

データベースS

第1回 データベースシステム

システム創成情報工学科 尾下 真樹

2018年度 Q2

授業担当

- 尾下 真樹 (おした まさき)
 - e-mail: oshita@ces.kyutech.ac.jp
 - 居室: 研究棟 W623
 - オフィスアワー: 水曜日 11:00～12:30
 - 研究内容
 - コンピュータアニメーション技術の研究
 - キャラクタの動作生成・制御、操作インターフェース、物理シミュレーション、アニメーション制作システム



今日の内容

- ガイダンス
 - 授業の達成目標、進め方、成績評価、授業内容
- データベースシステムの概要
- リレーショナルデータモデルの概要
- データベースシステムの応用

授業の概要

- データベースの基礎知識を学習(講義)
 - 講義形式で説明
 - 基本的にPowerPointの資料を使って説明
 - 毎回の講義中に演習問題
 - 期末テスト
- データベースを使えるようになる(演習)
 - PostgreSQLの使い方を講義形式で説明
 - 各自、CL端末室で実際にデータベースを作成
 - 期末レポート

本科目の達成目標(シラバスより)

- リレーショナルデータベースを扱う上で必要な、スキーマの設計方法やSQLの使い方などの基礎的な知識を理解させる。
- リレーショナルデータベースの内部で用いられる、データ格納方式や高速化のための基礎的な技術を理解させる。
- データベース設計・操作を体験させ、データベースを利用するための基礎的な技術を習得させる。

本科目の位置づけ

- 情報の必修科目
 - 本科目の単位を修得しなければ、卒業できない
- 学科の学習・教育目標(B)に対応
 - コンピュータ応用とシステム理論を学び、時代の要請に呼応した新たな情報システムを創造し、開発を行うための基礎能力を身に付ける

特殊な履修の注意

- **重複履修は、一切認められない**
 - 同一時間の1・2年生科目再履修との重複履修は、一切認められない
- **上級年次科目履修は、条件を満たせば許可**
 - 留年した2年生が、3年生の必修科目を履修
 - 同一時間に再履修科目がなく、今年度に進級の見込みがある場合に限り、履修を認める（本科目のルール、他の科目は担当教員次第）
 - 希望者は、次回の授業の後にまとめて対応するので、次回の授業に書類を揃えて来ること

履修に関する注意

- **授業に関する質問や、履修関連相談、書類手続き(押印)は、オフィスアワーのみ対応**
 - 研究棟 6F 西棟 W623 oshita@ces.kyutech.ac.jp
 - オフィスアワー: **毎週 水曜日 11:00~12:30**
 - オフィスアワー以外の時間は一切対応しないので、締切がある手続きの相談には余裕を持って来ること

講義の進め方

- PowerPointの資料を使って説明
- 資料は Moodle の本講義のページで公開
 - <http://ict-i.el.kyutech.ac.jp>
 - ハードコピーが必要な人は各自印刷する
 - 演習に必要な資料なども、前もって置いておく
 - 講義に関する連絡も、Moodle上で行う
- 講義の聴き方は、各自工夫すること
 - 重要な点は、適宜、ノートをとるようにすること
 - 資料を印刷しただけで、安心しないこと

成績評価

- 期末試験(40点)
 - 期末レポート(40点)
 - 毎回の授業の演習問題・演習課題(20点)
 - 授業中の演習問題、データベース演習課題
 - 出席
 - 成績には考慮しない、**一定回数の欠席で不合格**
- ※ 上記の説明以上の具体的な評価方法や配点、個別の成績評価状況に関する質問には、回答しない

期末試験・レポート

- **期末試験**
 - 授業で扱う全ての範囲から出題
 - 教科書・ノート持込 **不可**
- **期末レポート**
 - 与えられた演習課題を行ってレポートを提出
 - 期末試験後の締切を予定

演習問題

- **毎回の講義中に、復習のための演習問題を実施する**
 - 出席の確認もかねる
 - 点数は、成績に反映する
 - 全成績の20% (1回分で1・2点程度)
 - 1回分はそれほど大きくはないが、積み重ねは大きい
 - 講義をきちんと聞いていれば解ける問題を出題
- Moodleにも同じ演習問題を用意するので、**復習に活用すること**

