

データベースS 演習レポート課題

九州工業大学 情報工学部 システム創成情報工学科
講義担当：尾下真樹

レポートの提出方法

- レポートの締め切りは、8月上旬（期末試験終了後）を予定。（詳細は後日連絡）直前になって慌てなくても良いように、課題には、十分早く取りかかること。
- レポートは、Moodleの本講義のコースから提出すること。電子メールや紙での提出は、一切認めない。また、学科のPostgreSQLサーバ上にデータベースを作成して、自分のディレクトリに操作のウェブページを置いた状態にしておくこと。
- レポートのファイル形式は、MS Word または PDF とすること。
- レポートのファイル名は学生番号にすること。例えば、学生番号 13236001 であれば、“13236001.docx”、“13236001.doc”、“13236001.pdf” などのファイル名で提出する。
- レポートには、自分の作成した全ての html, php ファイルを ZIP 形式で圧縮して添付すること。こちら、ファイル名は学生番号とする（例えば、学生番号が 13236001 であれば、“13236001.zip”）。なお、提出する ZIP ファイルには、レポートに必要なファイル以外（*.php~ などのテキストエディタが作成する自動バックアップファイルや、employee_*.php などの講義の演習で使ったファイルなど）は含めないこと。
- レポートのファイルサイズが大きくなりすぎないようにすること。もしサイズが 10MB を超える場合は、あらかじめ相談すること。
- 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により評価する。
- 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、学科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、レポート提出後も各自が修正可能な状態にあるが、修正は許可しない。ウェブページについては、最終更新日が締め切り時刻以降に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、他人のレポートや資料を丸写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。（授業で資料として配布したプログラムについては、必要箇所をコピーして利用するのは構わない。）

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、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、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、最終的なリレーションスキーマは、最低でもテーブル数が 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（あまりテーブルの規模が小さいとレポートとして面白くないため）。

ヒント：

属性値を複数持つ場合は、下記の例のように、「～1、～2、・・・」など記述しておき、第 1 正規形を満たすように分解する（分るように書いてあれば、どのような書き方でも良い）。

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

ヒント：

どうしてもスキーマの案が思い浮かばない人のためのヒント。

下記のスキーマ案のうちの一つを参考にして、適宜属性を加えたり削除したりして使用しても良い（必ず何らかの変更を行い、そのままは使用しないこと）。

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）

自分の所有している CD のデータベース（アーティスト名、アーティストのホームページ、CD のタイトル、CD の曲数、CD の発売年、レコード会社、レコード会社のホームページ、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、・・・）

掲示板（発言番号、投稿者、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）

その他、マンガ本、映画、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良い。

評価基準：

基本的には、きちんと正規形を満たすスキーマが設計されており、スキーマが正規形を満たすことの説明や分解の過程がきちんとレポートで説明されていれば、ほぼ満点を与える。

逆に、最終的なスキーマが正規形を満たしていたとしても、途中の分解の理由が間違っていたり、正規形を満たすかどうかの説明がきちんと書かれていなかったりすれば、大きく減点する。

課題 2 データベースの作成

前課題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、データ数 20 件以上を入力すること。データの入力、次の課題で作成するデータ追加インターフェースを使って行っても構わない。

レポートには、テーブルの作成や制約の設定に使用した SQL 文と、データベースに格納した初期データの一覧を載せること。

注意：

インストールされている PostgreSQL では、テーブル名や属性名には日本語も使えることになっているが、これらに日本語を使用すると、面倒な問題が発生する可能性があるため、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、課題 1 では、テーブル名や属性名は必ず日本語で書くこと（中途半端に英語やローマ字を使うと、何の属性を意味しているのか分からない恐れがあるため）。

注意：

レポートとして提出されたデータベースは、基本的には尾下だけ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、少なくとも学科内のコンピュータからはアクセス可能であるため、他人に見られても問題ないようなデータベースの内容とすること。

評価基準：

前課題で設計したスキーマに応じて、正しくテーブルが作成されており、テスト用のデータがきちんと追加されていれば、ほぼ満点を与える。

データベースのテーマやデータの内容についてはそれほど問わないが、なるべくオリジナリティがあり、具体的なもの（単にレポートのためのデータベースではなく、実際に使ってみたら役に立ちそうなデータベースだと感じられるようなもの）であれば評価にプラスする。

