーブルへのデータ挿入、更新、参照

insert into テーブル名 values (値1, ..., 値n)

select 列1, ..., 列m from テーブル名 where 条件

update テーブル名 set 列名1 = 値1

delete from テーブル名 where 条件

SQL文

データ制御言語 (DCL)：データへのアクセス権限を指定

grant 権限 on テーブル名 to ユーザ名

revoke 権限 on テーブル名 to ユーザ名

例題：ある学校に通う1年生のデータベース

学生

学籍番号	クラス	氏名	電話番号	住所
210101	1組	鈴木一郎	03-3456-7890	東京都
210102				

クラス

クラス	担任	副担任
A	高橋花子	伊藤太郎
B		