演習問題の手順

- **マークシートによる演習問題**
 1. 講義中にマークシート・演習問題を配布
 - 配布の時点で、講義を聞いていない人(遅刻、退席、睡眠など)には、一切配布しない
 - いちいち言い訳を聞いているときりがないため
 - トイレ等は休み時間に済ませる
 2. 演習問題の回答をマークシートに記入し、回収時に提出する
 - 同じく、回収の時点で、講義を聞いていない人の分は、回収しない
 - 後から持って来て、一切受け取らない
 - 回収開始後も記入している人の分は、受け取らない

クリッカーの利用

- 講義中に、演習問題の答え合わせに利用
 - クリッカーを利用する回には、講義が始まる前に配布を行う
 - 指示をされたら、自分が回答した選択肢の番号のボタンを押す
- **きちんと授業に参加すること**
 - ただ出席するのではなく、きちんと聞いて理解し、考える
- **演習問題と同様、成績に考慮**



出席確認

1. 授業開始前に学生証(ICカード)で出席登録
 - 授業開始時刻以降に登録したものは無効
 - 授業の邪魔になるので、授業開始後に登録に来ないこと(開始時刻後の登録は、下記の2に関わらず、欠席扱いとする)
 - 学生証を不所持の場合は遅刻扱い(個別対応はしない)
 2. 演習問題への回答(マークシート)
 - 上記の1・2の両方を満たしたら、出席と判定する
 - 両方なしは欠席、2のみは遅刻(1/2回分の欠席)
 - 学修細則で認められる欠席・遅刻は、欠席・遅刻届けが提出されれば出席扱いとする
 - 本授業では、全体で1/3以上の欠席がなければ、欠席・遅刻届の提出は不要
- ※ 上記以上の詳細や、個別の出席状況に関する質問には回答しない
※ 出席不足のため不合格(0点)になった場合、万一、上記のルール範囲外で特別な事情がある場合には、成績公開に申し出ること

演習課題

- 何度かの講義では、講義後に各自で行う演習課題を出す
 - 指定された通りの演習を行い、回答・結果を指定のファイルに記入して、Moodleから提出すること
- 講義中の演習問題(マークシート・クリッカー)と合わせて成績評価に考慮
 - 合計で全成績の20%

Moodleの演習問題


- 講義中に行った演習問題 + 追加問題
- 復習に活用すること
 - 講義中に行った演習問題で間違えた問題等の復習をすること
 - 間違ったまま覚えると、演習問題の意味がない
 - 例年、きちんと Moodle の問題をやっている人ほど、試験の点数も高い
- **出席・成績には考慮しない**

教科書・参考書

- 「リレーショナルデータベース入門 [第3版]」
増永良文 著、サイエンス社 (3,200円)
- 「データベースシステム」
北川 博之 著、昭晃堂 出版 (3,200円)
 - なるべく買うことをすすめる
 - 内容はほぼ同じなので、どちらでも良い
 - 上の本の方が例が多い、やや新しい
 - 最近、改訂版(第3版)が刊行
 - 下の本の方が文章は分かりやすいかも、やや古い




演習の参考書

- 「PHP5 徹底攻略」
堀田 倫英、桑村 潤 著
ソフトバンクパブリッシング (3,800円) 
- 「PostgreSQLによるLinuxデータベース構築」
廉 升烈 著、翔泳社 出版 (2,200円)
 - 演習では基本的に最低限の資料は用意するので無理に買わなくても良い
 - 自分でいろいろやりたい人、自宅のパソコンで演習をやりたい人向け

学科端末室(CL) 利用案内

- Computer Laboratory
 - 研究棟 西棟 6階 北側
 - CL (W605): 端末32台
 - 平日のみ(週末・休日は閉室)
- 利用案内
 - システム創成プロジェクトI 等の授業で使用
 - アカウントは情報科学センタと共通
 - Windows 8、環境は情報科学センタとは異なる
 - トラブル等があれば、E526(技術職員室)へ

