帝塚山学院大学 TEZUKAYAMA GAKUIN UNIVERSITY

データベース概論
まとめ

中野秀男
情報メディア学科/ICTセンター長

今日の話

- 講義のスライド等を整理して話します
- 情報メディア学科の科目体系の中の「データベース概論」
- 情報社会における各種サービスとデータベース
- 図書館と検索エンジンとデータベース
- データ：実体、データの性質：属性、そしてデータの集合
  - 簡単なものならExcelやAccess
- 複数の利用者が同時に利用、権限制御などでSQLで統一
- レベルに応じて
  - Excel, Access, MySQL, PostgreSQL, SQLserver
- 検索、ACID特性、トランザクション
- 教科書は「データベース-ビッグデータ時代の基礎-」
  - 共立出版 未来へつなぐデジタルシリーズ

科目体系

情報社会におけるデータベース

- 教科書の図1.1
- 各種サービス
  - オンラインショッピング
    - 顧客データベース、在庫データベース、販売データベース
  - ネットバンキング
    - 口座データベース
  - 会社のデータベース
    - 決済データベース、財務データベース、人事給与データベース
  - 学校のデータベース
    - 学生データベース、成績データベース
- それぞれのサービスが連携
  - 会社はオンラインショッピングして、ネットバンキングで決済

データベースはデータの集まりを扱う

- データって
  - 実体(entity)
  - レコード(record)
- 具体的には
  - 学生
  - 受講科目
- データには性質(属性)がある
  - 学生：学籍番号、姓名、生年月日、性別、住所、出身校
  - 受講科目：科目名、状態(履修中か)、成績、教員、教室、種別
- 属性はフィールド(field)
- データが集まってテーブル、表、Book(Excelだと)

検索エンジンとデータベースシステム

- 「検索エンジンはなぜ見つけるのか—知っておきたいウェブ情報検索の基礎知識」森 大二郎著、日経BP社
- 最善の情報を最低のコストで最大多数の人に提供する
  - 近代図書館と検索エンジン
- 検索エンジン
  - 利用者の情報要求に対して、ウェブ上の情報資源の中から、最善の情報を見つけ出してくることを目標とするサービス
- 三つの要件
  - もれなく、すばやく、的確に
- そのためには
  - 集める、整理する、検索する
- データベースはこれに処理が加わる

テーブル、レコード、フィールド

- 具体的な例で
- レストランのデータベース設計(テーブル設計)
- レコード
  - 店員、お客、料理
- フィールド
  - 店員：性別、年齢、姓名、勤務日時、時給
  - お客：性別、おおよその年齢
  - 料理：カテゴリー、料金、出す時間帯
- テーブル
  - マスターテーブル：店員台帳、お客台帳、料理台帳
  - データテーブル：注文データテーブル

今までの整理(1)

- データベースは集める、整理する、処理する
  - 集めて整理する
  - 高速に処理したり、不具合がないように整理する
  - 簡単なものならExcelやAccessで十分
- データ：実体(entity), レコード
  - 実体は属性を持っている, フィールド
  - 実体が集まって表に
- 実際の情報システムは複数のデータの集合と関係を持っている
  - 履修モデル：学生、科目、教員、学科、学科長
  - 眼鏡屋：品物、顧客、販売、店員

Excel

- 行(row)と列(column)、セル(cell)、全体がブック(book)
- セル：文字列、数字、計算式、マクロ(VBA)
- マクロ：VBA
  - VBA:Visual Basic for Applications
  - ExcelやAccess等のアプリケーションの操作を自動化する
  - 条件を判断して処理内容を分けられる
  - 同じ処理を何度か繰り返すことができる
  - フォームを使ってユーザからの指示ができる
- グラフ表示
  - 目的にあったグラフ表示ができる
- VBA(Access等も含めて)やグラフ表示に多くのノウハウが

