# School of Computing and Information Technology

**Student to complete:**

| | |
|---|---|
| Family name | |
| Other names | |
| Student number | |
| Table number | |

# ISIT912
# Big Data Management
# Wollongong Campus
# Liverpool Campus

## Final Examination Paper
## Spring Session 2023

| | |
|---|---|
| Exam duration | 3 hours |
| Weighting | 50% |
| Items permitted by examiner | None |
| Aids supplied | None |
| Directions to students | 6 questions to be answered. |

# CHEAT SHEET

**Sample operations on HDFS**

```
$HADOOP_HOME/bin/hadoop fs -mkdir myfolder
$HADOOP_HOME/bin/hadoop fs -put $HADOOP_HOME/*.txt myfolder
$HADOOP_HOME/bin/hadoop fs -ls myfolder
$HADOOP_HOME/bin/hadoop fs -cat myfolder/README.txt
$HADOOP_HOME/bin/hadoop fs -copyToLocal myfolder/README.txt /home
$HADOOP_HOME/bin/hadoop fs -rm myfolder/README.txt
```

**Sample preparation and processing of Java application**

```
javac -cp $HADOOP_CLASSPATH FileSystemCat.java
jar cvf FileSystemCat.jar FileSystemCat*.class
$HADOOP_HOME/bin/hadoop jar WordCountTR.jar WordCountTR -conf
                    /home/bigdata/Desktop/hadoop-local.xml local-input local-output
```

**Sample HQL**

```
create table names(
 full_name VARCHAR(30),
 age        DECIMAL(3) )
row format delimited fields terminated by ',' stored as textfile;
load data local inpath '/home/bigdata/Desktop/names.tbl' into table names;
create external table names(
 full_name VARCHAR(30),
 age        DECIMAL(3) )
row format delimited fields terminated by ',' stored as textfile
stored as textfile location '/user/bigdata/a-new-hdfs-folder'
select part, customer, sum(amount)
from orders
group by part, customer with rollup/cube/grouping sets;
```

**Sample options of window framing**

```
ROWS BETWEEN 3 PRECEDING AND CURRENT ROW,
ROWS BETWEEN UNBOUNDED PRECEDING AND 2 FOLLOWING
ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING
ROWS BETWEEN 2 FOLLOWING AND UNBOUNDED FOLLOWING
```

**HBase**

```
create 'COURSEWORK', 'STUDENT'
alter 'COURSEWORK', {NAME=>'SUBJECT', VERSIONS=>'1'}
put 'COURSEWORK','student:007','STUDENT:snumber','007'
put 'COURSEWORK','student:007','STUDENT:first-name','James'
get 'COURSEWORK','student:007'
scan 'COURSEWORK',{COLUMN=>'STUDENT:first-name'}
```

**Pig**

```
orders = load '/user/bigdata/orders.txt' using PigStorage(',') as
(item:chararray,customer:chararray,quantity:int,year:int,month:int,day:int);
dump orders;
dates = foreach orders generate day, month, year;
biggerorders = filter orders by quantity > 100;
inner_join = join orders by item, items by item;
ordergrp = group orders by item;
itemscnt = foreach ordergrp generate group, COUNT(orders.item);
```

**Spark**

```
val text = spark.read.textFile("/user/bigdata/README.txt")
text.count()
text.first()
val peopleDF = spark.sparkContext.
  textFile("/user/bigdata/week10/people.txt").
  map(_.split(",")).
  map(attributes => Person(attributes(0), attributes(1).trim.toInt)).
  toDF()
peopleDF.show()
val df = spark.read.json(flightData)
df.createOrReplaceTempView("dfTable")
df.select("DEST_COUNTRY_NAME", "ORIGIN_COUNTRY_NAME").show(2)
import org.apache.spark.sql.functions.{expr, col}
df.select(col("DEST_COUNTRY_NAME"),expr("ORIGIN_COUNTRY_NAME")).show(2)
df.select(col("DEST_COUNTRY_NAME").alias("Destination"),expr("ORIGIN_COUNTRY_NAME").alias("Origin")).show
df.filter(col("count") < 2).show(2)
```

**QUESTION 1 (10 marks)**

Assume that a file `patients.tbl` contains information about the daily measurements of patients' temperature and oxygen levels at a hospital.

