CSCI 3700 - Database Management Systems
Data-intensive Application Development
Term Project

Jack Edwards and Oliver Chen

Contents

1 Data-intensive Application Selection

A data-intensive application is characterized by: lots of data; relatively little processing; large number of insertions, deletions, updates, and queries. Data is viewed as a corporate resource and is often used for tasks varying from marketing campaigns, new product strategies, inventory management and distribution logistics, to improving customer loyalty.

Ideally, the application should have: well-understood or well-designed business processes; sufficient documentation about the business processes; tasks are known; application boundary is well-demarcated; allows developing one organization-wide database schema from multiple department-wide database views.

You should either possess sufficient domain knowledge of the application area, or have access to a domain expert. Otherwise, it is extremely difficult to develop a successful data-intensive application.

Describe the data-intensive application that you have selected. What are its characteristics? What makes it data-intensive? Who is the sponsor of this project? Who are the end-users? How will they benefit from this application?

Answer: We have selected the Project Gutenberg as our topic, it has over 57,000 online eBooks. The project would be create a database that hold all the books, identify the entities, attributes and relationships. The database would act as a library allow users to do searches base on their need. Also the database system can be implement with information retrieval system.

2 Identification and Documentation of Use-cases

Identify various *classes of users* (aka *actors*) for the data-intensive application. Note that users can be human as well other systems. In some applications, *time* is a user. For example, end-of-day, end-of-week, end-of-month, end-of-quarter, and end-of-fiscal-year are all time-triggered events. Your application needs to respond to these time-triggered events.

Use-cases describe interactions between the users and the system. Some interactions can be normal (no error conditions), other interactions may entail additional processing (e.g., preferred customers receive additional services), and yet other interactions require error recovery due to various conditions such as erroneous input or device malfunctioning. Each path through a use-case is called a *scenario*. In other words, a use-case is a set of related scenarios.

Conceptually, a use-case represents a *unit of work* from an end-user perspective. A use-case involves executing a set of tasks in certain sequence.

How may user classes do you have? How many actors (including human, application, and abstract ones like time)? Name use-cases. Document them using the LaTeX template.

Answer: We will have two user classes End users and administrators. Administrators can add documents to the database as well as performing any actions available to end users. End users can stream all documents from the database, they can search documents by language, title, author, and year. We plan to provide java and python libraries to interface with the database.

3 Use-case Diagram

Use-case diagram is a pictorial representation of interactions between the application users and use-cases. It also shows relationships between use-cases such as one use-case being embedded in another use-case, or one use-case extending the functionality of another use-case.

Answer:

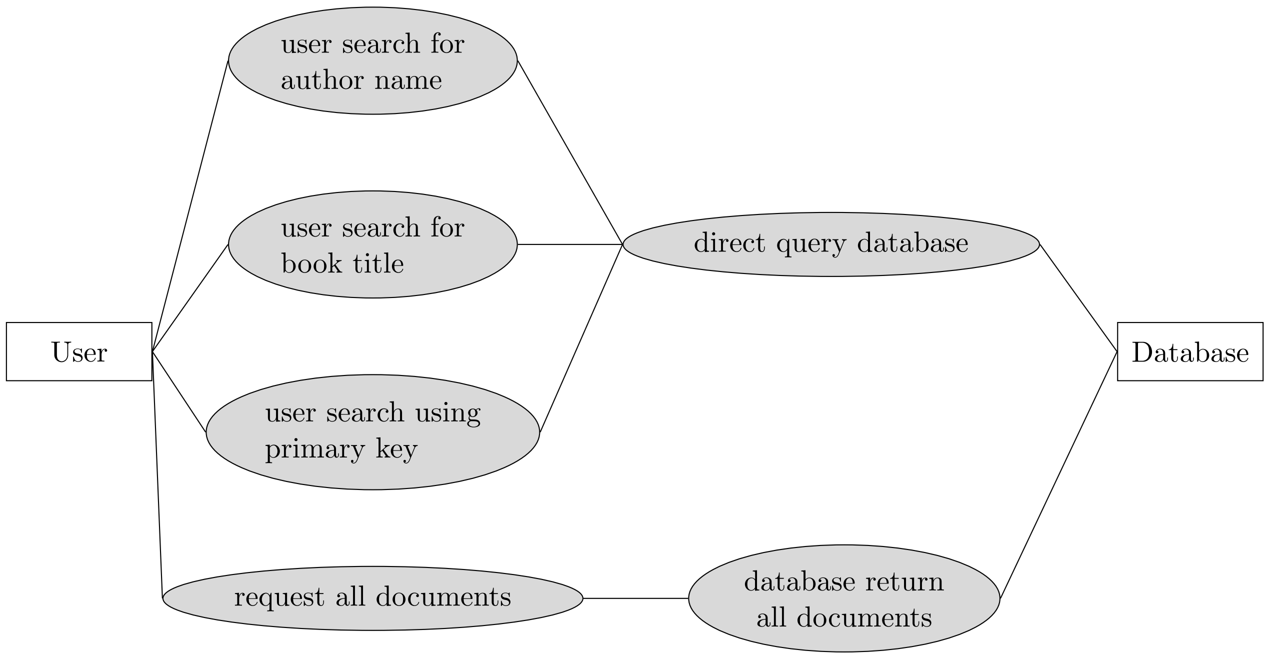

user search for author name
user search for book title
user search using primary key
request all documents
direct query database
database return all documents
User
Database

Figure 1: Use-case digram

4 Identification and Documentation of Data Tasks in the Application

Each use-case scenario requires executing a set of tasks. For each task identify and document inputs needed, and outputs generated. Also, specify possible error conditions that might occur as inputs are transformed into outputs.

Answer: User could query author name, book title, primary key of certain book, since these could be directly query from the database. We can do a direct query to gather these information. Also if user need to request all documents from database, the database will return all documents in json format with all attributes as the fields. It would be more complicated if user try to search by keywords. This process require information retrieval system to process text and analyze documents, the function will be add if the implementation is needed.
The possible errors these processes might occurs are, user type in wrong information, user misspell words, ir system could not find the information user need.

5 Identification and Documentation of Transactions

A *transaction* is a unit of work both from a database end-user perspective as well as from the database system perspective. A transaction requires executing all the tasks that comprise a unit of work in entirety all or nothing proposition. For each transaction specify its frequency of execution.

Answer: The main transaction will be adding new books to the database, the title, the release date, the language and the full text must be added. Books and authors are represented by separate entities therefore if a books author is not present in the database it must also be created. To create and Author the authors full nane is required. The transaction to create a book will be executed much more frequently than the transaction to create an author.

6 Identification and Documentation of Database Queries

Unlike transactions, database *queries* do not change the data in the database. Queries require only read access to the database. Some queries may take quite a bit of time to complete. Therefore, performance is often an issue for database queries.

Specify queries in plain English. For each query specify what data is to be retrieved (not how) as well as its frequency of execution.