Access

- 表面的にはExcelのように行と列
  - Accessでは行はレコード
  - Accessでは列はフィールド
- レコード毎に違う主キーを決める
- データ型
  - テキスト型、メモ型
  - 数字(いろいろな数字)
  - 日付型
  - 通貨型
- クエリ(問い合わせ)で条件にあったレコードを見つける
  - 例：生年月日が10月の学生を学籍場号順に見つける
- フォーム：入力画面を設計
- レポート：

ACCESSからみた業務(仕事)の流れ

- マスターテーブル
  - 業務の元となるテーブル
  - 顧客マスター、社員マスター、商品マスター、商品区分
- 売り上げ等のテーブル
  - 日々のもの(見積、売上げ)
  - 見積データ、見積明細
- クエリで条件にあったものや、演算(集計など)をする
- フォームで日々の業務にあった画面を作る
- レポートする

今までの整理(2)

- データの集合：表, テーブル
  - いろいろなデータの集合があるので、
    - それぞれに主キーを決めて、
    - 相互の関係は外部キーで関係をつける
  - マスターテーブル(台帳): 基本になるテーブル
  - データテーブル: 日々更新されるデータ
- SQL
  - 複数の利用者が同時に利用、権限制御などでSQLで統一
  - ネットを通して利用できるように：サーバとクライアント(端末)
  - いろいろな言語から使えるように
    - PHP, Perl, Python, Ruby, C, C++, C#, Java
- 高速に検索や処理のために便利なB-treeのデータ構造

リレーションスキーマ

- 3.2 リレーションスキーマ
- リレーション名:Accessだとテーブル、Excelだとブック
- リレーションはTupleの集合
  - Tupleは行
- Tupleは1つ以上の属性値(attribute value)を持つ
  - 属性は列
- 属性は取りうる範囲が決まっている
  - この範囲がドメイン(定義域、domain)
- 3.3.2 キー制約
  - 主キー 空値がない
  - 外部キー

5.SQL

- 5.1.1 標準言語としてのSQL
  - リレーショナルデータベースを扱う言語
  - 言語:SQLの場合は基本的には命令
- 5.1.2 SQLの仕様
  - リレーショナルデータモデルでは
    - リレーション、Tuple、属性
  - SQLでは
    - 表(Table), 行(row)、列(column)
  - テーブルの枠組みを定義するSQL文: create
  - データの操作をするSQL文: insert, delete, select
  - データのアクセス権限を指定するSQL文

ACID特性

- 本格的なデータベースに要求される条件
- A: atomicity(原子性)
  - トランザクションが更新後の状態になるか、まったく処理を行わなかった状態でおわるかのいずれか
- C: consistency(一貫性)
  - 整合性がトランザクションの完了後も保たれる
- I: isolation(分離性)
  - トランザクションが他のトランザクションに影響をあたえないように分離されて実行される
- D: durability(持続性)
  - 一旦コミットされたトランザクションの更新は、その後の障害などで失われることはない

いろいろなDBのテーブル

- 大規模コンビニチェーンのDB
  - コンビニストアが全国にたくさん
    - コンビニ毎のデータベース
  - 一箇所のデータセンターに全国のコンビニのデータを管理
  - 的確に管理と予測をして指示を
- いずれ、大手自動車会社のデータベースも
- 普通のお店のデータベースは

大規模コンビニチェーンのDB

- 大規模コンビニチェーンの各店舗のDB
  - 商品マスターテーブル
    - 商品バーコード
    - 商品名
    - 売り値
    - 在庫量
  - 販売データテーブル
    - 購入日時
    - 購入時刻
    - 購入商品・個数
    - 置いてある場所
    - 購入者種別
    - クーポン券利用の有無

大規模コンビニチェーンのDB

- 大規模コンビニチェーンのデータセンターのDB
  - 販売データテーブル
    - 店コード
    - 販売商品・数
    - 購入者種別
  - 店舗マスターテーブル
    - 店コード
    - 住所
    - 立地環境
    - 広さ
    - 店長

