

# リレーショナルデータモデル学習支援のためのSVGを用いた解説表示機能の試作

藤間 黄純<sup>1</sup> 岡田 信一郎<sup>2</sup>

**概要:** 膨大なデータの調査や管理が必要とされる現代の情報社会では、データベースについて学習することの重要性は高いといえる。筆者らは、データベースの基礎であるリレーショナルデータモデルについて習得するための「リレーショナルデータモデル演習システム」の開発と改良を行ってきた。本システムは、8種のリレーショナル代数問題と3種の正規化問題の自動生成と学習者解の正誤判定の機能を備え、多くの演習問題による学習を可能とするが、システム導入後も正規化問題を難しいと感じる学習者から問題に関する解説が欲しいという意見が寄せられていた。そこで今回筆者らは、学習者の理解度向上のために、自動生成された個々の問題に合わせた解説をSVGを用いて生成、表示する機能の試作を行った。

## 1. はじめに

膨大なデータの調査や管理が必要とされる現代の情報社会では、データベースについて学習することの重要性は高いといえる。本研究室では、現代のデータベースの基礎となっているリレーショナルデータモデルについて理解を深めるための「リレーショナルデータモデル演習システム」を開発し、本学の情報工学科2年生を対象に開講されている「データベース論」の授業で運用と評価を行ってきた[1]。本システムは、8種のリレーショナル代数問題と3種の正規化問題の自動生成と学習者解の正誤判定機能を備えており、多くの演習問題による学習が可能である。評価実験の結果[1]、本システム導入後も正規化問題を難しいと感じる学習者から問題に関する解説が欲しいという意見が寄せられていた。そこで今回筆者らは、学習者の理解度向上のために、自動生成された個々の問題に合わせた解説をSVGを用いて生成、表示する機能の試作を行った。

## 2. 背景

### 2.1 システム概要

リレーショナルデータモデル演習システムは、集合論に基づいて考案されたリレーショナルデータモデルの学習を支援するために開発されたデータベース初学者向けシステムである。本システムは、「リレーショナル代数」と「正規

化」に関する問題の自動生成機能と学習者解の自動判定機能を備えており、学習者はこれらの問題を繰り返し解くことでリレーショナルデータモデルについて理解を深めることができる。本システムはクライアントアプリケーションとサーバアプリケーションから構成されている。クライアント側は問題の表示や解答の入力を担当し、サーバ側では問題生成や正誤判定、学習履歴の管理を担当している。クライアントアプリケーションはJavaFXアプリケーションとして作成されており、GUIにはJavaFXの標準コントロールを用いている。クライアントアプリケーションの実行画面例を図??に示す。データベース教育に関する関連研究としては、「情報システムにおけるデータベースの仕組みを学ぶ共通教科「情報」の授業の開発と評価」[3]などがあるが、本システムはデータベース初学者向けのシステムであるが、情報分野を専門とする大学生を対象としており、リレーショナルデータモデルの理論を習得させ、技術者として活用できるまで習熟させることを目的としている。

### 2.2 機能概要

#### ● 問題生成機能

クライアント側からサーバ側に新しい問題の要求があった場合に、成績テーブルのデータを参照し、学習者が合格していない学習項目の問題を生成する機能である。「リレーショナル代数」には「和演算」「差演算」「商演算」「共通演算」「直積演算」「射影演算」「選択演算」「結合演算」の8つの学習項目が、「正規化」には「第2正規化」「第3正規化」「ボイスコード正規化」の3つの学習項目がある。

<sup>1</sup> 茨城大学大学院理工学研究科  
Graduate School of Science and Engineering, Ibaraki University

