



INSTITUTE OF TECHNOLOGY TRALEE

WINTER EXAMINATION AY 2017/ 2018

Advanced Database Programming CRN 48064

External Examiner: Sean McHugh

Internal Examiner: Mr. Peter Given

Duration: 2 Hours

Instructions to Candidates:

- i) Answer any **three** questions.
 - ii) All questions carry equal marks. Submit all your rough-work, marks may be lost otherwise.
-

Question 1:

- i) A CouchDB database contains JSON documents that contain a field for "*band name*" along with a field for "*albums*" which contains an array of embedded JSON objects which contain information about the album's "*name*" and "*year released*". Explain how you would write a query in CouchDB to find out how many albums a particular band has released and explain how the query works. **(13 marks)**
- ii) In comparison to a relational database such as PostgreSQL, discuss the weaknesses of the database design in part i). **(10 marks)**
- iii) Write a note on a) JSON and b) conflict resolution in CouchDB. **(10 marks)**

Question 2:

- i) Compare and contrast MongoDB with CouchDB, using examples where appropriate. **(12 marks)**
- ii) "*MongoDB is an easy transition for developers used to working with the relational model*". Discuss. **(12 marks)**
- iii) Discuss how and why Mongo uses Replica sets, giving examples where appropriate. **(9 marks)**

Question 3:

- i) Discuss the strengths and weaknesses of Neo4J and describe a use case for an application which would be suited to using a Neo4J database. **(13 marks)**
- ii) Appendix 1 shows the Cypher code used to create a Neo4j database. Draw a graph representing the resulting database and show queries that will retrieve a) the names of all the people Alice is friends with b) friends of friends of Alice
(10 marks)
- iii) Discuss the CAP theorem and explain where Neo4J High Availability sits in relation to this theory. **(10 marks)**

Question 4

- i) Discuss the strengths and weaknesses of HBase. **(12 marks)**
- ii) Compare and contrast a HBase table with a table in a relational database such as PostgreSQL. **(12 marks)**
- iii) *"With respect to CAP, HBase is decidedly CP"*. Discuss. **(9 marks)**

Appendix 1: Cypher Code to create a Neo4j Database

```
CREATE (p:Publication {name: "Wine Expert Monthly"})

MATCH (p:Publication {name: "Wine Expert Monthly"}),
(w:Wine {name: "Prancing Wolf", vintage: 2015})
CREATE (p)-[r:reported_on]->(w)

MATCH (p:Publication {name: "Wine Expert Monthly"}),
(w:Wine {name: "Prancing Wolf"})
CREATE (p)-[r:reported_on {rating: 97}]->(w)

CREATE (g:GrapeType {name: "Riesling"})

MATCH (w:Wine {name: "Prancing Wolf"}),(g:GrapeType {name: "Riesling"})
CREATE (w)-[r:grape_type]->(g)

CREATE (wr:Winery {name: "Prancing Wolf Winery"})

MATCH (w:Wine {name: "Prancing Wolf"}),(wr:Winery {name: "Prancing Wolf Winery"})
CREATE (wr)-[r:produced]->(w)

CREATE (w:Wine {name:"Prancing Wolf", style: "Kabinett", vintage: 2002});

CREATE (w:Wine {name: "Prancing Wolf", style: "Spätlese", vintage: 2010});

MATCH (wr:Winery {name: "Prancing Wolf"}),(w:Wine {name: "Prancing Wolf"})
CREATE (wr)-[r:produced]->(w)

MATCH (w:Wine),(g:GrapeType {name: "Riesling"})
CREATE (w)-[r:grape_type]->(g)

CREATE (p:Person {name: "Alice"})

MATCH (p:Person {name: "Alice"}),
(w:Wine {name: "Prancing Wolf", style: "ice wine"})
CREATE (p)-[r:likes]->(w)

CREATE (p: Person {name: "Tom"})

MATCH (p:Person {name: "Tom"}),(w:Wine {name: "Prancing Wolf", style: "ice wine"})
CREATE (p)-[r:likes]->(w)

MATCH (p:Person {name: "Tom"}),
(pub:Publication {name: "Wine Expert Monthly"})
CREATE (p)-[r:trusts]->(pub)

CREATE (p:Person {name: "Patty"})

MATCH (p1:Person {name: "Patty"}),
(p2:Person {name: "Tom"})
CREATE (p1)-[r:friends]->(p2)

MATCH (p1:Person {name: "Patty"}),
(p2:Person {name: "Alice"})
CREATE (p1)-[r:friends]->(p2)
```

```
CREATE (p1:Person {name: "Ahmed"}), (p2:Person {name: "Kofi"});
```

```
MATCH (p1:Person {name: "Ahmed"}),(p2:Person {name: "Alice"})  
CREATE (p1)-[r:friends]->(p2);
```

```
MATCH (p1:Person {name: "Kofi"}),(p2:Person {name: "Tom"});  
CREATE (p1)-[r:friends]->(p2);
```