

情報システムとデータベース

「情報II」第4章

Contents

1. 情報システムの開発	01
2. 情報システムの構成とDBMS	07
3. 関係モデル	13
4. 関係演算	21
5. データの操作	31

●本書の複製等について—本書のコピー、スキャン、デジタル化等の無断複製は著作権法上での例外を除き禁じられています。本書を代行業者等の第三者に依頼してスキャンやデジタル化することは、たとえ個人や家庭内の利用でも認められておりません。

クラス： 番号： 氏名：

情報システムの開発

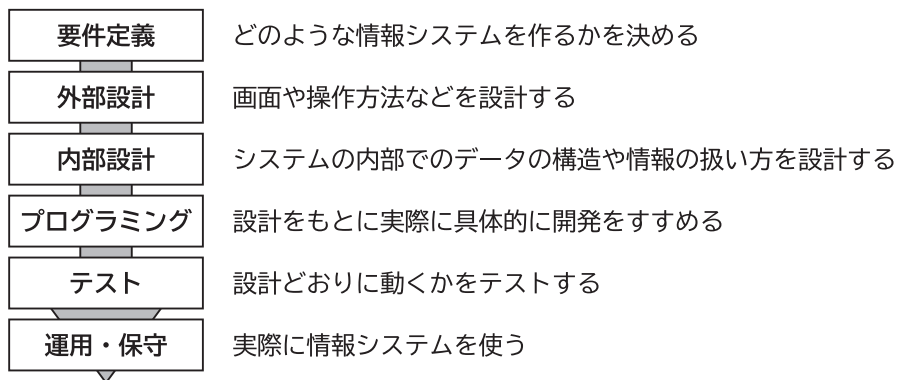
業務や活動を円滑に行えるように、情報を処理したり伝達したりする仕組みのことを情報システムといいます。ここでは、情報システムを開発するための流れや、システム設計をする際の表現方法などについて学びます。

(教科書II : p.100 – p.101 , p.110 – p.113)

■ 情報システムの開発

情報システムの開発

情報システムは、一般に次のような工程で開発される



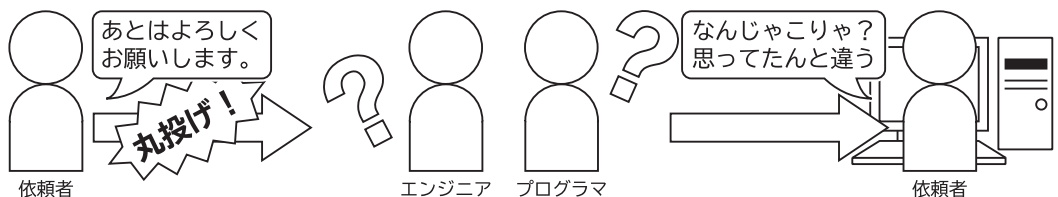
※各工程を確実に仕上げたから次の工程に進む方法 = [¹]
 →進捗の管理がしやすい ⇔ 前工程が終わらないと次の工程に進めない

情報システムに携わる主な職業

システムエンジニア	主にシステム開発の [²] に関わる技術者
プログラマー	[³] を行う技術者

※システムエンジニアは、テストや運用のサポートなども担う

エンジニアもプログラマーも、依頼者の業務内容まで把握しているわけではない



※依頼者と開発者が相互に理解を深めることで、現場で使える情報システムが開発できる
 →依頼する側としても、情報システムの開発に携わる基礎知識を持つようにしましょう

システム開発の手順

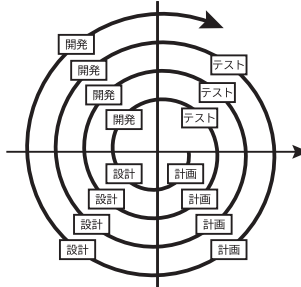
前頁に示した〔¹

〕以外にも次のような方法もある

スパイラルモデル

開発するシステムをいくつかのサブシステムに分割

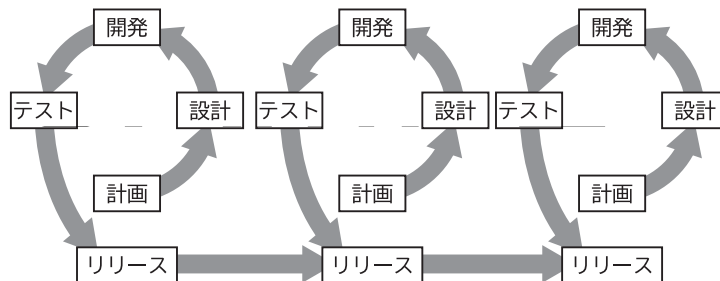
→分割したサブシステムを、順番に開発し、サイクルを繰り返すことで規模を拡大



アジャイル開発モデル

途中で仕様や設計の変更があることを前提にした開発手法 → ^{アジャイル}素早く開発

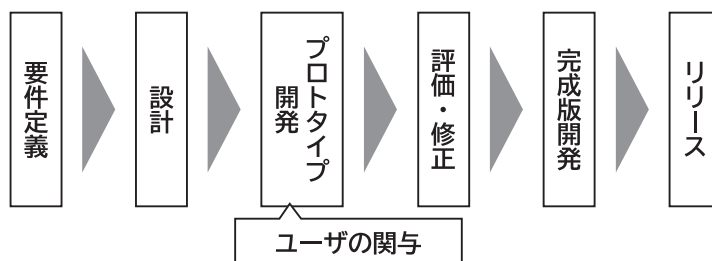
→小さな単位で開発を行い、都度リリースしながら、徐々に拡大していく手法



プロトタイプングモデル

開発の早い段階で^{プロトタイプ}試作品を作成し、ユーザからの意見を受けて改良しながら開発する手法

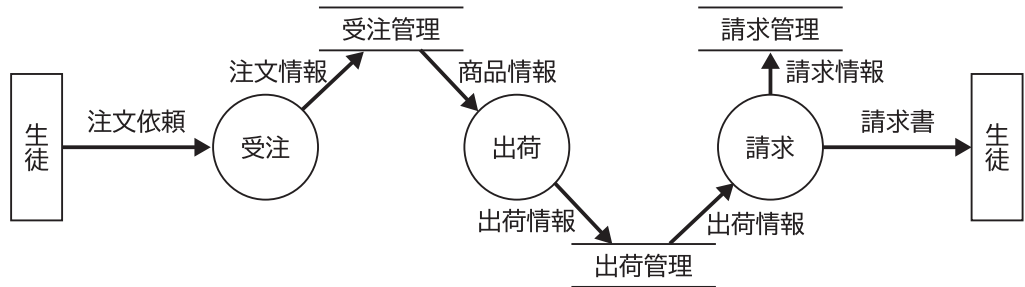
→完成品をよりユーザのニーズを反映したものにしやすい



■ システム設計の表現方法

データフローダイアグラム (DFD)

ユーザとシステムとのやりとりをデータの流りに着目して図に表したもの



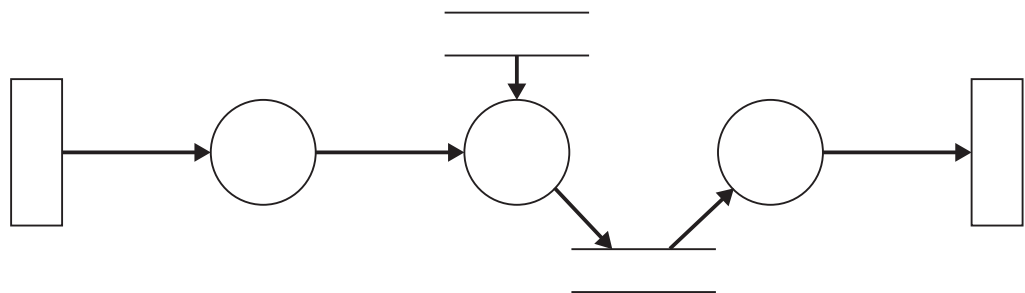
DFDの構成要素

プロセス	源泉	データストア	データの流れ
○	□	=	→
処理・機能	データの発生源 データの行き先	ファイル データベース	データの流れ

例題1

生徒が試験を受験し、成績通知されるまでの一連の処理をDFDで表現してみよう
下の図に、語群から適当な語を選び、図に書き入れよう（同じ語が複数入る場合もある）

4

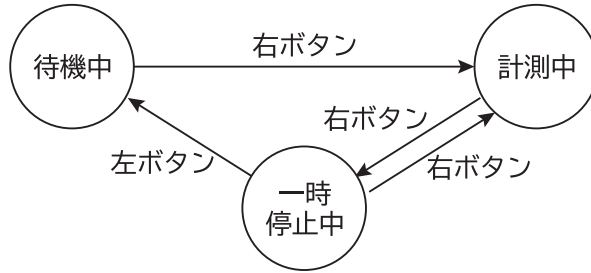


[語群]