<sup>2</sup> 茨城大学工学部  
Faculty of Engineering, Ibaraki University

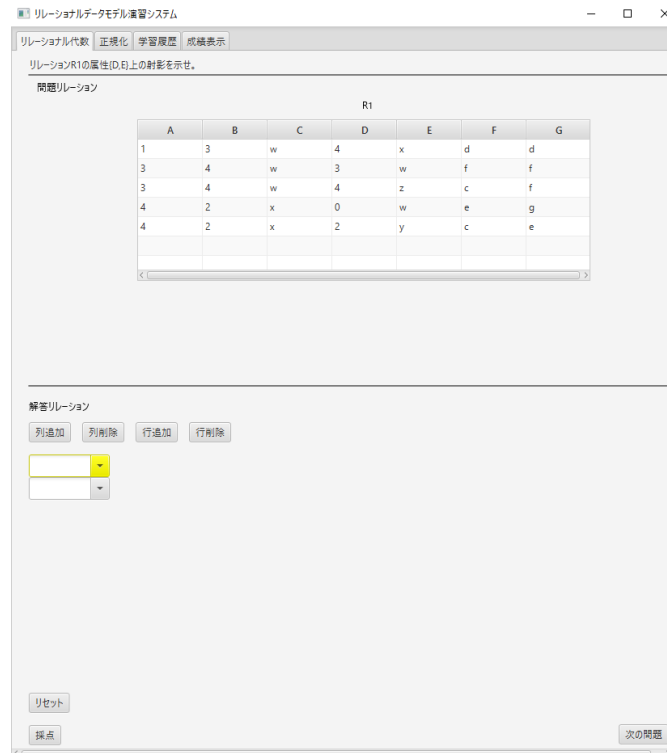


図 1 リレーショナルデータモデル演習システム

- 問題表示機能

クライアントアプリケーション画面に、問題生成機能で生成した問題を表示する機能である。問題文を表示するために Label コントロールを、問題リレーションを表示するために TableView コントロールを使用している。

- 解答入力機能

学習者が解答リレーションを入力するための機能であり、ドロップダウンリストかキーボードで入力する。

- 正誤判定機能

学習者が入力した解答リレーションと正解例リレーションを比較して正解かどうか判定する機能である。

- 正解表示機能

正誤判定機能による結果と正解例リレーションを表示する機能である。

- 学習履歴管理機能

学習者が解答した問題の履歴を学習履歴として管理、表示する機能である。学習履歴はデータベースに保管されていて、必要に応じて詳細を確認することができる。

- 成績管理機能

学習者の成績を管理、表示するための機能である。学習項目ごとに「出題回数」「正解数」「理解度」が表示される。学習項目はそれぞれの学習項目別に最新の 5 問のうち 3 問以上正解となると合格となる。

## 2.3 これまでの運用における要望

本システムは、本学情報工学科 2 年生を対象に開講されている「データベース論」の授業で演習課題として運用されている。学習者は本システムに関して説明を受けた後、自由時間に本システムを利用して学習を行う。演習課題の締め切りは提示日から 3,4 週間程度であり、期間内に全ての学習項目に合格することが求められている。運用後には本システムを利用した学生にアンケートを取り、システムの改良に役立てている。令和 1 年度に行った評価実験では、70 名からアンケートの回答が得られた。アンケート設問 14「その他、こうした方がいい、こうして欲しいと思う所、欲しい機能はありますか？」に対して、「なぜその何次正規形になっているかの解説が欲しかった。」「正規化の問題で、候補キーを与えたほうが良い。」といった意見が寄せられた。また、不合格のトピックがあった学習者から「正規化が正答できなかった」という回答があった。

## 3. 解説表示機能の試作

### 3.1 目的

本システムは学習者の解答入力に対して自動で正誤判定を行うことが可能であるが、正解表示画面では正解か不正解かという判定結果と正解例リレーションの表示されるのみであった。学習者は自身の入力と問題文、正解例リレーションを比較することで不正解の原因を理解し、リレーショナルデータモデルに関する各トピックについて理解を深めることは可能ではあったが、運用後に実施したアンケートで

は解説を表示して欲しかったという意見が寄せられていた。よって筆者らは、学習者の理解度向上を目的として、自動生成された個々の問題に合わせた解説を SVG を用いて生成し、表示する機能の試作を行うこととした。

