ISOM 671: Managing Big Data (Assignment 2)

Sean Jung

There are 5 numbered questions (1-point each).* Please submit your assignment as a single PDF or Word file by uploading it to course canvas page. *You should provide: SQL statements, results of any SQL statement (typically copy first 10 rows), and answers to questions, if any.

1. Use MySQL Workbench to create an EER diagram for a database that stores information about the downloads that users make.

   - Each user must have an email address, first name, and last name.
   - Each user can have one or more downloads.
   - Each download must have a filename and download date/time.
   - Each product can be related to one or more downloads.
   - Each product must have a name.

1.1. Then, export a script that creates the database and save this script in a file named hw2-q1.sql. Next, use MySQL Workbench to open this file and review it. Report the script here.

```
-- MySQL Script generated by MySQL Workbench

-- Mon Oct 12 21:35:13 2020

-- Model: New Model    Version: 1.0

-- MySQL Workbench Forward Engineering



SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET                                                                                    @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGI
NE_SUBSTITUTION';



-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------
-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `bigdata_hw2` DEFAULT CHARACTER SET utf8 ;

USE `bigdata_hw2` ;

-- -----------------------------------------------------

-- Table `mydb`.`Products`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bigdata_hw2`.`Products` (

  `Product` VARCHAR(45) NOT NULL,

  PRIMARY KEY (`Product`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Users`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bigdata_hw2`.`Users` (

  `first_name` VARCHAR(45) NOT NULL,

  `last_name` VARCHAR(45) NOT NULL,

  `email_address` VARCHAR(45) NOT NULL,

  PRIMARY KEY (`email_address`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Downloads`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bigdata_hw2`.`Downloads` (

  `filename` VARCHAR(45) NOT NULL,

  `download_date` VARCHAR(45) NOT NULL,

  `Users_email_address` VARCHAR(45) NOT NULL,
```

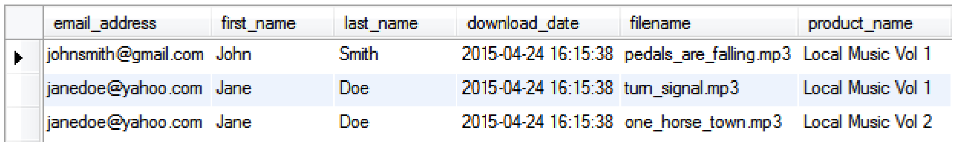
```sql
  `Products_Product` VARCHAR(45) NOT NULL,

  INDEX `fk_Downloads_Users1_idx` (`Users_email_address` ASC) VISIBLE,

  INDEX `fk_Downloads_Products1_idx` (`Products_Product` ASC) VISIBLE,

  CONSTRAINT `fk_Downloads_Users1`

    FOREIGN KEY (`Users_email_address`)

    REFERENCES `mydb`.`Users` (`email_address`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  CONSTRAINT `fk_Downloads_Products1`

    FOREIGN KEY (`Products_Product`)

    REFERENCES `mydb`.`Products` (`Product`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

2. Run the script you created in question 1 to create the database under the name my_web_db. Write a script that adds rows to the database. In particular,

   - Add two rows to the Users and Products tables.
   - Add three rows to the Downloads table:
   - one row for user 1 and product 1;
   - one row for user 2 and product 1;
   - and one row for user 2 and product 2.
   - Use the NOW function to insert the current date and time into the download_date column.

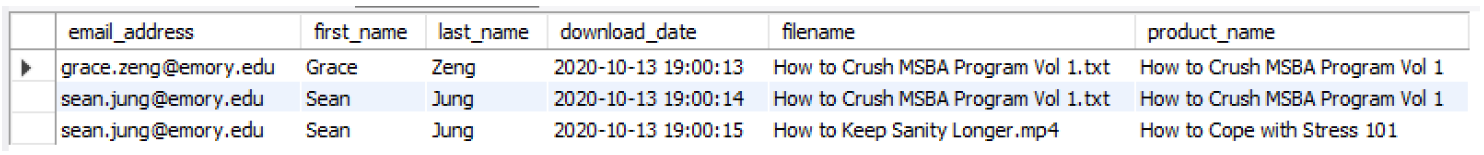
```sql
INSERT INTO products (product_name)
VALUES ('How to Crush MSBA Program Vol 1');

