

データベース 演習レポート課題

九州工業大学 情報工学部
講義担当：尾下真樹

レポートの提出方法

- レポートは、Moodle の本講義のコースの提出用のページから提出すること。
また、演習用の PostgreSQL サーバ上にデータベースを作成して、ウェブサーバの各自のディレクトリに作成したインターフェースのファイル一式を置き、動作する状態にしておくこと。
- レポートの提出締切は、Moodle の提出用ページの情報を参照すること（締切厳守）。
直前になって慌てなくても良いように、十分早くに課題に取りかかること。
- レポートのファイル形式は PDF として、ファイル名は学生番号にすること。
例えば、学生番号 212C1234 であれば、ファイル名を "212C1234.pdf" として提出する。
- 上記の提出方法を守れなかった場合、レポートが評価されなかったり（0点）、減点の対象となったりすることがあるので、注意すること。
- その他、変更点や補足事項があれば Moodle の本講義のコースで連絡をするので、適宜確認すること。

レポートの作成方法

- レポートの先頭ページには、科目名、提出日、学生番号、氏名を書くこと。
- レポートの本文は、2 ページ目から始めて、課題ごとに章立てし、指示された内容を説明すること。
長々と文章を書く必要はないので、指示された内容を適切に説明すること。
- レポートは、Moodle の本講義のコースで公開している LaTeX テンプレート（+サンプル PDF）にもとづいて、LaTeX を使用して作成することを推奨する。LaTeX を使用して作成することが難しい場合は、他のソフトウェアを使用して作成しても構わないが、レポートの形式や内容、章分け・節分けの構成は、テンプレートの指示に従うこと。フォントの大きさや余白・行間等は、大体一致していれば、厳密にテンプレートと一致していなくとも構わない。
- 本資料や LaTeX テンプレートで説明されている事項以外の詳細な書き方については、特に指定はないので、各自で判断して作成すること。

レポートの評価方法

- 本課題の成績は、提出されたレポート、及び、作成されたデータベースやインターフェースにもとづいて評価する。
- レポートの文章（正しい日本語で書かれているか）や書き方（きちんと章立てされているか、文字サイズ・行間隔等は適当か、実行結果が見やすく示されているか、など）も評価の対象とする。
- 本レポートでは、演習用のサーバ上に置かれたデータベース、および、ウェブページも評価の対象となる。これらは、レポート提出締切後も修正可能な状態にあるが、提出締切後の修正は許可しない。
ウェブページについては、最終更新日が締切日時以降となっているファイルは、締め切りを過ぎてから提出されたものとして扱う。
- カンニングは厳重な処罰や減点の対象となるので、絶対に行わないこと。例え一部分であっても、他人のレポートや外部資料を丸写ししたプログラムやレポートは、カンニングと判断される。どうしても分からないところを他人に相談することは構わないが、プログラムやレポートは全て自力で作成すること。
（講義資料のプログラムやレポートのテンプレートについては、同じ内容が提出レポートに含まれていても構わない。）

課題概要

本演習では、各自がデータベースに格納するデータを決めて、データベース+Web インターフェースを開発する。

データベースに格納する対象は、例えば、友人の住所録や所有する CD・マンガ本・ゲームソフトの管理簿、掲示板など、自分に関係するもので、データベースとして管理してあれば便利そうなものを設定する。何らかの具体的なテーマをもとにデータベースを実際に開発する、というのが本課題の主眼であるため、どのようなテーマを選択するかについては、それほど深く考えなくとも良い。

課題 1 データベーススキーマの設計

本課題では、自分が作成するデータベースのリレーションスキーマを設計する。

レポートには、まず、自分の対象とするデータについて、最初に思いついた全ての属性を書き出して、一つのリレーションスキーマとする。また、そのリレーションに格納されるデータの条件を書き出して、その条件にもとづき、リレーションの候補キー、および、属性同士の関数従属性を全て書き出す。

次に、第 1 正規形から第 5 正規形までの各正規形について順番に、スキーマが正規形を満たすかどうかを検証し、もし正規形を満たしていなければ、正規形を満たすように分解していく。正規形を満たしている場合は、正規形の条件にもとづいて、スキーマが各正規形を満たしていることを説明せよ。

レポートでは、正規形の検証とスキーマの分解の過程を説明し、最終的にどのようなスキーマが得られたかを示すこと。各スキーマを記述する際は、必ず、主キーとなる属性に下線を引くこと。また、参照整合性制約も明記すること。