### 3.2 開発方針

表示する解説は「データベース論」の授業で配布される資料を参考に、問題を解く際に重要となる注目すべきタプルや属性の強調表示や演算前と演算後のリレーションがどういった関係であるのかといったことを中心に作成した。本システムでは、学習を行う際に授業で配布された資料を参考に問題を解くことを想定しているが、自動生成された個々の問題に合わせた解説を表示することで、学習者が各トピックについて理解することの手助けとなることを目指す。また、今回試作機能を実装するにあたって、本システムの正解表示機能部の改良を行った。これまで問題リレーション、解答リレーション、正解例リレーションの表示に JavaFX の標準コントロールである TableView を用いていたが、TableView では表現に限界があるため、SVG 形式の解説を表示できるように WebView コンポーネントを用いることとした。WebView コンポーネントは JavaFX の UI コンポーネントの一つであり、HTML5 の機能をサポートしているため SVG タグ要素の描画が可能である。SVG は、XML をベースとした 2 次元ベクタ形式の画像ファイル形式の一つであり、XML に基づくマークアップ言語であるため、XML 特有の各種機能を定義した要素を持っている。画像を図形の集合として表現するベクタ形成であるため拡大、縮小しても品質を損なうことなく、レイヤ機能を有するため写真や挿絵などのビットマップデータと共存させることもできる。また、テキストデータとして、JavaScript で操作することも可能であるため、自動生成された問題に合わせた解説を生成するのに適切であると判断した。

### 3.3 生成する解説

今回試作した解説生成機能で、問題トピックごとにどのような解説を表示するのか以下に述べる。

#### 3.3.1 リレーショナル代数

リレーショナルデータモデルではリレーションに対する演算として、4 つの集合演算と 4 つのリレーショナル代数特有の演算を定義している。これらの演算によって、リレーショナルデータベースのデータを操作することが可能である。リレーショナル代数問題において、演算の対象となるリレーションをそれぞれ  $R_1, R_2$  と表記する。

##### (1) 和集合演算

和集合演算は次のように定義される。ここで、 $t$  はタプルである。

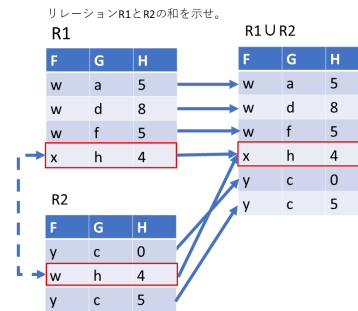


図 2 和集合演算例

$$R_1 \cup R_2 = \{t | t \in R_1 \vee t \in R_2\}$$

和両立を満たすことから、 $R_1$  と  $R_2$  の属性数と対応するドメインは等しくなっている。 $R_1$  と  $R_2$  に含まれるタプルを集めたリレーションが演算の結果となる。しかし、 $R_1$  と  $R_2$  に含まれるタプルであっても重複要素はみとめられないため、演算後のリレーションでは 1 つしか含まれない。解説は図 2 のように、 $R_1$  と  $R_2$  のタプルを比較し、重複タプルが存在したら rect 要素で強調表示するとともに点線矢印でどれどれが重複しているのか視覚的にわかりやすくする。また、それぞれのリレーションのタプルが演算後のリレーションのどのタプルに対応しているのかわかるように矢印を表示している。

##### (2) 差集合演算



図 3 差集合演算例

差集合演算は次のように定義される。ここで、 $t$  はタプルである。

$$R_1 - R_2 = \{t | t \in R_1 \wedge \neg(t \in R_2)\}$$

和両立を満たすことから、 $R_1$  と  $R_2$  の属性数と対応するドメインは等しくなっている。 $R_1$  に含まれているタプルかつ、 $R_2$  に含まれていないタプルを抜き出して合わせたリレーションが演算の結果となる。解説は和集合演算と同様に、 $R_1$  と  $R_2$  のタプルを比較し、重複タプルが存在したら rect 要素で強調表示するとともに、点線矢印でどれど

どれが重複してるのか視覚的にわかりやすくしている。また、 $R_1$  のタプルで、 $R_2$  との重複がないタプルから演算後のリレーションの対応するタプルへ矢印を表示している。

