

インフォマティクス1

データベース

石川 佳治

情報学研究科 知能システム学専攻

ishikawa@i.nagoya-u.ac.jp

<http://www.is.i.nagoya-u.ac.jp/>



目次

- データベースシステムとは
- リレーショナルデータモデルとSQL言語
- データベースシステムの必要性
- ビッグデータの時代



背景

- 世の中には大量のデータが存在
 - 成績・履修データ
 - 人事・営業・販売データ
 - 文書, 写真, 動画データ
 - 科学分野の実験・観測データ
- データの有効利用が求められる
 - 多数のユーザによる共有
 - 高速な検索・問合せ
 - データの加工・再利用
 - 大量のデータの安定した保存
- 解決策: **データベースシステムの利用**



ショッピングサイトの例 (amazon)

Firefox Amazon.co.jp: データベース

www.amazon.co.jp/s/ref=nb_sb_noss_1?__mk_ja_JP=%83%83%83%5E%83%83%83%83&url=search-alias%3Daps&field-keywords=%83%81%5B

amazon.co.jp プライム

期間限定 **本ポイント15%** 4/30まで
Amazon Student > 今すぐチェック

こんにちは、石川 佳治さん
アカウントサービス

検索

カート

ほしい物リスト

カテゴリからさがす

すべて データベース

カテゴリ

本

- コンピュータ・IT
- データベース処理
- データベース
- ビジネス・経済
- データベースアプリケーション

+ 続きを見る

Kindleストア

- 工学
- アプリケーション

+ 続きを見る

+ 全5カテゴリ

配送オプション

プライム対象

著者

- 速水 治夫
- 五十嵐 貴之
- ミック
- 谷尻 かおり
- 小泉 修

+ 続きを見る

フォーマット

- 単行本(ソフトカバー)
- 単行本
- Kindle版

"データベース"

関連サーチ: データベース設計, データベーススペシャリスト, データライヴ

検索結果8,592件中1件から16件までを表示

並び替えを有効にするには カテゴリ を選択してください

なかに見! 検索

マンガでわかるデータベース 高橋 麻奈、あづま 笙子、トレンドプロ (2005/12)

¥ 1,995 単行本 **プライム**

47分以内にご注文いただくと、2013/4/25 木曜日までにお届けします。

12点在庫あり。ご注文はお早めに。

こちらからもご購入いただけます - 単行本

¥ 1,326 中古品 (18 出品)

★★★★☆ (17)

本: 全5,165商品を見る

なかに見! 検索

SQL ゼロからはじめるデータベース操作 (CD-ROM付) (プログラミング学習シリーズ) ミック (2010/6/29)

¥ 1,974 大型本 **プライム**

22時間以内にご注文いただくと、2013/4/25 木曜日までにお届けします。

こちらからもご購入いただけます - 大型本

¥ 1,190 中古品 (9 出品)

★★★★☆ (5)

本: 全5,165商品を見る

なかに見! 検索

データベース 4500 完成英単語・熟語 荻野 治雄 (2008/12)

¥ 1,040 単行本 **プライム**

22時間以内にご注文いただくと、2013/4/25 木曜日までにお届けします。

こちらからもご購入いただけます - 単行本

¥ 1,040 中古品 (23 出品)

★★★★☆ (23)

本: 全5,165商品を見る

14:43
2013/04/24

航空券予約サイトの例(ANA)



割引運賃		アップグレード対象運賃		Premium Economy シート利用		普通運賃	
選択	運賃	運賃種別	座席	滞在日数	予約変更	払戻(未使用時)	積算マイル
<input type="radio"/>	92,000円～	エコ割50	エコノミー	5日～1ヶ月	できません	できません	4,774マイル～
<input checked="" type="radio"/>	118,000円～	スーパーエコ割	エコノミー	3日～1ヶ月	できません	条件あり	6,684マイル～
<input type="radio"/>	139,000円～	エコ割7	エコノミー	2日～2ヶ月	できません	条件あり	6,684マイル～
<input type="radio"/>	303,600円～	エコ割スタンダードライト	エコノミー	2日～3ヶ月	できません	条件あり	6,684マイル～
<input type="radio"/>	323,600円～	エコ割スタンダード	エコノミー	最長6ヶ月	条件あり	条件あり	6,684マイル～
<input type="radio"/>	規則適用外	エコ割プレミアム ライト	Premium Economy (一部機材を除く)	最長4ヶ月	条件あり	条件あり	11,063マイル～
<input type="radio"/>	規則適用外	エコ割プレミアム	Premium Economy (一部機材を除く)	最長6ヶ月	条件あり	可能です	9,550マイル～
<input type="radio"/>	規則適用外	エコ割ロングステイ	エコノミー	65日～12ヶ月	条件あり	条件あり	-
<input type="radio"/>	355,000円～	スーパービジ割21	ビジネス	3日～1ヶ月	できません	条件あり	11,936マイル～

空席状況と連動した運賃

*5月10日10時36分(日本時間)時点での運賃です

ご希望のフライトを選択してください。

★おススメの旅
運賃合計: 118,000円(1名様分)
※各種税金・料金などが別途かかります
積算マイル: 6,684マイル

*出発時間、到着時間は現地時間です。

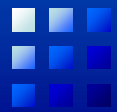
前日						翌日					
行きのフライト						帰りのフライト					
搭乗日	便名	区間	機種	空席状況		搭乗日	便名	区間	機種	空席状況	
5/24 (木)	NH7030	東京(成田) 16:55 ユナイテッド航空運航	シアトル 09:34	777	OK	5/30 (水)	NH7031	シアトル 12:56 ユナイテッド航空運航	東京(成田) 翌日 15:10	777	OK
5/24 (木)	NH008	東京(成田) 17:25 ANA	サンフランシスコ 10:45	77W	OK	5/30 (水)	NH7327	シアトル 07:32 ユナイテッド航空運航	サンフランシスコ 09:51	752	OK
5/24 (木)	NH7326	サンフランシスコ 13:47	シアトル 15:51	77W	OK	5/30 (水)	NH007	サンフランシスコ 12:25	東京(成田) 翌日 15:25	77W	OK