8.データモデリング

- 8.3 実体関連図(ER図)
  - E: Entity(実体), R: Relationship(関連)
  - (1) 実体、(2) 属性、(3) 主キーと外部キー、(4) 実体間の関連
- 履修モデル(図8.4) 関係は外部キーで
  - 学科:(主キーは学科ID),(関係:教員,学生)
  - 教員:(主キーは教員ID),(関係は学科,開講科目)
  - 学生:(主キーは学籍番号),(関係は学科,履修登録)
  - 履修登録:(主キーは学籍番号,科目ID),(関係は学生,科目)
  - 科目:(主キーは科目ID),(関係は教員,履修登録)
- 眼鏡屋の例(図8.6, 8.7)

9.データベース管理システム

- SQLのデータベース
  - Oracle
  - SQLserver(Microsoft)
  - MySQL
  - PostgreSQL
- インデックス方式
  - B-treeインデックス
  - ハッシュインデックス

検索

- データは整理されている
  - インデックス
- 検索アルゴリズム
  - 二分木探索(binary search)
  - ハッシュ探索(hash search)
- 二分木探索
  - データを順に並べておく
  - 真ん中のデータが探しているデータか？
    - OK: 検索終了(結果は真ん中のデータ)
    - 小さければ前半のデータセットでやり直し
    - 大きければ後半のデータセットでやり直し
    - データがない(結果は「データなし」)

トランザクション

- トランザクション
  - データベースの状態を、整合のある状態から、別の整合性のある状態に変化させるデータ操作の集合
  - 整合のない状態とは
    - 例：仕送り問題
- トランザクションはコミットされるか、アボートされる
  - コミット：正常終了
  - アボート：異常終了
- ロールバック
  - アボートした時に開始前の状態に戻す

ロック

- ロックによる同時実行制御
  - 操作対象のデータを施錠する
    - 例：同時に同じデータの書き換えが起こった場合など
  - 共有ロック：データ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
  - ロックの粒度：どの範囲でロックするか
    - 例：全体をロック、レコードをロック

MySQL

- 開発は一企業が行いデュアルライセンスで配布
  - 一企業：MySQL社 → Sun Microsystems社 → Oracle社
  - デュアルライセンス
    - コマーシャルライセンス
    - オープンソース: GPLv2(General Public License)
- LAMPのM
  - OS(Linux), Webサーバ(Apache), DB(MySQL)
  - 開発言語(Perl, PHP, Python)
- 多くのOSで動作
  - Windows, Linux, FreeBSD, Mac OS X
- いろいろな言語から利用可能
  - LL(Perl,Python,PHP),Ruby,Java,C#

PostgreSQL

- オープンソースのRDBMS
  - Relational DataBase Management System
- 1986年のPOSTGRESプロジェクトから
- 動作環境：LinuxベースのUNIX互換OS、Windows
- いろいろな言語からPosrgreSQLを使う
  - Java, Perl, Python, Ruby, ,Net Framework, Excel
- データベースの役割
  - データの管理
  - トランザクション管理、同時実行制御
  - セキュリテイ機能
  - 障害回復管理
- 開発のコアメンバー
- PostgreSQLユーザ会

SQL Server

- Windows ServerとWindowsで動作するRDBMS
- SQL Server: 2000, 2005, 2008, 2008 R2, 2012, 2014
  - それぞれにMicrosoftからサポート終了期限が
- 特徴
  - 使いやすさ
  - データ転送(インポート、エクスポート)/データ分析ツール
  - 開発ツール(Visual Studio)との連携
- 価格

今までの整理(3): Excel

- 表はBook
- 行がレコード
- 列がフィールド
  - 列を対応させることで他のアプリとインポートやエクスポート
  - csv形式
- だが行と列を変えたりできる
- 基本はセルで、セルの属性は数字、文字列、計算式
- 表レベルで良いとか、一人で利用する場合はExcelで十分
- VBAでさらに使いやすくなる
- 多くの利用者が同時に利用する場合はだめ
- 複数の表が相互に関係する場合もだめ