データベースの概要

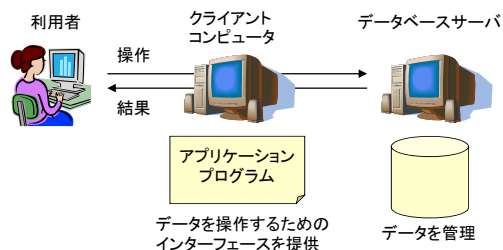
- データベースって何だろう？
 - 大量のデータを効率良く管理するためのシステム
 - データのモデリング、検索インターフェース、大量データ処理などの機能を提供
 - 他のプログラムからデータを利用できる 
- データベースの応用
 - 企業の顧客・売上データなど
 - 最近では、地理データベース、マルチメディア、DNAデータ、科学技術データなども扱われる
 - SEなど情報系に就職する人の多くは、何らかの形でデータベースシステムに関わる可能性が高い

データベースの社会での応用

- 店頭で販売されるソフトウェアはごく一部
- 実際には、企業からの注文で、企業内部で使われるソフトウェアの開発が多い
 - 企業のデータ(顧客・売り上げ・業務情報など)を管理するようなソフトウェア
 - データの管理部分にはデータベースシステムを利用し、ユーザーインターフェースのみを開発
 - データベースの設計・構築・最適化などの作業
 - データベースと密接に関わる仕事が多くなる

データベースの応用のイメージ

- データベース と アプリケーションプログラム



情報系の職種(おおまかな分類)

- プログラマ(コーダー)
 - 指示された通りにプログラムを書く人
- 開発者・設計者、システムインテグレータ
 - どのようにシステムを作るか設計をし、プログラムも書く
- システムエンジニア
 - データベースの構築や管理
 - コンピュータやネットワークの管理
 - 利用者の要求を聞き、ソフトウェアの仕様をまとめる
 - プログラマや設計者・開発者の仕事をすることもある
- プロジェクトマネージャ
 - システムの開発を管理

本科目はどのような役に立つか

- この授業を受講したからと言って、何らかの企業に簡単に就職できるようなことはない
- 本科目の範囲内で扱うのは、ごく基礎的な知識・演習のみ
 - 情報工学の基礎知識として、最低限知っておくべき内容
- 自分でより勉強したい人のための出発点

リレーショナルデータベース

- データをリレーション(表、テーブル)の集合によって管理
- リレーション同士の演算によって、さまざまな処理を実現
- 問い合わせ言語SQL

リレーション

属性1	属性2	...
データ1	データ1	データ1
データ2	データ2	データ1
...

リレーションの例(悪い例)

履修情報

学生番号	氏名	科目番号	科目名	成績
0123001	織田 信長	01	データベース	60
0123001	織田 信長	03	グラフィックス	90
0123002	豊臣 秀吉	03	グラフィックス	80
0123003	徳川 家康	01	データベース	70
...	...			

このリレーションをそのままデータベースに格納すると、データの不整合が起きる可能性がある

リレーションの例(良い例)

学生		科目	
学生番号	氏名	科目番号	科目名
0123001	織田 信長	01	データベース
0123002	豊臣 秀吉	03	コンピュータグラフィックス
0123003	徳川 家康
...	...		

履修		
科目番号	学生番号	成績
01	0123001	60
03	0123002	80
01	0123003	70
...	...	

複数の適切なリレーションに分割して管理(正規化)
必要に応じて動的に結合

SQLによる問い合わせの例

‘データベース’の履修者全員の氏名と成績の一覧を出力

```
SELECT 氏名, 成績
FROM 学生, 科目, 履修
WHERE 科目.科目名 = 'データベース'
      AND 履修.科目番号 = 科目.科目番号
      AND 学生.学籍番号 = 履修.学籍番号
```

一種のプログラミング言語

簡単な記述で、さまざまな条件にもとづいてデータを検索できる

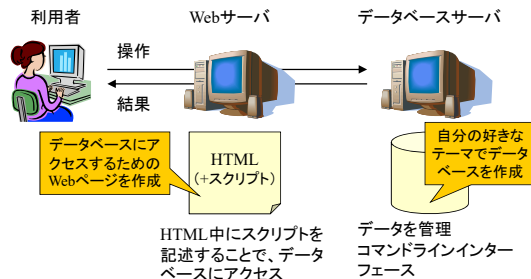
氏名	成績
織田 信長	80
豊臣 秀吉	70
...	...

