ISYS2095 – Assessment 2

Adam Mutimer (S3875753)

# Part A: SQL Programming

## Task 1: Non-Nested Queries

### Question 1.1:



Figure - Question 1.1 - Query Results

|  |
| --- |
| SELECT department.deptname,  count(department.deptname) AS DeptNameCount  FROM department  WHERE department.deptname IS NOT NULL AND  'DeptNameCoun' != 0  GROUP BY department.deptname  ORDER BY DeptNameCount DESC,  department.deptname ASC; |

### Question 1.2:



Figure - Question 1.2 - Query Results

|  |
| --- |
| SELECT academic.title,  academic.givename,  academic.famname,  Count(paper.title) AS PaperCount  FROM academic, interest, paper, author  WHERE academic.title like 'dr'  AND interest.descrip like '%database%'  AND interest.acnum = academic.acnum  AND paper.title like '%database%'  AND paper.panum = author.panum  AND author.acnum = academic.acnum  GROUP BY academic.acnum  ORDER BY PaperCount DESC, academic.famname ASC, academic.givename ASC; |

## Task 2: Nested Queries

### Question 2.1:



Figure - Question 2.1 Query Results

|  |
| --- |
| SELECT department.deptnum,  department.deptname,  department.state  FROM department  WHERE UPPER(department.state) IN ("VIC", "QLD")  AND department.deptnum NOT IN ( SELECT academic.deptnum FROM academic ); |

### Question 2.2:



Figure - Question 2.2 Query Results

|  |
| --- |
| SELECT department.deptnum,  department.deptname,  department.state  FROM department  WHERE NOT EXISTS( SELECT academic.deptnum FROM academic WHERE academic.deptnum = department.deptnum GROUP BY academic.deptnum )  AND UPPER(department.state) IN ("VIC", "QLD"); |

### Question 2.3:



Figure 5 - Question 2.3 Query Results

NOTE: Question specified only to show the full name of the academics, It did not specify if we needed to concatenate the 3 columns or display them individually. I assumed it required them to be concatenated. *Used: Title + Given Name + Family Name*

|  |
| --- |
| SELECT academic.acnum,  academic.title || '. ' || academic.givename || ' ' || academic.famname AS [Full Name]  FROM academic  WHERE academic.acnum IN (  SELECT author.acnum  FROM author  WHERE academic.acnum IN (SELECT acnum from author WHERE panum IN (SELECT panum from author WHERE ACNUM=202) GROUP BY ACNUM)  )  AND academic.acnum != 202  GROUP BY acnum  ORDER BY academic.FAMNAME, academic.givename |

### Question 2.4:



Figure 6 - Question 2.4 Query Result

|  |
| --- |
| SELECT deptname  FROM (  SELECT deptname,  MAX(occur)  FROM (  SELECT deptname,  Count(deptname) AS occur  FROM department  GROUP BY deptname  )  ); |

## Task 3: Set Operators

### Question 3.1:



Figure 7 - Question 3.1 Query Results

|  |
| --- |
| SELECT academic.acnum  FROM academic  EXCEPT  SELECT author.acnum  FROM author  INTERSECT  SELECT acnum  FROM (  SELECT acnum,  Count(interest.acnum) AS count  FROM INTEREST  GROUP BY interest.acnum  HAVING count >= 5  ); |

### Question 3.2:



Figure 8 - Question 3.2 Query Results

|  |
| --- |
| -- Select All academics:  SELECT academic.acnum  FROM academic  -- Remove Academics that have not authored any papers AND Remove 114:  INTERSECT  SELECT interest.acnum  FROM interest  WHERE interest.acnum != 114  -- Remove Academics with out matching interest fields and total matches matching 114 interest count  INTERSECT  SELECT interest.acnum  FROM interest  WHERE fieldnum IN (  SELECT interest.fieldnum  FROM interest  WHERE interest.acnum = 114  )  GROUP BY interest.acnum  HAVING Count(interest.acnum) = (  SELECT Count(interest.fieldnum)  FROM interest  WHERE interest.acnum = 114  ); |

# Part B: Normalisation

## Task 4: Relational Database Design

Diagram

Description automatically generated

Figure : University ER Diagram (Figure 2)

|  |
| --- |
| Course(CCode, Name)  CourseOffering(CCode\*,OCode,Start Date,Weeks,Has Break)  Contract(CNumber, Start Date, End Date, Salary,Is Full Time, Is Casual,SNo\*,Staff Name\*)  Staff(SNo,StaffName,Academic Level)  Lecture(CCode\*,OCode\*,SNo\*)  Tutor(CCode\*,OCode\*,SNo\*,Hours,Rate)  Coordinate(CCode\*,OCode\*,SNo\*,Hours) |