CCT College Dublin

Assessment Cover Page

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **Programme Title** | Higher Diploma in Science in Computing |
| **Module Titles** | Databases and Web Development |
| **Assignment Title** | Data Manipulation and Validation |
| **Lecturers** | Aldana Louzan and Mikhail Timofeev |
| **Student Full Name** | Maria Helena dos Santos Ferreira |
| **Student Number** | 2022443 |
| **Submission Deadline** | 04/12/2022 @23:59 |
| **Date of Submission:** | 09/12/2022 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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. |

[**https://github.com/MariaFerreiraCCT/webdev\_ca2**](https://github.com/MariaFerreiraCCT/webdev_ca2)

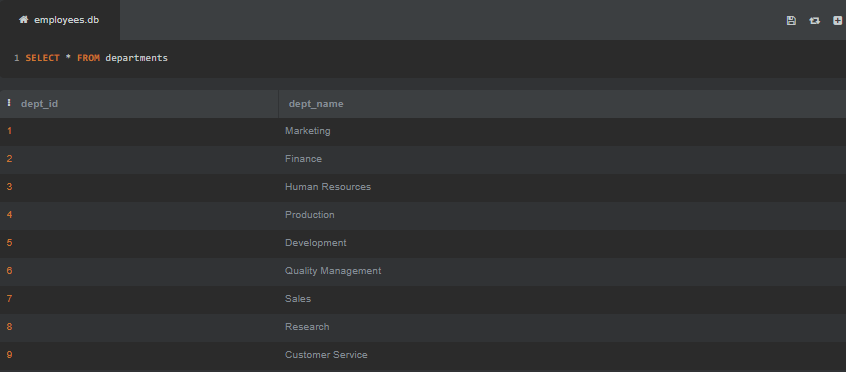
**1.1 Databases CA Part 1**

**1. List all attributes present in the department’s relation.**

**Query:**

Select \* From departments

**Screenshot:**

****

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

**Query:**

SELECT emp\_id, first\_name, last\_name from employees

**Screenshot:**

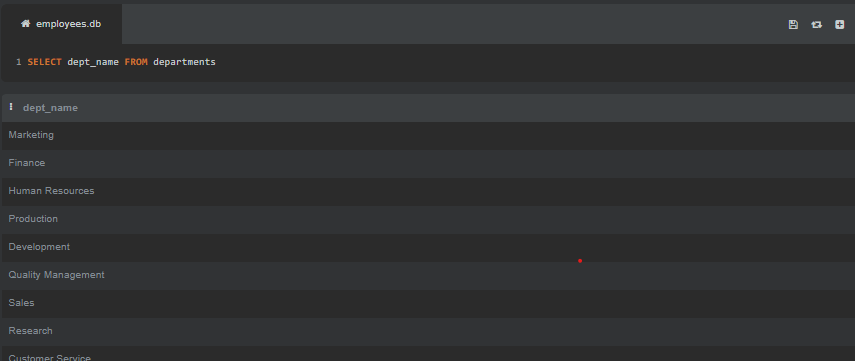