演習内容

- PostgreSQL によるデータベース演習
 - リレーショナルデータベースシステムの基本的な利用方法を学習
- PHPによる Webインターフェースの開発
 - 前のスライドで説明したような、Webインターフェースを、PHPというプログラミング言語を用いて開発する
- 各自、独自のデータベースと Webインターフェースを開発して、レポートとして提出

演習内容

• データベース作成と Webインターフェース



講義予定

• 講義・演習

- 第1回 ガイダンス、データベースシステム
- 第2回 データモデル
- 第3回 リレーショナル代数
- 第4回 データベース言語 SQL (1)
- 第5回 演習: PostgreSQLによるデータベース構築
- 第6回 データベース言語 SQL (2)
- 第7回 リレーショナルデータベースの設計 (1)
- 第8回 リレーショナルデータベースの設計 (2)
- 第9回 リレーショナルデータベースの設計 (3)

講義予定

• 講義(続き)

- 第10回 演習: PHPによるWebインターフェース(1)
- 第11回 演習: PHPによるWebインターフェース(2)
- 第12回 物理的データ格納方式
- 第13回 同時実行制御
- 第14回 問い合わせ処理
- 第15回 障害回復
- 期末試験
- レポート提出

今日の内容

• ガイダンス

- 授業の達成目標、進め方、成績評価、授業内容

• データベースシステムの概要

- データベースシステムの機能

• リレーショナルデータモデルの概要

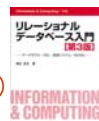
- スキーマとインスタンス

• データベースシステムの応用

教科書

• 「リレーショナルデータベース入門 [第3版]」

- 増永良文 著、サイエンス社 (3,200円)
- 1章(1.1~1.3)、2章(2.5)



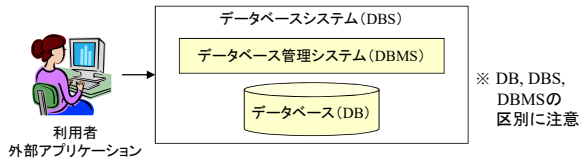
• 「データベースシステム」

- 北川 博之 著、昭晃堂 出版 (3,200円)
- 1章・2章

データベースシステムの概要

データベースシステム

- データベースシステム (Database System)
 - データベース (Database)
 - データを組織的・永続的に記録するための装置
 - データベース管理システム (Database Management System)
 - データベースを管理・利用するためのソフトウェア

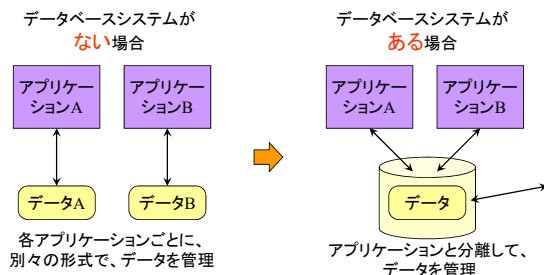


ファイルシステムとの比較

- 一般的なファイルシステムでも大量のデータを扱うことはできる
 - 例: Excelのファイルにデータを記述して保存
- ファイルシステムの問題点
 - データとアプリケーションの相互依存
 - データを作ったアプリケーションでないとデータの読み出しや修正ができない
 - データの整合性を一元管理できない
 - アプリケーションにデータ管理がまかされている
 - 複数のユーザの同時アクセスに対応できない
 - 機密保持や障害回復などの機能が不足

データベースシステムの利点

- データをアプリケーションと分離して管理



データベースシステムの利点

- データをアプリケーションと分離して管理
 - データをいろいろな目的に利用できるようになる
 - 関連するデータをひとつのデータベースに統合することで、重複や不整合を防げる
 - データだけを独立に整理することで、データの意味やその相互関係を把握することが容易になる
 - データの表現方法や管理方法を標準化できる

データベースシステムの特徴

- 大量のデータを効率的に管理・処理するためのしくみを提供
 - データを体系的・組織的に定義・管理するためのデータ記述方法(データモデル)、管理方法
 - 大量のデータを効率良く処理するための方法
- データを常に正しく保つためのしくみを提供
 - 実際の運用では、とにかく正しいデータを保つことが必須とされる
 - 整合性の維持、機密保護、障害回復