正規形の検証の過程で、少なくとも一度はリレーションスキーマの分解が行われるようにすること。また、最終的なリレーションスキーマは、最低でもリレーション数（テーブル数）が 2 つ以上、属性数は全テーブル合計で 6 つ以上になるようにすること。（リレーションスキーマの規模が極端に小さいと、レポートとして成立しないため。）ただし、リレーション数（テーブル数）が多くなると、後の Web インターフェースの作成が大変になるため、分解後のリレーション数（テーブル数）の数は 3 つ以内（2 or 3 個）にすることを推奨する。

ヒント：

初期スキーマにおいて、属性値を複数持つ属性がある場合は、下記の例のように、「～1、～2、・・・」など記述しておき、正規化の最初の段階で、第 1 正規形を満たすように分解する。

従業員（従業員番号、氏名、年齢、部門番号、部門名、担当顧客番号 1、担当顧客番号 2、…、電話番号）

→ 一人の従業員に対して複数の担当顧客があるため、第 1 正規形を満たすように、従業員（従業員番号、氏名、年齢、部門番号、部門名、電話番号）と担当（従業員番号、担当顧客番号）のように分解する。

ヒント：

自分のレポート課題で使用する初期スキーマに案が思い浮かばなければ、下記の例を参考にして、適宜属性を加えたり削除したりして使用しても良い。（下記の例をそのまま使用するのはではなく、必ず何らかの変更を加えること。）

友人の住所録（名前、住所、メールアドレス、所属学部名、所属学科名、所属サークル、所属サークルの部屋番号、電話番号）

※ 所属学科名が同じ友人は、所属学部名も同じである

自分が所有している書籍のデータベース（書籍番号、書籍名、著者、出版年、出版社番号、出版社名）

※ 出版社番号が同じ書籍は、出版社名も同じである。

自分が所有している CD のデータベース（アーティスト名、CD のタイトル、CD の発売年、レコード会社、

CD の 1 曲目のタイトル、CD の 2 曲目のタイトル、CD の 3 曲目のタイトル、…)

※ アーティストが同じ CD はレコード会社も同じである。一つの CD に対して複数の曲のタイトルがあるため、第 1 正規形を満たすように分解する。

電子掲示板の投稿のデータベース (投稿番号、投稿者名、投稿者のメールアドレス、投稿者のパスワード、発言タイトル、発言の本文)

※ 投稿者名が同じ投稿は、投稿者のメールアドレスや投稿者のパスワードも同じである。

その他、映画、芸能人、車、サークル関係のデータベースなど、データベースの対象は何でも構わない。

評価基準：

主に以下の項目を満たしているかにより評価する。(各項目の詳細な判断基準や点数に関する質問は受け付けないので、各自で判断すること。以降、他の評価基準についても同じ。)

- 初期スキーマ (各属性の説明やドメインを含む) が正しく示されている。
- 初期スキーマに格納されるデータの条件が、最初に正しく説明されている。
- 条件に従って、候補キーや関数従属性が正しく示されている。
- 第 1 正規形～第 5 正規形の全ての正規形の検証が行われている。
- 正規形を満たすか、満たさないかの判定が、正しく行われている。
- 正規形を満たす場合に、その理由が正しく説明されている。
- 正規形を満たさない場合に、その理由が正しく説明されている。
- 正規形を満たさない場合に、正しく分解が行われている。
- 正規形の検証の過程で、少なくとも 1 回以上の分解が行われている。
- 全体で候補キーや関数従属性が一貫している。(正規形の検証の過程で突然、最初に列挙されていなかった候補キーや関数従属性が出て来たり、途中で候補キーや関数従属性の定義が変わったりしない。)
- 最終的に全ての正規形を満たすリレーションスキーマが設計されている。
- 最終的なリレーションスキーマに対する参照整合性制約が正しく示されている。

以下の項目は必須ではないが、以上の項目を満たした上で、さらに以下の項目も満たしていれば、より高い評価とする。

- 第 2 正規形を満たさないリレーションスキーマや、ボイス・コッド正規形を満たさないリレーションスキーマが、初期スキーマとして用いられている。
- 2 回以上の分解 (異なる正規形の判定にもとづく分解) を行うようなリレーションスキーマが、初期スキーマとして用いられている。

