CCT College Dublin

Assessment Cover Page

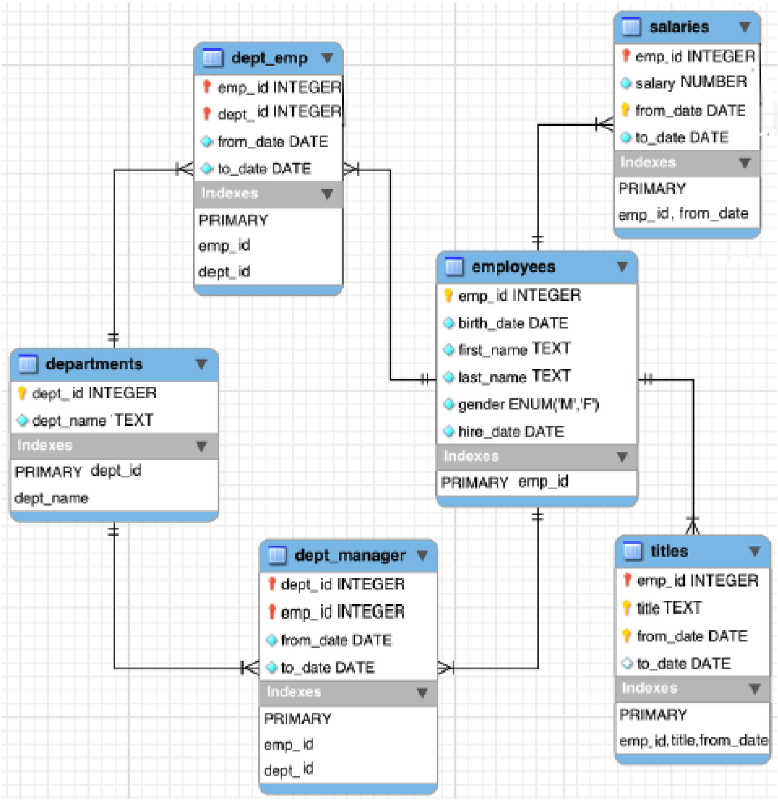
Module Title:	Databases, Web Dev.
Assessment Title:	Data Manipulation and Validation
Lecturer Name:	Mikhail Timofeev and Aldana Louzan
Student Full Name:	Burak Esit
Student Number:	2022361
Assessment Due Date:	04.12.2022
Date of Submission:	03.12.200

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.

Assignment Introduction

In this assignment, the **employees sample database** (created by Fusheng Wang and Carlo Zaniolo at Siemens Corporate Research), a large base of data spread over six separate tables and consisting of 4 million records in total that was created for system testing purposes. The following diagram provides an overview of the structure of the **employees.db**:



employees.db Schema

1.1 Databases CA Part 1 (20%)

1. List all attributes present in the **departments** relation. [4 marks]
2. List all **employee IDs** of all past/current employees, their **first** and **last names**. [4 marks]
3. List all department **titles** present in the database. [4 marks]
4. List all **unique job titles** found in the database, and **order them alphabetically.** [4 marks]
5. List all past/current **employees' names ordered alphabetically** in ascending order, i.e. first name and last name in alphabetical order. [4 marks]

Answers

1. SELECT dept_id,dept_name FROM departmens
2. SELECT first_name,last_name FROM employees
3. SELECT dept_name FROM departments
4. SELECT * FROM titles GROUP BY title
5. SELECT first_name || ' ' || last_name as fullname FROM employees ORDER BY fullname ASC

1.2 Database CA Part 2 (40%)

1. The number of all employees that started on **1991-05-01**. [4 marks]
2. List all **emp_no** who have had strictly **more than 2 titles** and display **the total number of the titles** they
have had. [4 marks]
3. List **female employees** (past/current) together with all other relation attributes. [4 marks]
4. List past/current **employees** hired prior to **1986-01-01** with the surname **Simmel** [4 marks]
5. How many past/current **employees' last name** begins with the capital letter **B**?
Use a column alias **total with B** to output your results. [4 marks]
6. 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)** [4 marks]