データベースシステムの機能

- 基盤となるデータ記述・操作系
- 効率の良いデータアクセス機構
- 整合性の維持
- 機密保護
- 同時実行制御
- 障害回復

基盤となるデータ記述・操作系

- データモデル(データ記述)
 - データベースに格納するデータ構造(スキーマ)を記述するための枠組み
 - 実際にどのようにファイルやメモリにデータが格納されるかといったことは気にせず、概念的なデータ構造を定義できる(詳しくは次回説明)
 - 各DBMSはある特定のデータモデルをサポート
 - リレーショナルモデルが代表的
 - これまでにさまざまなデータモデルが開発されてきた
 - データモデルに基づいた操作言語が存在

基盤となるデータ記述・操作系

- データベース言語(データ操作)
 - データモデルは、データの定義・作成・問い合わせ(検索)・更新などの操作のための言語を持つ
 - データベース言語による直接利用
 - 他のプログラミング言語からも利用可能
 - ODBC (Open Database Connectivity) や JDBC (Java Database Connectivity) などの標準インターフェース
 - リレーショナルデータベースの場合は、SQLがデータベース言語となる

効率の良いデータアクセス機構

- 指示された検索などの処理を効率的に実行
 - データモデルは、あくまで抽象的なデータ表現であり、物理的なデータ構造は規定しない
 - DBMSは最適な性能が得られるように実際のデータ構造などを工夫できる
 - ハードディスクへの配置・制御方法の工夫
 - ツリーやハッシュなどの補助的なデータの追加
 - 並列に処理を行うための工夫
 - データベースの利用者は、このような内部の構造を意識しなくともデータベースを使える

その他の機能

- 整合性の維持
 - データが常に正しい値を保つような制約を追加できる
- 機密保護
 - ユーザごとに実行できる操作を細かく設定可能
- 同時実行制御
 - 複数の処理を並列に処理できるような機能
- 障害回復
 - 途中で処理が中断しても整合性を失わないような機能
 - 故障が起きてもデータが失われないような工夫

リレーショナルデータモデルの概要

リレーショナルデータモデル

- リレーショナルデータモデル
 - 表形式のデータ構造(リレーション)によりデータを格納するデータモデル
 - リレーション同士の演算によって、さまざまな処理を実現できる
 - 他のデータモデルと比べて、単純、データ独立性が高い、といった利点がある
 - ただし、可変長のデータや、データ構造が複雑なデータには不向き

リレーショナルデータモデルの例

学生		科目	
学生番号	氏名	科目番号	科目名
0123001	織田 信長	01	データベース
0123002	豊臣 秀吉	03	コンピュータグラフィックス
0123003	徳川 家康
...	...		

履修		
科目番号	学生番号	成績
01	0123001	60
03	0123002	80
01	0123003	70
...	...	

スキーマとインスタンス

- スキーマ (scheme)
 - データベースに格納されるデータのデータ構造、データの型、データ同士の関連、各種制約を記述したもの
 - メタデータ(データについてのデータ)
- インスタンス (instance)
 - スキーマにもとづいて格納されたデータ
 - Javaのクラスとオブジェクトの関係と同じ (クラス→スキーマ、オブジェクト→インスタンス)

スキーマとインスタンス

- リレーショナルデータベースの例
 - リレーション
 - 複数の属性の組み合わせによりデータを表現
 - スキーマ
 - リレーションの項目の型、属性制約、キー制約など
 - インスタンス
 - それぞれのデータ、表の各行に相当

学生		履修		
学生番号	氏名	科目番号	学生番号	成績
0123001	織田 信長	01	0123001	60
0123002	豊臣 秀吉	03	0123002	80
0123003	徳川 家康	01	0123003	70

参考: Javaとの関連

- スキーマ・インスタンスの関係は、Javaのクラス・オブジェクトの関係に近い
- Java
 - クラスを定義(クラスがどのような属性を持つか)
 - データごとに new でクラスのオブジェクトを生成
- リレーショナルモデル
 - スキーマを定義(どのような属性を持つか)
 - ひとつのデータごとにインスタンスを追加

参考: Javaとの関連

Javaプログラムの例:

```

class Student
{
    public int    id;
    public string name;
    public Student( int i, string n ) { id=i; name=n; }
}
    
```

← クラスの定義

```

Student a, b, c;
a = new Student( 0123001, "織田 信長" );
b = new Student( 0123002, "豊臣 秀吉" );
c = new Student( 0123003, "徳川 家康" );
    
```