課題 2 データベースの作成

課題 1 で設計したスキーマにもとづき、テーブルを作成する。各テーブルには、適切な参照整合性制約を設定すること。テーブルを作成したら、実際にいくつかの初期データを入力する。全てのテーブルで合わせて、最低 20 件以上のデータ数を入力すること。

レポートには、テーブル作成や初期データの挿入に使用したコマンド・SQL と、何らかのテキストファイルを用いた場合はその内容を示すこと。

また、各テーブルに格納されているデータを確認するために、各テーブルの全てのデータを出力する SQL を実行して、その実行結果を示すこと。自分が開発したインターフェースを使って初期データ以外のデータを追加・変更・削除した場合は、それでも構わないので、レポート提出時点でテーブルに格納されている全てのデータを示すこと。

授業での説明通り、テーブル名や属性名は全てアルファベットにすること。(ただし、リレーションスキーマ

の設計の段階（課題 1）では、テーブル名や属性名は必ず日本語で記述すること。中途半端に英語やローマ字を使うと、属性の意味が分からない不明瞭になる恐れがあるため。）

データベースは、これまでの演習で作成した各自のデータベース（各自の九工大 ID と同じ名前のデータベース）を使用し、勝手に別の名前で作成したりしないこと、

注意：

レポートとして提出されたデータベースは、基本的には担当教員のみがレポート評価のためチェックするが、ウェブページの性質上、URL を入力すれば学内から誰でもアクセス可能であるため、他人に見られても問題がないようなデータベースの内容とすること。

評価基準：

主に以下の項目を満たしているかによって評価する。

- テーブルの作成が正しく行われている。
- テーブルに対して参照整合性制約が正しく設定されている。
- テーブルの作成に用いたコマンド・SQL やテキストファイルが、漏れなく正確に示されている。
- 初期データの挿入に用いたコマンド・SQL やテキストファイルが、漏れなく正確に示されている。
- テーブルに格納されている全てのデータが、漏れなく正確に示されている。
- テーブルに格納されるデータが課題 1 で示されたデータの条件に一致している。

課題 3 データベースへの問い合わせ

課題 2 で作成したデータベースに対して、そのデータベースを利用する際に使われる問い合わせの具体例を考えて、その問い合わせに対する SQL と実行結果を最低 3 つ示す。問い合わせの具体例は何でも構わないが、出力結果が想定した問い合わせと一致しているか（正しい結果が出力されているか）を確認すること。

複数の問い合わせの例としては、同じような問い合わせ（条件の数値だけを変えたような問い合わせ）ではなく、条件に使用する属性が異なる問い合わせや、結合を用いた問い合わせを含めること。なるべく GROUP BY (+HAVING) や入れ子型問い合わせ（相関あり・なし）などの、授業で学習したやや複雑な構文を用いた SQL を含めることが望ましい。無理に多くの（4 つ以上）の問い合わせを示す必要はないので、3 つの問い合わせに上記のような例を含めるようにする。

評価基準：

主に以下の項目を満たしているかによって評価する。

- 各問い合わせに対する正しい SQL が記述されている。
- 各問い合わせの SQL の実行結果が正しく示されている。
- 複数の問い合わせの中に、条件に使用する属性が異なる問い合わせや、結合を用いた問い合わせが含まれている。

以下の項目は必須ではないが、以上の項目を満たした上で、さらに以下の項目も満たしていれば、より高い評価とする。

- 複数の問い合わせの中に、GROUP BY (+HAVING) や入れ子型問い合わせ（相関あり・なし）などの、授業で学習したやや複雑な構文を用いた SQL が含まれている。

課題 4 Web インターフェースの作成

課題 2 で作成したデータベースに対して、Web 経由で主要な操作を行うためのインターフェースを開発する。まず、最低限の機能として、データベースのデータの一覧表示を行う機能や、少なくとも 1 つの主要なテーブルに対してデータ追加・削除・更新の操作のための機能を作成する。全ての操作について、本授業で学習したフォームの動的生成の方法を用いて、使いやすいインターフェースを作成すること。ただし、更新の必要性が少ないテーブルについては、これらの操作のための機能は作成しなくても良いものとする。(例えば、授業の演習の例では、**employee** テーブルに対しては追加・削除・更新のインターフェースを作成しているが、データ数が少なく更新頻度も少ない **department** テーブルに対しては、これらのインターフェースを作成していない。)

