**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Databases |
| **Assessment Title:** | Data Manipulation and Validation |
| **Lecturer Name:** | Aldana Louzan |
| **Student Full Name:** | Halil UGUR |
| **Student Number:** | 2022389 |
| **Assessment Due Date:** | 04/12/2022 @23:59 |
| **Date of Submission:** | 03.12.2022 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

Contents

[Abstract 4](#_Toc120991301)

[Part 1 5](#_Toc120991302)

[Part 2 8](#_Toc120991303)

[Part 3 12](#_Toc120991304)

[GitHub Repository 15](#_Toc120991305)

[References 15](#_Toc120991306)

**Pictures**

[Picture 1: Department's relations query result 5](#_Toc120991224)

[Picture 2: Employees information query result 5](#_Toc120991225)

[Picture 3: Department titles query result 6](#_Toc120991226)

[Picture 4: Job titles query result 6](#_Toc120991227)

[Picture 5: List of employees sorted as name and last name 7](#_Toc120991228)

[Picture 6: Employee search size result 8](#_Toc120991229)

[Picture 7: List of employees that have more than two more titles 8](#_Toc120991230)

[Picture 8: List all tables 9](#_Toc120991231)

[Picture 9: List employees that surname is "Simmel" 9](#_Toc120991232)

[Picture 10: Employee size that surname start with B 9](#_Toc120991233)

[Picture 11: Created employee training table 10](#_Toc120991234)

[Picture 12: Added new data into training table 10](#_Toc120991235)

[Picture 13: Employees training table was deleted 11](#_Toc120991236)

[Picture 14: email\_address column added to employees table 11](#_Toc120991237)

[Picture 15: Added new email address into email\_address column for 10001 employee 11](#_Toc120991238)

[Picture 16: Manager size by gender on departments 12](#_Toc120991239)

[Picture 17: Salary amount by gender 12](#_Toc120991240)

[Picture 18: Size of employees by salary range 13](#_Toc120991241)

[Picture 19: List of employees group by last name and first name 13](#_Toc120991242)

[Picture 20: Salary list of the person whose employee id is 10012 14](#_Toc120991243)

# Abstract

The Employee Sample Database (created by Fusheng Wang and Carlo Zaniolo at Siemens Corporate Research) is an extensive database of four million records spread across six tables, created for system testing purposes. This article was written to share the results of queries made on the employees database. The queries consist of three parts, and each is presented under a separate heading. In the first part, the aim is to query the existing tables and to obtain new, understandable data from the available data. In the second part, the aim is to add a new column to the existing table or to add a new table to the database. In the third part, data analysis is aimed at by running complex queries on tables.

# Part 1

1. List all attributes present in the departments relation.

SELECT \*  
FROM departments AS d  
 JOIN dept\_manager AS dm ON d.dept\_id = dm.dept\_id  
 JOIN dept\_emp AS de ON d.dept\_id = de.dept\_id;

A picture containing text, electronics, screenshot

Description automatically generated

Picture : Department's relations query result

1. List all employee IDs of all past/current employees, their first and last names.

SELECT emp\_id, first\_name, last\_name  
FROM employees;

A picture containing graphical user interface

Description automatically generated

Picture : Employees information query result

1. List all department titles present in the database.

SELECT dept\_name  
FROM departments;

A screenshot of a phone

Description automatically generated with medium confidence

Picture : Department titles query result

1. List all unique job titles found in the database, and order them alphabetically.

SELECT DISTINCT title  
FROM titles  
ORDER BY title ASC;

Graphical user interface, application

Description automatically generated

Picture : Job titles query result

1. List all past/current employees’ names ordered alphabetically in ascending order, i.e. first name and last name in alphabetical order.