生徒 / 学力 / 評定 / 正答 / 模範解答 / 答案用紙 / 受験 / 点数 / 成績通知 / 採点 / 成績管理

状態遷移図

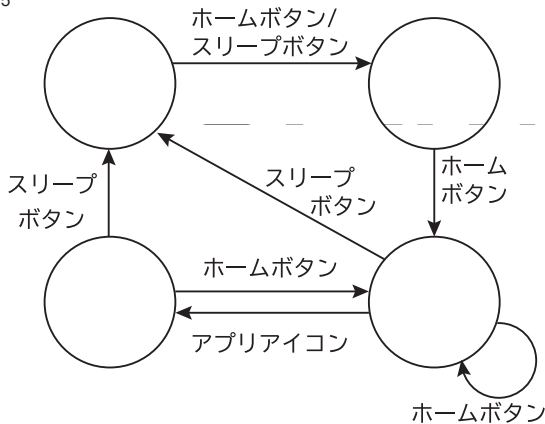
時間変化や状態変化による状態の遷移を矢印で表したもの



例題2

下図は、スマートフォンの**スリープボタン**、**ホームボタン**の動作に関する状態遷移図
 どのような状態からどのように遷移するかは、表にまとめてある
 表を参照しながら、状態遷移図を完成させよう
 ただし、この図は未完成であり、状態遷移（矢印）を1つ追加する必要がある
 追加する状態遷移も図に書き入れよう

5

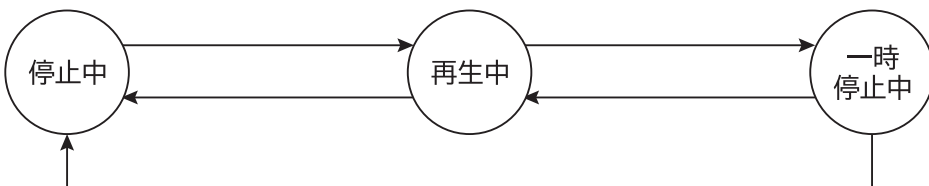


	ホームボタン	スリープボタン	アプリアイコン
スリープ状態	ロック画面	ロック画面	—
ロック画面	ホーム画面	スリープ状態	—
ホーム画面	ホーム画面	スリープ状態	アプリ画面
アプリ画面	ホーム画面	スリープ状態	—

問題1

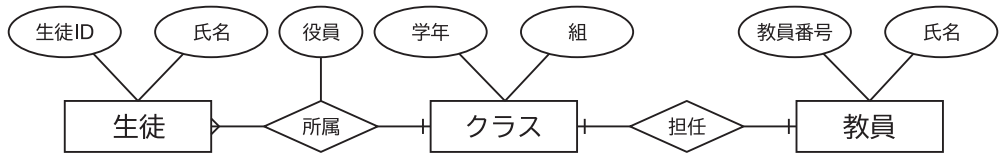
DVDデッキには、**再生ボタン**、**一時停止ボタン**、**停止ボタン**の3つのボタンがある
 各ボタンによる動作の状態遷移図を描いてみよう

6



ER図

エンティティ
実体と実体との間の**リレーションシップ**
関連に着目して図に表したもの



ER図の構成要素

実体	属性	関連	対応
□	○	◇	—
—+	1個の対応関係		
—<	1個以上の対応関係		

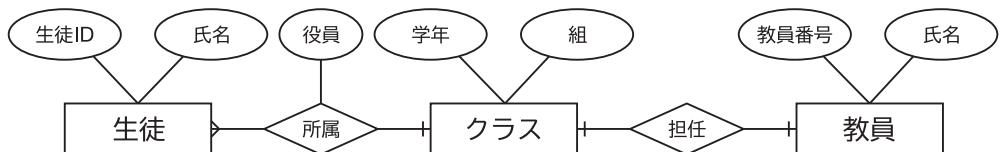
例題3

上のER図に更に科目の履修関係を追加してみよう

生徒は、科目を複数履修することができ、履修科目ごとに成績と出席状況が記録される
科目には、科目番号と科目名、単位数が設定されている

また、1科目につき1名の教員が担当するものとする

7



問題2

部活動の管理をするシステムのER図を作ってみよう

生徒は、複数の部活に所属することができ、所属に対して役職を記録しておくものとする

生徒は、生徒番号、クラス、氏名で管理されている

部活は、名称、種別で管理されている

1つの部活は、大会名、開催日、開催場所で管理された大会に複数出場することがある

大会は、1回の出場につき、1つの実績が記録される



振り返り

次の各観点が達成されていれば□を塗りつぶしましょう。

情報システムを開発する手順について理解することができた

さまざまなシステム設計の表現方法について、それぞれの考え方を理解できた

システム設計の表現方法で図を描くことで、システム設計の考え方が身に付いた

今日の授業を受けて思ったこと、感じたこと、新たに学んだことなどを書いてください。

.....

.....

.....

情報システムの構成と DBMS

わたしたちは日々、情報システムを利用する中で、非常に多くの情報を生み出しています。このようにして生み出された情報はデータベースに蓄積され、管理、運用されます。データベースを利用するには、データベース管理システム（DBMS）を用います。

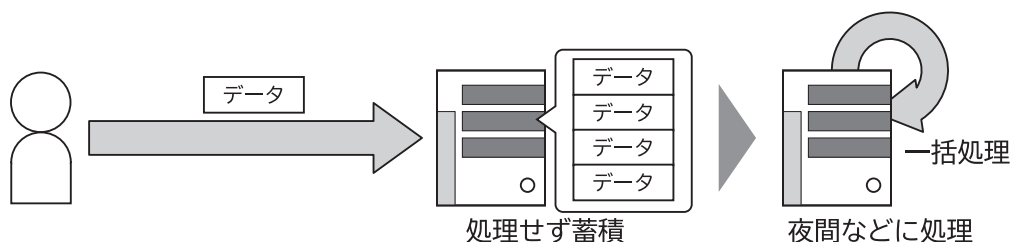
(教科書I : p.186 – p.189 , 教科書II : p.102 – p.103)

■ 情報システムの処理形態

バッチ処理とオンライン処理

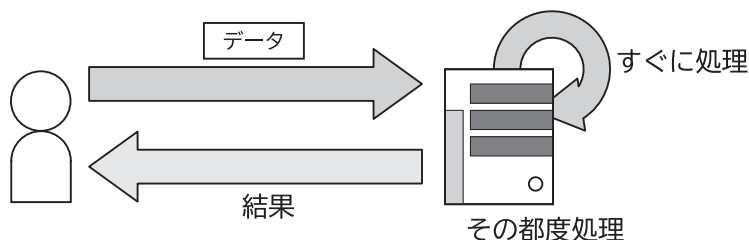
バッチ処理

データを処理せず貯めておき、夜間や月末などのタイミングで一括して処理する
→売上管理や在庫管理などに利用されている



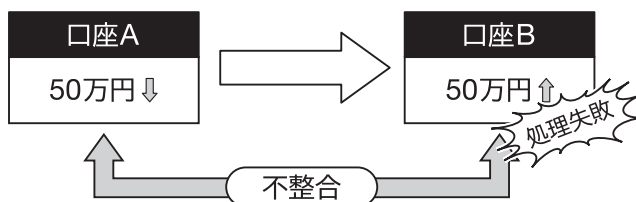
オンライン処理

データが発生する度に、即座に処理し、結果を返す
→銀行の〔¹ 〕や交通機関の〔² 〕などに利用



互いに関連性のある一連の処理を〔³ 〕という

例) 銀行の口座Aから口座Bに50万円を送付する場合、次のような現象に注意

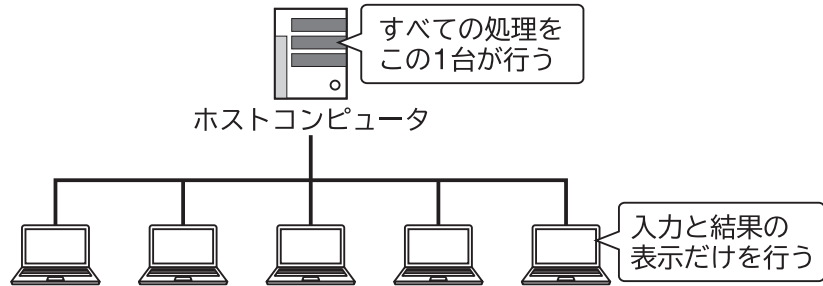


※このような不整合が起こらないようにすべての処理が成功した場合のみ確定する

集中処理と分散処理

集中処理

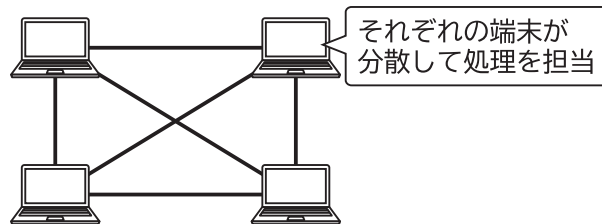
1台の〔⁴ 集中型システム 〕に複数台の端末を接続
 →要求された〔⁵ 処理 〕の処理をホストコンピュータだけで処理を行う



- ※セキュリティの確保や運用管理・保守が〔⁶ 容易 ・ 困難 〕
- ※ホストコンピュータの負荷が〔⁷ 大きい ・ 小さい 〕
- ※ホストコンピュータの故障により、システム〔⁸ 停止 ・ 稼働 〕してしまう