****

**3. List all department titles present in the database.**

**Query:**

SELECT dept\_name FROM departments

**Screenshot:**

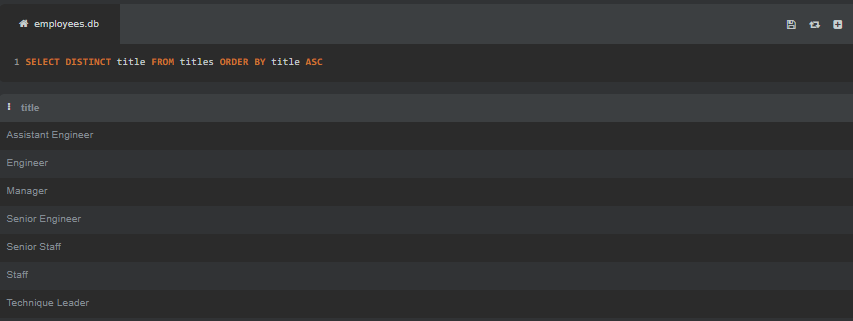
****

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

**Query:**

SELECT DISTINCT title FROM titles ORDER BY title ASC

**Screenshot:**

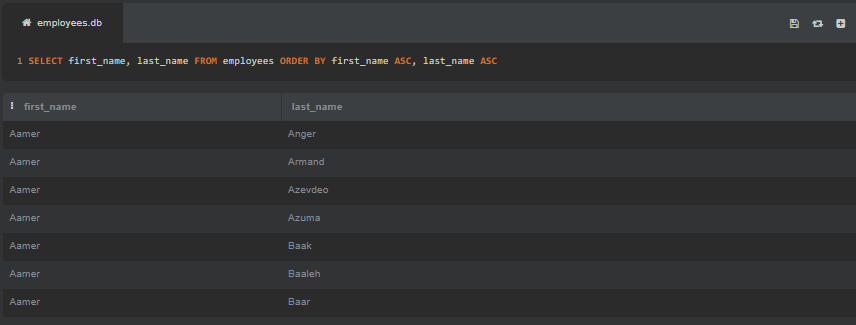
****

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

**Query:**

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

**Screenshot:**

****

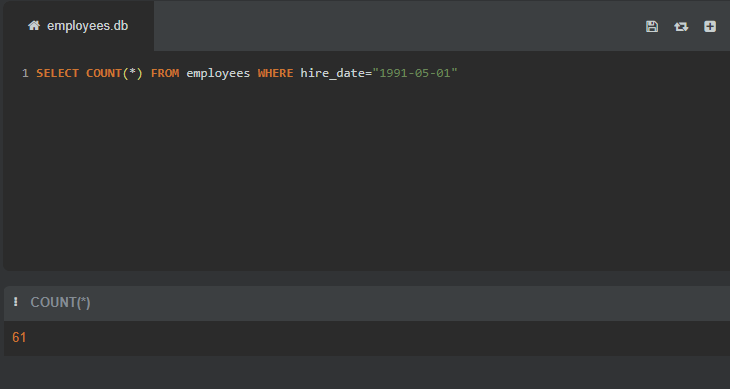
**1.2 Database CA Part 2**

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

**Query:**

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

**Screenshot:**

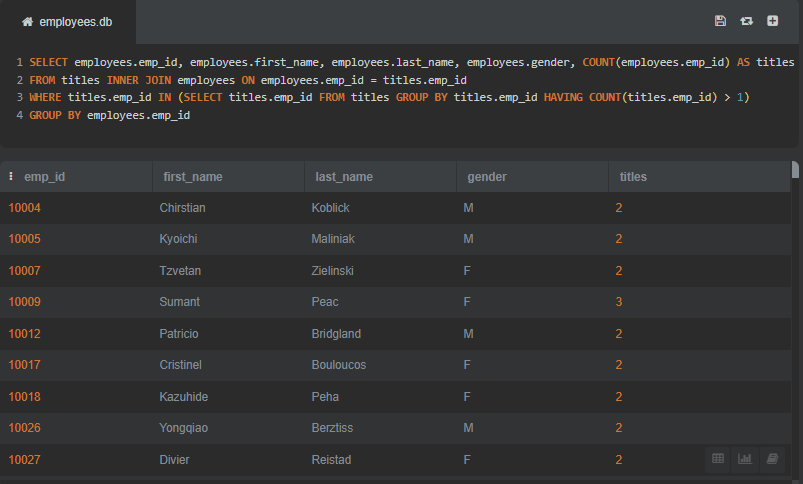
****

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

**Query:**

SELECT employees.emp\_id, employees.first\_name, employees.last\_name, employees.gender, COUNT(employees.emp\_id) AS titles FROM titles INNER JOIN employees ON employees.emp\_id = titles.emp\_id WHERE titles.emp\_id in (SELECT titles.emp\_id FROM titles GROUP BY titles.emp\_id HAVING COUNT(titles.emp\_id) > 1) GROUP BY employees.emp\_id