SELECT first\_name, last\_name  
FROM employees  
ORDER BY first\_name ASC, last\_name ASC;

A screenshot of a computer

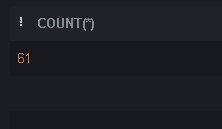
Description automatically generated with medium confidence

Picture : List of employees sorted as name and last name

# Part 2

1. The number of all employees that started on 1991-05-01.

SELECT *COUNT*(\*)  
FROM employees  
WHERE hire\_date = "1991-05-01";



Picture : Employee search size result

1. List all emp\_no who have had strictly more than 2 titles and display the total number of the titles they have had.

SELECT e.emp\_id, *COUNT*(e.emp\_id) AS title\_size  
FROM employees AS e  
 JOIN titles AS t ON e.emp\_id = t.emp\_id  
GROUP BY e.emp\_id  
HAVING *COUNT*(e.emp\_id) > 2;

Background pattern

Description automatically generated

Picture : List of employees that have more than two more titles

1. List female employees (past/current) together with all other relation attributes.

SELECT \*  
FROM employees AS e  
 JOIN titles AS t ON e.emp\_id = t.emp\_id  
 JOIN dept\_emp AS de ON de.emp\_id = e.emp\_id  
 JOIN dept\_manager AS dm ON dm.emp\_id = e.emp\_id  
 JOIN departments AS d ON d.dept\_id = de.dept\_id OR d.dept\_id = dm.dept\_id  
 JOIN salaries AS s ON s.emp\_id = e.emp\_id  
WHERE gender = "F";

A picture containing calendar

Description automatically generated

Picture : List all tables

1. List past/current employees hired prior to 1986-01-01 with the surname Simmel.

SELECT \*  
FROM employees  
WHERE last\_name = "Simmel"  
 AND hire\_date < "1986-01-01";

A picture containing text, indoor, electronics

Description automatically generated

Picture : List employees that surname is "Simmel"

1. How many past/current employees’ last name begins with the capital letter B? Use a column alias total with B to output your results.

SELECT *COUNT*(\*) AS "total with B"  
FROM employees  
WHERE last\_name LIKE "B%";

Graphical user interface, application

Description automatically generated

Picture : Employee size that surname start with B

1. Create a new table called emp\_training with 3 columns:
   * trainer\_no: this should be the primary key and is of type integer and is an auto-increment.
   * first\_name: this data type is varchar(30) and should not be NULL
   * last\_name: this data type is varchar(30) and should not be NULL
   * t\_module: this data type is varchar(20)

DROP TABLE IF EXISTS emp\_training;  
CREATE TABLE IF NOT EXISTS emp\_training  
(  
 trainer\_no INTEGER PRIMARY KEY AUTOINCREMENT,  
 first\_name TEXT NOT NULL,  
 last\_name TEXT NOT NULL,  
 t\_module TEXT  
);

Text

Description automatically generated

Picture : Created employee training table

1. Insert 2 new rows into the emp\_training table:
   * Row 1:
     + fname: Joe
     + lname: Bloggs
     + module: Google Docs
   * Row 2:
     + fname: Fred
     + lname: Bloggs
     + module: Google Sheets

INSERT INTO emp\_training (first\_name, last\_name, t\_module)  
VALUES ('Joe', 'Bloggs', 'Google Docs');  
  
INSERT INTO emp\_training (first\_name, last\_name, t\_module)  
VALUES ('Fred', 'Bloggs', 'Google Sheets');

Graphical user interface

Description automatically generated

Picture : Added new data into training table

1. The organisation no longer wishes to record the employees training within the database. Therefore, delete the newly created emp\_training table.

DROP TABLE IF EXISTS emp\_training;

A picture containing text

Description automatically generated

Picture : Employees training table was deleted

1. Alter the employees table to include an email\_address field of type varchar(20).

ALTER TABLE employees  
 ADD COLUMN email\_address VARCHAR(20);