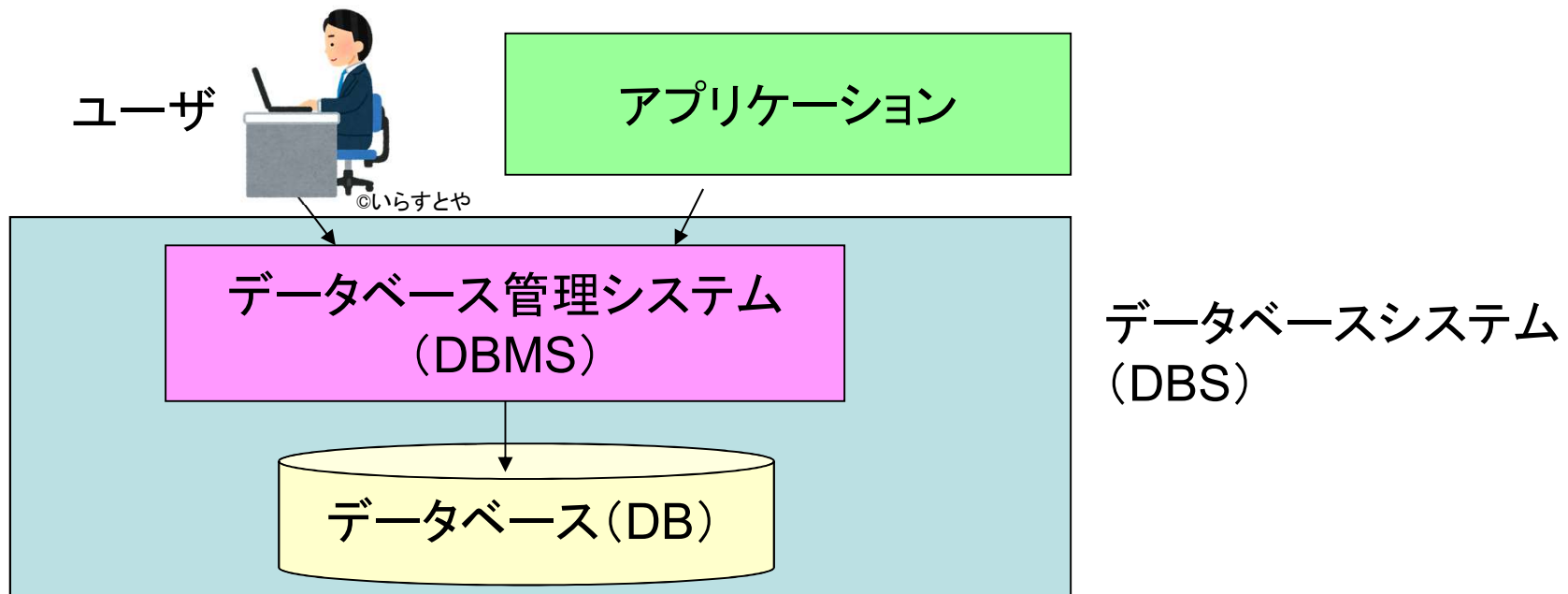
データベースシステムとは

- **データベースシステム** (database system)
 - 各種アプリケーションが扱うデータ資源を統合して蓄積管理
 - 効率的な共有, 高度な利用
- **アプリケーションシステムの例**
 - ウェブサイト: ショッピングサイトなど
 - 人事管理, 成績管理システム
- **データベース** (database, DB)
 - 複数の応用目的での共有を意図して組織的かつ永続的に格納されたデータ群



データベース管理システム

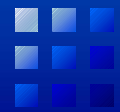
- **データベース管理システム** (database management system, DBMS)
 - データベースを管理するためのソフトウェア
 - データベースシステム: DBMSとそれにより管理されるデータ群





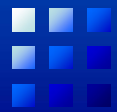
主要なDBMS

- **リレーショナルデータベース管理システム**
(RDBMS)と呼ばれるシステムが主流
 - リレーショナルデータモデル(後述)に基づく
- 商用のRDBMS
 - Oracle, Microsoft SQL Server, IBM DB2が三大DBMS
 - 日本では日立のHiRDB
- フリーソフトウェアのRDBMS
 - MySQL, PostgreSQL, Firebird
- 個人向けRDBMS: Microsoft Accessなど
- 組み込み向けRDBMS: SQLite
 - 実世界ではWindowsよりも多い数のSQLiteが動く

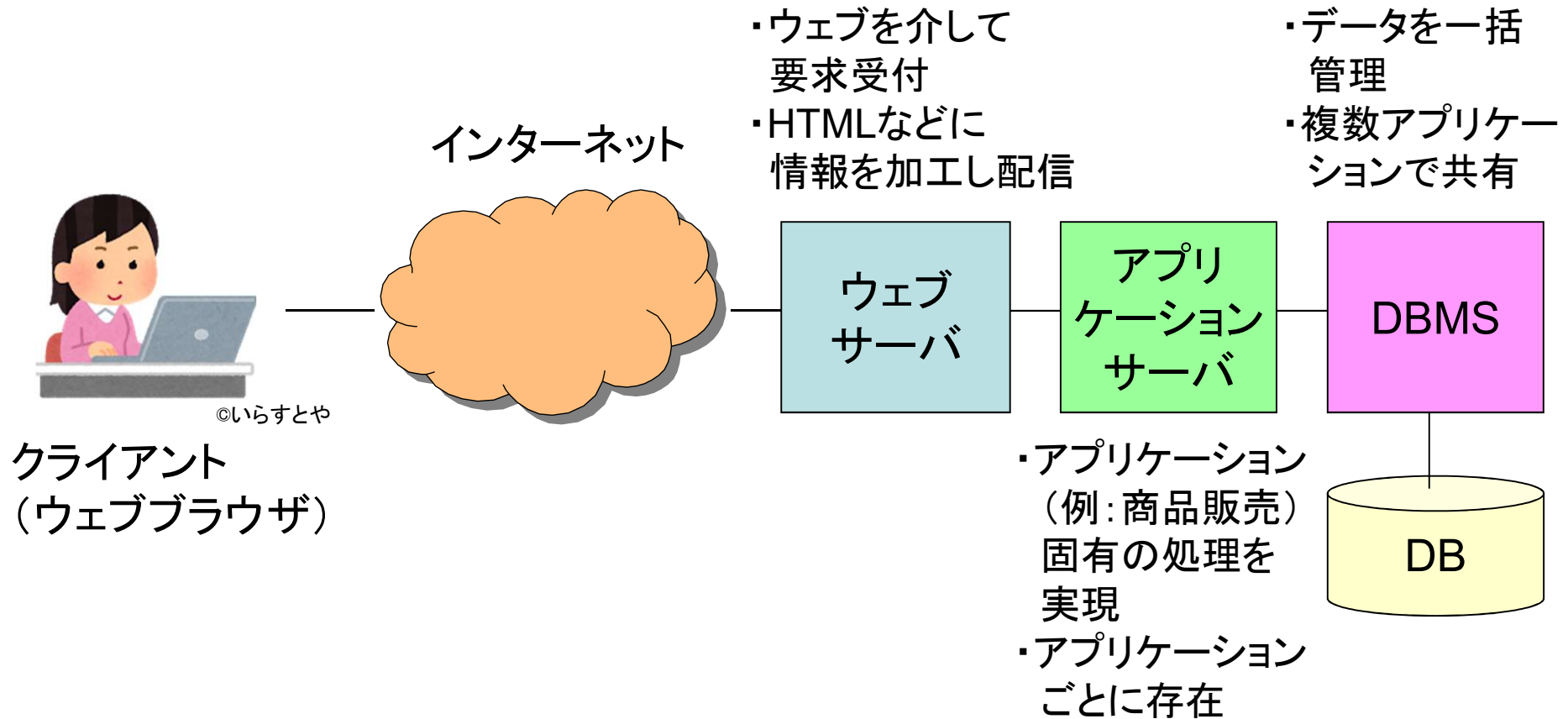


3階層モデル(three tier model) (1)

- インターネットサービスで一般的なシステム構成
- システムの拡張性, 柔軟性を実現
- 3階層のシステム構成
 - プレゼンテーション層(フロントエンド)
 - ウェブサーバによる実現: ユーザインタフェースを提供
 - アプリケーション層(ミッドティア)
 - アプリケーションサーバによる実現
 - アプリケーション固有の処理
 - データベース層(バックエンド)
 - DBMSによる実現: データを管理



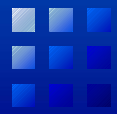
3階層モデル(three tier model) (2)





目次

- データベースシステムとは
- リレーショナルデータモデルとSQL言語
- データベースシステムの必要性
- ビッグデータの時代



データモデル(data model)

- 実世界に存在するデータは複雑
 - 例: CDの販売サイト
 - CDには, アーティスト, タイトルなどの情報がある
 - 顧客の情報(クレジットカードなど)の管理が必要
 - ...
- どのようにコンピュータ上に表現するか
 - 何らかの**枠組み**が必要:それが**データモデル**
 - データモデルに求められるもの
 - 表現能力:対象データをうまく表せるか
 - 実現可能性:コンピュータ上に効率的に実装できるか
 - 両者にはトレードオフが存在



リレーショナルデータモデル

- テーブル(リレーション)の形でデータを表現

CD

CD番号	アーティスト	タイトル	価格
00001	The Beatles	Abbey Road	2000
00002	The Beatles	Rubber Soul	1800
00003	Eric Clapton	Unplugged	1900
...