INSERT INTO products (product_name)
VALUES ('How to Cope with Stress 101');

INSERT INTO users (first_name, last_name, email_address)
VALUES ('Sean', 'Jung', 'sean.jung@emory.edu');

INSERT INTO users (first_name, last_name, email_address)
VALUES ('Grace', 'Zeng', 'grace.zeng@emory.edu');

INSERT INTO downloads (filename, download_date, Users_email_address, Products_Product)
VALUES ('How to Crush MSBA Program Vol 1.txt', now(), 'grace.zeng@emory.edu', 'How to Crush MSBA Program Vol 1');

INSERT INTO downloads (filename, download_date, Users_email_address, Products_Product)
VALUES ('How to Crush MSBA Program Vol 1.txt', now(), 'sean.jung@emory.edu', 'How to Crush MSBA Program Vol 1');

INSERT INTO downloads (filename, download_date, Users_email_address, Products_Product)
VALUES ('How to Keep Sanity Longer.mp4', now(), 'sean.jung@emory.edu', 'How to Cope with Stress 101');
```

2.1. Write a SELECT statement that joins the three tables and retrieves the data from these tables like this:



	email_address	first_name	last_name	download_date	filename	product_name
▶	johnsmith@gmail.com	John	Smith	2015-04-24 16:15:38	pedals_are_falling.mp3	Local Music Vol 1
	janedoe@yahoo.com	Jane	Doe	2015-04-24 16:15:38	turn_signal.mp3	Local Music Vol 1
	janedoe@yahoo.com	Jane	Doe	2015-04-24 16:15:38	one_horse_town.mp3	Local Music Vol 2

```
SELECT a.email_address, first_name, last_name, download_date, filename, product_name
FROM Users as a
LEFT JOIN Downloads as b
ON a.email_address = b.Users_email_address
LEFT JOIN Products as c
ON b.Products_Product = c.product_name;
```

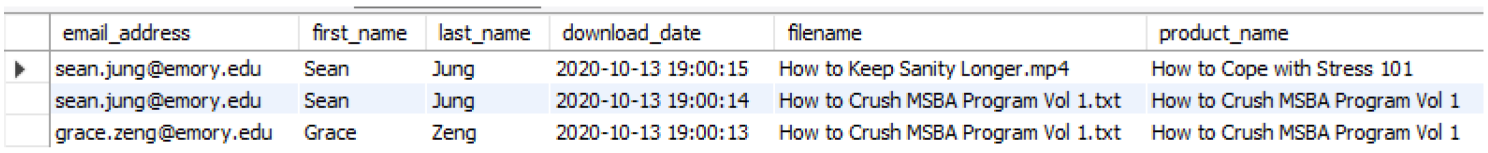


	email_address	first_name	last_name	download_date	filename	product_name
▶	grace.zeng@emory.edu	Grace	Zeng	2020-10-13 19:00:13	How to Crush MSBA Program Vol 1.txt	How to Crush MSBA Program Vol 1
	sean.jung@emory.edu	Sean	Jung	2020-10-13 19:00:14	How to Crush MSBA Program Vol 1.txt	How to Crush MSBA Program Vol 1
	sean.jung@emory.edu	Sean	Jung	2020-10-13 19:00:15	How to Keep Sanity Longer.mp4	How to Cope with Stress 101

Please notice how user 1 is related to product 1, user 2 is related to product 1, and user 2 is related to product 2

2.2. Sort the results by the email address in descending sequence and the product name in ascending sequence.