For example, the first six rows in a sample file with the measurements are listed below. The meanings of the values in each column are the following.

```
Measurement date   Patient ID   Temperature   Oxygen level

15-OCT-2023        007          40            45
15-OCT-2023        008          38            55
16-OCT-2023        007          39            50
16-OCT-2023        008          36            55
17-OCT-2023        007          37            55
17-OCT-2023        008          36            60
...                ...          ...           ...
```

Assume that a file `patients.tbl` is sorted in the ascending order of measurement dates.

Assume that a file `patients.tbl` is located in a local file system in a folder `/user/bigdata/measurements`.

Explain how to implement a MapReduce application that finds the largest increment in daily recorded temperatures between two adjacent measurements for each patient admitted to a hospital in a given year.

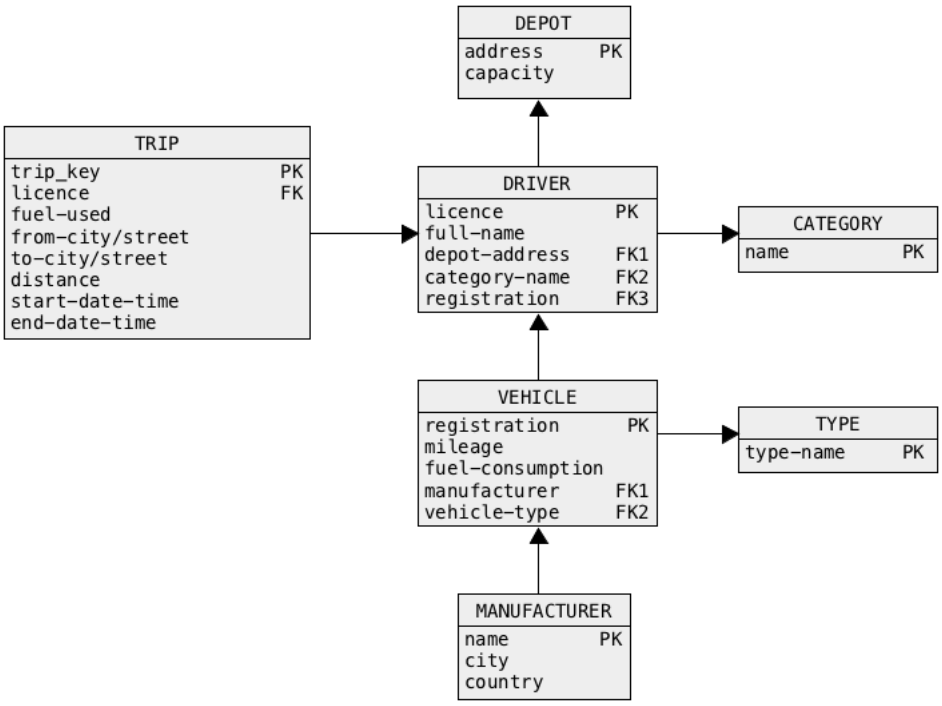
To simplify a problem, assume that a period of time when a patient stays at a hospital does not span over two or more years and each patient is admitted to a hospital only one time.

Answer the following questions in your explanations.

(i) What are the parameters of your application ?
(ii) What information is filtered out by a mapper ?
(iii) What key-value pairs must be created by a mapper ?
(iv) How key-value pairs are created by a mapper ?
(v) What key-value pairs are processed by a reducer ?
(vi) How key-value pairs are processed by a reducer ?
(vii) What is the format of the final results ?
(viii) How to upload a file `patients.tbl` to HDFS ?
(ix) How to prepare your application for processing ?
(x) How to list the final results ?

**QUESTION 2 (8 marks)**

A transportation company owns and maintains an operational database that contains information *a*bout present activities. A logical schema the database (a collection of relational schemas) is visualized below.