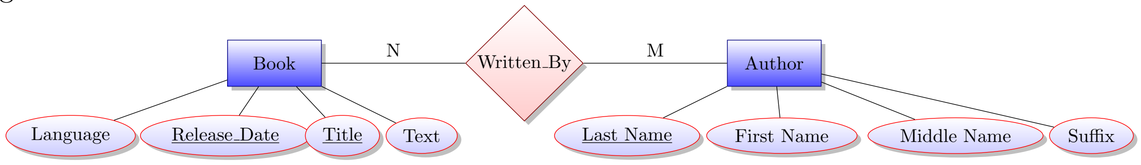
Describe your database queries (in English, not in SQL) here.

Answer: Queries sorted by frequency,

1. Stream the full text of all books in the database.
2. Look up book by primary key (release date, title) returns all information about the book
3. Look up author by primary key (full name) returns all book written by the author.
4. Look up books by language, all book in the specified language are returned.
5. Look up all books released in a certain year all books written in that year are returned

7 Conceptual Data Model

Start with Entity-Relationship (E-R) diagram for department-wise transactions and queries. The number of departments you will have (e.g., registrar, library, financial aid, campus housing) depends on the scope of the data-intensive application. In the second step, integrate these department-wise diagrams into one corporate-wide ER diagram. Follow established diagrammatic conventions. Use SQL Power Architect tool (or similar) for developing E-R diagrams.



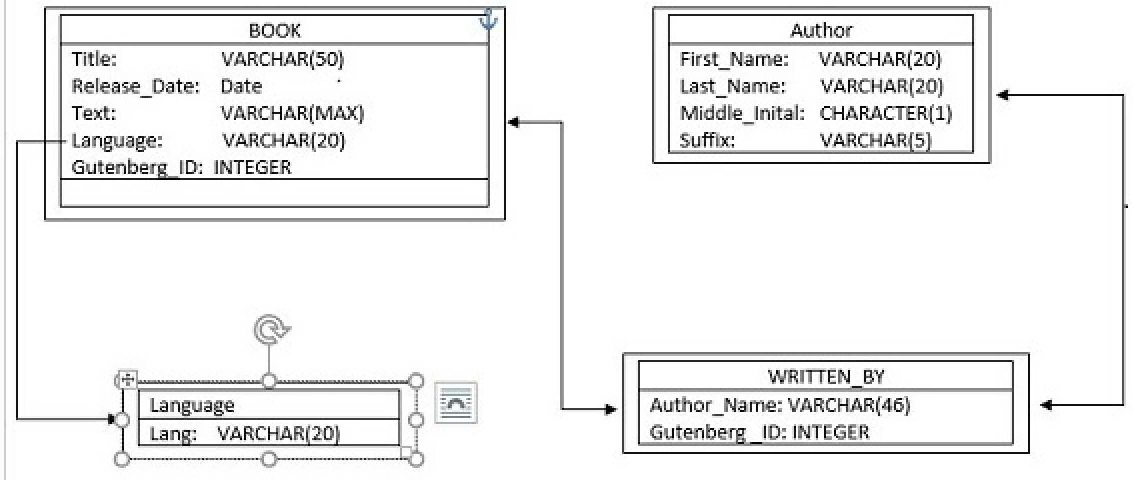
8 Logical Data Model

Identify functional and multivalued dependencies. Transform ER/EER diagrams into a relational schema. Determine functional dependencies and perform normalization. Transform each table into 3NF or BCNF using the functional dependencies and normalization rules. You may use Database Design (DBD) tool for this task. For each table in the final schema, specify primary and foreign keys. Also specify data integrity constraints.

Answers:
book(title, text, release_date, language)
attribute language can be mutivalued.
written_by(book_key, author_key)
author(last_name, first_name, middle_name, suffix)

9 Physical Data Model

For each database file, specify *initial* storage structures and access paths. Typically, these storage structures and access paths need modifications based on *observed performance* once the database is in operation (aka database tuning).



10 Database Creation and Data Loading

Now that your logical database schema and physical database design is in place, write SQL scripts to create the database using PostgreSQL. Load existing data into the tables using either SQL statements or *bulk loading.* Resolve any data integrity constraint violations.

Database loading script is located in code listings.

11 Implementing Database Transactions and Queries

Write SQL code for transactions and queries. Verify and validate all transactions and queries. Comment SQL code sensibly.

Here is an LaTeX markup for typesetting SQL code.

```
SELECT name AS "Country␣Name", population AS "Population",
   lifeexpectancy AS "Life␣Expectancy"
FROM   country
WHERE  lifeexpectancy IS NOT NULL
ORDER  BY lifeexpectancy DESC;
```

For the uninitiated, you may use the verbatim command for an uninspiring typeset.

```
SELECT name AS "Country Name",
       population     AS "Population",
       lifeexpectancy AS "Life Expectancy"
FROM   country
WHERE  lifeexpectancy IS NOT NULL
ORDER  BY lifeexpectancy DESC;
```

```
CREATE DATABASE gutenburg
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    CONNECTION LIMIT = -1;

CREATE TABLE public."Book"
(
    gutenberg_id text NOT NULL,
    release_date date,
    full_text text,
    language text NOT NULL,
    title text NOT NULL,
    PRIMARY KEY (gutenberg_id)
)
WITH (
    OIDS = FALSE
);

ALTER TABLE public."Book"
    OWNER to postgres;

CREATE TABLE public."Author"
(
    author_id serial,
    first_name text NOT NULL,
    last_name text,
    middle_name text,
    suffix text,
    prefix text,
    PRIMARY KEY (author_id)
)
WITH (
    OIDS = FALSE
);

ALTER TABLE public."Author"
    OWNER to postgres;

CREATE TABLE public."Written_By"
(
    author_id integer NOT NULL,
    gutenberg_id text NOT NULL,
    PRIMARY KEY (author_id, gutenberg_id),
    CONSTRAINT gutenberg_id FOREIGN KEY (gutenberg_id)
        REFERENCES public."Book" (gutenberg_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT author_id FOREIGN KEY (author_id)
        REFERENCES public."Author" (author_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
);

ALTER TABLE public."Written_By"
    OWNER to postgres;

--the symbol ? will represent a variable to be replaced with
   the user's actual information need

--get a book by gutenberg_id
Select * From public."Book"
Where gutenberg_id = ?

--get book by title
Select * From public."Book"
```