7. 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**
[4 marks]

8. The organisation no longer wishes to record the employees training within the database. Therefore, delete the newly created **emp_training** table. [4 marks]
9. Alter the **employees** table to include an **email_address** field of type **varchar(20)**. [4 marks]
10. Update the email address of **Georgi Facello** to **gfacello@gmail.com**, where **emp_no** equals to **10001**.
[4 marks]

Answers

1. SELECT Count(emp_id) FROM employees WHERE hire_date = '1991-05-01'
2. SELECT emp_id,title,COUNT(title) as amountofCount From titles Group by emp_id HAVING amountofCount > 2
3. SELECT * from employees WHERE gender='F'
4. SELECT * from employees WHERE hire_date<'1986-01-01' and last_name='Simmel'
5. SELECT Count(emp_id) as totalwithB from employees where last_name like 'B%'
6. CREATE TABLE emp_training (trainer_no integer PRIMARY KEY AUTOINCREMENT,

first_name varchar(30) Not NULL,last_name varchar(30)NOT NULL,t_module varchar(20));

7. 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');

8. DROP TABLE emp_training;
9. ALTER TABLE employees ADD email_address varchar(20);
10. UPDATE employees SET email_address="gfacello@gmail.com" WHERE emp_id=10001

1.3 Database CA Part 3 (40%)

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. [4 marks]

2. 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. [4 marks]

3. The number of employees that have a current salary (i.e., **to_date** equals to **9999-01-01**) between **90000** and **90040**. [4 marks]

4. 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 also displaying their current salaries (using the **INNER JOIN** method). [4 marks]

5. First name, last name, all salary dates and related amounts for the employee with employee number **10012**. [6 marks]

6. In relation to the table named salaries in **Figure 1** above. Answer in text:

a) What is the **degree** of this table?

b) What column(s), if any, make(s) up the **primary key**?

c) What column(s), if any, make(s) up the **foreign key**? [6 marks]

7. 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 [12 marks]

Answers

1. SELECT d.dept_id, e.gender, COUNT(e.emp_id) AS num_empGender FROM departments AS d JOIN dept_manager AS dm ON dm.dept_id=d.dept_id JOIN employees AS e ON dm.emp_id=e.emp_id GROUP BY d.dept_id, e.gender ORDER BY d.dept_id ASC

2.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

3.SELECT COUNT(*) AS current_employees FROM employees AS e JOIN salaries AS s ON e.emp_id=s.emp_id WHERE CURRENT_DATE < s.to_date and s.salary > 90000 and s.salary < 90040

4.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

5.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

6.The degree of the table is 4

The primary key is emp_id and from_date

The foreıgn key is emp_id.

7.At the same time we dont want to one employee should be a manager of two departmens.

SELECT Count(employees.emp_id) as total_no,title,first_name,last_name,employees.emp_id

from employees INNER join titles ON employees.emp_id=titles.emp_id

GROUP by employees.emp_id

GitHub Repository

All the materials we use can be accessed from this repo: GitHub Repo

References

1. https://github.com/googlearchive/code-prettify
2. https://codemirror.net/
3. https://getbootstrap.com/
4. https://www.geeksforgeeks.org/bootstrap/
5. https://www.w3schools.com/bootstrap/bootstrap_ver.asp
6. https://sql.js.org/#/
7. https://sql.js.org/examples/GUI/
8. https://stackoverflow.com/

References

1. https://github.com/googlearchive/code-prettify
2. https://codemirror.net/
3. https://getbootstrap.com/
4. https://www.geeksforgeeks.org/bootstrap/
5. https://www.w3schools.com/bootstrap/bootstrap_ver.asp
6. https://sql.js.org/#/
7. https://sql.js.org/examples/GUI/
8. https://stackoverflow.com/