```
SELECT a.email_address, first_name, last_name, download_date, filename, product_name
FROM Users as a
LEFT JOIN Downloads as b
ON a.email_address = b.Users_email_address
LEFT JOIN Products as c
ON b.Products_Product = c.product_name
ORDER BY email_address DESC, product_name ASC;
```



	email_address	first_name	last_name	download_date	filename	product_name
▶	sean.jung@emory.edu	Sean	Jung	2020-10-13 19:00:15	How to Keep Sanity Longer.mp4	How to Cope with Stress 101
	sean.jung@emory.edu	Sean	Jung	2020-10-13 19:00:14	How to Crush MSBA Program Vol 1.txt	How to Crush MSBA Program Vol 1
	grace.zeng@emory.edu	Grace	Zeng	2020-10-13 19:00:13	How to Crush MSBA Program Vol 1.txt	How to Crush MSBA Program Vol 1

3. Use MySQL Workbench to open the EER diagram (for OLTP schema) that you created for code 1 submission 1 (retail store). Then, export a script that creates the database and save this script in a file named hw2-q4.sql. Next, use MySQL Workbench to open this file to create database my_retail_oltp_db. Report the script here.

```
 -- MySQL Script generated by MySQL Workbench

-- Tue Oct 13 19:14:02 2020

-- Model: New Model    Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET                                                                        @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_
ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema my_retail_oltp_db

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema my_retail_oltp_db

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `my_retail_oltp_db` DEFAULT CHARACTER SET latin1 ;

USE `my_retail_oltp_db` ;

-- -----------------------------------------------------

-- Table `my_retail_oltp_db`.`offices`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`offices` (

  `officeCode` VARCHAR(10) NOT NULL,

  `city` VARCHAR(50) NOT NULL,
  `phone` VARCHAR(50) NOT NULL,

  `addressLine1` VARCHAR(50) NOT NULL,

  `addressLine2` VARCHAR(50) NULL DEFAULT NULL,

  `state` VARCHAR(50) NULL DEFAULT NULL,

  `country` VARCHAR(50) NOT NULL,

  `postalCode` VARCHAR(15) NOT NULL,

  `territory` VARCHAR(10) NOT NULL,

  PRIMARY KEY (`officeCode`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- -----------------------------------------------------

-- Table `my_retail_oltp_db`.`employees`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`employees` (

  `employeeNumber` INT NOT NULL,

  `lastName` VARCHAR(50) NOT NULL,

  `firstName` VARCHAR(50) NOT NULL,

  `extension` VARCHAR(10) NOT NULL,

  `email` VARCHAR(100) NOT NULL,

  `officeCode` VARCHAR(10) NOT NULL,

  `reportsTo` INT NULL DEFAULT NULL,

  `jobTitle` VARCHAR(50) NOT NULL,

  PRIMARY KEY (`employeeNumber`),

  INDEX `reportsTo` (`reportsTo` ASC) VISIBLE,

  INDEX `officeCode` (`officeCode` ASC) VISIBLE,

  CONSTRAINT `employees_ibfk_1`

    FOREIGN KEY (`reportsTo`)
    REFERENCES `my_retail_oltp_db`.`employees` (`employeeNumber`),

  CONSTRAINT `employees_ibfk_2`

    FOREIGN KEY (`officeCode`)

    REFERENCES `my_retail_oltp_db`.`offices` (`officeCode`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- -----------------------------------------------------

-- Table `my_retail_oltp_db`.`customers`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`customers` (

  `customerNumber` INT NOT NULL,

  `customerName` VARCHAR(50) NOT NULL,

  `contactLastName` VARCHAR(50) NOT NULL,

  `contactFirstName` VARCHAR(50) NOT NULL,

  `phone` VARCHAR(50) NOT NULL,

  `addressLine1` VARCHAR(50) NOT NULL,

  `addressLine2` VARCHAR(50) NULL DEFAULT NULL,

  `city` VARCHAR(50) NOT NULL,

  `state` VARCHAR(50) NULL DEFAULT NULL,

  `postalCode` VARCHAR(15) NULL DEFAULT NULL,

  `country` VARCHAR(50) NOT NULL,

  `salesRepEmployeeNumber` INT NULL DEFAULT NULL,

  `creditLimit` DECIMAL(10,2) NULL DEFAULT NULL,

  PRIMARY KEY (`customerNumber`),

  INDEX `salesRepEmployeeNumber` (`salesRepEmployeeNumber` ASC) VISIBLE,

  CONSTRAINT `customers_ibfk_1`

    FOREIGN KEY (`salesRepEmployeeNumber`)
    REFERENCES `my_retail_oltp_db`.`employees` (`employeeNumber`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- -----------------------------------------------------

-- Table `my_retail_oltp_db`.`orders`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`orders` (

