# ISOM 671: Managing Big Data (Assignment 2)

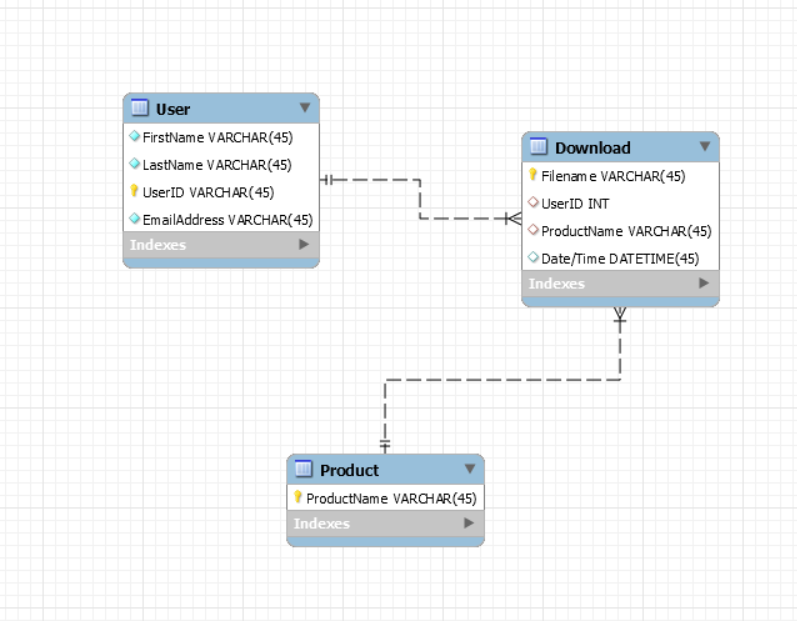
Name

Email

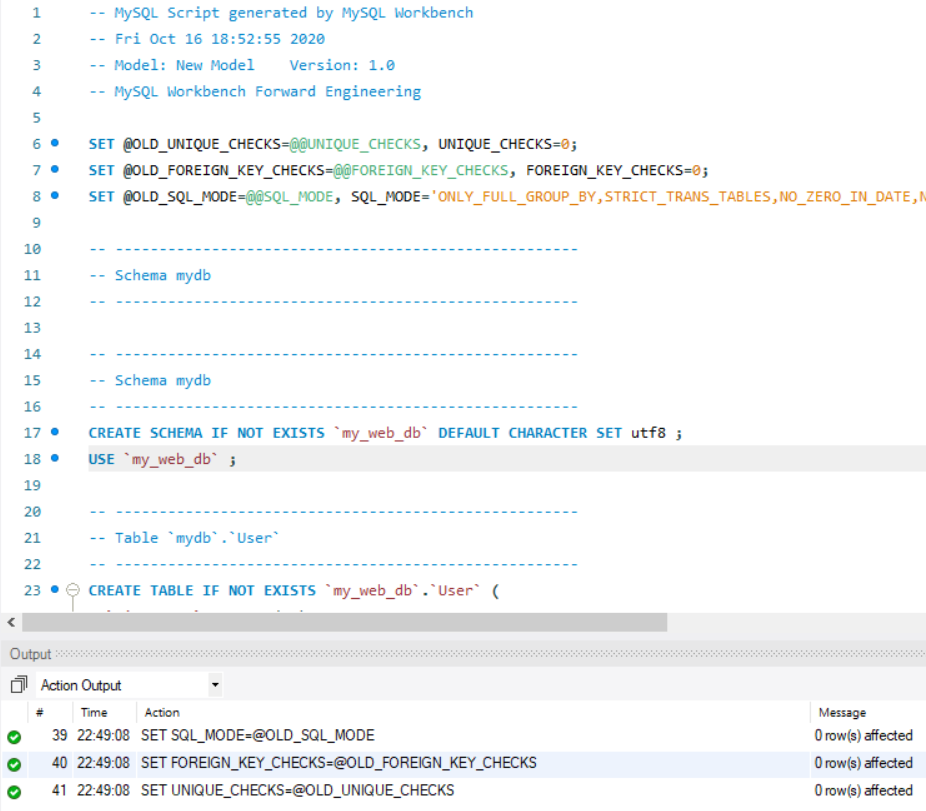
***There are 5 numbered questions (1-point each).* Please submit your assignment as a single PDF or Word file by uploading it to course canvas page. *You should provide: SQL statements, results of any SQL statement (typically copy first 10 rows), and answers to questions, if any.***