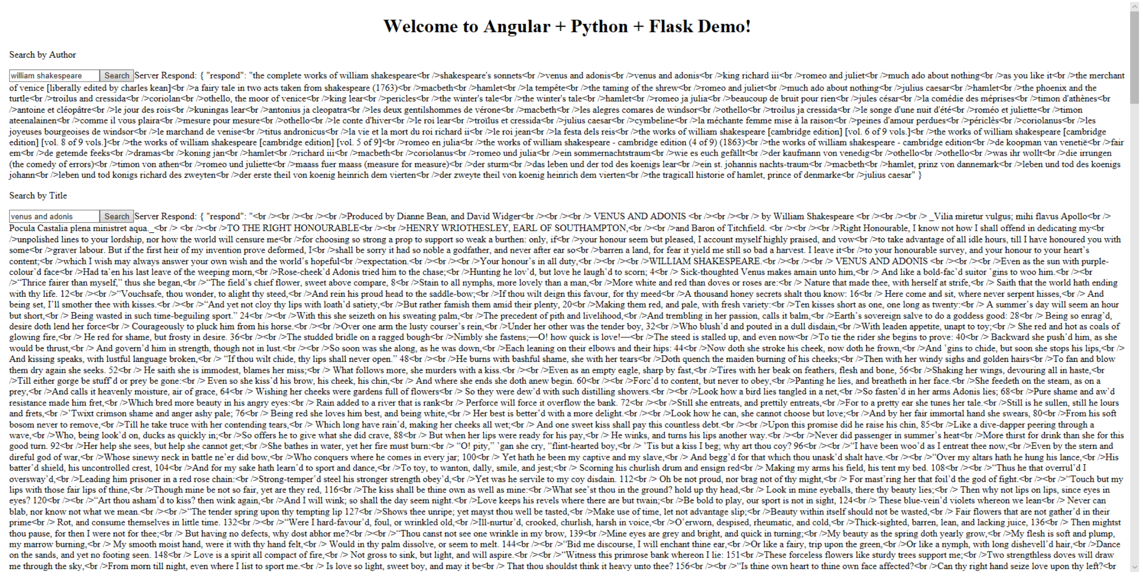
```
Where title = ?

-- get all books in a certain language
Select * from public."Book"
Where language = ?

-- get author by name
Select * From public."Author"
Where first_name = ? And middle_name = ?
And last_name = ? And suffix = ?
And prefix = ?

--get author by author_id
Select * from public."Author"
Where author_id = ?

--get all books by an author with author_id
select * from public."Book"
natural join public."Written_By"
where author_id = ?

--get all authors of a book with gutenberg_id
select * from public."Author"
natural join public."Written_By"
where public."Written_By".gutenberg_id = ?

--get all available languages
select distinct language from public."Book"
```

12 Developing Database Applications

This step involves writing database applications using Java or scripting languages such as JSP, PHP, and ASP.NET. Include rationale for choosing a specific language for developing the database applications. Students should not choose a scripting language unless they are already familiar with it. Simply there is no time to learn a new scripting language. Demonstrate a simple Web application based on the database that you have developed.

Discuss the design and implementation details of the database application here. Please do not include actual code. You may include code in Appendix.

Answer:
We had decided to use Angular as our frontend, and flask as our backend. Postgresql would be the database. Angular is written in typescript which is pretty much the javascript and Flask is written in python. We would create the functionality base on the application functions that we discuss in the previous section.



Welcome to Angular + Python + Flask Demo!

Search by Author

william shakespeare Search

```
Server Respond: { "respond": "the complete works of william shakespeare<br />shakespeare's sonnets<br />venus and adonis<br />venus and adonis<br />king richard iii<br />romeo and juliet<br />much ado about nothing<br />as you like it<br />the merchant
of venice [liberally edited by charles kean]<br />a fairy tale in two acts taken from shakespeare (1763)<br />macbeth<br />hamlet<br />la tempête<br />the taming of the shrew<br />romeo and juliet<br />much ado about nothing<br />julius caesar<br />hamlet<br />the phoenix and the
turtle<br />troilus and cressida<br />coriolan<br />othello, the moor of venice<br />king lear<br />pericles<br />the winter's tale<br />the winter's tale<br />hamlet<br />romeo ja julia<br />beaucoup de bruit pour rien<br />jules césar<br />la comédie des méprises<br />timon d'athènes<br
/>antoine et cléopâtre<br />le jour des rois<br />kuningas lear<br />antonius ja cleopatra<br />les deux gentilshommes de vérone<br />macbeth<br />les alegres comares de windsor<br />othello<br />troilus ja cressida<br />le songe d'une nuit d'été<br />roméo et juliette<br />timon
ateenalainen<br />comme il vous plaira<br />mesure pour mesure<br />othello<br />le conte d'hiver<br />le roi lear<br />troïlus et cressida<br />julius caesar<br />cymbeline<br />la méchante femme mise à la raison<br />peines d'amour perdues<br />périclès<br />coriolanus<br />les
joyeuses bourgeoises de windsor<br />le marchand de venise<br />titus andronicus<br />la vie et la mort du roi richard ii<br />le roi jean<br />la festa dels reis<br />the works of william shakespeare [cambridge edition] [vol. 6 of 9 vols.]<br />the works of william shakespeare [cambridge
edition] [vol. 8 of 9 vols.]<br />the works of william shakespeare [cambridge edition] [vol. 5 of 9]<br />romeo en julia<br />the works of william shakespeare - cambridge edition (4 of 9) (1863)<br />the works of william shakespeare - cambridge edition<br />de koopman van venetië<br />fair
em<br />de getemde feeks<br />dramas<br />koning jan<br />hamlet<br />richard iii<br />macbeth<br />coriolanus<br />romeo und julia<br />ein sommernachtstraum<br />wie es euch gefällt<br />der kaufmann von venedig<br />othello<br />othello<br />was ihr wollt<br />die irrungen
(the comedy of errors)<br />timon von athen<br />romeo und juliette<br />maass fuer maass (measure for measure)<br />der sturm<br />das leben und der tod des koenigs lear<br />ein st. johannis nachts-traum<br />macbeth<br />hamlet, prinz von dannemark<br />leben und tod des koenigs
johann<br />leben und tod konigs richard des zweyten<br />der erste theil von koenig heinrich dem vierten<br />der zweyte theil von koenig heinrich dem vierten<br />the tragicall historie of hamlet, prince of denmarke<br />julius caesar" }
```