  `orderNumber` INT NOT NULL,

  `orderDate` DATE NOT NULL,

  `requiredDate` DATE NOT NULL,

  `shippedDate` DATE NULL DEFAULT NULL,

  `status` VARCHAR(15) NOT NULL,

  `comments` TEXT NULL DEFAULT NULL,

  `customerNumber` INT NOT NULL,

  PRIMARY KEY (`orderNumber`),

  INDEX `customerNumber` (`customerNumber` ASC) VISIBLE,

  CONSTRAINT `orders_ibfk_1`

    FOREIGN KEY (`customerNumber`)

    REFERENCES `my_retail_oltp_db`.`customers` (`customerNumber`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- -----------------------------------------------------

-- Table `my_retail_oltp_db`.`productlines`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`productlines` (

  `productLine` VARCHAR(50) NOT NULL,

  `textDescription` VARCHAR(4000) NULL DEFAULT NULL,
  `htmlDescription` MEDIUMTEXT NULL DEFAULT NULL,

  `image` MEDIUMBLOB NULL DEFAULT NULL,

  PRIMARY KEY (`productLine`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- -----------------------------------------------------

-- Table `my_retail_oltp_db`.`products`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`products` (

  `productCode` VARCHAR(15) NOT NULL,

  `productName` VARCHAR(70) NOT NULL,

  `productLine` VARCHAR(50) NOT NULL,

  `productScale` VARCHAR(10) NOT NULL,

  `productVendor` VARCHAR(50) NOT NULL,

  `productDescription` TEXT NOT NULL,

  `quantityInStock` SMALLINT NOT NULL,

  `buyPrice` DECIMAL(10,2) NOT NULL,

  `MSRP` DECIMAL(10,2) NOT NULL,

  PRIMARY KEY (`productCode`),

  INDEX `productLine` (`productLine` ASC) VISIBLE,

  CONSTRAINT `products_ibfk_1`

    FOREIGN KEY (`productLine`)

    REFERENCES `my_retail_oltp_db`.`productlines` (`productLine`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;
```

```
-- -----------------------------------------------------
-- Table `my_retail_oltp_db`.`orderdetails`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`orderdetails` (
  `orderNumber` INT NOT NULL,
  `productCode` VARCHAR(15) NOT NULL,
  `quantityOrdered` INT NOT NULL,
  `priceEach` DECIMAL(10,2) NOT NULL,
  `orderLineNumber` SMALLINT NOT NULL,
  PRIMARY KEY (`orderNumber`, `productCode`),
  INDEX `productCode` (`productCode` ASC) VISIBLE,
  CONSTRAINT `orderdetails_ibfk_1`
    FOREIGN KEY (`orderNumber`)
    REFERENCES `my_retail_oltp_db`.`orders` (`orderNumber`),
  CONSTRAINT `orderdetails_ibfk_2`
    FOREIGN KEY (`productCode`)
    REFERENCES `my_retail_oltp_db`.`products` (`productCode`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-- -----------------------------------------------------
-- Table `my_retail_oltp_db`.`payments`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`payments` (
  `customerNumber` INT NOT NULL,
  `checkNumber` VARCHAR(50) NOT NULL,
  `paymentDate` DATE NOT NULL,
  `amount` DECIMAL(10,2) NOT NULL,
  PRIMARY KEY (`customerNumber`, `checkNumber`),

  CONSTRAINT `payments_ibfk_1`

    FOREIGN KEY (`customerNumber`)

    REFERENCES `my_retail_oltp_db`.`customers` (`customerNumber`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

3.1. Starting with this my_retail_oltp_db, write a SQL query that creates the fact table that you created in your OLAP model. Report your query here.