A transportation company would like to create a data warehouse such that the following information can be retrieved/computed from the data warehouse later on.

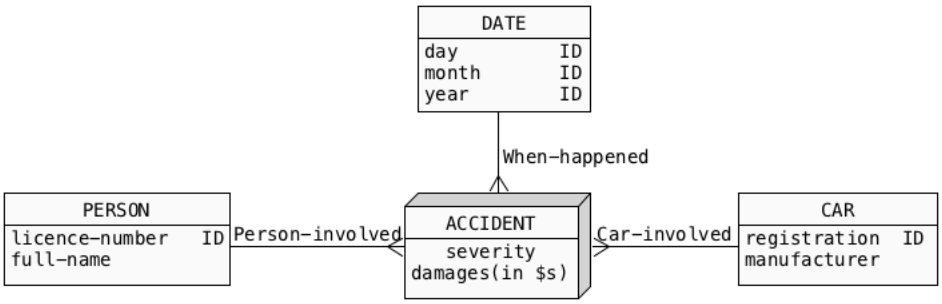
*(1) Find the total number of trips performed by a driver per day, per month and per year.*
*(2) Find the total time spent on driving per driver, per day, per month and per year.*
*(3) Find the total distance travelled per driver, per day, per month and per year.*
*(4) Find the total number of trips per car and per bus.*
*(6) Find the total amount fuel consumed per car and per bus, per day, per month. per year.*
*(7) Find the total amount fuel consumed per car and per bus, per day, per month. per year.*
*(8) Find the total number of trips made per city/street of origin and per city/street of destination.*
*(9) Find the total time spent on the trips per car and per bus.*
*(10) Find an average distance travelled per car manufacturer and per driver category.*

Your task is to draw a conceptual schema of a sample data warehouse domain listed above.

To draw a conceptual schema, use a graphical notation explained to you during the lecture classes in a subject ISIT912.

**QUESTION 3 (8 marks)**

Consider the following three-dimensional data cube.




The data cube contains information about accidents that involved people and cars.

Assume that, the text files `dates.tbl`, `people.tbl`, `cars.tbl` and `accidents.tbl` contain data obtained from the police reports for the car accidents. The sample contents of the files `dates.tbl`, `people.tbl`, `cars.tbl` and `accidents.` are given below.

`dates.tbl`
```
01,JAN,2020
02,JAN,2020
03,JAN,2020
… … … … … …
```

`cars.tbl`
```
PKR856,Rolls Royce
UUQ076,Toyota
XYZ007,Gogo Mobile
… … … … … …
```

`accidents.tbl`
```
25,SEP,2020,PKR856,Licence007,Victoria St.,Sydney,serious,1256.67
13,MAR,2021,UUQ076,Licence666,Bong Bong St.,Dapto,minor,100.00
01,FEB,2022,XYZ007,Licence999,Liberation Ave.,average,894.50
… … … … … …
```

`people.tbl`
```
Licence007,James Bond
Licence666,Harry Potter
Licence999,Robin Hood
… … … … … …
```

**(1) 4 marks**
Assume, that the files `dates.tbl`, `people.tbl`, `cars.tbl` and `accidents.tbl` are uploaded to HDFS. The locations of the files in HDFS is up to you. The types of all values are up to you.

Write HQL statements to create Hive tables that can be used to access the contents of the files `dates.tbl`, `people.tbl`, `cars.tbl` and `accidents.tbl` in HQL. Your solution must NOT replicate data included in the files `dates.tbl`, `people.tbl`, `cars.tbl` and `accidents.tbl`. All other implementation details are up to you.

**(2) 4 marks**
Write HQL statements that retrieve the following information from the files `dates.tbl`, `people.tbl`, `cars.tbl` and `accidents.tbl`.

(i) Find the total number of accidents aggregated per car model, per year, per car manufacturer and year, and the total number of accidents.