**Screenshot:**

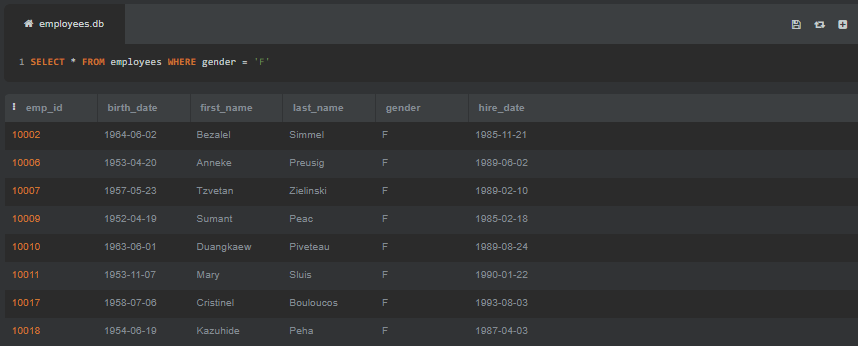
****

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

**Query:**

SELECT \* FROM employees WHERE gender = 'F'

**Screenshot:**

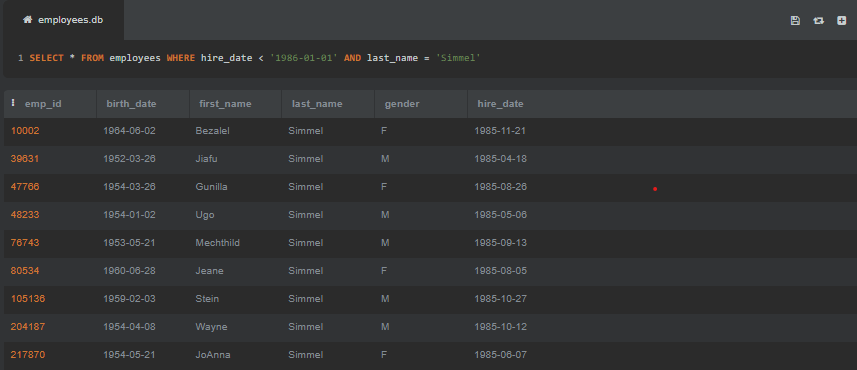
****

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

**Query:**

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

**Screenshot:**

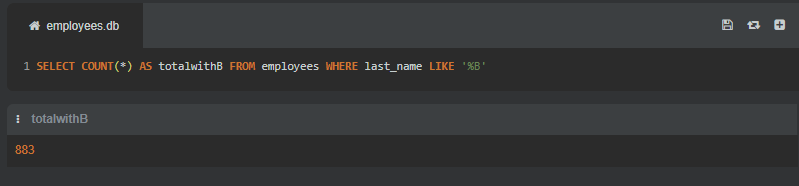
****

**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.**

**Query:**

SELECT COUNT(\*) AS totalwithB FROM employees WHERE last\_name LIKE '%B'

**Screenshot:**

****

**6. Create a new table called emp\_training with 3 columns.**

**Query:**

CREATE TABLE emp\_training (

trainer\_no INTEGER PRIMARY KEY AUTOINCREMENT,

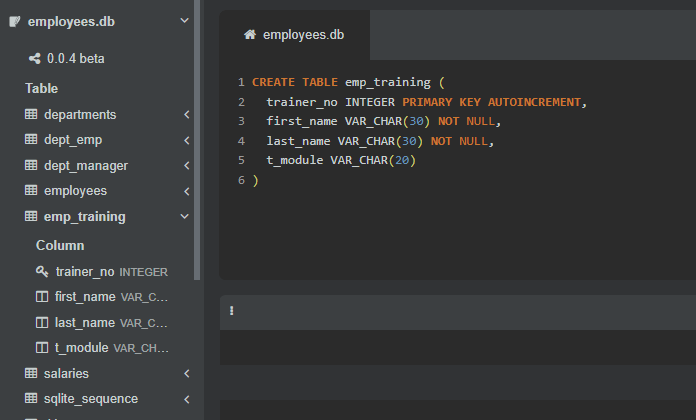
first\_name VAR\_CHAR(30) NOT NULL,

last\_name VAR\_CHAR(30) NOT NULL,

t\_module VAR\_CHAR(30) NOT NULL

)

