Module 30 Managing Big Data

Module title	Managing Big Data
Module NFQ level (only if an NFQ level can be demonstrated)	8
Module number/reference	BSCH-MBD
Parent programme(s)	Bachelor of Science (Honours) in Computing Science
Stage of parent programme	Award stage
Semester (semester1/semester2 if applicable)	Semester 2
Module credit units (FET/HET/ECTS)	ECTS
Module credit number of units	5
List the teaching and learning modes	Direct, Blended
Entry requirements (statement of knowledge, skill and competence)	Learners must have achieved programme entry requirements.
Pre-requisite module titles	BSCH-RD
Co-requisite module titles	None
Is this a capstone module? (Yes or No)	No
Specification of the qualifications (academic, pedagogical and professional/occupational) and experience required of staff (staff includes workplace personnel who are responsible for learners such as apprentices, trainees and learners in clinical placements)	Qualified to as least a Bachelor of Science (Honours) level in Computer Science or equivalent and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.
Maximum number of learners per centre (or instance of the module)	60
Duration of the module	One academic semester, 12 weeks teaching
Average (over the duration of the module) of the contact hours per week	3
Module-specific physical resources and support required per centre (or instance of the module)	One class room with capacity for 60 learners along with one computer lab with capacity for 25 learners for each group of 25 learners

Analysis of required learning effort		
	Minimum ratio teacher / learner	**Hours**
Effort while in contact with staff		
Classroom and demonstrations	1:60	18
Monitoring and small-group teaching	1:25	18
Other (specify)		
Independent Learning		
Directed e-learning		
Independent Learning		44
Other hours (worksheets and assignments)		45
Work-based learning – learning effort		
Total Effort		125

Allocation of marks (within the module)					
	Continuous assessment	**Supervised project**	**Proctored practical examination**	**Proctored written examination**	**Total**
Percentage contribution	40%			60%	100%

Module aims and objectives

There are two aims to this module: to expose the learner to practical issues in database management systems such as database administration and query optimisation; and to give the learner a flavour of the procedures and considerations in handling Big Data. In order to gain an understanding of how to work with Big Data, the learner gains an understanding of the core concepts required such as Data Mining, Data Warehousing, and Data Analytics.

Minimum intended module learning outcomes

On successful completion of this module, the learner will be able to:

1. Perform the duties of a DBA
2. Implement query optimisation strategies
3. Discuss the important role of efficient transaction management with regards to concurrency control, database recovery and deadlock detection/prevention.
4. Discuss the considerations surrounding the processing of Big Data
5. Describe and implement various strategies in Data Mining and Warehousing
6. Apply data analytics on big data

Rationale for inclusion of the module in the programme and its contribution to the overall MIPLOs

Database internal, Big Data components, and NoSQL are a major pillar of middleware infrastructure, knowledge of how it work and to build and optimize databases for performance in relational model or NoSQL model is an important skill in any computer science related business.

Appendix 1 of the programme document maps MIPLOs to the modules through which they are delivered.

Information provided to learners about the module

Learners receive a programme handbook to include module descriptor, module learning outcomes (MIMLO), class plan, assignment briefs, assessment strategy, and reading materials.

Module content, organisation and structure

Database Management

- The role of the DBA / Security / User Management / Physical Database Issues

Query Optimisation

- Use of indexing and keys / Optimising Joins / Optimising queries in a RDBMS

Transaction Processing and Concurrency

- Transactions Stages; commit, abort, etc. / ACID properties / Concurrency problems; lost update, incorrect summary, dirty read etc. / Locking / Deadlock detection and prevention

Introduction to Big Data

- Data Model / Data Storage / Data Warehousing / Data Extraction, Transforming and Loading / Batch Processing / Scalability / NoSQL / Managing Big Data, Online Analytical Processing

Data Mining

- Structural Pattern Recognition / Input-Output / Clustering / Managing Data Warehousing Models (Bottom-Up, Top-Down, etc.) / Data Transformation Models

Introduction to Data Analytics

- Extracting information from Big Data / Statistics / Case Studies

Module teaching and learning (including formative assessment) strategy
The module is taught as a combination of lectures and lab sessions. The lecture sessions discuss and explain to learners the theoretical various other advanced topics, including query optimization, concurrency recovery, data warehouses, NoSQL and Big Data components, and usage. While relational databases introduced the basics of database systems, the additional topics covered in this module helps learners become more proficient in writing queries and expands their knowledge base so that they have a better understanding of the field.