顧客

顧客番号	名前	メール
00001	山田一郎	ichiro@foo.co.jp
00002	佐藤優子	yuko@bar.com
...

販売

日付	CD番号	顧客番号
10/10	13579	01234
10/10	33434	00251
10/12	28995	25252
...

在庫

CD番号	倉庫	在庫数
00001	倉庫A	322
00001	倉庫B	815
00003	倉庫B	19
...

自然で
エレガントな
アイデア



SQL言語(1)

- リレーショナルデータベースのための言語
- 1970年代に提案され, いまも拡張されている
- ISO(日本ではJIS)による標準化
- **SQLの利点**
 - 高いレベルでデータベースの操作(問合せ, 更新, 各種定義・制御)が記述可能
 - 標準化されているので, どのRDBMSでも利用することが可能
 - プログラムの移植・連携が容易
 - システムごとに別の言語を学ぶ必要がない



SQL言語(2)

- Q1:ビートルズのCDのCD番号とタイトルは？

```
SELECT CD番号, タイトル
FROM CD
WHERE アーティスト = 'The Beatles'
```

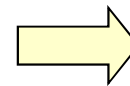
出力したい
列を列挙

対象とする
テーブル名を
列挙する

条件の指定(絞り込み)

CD

CD番号	アーティスト	タイトル	価格
00001	The Beatles	Abbey Road	2000
00002	The Beatles	Rubber Soul	1800
00003	Eric Clapton	Unplugged	1900
...



CD番号	タイトル
00001	Abbey Road
00002	Rubber Soul
...	...

条件を満たす行・列を抜き出す



SQL言語(3)

• Q2:ビートルズのCDの在庫状況

```
SELECT タイトル, 倉庫, 在庫数
FROM   CD, 在庫
WHERE  CD.CD番号=在庫.CD番号
      AND アーティスト='The Beatles'
```

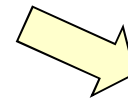
属性名が曖昧な場合,
テーブル名 + '.' を付与

在庫

CD番号	倉庫	在庫数
00001	倉庫A	322
00001	倉庫B	815
00003	倉庫B	19
...

CD

CD番号	アーティスト	タイトル	価格
00001	The Beatles	Abbey Road	2000
00002	The Beatles	Rubber Soul	1800
00003	Eric Clapton	Unplugged	1900
...



複数のテーブルの情報を
組み合わせる

タイトル	倉庫	在庫数
Abbey Road	倉庫A	322
Abbey Road	倉庫B	815
...



SQL言語(4)

- いったん2つの表を統合

CD.CD番号	アーティスト	タイトル	価格	倉庫.CD番号	倉庫	在庫数
00001	The Beatles	Abbey Road	2000	00001	倉庫A	322
00001	The Beatles	Abbey Road	2000	00001	倉庫B	815
00003	Eric Clapton	Unplugged	1900	00003	倉庫B	19
...

- 条件(タイトル='The Beatles')に合致した行を選択

CD.CD番号	アーティスト	タイトル	価格	倉庫.CD番号	倉庫	在庫数
00001	The Beatles	Abbey Road	2000	00001	倉庫A	322
00001	The Beatles	Abbey Road	2000	00001	倉庫B	815
...

- 不要な列を削除

直観的な説明
実際はもっと効率的に処理

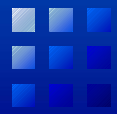


SQL言語(5)

- Q3: 顧客「山田一郎」がこれまで購入したCDのタイトルは?

```
SELECT タイトル
FROM    CD, 顧客, 販売
WHERE   CD.CD番号=販売.CD番号 AND
        顧客.顧客番号=販売.顧客番号 AND
        顧客.名前='山田一郎'
```

– 3つのテーブルを統合する処理となる

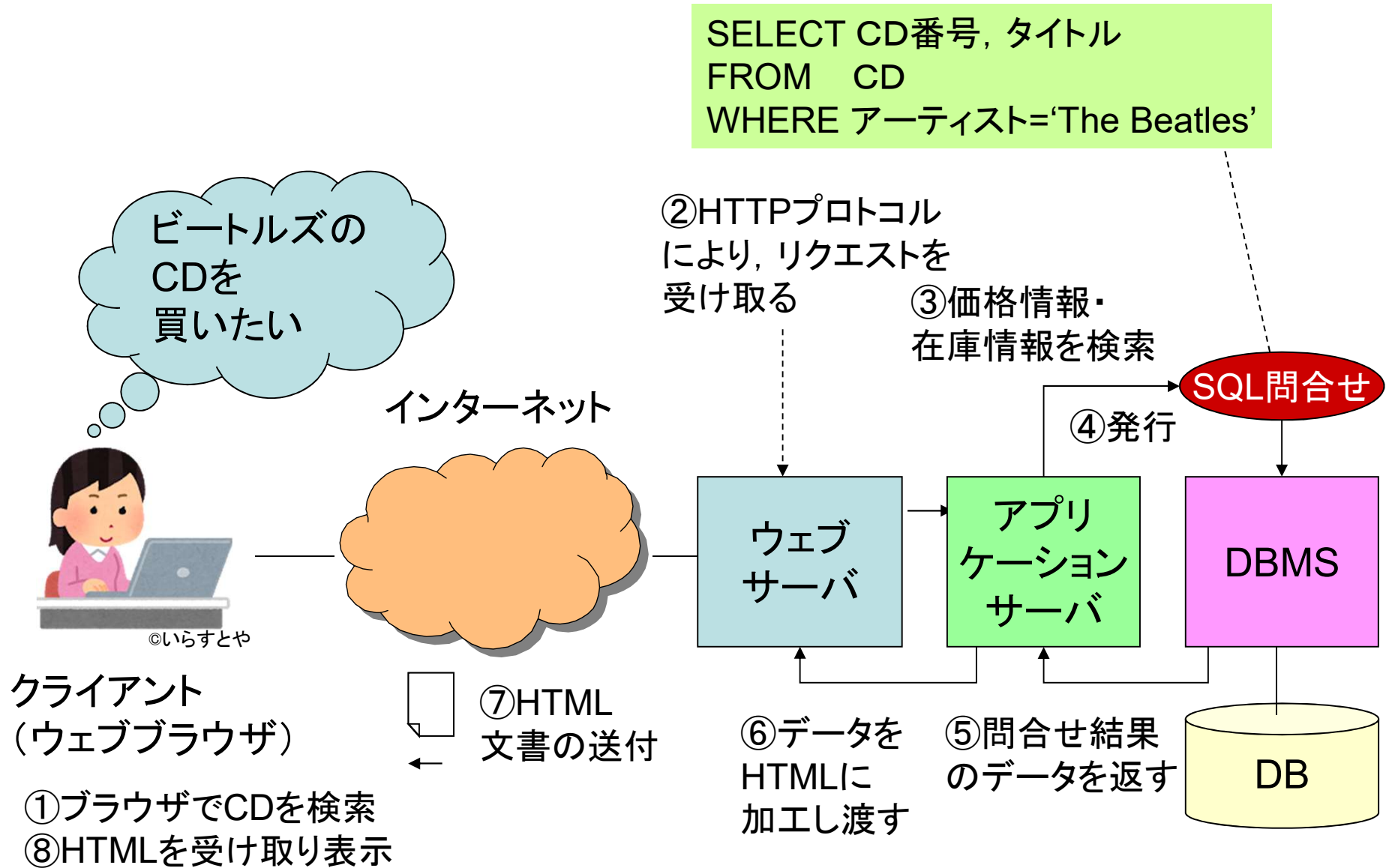


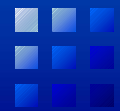
SQL言語(6)