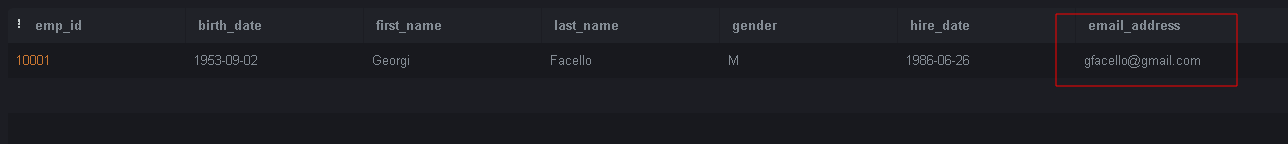
Graphical user interface, application

Description automatically generated

Picture : email\_address column added to employees table

1. Update the email address of Georgi Facello to gfacello@gmail.com, where emp\_no equals to 10001.

UPDATE employees  
SET email\_address="gfacello@gmail.com"  
WHERE emp\_id = 10001;



Picture : Added new email address into email\_address column for 10001 employee

# Part 3

1. List the number of male managers and female managers who work for each department. Make sure to display the gender, the number of employees (renamed as num\_empGender) and dept\_no, ordered by department number in an ascendant order.

SELECT d.dept\_id, e.gender, *COUNT*(e.emp\_id) AS num\_empGender  
FROM departments AS d  
 JOIN dept\_manager AS dm ON d.dept\_id = dm.dept\_id  
 JOIN employees AS e ON e.emp\_id = dm.emp\_id  
GROUP BY d.dept\_id, e.gender  
ORDER BY d.dept\_id ASC;

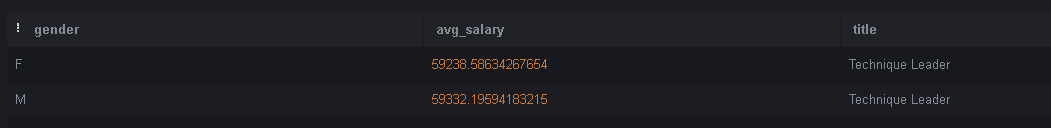
A screenshot of a computer

Description automatically generated with medium confidence

Picture : Manager size by gender on departments

1. List the average salary of male and female employees whose title is "Technique Leader". In your result table should appear, gender, average salary named as avg\_salary and title.

SELECT e.gender, *AVG*(s.salary) AS avg\_salary, t.title  
FROM employees AS e  
 JOIN titles AS t ON e.emp\_id = t.emp\_id  
 JOIN salaries AS s ON e.emp\_id = s.emp\_id  
WHERE t.title = "Technique Leader"  
GROUP BY gender;



Picture : Salary amount by gender

1. The number of employees that have a current salary (i.e., to\_date equals to 9999-01-01) between 90000 and 90040.

SELECT *COUNT*(\*)  
FROM employees AS e  
 JOIN salaries AS s ON e.emp\_id = s.emp\_id  
WHERE *CURRENT\_DATE* < s.to\_date  
 AND s.salary BETWEEN 90000 AND 90040;

