CSC 380/530 — Advanced Database
Take-Home Midterm Exam (document version 1.0)
SQL and PL/SQL

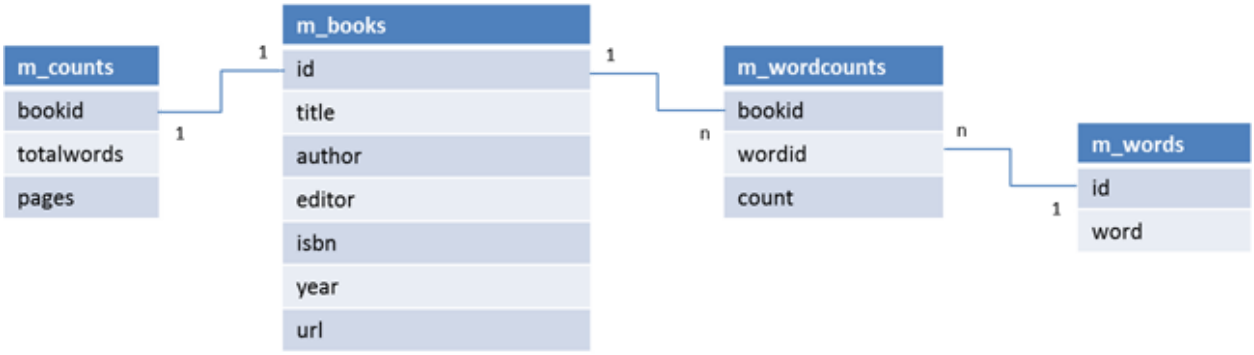
- The take-home midterm exam is due by 11:59:59 PM on Thursday, November 5, 2015 and must be submitted electronically.
- The take-home midterm exam will count as 20% of your final course grade.
- The take-home midterm exam is to be completed **individually**. Do not share your work with anyone else.
- **No late submissions will be accepted.**

Getting Started

Download the `midterm-sql.zip` file from the course website, then execute the scripts within your Oracle environment by executing them in this order: `midterm.sql`, `midterm-data.sql`, `insert-words.sql`, `insert-counts.sql`.

These scripts create and populate four tables (shown in the database diagram below), i.e., `m_books`, `m_counts`, `m_wordcounts`, and `m_words`.

The `m_wordcounts` table represents a many-to-many relationship between `m_books` and `m_words`, meaning that a given word in the `m_words` table may appear one or more times (as indicated by the `m_wordcounts.count` field) in one or more books.



Browse through these tables to understand them, emailing me with any questions.

Note that the data in these table was obtained programmatically (via Python) from publicly available texts of classic works (all from `http://bartleby.com/`).

General SQL Queries

Write SQL queries to answer the questions below. Note that your SQL code should be as general-purpose as possible (i.e., assume that the sample tables could have many more rows than what has been provided).

1. How many unique words are there across all books? Note that to be unique, a word must only appear once.
2. How many unique words are there per book? Design your query to display the results ordered alphabetically by book title.
3. What are the top eight most frequently occurring words across all books? Design your query to display the results in order starting from the most frequently occurring word.
4. How many unique words are there per author? How many unique words are there per editor? Design your query to combine the above results and display the results ordered alphabetically by author/editor name.
5. What is the percentage of word counts to total unique words per book? For example, if a word occurs 47 times in a work that contains 4700 unique words, the percentage for this word is 1%.

SQL Queries to Handle Stopwords

A "stopword" is a word that search engines often ignore. Typical stopwords include: the; of; and; to; or; not; that; there; and so on. From question 3 above, you probably found a lot of these stopwords. In this section, we will work further with stopwords to improve the query results of question 3.

Write SQL creation scripts and queries as specified below. As above, your SQL code should be as general-purpose as possible.

6. Write a creation script for a new table called `m_stopwords` that has a unique `id` field and a foreign key `wordid` field that references the `m_words` table.
7. Write a single SQL query that inserts rows into the (assumed-to-be-empty) `m_stopwords` table by selecting the top 20 most frequently occurring words across all books.
8. Taking stopwords from the `m_stopwords` table into account, write an SQL query that selects the top 20 most frequently occurring words across all books, i.e., ignore stopwords. Be sure you do not delete or change any of the existing tables or their data (i.e, do not simply delete stopwords from `m_wordcounts`).

Oracle PL/SQL Functions and Procedures

Write PL/SQL functions and procedures as specified below.

9. Create a PL/SQL function called `generatePassword()` that generates (and returns) a random password based on the words data. Generated passwords have the following general format:

   `<word-1><integer><word-2><integer>...<integer><word-n>`

   This function must take two integer inputs.

   The first input is `n`, which specifies the number of words to use in the generated password. More specifically, your function must randomly select `n` words that are unique across all books in the database (ignoring stopwords). Also note that a generated password must contain exactly `n` unique words (i.e., no duplicate words in a generated password).

   The second input is `m`, which specifies the number of digits the randomly generated integers should be between each word.

   As an example, if `n` is `4` and `m` is `2`, possible randomly generated passwords are:

   ```
   requiem97heather59switzerland83baboon
   distinguishing21sharpens18zephyrs81caves
   influenced12palms19newark45hosting
   ```

10. Create a PL/SQL procedure called `displayFrequencyReport()` that displays a summary report for either a given book or all data within the database. The input to this procedure is `bookid`, a key to the `m_books` table. If the `bookid` input is `0`, then data for all books should be displayed.

    In your summary report, display the word count for each word. Note that words with the same word count should be grouped together in your report. Therefore, also display the number of words in each group (i.e., the number of words with the given word count). Since the number of words in a group might be very large, display at most four of the words in each group.

    Order your output by word count, from highest to lowest.

    Your procedure must output the following (using the format shown here):

```
Data analysis report as of <current-date>

Unique words: 5717

 Word Count    # Words in Group    Words (at most four)
------------  ------------------  ---------------------------------
       1822                   1    the
       1258                   1    of
       1102                   1    and
        607                   1    to
        572                   1    his
        548                   1    in
        463                   1    he
        360                   1    was
        282                   1    with
        266                   1    that
        215                   1    it
        184                   2    is, by
        181                   1    at
        174                   1    as
        168                   1    on
        160                   3    but, which, for
        152                   1    had

        ...                 ...    ...

          3                 405    autumnal, leaves, gleam, mended
          2                 906    pound, limb, done, homeward
          1                3281    groan, beautiful, resist, extra
------------  ------------------  ---------------------------------
```

    Note that the above data is based on a different dataset (so use it only as an example).

Submission Instructions

To submit your work, create a single ZIP file (or compressed folder) containing all of your source files, including the code used to generate the SQL `insert` statements for the `zipcodes` table. Use your Saint Rose ID (e.g., `goldschmidtd168`) as the name of the ZIP file (i.e., `goldschmidtd168.zip`).

Though entirely optional, you can include a simple `README.txt` file with notes or instructions.

Email your ZIP file to `goldschmidt@gmail.com` (with a subject of "`CSC 380/530 Midterm Exam`").