**Screenshot:**

****

**7. Insert 2 new rows into the emp\_training table.**

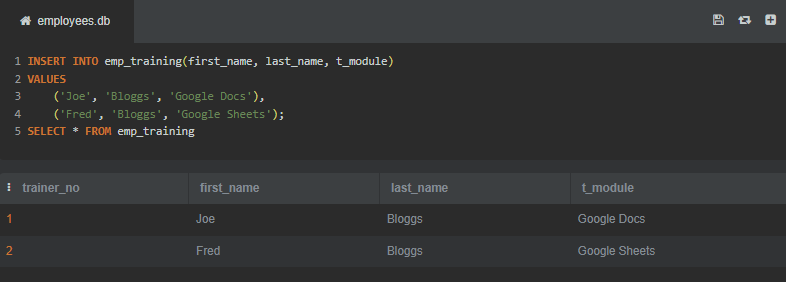
**Query:**

INSERT INTO emp\_training(first\_name, last\_name, t\_module) VALUES

(‘Joe’, ‘Bloggs’, ‘Google Docs’),

(‘Fred’, ‘Bloggs’, ‘Google Sheets’)

**Screenshot:**

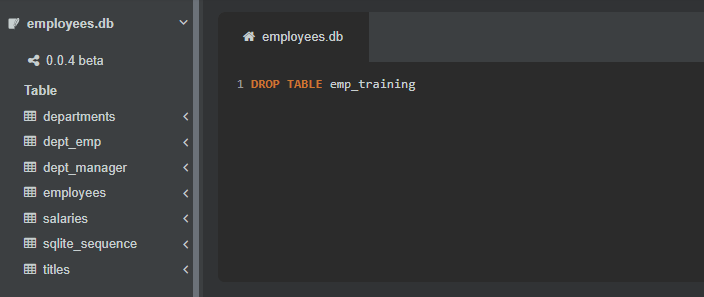
****

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

**Query:**

DROP TABLE emp\_training

**Screenshot:**

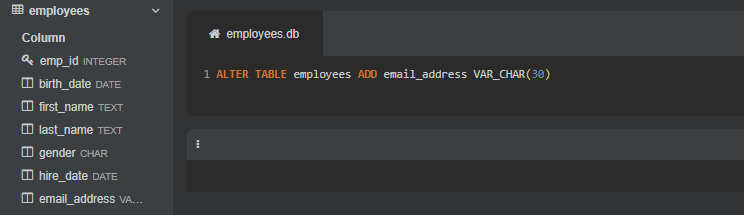
****

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

**Query:**

ALTER TABLE employees ADD email\_address VAR\_CHAR(30)

**Screenshot:**

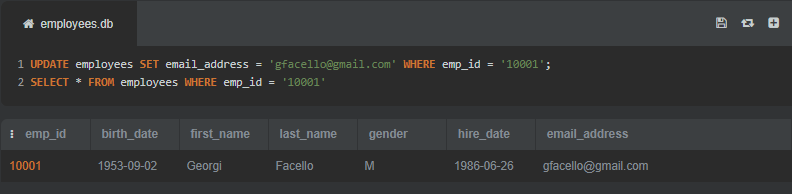
****

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

**Query:**

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

**Screenshot:**

****

**1.3 Database CA 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.**

**Query:**

SELECT employees.emp\_id, employees.first\_name, employees.last\_name, employees.gender, departments.dept\_name, dept\_manager.dept\_id,