分散処理

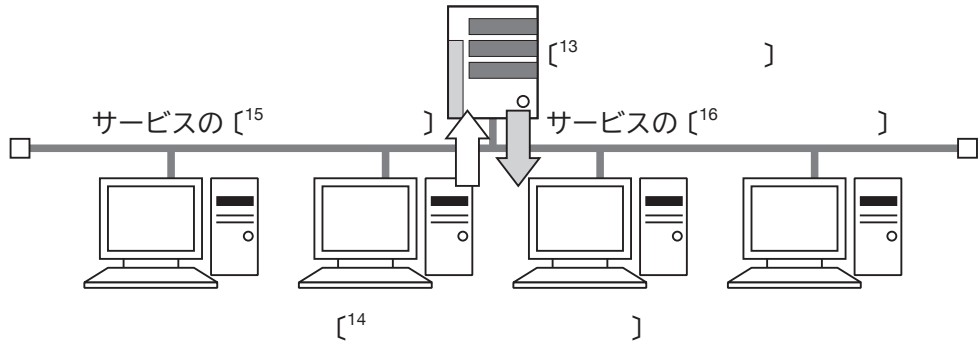
多数のコンピュータをネットワークで接続し、個別の端末に処理を〔⁹ 分散 〕させる



- ※システムの拡張が〔¹⁰ 容易 ・ 困難 〕
- ※1台の端末が故障しても、システム全体への影響は〔¹¹ 大きい ・ 小さい 〕
- ※セキュリティの確保や運用管理・保守が〔¹² 容易 ・ 困難 〕

クライアントサーバシステム

集中処理と分散処理の両方の特徴を併せ持ったシステム



※基本的には各端末が処理を担当する意味で [17] 的

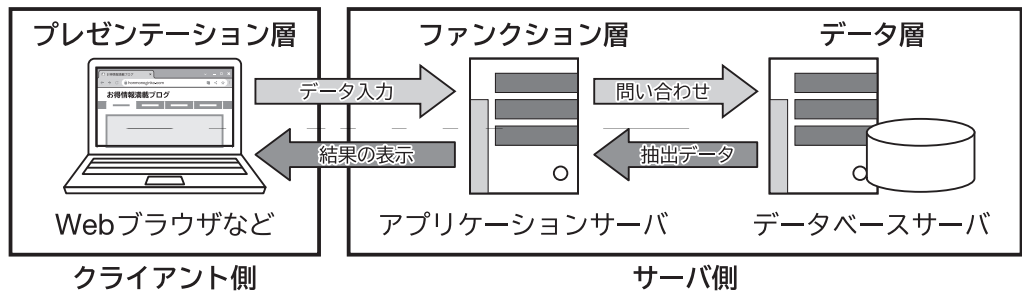
※集中して管理をしたい部分をサーバに担わせる点が [18] 的

3層クライアントサーバシステム

従来のクライアントサーバシステムは、クライアント側でアプリケーションを動作

→クライアント側に処理や管理の負担がかかっていた

→クライアントは [19] に特化、処理機能を [20] に



21	ユーザが直接入出力に触れる層
22	機能を実現するための処理を行う層
23	データベースを用いてデータの保管、管理を行う層

※各層が独立しているため、一部の変更や機能追加が他の層に影響を与えにくい

→システムの保守や拡張が [24] 容易 ・ 困難]

→セキュリティ対策も各層ごとに行えばよいため、[25] 容易 ・ 困難]

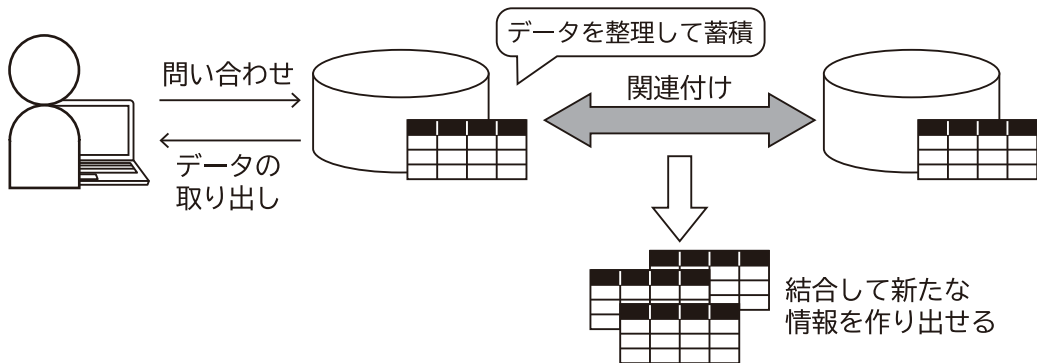
現代の情報システムのほとんどがこの考え方に基づいて開発されている

■ 情報システムとデータベース

データベースとは

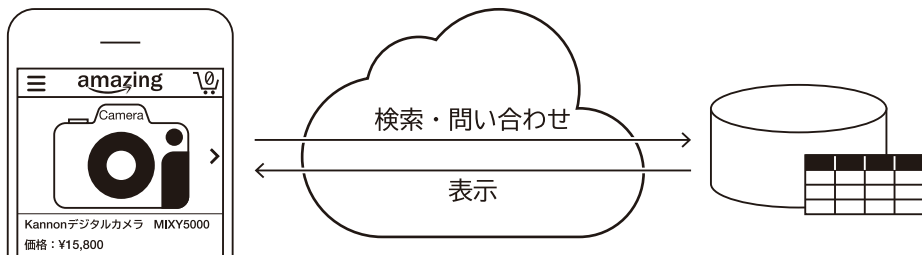
データベース = 大量のデータを扱いやすいように整理し蓄積したもの

※必要な情報を検索したり、データの追加・削除・更新などができる

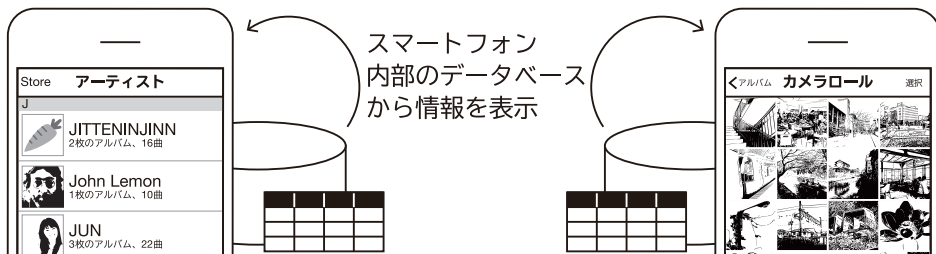


身近なデータベースシステム

●Webサービス

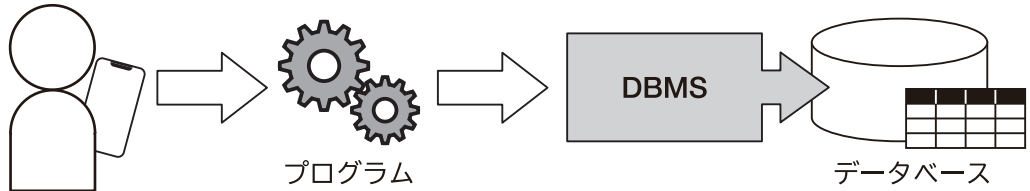


●アプリケーション



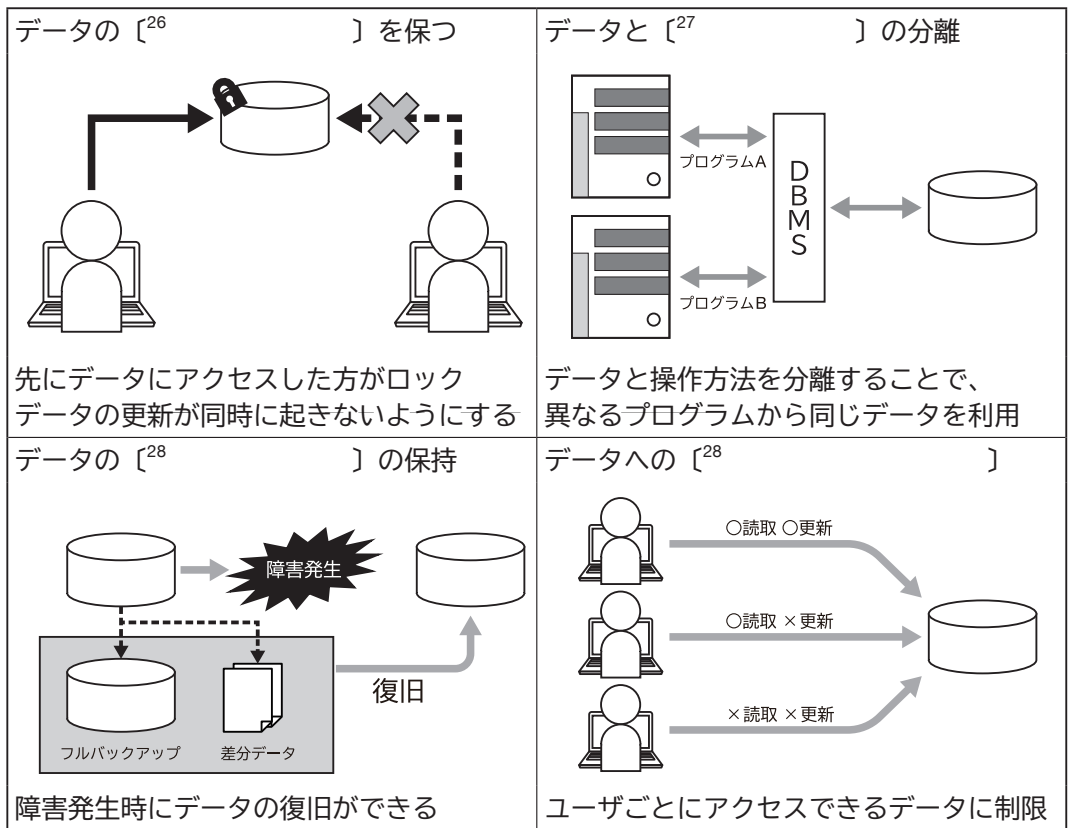
データベース管理システム (DBMS)

