School of Computing and Information Technology
University of Wollongong

Session: **Spring 2024**
Lecturer: **Janusz R. Getta**

ISIT312 Big Data Management
Assignment 3

Published on 23 September 2024

Scope

This assignment includes the tasks related to querying a data cube, design and implementation of HBase table, querying and manipulating data in HBase table, data processing with Pig, and data processing with Spark.

This assignment is due on **Saturday, 26 October 2024, 7:00pm** (sharp).

This assignment is worth **20%** of the total evaluation in the subject.

The assignment consists of 4 tasks and specification of each task starts from a new page.

Only electronic submission through Moodle at:
`https://moodle.uowplatform.edu.au/login/index.php`
will be accepted. A submission procedure is explained at the end of Assignment 1 specification.

A policy regarding late submissions is included in the subject outline.

Only one submission of Assignment 3 is allowed and only one submission per student is accepted.

A submission marked by Moodle as "late" is always treated as a late submission no matter how many seconds it is late.

A submission that contains an incorrect file attached is treated as a correct submission with all consequences coming from the evaluation of the file attached.

All files left on Moodle in a state "`Draft(not submitted)`" will not be evaluated.

A submission of compressed files (zipped, gzipped, rared, tared, 7-zipped, lhzed, … etc) is not allowed. The compressed files will not be evaluated.

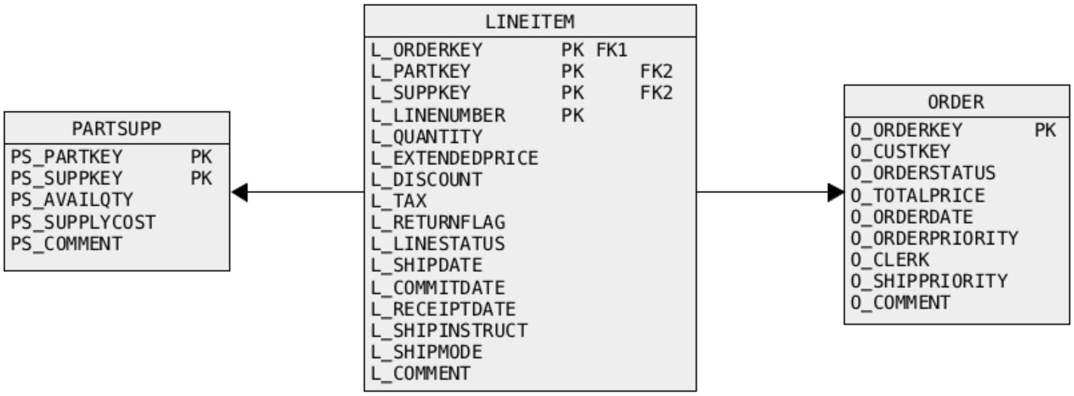
An implementation that does not compile well due to one or more syntactical and/or run time errors scores no marks.

Using any sort of Generative Artificial Intelligence (GenAI) for this assignment is NOT allowed !

It is expected that all tasks included within **Assignment 3** will be solved **individually without any cooperation** with the other students. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or office hours. Plagiarism will result in a **FAIL** grade being recorded for the assessment task.

Task 1 (5 marks)
Querying a data cube

Consider the following logical schema implementing a two-dimensional data cube.



The original data cube consists of a fact entity that represents the parts included in the orders, a dimension of orders and a dimension of parts supplied by suppliers.

The relational tables `PARTSUPP` and `ORDERS` implement the dimensions of parts supplied by suppliers and orders. A relational table `LINEITEM` implements a fact entity of a data cube.

Download a file `task1.zip` and unzip the file. You should obtain a folder `task1` with the following files: `dbcreate.hql`, `dbdrop.hql`, `partsupp.tbl`, `lineitem.tbl`, and `orders.tbl`.

A file `orders.tbl` contains information about the orders submitted by the customers. A file `lineitem.tbl` contains information about the parts included in the orders. A file `partsupp.tbl` contains information about the parts and suppliers of parts included in the orders.

Open Terminal window and use `cd` command to navigate to a folder that contains the files `dbcreate.hql`, `dbdrop.hql`, `partsupp.tbl`, `lineitem.tbl`, and `orders.tbl`.

Start Hive Server 2 in the terminal window (remember to start Hadoop and Metastore first). Then start `beeline` client.

