**COMP2714 Tutorial 3.2: SQL Basic –Worksheet**

# Introduction

**Purpose**: The purpose of this tutorial is to provide you with experience writing basic SQL SELECT queries. You will be able to apply techniques and functions such as aggregation, grouping and multiple relation queries to return specific subsets of data from a database.

**Learning Outcomes:** At the end of this tutorial, you will be able to:

Correctly use aggregate functions

Correctly use groupings in SELECT statements

Identify the difference between the keywords WHERE and HAVING

Write multiple relation queries to reconcile and return data from a variety of tables correctly

# Section A: Aggregation

The following is the revised relational schema for the blogging database from

tutorial 3.1. Based on this schema, write SQL queries which employ aggregate

functions to complete the tasks in this section.

Blog [blogSite, owner, dateCreated]

Article [blogSite, articleTitle, articleType, lengthInWords, authorName]

Foreign Keys:

Article.blogSite references Blog.blogSite

**A.1** Find the average word length of all articles of type ‘tech’ in blog site

‘howtosql.com’.

**A.1**

图形用户界面, 文本, 应用程序

描述已自动生成SELECT *AVG*(*Article*.lengthInWords) AS 'Average Word Length'  
FROM *Article*WHERE articleType = 'tech' AND blogSite = 'howtosql.com';



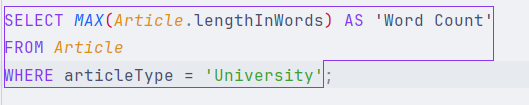
**A.2** How many articles does the blog site ‘howtosql.com’ have?

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

SELECT *COUNT*(*\**) AS Articles  
FROM *Article*WHERE blogSite = 'howtosql.com';

**A.3** What is the word count for the longest article of type “University”?



SELECT MAX(Article.lengthInWords) AS 'Word Count'

FROM Article

WHERE articleType = 'University';

**TUTORIAL** **CONTINUES** **ON** **NEXT** **PAGE**

# Section B: Grouping and Having

Based on the following schema, write SQL queries which employ the GROUP BY and/or HAVING keywords to complete the tasks in this section.

Blog [blogSite, owner, dateCreated]

Article [blogSite, articleTitle, articleType, lengthInWords, authorName]

Foreign Keys:

Article.blogSite references Blog.blogsite

* 1. Return a list with the number of articles for each blog site.

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

SELECT Article.blogSite, COUNT(\*) As 'Article Number'  
FROM Article  
GROUP BY blogSite;

* 1. Which owner(s) have at least 4 blog sites?

图形用户界面, 文本, 应用程序

中度可信度描述已自动生成

SELECT *Blog*.owner AS owner  
FROM *Blog*GROUP BY *Blog*.owner  
HAVING *COUNT*(blogSite) >= 4;

* 1. Return a list of all authors who wrote more than 3 articles, excluding the author “John Daniels”.

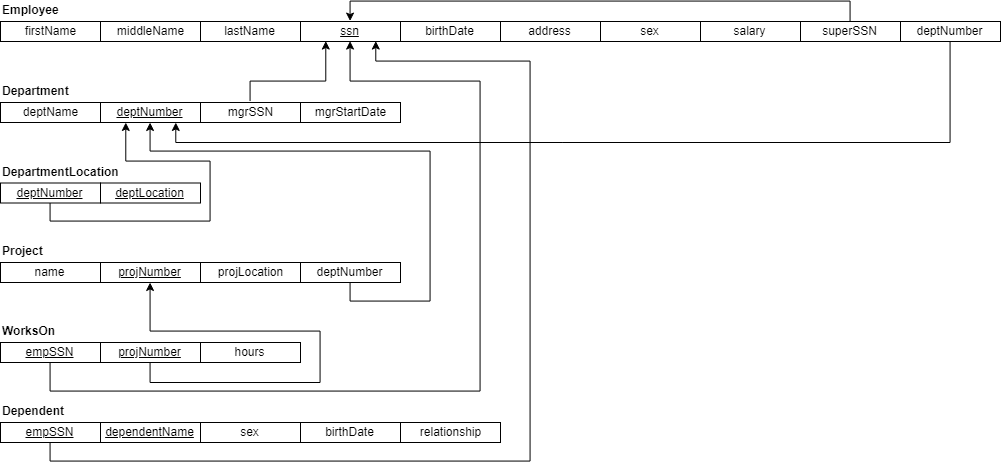
图形用户界面, 文本, 应用程序

描述已自动生成

SELECT *Article*.authorName AS author  
FROM *Article*WHERE authorName != 'John Daniels'  
GROUP BY *Article*.authorName  
HAVING *COUNT*(articleTitle) > 3;

# Section C: Multiple Relation Queries

Using the following relation schema diagram for a company database, write SQL multiple relation queries to complete the tasks in this section.1



* 1. List the first name and last name of employees who have a dependent with the same first name as themselves.

图形用户界面, 文本, 网站

中度可信度描述已自动生成

SELECT *Employee*.firstName, *Employee*.lastName  
FROM *Employee*JOIN *Department* ON *Employee*.ssn = *Department*.deptNumber  
WHERE *Employee*.firstName = *Department*.deptName;

* 1. The Australian Bureau of Statistics has requested a list of all the first names stored in our systems. Returned the combined list of employees’ first names and dependent names. You may assume they are of the same data type and duplicates should be returned.

图形用户界面, 文本

中度可信度描述已自动生成

SELECT Employee.firstName  
FROM Employee  
UNION ALL  
SELECT Dependent.dependentName  
FROM *Dependent*;

**C.3** Retrieve the names of employees in department 5 who work more than 10 hours per week on the 'ProductX' project.

图形用户界面, 文本, 应用程序

描述已自动生成

SELECT *Employee*.firstName, *Employee*.lastName  
FROM *Employee*JOIN *WorksOn* ON *Employee*.ssn = *WorksOn*.empSSN  
JOIN *Project* ON *WorksOn*.projNumber = *Project*.projNumber  
WHERE *Employee*.deptNumber = 5  
 AND *Project*.name = 'ProductX'  
 AND *WorksOn*.hours > 10;

* 1. For each project, list the project number and the total hours per week (by all employees) spent on that project.
  2. For each department whose average employee salary is more than $30000, retrieve the department name and the number of employees working for that department. You may assume Department.name is a candidate key.
  3. Return a list showing the salary difference between each employee and their supervisor if they have one. Your return should be of the format [ssn, salaryDif].