データベース管理システム = データベースの作成、運用、管理を行うシステム



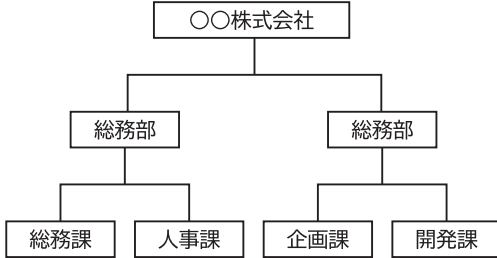
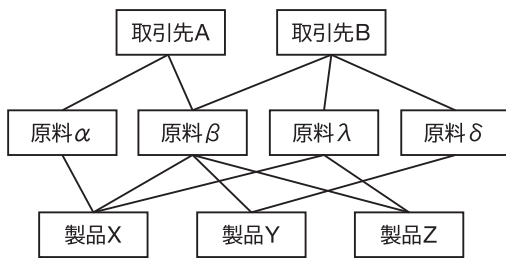
※データベースへは、DBMSを介してアクセスする

DBMSのおもな機能



データモデル

データベースは、データ相互の関係性を図や表で表した**データモデル**に基づいて作成

<p>階層型データベース</p>  <p>木のような階層構造で表現</p>	<p>ネットワーク型データベース</p>  <p>格子状のネットワーク構造をしている</p>																																									
<p>関係データベース</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>客名</th> <th>年齢</th> <th>品名</th> <th>メーカー</th> </tr> </thead> <tbody> <tr> <td>手塚くん</td> <td>29</td> <td>すくな〜いお茶</td> <td>佐藤園</td> </tr> <tr> <td>赤塚くん</td> <td>20</td> <td>たのしい牛乳</td> <td>ネイジ</td> </tr> <tr> <td>藤本くん</td> <td>21</td> <td>鳥龍茶</td> <td>ヨントリー</td> </tr> <tr> <td>我孫子くん</td> <td>22</td> <td>ハクシコーラ</td> <td>ハクシコーラ</td> </tr> </tbody> </table> <p>データ同士の関係を表の形式で整理</p>	客名	年齢	品名	メーカー	手塚くん	29	すくな〜いお茶	佐藤園	赤塚くん	20	たのしい牛乳	ネイジ	藤本くん	21	鳥龍茶	ヨントリー	我孫子くん	22	ハクシコーラ	ハクシコーラ	<p>オブジェクト指向データベース</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>生徒</th> <th>A太</th> <th>B美</th> </tr> </thead> <tbody> <tr> <td>クラス</td> <td>2年3組</td> <td>2年5組</td> </tr> <tr> <td>性別</td> <td>男</td> <td>女</td> </tr> <tr> <td>部活</td> <td>マンガ部</td> <td>テニス部</td> </tr> <tr> <td>授業を受ける○</td> <td>授業を受ける○</td> <td>授業を受ける○</td> </tr> <tr> <td>弁当を食べる○</td> <td>弁当を食べる○</td> <td>弁当を食べる○</td> </tr> <tr> <td>登校する○</td> <td>登校する○</td> <td>登校する○</td> </tr> </tbody> </table> <p>データと処理方法を一つのオブジェクトとして捉える</p>	生徒	A太	B美	クラス	2年3組	2年5組	性別	男	女	部活	マンガ部	テニス部	授業を受ける○	授業を受ける○	授業を受ける○	弁当を食べる○	弁当を食べる○	弁当を食べる○	登校する○	登校する○	登校する○
客名	年齢	品名	メーカー																																							
手塚くん	29	すくな〜いお茶	佐藤園																																							
赤塚くん	20	たのしい牛乳	ネイジ																																							
藤本くん	21	鳥龍茶	ヨントリー																																							
我孫子くん	22	ハクシコーラ	ハクシコーラ																																							
生徒	A太	B美																																								
クラス	2年3組	2年5組																																								
性別	男	女																																								
部活	マンガ部	テニス部																																								
授業を受ける○	授業を受ける○	授業を受ける○																																								
弁当を食べる○	弁当を食べる○	弁当を食べる○																																								
登校する○	登校する○	登校する○																																								

振り返り

次の各観点が達成されていれば□を塗りつぶしましょう。

- 情報システムの処理形態にはさまざまな種類のものがあることを理解した
- 3層クライアントサーバシステムとはどのようなものかを理解した
- データベースとは何かについての基本的な考え方を理解した
- データベース管理システムの役割を理解した

今日の授業を受けて思ったこと、感じたこと、新たに学んだことなどを書いてください。

.....

.....

.....

関係モデル

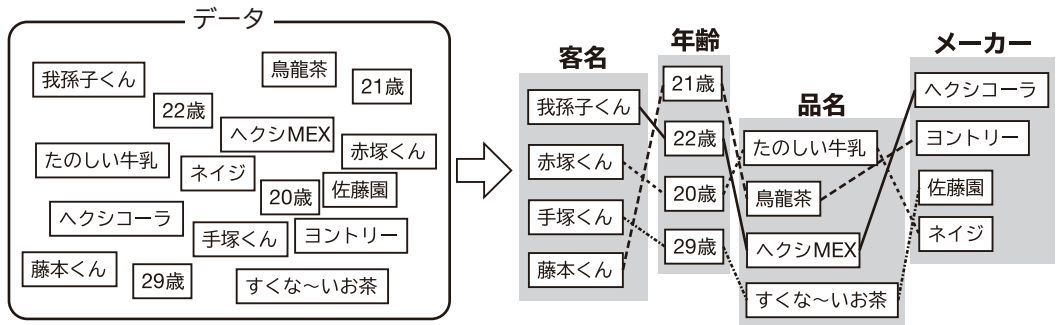
前節では、データベースの基本的な考え方について学びました。その中で、データベースにはさまざまなデータモデルがあることを紹介しましたが、現在、その中でもっともよく使われているのが関係モデルです。ここでは、関係モデルの考え方について学びます。

(教科書I : p.188 – p.189 , 教科書II : p.62 – p.p.65)

■ 関係モデル

関係（リレーション）モデルとは

一つひとつバラバラの〔¹ 〕の相互の**関係**をまとめると〔² 〕になる



↓ **関係を**〔⁴ 〕の形式にまとめる

客名	年齢	品名	メーカー
手塚くん	29	すくな〜いお茶	佐藤園
赤塚くん	20	たのしい牛乳	ネイジ
藤本くん	21	鳥龍茶	ヨントリー
我孫子くん	22	ハクシMEX	ハクシコーラ

〔⁵ 〕

〔⁶ 〕

※関係を表す表をテーブルという

関係スキーマ

関係スキーマ = 表の属性名を並べて記述したもの（表の見出しに相当する部分）

上記の関係における関係スキーマは次のようになる

購買表	⁷	⁸	⁹	¹⁰
------------	--------------	--------------	--------------	---------------

関連（リレーションシップ）

異なる関係の同じ属性を参照することで、関係同士の間に関連性を持たせることができる

関係【社員テーブル】

社員ID	氏名	部門コード	年齢
1001	佐藤 聡志	B01	35
1002	鈴木 錫夫	B02	37
1003	高橋 孝男	B03	29
1004	田中多奈美	B01	23
1005	伊藤唯斗志	B03	29

関係【部門テーブル】

部門コード	部門名
B01	総務部
B02	営業部
B03	開発部

{¹¹} ↑

{¹²}

主キー

主キー = 関係のレコードを一意に特定するためのフィールド

社員テーブルの主キー → {¹³}

部門テーブルの主キー → {¹⁴}

※主キーは、複数のフィールドで構成される場合もある

関連で参照する側のフィールドを**外部キー**という（社員表の**部門コード**）

実体関連図（ER図）

関係スキーマだけで表現すると、規模が大きくなると図が複雑になってしまう

→関係同士の間を線で表現

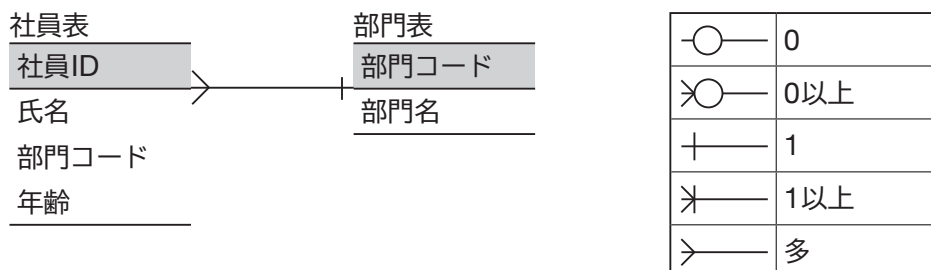
上の図から、実体関連図（E-R図）を描くと、次のようになる

15



IE記法

ER図は、次のように描かれることもある



※IE記法 = 情報エンジニアリング記法として、もっとも一般的に使用される記法

問題

次のように関係と関連が設定されているとき、下の各設問に教えてください。
ただし、すべての問題で解答欄はすべての欄が埋まるとは限りません。



[設問1]