When ready process a script file `dbcreate.hql` to create the internal relational tables and to load data into the tables. You can use either `beeline` or `Zeppelin`. A script `dbdrop.hql` can be used to drop the tables.

(1) 0.5 mark
Implement the following query using `GROUP BY` clause with `CUBE` operator.

For the order clerks (O_CLERK) Clerk#000000988, Clerk#000000992, find the total number of ordered parts per customer (O_CUSTKEY), per supplier (L_SUPPKEY), per customer and supplier (O_CUSTKEY, L_SUPPKEY), and the total number of ordered parts.

(2) 0.5 mark
Implement the following query using GROUP BY clause with ROLLUP operator.

For the parts with the keys (L_PARTKEY) 18, 19,20 find the largest discount applied (L_DISCOUNT) per part key (L_PARTKEY) and per part key and supplier key (L_PARTKEY, L_SUPPKEY) and the largest discount applied at all.

(3) 1 mark
Implement the following query using GROUP BY clause with GROUPING SETS operator.

Find the smallest price (L_EXTENDEDPRICE) per order year (O_ORDERDATE), and order clerk (O_CLERK).

Implement the following SQL queries as SELECT statements using window partitioning technique.

(4) 1 mark
For each part list its key (PS_PARTKEY), all its available quantities (PS_AVAILQTY), the smallest available quantity, and the average available quantity. Consider only the parts with the keys 18, 19 and 20.

(5) 1 mark
For each part list its key (PS_PARTKEY) and all its available quantities (PS_AVAILQTY) sorted in descending order and a rank (position number in an ascending order) of each quantity. Consider only the parts with the keys 18, 19 and 20. Use an analytic function ROW_NUMBER().

(6) 1 mark
For each part list its key (PS_PARTKEY), its available quantity, and an average available quantity (PS_AVAILQTY) of the current quantity and all previous quantities in the ascending order of available quantities. Consider only the parts with the keys 18, 19 and 20. Use ROWS UNBOUNDED PRECEEDING sub-clause within PARTITION BY clause.

When ready, save your SELECT statements in a file solution1.hql. Then, process a script file solution1.hql and save the results in a report solution1.txt.

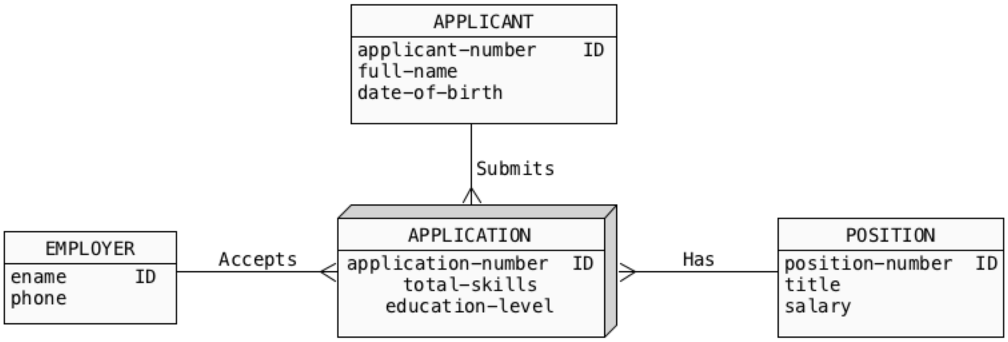
Deliverables

A file `solution1.txt` that contains a report from processing of `SELECT` statements implementing the queries listed above.

Task 2 (5 marks)

Design and implementation of HBase table (3 marks)

(1) Consider a conceptual schema of sample data cube given below. The schema represents a data cube where applicants submit applications for positions offered by employers.



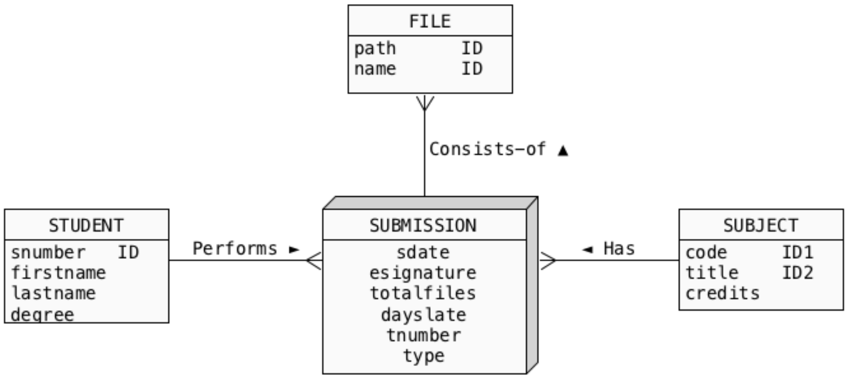
Design a single HBase table a database that contains information described by a conceptual schema given above.