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. 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

-- Fri Oct 16 18:52:55 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 mydb

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

CREATE SCHEMA IF NOT EXISTS `my\_web\_db` DEFAULT CHARACTER SET utf8 ;

USE `my\_web\_db` ;

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

-- Table `mydb`.`User`

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

CREATE TABLE IF NOT EXISTS `my\_web\_db`.`User` (

`FirstName` VARCHAR(45) NOT NULL,

`LastName` VARCHAR(45) NOT NULL,

`UserID` INT NOT NULL,

`EmailAddress` VARCHAR(45) NOT NULL,

PRIMARY KEY (`UserID`),

UNIQUE INDEX `UserID\_UNIQUE` (`UserID` ASC) VISIBLE)

ENGINE = InnoDB;

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

-- Table `mydb`.`Product`

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

CREATE TABLE IF NOT EXISTS `my\_web\_db`.`Product` (

`ProductName` VARCHAR(45),

`ProductID` INT NOT NULL,

PRIMARY KEY (`ProductID`))

ENGINE = InnoDB;

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

-- Table `mydb`.`Download`

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

CREATE TABLE IF NOT EXISTS `my\_web\_db`.`Download` (

`Filename` VARCHAR(45) NOT NULL,

`UserID` INT NOT NULL,

`ProductID` INT NOT NULL,

`Date/Time` DATETIME(0) NOT NULL,

PRIMARY KEY (`Filename`),

INDEX `fk\_Download\_Product\_idx` (`ProductID` ASC) VISIBLE,

INDEX `fk\_Download\_User1\_idx` (`UserID` ASC) VISIBLE,

CONSTRAINT `fk\_Download\_Product`

FOREIGN KEY (`ProductID`)

REFERENCES `my\_web\_db`.`Product` (`ProductID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Download\_User1`

FOREIGN KEY (`UserID`)

REFERENCES `my\_web\_db`.`User` (`UserID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

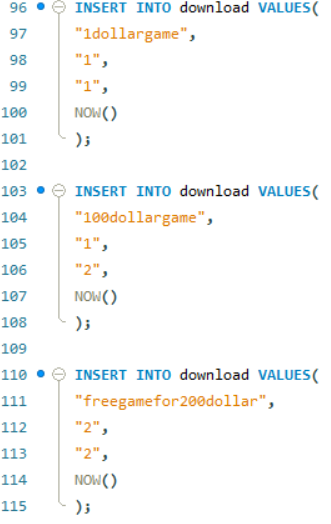
SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

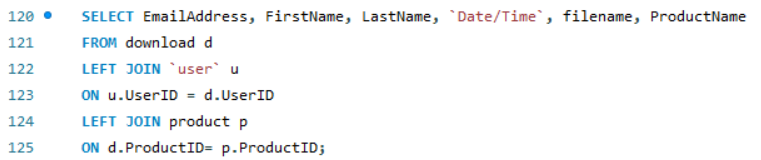
SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

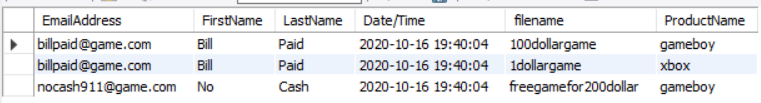
1. Run the script you created in question 1 to create the database under the name my\_web\_db. Write a script that adds rows to the database. In particular,

* 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.

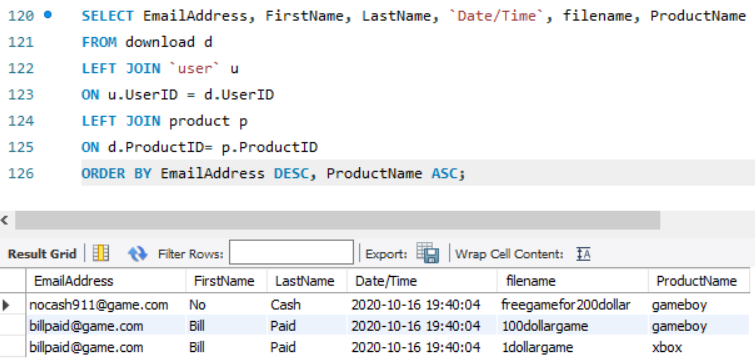
 

* 1. Write a SELECT statement that joins the three tables and retrieves the data from these tables like this:

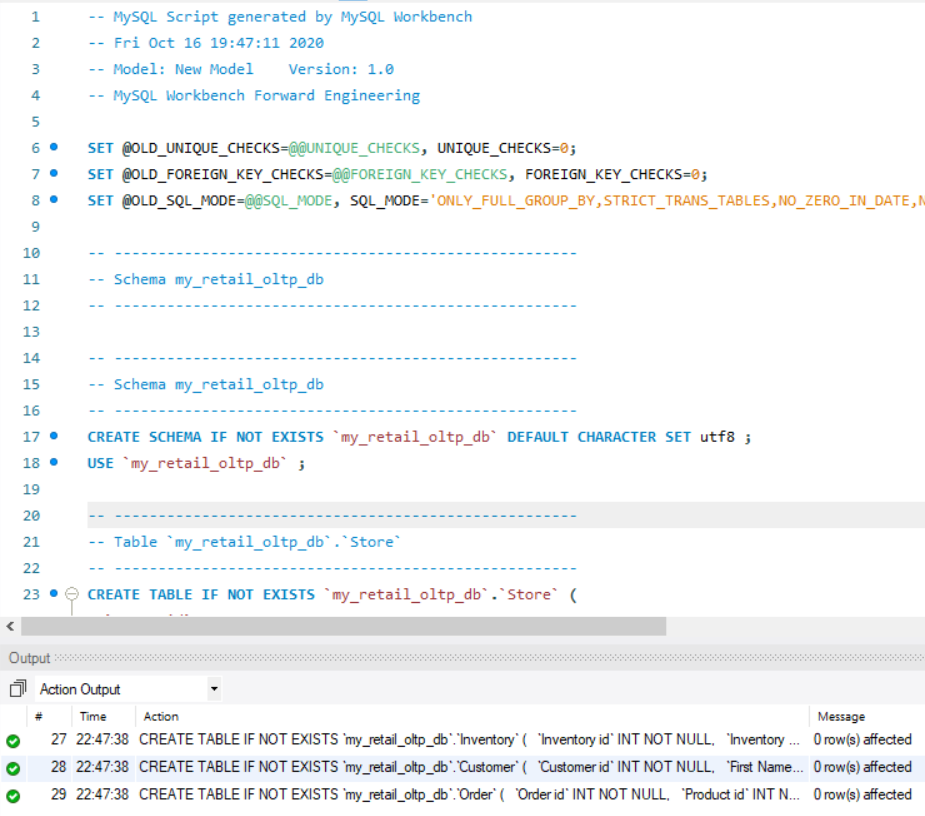




* 1. Sort the results by the email address in descending sequence and the product name in ascending sequence.



1. 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

-- Fri Oct 16 19:47:11 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 my\_retail\_oltp\_db

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

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

-- Schema my\_retail\_oltp\_db

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

CREATE SCHEMA IF NOT EXISTS `my\_retail\_oltp\_db` DEFAULT CHARACTER SET utf8 ;

USE `my\_retail\_oltp\_db` ;

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

-- Table `my\_retail\_oltp\_db`.`Store`

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

CREATE TABLE IF NOT EXISTS `my\_retail\_oltp\_db`.`Store` (

`Store id` INT NOT NULL,

`Store name` VARCHAR(45) NOT NULL,

`Operation Status` BINARY(10) NOT NULL,

`Phone Number` VARCHAR(10) NOT NULL,

`Address` VARCHAR(45) NULL,

`Email` VARCHAR(45) NULL,

PRIMARY KEY (`Store id`),

UNIQUE INDEX `Employee id\_UNIQUE` (`Store id` ASC) VISIBLE)

ENGINE = InnoDB;

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

-- Table `my\_retail\_oltp\_db`.`Department`

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

CREATE TABLE IF NOT EXISTS `my\_retail\_oltp\_db`.`Department` (

`Department id` INT NOT NULL,

`Department name` VARCHAR(45) NOT NULL,

`Store id` INT NOT NULL,

`Phone Number` VARCHAR(10) NOT NULL,

`Email` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Department id`),

UNIQUE INDEX `Department id\_UNIQUE` (`Department id` ASC) VISIBLE,

INDEX `fk\_Department\_Store1\_idx` (`Store id` ASC) VISIBLE,

CONSTRAINT `fk\_Department\_Store1`

FOREIGN KEY (`Store id`)

REFERENCES `my\_retail\_oltp\_db`.`Store` (`Store id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `my\_retail\_oltp\_db`.`Employee`