また、データベースからデータ検索を行うためのインターフェースを追加する。検索機能はなるべく実用的であると考えられるものを作成すること。例えば、本授業の演習で作成した従業員のデータベースであれば、従業員番号での検索だけでなく、部門名や年齢を条件とする検索、またそれらの条件を組み合わせた検索など。一般に、検索については、ひとつの条件だけでの検索ではなく、複数の条件を組み合わせた (AND や OR) 検索や、さらにどの条件を使用するかを利用者が自由に選択できるような任意の複数の条件の組み合わせでの検索が行えるようになっており、非常に有用である。このような検索機能まで作成されていれば、より高く評価する。

レポートには、必ず、自分の作成したデータベースのトップページの URL を記述し、本課題で作成した全ての **php**, **html** ファイルの構成・階層構造と、それぞれのファイルの役割を簡単に説明すること。URL は、必ず、正しいものを記述すること。URL が書かれていない場合、あるいは、アクセス不能な URL が書かれていた場合は、レポートが評価できないため、0 点となる。

次に、各 **php**, **html** ファイルについて、プログラムの全体を引用した上で、前のページから渡される入力や、そのページで使用する **SQL** とその作成方法、結果の表示方法等の重要な点について説明する。単純にプログラム全体をレポートに引用するだけでは不十分なので、どのような処理を行っているのかがきちんと分かるように、レポートの文章で説明すること。また、各機能について、実行例が分かるスクリーンショットを含めること。

注意：

講義で使用したサンプルプログラムをもとにして作成するのは構わないが、ソースファイルが見にくくならないように、たとえ動作に支障はなくても、レポートに関係のない不要な処理やコメントは削除すること。また、提出する **html**, **php** ファイルの中には、不要なファイル (バックアップファイルなどの関係ないファイル) が含まれないようにすること。

不要なファイル・処理が含まれている場合は、減点の対象とする。

評価基準：

作成されたインターフェースについて、主に以下の項目を満たしているかにより評価する。

- メニューページが正しく作成されている。
- 一覧表示 (全テーブルを組み合わせた一覧表示) が正しく作成されている。
- 主要なテーブルに対して、追加インターフェースが正しく作成されている。
- 追加インターフェースが、選択肢の動的生成に対応している。
- 主要なテーブルに対して、削除インターフェースが正しく作成されている。
- 削除インターフェースが、選択肢の動的生成に対応している。
- 主要なテーブルに対して、更新インターフェースが正しく作成されている。
- 更新インターフェースが、選択肢の動的生成に対応している。
- 更新インターフェースで、更新前の属性値が初期値としてフォームに表示されるようになっている。
- 検索インターフェースが作成されている。

- 検索インターフェースが、選択肢の動的生成に対応している。
- 実用的と考えられる検索インターフェースが作成されている。

以下の項目は必須ではないが、以上の項目を満たした上で、さらに以下の項目も満たしていれば、より高い評価とする。

- 主要な1つのテーブルだけでなく、複数の（全ての）テーブルに対する追加・削除・更新インターフェースが作成されている。
- 複数の条件での組み合わせた検索インターフェースが作成されている。
- 任意の複数の条件の組み合わせでの検索インターフェースが作成されている。

また、レポートの書き方について、主に以下の項目を満たしているかにより評価する。

- テンプレートの指定通りの形式や章・節・小節構成のレポートが作成されている。
- レポートの内容が読む人に伝わるような、正しい日本語や形式で書かれている。（誤字脱字や文法がおかしい文章が含まれていない。章・節・小節分けや、段落分け、参照が適切に行われている。）
- 作成したインターフェースのトップページの URL が正しく示されている。
- 作成した全てのページの構成・階層構造や、それぞれのページの役割が正しく説明されている。
- インターフェースの全機能と、各機能を構成する全てのページが、指示通りの節・小節構成で説明されている。
- 作成したインターフェースの使用例が、スクリーンショットも含めて、説明されている。また、スクリーンショット等の図が、適切な大きさを示されている。
- 各ページの説明で、ソースファイルが正しく示されている。
- ソースファイルの中に、授業のサンプルプログラムから引き継がれた不要な処理やコメントが残っていない。また、「従業員」などの、授業のサンプルプログラムで使った文字列が残っていない。
- 各ページの説明で、前のページから渡される入力や、そのページで使用する SQL とその作成方法、結果の表示方法等の重要な点が正しく説明されている。
- レポートで説明されている内容と、作成されたインターフェースの動作が一致している。
- 演習方法・レポート提出方法の指示が正しく守られている。（データベース名、提出方法、提出ファイル形式、提出ファイル名、など。）