Assessment is divided into three. There are two take home assignments that build the learner's skills with database scheduler transactions, recovery, query optimization, Hadoop mapreduce and database tuning. Finally, there is an end of semester exam that tests the learners understanding of the theoretical material.

In addition to classes, they need to put in at least four hours of study and homework each week. Lab work is used to practice some of the concepts explained in class with practical databases implementations.

Timetabling, learner effort and credit
The module is timetabled as one 1.5-hour lecture and one 1.5-hour labs per week.

The number of 5 ECTS credits assigned to this module is our assessment of the amount of learner effort required. Continuous assessment spreads the learner effort to focus on small steps before integrating all steps into the overall process of developing and deploying a cloud application.

There are 36 contact hours made up of 12 lectures delivered over 12 weeks with classes taking place in a classroom. There are also 12 lab sessions delivered over 12 weeks taking place in a fully equipped computer lab. The learner will need 45 hours of independent effort to further develop the skills and knowledge gained through the contact hours. An additional 44 hours are set aside for learners to work on worksheets and assignments that must be completed for the module as a part of the continuous assessment.

The team believes that 125 hours of learner effort are required by learners to achieve the MIMLOs and justify the award of 5 ECTS credits at this stage of the programme.

Work-based learning and practice-placement
There is no work based learning or practice placement involved in the module.

E-learning

The college VLE is used to disseminate notes, advice, and online resources to support the learners. The learners are also given access to Lynda.com as a resource for reference.

Module physical resource requirements

Requirements are for a classroom for 60 learners equipped with a projector, and a 25-seater computer lab for practical sessions with access to MySQL server and virtual box to run Big Data virtual image provided from Cloudera.

Reading lists and other information resources

Recommended Text

Elmasri, R. and Navathe, S. B. (2016) *Fundamentals of Database Systems.* Boston: Pearson.

Secondary Reading

Garcia-Molina, H., Ullman, J. D. and Widom, J. (2014) *Database systems: the complete book*. Harlow: Pearson.

Marz, N. and Warren, J. (2015) *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. New York: Manning Publications.

Specifications for module staffing requirements

For each instance of the module, one lecturer qualified to at least Bachelor of Science (Honours) in Computer Science or equivalent, and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.. Industry experience would be a benefit but is not a requirement.

Learners also benefit from the support of the programme director, programme administrator, learner representative and the Student Union and Counselling Service.

Module Assessment Strategy

The assignments constitute the overall grade achieved, and are based on each individual learner's work. The continuous assessments provide for ongoing feedback to the learner and relates to the module curriculum.

No.	Description	MIMLOs	Weighting
1	Take Home assignment: DB design, Recovery, Transactions and recovery	1,23	20%
2	Take Home assignment: Hadoop Programming	4-6	20%
3	Written exam that tests the theoretical aspects of the module		60%

All repeat work is capped at 40%.

Sample assessment materials

Note: All assignment briefs are subject to change in order to maintain current content.

Assignment Brief 1

Course:	Bachelor of Computer Science
Module:	Managing Big data
Assignment Number:	**1**
Assignment Title:	Recovery, Transactions, SQL and DB design

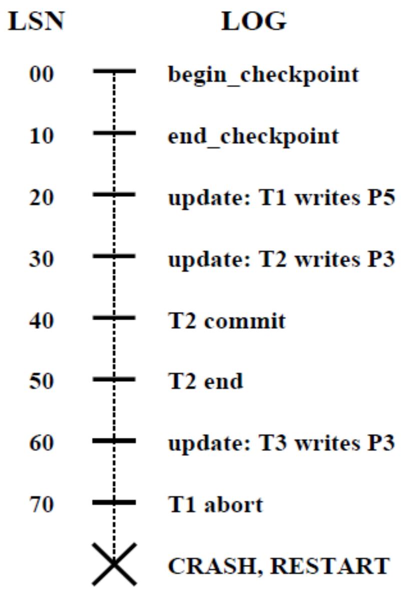
Question 1

Consider the execution shown in Figure 1 below.