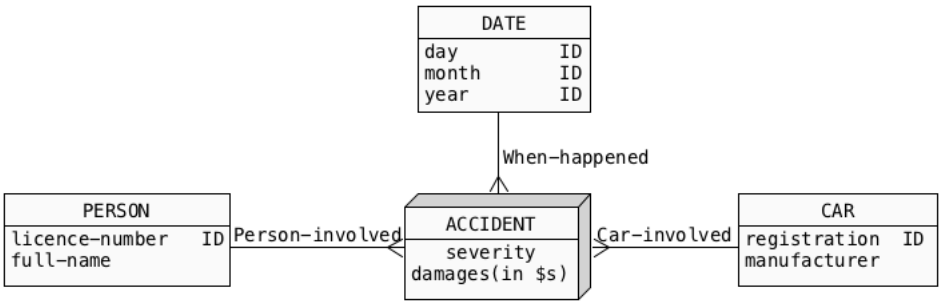
(ii) Find the average damages of all accidents that involved Toyota cars aggregated per year and per month, per year, and average damages of all accidents that involved Toyota cars.

(iii) Find the total damages aggregated per year, per car manufacturer and per city.

(iv) Find the full names of people involved in at least 3 accidents.

**QUESTION 4 (8 marks)**

Consider the following three-dimensional data cube.



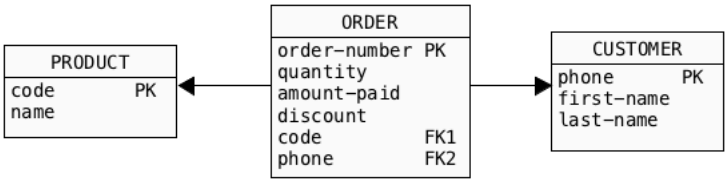

The data cube contains information about accidents that involved people and cars.

**(1) 1 mark**
Use HBase shell command language to write the commands that create HBase table implementing a conceptual schema given above.

**(2) 7 marks**
Use HBase shell command language to write the commands that insert into HBase table created in the previous step information about at least one accident that involved a person one a car on a given day. All values of attributes are up to you.

**QUESTION 5 (8 marks)**

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




Assume, that the files `product.txt`, `customer.txt` and `order.txt` contain data consistent with a logical schema of two-dimensional data cube given above.

Internal format of each file is a sequence of values separated with the commas (CSV format). Assume, that the files are uploaded to HDFS. The locations of the files in HDFS are up to you. The types of all values are up to you.

(1) Write the Pig-Latin statements that load the contents of files `product.txt`, `customer.txt` and `order.txt` into Pig storage.

Write Pig-Latin statements that implement the following queries.

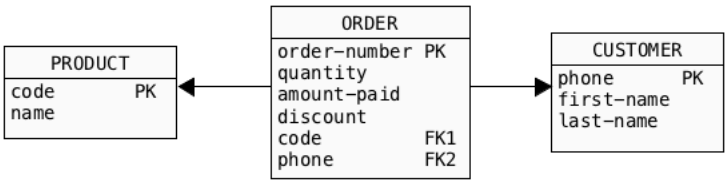
(2) Find the phone numbers of customers who purchased a product named `bolt` at the quantity greater than `500`.

(3) Find the phone numbers of customers who have not purchased any products yet. Display the phone number in the ascending order.

(4) Find the codes of products together with the total number of orders submitted for each product.

**QUESTION 6 (8 marks)**

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




Assume, that the files `product.txt`, `customer.txt` and `order.txt` contain data consistent with a logical schema of two-dimensional data cube given above.

Internal format of each file is a sequence of values separated with the commas (CSV format). Assume, that the files are uploaded to HDFS. The locations of the files in HDFS are up to you. The types of all values are up to you.

Write implementation of the following Spark-shell operations.

(1) Create the DataFrames that contain information about `products`, `orders`, and `customers`.
(2) Find the total number of orders that have a `quantity` is greater than `100`.
(3) For each product list its `code` and the total number of orders related to a product.
(4) Register a DataFrame that contains information about the orders as SQL temporary view.
(5) Use SQL view created in the previous step to find the total quantities per each product ordered.

# End of Examination Paper