各関係の関係スキーマを記述してください。また、関連を矢印で表してください。
主キー項目を太線で囲んでください。

20 (関連)

受注	16			顧客	17	
受注明細	18			商品	19	

[設問2]

関係と関連をもとに実体関連図 (E-R図) を描いてください。

21



[設問3]

- ①「6/27」に発注した顧客を教えてください。②「テレビA」、「テレビB」の受注数の合計を教えてください。

²²	顧客名

²³	商品名	受注数 (合計)
	テレビA	
	テレビB	

- ③「ステレオB」を発注した顧客を教えてください④「海山商事」が発注した商品名と、その商品の発注数の合計を教えてください。

²⁴	顧客名

²⁵	商品名	受注数 (合計)

- ⑤「川島電気」への請求金額を計算するために、下の表の各フィールドを埋めてください。

²⁶	商品名	単価	受注数	単価×受注数

- ⑥⑤より、「川島電気」への請求金額の合計を求めてください。

²⁷	円
---------------	---

■ データの正規化

データの正規化

テーブルからデータの冗長性や不整合を減らしていくことを**データの正規化**という
 次のようなデータから、データの正規化をする手順を説明する

科目コード	教科名	科目名	受講生		担当者
			生徒番号	氏名	
101	国語	現代文	001	伊藤 太郎	富田
			002	佐藤 花子	
			003	山田 一郎	
102	国語	古典	001	伊藤 太郎	山本
			002	佐藤 花子	
201	理科	化学	001	伊藤 太郎	鈴木
202	理科	物理	002	佐藤 花子	浜田
			003	山田 一郎	

第1正規形

表が単純に定義されている

科目コード	教科名	科目名	受講生		担当者
			生徒番号	氏名	
101	国語	現代文	001	伊藤 太郎	富田
			002	佐藤 花子	
			003	山田 一郎	
102	国語	古典	001	伊藤 太郎	山本
			002	佐藤 花子	
201	理科	化学	001	伊藤 太郎	鈴木
202	理科	物理	002	佐藤 花子	浜田
			003	山田 一郎	

フィールドが複合項目になっている

フィールドに繰り返しがある



科目コード	教科名	科目名	生徒番号	氏名	担当者
101	国語	現代文	001	伊藤 太郎	富田
101	国語	現代文	002	佐藤 花子	富田
101	国語	現代文	003	山田 一郎	富田
102	国語	古典	001	伊藤 太郎	山本
102	国語	古典	002	佐藤 花子	山本
201	理科	化学	001	伊藤 太郎	鈴木
202	理科	物理	002	佐藤 花子	浜田
202	理科	物理	003	山田 一郎	浜田

※見た目の分かりやすさを重視するあまり、非正規形のような表をつくってしまいがち
 →情報管理のためには、少なくとも第1正規形にはしておく必要がある

※第1正規形にするには、フィールド名はすべて1行で表記、同じデータもすべて表記
 →1行だけを取り出してもデータとして成立するようにする

第2正規形

あるフィールドの値が決まれば他のフィールドの値がすべて決まるように表を構成

生徒番号	組	氏名	科目コード	選択科目	担当	教科コード	教科名	点数
001	1組	伊藤 太郎	101	現代文	富田	1	国語	80
001	1組	伊藤 太郎	201	化学	鈴木	2	理科	96
002	2組	佐藤 花子	102	古典	山本	1	国語	50
002	2組	佐藤 花子	202	物理	浜田	2	理科	48
003	2組	山田 一郎	101	現代文	富田	1	国語	83
003	2組	山田 一郎	202	物理	浜田	2	理科	90



【成績テーブル】

生徒番号	科目コード	点数
001	101	80
001	201	96
002	102	50
002	202	48
003	101	83
003	202	90

【生徒テーブル】

生徒番号	組	氏名
001	1組	伊藤 太郎
002	2組	佐藤 花子
003	2組	山田 一郎

【科目テーブル】

科目コード	選択科目	担当	教科コード	教科名
101	現代文	富田	1	国語
102	古典	山本	1	国語
201	化学	鈴木	2	理科
202	物理	浜田	2	理科

※主キーを太枠で囲んで示した

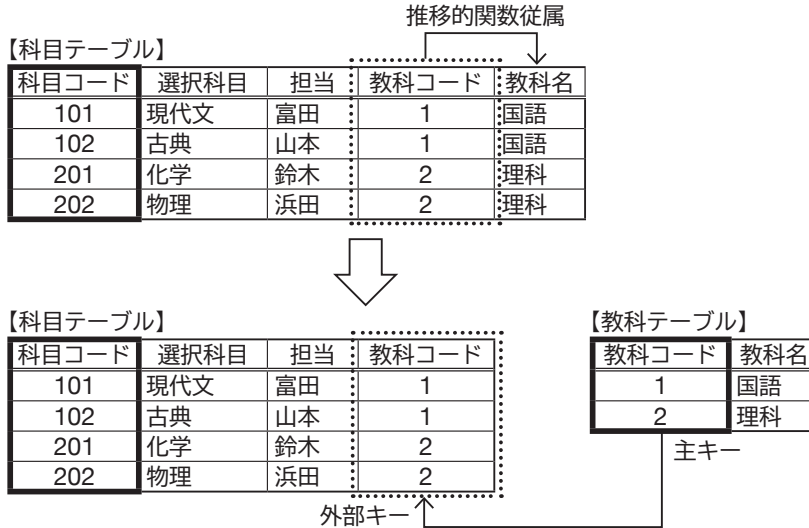
※あるフィールドの値により他のフィールドの値がひも付いていることを**関数従属**という

※主キー項目でテーブルは**完全関数従属**しているという

第3正規形

完全関数従属であり、**推移的関数従属**が存在しないように構成する

※**推移的関数従属** = 完全関数従属内で、あるフィールドから更に関数従属している



※第2正規形の科目テーブルでは、科目コードが決定すれば教科コードも教科名も決定
 →同テーブル内で教科コードが決定すれば、教科名も自動的に決定 → **推移的関数従属**

主キーと外部キー

主キー = 他のフィールドの値をすべて決定するようなフィールド

外部キー = 他のテーブルの主キーと同じ値でテーブルを結合できるようなフィールド

テーブルを正規化した際には、必ず**主キー**を設定するようにする

→テーブル内に主キーとなるような項目がない場合、新たにキーとなるフィールドを作る

→単純な連番を格納するだけのフィールドでも構わない

問題

次のテーブルを正規化してください。

主キー項目を太枠で囲んでください。

利用者ID	利用者名	商品ID	タイトル	ジャンル	貸出日
K113	草津 春夫	E235	花火	邦画	10/25
K113	草津 春夫	W225	三ツ矢怪談	ホラー	10/25
K113	草津 春夫	C313	Solar Wars	洋画	10/27
R201	栗東 夏子	K821	ポテト・チップス	コメディ	10/28
R201	栗東 夏子	E235	花火	邦画	10/28
M117	守山 秋江	C313	Solar Wars	洋画	10/29

①関係スキーマと関連を図にしてください。

28

利用者

--	--

31 (関連)

29

貸出

--	--	--

30

商品

--	--	--

②実体関連図 (ER図) を描いてください。

32



振り返り

次の各観点が達成されていれば□を塗りつぶしましょう。

関係モデルの基本的な考え方を理解できた

関係と関連から情報を読み取ることができた

データの正規化の手順を理解し、データを正規化することができるようになった

今日の授業を受けて思ったこと、感じたこと、新たに学んだことなどを書いてください。

.....

.....

.....

関係演算

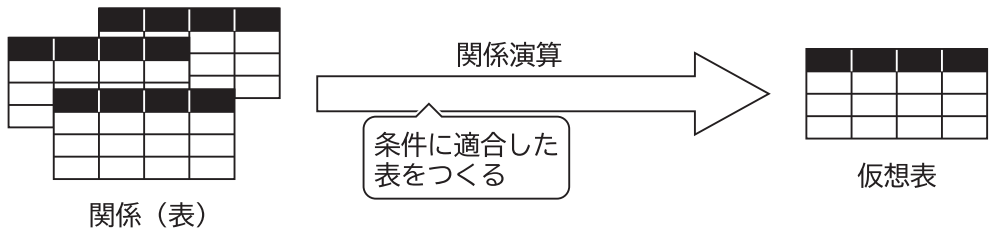
データベースは、蓄積されたデータから必要なデータを取り出して再利用することに意味があります。ここでは、関係演算と呼ばれるデータベースの演算を使い、大量のデータの中から必要なデータを取り出し、有用な「情報」にしていく術を学びます。

(教科書I : p.188 – p.189 , 教科書II : p.66 – p.67)

■ 関係演算

関係演算とは

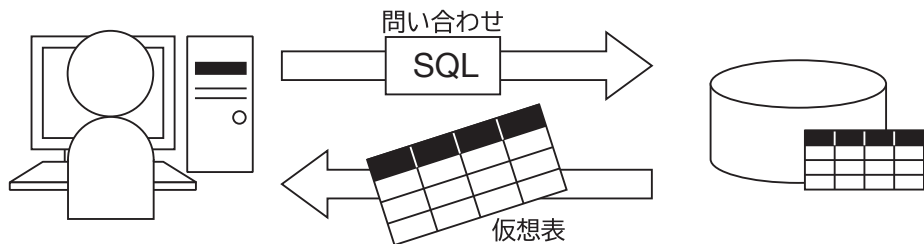
関係演算 = データベースの関係 (表) から条件に適合したデータを取り出す演算