(a) What is done during Analysis? (Be precise about the points at which Analysis begins and ends and describe the contents of any tables constructed in this phase.)

(b) What is done during Redo? (Be precise about the points at which Redo begins and ends.)

(c) What is done during Undo? (Be precise about the points at which Undo begins and ends.)



LSN	LOG
00	begin_checkpoint
10	end_checkpoint
20	update: T1 writes P5
30	update: T2 writes P3
40	T2 commit
50	T2 end
60	update: T3 writes P3
70	T1 abort
	CRASH, RESTART

Question 2

Given the following database schema in figure 1 below, provide SQL for the following queries/requests below.

a) Obtain the first names and last names of all customers that have subscribed for the fitness class with the name "Beginner Sumba"

b) Retrieve all fitness classes that cost €7 or less and are taught by the instructor Fiona Smith

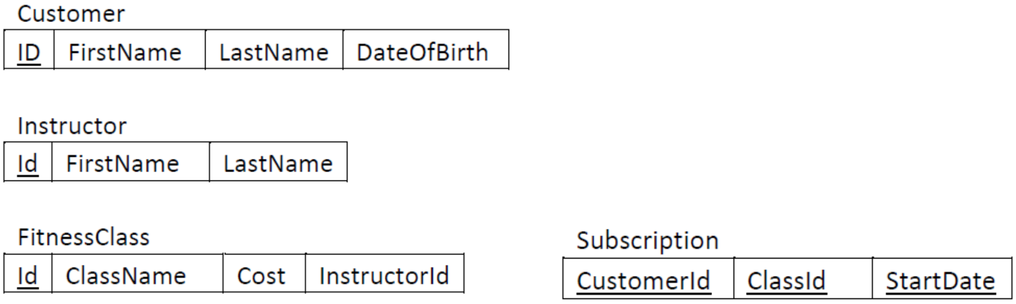
c) Customer DateOfBirth is sensitive information and should not be visible to all users of the database. Create a new database user Mary that has access to the CustomerId, FirstName and Lastname but cannot access DateOfBirth. Mary should only be able to read customer information but should not be able to make any changes to the information in the Customer table. Mary should not have access to any other information in the DBMS.

d) List the average cost of a Fitness Class by instructor. Show the Instructor Id and the Instructor Name.

e) List all the Fitness Classes which are at least 50 cent above the lowest class price. Show the Class id and class name.

f) Retrieve a list of all classes (Id and ClassName) that both Mary Jones and Tina Lenihan have subscribed to (ie Mary Jones and Tina Lenihan have been in the same Fitness Class).

Figure1:Database Schema



Customer

ID	FirstName	LastName	DateOfBirth

Instructor

Id	FirstName	LastName

FitnessClass

Id	ClassName	Cost	InstructorId

Subscription

CustomerId	ClassId	StartDate

Question 3

Consider a concurrency control manager that uses strict two phase locking that

schedules three transactions:

T1 : R1(A);R1(B);W1(A);W1(B);Co1

T2 : R2(B);W2(B);R2(C);W2(C);Co2

T3 : R3(C);W3(C);R3(A);W3(A);Co3

Each transaction begins with its first read operation, and commits with the Co statement. Answer the following questions for each of the schedules below:

Schedule 1:

R2(B);W2(B);R3(C);W3(C);R3(A);W3(A);Co3;R2(C);W2(C);Co2;R1(A);R1(B);W1(A);W1(B);Co1

Schedule 2:

R2(B);W2(B);R3(C);W3(C);R1(A);R1(B);W1(A);W1(B);Co1;R2(C);W2(C);Co2;R3(A);W3(A);Co3

a. Is the schedule conflict-serializable? If yes, indicate a serialization order.

b. Is this schedule possible under a strict 2PL protocol?

c. If strict 2PL does not allow this schedule because it denies a read or a write request, is the system in a deadlock at the time when the request is denied?