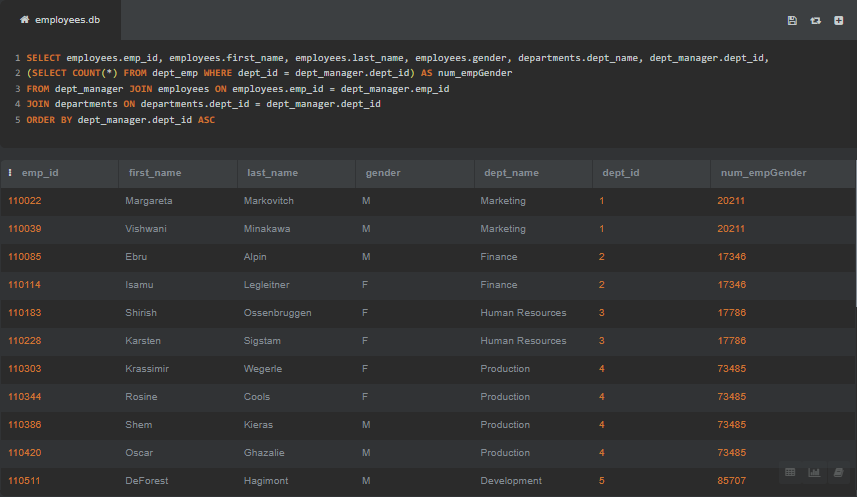
(SELECT COUNT(\*) FROM dept\_emp WHERE dept\_id = dept\_manager.dept\_id) AS num\_empGender

FROM dept\_manager JOIN employees ON employees.emp\_id = dept\_manager.emp\_id

JOIN departments ON departments.dept\_id = dept\_manager.dept\_id

ORDER by dept\_manager.dept\_id ASC

**Screenshot:**

****

**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.**

**Query:**

SELECT DISTINCT employees.gender, (SELECT AVG(salaries.salary)) as avg\_salary, titles.title

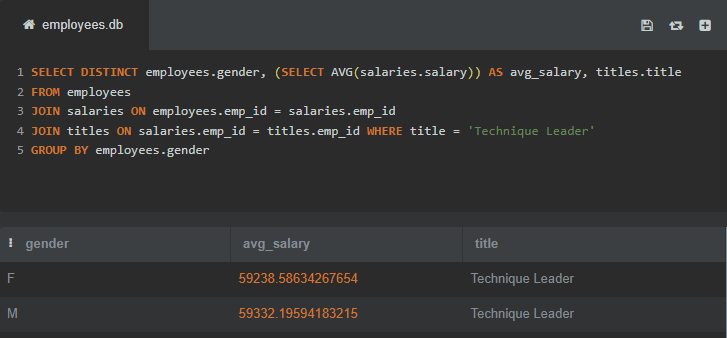
FROM employees

JOIN salaries ON employees.emp\_id = salaries.emp\_id

JOIN titles ON salaries.emp\_id = titles.emp\_id WHERE title = 'Technique Leader'

GROUP BY employees.gender

**Screenshot:**

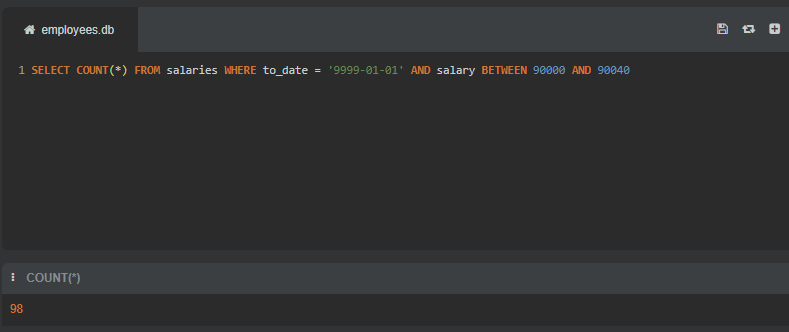
****

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

**Query:**

SELECT COUNT(\*) FROM salaries WHERE to\_date = ‘9999-01-01’ AND salary BETWEEN 90000 AND 90040