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

CREATE TABLE IF NOT EXISTS `my\_retail\_oltp\_db`.`Employee` (

`Employee id` INT NOT NULL,

`First Name` VARCHAR(45) NOT NULL,

`Last Name` VARCHAR(45) NOT NULL,

`Department id` INT NOT NULL,

`Salary` DECIMAL(2) NULL,

`Phone Number` VARCHAR(10) NOT NULL,

`Address` VARCHAR(45) NULL,

`Email` VARCHAR(45) NULL,

`Position` VARCHAR(45) NULL,

UNIQUE INDEX `Employee id\_UNIQUE` (`Employee id` ASC) VISIBLE,

PRIMARY KEY (`Employee id`),

INDEX `fk\_Employee\_Department\_idx` (`Department id` ASC) VISIBLE,

CONSTRAINT `fk\_Employee\_Department`

FOREIGN KEY (`Department id`)

REFERENCES `my\_retail\_oltp\_db`.`Department` (`Department id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `my\_retail\_oltp\_db`.`Product`

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

CREATE TABLE IF NOT EXISTS `my\_retail\_oltp\_db`.`Product` (

`Product id` INT NOT NULL,

`Product Name` VARCHAR(45) NULL,

`Description` VARCHAR(45) NULL,

`Price` VARCHAR(45) NULL,

`Category` VARCHAR(45) NULL,

UNIQUE INDEX `Product id\_UNIQUE` (`Product id` ASC) VISIBLE,

PRIMARY KEY (`Product id`))

ENGINE = InnoDB;

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

-- Table `my\_retail\_oltp\_db`.`Inventory`

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

CREATE TABLE IF NOT EXISTS `my\_retail\_oltp\_db`.`Inventory` (

`Inventory id` INT NOT NULL,

`Inventory Name` VARCHAR(45) NULL,

`Store id` INT NULL,

`Quantity` INT NULL,

`Product id` INT NOT NULL,

PRIMARY KEY (`Inventory id`),

INDEX `fk\_Inventory\_Store1\_idx` (`Store id` ASC) VISIBLE,

INDEX `fk\_Inventory\_Product1\_idx` (`Product id` ASC) VISIBLE,

CONSTRAINT `fk\_Inventory\_Store1`

FOREIGN KEY (`Store id`)

REFERENCES `my\_retail\_oltp\_db`.`Store` (`Store id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Inventory\_Product1`

FOREIGN KEY (`Product id`)

REFERENCES `my\_retail\_oltp\_db`.`Product` (`Product id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `my\_retail\_oltp\_db`.`Customer`

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

CREATE TABLE IF NOT EXISTS `my\_retail\_oltp\_db`.`Customer` (

`Customer id` INT NOT NULL,

`First Name` VARCHAR(45) NOT NULL,

`Last Name` VARCHAR(45) NOT NULL,

`Phone Number` VARCHAR(10) NOT NULL,

`Address` VARCHAR(45) NULL,

`Email` VARCHAR(45) NULL,

UNIQUE INDEX `Employee id\_UNIQUE` (`Customer id` ASC) VISIBLE,

PRIMARY KEY (`Customer id`))

ENGINE = InnoDB;

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

-- Table `my\_retail\_oltp\_db`.`Order`

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

CREATE TABLE IF NOT EXISTS `my\_retail\_oltp\_db`.`Order` (

`Order id` INT NOT NULL,

`Product id` INT NULL,

`Quantity` INT NULL,

`Customer Review` INT NULL,

`Employee id` INT NULL,

`Time` DATETIME(0) NULL,

`Customer id` INT NULL,

`Transaction Info` VARCHAR(45) NULL,

PRIMARY KEY (`Order id`),

INDEX `fk\_Order\_Employee1\_idx` (`Employee id` ASC) VISIBLE,

INDEX `fk\_Order\_Product1\_idx` (`Product id` ASC) VISIBLE,

INDEX `fk\_Order\_Customer1\_idx` (`Customer id` ASC) VISIBLE,

CONSTRAINT `fk\_Order\_Employee1`

FOREIGN KEY (`Employee id`)

REFERENCES `my\_retail\_oltp\_db`.`Employee` (`Employee id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Order\_Product1`

FOREIGN KEY (`Product id`)

REFERENCES `my\_retail\_oltp\_db`.`Product` (`Product id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Order\_Customer1`

FOREIGN KEY (`Customer id`)