(3) 共通集合演算

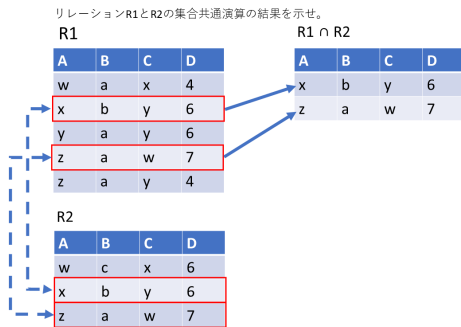


図 4 共通集合演算例

共通集合演算は次のように定義される。ここで、 $t$  はタプルである。

$$R_1 \cap R_2 = \{t | t \in R_1 \wedge t \in R_2\}$$

和両立を満たすことから、 $R_1$  と  $R_2$  の属性数と対応するドメインは等しくなっている。 $R_1$  と  $R_2$  の両方に含まれているタプルを抜き出して合わせたリレーションが演算の結果となる。解説は図 4 のように、 $R_1$  と  $R_2$  の重複タプルを点線矢印で表示し、これらのタプルが演算後リレーションのどのタプルと対応しているのかを矢印で表示した。

(4) 直積集合演算

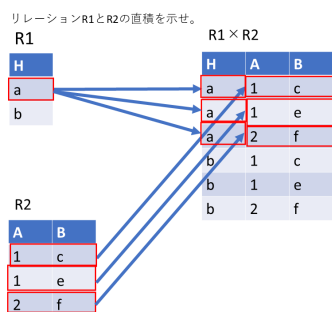


図 5 直積集合演算例

直積集合演算は次のように定義される。ここで、 $t_1, t_2$  は  $R_1, R_2$  のタプルである。

$$R_1 \times R_2 = \{(t_1, t_2) | t_1 \in R_1 \wedge t_2 \in R_2\}$$

ここで、

$$t_1 = (a_1, a_2, \dots, a_n), t_2 = (b_1, b_2, \dots, b_m)$$

とすると

$$(t_1, t_2) = (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)$$

は  $n+m$  項のタプルとなる。直積集合演算では、 $R_1$  と  $R_2$  に含まれるタプルをすべて掛け合わせて新しいリレーションを形成する。解説は図 5 のように、演算後のリレーションのどの要素がどの  $R_1$  のタプルと対応しているのか、また同様に  $R_2$  のどのタプルが対応しているのか矢印で表示した。

(5) 射影演算



図 6 射影演算例

$R_1$  の全属性集合の部分集合を  $X = \{A_{i1}, A_{i2}, \dots, A_{ik}\}$  とする。このとき、 $R_1$  の  $X$  上の射影  $R_X$  は次のように定義される。

$$R_X = \{ulu \in dom(A_{i1}) \times dom(A_{i2}) \times \dots \times dom(A_{ik}) \wedge (\exists t \in R_1)\}$$

射影演算は、 $R_1$  の属性を指定してリレーションを縦方向に切り出している。ここで、指定する属性は単体でも複数でもよい。解説は図 6 のように、切り出す属性を強調表示し、解答リレーションはこれらの属性を合わせたリレーションであることがわかるようにした。また、属性で切り出したことによって重複タプルが発生した場合には、解答リレーションでは1つにまとめなければいけないため、重複タプルが発生していることがわかるように点線矢印で表示した。

(6) 選択演算

リレーションから条件を満たすタプルを抜き出す演算である。 $A_i$  と  $A_j$  が比較可能であるとき、 $R_1$  の属性  $A_i$  と  $A_j$  選択演算を  $R[A_i \theta A_j]$  と表す。選択演算  $R[A_i \theta A_j]$  は次のように定義される。