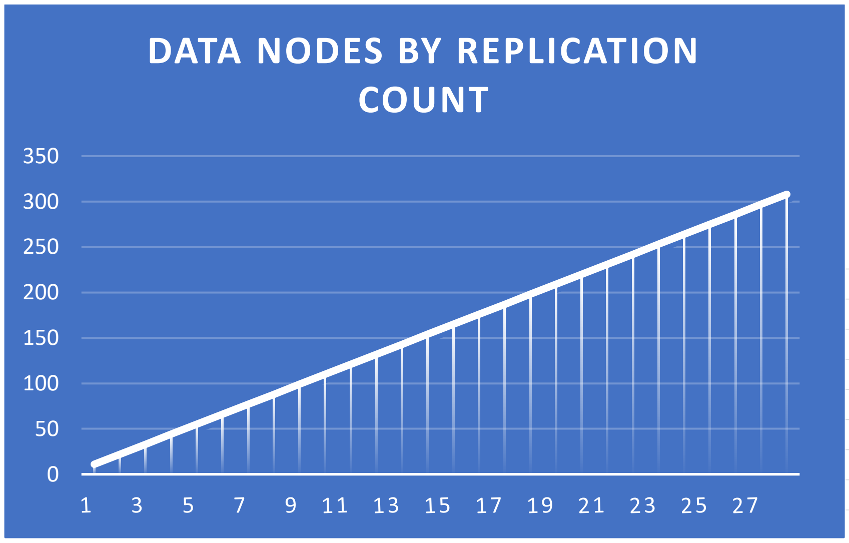
```
##### Create FACT Table

CREATE TABLE IF NOT EXISTS `my_retail_oltp_db`.`Retail_Fact_Table` (

  `Surr_Fact` VARCHAR(10) NOT NULL,

  `offices` VARCHAR(50) NOT NULL,

  `employees` VARCHAR(50) NOT NULL,

  `customers` VARCHAR(50) NOT NULL,

  `orders` VARCHAR(50) NULL DEFAULT NULL,

  `productlines` VARCHAR(50) NULL DEFAULT NULL,

  `products` VARCHAR(50) NOT NULL,

  `orderdetails` VARCHAR(15) NOT NULL,

  `payments` VARCHAR(10) NOT NULL,

  PRIMARY KEY (`Surr_Fact`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;
```

The following two questions are based on the NoSQL and Hadoop content of course.

4. Assuming a company is planning to migrate their existing file servers and databases (500TB) to HDFS platform where each data node is of size 64TB. The company believes they will be utilizing 25% of data node storage for intermediate tasks.

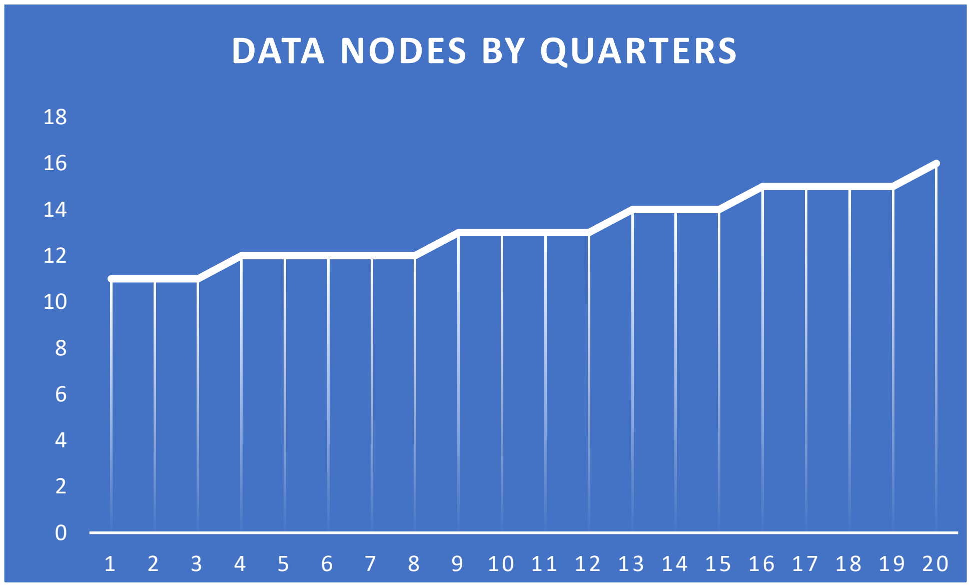
Question 4 is saying there's 500 TB of data to store on nodes that are 64 TB, but each node can only hold 75% of data. So, technically each node can only hold 48 TB. So, 500/48 = 10.46 (or 11 to round up) will be the number of nodes you'll need for 1 replication