※関係演算で取り出した関係 (表) を**仮想表**という

SQL

SQL = 関係データベースにおいて、データの操作を行うための問い合わせ言語



※SQLを使えば、様々なプログラムで同じデータを呼び出して使うことができる

■ SQL Online IDEの利用

SQL Online IDEの立ち上げ方と利用方法

①Webブラウザから下記URLにアクセスする

<https://sqliteonline.com>

②画面右上の「File」から「Open DB」を開く

③事前にClassroomからダウンロードしたDBファイルを選択し、「開く」を押す

用意したデータセット

あらかじめ、今回の実習を行なうためのデータセットを準備しておいた

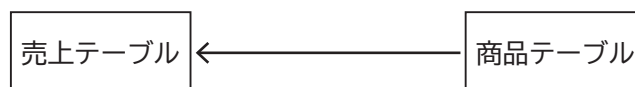
関係【売上テーブル】

売上コード	売上日	時間帯	年齢層	性別	商品コード
040111	4/1	朝	成年	女	S221
040112	4/1	朝	熟年	女	O113
040113	4/1	朝	熟年	男	S225
040114	4/1	朝	子ども	男	S221
040115	4/1	朝	若者	女	O115
040116	4/1	朝	子ども	男	S225
040121	4/1	昼	熟年	女	K117
040122	4/1	昼	若者	女	S225
040123	4/1	昼	熟年	女	P223
040124	4/1	昼	子ども	男	S225
040125	4/1	昼	子ども	男	O113
040131	4/1	夕方	成年	男	C207
040132	4/1	夕方	熟年	女	P223
040133	4/1	夕方	成年	男	C207
040134	4/1	夕方	若者	女	O115
040141	4/1	夜	若者	女	G201
040142	4/1	夜	若者	女	P223
040143	4/1	夜	子ども	男	P223
040144	4/1	夜	子ども	女	O115
040145	4/1	夜	成年	男	S225
.....

関係【商品テーブル】

商品コード	商品名	種類	価格
O113	おにぎり	ファストフード	120
O115	お弁当	ファストフード	360
K117	カップ麺	加工食品	110
S221	サンドイッチ	ファストフード	210
P223	パン	日配食品	120
S225	スナック菓子	加工食品	140
I321	アイスクリーム	日配食品	130
J103	ジュース	加工食品	150
G201	牛乳	日配食品	200
C207	コーヒー	加工食品	120

【実体関連図 (E-R図)】



■ SELECT文

SELECT文

関係データベースからデータを抽出するとき、SELECT文を使う
FROM句に関係名を指定すると、その関係のデータが取り出される

```
SELECT 列名
FROM 関係名
```

- ※各コマンドは必ず【半角】で入力すること
- ※コマンドとコマンドの間は【半角スペース】で間を開けること
- ※エラーメッセージが表示されても、慌てずやり直そう

関係の全データの表示

関係（表）に入っている全てのデータを表示させるには、列名に「*」を入力する

例題1

試しに、売上テーブルを表示してみよう

```
SELECT *
FROM 売上テーブル
```

NoSQL

Not only SQLの略称で、非構造化データを集めて蓄積する方法
関係データベース以外のデータベース管理システムの総称
データベースの構造を固定せず、構造の異なるデータを柔軟に扱う

1レコード	<pre>{ キー バリュー "名前": "ソードゴブリン", "HP": "149", "使用技": ["斬る", "流し斬り", "ゴブリンストライク"], "ドロップ": "ゴブリンソード" },</pre>	
1レコード	<pre>{ "名前": "ウェアウルフ", "HP": "382", "弱点": "冷", "使用技": ["アーマーブレイク", "ハウリング", "火炎放射"] }</pre>	異なるキーが存在

結合演算

結合演算 = ^{Join} 複数の関係（表）を1つの表にまとめる操作

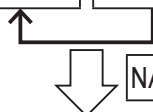
関係同士を接続するには、NATURAL JOIN句を使う

関係【売上テーブル】

売上コード	売上日	時間帯	年齢層	性別	商品コード
040111	4/1	朝	成年	女	S221
040112	4/1	朝	熟年	女	O113
040113	4/1	朝	熟年	男	S225
040114	4/1	朝	子ども	男	S221
040115	4/1	朝	若者	女	O115
.....

関係【商品テーブル】

商品コード	商品名	種類	価格
O113	おにぎり	ファストフード	120
O115	お弁当	ファストフード	360
K117	カップ麺	加工食品	110
S221	サンドイッチ	ファストフード	210
P223	パン	日配食品	120
.....



NATURAL JOIN 商品テーブル

売上コード	売上日	時間帯	年齢層	性別	商品コード	商品名	種類	価格
040111	4/1	朝	成年	女	S221	サンドイッチ	ファストフード	210
040112	4/1	朝	熟年	女	O113	おにぎり	ファストフード	120
040113	4/1	朝	熟年	男	S225	スナック菓子	加工食品	140
040114	4/1	朝	子ども	男	S221	サンドイッチ	ファストフード	210
040115	4/1	朝	若者	女	O115	お弁当	ファストフード	360
.....

※関係【売上テーブル】と関係【商品テーブル】をもとに1つの表にまとめた

例題2

上の図のように、【売上テーブル】と【商品テーブル】を結合してみよう

```
SELECT *
FROM 売上テーブル
NATURAL JOIN 商品テーブル
```

SQLで使う主な記号の意味

- 「*」（アスタリスク）：すべてを選択するときに使う
 - 「;」（セミコロン）：SQL文を終わりにするときに使う
 - 「'」（シングルクォーテーション）：文字列を指定するときに使う
- いずれも、記号は**半角**で入力するのを間違えないように！

射影演算

射影演算 = Projection 関係（表）の中から必要な**フィールド**（列）だけを取り出す操作

たとえば、【商品テーブル】の【商品名】と【価格】だけを**射影**するとは

商品コード	商品名	種類	価格
O113	おにぎり	ファストフード	120
O115	お弁当	ファストフード	360
K117	カップ麺	加工食品	110
S221	サンドイッチ	ファストフード	210
P223	パン	日配食品	120
.....



SELECT 商品名, 価格

商品名	価格
おにぎり	120
お弁当	360
カップ麺	110
サンドイッチ	210
パン	120
.....

※【商品名】と【価格】だけの表を取り出すことができた

※列名を並べるには、「,（半角のカンマ）」で区切る

例題3

【例題3】の状態から、【売上日】、【時間帯】、【商品名】だけの表を取り出してみよう

```
SELECT 売上日, 時間帯, 商品名
FROM 売上テーブル
NATURAL JOIN 商品テーブル
```

SQLは何の略？

SQLは1987年に国際標準化機構（ISO）により統一標準規格とされた標準規格としてのSQLは公式には「何かの略称ではない」とされている

SQLはIBMのデータベース操作言語「シークエSEQUEL」にちなんでいる

→構造化された英語で問い合わせる言語SEQUEL = Structured English QUERy Language

→名前通り、英語でデータを取り出すことができる言語として開発された

選択演算

^{Selection}
選択演算 = 関係（表）の中から条件に適合した**レコード**（行）だけを取り出す操作

たとえば、**[種類]** が '**ファストフード**' であるレコードを**選択**するとは

売上コード	売上日	時間帯	年齢層	性別	商品コード	商品名	種類	価格
040111	4/1	朝	成年	女	S221	サンドイッチ	ファストフード	210
040112	4/1	朝	熟年	女	O113	おにぎり	ファストフード	120
040113	4/1	朝	熟年	男	S225	スナック菓子	加工食品	140
040114	4/1	朝	子ども	男	S221	サンドイッチ	ファストフード	210
040115	4/1	朝	若者	女	O115	お弁当	ファストフード	360
.....

↓ WHERE 種類 = 'ファストフード'

売上コード	売上日	時間帯	年齢層	性別	商品コード	商品名	種類	価格
040111	4/1	朝	成年	女	S221	サンドイッチ	ファストフード	210
040112	4/1	朝	熟年	女	O113	おにぎり	ファストフード	120
040114	4/1	朝	子ども	男	S221	サンドイッチ	ファストフード	210
040115	4/1	朝	若者	女	O115	お弁当	ファストフード	360
040125	4/1	昼	子ども	男	O113	おにぎり	ファストフード	120
.....