付録 1. LaTeX の使用方法

ここでは、レポートを作成する上で必要となる LaTeX の最低限の使用法のみを説明する。より詳しい使用方法については、テンプレートの使用例を参照したり、1 年生の「プログラミング」の授業で学習した内容を復習したり、各自で書籍やインターネットで調べたりすること。

LaTeX コンパイル方法

BYOD 端末の Ubuntu 仮想環境には、LaTeX のソフトウェアがインストールされており、リテラシーの授業で学習した通り、下記の手順で LaTeX ソースファイルをコンパイルして PDF を作成できる。

1. ソースファイル (.tex) をコンパイルして dvi ファイルを生成する。

```
$ platex ファイル名.tex
```

2. dvi ファイルを PDF に変換する。

```
$ dvipdfmx ファイル名.dvi
```

レポートのテンプレートでは、プログラムリストの表示に `jlisting` パッケージを使用しているため、コンパイルを行うためには、テンプレート一式に含まれている `jlisting.sty` が、ソースファイルと同じディレクトリに置かれている必要がある。

また、ソースファイルから読み込んでいるプログラムリストや図も、(読み込み時にファイルが置かれているパスを設定していない場合は) ソースファイルと同じディレクトリに置かれている必要がある。

本文の記述方法

基本的には、ソースファイルに記述したテキストが、そのまま本文として出力される。

ただし、ソースファイル中のインデントや改行は無視されるので、改行したい場合は、`¥¥` を付けるか、一行空行を空ける。

アンダーバー (`_`) やダブルクォート (`"`) や中括弧 (`{}`) などの特別な記号は、そのまま記述しても表示されない (コンパイルエラーが出る) ので、各記号の前に `¥` を付ける。

環境によっては、`¥` 記号は、`\` (バックスラッシュ) として表示されるので、注意する。

章・節・小節

`¥chapter{見出し}`, `¥section{見出し}`, `¥subsection{見出し}` を使って、章・節・小節を分けることができる。

`¥newpage` コマンドを使うと、改ページを行える。各章の前には、自動的に改ページが行われる。

箇条書き

`¥begin{itemize}` ~ `¥end{itemize}` の内側に、各項目を `¥item` ~ として記述することで、箇条書きにすることができる。箇条書きの内側に、さらに別の箇条書きを入れることで、階層化することもできる。

`¥begin{enumerate}` ~ `¥end{enumerate}` を使うと、番号付きの箇条書きになる。

`¥begin{description}` ~ `¥end{description}` を使い、各項目を `¥item [見出し]` ~ として記述することで、各見出しを説明するような、見出し付き箇条書きになる。

図の挿入

`¥begin{figure}` ~ `¥end{figure}` や `¥includegraphics[オプション]{画像ファイル名}` を使用することで、画像を挿入できる。画像ファイルの形式は、EPS 形式にする。(PNG 形式などの他の形式の画像も読み込むことができるが、少し難しいため、EPS 形式を推奨する。) また、オプションで図の大きさを指定できる。図の大きさは、絶対的な大きさ (cm など) や、ページの表示幅に対する相対的な比率 (ページの表示幅の 0.5 倍など) で指定することができる。

`¥caption{キャプション}` を使うことで、図のキャプションを設定できる。図が何を表しているのかが分かるように、適切なキャプションを設定する。

`\label{ラベル文字列}` を使うことで、図を本文から参照できるようになる。本文中に `\ref{ラベル文字列}` を記述すると、コンパイル時にその図の図番号に置き換えて出力される。全体で図を追加・削除して図番号が変わったときにも、正しい図番号で参照することができるため、図を参照するときにはラベルを使う方が良い。ただし、参照する全ての図に対して、個別のラベル文字列を設定する必要がある。

`\begin{center} ~ \end{center}` を使うことで図のセンタリングを行える。

`\fbox{ ~ }` を使うことで図に枠を付けて表示することができる。

LaTeX では、図は自動的に配置されるため、図の細かい配置を気にする必要はない。図が後方に配置されて気になる場合は、`\clearpage` コマンドを使用すると、コマンドよりも上の図を全て出力した上で改ページを行うことができる。

プログラムリストの挿入

`listings` の機能を使うことで、プログラムリストなどのテキストファイルを挿入できる。本文中にプログラムリストを直接記述する書き方では、インデントや改行が無視されたり、記号が正しく表示されなかったりして、プログラムリストを示すことが難しいが、`listings` を使うことで、プログラムリストをそのまま挿入することができる。