Create HBase script `solution2-1.hb` with HBase shell commands that create HBase table and load sample data into the table. Load into the table information about at least two applicants, two positions, two employers and three applications.

When ready use HBase shell to process a script file `solution2-1.hb` and to save a report from processing in a file `solution2-1.txt`.

Querying HBase table (2 marks)

(2) Consider a conceptual schema given below. The schema represents a data cube where students submit assignments and each submission consists of several files and it is related to one subject.



Download a file `task2-2.hb` with HBase shell commands. Process a script `task2-2.hb`. Processing of the script creates HBase table `task2-2` and loads some data into it.

Use HBase shell to implement the queries and data manipulations listed below. Implementation of each step is worth 0.4 of a mark.

Save the queries and data manipulations in a file `solution2-2.hb`. Note that implementation of the queries and data manipulations listed below may need more than one command of HBase shell.

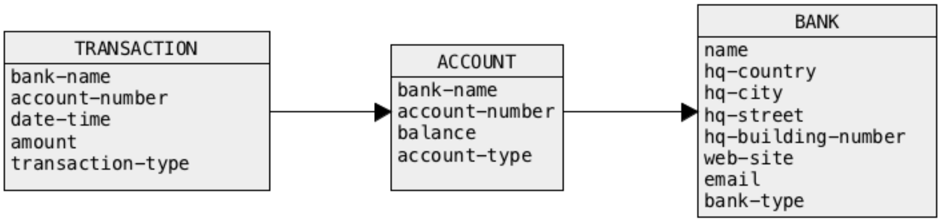
(i) Find all information included in a column family `SUBJECT` qualified by `code` and column family `FILES` qualified by `fnumber1` and `fnumber2`.
(ii) Find all information about a subject that has a code `312`, list two versions per cell.
(iii) Find all information about a submission of assignment `1` performed by a student `007` in a subject `312`, list one version per cell.
(iv) Replace a submission date of assignment `1` performed by a student `007` in a subject `312` with a date `02-APR-2019` and then list a column family `SUBMISSION` to verify the results.
(v) Add a column family `DEGREE` that contains information about titles of degrees enrolled by the students. Assume that a student can enrol only one degree. Then add information about a title of degree enrolled by a student with a number `007`. A degree title is up to you. List all information about a student with a number `007`.

When ready, start HBase shell and process a script file `solution2-2.hb` with the Hbase shell commands. Save report from processing of the script in a file `solution2-2.txt`.

Deliverables
A file `solution2-1.txt` with a listing from processing of a script file `solution2-1.hb`.
A file `solution2-2.txt` with a listing from processing of a script file `solution2-2.hb`.

Task 3 (5 marks)
Data processing with Pig Latin

Consider the following logical schema of one-dimensional data cube.



Download a file `task3.zip` published on Moodle together with a specification of Assignment 3 and unzip it. You should obtain a folder `task3` with the following files: `bank.tbl`, `account.tbl`, and `transaction.tbl`.

Use a text editor to examine the contents of the files.

Upload the files into HDFS.

Open `Terminal` window and start `pig` command line interface to Pig. Use `pig` command line interface to implement the following actions. Implementation of each step is worth 1 mark.

(1) Use `load` command to load the files `bank.tbl`, `account.tbl`, and `transaction.tbl` from HDFS into a Pig storage.

Implement and process the following queries.

(2) Find the names (`name`) of banks (`bank-type`) that have headquarters located in Japan (`hq-country`).

(3) Find the account numbers (`account-number`) opened in any construction (`bank-type`) bank.

(4) Find the names of banks (`bank-name`), that have no accounts opened in the banks.

(5) Find the total number of accounts opened in each bank located in Japan (`hq-country`).