4.1. Plot a chart showing the number of data nodes needed (Y-axis) based data replication count (X-axis).



Replication	Data Nodes
1	11
2	22
3	33
4	44
5	55
6	66
7	77
8	88
9	99

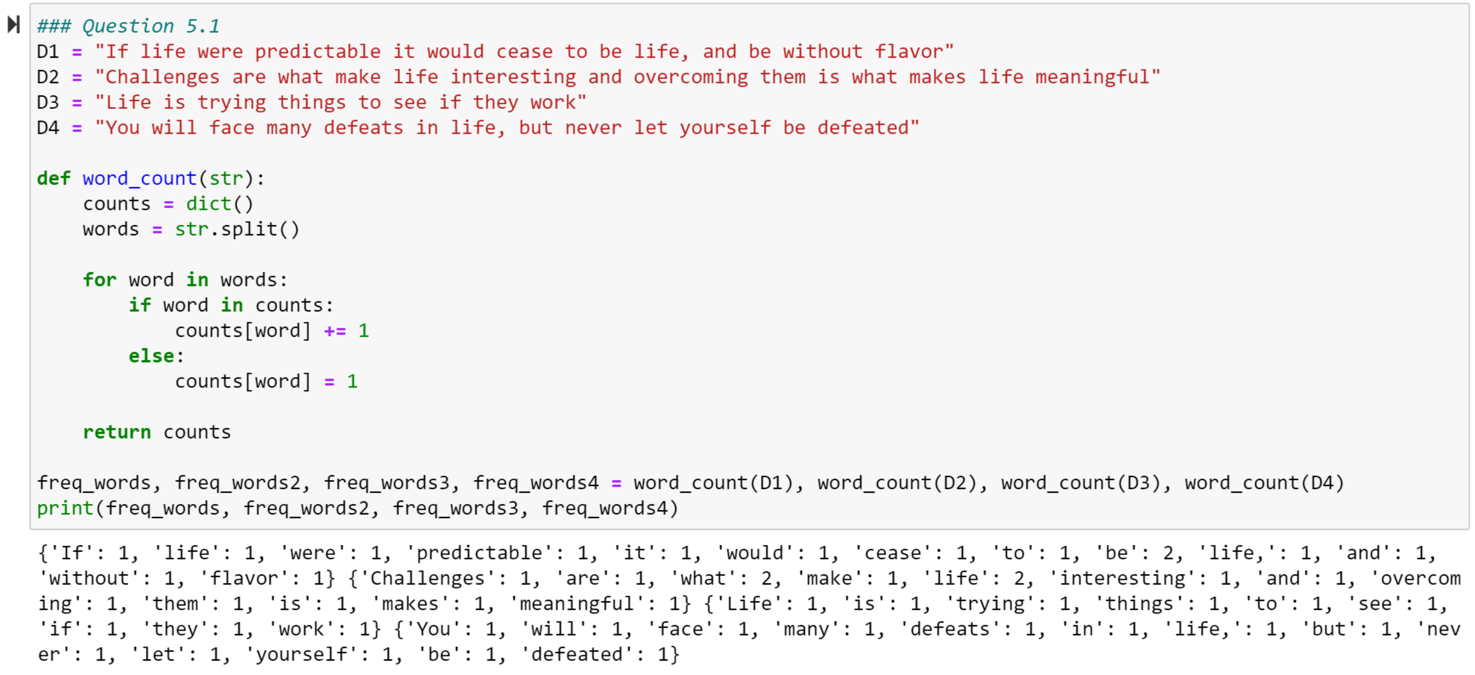
4.2. For a selected data replication count, plot a chart for change in data node need every quarter for next 20 quarters - assuming their data increases by 2% every quarter.