課題 3 Web インターフェースの作成

前課題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、主要なテーブルのデータ表示・データ追加を行うページを作成すること。ただし、あまり後からデータを追加する必要のないテーブルについては、データ追加のインターフェースは作成しなくても良いものとする。例えば、演習資料の例では、データ数が少なく、変更もたまにしか起こらない **department** テーブルには、追加・修正のインターフェースを作成していない。

次に、検索の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だと思われるものを作成する。例えば、CD のデータベースであれば、アーティスト名や、リリース期間（いつからいつまでの間）で検索など。また、掲示板の例であれば、最新??件の表示や、投稿者??のこれまでの発言を全て表示など。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）検索ができるようになっていて、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、評価にプラスする。

最後に、主要なテーブルのデータ削除・修正のためのインターフェースも作成せよ。

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 **html**, **php** ファイルの構成と、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 点とする。

次に、各 **php** プログラムについて、前のページからどのような引数を受け取るのか、内部でどのような **PHP** プログラムや **SQL** を実行しているかなど、主要な処理を説明すること。

PHP プログラムの全体を引用したり、**PHP** プログラムを一行一行詳しく説明したりする必要はないが、プログラム中の主要な部分を引用しながら、プログラムでどのような手順・内容の処理を行なっているのかが分かるように説明すること。

単に、プログラムの一部、あるいは全体をレポートに貼り付けるだけでは、不十分。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。

なお、レポートには、作成した全ての **html**, **php** ファイルを圧縮した **ZIP** ファイルを添付して提出すること。

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とも、レポートに関係のない不要な処理は削っておくこと。

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：

評価は、**html**, **php** ファイルに間違いなどがいないかを確認し、実際に作成されたウェブインターフェースを使ってみて行う。

基本的には、正しく動作するウェブインターフェースが一通り（主要なテーブルに対する表示・追加・検索・削除・修正）実現できており、各 **html**, **php** ファイルの内容がレポートできちんと説明されていれば、ほぼ満点を与える。

インターフェースが作成されていても、その内容がレポートできちんと説明されていなければ、かなり低い点数とするので、注意すること。レポートの評価方法で書いた通り、レポートの文章はもちろん、レポートの書き方も評価の対象とする。きちんと章立て・段落分けをしたり、余白・行間を適切に取ったり、といった基本的なレポートの書き方にも、十分注意すること。また、例えば、見出しは文字の大きさを変える、スキーマや **SQL** などを引用する場合は枠で囲んで本文との区別をつけやすくする、枠の大きさやインデントは全体で統一する、などの読みやすくする工夫も必要である。

講義で学習したように、追加や検索のフォームは、選択肢や初期値の動的生成に対応していると便利である。作成されたページが、これらの機能に対応していれば高得点、逆に、固定の **html** ファイルであれば低い得点を与える。さらに、作成された検索インターフェースなどが実用的だと感じられるものになっていれば、さらに高い評価を与える。

付録 1. ZIP ファイルの作成方法

ZIP ファイルの作成方法が分からない人は、CL 端末には **Windows** 用のフリーソフト **Lhaplus** がインストールされているので、以下の説明を使って **ZIP** ファイルを作成すること。別の作成方法を知っている人は、以下の方法は使用せずに、自分の好きなツールを使って作成して構わない。

1. 適当な場所に、「自分の学生番号」の名前でフォルダを作成する。
2. 作成したフォルダの下に、提出するファイル一式をコピーする。
3. 作成したフォルダを、デスクトップの **Lhaplus** にドラッグ&ドロップする。
4. デスクトップに、「自分の学生番号.zip」のファイルが作成されるので、このファイルを提出する。
5. 提出する前に、自分の作成した **ZIP** ファイルを **Lhaplus** にドラッグ&ドロップして、ファイルが問題なく展開でき、内容が正しいことを確認した上で、提出すること。

付録 2. よくある間違い

基本的には、以上の説明に従ってレポートを作成すれば問題ないが、以下に、過去のレポートでよくある間違いを挙げておくので、特にこのような間違いをしないように十分注意すること。(これら以外の間違いにも、注意すること。)

課題 1 スキーマの設計

- 最初に候補キーや関数従属性の定義がきちんと書かれていない。
何度も講義で説明した通り、候補キーや関数従属性をきちんと定義されていなければ、スキーマの正規形の検証はできないため、候補キー属性や関数従属性がきちんと書かれていないと話にならない。
- 候補キー属性や関数従属性の定義がおかしい。
正規化自体は間違っていないものの、例えば、候補キー属性の定義と関数従属性の定義が矛盾する、関数従属性の定義が常識的に考えてありえない、など。
- 正規化の間違い。
分解の理由の説明の間違い、分解の必要がない(正規形を満たしている)のに分解している、すべての正規形を満たしているかが検証されていない、など。