※ **[種類]** フィールドの値が '**ファストフード**' のものが取り出された

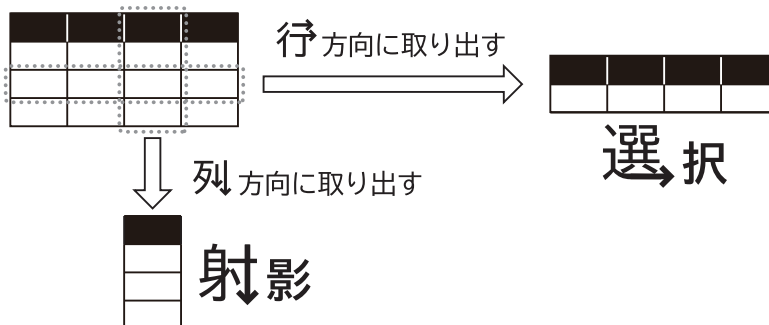
※値は必ず ' ' で囲む必要がある

例題4

上の図のように、**[種類]** が '**ファストフード**' であるレコードを選択してみよう

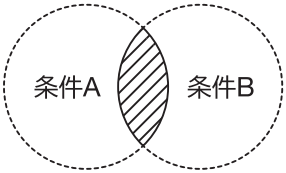
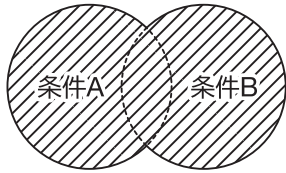
```
SELECT *
FROM 売上テーブル
NATURAL JOIN 商品テーブル
WHERE 種類 = 'ファストフード'
```

選択と射影の覚え方



AND検索、OR検索

WHERE句で**選択**する際、更に複数の条件を指定することができる

AND検索	OR検索
	
条件Aかつ条件Bの両方を満たすもの	条件Aまたは条件Bのどちらかを満たすもの
WHERE 年齢層 = '若者' AND 性別 = '女'	WHERE 時間帯 = '夕方' OR 時間帯 = '夜'

例題5

[年齢層] が '若者' で**かつ** [性別] が '女' のデータを取り出してみよう

```
SELECT *
FROM 売上テーブル
NATURAL JOIN 商品テーブル
WHERE 年齢層 = '若者' AND 性別 = '女'
```

例題6

[時間帯] が '夕方'**または** [時間帯] が '夜' のデータを取り出してみよう

```
SELECT *
FROM 売上テーブル
NATURAL JOIN 商品テーブル
WHERE 時間帯 = '夕方' OR 時間帯 = '夜'
```

問題1

- (1) [商品名] が 'おにぎり' で**かつ** [時間帯] が '夕方' に買ったのはどのような年齢層の人ですか。
- (2) [商品名] が 'アイスクリーム'**または** [商品名] が 'ジュース' の一覧の中で、4/3の朝に売れたのは、アイスクリームとジュースのどちらですか。

集計関数

SELECT文を使って集計をするための関数がいくつか用意されている
 集計関数を使う場合、フィールドごとにグループ化することが必要
 たとえば、この期間に種類ごとにいくつずつ売れたかを集計したい場合、

売上コード	売上日	時間帯	年齢層	性別	商品コード	商品名	種類	価格
040111	4/1	朝	成年	女	S221	サンドイッチ	ファストフード	210
040112	4/1	朝	熟年	女	O113	おにぎり	ファストフード	120
040113	4/1	朝	熟年	男	S225	スナック菓子	加工食品	140
040114	4/1	朝	子ども	男	S221	サンドイッチ	ファストフード	210
040115	4/1	朝	若者	女	O115	お弁当	ファストフード	360
.....

GROUP BY 種類

グループ化

種類	count(価格)
ファストフード	23
加工食品	32
日配食品	22

グループごとに
個数をカウント

SELECT 種類, COUNT(価格)

例題7

上の例のように、[種類] ごとの件数をカウントしてみよう

```
SELECT 種類, COUNT(価格)
FROM 売上テーブル
NATURAL JOIN 商品テーブル
GROUP BY 種類
```

問題2

- (1) 商品名ごとの件数をカウントし、お弁当の数を教えてください。
- (2) 年齢層ごとの件数をカウントし、子どもの数を教えてください。
- (3) 売上日, 時間帯ごとの件数をカウントし、4/3夜の件数を教えてください。

※GROUP BY句は最終行に書くようにしよう (WHERE句の方が先に書く)

合計 (SUM関数)

あるフィールドの値をグループごとに合計するには、SUM関数を使う
→引数には合計の計算をしたいフィールド名を指定する

```
SELECT グループ化フィールド, SUM(フィールド名)
FROM 関係1
NATURAL JOIN 関係2
GROUP BY グループ化フィールド
```

※グループ化するフィールドは、GROUP BY句で指定する

平均 (AVG関数)

あるフィールドの値をグループごとに平均するには、AVG関数を使う
→引数には平均の計算をしたいフィールド名を指定する

```
SELECT グループ化フィールド, AVG(フィールド名)
FROM 関係1
NATURAL JOIN 関係2
GROUP BY グループ化フィールド
```

※グループ化するフィールドは、GROUP BY句で指定する

例題8

(1) [種類] ごとに [価格] の合計を計算してみよう

```
SELECT 種類, SUM(価格)
FROM 売上テーブル
NATURAL JOIN 商品テーブル - - - - -
GROUP BY 種類
```

(2) [種類] ごとに [価格] の平均を計算してみよう

```
SELECT 種類, AVG(価格)
FROM 売上テーブル
NATURAL JOIN 商品テーブル
GROUP BY 種類
```

問題3

- (1) [商品名] ごとの [価格] の**合計**を計算し、お弁当の金額を求めてください。
- (2) [売上日] [時間帯] ごとの [価格] の**合計**を計算し、最も売上の高かった日と時間帯を求めてください。
- (3) [時間帯] ごとの [価格] の**平均**を計算し、最も売上平均の高かった時間帯を求めてください。
- (4) [年齢層] [性別] ごとの [価格] の**平均**を計算し、平均してもっとも多くの買い物をした年齢層と性別の組み合わせを教えてください。

身近な関係演算

内部にデータベースを持つアプリはたくさんある
→「ミュージック」アプリなどはその典型



アーティストをタップすると、アーティストの一覧に（射影演算）
さらにアルバムをタップすると、楽曲の一覧に → これらは**選択演算**
その後、アルバムを**選択**したり、楽曲を**選択**したり → **選択演算**

—— **身近なアプリにも関係演算が使われている**

振り返り

- 次の各観点が達成されていれば□を塗りつぶしましょう。
- 大量に蓄えられたデータから必要なデータを取り出すことができた。
- SQLの基本的な構文を理解し、使うことができるようになった。
- 失敗と試行を繰り返しながらチャレンジする態度が身に付いた。

今日の授業を受けて思ったこと、感じたこと、新たに学んだことなどを書いてください。

.....

.....

.....

データの操作

前節では、関係演算としてSQLのSELECT文について学び、すでにあるデータベースからデータを取り出す方法を学びました。ここでは、1からデータベースを定義し、データを挿入したり更新したりといったデータの操作方法について学びます。

(教科書II : p.86 – p.89)

■ データベースの定義

データの型

関係データベースの各フィールドは、入れることのできるデータの型が決まっている
→関係データベースの設計の際に、データの型を決めておかなければならない

主なデータ型

TEXT型	文字列型 ※ここに入った数字は文字としての数字が格納される
INTEGER型	整数型 ※小数点を含む数値を格納することはできない
REAL型	実数型 ※小数点を含む実数を格納することができる
NUMERIC型	数値型 ※整数は整数として、実数は実数として格納することができる
BLOB型	ファイル ※画像やテキストファイルなど、大きなファイルを格納

※他にも、一般的には日時を格納するDATE型やTIME型なども使われることがある

ナ^ル NULL値

NULL = 何もない、無の状態を意味する (ヌルと読む人もいる)

※数値の0とNULLは意味がまったく異なる → 0は計算が可能、NULLは計算不可

制約

NOT NULL	NULL値を禁止するには、このオプションを追加する
UNIQUE	フィールドに重複データを許容しない場合、このオプションを追加する

テーブルの定義

テーブルを定義するには、CREATE TABLE文を使う

```
CREATE TABLE テーブル名
(フィールド名1 型 オプション, ..., PRIMARY KEY(フィールド名))
```


やってみよう1

課題設定

ここはあるゲームセンター「わちゃわちゃタウン」です。「わちゃわちゃタウン」では、会員になると利用日に利用したゲームのハイスコアを記録してくれるサービスを行なっています。会員情報と会員のスコア情報を蓄えるデータベースを次のように作成しました。

データベース設計

各テーブルを次のように定義しました。

【会員】

フィールド名	データ型	制約
会員番号	TEXT	PRIMARY KEY
会員氏名	TEXT	NOT NULL
性別	TEXT	

【記録】

フィールド名	データ型	制約
会員番号	TEXT	PRIMARY KEY
ゲームID	TEXT	PRIMARY KEY
利用日	TEXT	PRIMARY KEY
プレイ時間	INTEGER	
スコア	NUMERIC	

【ゲーム】

フィールド名	データ型	制約
ゲームID	TEXT	PRIMARY KEY
ゲーム名	TEXT	NOT NULL

データベースの定義

CREATE TABLE文を使って、設計されたテーブルを作成してみよう。

```
CREATE TABLE 会員(
  会員番号 TEXT,
  会員氏名 TEXT NOT NULL,
  性別 TEXT,
  PRIMARY KEY (会員番号)
);
```

テーブル名
フィールド名 制約を並べ、で区切る
主キーの設定
1行目の(を閉じるための)

```
CREATE TABLE 記録(
  会員番号 TEXT,
  ゲームID TEXT,
  利用日 TEXT,
  プレイ時間 INTEGER,
  スコア NUMERIC,
  PRIMARY KEY (会員番号, ゲームID, 利用日)
);
```

```
CREATE TABLE ゲーム(
  ゲームID TEXT,
  ゲーム名 TEXT NOT NULL,
  PRIMARY KEY (ゲームID)
);
```

■ データベースの操作

データの挿入

データベースへのデータの登録をするには、INSERT INTO文を使う

```
INSERT INTO テーブル名 VALUES (値1,値2,...),(値1,値2,...)
```