- その他, SQLで記述できる問合せの例(ここで紹介していない機能も利用)
 - ビートルズのCD「Abbey Road」の総在庫数は?
 - ビートルズのCDで最も安いものは?
 - ビートルズのCDの価格の平均は?
 - ビートルズのCDを買ったがエリック・クラプトンのCDを買っていない人は?
 - ビートルズのすべてのCDを買った人は?
 - 2007年10月のCDの販売数と売上金額は?



3階層モデルでのSQLの役割





問合せ結果のHTML文書の例

- ステップ⑦でサーバから送られる

```
<html>
<h1>問合せ結果</h1>

<ul>
<li>00001, Abbey Road</li>
<li>00002, Rubber Soul</li>
</ul>

</html>
```

- ブラウザで表示 ⇒





HTML文書の生成(サーバ側)

- HTML文書のひな型にデータベース問合せ結果を埋め込む: PHP言語の例(やや簡略化)

```
<html>
<h1>問合せ結果</h1>

<ul>
$result = exec("SELECT CD番号, タイトル FROM CD WHERE ...");
if ($result) {
  do {
    @$row = fetch_row($result, $i);
    if ($row) {
      print('<li>'. $row[0] . ', ' . $row[1] . '</li>' . "\n");
    }
  } while ($row);
}
</ul>

</html>
```

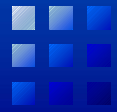
`exec()` は、引数で与えられた SQL文を実行
`$result`変数には問合せ結果が入る

`fetch_row()` は、問合せ結果から 次の1行を読み込む

`print()` で出力する
' ' は文字列を結合することを表す
"¥n" は改行文字

このひな型をPHPインタプリタ(PHP言語の処理プログラム)で実行すると、先のHTML文書が得られる

実際には、青字のSQL文も固定ではなく、その場で生成して埋め込む



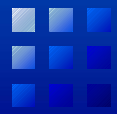
目次

- データベースシステムとは
- リレーショナルデータモデルとSQL言語
- データベースシステムの必要性
- ビッグデータの時代



データベースシステムの必要性(1)

- DBMSを用いないデータ管理にはいくつかの問題が存在
 - 1. データとアプリケーションの相互依存
 - アプリケーションとファイル構造が相互依存: 異なったファイル構造で格納されたデータを利用できない
 - 例: 学生の成績はExcelで管理しているが, 学生の指導教員データはテキストファイル, 教員の教務データは専用プログラムで独自のファイル形式など
 - 統合利用(例: 留年可能性大の学生について, 関連する授業を担当している教員を知りたい)が難しい
 - 不統一性, 不整合性, 冗長性の原因



データベースシステムの必要性(2)

2. 整合性維持機能の欠如

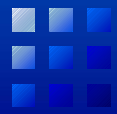
- 不正な更新, 誤った入力の排除が難しい(例: 複数ファイルで学生の学籍番号が不一致)
- アプリケーションプログラムでチェックが必要

3. 不十分な機密保護

- 複数ユーザによるデータ共有ではアクセス制御(読出し, 書込みの制限)が必要
- ファイル単位のアクセス制御は可能だが, データ単位のアクセス制御は困難

4. 処理速度

- 効率的な問合せ処理は容易でない



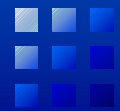
データベースシステムの必要性(3)

5. 複数ユーザの同時アクセス

- 複数ユーザによる更新を同時に行うとさまざまな異状, 矛盾が発生
- 有用性を維持しつつ, 複数ユーザへの対処を図ることが重要

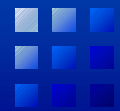
6. 不十分な障害時データ保護

- システム障害への対応は容易ではない
 - プログラムのエラー
 - システムダウン
 - ディスククラッシュ
- 障害時にもデータを保護し, すみやかな復旧を図る機能が求められる



問合せ処理(1)

- amazon.co.jpの場合,「本」のカテゴリだけで500万件以上
- キーワード検索で,なぜ一瞬(< 0.5秒)で結果が返るのか?
 - ネットワーク上のデータ転送速度も考えると, 0.1秒程度での処理
- 500万件のデータを頭から探しては到底不可能
 - 例: 1TBのデータがHDDにあるとき, 250MB/sで全データを読みだすのに1時間以上かかる



問合せ処理(2)

- Q1:ビートルズのCDのCD番号とタイトルは？

```
SELECT CD番号, タイトル
FROM CD
WHERE アーティスト = 'The Beatles'
```

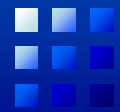
CD

CD番号	アーティスト	タイトル	価格
00001	The Beatles	Abbey Road	2000
00002	The Beatles	Rubber Soul	1800
00003	Eric Clapton	Unplugged	1900
...