Search by Title

venus and adonis Search

```
Server Respond: { "respond": "<br /><br /><br /><br />Produced by Dianne Bean, and David Widger<br /><br /><br /> VENUS AND ADONIS <br /><br /><br /> by William Shakespeare <br /><br /><br /> _Vilia miretur vulgus; mihi flavus Apollo<br />
Pocula Castalia plena ministret aqua._<br /> <br /><br />TO THE RIGHT HONOURABLE<br /><br />HENRY WRIOTHESLEY, EARL OF SOUTHAMPTON,<br /><br />and Baron of Titchfield. <br /><br /><br />Right Honourable, I know not how I shall offend in dedicating my<br
/>unpolished lines to your lordship, nor how the world will censure me<br />for choosing so strong a prop to support so weak a burthen: only, if<br />your honour seem but pleased, I account myself highly praised, and vow<br />to take advantage of all idle hours, till I have honoured you with
some<br />graver labour. But if the first heir of my invention prove deformed, I<br />shall be sorry it had so noble a godfather, and never after ear so<br />barren a land, for fear it yield me still so bad a harvest. I leave it<br />to your honourable survey, and your honour to your heart's
content;<br />which I wish may always answer your own wish and the world's hopeful<br />expectation.<br /><br /><br />Your honour's in all duty,<br /><br /><br />WILLIAM SHAKESPEARE.<br /><br /><br /> VENUS AND ADONIS <br /><br /><br />Even as the sun with purple-
colour'd face<br />Had ta'en his last leave of the weeping morn,<br />Rose-cheek'd Adonis tried him to the chase;<br />Hunting he lov'd, but love he laugh'd to scorn; 4<br /> Sick-thoughted Venus makes amain unto him,<br /> And like a bold-fac'd suitor 'gins to woo him.<br /><br
/>"Thrice fairer than myself," thus she began,<br />"The field's chief flower, sweet above compare, 8<br />Stain to all nymphs, more lovely than a man,<br />More white and red than doves or roses are:<br /> Nature that made thee, with herself at strife,<br /> Saith that the world hath ending
with thy life. 12<br /><br />"Vouchsafe, thou wonder, to alight thy steed,<br />And rein his proud head to the saddle-bow;<br />If thou wilt deign this favour, for thy meed<br />A thousand honey secrets shalt thou know: 16<br /> Here come and sit, where never serpent hisses,<br /> And
being set, I'll smother thee with kisses.<br /><br />"And yet not cloy thy lips with loath'd satiety,<br />But rather famish them amid their plenty, 20<br />Making them red, and pale, with fresh variety:<br />Ten kisses short as one, one long as twenty:<br /> A summer's day will seem an hour
but short,<br /> Being wasted in such time-beguiling sport." 24<br /><br />With this she seizeth on his sweating palm,<br />The precedent of pith and livelihood,<br />And trembling in her passion, calls it balm,<br />Earth's sovereign salve to do a goddess good: 28<br /> Being so enrag'd,
desire doth lend her force<br /> Courageously to pluck him from his horse.<br /><br />Over one arm the lusty courser's rein,<br />Under her other was the tender boy, 32<br />Who blush'd and pouted in a dull disdain,<br />With leaden appetite, unapt to toy;<br /> She red and hot as coals of
glowing fire,<br /> He red for shame, but frosty in desire. 36<br /><br />The studded bridle on a ragged bough<br />Nimbly she fastens;—O! how quick is love!—<br />The steed is stalled up, and even now<br />To tie the rider she begins to prove: 40<br /> Backward she push'd him, as she
would be thrust,<br /> And govern'd him in strength, though not in lust.<br /><br />So soon was she along, as he was down,<br />Each leaning on their elbows and their hips: 44<br />Now doth she stroke his cheek, now doth he frown,<br />And 'gins to chide, but soon she stops his lips,<br />
And kissing speaks, with lustful language broken,<br /> "If thou wilt chide, thy lips shall never open." 48<br /><br />He burns with bashful shame, she with her tears<br />Doth quench the maiden burning of his cheeks;<br />Then with her windy sighs and golden hairs<br />To fan and blow
them dry again she seeks. 52<br /> He saith she is immodest, blames her miss;<br /> What follows more, she murders with a kiss.<br /><br />Even as an empty eagle, sharp by fast,<br />Tires with her beak on feathers, flesh and bone, 56<br />Shaking her wings, devouring all in haste,<br
/>Till either gorge be stuff'd or prey be gone:<br /> Even so she kiss'd his brow, his cheek, his chin,<br /> And where she ends she doth anew begin. 60<br /><br />Forc'd to content, but never to obey,<br />Panting he lies, and breatheth in her face.<br />She feedeth on the steam, as on a
prey,<br />And calls it heavenly moisture, air of grace, 64<br /> Wishing her cheeks were gardens full of flowers<br /> So they were dew'd with such distilling showers.<br /><br />Look how a bird lies tangled in a net,<br />So fasten'd in her arms Adonis lies; 68<br />Pure shame and aw'd
resistance made him fret,<br />Which bred more beauty in his angry eyes:<br /> Rain added to a river that is rank<br /> Perforce will force it overflow the bank. 72<br /><br />Still she entreats, and prettily entreats,<br />For to a pretty ear she tunes her tale.<br />Still is he sullen, still he lours
and frets,<br />'Twixt crimson shame and anger ashy pale; 76<br /> Being red she loves him best, and being white,<br /> Her best is better'd with a more delight.<br /><br />Look how he can, she cannot choose but love;<br />And by her fair immortal hand she swears, 80<br />From his soft
bosom never to remove,<br />Till he take truce with her contending tears,<br /> Which long have rain'd, making her cheeks all wet;<br /> And one sweet kiss shall pay this countless debt.<br /><br />Upon this promise did he raise his chin, 85<br />Like a dive-dapper peering through a
wave,<br />Who, being look'd on, ducks as quickly in;<br />So offers he to give what she did crave, 88<br /> But when her lips were ready for his pay,<br /> He winks, and turns his lips another way.<br /><br />Never did passenger in summer's heat<br />More thirst for drink than she for this
good turn. 92<br />Her help she sees, but help she cannot get;<br />She bathes in water, yet her fire must burn:<br /> "O! pity," 'gan she cry, "flint-hearted boy,<br /> 'Tis but a kiss I beg; why art thou coy? 96<br /><br />"I have been woo'd as I entreat thee now,<br />Even by the stern and
direful god of war,<br />Whose sinewy neck in battle ne'er did bow,<br />Who conquers where he comes in every jar; 100<br /> Yet hath he been my captive and my slave,<br /> And begg'd for that which thou unask'd shalt have.<br /><br />"Over my altars hath he hung his lance,<br />His
batter'd shield, his uncontrolled crest, 104<br />And for my sake hath learn'd to sport and dance,<br />To toy, to wanton, dally, smile, and jest;<br /> Scorning his churlish drum and ensign red<br /> Making my arms his field, his tent my bed. 108<br /><br />"Thus he that overrul'd I
oversway'd,<br />Leading him prisoner in a red rose chain:<br />Strong-temper'd steel his stronger strength obey'd,<br />Yet was he servile to my coy disdain. 112<br /> Oh be not proud, nor brag not of thy might,<br /> For mast'ring her that foil'd the god of fight.<br /><br />"Touch but my
lips with those fair lips of thine,<br />Though mine be not so fair, yet are they red, 116<br />The kiss shall be thine own as well as mine:<br />What see'st thou in the ground? hold up thy head,<br /> Look in mine eyeballs, there thy beauty lies;<br /> Then why not lips on lips, since eyes in
eyes? 120<br /><br />"Art thou asham'd to kiss? then wink again,<br />And I will wink; so shall the day seem night.<br />Love keeps his revels where there are but twain;<br />Be bold to play, our sport is not in sight, 124<br /> These blue-vein'd violets whereon we lean<br /> Never can
blab, nor know not what we mean.<br /><br />"The tender spring upon thy tempting lip 127<br />Shows thee unripe; yet mayst thou well be tasted,<br />Make use of time, let not advantage slip;<br />Beauty within itself should not be wasted,<br /> Fair flowers that are not gather'd in their
prime<br /> Rot, and consume themselves in little time. 132<br /><br />"Were I hard-favour'd, foul, or wrinkled old,<br />Ill-nurtur'd, crooked, churlish, harsh in voice,<br />O'erworn, despised, rheumatic, and cold,<br />Thick-sighted, barren, lean, and lacking juice, 136<br /> Then mightst
thou pause, for then I were not for thee;<br /> But having no defects, why dost abhor me?<br /><br />"Thou canst not see one wrinkle in my brow, 139<br />Mine eyes are grey and bright, and quick in turning;<br />My beauty as the spring doth yearly grow,<br />My flesh is soft and plump,
my marrow burning,<br /> My smooth moist hand, were it with thy hand felt,<br /> Would in thy palm dissolve, or seem to melt. 144<br /><br />"Bid me discourse, I will enchant thine ear,<br />Or like a fairy, trip upon the green,<br />Or like a nymph, with long dishevell'd hair,<br />Dance
on the sands, and yet no footing seen. 148<br /> Love is a spirit all compact of fire,<br /> Not gross to sink, but light, and will aspire.<br /><br />"Witness this primrose bank whereon I lie: 151<br />These forceless flowers like sturdy trees support me;<br />Two strengthless doves will draw
me through the sky,<br />From morn till night, even where I list to sport me.<br /> Is love so light, sweet boy, and may it be<br /> That thou shouldst think it heavy unto thee? 156<br /><br />"Is thine own heart to thine own face affected?<br />Can thy right hand seize love upon thy left?<br
```

13 Summary of Revisions

Briefly describe who critiqued your document (e.g., instructor, peer, friend) and provided suggestions for improvement, and how you have incorporated the suggestions and revised the document.