やってみよう2

INSERT INTO文を使って、次のデータを挿入してみよう。

```
INSERT INTO ゲーム VALUES _____テーブル名
('101','ワニワニハンマー'), _____,で区切ってデータを順に登録
('103','ポケットストライク'),
('301','ストライクガンマン')
```

```
INSERT INTO 会員 VALUES
('07001','草津 春夫','男'),
('08002','栗東 夏子','女'),
('09005','守山 秋江','女')
```

```
INSERT INTO 記録 VALUES
('07001','101','08/17',20,40),
('07001','103','08/17',10,25),
('07001','301','08/17',10,5.6),
('08002','101','08/17',15,55),
('08002','301','08/17',10,3.4),
('09005','103','08/18',5,23),
('09005','301','08/18',15,7.4)
```

やってみよう3

[やってみよう2] を参考にして、各テーブルにデータを追加してみよう

- ① **ゲームテーブル**に、新しいゲームを1つ考えて追加しよう
- ② **会員テーブル**には、自分自身を登録してみよう
※性別はNULL値も登録可 (NULL値を登録する場合、' 'とすればOK)
- ③ **記録テーブル**には、新しいゲームも含めてスコアを記録してみよう
※スコアは、好きに設定してもよい (小数点も大丈夫)

データの更新

データベースのデータを書き換えるには、UPDATE文を使う

```
UPDATE 表名 SET フィールド名 = 値 WHERE 条件
```

※WHERE句で特定のレコードを選び出すところがポイント

※条件で複数行が選ばれた場合、すべてのレコードが書き換わるので注意

やってみよう4

UPDATE文を使って、次のデータを更新してみよう

草津春夫（07001）さんの、ワニワニハンマー（101）のスコアを45点に更新する

```
UPDATE 記録           _____テーブル名
SET   スコア = 45      _____書き換えるデータ
WHERE 会員番号 = '07001' AND ゲームID = '101'  _____レコードを選択
```

やってみよう5

[やってみよう3] で登録したデータを1つ更新してみよう

提出

メニューの「☰File→💾Save DB」を選ぶと、「sqlite.db」がダウンロード
→このファイルをClassroomからフォーム経由で提出

振り返り

次の各観点が達成されていれば□を塗りつぶしましょう。

データの型とはどういうことかを理解した

テーブルの定義をすることができた

データベースにデータを挿入し、データの操作をすることができた

今日の授業を受けて思ったこと、感じたこと、新たに学んだことなどを書いてください。

.....

.....

.....

集合演算 (和・差・積)

和集合演算

和集合演算 = 2つのテーブルのすべての行を合わせたテーブルを生成

商品コード	商品名	価格
O113	おにぎり	120
O115	お弁当	360
S221	サンドイッチ	210

+

商品コード	商品名	価格
S221	サンドイッチ	210
P223	パン	120
S225	スナック菓子	140



商品コード	商品名	価格
O113	おにぎり	120
O115	お弁当	360
S221	サンドイッチ	210
P223	パン	120
S225	スナック菓子	140

※共通のレコードは1行にまとめる

差集合演算

和集合演算 = 引くテーブルにある同一レコードを取り除く

商品コード	商品名	価格
O113	おにぎり	120
O115	お弁当	360
S221	サンドイッチ	210

-

商品コード	商品名	価格
S221	サンドイッチ	210
P223	パン	120
S225	スナック菓子	140



商品コード	商品名	価格
O113	おにぎり	120
O115	お弁当	360

積集合演算

積集合演算 = 両方に共通したレコードだけを取り出す

商品コード	商品名	価格
O113	おにぎり	120
O115	お弁当	360
S221	サンドイッチ	210

∩

商品コード	商品名	価格
S221	サンドイッチ	210
P223	パン	120
S225	スナック菓子	140



商品コード	商品名	価格
S221	サンドイッチ	210

SNSのしくみ

SNSの正体はデータベースシステム



ポストとフォローを実現する関係

XのようなSNSの場合、次のような関係で実現できる

関係【ポスト】

ID	userID	ポスト	日時
32	userA	今日は寒いな	01/07 13:12
33	userB	石がストーン!!	01/07 13:12
34	userA	カップルまじ爆発しろ	01/07 13:13
35	userC	久しぶりに雨降ってる	01/07 13:13
36	userD	昨日、『浅イイ話』観た人いる？	01/07 13:14
37	userB	二本の日本刀マジウケるwww	01/07 13:15

関係【フォロー】

userID	followID
userB	userC
userB	userD
userC	userB
userC	userE
userE	userB
userE	userC

フォローユーザのポストを取り出す関係演算

たとえば、userEがフォローしているユーザーのポストを表示する場合

①SELECT * FROM フォロー WHERE userID = 'userE'

userID	followID
userE	userB
userE	userC

②SELECT ID,userID,ポスト,日時 FROM ポスト

INNER JOIN フォロー

ON ポスト.userID = フォロー.followID 二つの関係を結合

WHERE フォロー.userID = 'userE' ①の選択と同じ

関係【ポスト】

ID	userID	ポスト	日時
33	userB	石がストーン!!	01/07 13:12
35	userC	久しぶりに雨降ってる	01/07 13:13
37	userB	二本の日本刀マジウケるwww	01/07 13:15

followID = 'userE'で

取り出された条件

章末問題

[問題]

次の図のような関係があったとき、各問に答えてください。

関係【生徒データ】

生徒番号	年	組	番	生徒名	委員会番号
17216	1	2	26	春山真琴	5
17189	1	4	23	夏生美織	1
17011	1	4	2	秋本葉月	9
17237	1	3	24	冬柴茉実	5
17083	1	4	7	金田由孝	5
.....

関係【委員会データ】

委員会番号	委員会
1	クラスリーダー
5	図書委員会
9	文化祭実行委員
.....

- (1) 秋本葉月の所属している委員会を答えてください。

- (2) 各関係の関係スキーマを書き、関係スキーマに関連の矢印を引いてください。また、主キー項目を太枠で囲んでください。ただし、すべての欄が埋まるとは限りません。

生徒データ

--	--	--	--	--	--

委員会データ

--	--	--	--	--	--

- (3) 実体関連図を完成させてください。

生徒データ

委員会データ

- (4) 図書委員会の名簿を作成するためのSQL文として正しいものを選んでください。

ア.SELECT * FROM 委員会データ

イ.SELECT 年,組,番,氏名 FROM 生徒データ WHERE 組='4'

ウ.SELECT 委員会,年,組,番,氏名 FROM 生徒データ

NATURAL JOIN 委員会データ WHERE 委員会='図書委員会'

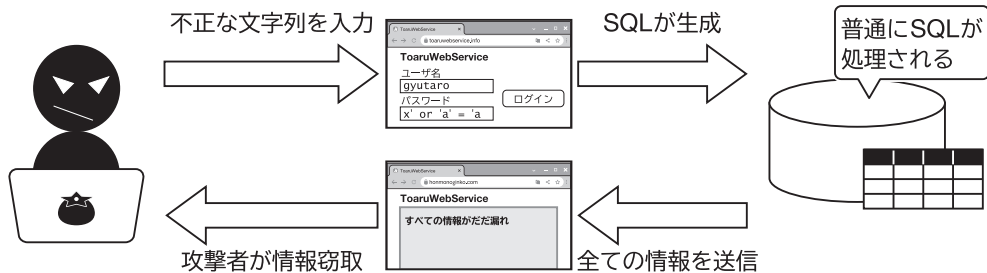
エ.SELECT 委員会,COUNT(*) FROM 生徒データ NATURAL JOIN 委員会データ
GROUP BY 委員会

コラム～SQLインジェクション攻撃

■SQLインジェクション攻撃

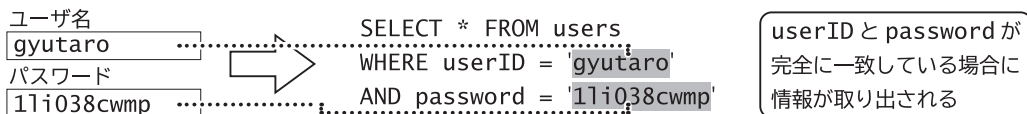
SQLインジェクション攻撃

SQLインジェクション攻撃 = 不正なSQL文を送信し、情報窃取、改ざんなどの攻撃



SQLインジェクション攻撃の仕組み

ユーザ名、パスワードを入力すると、アプリが入力文字列をもとにSQL文を生成



ここで、パスワードに「x' or 'a'='a」と入力すると、生成されるSQL文は

```
SELECT * FROM users
WHERE userID = 'gyutaro'
AND password = 'x' OR 'a'='a'  ———— 'a'='a'は必ず真
```

userID='gyutaro' AND password='x'はパスワードが違うため偽となるが、ORにある'a'='a'が必ず真になるので、WHERE句の条件は全体としては真となる
→結果的にパスワードはスルーされ、真であるレコードがすべて取り出されることに

SQLインジェクション攻撃への対策

例えば入力された'記号を"に置き換えるなどの対策が考えられる

```
SELECT * FROM users
WHERE userID = 'gyutaro'
AND password = 'x" OR "a"="a'  ———— 網掛け部分がパスワード扱いに
```

SQLでは'はただの記号として扱われる→パスワードが「x" OR "a"="a」と扱われる
→userIDとpasswordが一致しないため、このSQL文は1行も結果を返さない