**Screenshot:**

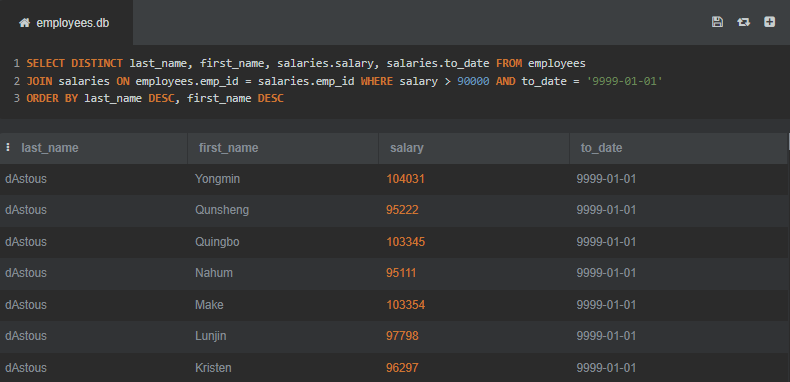
****

**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).**

**Query:**

SELECT DISTINCT last\_name, first\_name, salaries.salary, salaries.to\_date FROM employees JOIN salaries ON employees.emp\_id = salaries.emp\_id WHERE salary > 90000 AND to\_date = '9999-01-01' ORDER BY last\_name DESC, first\_name DESC

**Screenshot:**

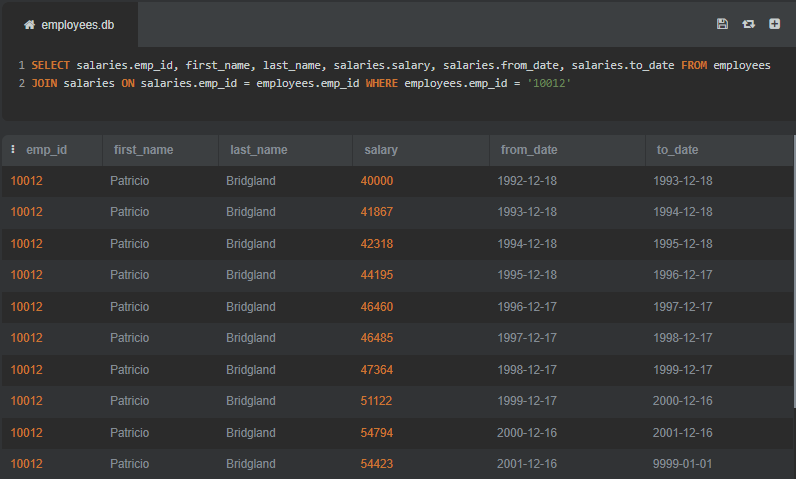
****

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

**Query:**

SELECT salaries.emp\_id, first\_name, last\_name, salaries.salary, salaries.from\_date, salaries.to\_date FROM employees JOIN salaries ON salaries.emp\_id = employees.emp\_id WHERE employees.emp\_id = ‘10012’

**Screenshot:**

****

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

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

**Answer:** Since there are four columns in salaries table so its degree is **4**.

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

**Answer:** The **emp\_id** column makes up the primary key.

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

**Answer:** The **from\_date** column makes up foreign key.

**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.**

**Answer:**

In SQL tables, sometimes we don’t have a unique key so we combine two or more columns to use as unique primary key. This is called composite key.

In dept\_emp, we have emp\_id and dept\_id. The emp\_id is used to identify employee from employees table while dept\_id is used to identify department in the departments table. Same case is with dept\_manager table where we have used dept\_id and emp\_id as composite key. In salaries we have emp\_id and from\_date as composite key. emp\_id is used to identify employee from employees table while from\_date is used in relation with titles table.

In titles table, we have used 3 columns as composite key, emp\_id, title, from\_date. emp\_id is used for employee; title is used to identify each role and from\_date is used in relation to salaries to identify employee’s starting date.