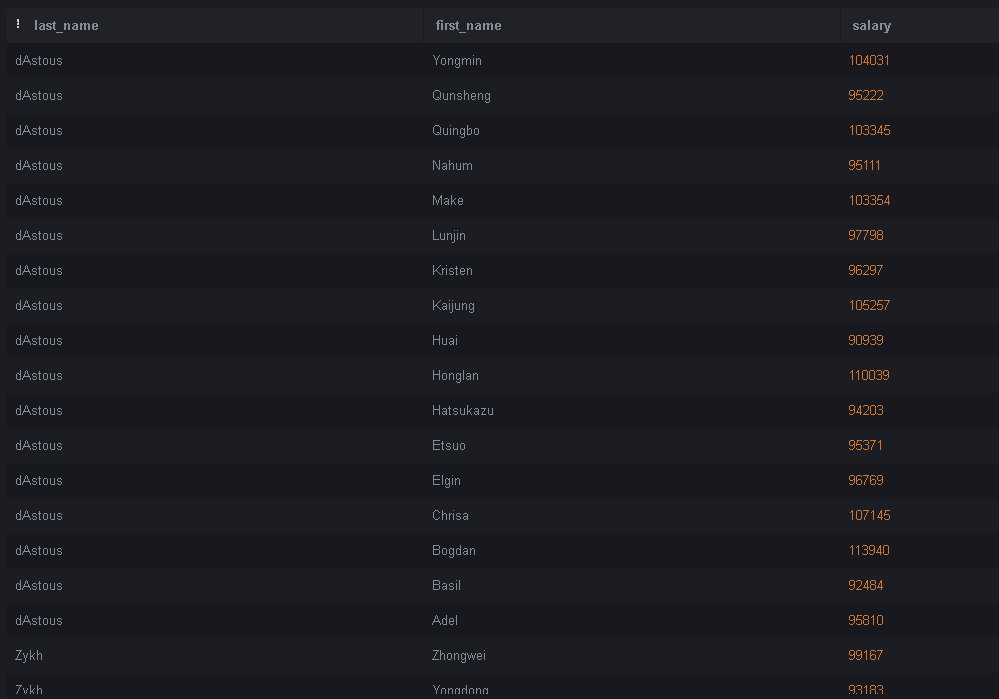
Graphical user interface, application

Description automatically generated

Picture : Size of employees by salary range

1. List all unique employees’ last and first names (using GROUP BY method) that have a current salary (i.e., to\_date equals to 9999-01-01) greater than 90000, outputting both names in descending order (sort by the last name first and then the first name) and displaying their current salaries (using the INNER JOIN method).

SELECT e.last\_name, e.first\_name, s.salary  
FROM employees AS e  
 INNER JOIN salaries AS s on e.emp\_id = s.emp\_id  
WHERE *CURRENT\_DATE* < s.to\_date  
 AND s.salary > 90000  
GROUP BY e.last\_name, e.first\_name  
ORDER BY e.last\_name DESC, e.first\_name DESC;



Picture : List of employees group by last name and first name

1. First name, last name, all salary dates and related amounts for the employee with employee number 10012.

SELECT e.first\_name, e.last\_name, s.from\_date, s.to\_date, s.salary  
FROM employees AS e  
 JOIN salaries AS s ON e.emp\_id = s.emp\_id  
WHERE e.emp\_id = 10012;

A picture containing graphical user interface

Description automatically generated

Picture : Salary list of the person whose employee id is 10012

1. In relation to the table named salaries in Figure 1 above. Answer in text:
   * What is the degree of this table?
   * What column(s), if any, make(s) up the primary key?
   * What column(s), if any, make(s) up the foreign key?

**Answer**:

* The table of degree is 4. Because there are 4 attributes (columns) in the salaries table.
* There is no primary key. But there is composite key.
* emp\_id is a foreign key for the salaries table

1. In the given schema, the tables dept\_emp, dept\_manager, salaries, titles have composite keys. Explain for each relation why this is the case? Support your answer with appropriate references.

**Answer**:

A composite key is two or more columns in a table that are combined to uniquely identify each row in the table. The columns must be combined to create a unique identifier, but individually they do not guarantee uniqueness [1].

For dept\_manager and dept\_emp tables: There are emp\_id and dept\_id columns. These columns are likely to repeat the same data, but when combined they form a unique key.

For salaries and titlies tables: In this table, we need to consider the emp\_id and dates. Because the employee may have different duties or salaries on different dates.

# GitHub Repository

All the materials we use can be accessed from this repo: [GitHub Repo](https://github.com/halilugur/database_ca2)

# References

1. geeksforgeeks (2021). *Composite Key in SQL*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/composite-key-in-sql/
2. javatpoint (n.d.). *Learn SQL Tutorial - Javatpoint*. [online] www.javatpoint.com. Available at: https://www.javatpoint.com/sql-tutorial
3. sqltutorial (2020). *SQL Tutorial - Essential SQL for the Beginners*. [online] SQL Tutorial. Available at: https://www.sqltutorial.org/
4. w3schools (2019). *SQL Tutorial*. [online] W3schools.com. Available at: https://www.w3schools.com/