`\begin{lstlisting}[caption=キャプション] ~ \end{lstlisting}` の形式で、プログラムリストを記述する。もしくは、`\lstinputlisting[caption=キャプション]{ファイル名}` の形式で、同じディレクトリにある別のファイルを挿入することもできる。ファイル全体を挿入する場合は、こちらの方法を使う方が良い。

整形済みテキストの挿入

SQL の実行結果などの整形済みテキストを本文中に含めるときには、`\begin{verbatim} ~ \end{verbatim}` を使用する。ただし、`verbatim` 環境を使うと、文字数が多い行はページの横幅を超えてしまうため、その場合、手作業で適当に改行を挿入してページ内に収まるように調整する。

参照

本文から、図やプログラムリストや章・節・小節を参照するときには、ラベルを使用することで、自動的に正しい番号を挿入することができる。

図の場合は `\begin{figure} ~ \end{figure}` の内側、章・節・小節の場合は見出しの直後に、`\label{ラベル名}` を追加することで、本文中で `\ref{ラベル名}` とすると、そのラベルに対応する番号を自動的に挿入することができる。プログラムリストの場合は、`\lstinputlisting[caption=キャプション,label=ラベル名]{ファイル名}` のようにして、ラベルを設定する。

ラベル名には、参照する各要素を区別できるように、各要素に異なる文字列を設定する必要がある。

コンパイル時のエラーへの対処方法

LaTeX のコンパイル時にエラーが表示され、コンパイルを行うことができなかった場合は、エラーメッセージの説明を参照して、ソースファイルの問題を修正する。

よくある間違いと対処方法は、以下の通り。

- ・コンパイルに必要なファイルを読み込むことができない。
上記の「LaTeX コンパイル方法」の説明の通り、テンプレート一式に含まれている全てのファイルをソースファイルと同じディレクトリに置く必要がある。また、ソースファイルから読み込む図やソースファイルなどを追加した場合は、ファイル名が正しいことを確認する。
- ・本文中に `_` などの記号が含まれている。
`_` のような特殊文字は、LaTeX のコマンドの一部として扱われるため、そのまま本文中に記述することはできない。ソースファイル名を記述する場合など、本文中に `_` を使用したい場合は、前に `\` 記号を付けて、`_` のように記述する。
- ・図やソースファイルを参照しているときに、参照番号が正しく更新されない。
参照番号が正しく反映されるようにするためには、複数回のコンパイルを行う必要がある。
また、複数の図やソースファイルに同じラベル名を設定していると、正しい参照番号を表示することができないため、全ての図やソースファイルには、独立したラベル名を設定する必要がある。

付録 2.スクリーンショット画像（EPS 形式）の作成方法

LaTeX を使ってレポートを作成する上で必要となる、スクリーンショット画像（EPS 形式）の作成方法を説明する。ここでは、BYOD 端末の Ubuntu 仮想環境にインストールされているフリーの画像編集ソフトウェア GIMP を使った方法を紹介する。なお、使用する環境やバージョンによって、細かい表示や操作方法は異なる可能性があるので、適宜、読み替えて操作すること。

GIMP 以外のソフトウェアや別の方法を用いて画像を作成しても構わない。

1. **PrtScn** キーを押して、スクリーンショットを取得し、クリップボードに保存する
PrtScn キーのみを押すと画面全体、**Ctrl+Alt+PrtScn** キーを押すと現在アクティブなウィンドウのスクリーンショットを取得できる。
2. GIMP を起動する。
BYOD 端末の Ubuntu 仮想環境では、ターミナルから **gimp** のコマンドを実行して起動するか、「コンピュータを検索」から「**gimp**」で検索して表示されるプログラムを選択して起動する。
3. GIMP 上で新規作成した画像に、スクリーンショットを貼り付ける。
メニューから、「編集」→「貼り付け」を選択。
4. 画像中の保存したい範囲を切り出す。
ツールボックスから「矩形選択」を選択し、画像中の切り出す範囲を選択。その後、メニューから、「画像」→「選択範囲で切り抜き」を選択。
5. EPS 形式で画像を保存する。
メニューから、「ファイル」→「名前を付けて画像をエクスポート」を選択。その後、ファイルの形式の選択 から、EPS 形式を選択し、適当なディレクトリに適当なファイル名でエクスポートする。