Include information on: who critiqued your document, what suggestions were made, and how you incorporated the suggestions and revised the document.

Answer:
After we complete the website to communicate with database, we had talked to Dr.Gudivada for some suggestions and all we need to improve is to add in the css style make the website look more pleasant.

14 Metacognitive Reflection

We learn how to learn through metacognitive reflection by actively planning, monitoring, and evaluating our own thinking and learning.

Learning how to learn involves going beyond the cognitive and into the realm of the metacognitive. In the context of this assignment, cognitive part is the development of the data-intensive application. Metacognitive part refers to the strategies, techniques, and tools you have used to accomplish these tasks.

Perform metacognitive reflection on this assignment by answering the following questions:

1. Did I solve the right problem?

   Yes, we did.

2. Did I solve the problem right?

   Yes, we did.

3. How did I approach solutions to the problems?

   Setting up database and website to show deliverable.

4. What strategies and techniques did I draw upon?

   We drew upon prior knowledge when cleaning out data. We use methods learned in class to load data into out database .

5. Did I learn a new strategy in completing this assignment? If so, how is it different from and similar to the repertoire of techniques that I have already acquired?

   Yes we learned how to interface with a PostgreSQL database from an application program, We also learned how to create a simple web app with angularJS and flask and have it interface with our database

6. Any other information you may wish to add ···

   This assignment provided a great opprotunity to apply concepts learned in class.

15 Self-assessment

You need to assign a grade for this assignment yourself. Use the rubric listed below to come up with a score. The instructor will also assign a score. Without this section, assignment will be returned with a score of 0.

The first two traits correspond to writing and the remaining ones relate to domain aspects of the project.

Perf Level / Trait	Poor	Fair	Good	Outstanding
Diction	Chooses non-technical vocabulary that inadequately conveys the intended meaning of the communication.	Chooses technical vocabulary that conveys the intended meaning of the communication.	Chooses appropriate, technical, and varied vocabulary that conveys the intended meaning of the communication.	Chooses lively, precise, technical, and compelling vocabulary and skillfully communicates the message.
Communication Style	Has only a few (but noticeable) errors in style, mechanics, or other issues that might distract from the message.	Is virtually free of mechanical, stylistic or other issues.	Uses complex and varied sentence styles, concepts, or visual representations.	Creates a distinctive communication style by combining a variety of materials, ideas, or visual representations.
Application Selection	Not a data-intensive application.	Application is somewhat data-intensive	Application is data-intensive but limited access to domain expertise.	Application is data-intensive with adequate access to domain expertise.
Use-cases	Less than 50% of the use-cases are identified, and documented poorly.	Over 75% of the uses-cases are identified and documented using a standard template.	All the use-cases are identified, but detail is missing for some use-cases.	All the use-cases are identified, well-documented using a standard template, and verified against application requirements.
Data Tasks	Inputs, outputs, and possible error conditions are documented for less than 50% of data tasks.	Inputs, outputs, and possible error conditions are documented for less than 75% of data tasks.	Inputs, outputs, and possible error conditions are documented for all data tasks.	Inputs, outputs, and possible error conditions are documented for all data tasks. Processing logic (or high-level algorithms) for transforming inputs into outputs is also described.
Transactions and Queries	Less than 50% of the transactions and queries are identified and described.	Less than 75% of the transactions and queries are identified and described.	All the transactions and queries are identified and described.	All the transactions and queries are identified and described including their frequency of execution.

Perf Level / Trait	Poor	Fair	Good	Outstanding
Data Models	Only conceptual data model is described in detail. Cursory treat of logical data model. Physical data model design is missing.	Conceptual and logical data models are described in detail. Physical data model design is missing.	Conceptual, logical, and physical data models are described completely and precisely.	Conceptual, logical, and physical data models are described completely and precisely. Database normalization based on functional dependencies is discussed in detail.
Creation and Loading	SQL scripts are written and executed to create the database and load the data. Data in the database is trivial in size.	SQL scripts are written and executed to create the database and load the data. Data in the database is moderate in size.	Conceptual, logical, and physical data models are described completely and precisely. Data in the database is huge in size – in the order of millions of rows.	Conceptual, logical, and physical data models are described completely and precisely. Data in the database is huge in size – in the order of millions of rows. Detail evidence is provided on how referential integrity constraints are resolved.
Implementing Transactions and Queries	Less than 50% of the transactions and queries are implemented.	Less than 75% of the transactions and queries are implemented.	All the transactions and queries are implemented; run and execute correctly.	All the transactions and queries are implemented; run and execute correctly. There is also written evidence that transactions and queries are tested.
Revisions	Only peer or instructor/grader feedback is solicited, but not incorporated.	Both peer and instructor/grader feedback is solicited but not incorporated.	Both peer and instructor/grader feedback is solicited and incorporated.	Both peer and instructor/grader feedback solicited and incorporated. Evidence is presented to show how the feedback improved the document.
Meta-cognitive Reflection	Not performed.	Is shallow and incomplete.	Is complete but not thorough.	Is complete and thorough.

Use the following table to score your solution. Circle the appropriate number in each row. For example, to circle 4, use the LaTeX markup code `\circled{4}`, which produces $\textcircled{4}$.

Perf Level	*Poor*	*Fair*	*Good*	*Outstanding*
Trait				
Diction	2	3	4	5
Communication Style	2	3	4	5
Application Selection	4	6	10	15
Use-cases	4	6	8	10
Data Tasks	4	6	8	10
Transactions and Queries	4	6	8	10
Data Models	4	6	8	10
Creation and Loading	4	6	8	10
Implementing Transactions and Queries	4	6	8	10
Revisions	4	6	8	10
Meta-cognitive Reflection	2	3	4	5

Total score: 100 / 100.