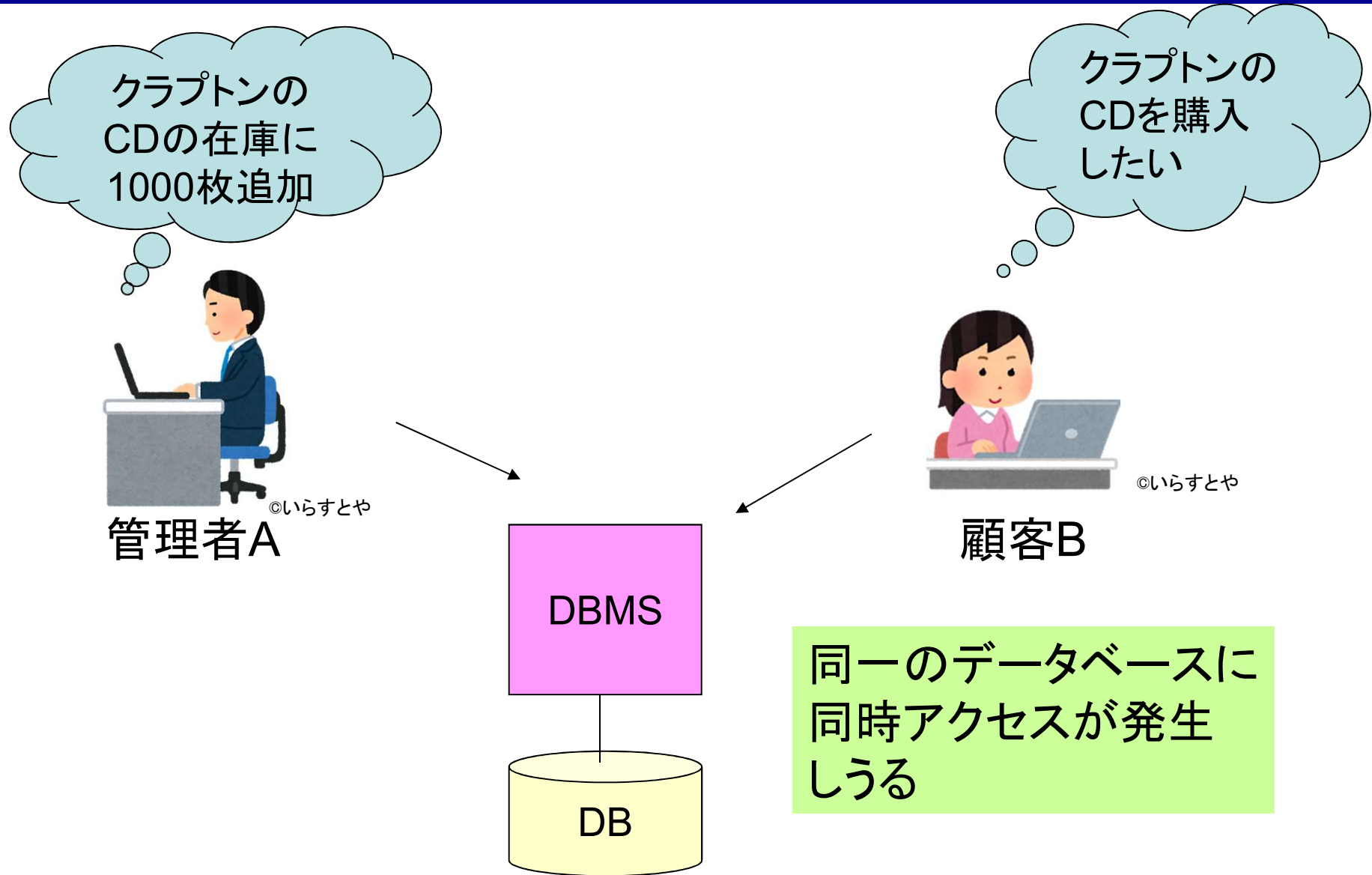
索引(インデックス)

アーティスト	CD番号
The Beatles	0001, 0002
Eric Clapton	0003
...	...

- 索引を事前に作っておく.
- 問合せ時にはまず索引を見て, この場合であれば0001, 0002というCD番号を得る. CD番号をもとに直接対象のデータを取得する
- 効率的な索引を作るための技術が存在



同時アクセスへの対応(1)

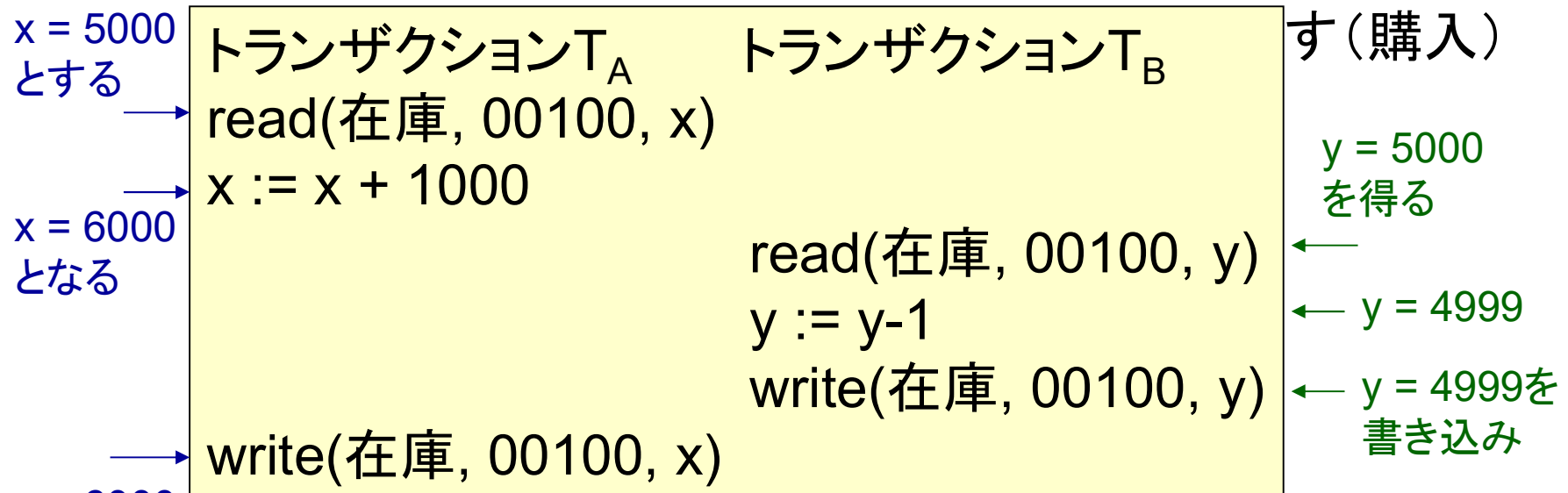




同時アクセスへの対応(2)

- 「在庫」テーブルに対する同時アクセス
 - トランザクション: データベースに対する一連の処理

- T_A : 商品番号00100の商品の在庫数を1000増やす



$x = 6000$
を書き込む
上書きしている!

在庫数は5999になるべきなのに、
誤って6000となってしまう!



同時アクセスへの対応(3)

- **同時実行制御**

- 競合するデータへのアクセスを制御
- 更新による不整合の発生を防ぐ
- 一方で, できるだけ並行性を向上させる
 - 競合しない要求は同時に実行する
 - 競合しない例
 - 管理者A: CDの在庫を増やす
 - 顧客B: CDの価格をチェック

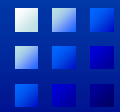
- **整合性 (consistency) の維持管理**

- 同時実行制御と関連: ますます重要に
- 世界中に存在するデータセンター間で, データの同期をどのようにとるか



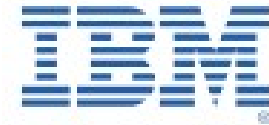
目次

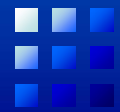
- データベースシステムとは
- リレーショナルデータモデルとSQL言語
- データベースシステムの必要性
- **ビッグデータの時代**



情報分野を代表する企業は？

- 1970～80年代 : IBM
 - 大型計算機からPCまで
 - コンピュータ(ハードウェア)を作り提供
- 1980～90年代 : Microsoft
 - Windowsオペレーティングシステム
 - ソフトウェアを作りサービスを提供
- 2000年代 : Google
 - サーチエンジン
 - ウェブから情報(データ)を集めて提供
- 2010年代 : ?
 - GAFA (Google, Apple, Facebook, Amazon)
 - 情報サービス(ビッグデータ含む)を世界規模で

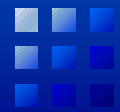




ビッグデータ出現の理由(1)

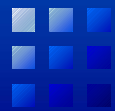
- 情報(データ)を蓄積するコストが**どんどん小さくなっている**
 - 1TBの外付けHDDが5,000円以下で買える
 - 新聞(文字のみ)1部は500キロバイト程度:1TBのディスクに約5,500年分
 - 圧縮方式などにもよるが, 1TBにはおよそ25万曲:1曲あたり単価0.02円
- 無駄な情報を蓄積するコストよりも, 整理をするコストの方が大
- **無料のサービス**(例: Gmail, Facebook)などの出現の理由