今までの整理(4): Accesss

- 表はテーブル
- 行がレコード
- 列がフィールド
  - フィールドには数字、テキスト、日付などの属性を持たせる
    - Excelの場合は数字と文字列と計算式だけ
- 複数のテーブルが作れて、各テーブルに主キー
- 各テーブル間の関係のリンクができる
- フォームで便利な入力画面が作れる
- レポート機能でいろいろなレポートが出せる
- 多くの利用者が同時に利用する場合はだめ

今までの整理(5): 本格的なデータベース

- MySQL, PostgreSQL, SQLserver, Oracle
- 表はテーブル
- 行がレコード
- 列がフィールド
- 複数のテーブルが作れて、各テーブルに主キー
- テーブルは外部キーを含ませることで他のテーブルと関連
- SQLをサポートしているので
  - テーブルの枠組みを定義するSQL文: create
  - データの操作をするSQL文: insert, delete, select
  - データのアクセス権限を指定するSQL文
  - 複数の言語から利用できる
  - ネットを通していろいろな場所からデータベースにアクセスできる
- 多くの利用者が同時に正しく利用できる

今までの整理(6): 本格的なデータベース

- MySQL, PostgreSQL
  - 両者ともオープンソースソフトウェア(OSS)
  - MySQLは商用ラインセンスもある
  - PostgreSQLは完全にコミュニティベースのソフトウェア
- SQLserver
  - Windows ServerやWindowsで動くソフトウェア
- Oracle
  - 業界標準的なデータベース
- DB2

ACID特性

- 本格的なデータベースに要求される条件
- A: atomicity(原子性)
  - トランザクションが更新後の状態になるか、まったく処理を行わなかった状態でおわるかのいずれか
- C: consistency(一貫性)
  - 整合性がトランザクションの完了後も保たれる
- I: isolation(分離性)
  - トランザクションが他のトランザクションに影響をあたえないように分離されて実行される
- D: durability(持続性)
  - 一旦コミットされたトランザクションの更新は、その後の障害などで失われることはない

半構造

- SQLの限界
  - あいまいな記述ができない
    - 例：メールアドレスが二つある
    - 例：携帯電話を二台持っているので電話番号が二つある
  - リンクしとけば簡単に検索できるのに、そんな機能がない
- 半構造
  - 自己記述にしておく：データにラベルをつけておく
  - リンクも記述する

13.1 マルチメデイア

- 一般にマルチメディアは
  - シンプルテキスト: plain text
  - 書式付きテキスト: rich text
  - 静止画像、動画像、音声、音楽
- 非テキスト型メディア
  - 静止画像、動画像、音声、音楽
- 非テキスト型メディアのデータベースの特徴
  - データ量が膨大
  - コンピュータが扱いにくいデータ
- メタデータ
  - マルチメディアの中身を表すデータ

13.2 マルチメデイアDBシステムの困難性

- 画像データベースの処理の流れ
  - 1.ユーザが「どんな画像が欲しいか」をDBに伝える
  - 2.DBシステムがユーザの検索要求を解析
  - 3.DBシステムがユーザの要求に近いメタデータを探し選択
  - 4.DBシステムが検索結果(候補画像)をユーザに提示
- 要求の難しさ
  - 要求があいまい
- 13.3 Web画像を対象としたメタデータ自動生成

Googleのデータベース

- 世界中のウェブを自動巡回プログラム(ロボット)で収集
- 安価なコンピュータや補助記憶でDBを構成
- 列(フィールド)の単位で整理せずに、行の単位で整理
- 複数の行(文書)を複数のコンピュータに保存
- 検索要求(クエリ)はありそうな複数のコンピュータに要求
- 多くの同時にくる大量のクエリに平均的に対応
- シャーディング：データの断片化
- レプリケーション：複製
- ページランク

CAP定理

- 多くのコンピュータでデータベースを処理することになったことででてきた考え方
- 以下の3つを同時に満たすことはできない
- 整合性
  - 更新クエリの実行のあとは常に最新のデータが検索できる
- 可用性
  - いつクエリが来ても実行できる
- 分断耐性
  - 障害でデータが分断されてもデータベースとして動作する