Question 4

Design the Data model for the following business case with description as:

a) A Company has car rental service rents its cars (VID) to the customers (C_ID), who can be private or business.

b) Each customer has a driver's license (LN) with one or more categories (Cat_ID). These categories allow the person to drive certain vehicle classes (VC_ID).

c) Special business programs (CBP) exist for business clients. These business clients can be assigned to one of these programs.

d) Each car belongs to one vehicle class, needs a certain fuel type (F_T) and belongs to a specific price class (Price_Class).

e) These cars have different features (Feature_ID) (Roof top,ESP, ABS , GPS …).

f) The vehicle is available for rentals for a specified time (TP_ID). There are fixed prices (Price) for several periods (eg. price per weekend days, price per day, price per week).

g) The rental agreement contains few options (insurance type, driver included, etc.) The client can choose from these options (OPT). The prices of the different options may change, depending on the rental period (TP_ID).

Deliverables:

1. ERD diagram (show cardinality, etc.)

2. Logical schemas with primary keys/Foreign keys indicated.

3. Is there any need to normalize the design, explain

Assignment Brief 2	
Course:	Bachelor of Computer Science
Module:	Managing Big data
Assignment Number:	**2**
Assignment Title:	Hadoop programming

Programming Java with mapreduce.
Use the sample count program in java on moodle as a base code and add more classes to achieve the tasks required.
(a) Write a MapReduce program to count all the words in the input file (COMPOUND.TXT) on moodle and outputs the 100 most frequent words.
(b) Add to your code a method that uses MapReduce to count the number of words with a given length. The output should contain the 100 most frequent words in decreasing order of frequency. For example: words with 2, 3, 4, … letters.

```
and 2700
not 860
...
```

c) Add to your code a method counts the number of words that have a given prefix and outputs the 100 most frequent words with the given prefix in decreasing order of frequency. For example: count how many words start with "dis".

```
disconnect 100
disappear 50
...
```

You must write your code in Java. In addition to the source code, you will submit a JAR file that can be called using the following command:
hadoop -jar wordcount.jar WordCount -input <> -output <> [-combiner] [-wordlength x] [-prefix yyy]
The inputs of your Java program should be the following:
WordCount: classname
-input: Input
-output: Output
-combiner: (Extras) Use or not use combiner
-word-length: (Extras) Consider word with x characters only
-prefix: (Extras) Consider word with yyy prefix only
What to submit:
You create and submit a ZIP file that contains:
1) Java source code and jar file: WordCount.java WordCount.jar
2) Output files. You will submit a file called Results containing the outputs.
For each problem that involves counting you will create a text file with names a_output, b_output and c_output respectively, where each line contains a word and the count of how often it occurred separated by tab, in decreasing order of frequency.
For item (b), create a text file, named b_output, containing the two performance numbers separated by a tab in one line.

GRIFFITH COLLEGE DUBLIN

QUALITY AND QUALIFICATIONS IRELAND
EXAMINATION

MANAGING BIG DATA

Lecturer(s):

External Examiner(s):

Date: **XXXXXXXX** **Time:** **XXXXXXX**

THIS PAPER CONSISTS OF FIVE QUESTIONS
FOUR QUESTIONS TO BE ATTEMPTED
ALL QUESTIONS CARRY EQUAL MARKS

QUESTION 1

(a) Consider the following log of a database system that is running undo/redo logging with checkpointing:

```
<T1 Start>
<T1, A, 45, 10>
<T2 Start>
<T2, B, 5, 15>
<T2, C, 35, 10>
<T1, D, 15, 5>
<T1 Commit>
<T3 Start>
<T3, A, 10, 15>
<BEGIN CHKPT>
<T2, D, 5, 20>
<T2 Commit>
<END CHKPT>
<T4 Start>
<T4, D, 20, 30>
<T3, C, 10, 15>
<T3 Commit>
<T4 Commit>
```

Assume the update log records are in the format:

<Tid, element, old value, new value>

What is the value of the data elements *A, B, C,* and *D* on disk after recovery?