← オブジェクトの生成

データベースシステムの応用

データベースの応用

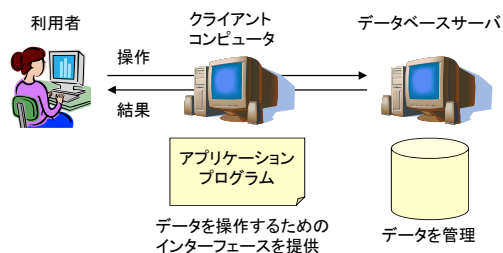
- データベースシステムの主な応用
 - 企業内部のデータ管理
 - 人事、在庫、販売、生産、顧客情報など
- 就職との関係
 - 情報系にシステムエンジニア(SE)などとして就職する人の多くは、実際にはデータベースを扱うことになる
 - 情報系の会社に限らず、ほとんどの会社では社内のデータをデータベース化する必要があるため、データベースを扱えることが期待される

データベースの応用

- データベースシステムの実際の運用形態
 - 会社や部署ごとに、データベースの専門家が、そこで管理されるデータに合ったデータベースを設計
 - データベースを設計するためには、業務の内容を熟知している必要がある
 - 業務の専門家とデータベースシステムの専門家の意思疎通が必要
 - ユーザのためのインターフェースも開発
- 現在使われているDBMSのほとんどはリレーショナルデータベース

データベースの応用のイメージ

- データベースとアプリケーションプログラム



リレーショナルデータベースの利点

- 運用実績
 - 企業ではとにかく信頼性が第一
 - これまで長い間使われてきている
- しくみが単純
 - データが表形式で表せるので簡単
 - 人間が出力を見たときに分かりやすい
 - それぞれの処理が明快
 - 手動でバックアップなどもできる

データベース応用の現実

- データベースシステムを使わない、COBOLなどの言語で書かれたプログラムも、まだ現役で使われている
- リレーショナルデータベースがほとんど
 - 商用システムとしては、Oracleが代表的
 - Oracleなどの商用システムは、SQLなどの操作言語とは別に独自の言語を提供している
 - 実際にはそれらシステム独自の使い方も勉強する必要はある

リレーショナルデータベース

- 商用のシステム
 - Oracle
 - 高速なコンピュータと大量のメモリ・ハードディスクの組み合わせにより高性能を実現
 - Microsoft SQL
- フリーのシステム
 - PostgreSQL
 - MySQL
 - 大量のデータを処理するのであれば、これらのシステムでも十分に実用的に使える
 - SQLite(組み込み型で動作する軽量なシステム)

データベースを使った応用

- **データマイニング**
 - データベースの中にあるデータから隠れた知識を見つけた技術
 - 例えば、コンビニの売り上げデータベースから、よく売れる商品の組み合わせを発見して商品陳列に利用するなど
 - 近年は特に、ビッグデータの解析技術が注目されている
- **OLAP (On-Line Analytical Processing)**
 - 数値データをさまざまな条件で対話的に集計
 - 例えば、商品ごと部門ごと期間ごとなどのいろんな条件で売り上げ結果を集計することで分析に役立てる
 - キューブモデルなどの特殊な物理データ構造が用いられる

データベースの新しい応用分野

- **エンジニアリングデータ (CAM: Computer Aided Manufacturing)**
 - 工業製品のモデルデータ
 - 機器のメンテナンス情報の管理など
- **科学技術分野での利用**
 - タンパク質データベース
 - DNAデータベース
- **地理データベース**
 - 地図作成、カーナビなど

マルチメディアデータベース

- **マルチメディアデータベース**
 - 前スライドで紹介した、3次元立体モデル、科学技術計算データ、地理データ、動画、音楽など
 - 従来のリレーショナルデータベースは、テキスト・数値データなどしか扱えない
 - 現在ではアプリケーションごとに特化したシステムが開発されている

まとめ

- **ガイダンス**
 - 授業の達成目標、進め方、成績評価、授業内容
- **データベースシステムの概要**
 - データベースシステムの機能
- **リレーショナルデータモデルの概要**
 - スキーマとインスタンス
- **データベースシステムの応用**

次回予告

- **リレーショナルデータモデル**
 - データモデル
 - リレーショナルデータモデル
 - リレーションの整合性制約