A Code Listings

```
#Code for extracting data from Gutenberg corpus
#Directory set up:
#This program need:
#   input folder: for input files
#   output folder: for output files
#   duplicate folder: for duplicate files
#   organized folder: for organized files
#       txt folder: for organized text files
#       other foler: for organized other files
#       orgDuplicate folder: for organized duplicate files
#   extract folder: for extracted data store
#       _data folder: extracted information
#       encodeErr folder: encode error files
#       exist folder: data exist
#       nonExist folder: data could not be determine author, title, language exist or not


#All user need to do is put input file insdie input folder and run the program

'''
directory structure:
.
+--extract.py
+--input
+--output
+--duplicate
+--organized
   +--txt
   +--other
   +--orgDuplicate
+--extract
   +--_data
   +--exist
      +--missInfo
   +--nonExist

'''
# =========================================================================

#imports
import os
import zipfile

def makeDir():
    os.mkdir("input")
    os.mkdir("output")
    os.mkdir("duplicate")
    os.mkdir("organized")
    os.mkdir("organized/txt")
    os.mkdir("organized/other")
    os.mkdir("organized/orgDuplicate")
    os.mkdir("extract")
    os.mkdir("extract/_data")
    os.mkdir("extract/exist")
    os.mkdir("extract/exist/missInfo")
    os.mkdir("extract/nonExist")
# ========================================================================

#Use Recursion to loop into most inner file if it's zip unzip and step back one directory and go to second, so on.....
def unzipFiles(inputDir,outputDir,fileNameList,duplicateCounter,otherFile,duplicateFile,duplicateDir):
    fileList = os.listdir(inputDir)

    for fileName in fileList:
        #if it's directory
        if os.path.isdir(os.path.join(inputDir,fileName)):
            newInputDir = os.path.join(inputDir, fileName)
            unzipFiles(newInputDir,outputDir,fileNameList,duplicateCounter,otherFile,duplicateFile,duplicateDir)
        #if it's zip files
        elif ".zip" in fileName:
            #if file is duplicate unzip to duplicate
            if fileName in fileNameList:
                duplicateCounter+=1
                duplicateFile.write(fileName+"␣"+str(duplicateCounter)+"␣\n")
                unzipDir = os.path.join(inputDir,fileName)
                zip_ref = zipfile.ZipFile(unzipDir, 'r')
                zip_ref.extractall(duplicateDir)
                print(fileName+"␣DONE")
                zip_ref.close()
            #else unzip file to output folder
            else:
                fileNameList.append(fileName)
                unzipDir = os.path.join(inputDir,fileName)
                zip_ref = zipfile.ZipFile(unzipDir, 'r')
                zip_ref.extractall(outputDir)
                print(fileName+"␣DONE")
                zip_ref.close()
        #else not zip file not directory, record it
        else:
            print("other␣files")
            otherFile.write(fileName+"\n")
# =========================================================================

#Organize text file into organized folder, categorize file into text or other.
def organize(inputDir,categorizedTxtDir,categorizedOtherDir,categorizedDuplicateDir,fileNameList):
    outputFileList = os.listdir(inputDir)

    for fileName in outputFileList:
        if os.path.isdir(os.path.join(inputDir,fileName)):
            newInputDir = os.path.join(inputDir, fileName)
            organize(newInputDir,categorizedTxtDir,categorizedOtherDir,categorizedDuplicateDir,fileNameList)
        elif ".txt" or ".TXT" in fileName:
            fileName = fileName.replace(".TXT",".txt")
            if fileName in fileNameList:
                currentFile = os.path.join(inputDir,fileName)
                newFile = os.path.join(categorizedDuplicateDir,fileName)
                os.rename(currentFile,newFile)
                print(fileName+"␣DONE")
            else:
                fileNameList.append(fileName)
                currentFile = os.path.join(inputDir,fileName)
                newFile = os.path.join(categorizedTxtDir,fileName)
                os.rename(currentFile,newFile)
                print(fileName+"␣DONE")
        else:
            currentFile = os.path.join(inputDir,fileName)
            newFile = os.path.join(categorizedOtherDir,fileName)
            os.rename(currentFile,newFile)
            print(fileName+" DONE")

# =========================================================================

def extractData(categorizedTxtDir,extractExistDir,extractNonExistDir,extractDataDir,extractExistMissDir,encodeErrDir):
    categorizedTxtFileList = os.listdir(categorizedTxtDir)

    for fileName in categorizedTxtFileList:
        title =""
        author =""
        release =""
        language =""
        text =""
        asciiErr = False
        utf8Err = False

        currentFileDir = os.path.join(categorizedTxtDir,fileName)
        currentFile = open(currentFileDir,"r",encoding='ascii')
        currentFile = open(currentFileDir,"r",encoding='utf-8')
        try:
            ascContent = currentFile.read()
        except:
            asciiErr=True

        try:
            utfContent = currentFile.read()
        except:
            utf8Err=True

        if asciiErr == False:
            content = ascContent
        elif utf8Err == False:
            content = utfContent
        else:
            currentFile = os.path.join(categorizedTxtDir,fileName)
            newFile = os.path.join(encodeErrDir,fileName)
            os.rename(currentFile,newFile)
            continue


        splitFile = content.split("*** START OF THIS PROJECT GUTENBERG EBOOK")

        if len(splitFile) == 2:
            info = splitFile[0]
            text = splitFile[1]
            splitLines = info.split("\n")
            for eachLine in splitLines:
                if "Title:" in eachLine:
                    title = eachLine.replace("\n","")
                elif "Author:" in eachLine:
                    author = eachLine.replace("\n","")
                elif "Release Data:" in eachLine:
                    release = eachLine.replace("\n","")
                elif "Language:" in eachLine:
                    language = eachLine.replace("\n","")

            if title == "" or author == "" or release == "" or language == "":
                currentFile = os.path.join(categorizedTxtDir,fileName)
                newFile = os.path.join(extractExistMissDir,fileName)