$$R[A_i \theta] = \{t | t \in R \wedge t[A_i] \theta t[A_j]\}$$

選択演算には、リレーションの属性と属性を比較するパターンと属性と固定値を比較するパターンがある。属性の条件

リレーションR1に対する属性EとGの選択演算(属性Eの値=属性Gの値を条件とする選択演算)の結果を示せ。

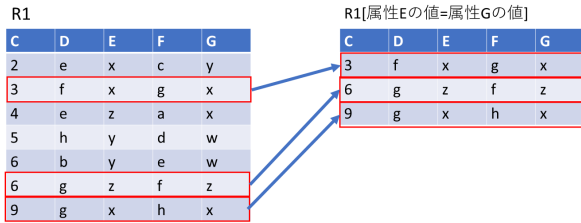


図 7 選択演算例

となる演算子は、文字列の比較であるなら”=”か”neq”のみとなるが、数字であった場合は”>”, ”<”, ”≤”, ”≥”の中から選択される。 $\theta$  はこれらの比較演算子の何れかであることを表している。解説は図 7 のように、条件を満たしているタプルを強調表示し、演算後のリレーションのどのタプルに対応しているか矢印で表示している。

(7) 結合演算

リレーションR1とR2の結合を示せ。なお、R1のHとR2のCを結合属性とする。

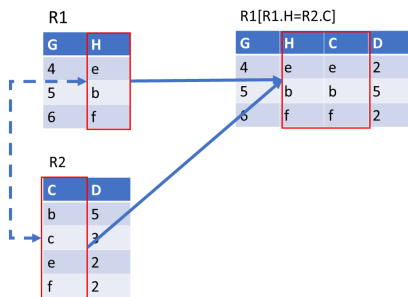


図 8 結合演算例

$R_1$  と  $R_2$  の条件を満たしているタプル同士を結合して、1つのリレーションにする演算である。 $R_1(A_1, A_2, \dots, A_n)$  と  $R_2(B_1, B_2, \dots, B_m)$  の  $A_i$  と  $B_j$  が比較可能であるとき結合演算は次のように定義される。

$$R_1[A_i \theta B_j] R_2 = \{(t, u) | t \in R_1 \wedge u \in R_2 \wedge t[A_i] \theta u[B_j]\}$$

解説は図 8 のように、結合属性として指定された  $R_1$  の属性と  $R_2$  の属性のうち条件を満たしている要素を強調表示し、演算後のリレーションのどのタプルに対応しているのかを矢印で表示している。

(8) 商演算

商演算は、 $R_2$  の全ての属性の値を条件に  $R_1$  のタプルを抜き出して 1 つにまとめる演算である。 $R_1(A_1, A_2, \dots, A_{n-m}, B_1, B_2, \dots, B_m)$  を  $n$  次、 $R_2(B_1, B_2, \dots, B_m)$  を  $m(m \leq n)$  次のリレーションとする。 $R_1$  を  $R_2$  で割った商は次のように定義される。

リレーションR1とR2の商を示せ。

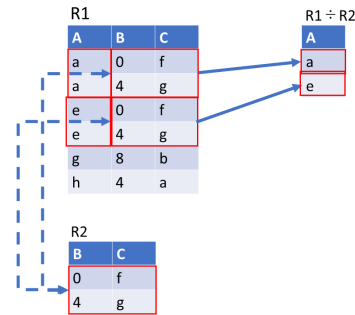


図 9 商演算例

$$R_1 \div R_2 = \{t | t \in R_1(A_1, A_2, \dots, A_{n-m}) \wedge (\forall u \in R_2)((t, u) \in R_1)\}$$

ここで、解答リレーションを  $S$  とすると  $R_1 \div R_2 = S$  となり、 $R_1 = R_2 \times S$  と変形できる。本システムでは、この変形を利用し、 $R_2$  と解答リレーションの直積をとってから、行を追加することで問題リレーション  $R_1$  を生成している。解説は図 9 のように、 $R_2$  の属性の値を組として条件を満たしている  $R_1$  のタプルを強調表示し、演算後のリレーションのどのタプルに対応しているかを矢印で表示している。

3.3.2 正規化

正規化問題を解くには、リレーションがどのような候補キーを持っているか理解することが重要となる。候補キーとは、その値の組がリレーションのタプルを唯一に識別できる属性の組のうち、1 つでも欠けると 1 意識別能力を持たなくなる属性の組のことである。候補キーはリレーションに少なくとも 1 つ存在し、2 つ以上の属性の組となる場合もある。本システムでは、まず解答リレーション  $R_1, R_2$  を生成し、それらを結合することで問題リレーションを生成している。2 つの解答リレーションをはそれぞれ問題の条件に合わせた関数従属性が存在するように要素が設定されており、結合するための属性を持っている。またこのとき、問題リレーションと解答リレーションの間には、情報無損失分解が成り立っている。