(i) If the system crashes just before line 6 is written to disk?

(ii) If the system crashes just before line 10 is written to disk?

(iii) If the system crashes just before line 12 is written to disk?

(iv) If the system crashes just before line 13 is written to disk?

(v) If the system crashes just before line 16 is written to disk?

(vi) If the system crashes just before line 18 is written to disk?

(12 marks)

(b) Describe in detail the ACID properties.

(8 marks)

(c) Compare the basic *Binary Lock* model to the *Read/Write Lock* model in *Concurrency Control*. Make reference to the basic rules for each. Conclude by constructing pseudo code to implement the *Read/Write Lock* model.

(5 marks)

Total (25 marks)

QUESTION 2

(a) SQL supports four isolation-levels and two access-modes, for a total of eight combinations of isolation-level and access-mode. Each combination implicitly defines a class of transactions; the following questions refer to these eight classes:

(i) Consider the four SQL isolation levels. Describe which of the phenomena can occur at each of these isolation levels: dirty read, unrepeatable read, phantom problem.

(4 marks)

(ii) For each of the four isolation levels, give examples of transactions that could be run safely at that level.

(16 marks)

(b) Describe and contrast the methods of *Deadlock Prevention Protocols* and *Deadlock Detection* making specific reference to when one should be chosen over the other.

(5 marks)

Total (25 marks)

QUESTION 3

(a) Consider the schema

Airport (code, name, city, country)

Flight (number, airline, from_airport_code, to_airport_code)

Reservation(flight_number, seat_number, date, passenger_name)

Answer the following using relational algebra

(i) List the flight numbers of flights that take off from India.

(4 marks)

(ii) List the passenger who are on flight number ‘SA 747’.

(4 marks)

(iii) List all the flight information for Indian Airlines and Jet Airways.

(2 marks)

(b) For the following schedules: Answer the following questions:

$r_1(A)$; $r_2(A)$; $r_3(B)$; $w_1(A)$; $r_2(C)$; $r_2(B)$; $w_2(B)$; $w_1(C)$;

(i) What is the precedence graph for the schedule?

(5 marks)

(ii) Is the schedule conflict-serializable? If so, what are all the equivalent serial schedules?

(5 marks)

(c) What is the difference between primary index and secondary index?

(5 marks)

Total (25 marks)

QUESTION 4

Key value stores and document stores are examples of data models used in No-SQL databases. Compare databases using these two data models.

(a) Explain the structure of each of these data models, give examples of each and clearly evaluate the advantages and disadvantages of each.

(15 marks)

(b) Define Data Warehousing.

(5 marks)

(c) What are ETL tools in the context of data warehousing. Explain why they are needed.

(5 marks)

Total (25 marks)

QUESTION 5

The database below records products available for sale in the bakery and bakery orders.

Product(id, title, description)

BakeryOrders(id, p_id, price, collection_date, customer_name)

Primary keys are underlined. P_id is a foreign key for the id in the product table.

(a) Write the SQL to list the products that have had no bakery orders since 1st January 2017.

(5 marks)

(b) Write the SQL to list the total value of bakery orders by product title. The results should include products which do not have any bakery orders.

(4 marks)

(c) Explain, using examples, the terms transitive dependency and partial dependency.

(8 marks)

(d) Consider the following relational schema and set of functional dependencies.

R(A,B,C,D,E) with functional dependencies AE→ C and D →B.

(i) Determine the key (i.e., a minimal superkey) for this relation?

(ii) Decompose R into BCNF.

For full marks, justify your answers.

(8 marks)

Total (25 marks)