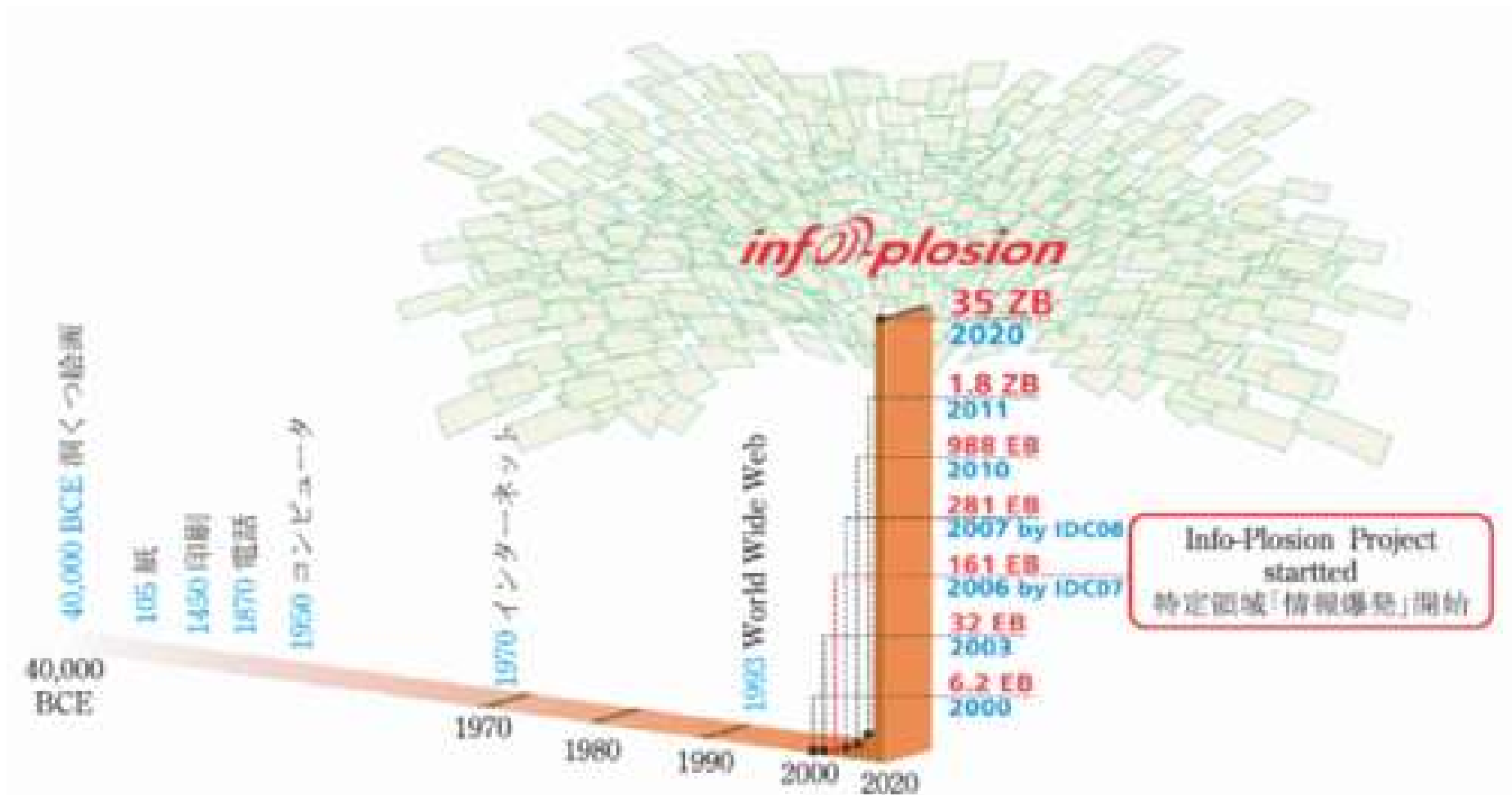


ビッグデータ出現の理由(2)

- インターネット時代になり, さまざまな情報サービスが出現
- Web
 - 1兆以上のページが存在し, 増え続ける
 - **Hidden Web**: Webサイトの背後にあるデータであり, 検索などを通じて見ることができる
- センサデータ
 - GPS, 監視画像, ICタグ, コンビニ・スーパーのPOS (Point of Sales) データなど
- IoT (Internet of Things)
 - モノのインターネット



情報爆発



情報爆発 (出典: Horizon Information Strategies, cited from Storage New Game New Rules, p. 34 (www.horizon.com), IDC, The Diverse and Exploding Digital Universe 2020 (http://www.emc.com/collateral/demos/microsites/idc-digital-universe/iview.htm))

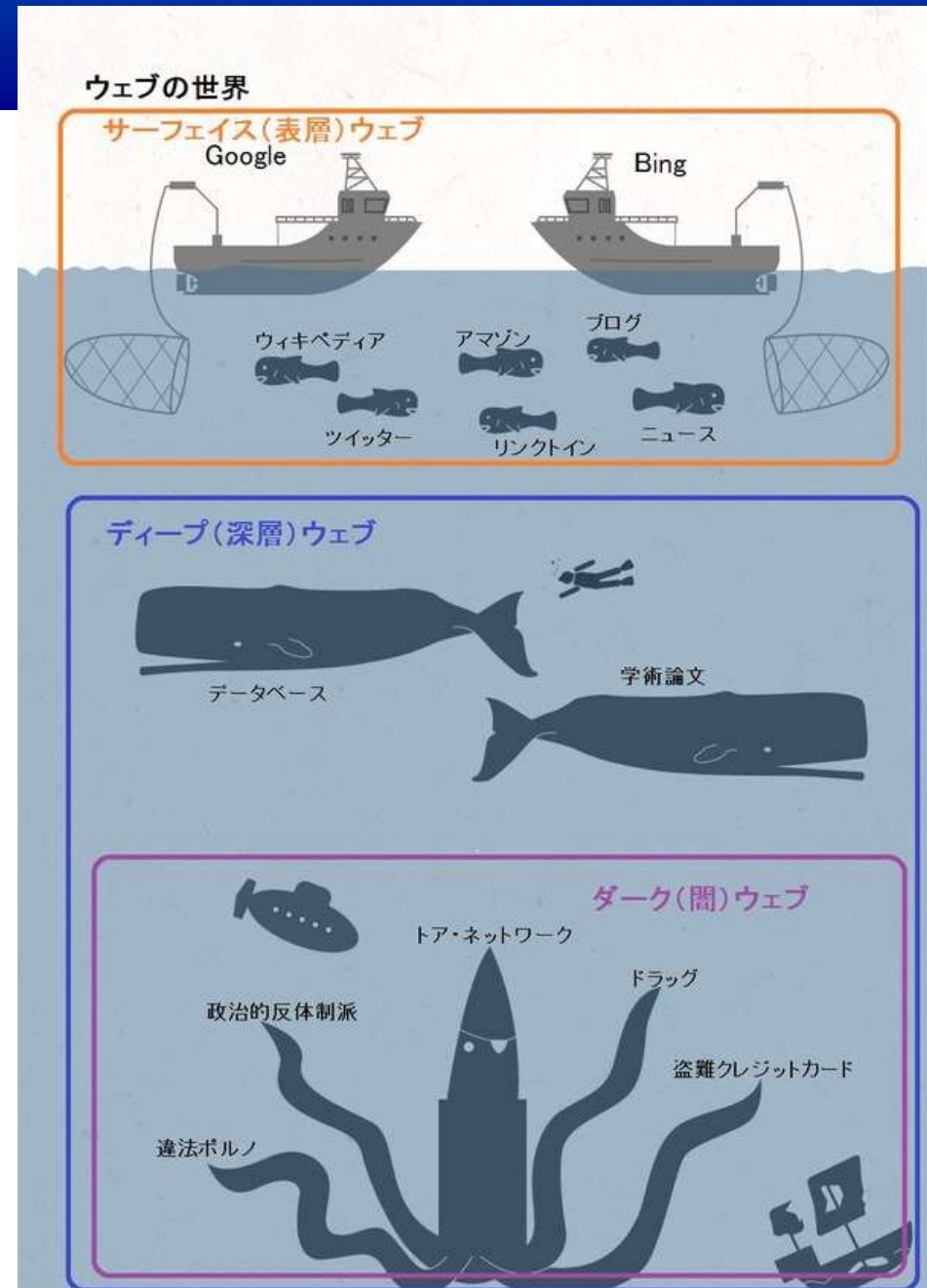
出典: http://japan.zdnet.com/storage/2013/07/08/93f8ed4c3d1f607fe261ed8e747490cf/130708_saitou1.png