(1) 第 2 正規化

リレーションRは(甲,乙)→(丙)、(甲,乙)→(丁)、(甲)→(丁)なる関数従属性を持つ。Rをより高次の正規形へ正規化せよ。また、この問題リレーションは最大何次元正規形か答えよ。

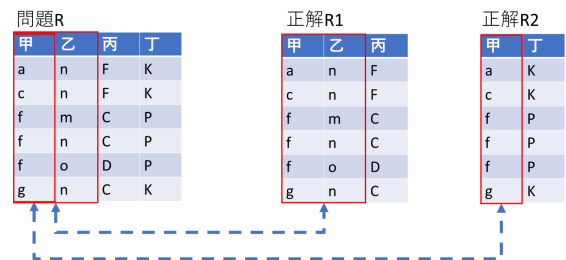


図 10 第 2 正規化例

リレーションが第 2 正規形であるためには、リレーション

が第1正規形の条件を満たしていることとリレーシヨンの全ての非キー属性がリレーシヨンの各候補キーに完全関数従属している必要がある。ここで、非キー属性とは、いかなる候補キーにも属していない属性のことである。第2正規化問題の問題リレーシヨンには、第2正規形違反が発生しており、候補キーの一部に関数従属性をもつ部分関数従属があり、これを取り除くことで第2正規形に正規化が可能となっている。図10の例では、問題リレーシヨンRは{甲, 乙} → {丙}, {甲, 乙} → {丁}, {甲} → {丁}なる関数従属性を持っている。つまり、属性甲, 乙の値に対して属性丙及び属性丁の値が、属性甲に対して丁の値が唯一に決まることを意味している。ここで、非キー属性である属性丁が、候補キーの一部である属性甲に関数従属しているため、これを取り除くことで第2正規形に正規化が可能である。解説は図10のように、問題リレーシヨンにおける候補キーを強調表示するとともに、正規化後のリレーシヨンのどの属性と対応しているか点線矢印を表示することで第2正規形違反を理解しやすいようにした。

(2) 第3正規化

リレーシヨンRは{甲}→{乙}, {甲}→{丙}, {乙}→{丁}なる関数従属性を持つ。Rをより高次の正規形へ正規化せよ。また、この問題リレーシヨンは最大何次正規形か答えよ。

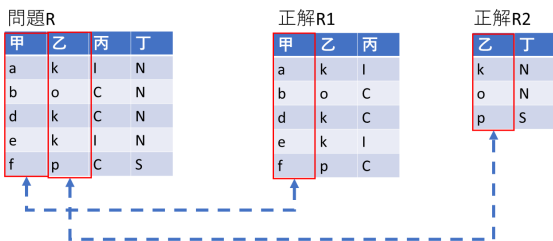


図11 第3正規化例

リレーシヨンが第3正規形であるためには、リレーシヨンの全ての非キー属性はリレーシヨンのいかなる候補キーにも推移的関数従属していない必要がある。つまり第3正規形違反は、推移的関数従属が存在することであり、問題リレーシヨンにおいてこれを取り除くことでより高次の正規形に正規化することができる。解説は図11のように、問題リレーシヨンにおける候補キーを強調表示するとともに、正規化後のリレーシヨンのどの属性と対応しているか点線矢印を表示している。

(3) ボイスコード正規化

$X \rightarrow Y$  をリレーシヨンの関数従属性とするとき、 $X \rightarrow Y$  は自明な関数従属性であるとともに属性 X がリレーシヨンのスーパーキーであるなら、リレーシヨンはボイスコード正規形であるといえる。スーパーキーとは、その値がリレーシヨンのタプルを唯一識別できる属性集合のことである。ボイスコード正規形違反は、非キー属性から候補キー属性