                os.rename(currentFile,newFile)
            else:
                writeNewFileName = os.path.join(extractDataDir,fileName)
                writeNewFile = open(writeNewFileName,"w")
                writeNewFile.write(title+"\n"+author+"\n"+release+"\n"+language+"\n Text: "+text)
                writeNewFile.close()
                currentFile = os.path.join(categorizedTxtDir,fileName)
                newFile = os.path.join(extractExistDir,fileName)
                os.rename(currentFile,newFile)
        else:
            currentFile = os.path.join(categorizedTxtDir,fileName)
            newFile = os.path.join(extractNonExistDir,fileName)
            os.rename(currentFile,newFile)
        print(fileName+"␣Done")
# ================================================================

#main
#1
currentDir = os.getcwd()
inputDir = os.path.join(currentDir, "input")
outputDir = os.path.join(currentDir, "output")
otherFile = open(os.path.join(currentDir,"other.txt"),"w")
duplicateDir = os.path.join(currentDir, "duplicate")
duplicateFile = open(os.path.join(currentDir,"duplicate.txt"),"w")
fileNameList = []
duplicateCounter = 0
#2
categorizedTxtDir = os.path.join(currentDir,"organized","txt")
categorizedOtherDir = os.path.join(currentDir,"organized","other")
categorizedDuplicateDir = os.path.join(currentDir,"organized","orgDuplicate")
#3
extractExistDir = os.path.join(currentDir,"extract","exist")
extractExistMissDir = os.path.join(currentDir,"extract","exist","missInfo")
extractNonExistDir = os.path.join(currentDir,"extract","nonExist")
extractDataDir = os.path.join(currentDir,"extract","_data")
encodeErrDir = os.path.join(currentDir,"extract","encodeErr")

print("Select␣your␣action:")
print("1.␣create␣file␣directories")
print("2.␣Unzip␣all␣files␣from␣input␣directory")
print("3.␣organize␣output␣folder,␣categorize␣it␣into␣text␣files␣and␣other␣files")
print("4.␣extract␣data␣from␣organized/txt␣and␣move␣file␣to␣extract/exist␣or␣extract/nonExist")
print("0.␣quit")

userInput = input()
userInputInt = int(userInput)
if userInputInt == 1:
    makeDir()
```

```
elif userInputInt == 2:
    unzipFiles(inputDir,outputDir, fileNameList,
       duplicateCounter,otherFile,duplicateFile,duplicateDir)
elif userInputInt == 3:
    otherFile.close()
    duplicateFile.close()
    organize(outputDir,categorizedTxtDir,categorizedOtherDir,
       categorizedDuplicateDir,fileNameList)
elif userInputInt == 4:
    extractData(categorizedTxtDir,extractExistDir,
       extractNonExistDir,extractDataDir,extractExistMissDir,
       encodeErrDir)
elif userInputInt == 0:
    exit()


#------------------------------------------------------------#

#code for loading data in to database
import re
import os
import sys

import psycopg2 as pg
import pandas.io.sql as psql
from pprint import pprint


def extractDataFromHeaderLine(dataDescription, line):
    lineS = re.split('^' + dataDescription, line, maxsplit=1)
    #print('lineS is: ')
    #pprint(lineS)
    if(len(lineS) == 2):
        return lineS[1].strip().lower()
    else:
        return ""



#for well formed docs
def getData(file):
    title =""
    author =""
    release =""
    language =""
    text =""
    gutenberId = ""
    charSetEncode=""
    asciiErr = False
    utf8Err = False
    defaultErr = False

    content = None

    currentFileAscii = open(file,"r",encoding='ascii')
    currentFileUtf8 = open(file,"r",encoding='utf-8')
    currentFileDefult = open(file,"r")

    try:
        ascContent = currentFileAscii.read()
    except:
        asciiErr=True
    try:
        utfContent = currentFileUtf8.read()
    except:
        utf8Err=True

    try:
        defaultContent = currentFileDefult.read()
    except:
        defaultErr = True

    if asciiErr == False:
        content = ascContent
    elif utf8Err == False:
        content = utfContent
    elif defaultErr == False:
        content = defaultContent
    else:
        currentFileAscii.close()
        currentFileUtf8.close()
        currentFileDefult.close()

        print('error␣reading␣file:␣' + file)

    #extract the data

    headerAndContent = re.split('\*\*\*', content, maxsplit=1)
    #print(headerAndContent[0])

    header = headerAndContent[0]
    fullText = headerAndContent[1]
    #print(fullText)

    headerSplit = header.split('\n')

    for line in headerSplit:
        if "Title:" in line:
            title = extractDataFromHeaderLine('Title:',line)
        elif "Author:" in line:
            author = extractDataFromHeaderLine('Author:',line)

        elif "Release␣Date:" in line:
            release = extractDataFromHeaderLine('Release␣Date:',line)
            if('[' in release):
                releaseS = release.split('[')
                release = releaseS[0].strip()

        elif "Language:" in line:
            language = extractDataFromHeaderLine('Language:',line)

    '''
    print('The title is: ' + title)
    print('Author: ' + author)
    print('Date: ' + release)
    #pprint(release)
    print('Language: ' + language)
    '''

    return title, author, release, language, fullText


def processAuthorName(authorName):
    firstName = ''
    middleName = ''
    lastName = ''
    suffix = ''
    prefix = ''
    nameL = authorName.split()

    i = 0
    while(i < len(nameL)):
        if(')' in nameL[i] or '(' in nameL[i]):
            del nameL[i]
            i -=1
        i += 1

    nameLen = len(nameL)

    #check for prefix
    comPrefixes = ['mr', 'mrs', 'miss', 'sir', 'lord', 'ms']
    comSuffixes = ['sr', 'jr', 'ii', 'iii', 'iv', 'v']


    if(nameLen > 1):
        if(nameL[0].strip('.') in comPrefixes):
            prefix = nameL[0]
            del nameL[0]
            nameLen = len(nameL)

    if(nameLen > 1):
        if(nameL[nameLen - 1].strip('.') in comSuffixes):
            suffix = nameL[nameLen - 1]
            del nameL[nameLen - 1]
            nameLen = len(nameL)

    if(nameLen == 0):
        firstName = 'anonymous'
    elif(nameLen == 1):
        firstName = nameL[0]
    elif(nameLen == 2):
        firstName = nameL[0]
        lastName = nameL[1]

    elif(nameLen == 3):
        firstName = nameL[0]
        middleName = nameL[1]
        lastName = nameL[2]
    else:
        firstName = nameL[0]
        lastName = nameL[nameLen - 1]
        del nameL[nameLen - 1]
        del nameL[0]
        middleName = '␣'.join(nameL)

    '''
    print('first name: ' + firstName)
    print('middle name: ' + middleName)
```