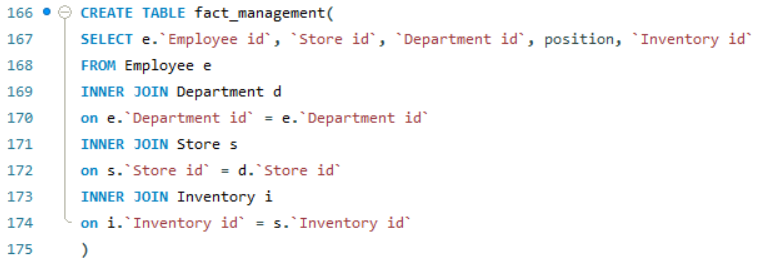
REFERENCES `my\_retail\_oltp\_db`.`Customer` (`Customer id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

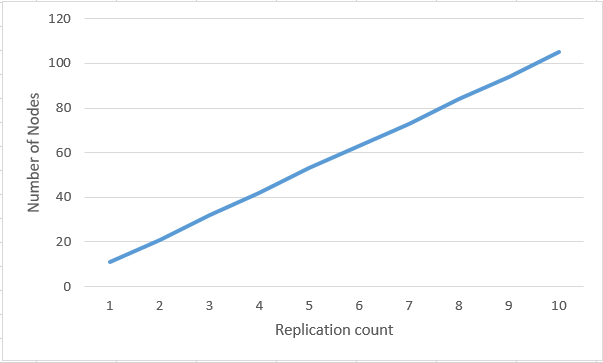
ENGINE = InnoDB;

* 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.

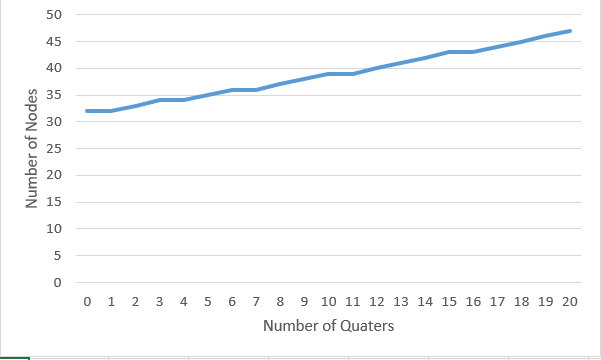


**The following two questions are based on the NoSQL and Hadoop content of course.**

1. Assuming a company is planning to migrate their existing file servers and databases (500TB) to HDFS platform where each datanode if of size 64TB. The company believes they will be utilizing 25% of datanode storage for intermediate tasks.
   1. Plot a chart showing the number of datanodes needed (Y-axis) based data replication count (X-axis).

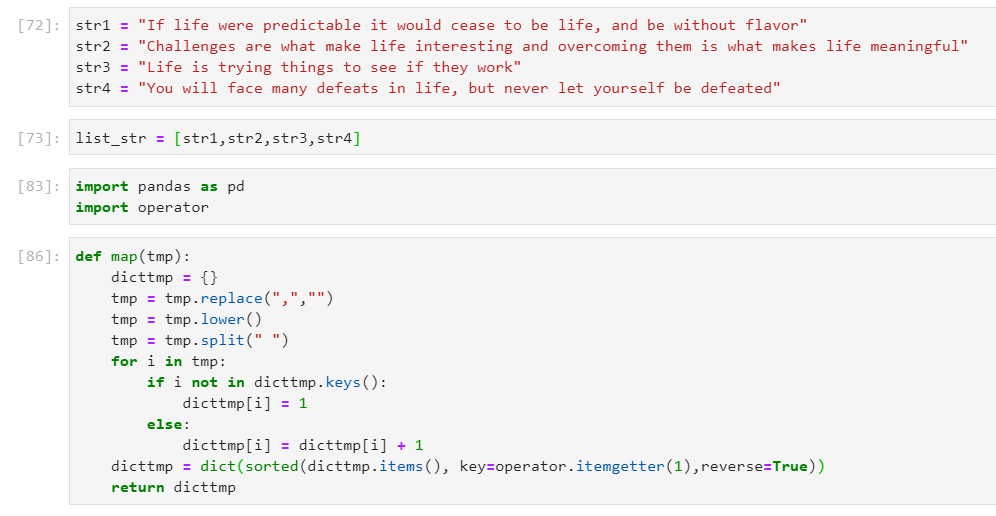


* 1. For a selected data replication count, plot a chart for change in datanode need every quarter for next 20 quarters - assuming their data increases by 2% every quarter.



1. Assuming you have following 4 data items on a datanode.

|  |  |  |
| --- | --- | --- |
| 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 |

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