リレーシヨンRは{甲,乙}→{丙}, {丙}→{乙}なる関数従属性を持つ。Rをより高次の正規形へ正規化せよ。また、この問題リレーシヨンは最大何次正規形か答えよ。

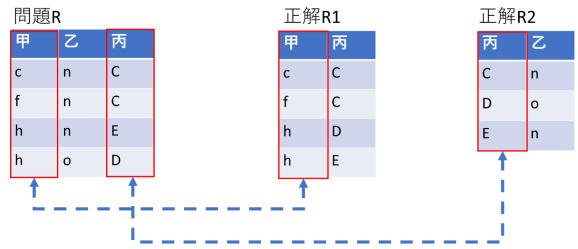


図12 BC正規化例

に対して関数従属性があることであり、問題リレーシヨンにおいてこれを取り除くことでより高次の正規形に正規化することができる。解説は図12のように、問題リレーシヨンにおける候補キーを強調表示するとともに、正規化後のリレーシヨンのどの属性と対応しているか点線矢印を表示している。

4. 実現方式

4.1 システム構成

解説を生成, 表示するシステムの構成は図13のようになっている。

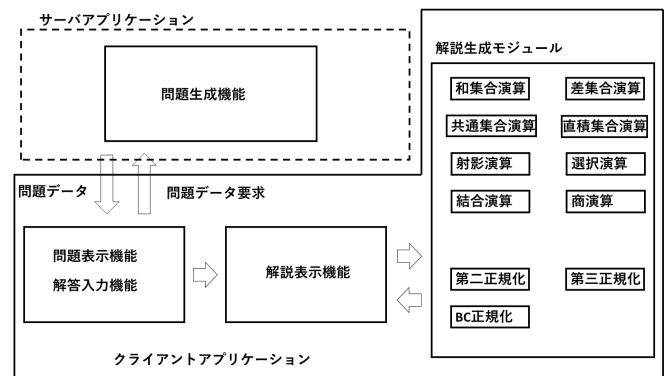


図13 システム構成

学習者が解答を入力し、サーバアプリケーションに送信すると、自動で正誤判定が行われる。正誤判定をクライアントアプリケーションが受けると、解説表示機能に解説生成の要求が送られ、解説生成モジュールで問題トピックごとの解説が自動で生成される。解説は自動生成された問題のトピック情報及び、問題リレーシヨンと正解例リレーシヨンを分析することで生成している。解説表示機能の画面構成は図14のようになっている。生成した解説は解説表示エリアで表示される。また、従来のシステムでは問題リレーシヨン, 解答リレーシヨン, 正解例リレーシヨンは

TableView を用いて表示していたが,SVG を用いて表示することとした. これは, TableView を用いると見やすいリレーションを表示するためには, 画面領域の確保が必要であり, 解説表示エリアを追加したことによって, 画面領域が圧迫される問題を解決するためである. 図 14 に示した画面はリレーショナル代数問題の場合であり, 正規化問題の場合には若干異なる構成となっている. 正規化問題では, 解答リレーションと正解例リレーションは 2 つとなるため, それらを上下に配置している. また, 問題リレーションの最大正規形を答える問題の正誤判定結果を表示するエリアを解答リレーションの正誤判定表示エリアの下に配置している.