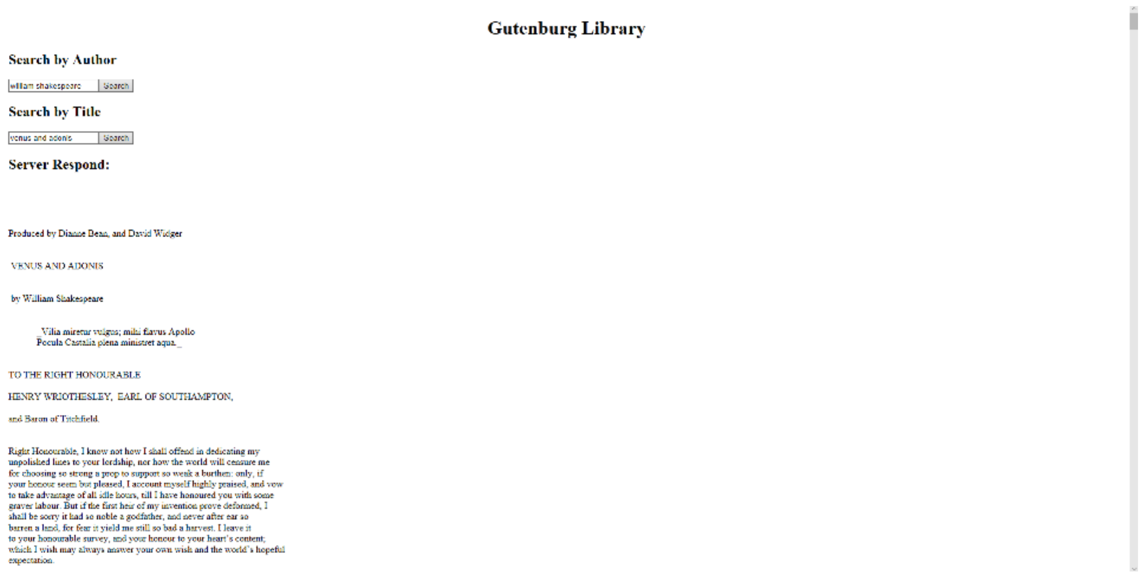
```python
    print('last name: ' + lastName)
    print('suffix: ' + suffix)
    print('prefix: ' + prefix)
    '''

    return firstName, middleName, lastName, suffix, prefix


########## Database Querys

def getBookByPrimaryKey(gutenbergId, cur):
    cur.execute('Select␣*␣From␣public."Book"␣Where␣gutenberg_id␣=␣%s', (gutenbergId,))
    result = cur.fetchall()
    #print(result)
    return result

def bookIsInDatabase(gutenbergId, conn):
    result = getBookByPrimaryKey(gutenbergId, conn)
    #print('book is:')
    #print(result)
    #sys.exit()

    if(len(result) > 0):
        return True
    else:
        return False

def getAuthorByName(firstName, middleName, lastName, suffix, prefix, cur):
    cur.execute('Select␣*␣From␣public."Author"␣Where␣first_name␣=␣%s␣And␣middle_name␣=␣%s␣And␣last_name␣=␣%s␣and␣suffix␣=␣%s␣and␣prefix␣=␣%s',
                (firstName, middleName, lastName, suffix, prefix))

    result = cur.fetchall()
    #print(result)
    return result

def authorIsInDatabase(firstName, middleName, lastName, suffix, prefix, cur):
    result = getAuthorByName(firstName, middleName, lastName, suffix, prefix, cur)

    if(len(result) > 0):
        return True
    else:
        return False


def getWrittenBy(gutenbergId, authorId, cur):
    cur.execute('Select␣*␣From␣public."Written_By"␣Where␣author_id␣=␣%s␣And␣gutenberg_id␣=␣%s', (authorId, gutenbergId))
    result = cur.fetchall()
    return result


def writtenByRelationInDatabase(gutenbergId, authorId, cur):
    result = getWrittenBy(gutenbergId, authorId, cur)

    if(len(result) > 0):
        return True
    else:
        return False


def insertIntoDatabase(gId, title, release, language, author, fullText, conn, cur, brokenFiles):

    #gId = str(gId) + '-' + language
    try:


        if(not bookIsInDatabase(gId, cur)):
            try:
                #inset book
                cur.execute('INSERT␣INTO␣public."Book"␣(gutenberg_id,release_date,full_text,language,title)␣VALUES␣(%s,%s,%s,%s,%s);',
                        (gId, release, fullText, language, title))
                #print('loaded book')
                conn.commit()
            except pg.DataError:
                conn.rollback()
                cur.execute('INSERT␣INTO␣public."Book"␣(gutenberg_id,release_date,full_text,language,title)␣VALUES␣(%s,%s,%s,%s,%s);',
```

```python
                            (gId, None, fullText, language, title))
                print('loaded␣book␣wth␣null␣date:␣' + str(gId))
                conn.commit()

        else:
            print('book␣is␣in␣database,␣gid:␣' + str(gId))

        if('␣and␣' in author):
            authorNameS = re.split('␣and␣', author)
        elif(',' in author):
            authorNameS = re.split(',', author)
        else:
            authorNameS = [author]

        authorNameS = [name.strip() for name in authorNameS if(name != '' or name != None)]

        if(len(authorNameS) > 1):
            pprint(authorNameS)
        for author in authorNameS:

            firstName, middleName, lastName, suffix, prefix = processAuthorName(author)
            #TODO fix rest of querys
            if(not authorIsInDatabase(firstName, middleName, lastName, suffix, prefix, cur)):

            #inset author
                cur.execute('insert␣into␣public."Author"␣(first_name,last_name,middle_name,suffix,prefix)␣VALUES␣(%s,␣%s,␣%s,␣%s,␣%s)␣returning␣author_id',
                            (firstName, lastName, middleName, suffix, prefix))
                #print('loaded author')
                authorId = cur.fetchone()
                conn.commit()
                #print('auth id is: ' + str(authorId[0]))
            else:

                authorId = getAuthorByName(firstName, middleName, lastName, suffix, prefix, cur)[0][0]
                print('author␣is␣in␣database,␣author_id:␣' +
                    str(authorId))

            if(not writtenByRelationInDatabase(gId, authorId, cur)):
                cur.execute('INSERT␣INTO␣public."Written_By"␣(author_id,␣gutenberg_id)␣VALUES␣(%s,␣%s)', (authorId, gId))
                conn.commit()
                #print('loaded written by')
            else:
                wb = getWrittenBy(gId, authorId, cur)
                #print(wb[0])
                #print('written by relation in database: ' + str(wb[0][0]), + ', ' + str(wb[0][1]))


    #    #inset written_by
    except Exception:
        #pass
        brokenFiles.append(gId)
        print(Exception)
        print(gId)



#            MAIN             #

dbname = "gutenburg"
dbhost = "localhost"
dbport = "5432"
dbuser = "postgres"
dbpassword = "postgres"


conn = pg.connect(database=dbname, host=dbhost, port=dbport, user=dbuser, password=dbpassword)
cur = conn.cursor()
#dirToProcess = '/Users/edwardsja15/desktop/gutenberg/load/extract/_data'
#dirToprocessWindows = '/home/reilly/database/load/_data'

dirToProcess = '/home/reilly/database/load/_data'


fileList = os.listdir(dirToProcess)
```

```python
#print(fileList[:10])
i = 0
brokenFiles = []
invalidIds = []
for filename in fileList:
    #can load
    if('.txt' in filename):
        fullfilename = os.path.join(dirToProcess, filename)

        title, author, release, language, fullText = getData(
           fullfilename)

        if(language == 'en'):
            language = 'english'

        filenameS = filename.split('.')
        isValid = False

        if('-' in filenameS[0]):
            idSplit = filenameS[0].split('-')
            try:
                int(idSplit[1])
                isValid = True
            except ValueError:
                isValid = False

                invalidIds.append(filename)

        else:
            isValid = True

        gutenbergId = filenameS[0]
        print('filename␣is:␣' + filename)

        if(isValid):
            insertIntoDatabase(gutenbergId, title, release,
               language, author, fullText, conn, cur,
               brokenFiles)
        #i += 1
        #if(i > 1050):
        #    break
print('invalid␣ids:')
pprint(invalidIds)
print('broken␣files:')
pprint(brokenFiles)
```

```
conn.close()
```

B Test Cases



Produced by Dianne Bean, and David Widger

VENUS AND ADONIS

by William Shakespeare

_Vilia miretur vulgus; mihi flavus Apollo
Pocula Castalia plena ministret aqua._

TO THE RIGHT HONOURABLE

HENRY WRIOTHESLEY, EARL OF SOUTHAMPTON,

and Baron of Titchfield.

Right Honourable, I know not how I shall offend in dedicating my
unpolished lines to your lordship, nor how the world will censure me
for choosing so strong a prop to support so weak a burthen: only, if
your honour seem but pleased, I account myself highly praised, and vow
to take advantage of all idle hours, till I have honoured you with some
graver labour. But if the first heir of my invention prove deformed, I
shall be sorry it had so noble a godfather, and never after ear so
barren a land, for fear it yield me still so bad a harvest. I leave it
to your honourable survey, and your honour to your heart's content;
which I wish may always answer your own wish and the world's hopeful
expectation.

C Other

Full code listings can be seen at:
https://github.com/reiman2222/gutenberg-corpus-for-learning-SQL-and-IR