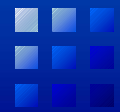


Hidden Web

- Invisible Webないし Deep Webともいう
- サーチエンジンや検索インタフェースを介してアクセス可能な場合も
- 我々が目にしている Webはほんの一部

<https://hajimeru-bitcoin.com/deepweb>





IoT (Internet of Things)

- 「モノのインターネット」と呼ばれる
- すべてのモノが
ネットを介して
つながる

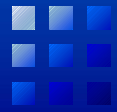


出典：http://www.3g.co.uk/g_phones/large/internet-of-things-everything-you-need-to-know.jpg



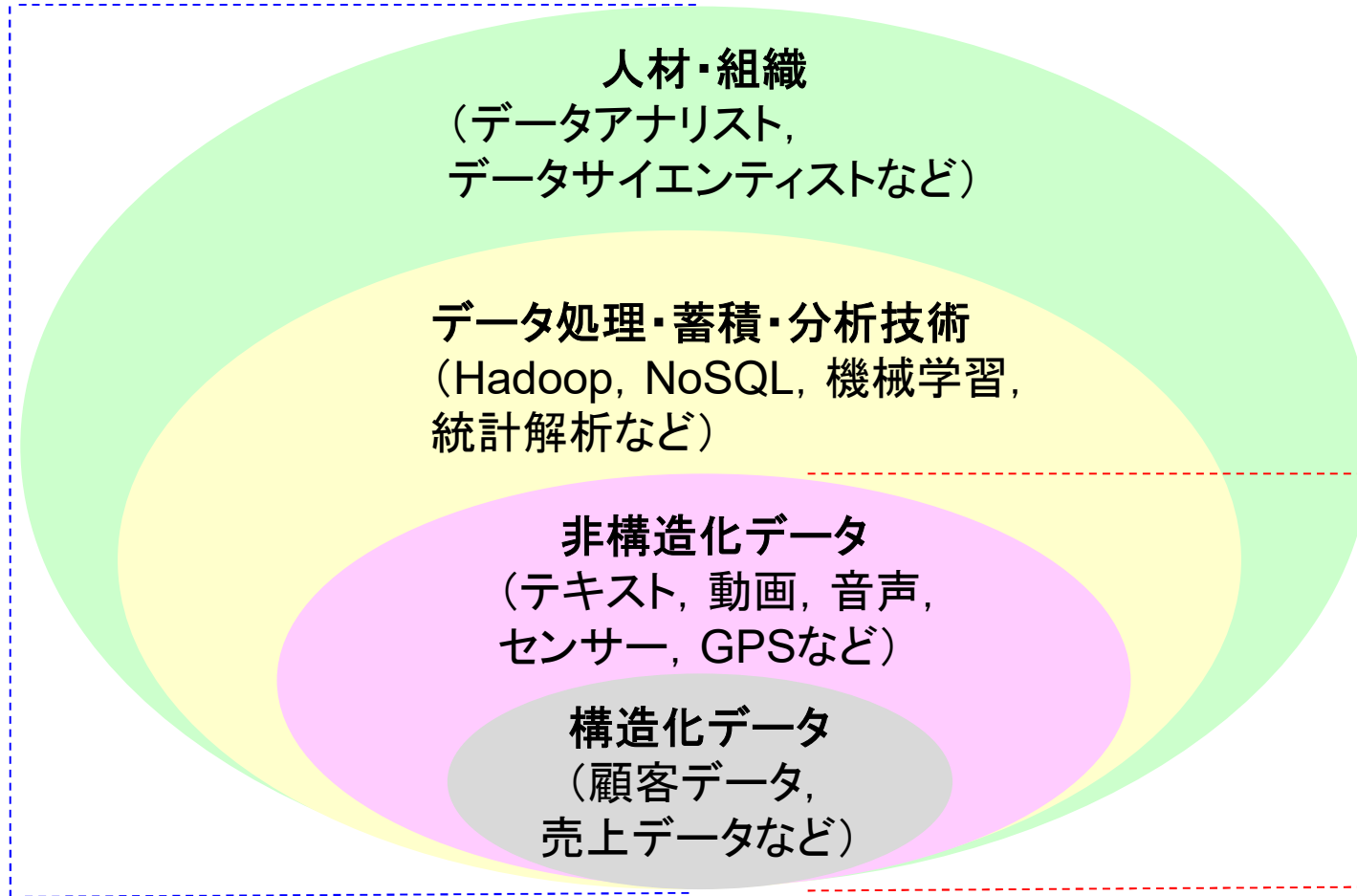
ビッグデータとは(1)

- 一般的な定義: 既存の技術では管理することが困難な大量のデータ
- 管理が困難な要因: **3V**
 - ボリューム (Volume): データの量
 - 多様性 (Variety): さまざまなデータが扱われる
 - 発生頻度・更新頻度 (Velocity): オンライン取引データ, 交通データなど
 - データの正確さ (Veracity) を加え **4V** とも言う
- より広い定義: 「ビッグデータ」を蓄積・処理・分析するための技術や, 「ビッグデータ」を分析する人材・組織を含む包括的な概念

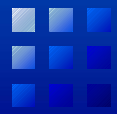


ビッグデータとは(2)

広義のビッグデータ



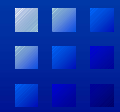
狭義のビッグデータ



ビッグデータと産業革命

- **産業革命**とは: 生産・流通の手段に大きな変革があり, 産業の構造やコストに大幅な変化があること
- 第一次産業革命: 18世紀末
 - 蒸気機関の出現: 手工業の衰退
- 第二次産業革命: 19世紀末
 - 石油と電気: 自動車・プラスチックによる大量生産・大量輸送
- 第三次産業革命: 20世紀後半
 - エレクトロニクス・コンピュータ
- **第四次産業革命** (いま!)
 - 情報 (**ビッグデータ**) の蓄積・流通の**コストが激減**
 - 情報技術 (例: 人工知能など) の進展

情報技術は
ますます重要に



ビッグデータと人工知能

- 人工知能の急速な発展の理由
 - ディープラーニング・機械学習などの技術の進展
 - コンピュータの性能向上
 - **ビッグデータの出現**: 高度な人工知能は、大量のデータの学習を前提としている