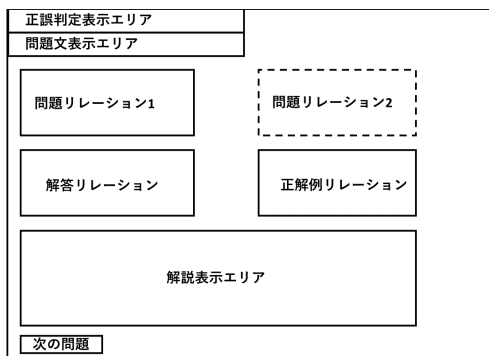


図 14 解説表示画面

#### 4.2 解説生成

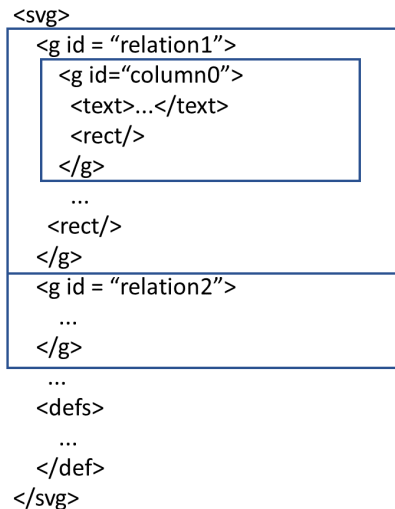


図 15 SVG データの構造

生成する解説の SVG データの大まかな構造は図 15 のようになっている. 1 番外側の要素は svg 要素であり,SVG データとしてのサイズ定義や viewBox 属性の設定をして

いる. リレーションは問題リレーションと正解例リレーションのように複数存在するため, それぞれを g 要素を用いてグループ化している. グループ化しておくことで, リレーション単位での検索や操作が可能となっている. リレーションの属性名やセルは rect 要素と text 要素を組み合わせることで表現している. これらの要素もまた, g 要素でグループ化しているため, 属性名単位やセル単位での操作が可能である. defs 要素は解説要素として使用する矢印といった描画オブジェクトを定義しておくための要素である. グループ化したリレーションは原点座標を基準に描画した後, transform 属性を設定することで適切な位置になるよう移動している. リレーションとリレーションの関係性を示すための矢印などの要素は, グループ化してしまうと transform 属性で一緒に移動してしまい, 表示が崩れるので, svg 要素の下に配置している. SVG 要素の生成には, Apache Batik の提供する Java のライブラリ [5] を用いており, SVG を Documet データとして生成し, それを HTML のコンテンツとして WebView コンポーネントに表示する.

#### 5. まとめと今後の課題

本研究では, リレーショナルデータモデル演習システムの学習効果を向上させるために, 解説表示機能の試作と実装を行った. 今回実装したのは, 問題のトピックに合わせた解説を SVG を用いて生成する機能であり, SVG を用いるために SVG の要素を生成する機能の実装も行った. また, 生成した解説を表示する領域を確保するために, 正解表示機能における表示画面をサブウィンドウに表示するようするという変更を行った. 今回実装した解説表示機能により, 自動生成された個々の問題に合わせた解説を表示することで, その問題に対する理解度を向上させられると考える. 今後課題としては, 今回試作した解説はその問題に解答するにあたり重要なタプルの強調表示や関係性を理解しやすくするための矢印などといった簡単なものでしかなく, 学習者がリレーショナルデータモデルについて理解する助けになるのに十分であるとはいえない状態であるため, さらに高度な解説を生成することを目指すことが挙げられる. また, 来年度の「データベース論」の授業で解説表示機能を実装したリレーショナルデータモデル演習システムを運用評価し, 学習向上効果があるのか検証するとともに, 学習者から意見を求め, 機能のさらなる向上を目指したい.

#### 参考文献

- [1] 岡田信 1 郎, 阿部孝昭, 山縣大輔, “リレーショナルデータモデル演習システムのための正規化問題生成機能の開発と運用評価,” 情報処理学会研究報告, Vol.2017-CE-138, No.17, 2017 年 2 月.
- [2] 伊藤豪, 岡田信 1 郎, “リレーショナルデータモデル演習システムの試作と評価,” 2014 年電子情報通信学会総合大

- 会, D-15-7(2014).
- [3] 白井詩沙香, 長瀧寛之, 竹中 1 平, 武本康宏, 田邊則彦, 兼宗進, “情報システムにおけるデータベースの仕組みを学ぶ共通教科「情報」の授業の開発と評価,” 情報処理学会論文誌 教育とコンピュータ, Vol.5, No.3, pp.23-34, (Oct. 2019).
  - [4] 増永良文: リレーショナルデータベース入門 [第 3 版]—データモデル・SQL・管理システム・NoSQL—, サイエンス社 (2017).
  - [5] Apache™ Batik SVG Toolkit 入手先 (<https://xmlgraphics.apache.org/batik/>) (2021 年 2 月閲覧).