5. Assuming you have following 4 data items on a data node.

NodeID	DocumentID	Data
N1	D1	If life were predictable it would cease to be life, and be without flavor
N1	D2	Challenges are what make life interesting and overcoming them is what makes life meaningful
N1	D3	Life is trying things to see if they work
N1	D4	You will face many defeats in life, but never let yourself be defeated

5.1. Write and submit a mapper code (in Python or pseudocode) that creates a sorted dictionary of each word (key) and frequency (value) in each document.



```python
### Question 5.1
D1 = "If life were predictable it would cease to be life, and be without flavor"
D2 = "Challenges are what make life interesting and overcoming them is what makes life meaningful"
D3 = "Life is trying things to see if they work"
D4 = "You will face many defeats in life, but never let yourself be defeated"

def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts

freq_words, freq_words2, freq_words3, freq_words4 = word_count(D1), word_count(D2), word_count(D3), word_count(D4)
print(freq_words, freq_words2, freq_words3, freq_words4)
```

```
{'If': 1, 'life': 1, 'were': 1, 'predictable': 1, 'it': 1, 'would': 1, 'cease': 1, 'to': 1, 'be': 2, 'life,': 1, 'and': 1,
'without': 1, 'flavor': 1} {'Challenges': 1, 'are': 1, 'what': 2, 'make': 1, 'life': 2, 'interesting': 1, 'and': 1, 'overcom
ing': 1, 'them': 1, 'is': 1, 'makes': 1, 'meaningful': 1} {'Life': 1, 'is': 1, 'trying': 1, 'things': 1, 'to': 1, 'see': 1,
'if': 1, 'they': 1, 'work': 1} {'You': 1, 'will': 1, 'face': 1, 'many': 1, 'defeats': 1, 'in': 1, 'life,': 1, 'but': 1, 'nev
er': 1, 'let': 1, 'yourself': 1, 'be': 1, 'defeated': 1}
```

5.2. Write and submit a reducer code (in Python or pseudocode) that takes the mapped dictionaries and gives the TOTAL frequency of ALL words.

```
### Question 5.2
wordstring1 = "If life were predictable it would cease to be life and be without flavor "
wordstring2 = "Challenges are what make life interesting and overcoming them is what makes life meaningful "
wordstring3 = "Life is trying things to see if they work "
wordstring4 = "You will face many defeats in life but never let yourself be defeated"
wordlist = wordstring1 + wordstring2 + wordstring3 + wordstring4

# Function to convert (From List to String)
def listToString(s):

    # initialize an empty string
    str1 = ""

    # traverse in the string
    for ele in s:
        str1 += ele

    # return string
    return str1

s = wordlist
wordlist = listToString(s)

# Convert to upper case for consistency in word frequency search
WORDLIST = wordlist.upper()

def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts

freq_words = word_count(WORDLIST)
print(freq_words)
```

```
{'IF': 2, 'LIFE': 6, 'WERE': 1, 'PREDICTABLE': 1, 'IT': 1, 'WOULD': 1, 'CEASE': 1, 'TO': 2, 'BE': 3, 'AND': 2, 'WITHOUT': 1,
'FLAVOR': 1, 'CHALLENGES': 1, 'ARE': 1, 'WHAT': 2, 'MAKE': 1, 'INTERESTING': 1, 'OVERCOMING': 1, 'THEM': 1, 'IS': 2, 'MAKE
S': 1, 'MEANINGFUL': 1, 'TRYING': 1, 'THINGS': 1, 'SEE': 1, 'THEY': 1, 'WORK': 1, 'YOU': 1, 'WILL': 1, 'FACE': 1, 'MANY': 1,
'DEFEATS': 1, 'IN': 1, 'BUT': 1, 'NEVER': 1, 'LET': 1, 'YOURSELF': 1, 'DEFEATED': 1}
```

```
5.3. Write and submit a reducer code (in Python or pseudocode) that takes the mapped
     dictionaries and gives the AVERAGE frequency of word “life” across all documents.
```

```python
# Queston 5.3

# Python code to find frequency of each word
def freq(str):

    # break the string into list of words
    str = str.split()
    str2 = []

    # loop till string values present in list str
    for i in str:

        # checking for the duplicacy
        if i not in str2:

            # insert value in str2
            str2.append(i)

    for i in range(0, len(str2)):

        # count the frequency of each word(present
        # in str2) in str and print
        print('Frequency of', str2[i], 'is :', str.count(str2[i]))

def main():
    str = WORDLIST
    freq(str)

if __name__=="__main__":
    main()              # call main function
```

```
Frequency of IF is : 2
Frequency of LIFE is : 6
```