- Big data fuels AI

- 「**データは21世紀の石油**」
 - 今日、最も価値のある資源
 - EU一般**データ保護規則**
(**GDPR**)などの動き

https://infocus.dellemc.com/william_schmarzo/data-is-new-currency/





データサイエンティスト

- ビッグデータの分析
 - ビッグデータを分析して、価値ある情報を手に入れる
 - 例：自動車の移動パターン、ウェブアクセスの傾向、商品の売り上げの分析
 - データマイニング（データの発掘）とも呼ばれる
- データサイエンティスト（データアナリスト）
 - 大規模なデータが扱える
 - プログラミングなどの豊富な知識を有する
 - 対象分野の知識がある
 - 統計・機械学習などのツールが使える
 - 今後ますます重要に



データエンジニア

- 人工知能技術の活用には, ビッグデータの処理が欠かせない
 - データクリーニング (data cleaning)
 - データの変換や修正を行い, 学習に活用可能とする
 - データクレンジング (data cleansing) ともいう
 - データラングリング (data wrangling)
 - データを「飼いならす」という意味
 - データの獲得プロセスを含めたより広い意味
- **データエンジニア**
 - データベース技術などのシステム技術と人工知能技術を駆使して, ビッグデータに立ち向かうエンジニア

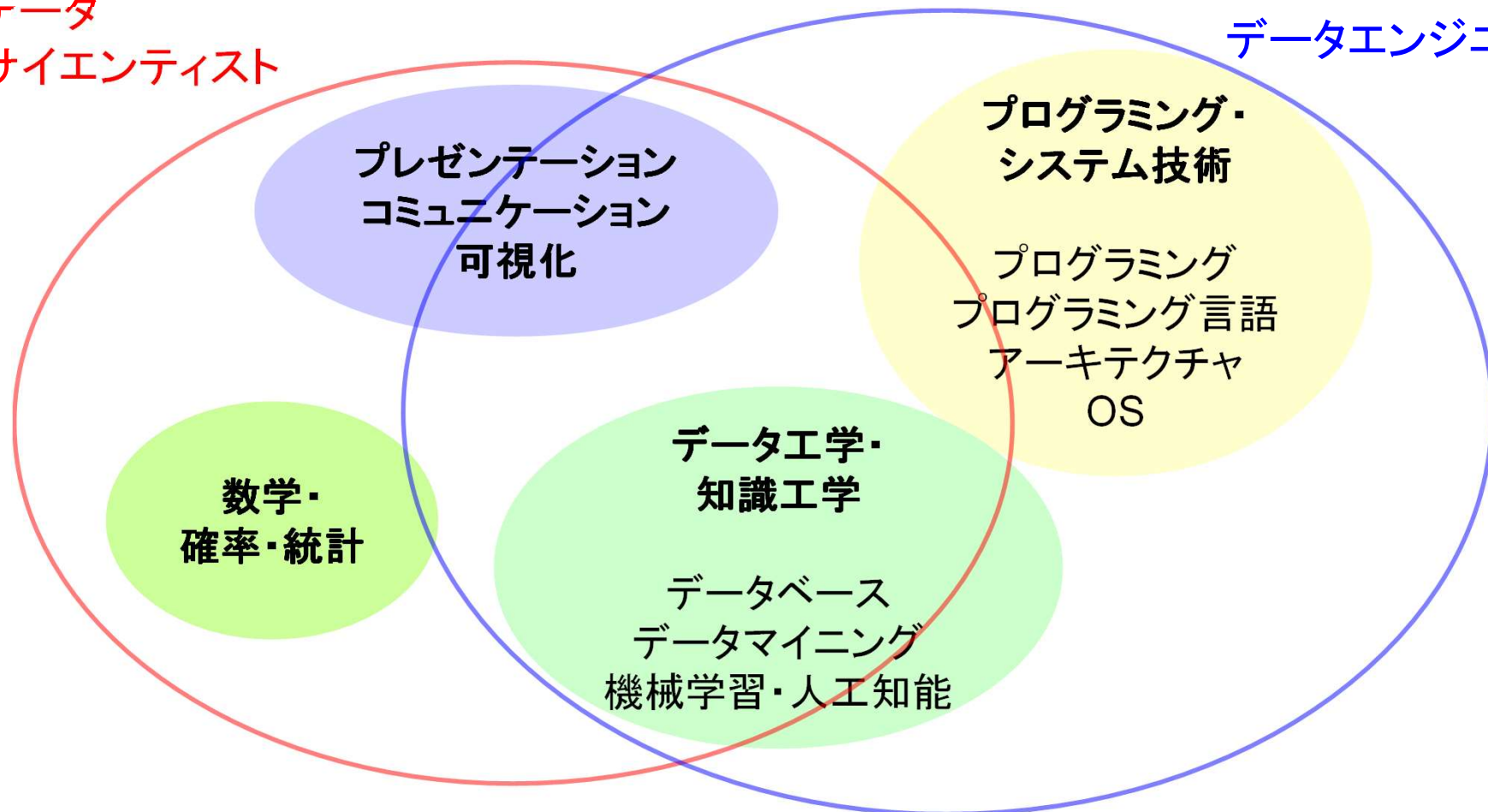


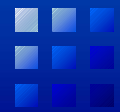
データサイエンティスト／エンジニアを目指す

- 総合的なスキルが必要
 - 情報学部(＋情報学研究科)でその多くをカバー

データ
サイエンティスト

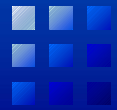
データエンジニア





レポート課題(1)

- 以下の問題に答えること. **A4版**のレポート用紙に回答し, **次回の講義(村瀬勉先生)**の開始時に提出すること.
- レポートには, **学生番号, 氏名, 担当教員名(石川)**を明記すること
- **課題1**
 - 次に示すデータベースの例を考える
 - 「科目」は科目情報, 「学生」は学生の情報, 「履修」はどの科目をどの学生が受講し, 成績が何点であったかを示す



レポート課題(2)

科目

科目番号	科目名	単位数
001	哲学概論	2
002	解析学	3
003	代数学	3

学生

学生番号	氏名	学部
00001	山田太郎	工学部
00002	鈴木明	工学部
00003	渡辺花子	理学部

履修

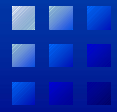
科目番号	学生番号	成績
001	00001	90
001	00002	80
002	00001	90
002	00003	60
003	00001	80
003	00002	95
003	00003	70



レポート課題(3)

- 以下の問合せをSQLで表現し, このデータベース例に適用した結果とともに示せ
 - ①単位数が3単位の科目の科目番号と科目名は?
 - ②「哲学概論」を受講している学生の学生番号と成績は?
 - ③「哲学概論」を受講している学生の名前は?
- 注意
 - 問合せ結果は表形式で書く必要はない
 - たとえば①については右のように書いてよい(右は正解とは限らない)

科目番号	科目名
001	哲学概論
002	解析学

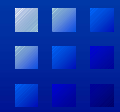


レポート課題(4)

• 課題2

- 以下のテーブルは, 先に示した三つのテーブルを一つにまとめたものである.
- 一般には, 先のようにテーブルを分割する方が推奨される. その理由(複数可)について考察せよ.

科目番号	科目名	学生番号	氏名	学部	成績
001	哲学概論	00001	山田太郎	情報学部	90
001	哲学概論	00002	鈴木明	工学部	80
002	解析学	00001	山田太郎	情報学部	90
002	解析学	00003	渡辺花子	理学部	60
003	代数学	00001	山田太郎	情報学部	80
003	代数学	00002	鈴木明	工学部	95
003	代数学	00003	渡辺花子	理学部	70



レポート課題(5)

- 課題3

- 本日の講義に対する感想・コメントを自由に書いてください。