



## Semester 1 Examination, 2019-2020

<b>Exam Code(s)</b>	3BCT1 1EM1
<b>Exam(s)</b>	Third Year Computer Science and IT Erasmus
<b>Module Code(s)</b>	CT3532
<b>Module(s)</b>	Computer Systems and Organization
Paper No.	1
Repeat Paper	No
External Examiner(s)	Dr. Jacob Howe
<b>Internal Examiner(s)</b>	Prof. Michael Madden *Mr. Donal Kelly
<b>Instructions:</b>	Answer any THREE questions. All questions carry equal marks.
<b>Duration:</b>	2 hours
<b>No. of Pages</b>	3
<b>Discipline(s)</b>	Computer Science
<b>Course Co-ordinator(s)</b>	Mr. Donal Kelly
<b>Requirements:</b>	
Release in Exam Venue	Yes
MCQ	No
Handout	None
Statistical/ Log Tables	None
Cambridge Tables	None
Graph Paper	None
Log Graph Paper	None
Other Materials	None
Graphic material in colour	No

## Q.1.

- a) Describe a B+tree leaf node and what it contains. How is it different from a B-tree leaf node?

(7)

- b) Adding the sorted list [1,2,3,4,5,6,7,8] to a Binary Search Tree will give you poor search performance. Will you have the same problem with a B-tree? Show what this list will look like if inserted into a B-tree with order 3. How many comparisons are needed to search for number 3?

(12)

- c) What structure does the MySQL InnoDB storage engine use to store its indexes? If given a table `Student` with columns `student_id`, `first_name`, `last_name`, `department_id`, and `email`, suggest an index that could work well for the query `SELECT first_name, last_name FROM student WHERE department_id = 4 AND last_name IN ('O'Reilly', 'Reilly', 'O' Reilly', 'O Reilly');` What syntax would you use to make this index in MySQL?

(6)

## Q.2.

- a) In hashing, what is a collision? Describe an example. Why can a simple chaining collision resolution policy (CRP) result in poor search performance?

(6)

- b) Describe how entries are inserted using a linear probing method. With this CRP, why do buckets need to be marked as *'tombstones'* after items are deleted from them?

(7)

- c) Extendible Hashing and Linear Hashing are two kinds of dynamic hashing techniques. What is 'dynamic' about these approaches? From a high-level view, how do extendible and linear dynamic hashing differ? Pick one and describe how it inserts values. You may use pseudocode and/or diagrams.

(12)

PTO

### Q.3.

- a) Why is concurrency control needed? Describe an example where things can go wrong without concurrency control.

(8)

- b) What is the two-phase locking protocol? How does it guarantee serializability?

(8)

- c) MySQL uses Multiversion Concurrency Control techniques (MVCC) with its default storage engine. Describe how MVCC works. What does MySQL add to tables to allow it to use MVCC?

(9)

### Q.4.

- a) How is Boyce-Codd Normal Form (BCNF) different from 3NF? If you have a BCNF violation given table1 with columns  $A, B, C$  and functional dependencies  $AB \rightarrow C$  and  $C \rightarrow A$ , how should you decompose the table to remove it?

(8)

- b) Given the following relation  $R$ , and the functional dependencies  $F$ , decompose  $R$  such that the resulting relations satisfy 3NF.

$$R = \{A, B, C, D, W, X, Y, Z\}$$

$$F = \{AB \rightarrow C, C \rightarrow D, B \rightarrow WX, X \rightarrow YZ\}$$

(8)

- c) Given the following set of functional dependencies,  $F$ , generate the minimal cover set.  
 $F = \{X \rightarrow WD, AB \rightarrow Z, A \rightarrow BC, W \rightarrow D, Z \rightarrow C\}$

(9)