課題 2 データベースの作成

- 課題 1 で設計したテーブルと、課題 2 で作成されたデータベースのテーブルが食い違っている。
- テーブルに適切な参照整合性制約が設定されていない。

課題 3 Web インターフェースの作成

- データの追加・検索・修正・削除のインターフェースで、フォームの動的生成(選択肢の動的生成)に対応していない。
例えば、従業員の部門を部門番号で入れなければならなかったり、あらかじめ固定の選択肢が HTML に直接書かれている(部門のデータを追加しても選択肢が変化しない)、など。
選択肢を動的に生成するのは、入力のしやすさの点では非常に重要であるため、これらができていないと、評価は低くなる。
また、選択肢の動的生成を用いていないと、例えば存在しない部門番号などが入力されてしまったときに、参照整合性制約のため挿入時にエラーが生じたり、データベースに不整合が生じてしまったりする可能性があるため、このようなことが起こらないようにするためにも、選択肢から選択できるようなインターフェースにすることが必要。
- 検索インターフェースとして、ごく簡単なものしか作成されていない。
例えば、従業員番号を入れると、その従業員の情報を表示するといった、単純なインターフェースのみのもの。単にある従業員の情報を見たいだけであれば、一覧表示を表示すれば十分なので、このような検索は全く実用的ではない。
せっかく複数のテーブル・属性を作成しているので、例えば、所属部門で検索したり、年齢で検索したり、検索結果も年齢順に並べて表示したり、といったインターフェースが作成されていると良い。特に、課題の説明に書かれている通り、属性値の範囲での検索や、複数の条件を組み合わせた AND 検索など、検索が工夫されていれば、高い評価となる。
- サンプルプログラムの不要な内容が残ったままになっている。
たとえば、ページのタイトル名やリンクが、サンプルの従業員のままになっている、など。

レポートの説明や書き方

- 全体的に説明不足、説明の文章が分かりにくい、など。
レポートでは、単に作成したシステムが動けば良いというものではなく、どのような内容をどのようにして作成したのか、他人が分かるように、きちんと文章で説明できることが非常に重要である。
ときどき、データベースやインターフェースは非常に良くできているにも関わらず、レポートのできが非常に悪く、低い点数になる人もいるので、システムとレポートの両方をきちんと作成すること。
文章についても、きちんと他人が読んで意味が理解できる文章を書くことが必要である。かなり説明が支離滅裂な人もいるので、レポートを書いた後は、一度、他人に読んでもらったり、全く内容を知らない別人になったつもりで自分で読み返してみたりして、きちんと意味の通じる文章になっているかどうか確認した方がよい。また、誤字脱字や変換ミスにも注意すること。
- 処理の説明が不十分。
PHP ファイルで使用している引数や SQL を単純に列挙するだけでは、不十分。それだけでは、どのようにしてインターフェースを実現したのか、他人がレポートを読んでも全く分からない。
プログラムを1行1行説明する必要はないが、どのような処理のために SQL を使用しているのか、SQL を使って取得した結果をどのように表示しているのか、フォームへの入力情報をどのように受け渡ししているのか、など、全体処理の流れが分かるような最低限の説明は必要である。
- ファイル構成の説明が不十分。
一般に、データベースのインターフェースを実現するためには、複数の html, php ファイルが必要となる。また、検索機能など、複数のファイルの組み合わせによって実現される機能もあり、どのファイルからどのファイルが呼び出されるかなど、ファイル間の関連（全体の階層構造）も複雑となる。
課題で指示されている通り、自分の作成した内容をきちんと説明するために、このようなファイル構成を図などを使って説明することが必要。
- レポートの体裁や見易さ。
レポートの文章だけでなく、レポートを見やすくするための体裁も重要である。
行間を適切にとる、段落をきちんと分ける、適切な余白をとるなど、基本的な体裁が整っていないと、レポートが読みにくく、減点の対象となる。
また、例えば、見出しは文字の大きさを変える、プログラムやスキーマや SQL などを引用するときは枠で囲んだりインデントをつけたりして分かりやすくする、枠の大きさやインデントの幅はレポート全体で統一する、などのレポートを読みやすくする工夫も必要である。
- 作成したインターフェースのトップページの URL が書かれていない。