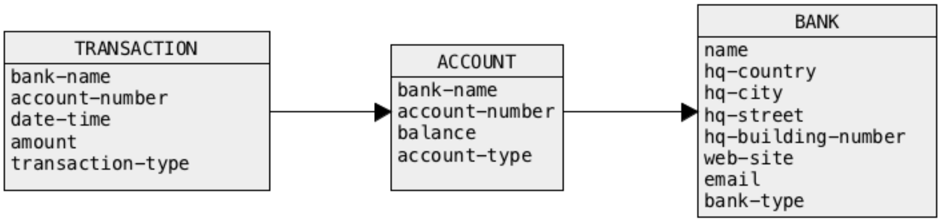
When ready, Copy into a clipboard the contents of `Terminal` window with the data loadings and queries processed above and the messages and results listed in the window and Paste the results from a clipboard into a text file `solution3.txt`.

Deliverables

A file `solution3.txt` that contains a listing of data loadings and queries performed above, the messages and results of operations. A file `solution3.txt` must be created through Copy/Paste of the contents of `Terminal` window into a file `solution3.txt`. No screen dumps are allowed and no screen dumps will be evaluated.

Task 4 (5 marks)
Data processing with Spark

Consider the following denormalized logical schema of one-dimensional data cube.



Download a file `task4.zip` published on Moodle together with a specification of Assignment 3 and unzip it. You should obtain a folder `task4` with the following files: `bank.csv`, `account.csv`, and `transaction.csv`.

Use a text editor to examine the contents of the files.

Upload the files `bank.csv`, `account.csv`, and `transaction.csv` to HDFS.

Open `Terminal` window and start `pyspark` command line interface to Spark. Use `pyspark` command line interface to implement the following actions. Implementation of each action is worth 1 mark.

(1) Create the schemas for the files `bank.csv`, `account.csv`, and `transaction.csv`.

(2) Create the data frames with the contents of the files `bank.csv`, `account.csv`, and `transaction.csv` using the schemas created in the previous step.

Count the total number of rows in each frame and then list the contents of each frame.

(3) Implement the following query on a data frame that contains information about the transactions.

*Find the total amount of money involved in the deposit transactions per each bank. You can skip the banks that had no deposit transaction*s. *Sort the results in the ascending order of the total amount of money found.*

(4) Create a temporary view over a data frame with information about the transactions.

(5) Implement the following query on a temporary view that contains information about the transactions.

> *Find the total amount of money involved in the deposit transactions per each bank. You can skip the banks that had no deposit transaction*s. *Sort the results in the ascending order of the total amount of money found.*

When ready, Copy into a clipboard the contents of `Terminal` window with the operations processed above and the results listed in the window and Paste the results from a clipboard into a text file `solution4.txt`.

Deliverables

A file `solution4.txt` that contains a listing of operations performed above and the results of operations. A file `solution4.txt` must be created through Copy/Paste of the contents of `Terminal` window into a file `solution4.txt`. No screen dumps are allowed and no screen dumps will be evaluated.

Submission of Assignment 3

Note, that you have only one submission. So, make it absolutely sure that you submit the correct files with the correct contents. No other submission is possible !

Submit the files `solution1.txt`, `solution2-1.txt`, `solution2-2.txt`, `solution3.txt`, and `solution4.txt` through Moodle in the following way:

(1) Access Moodle at **`http://moodle.uowplatform.edu.au/`**
(2) To login use a **`Login`** link located in the right upper corner the Web page or in the middle of the bottom of the Web page
(3) When logged select a site **`ISIT912/312 (S224) Big Data Management`**
(4) Scroll down to a section **`Assessment items (Assignments)`**
(5) Click at **`In this place you can submit the outcomes of your work on the tasks included in Assignment 3 for ISIT312 students`** link.
(6) Click at a button **`Add Submission`**
(7) Move a file **`solution1.txt`** into an area **`File submissions`**. You can also use a link **`Add`**...
(8) Repeat a step (7) for the files **`solution2-1.txt`**, **`solution2-2.txt`**, **`solution3.txt`**, and **`solution4.txt`**.
(9) Click at a button **`Submit assignment.`**
(10) Click at the checkbox with a text attached: **`By checking this box, I confirm that this submission is my own work, I accept responsibility for any copyright infringement that may occur as a result of this submission, and I acknowledge that this submission may be forwarded to a text-matching service`**.
(11) Click at a button **`Continue`**
(12) Check if **`Submission status`** is **`Submitted for grading`**.

End of specification