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

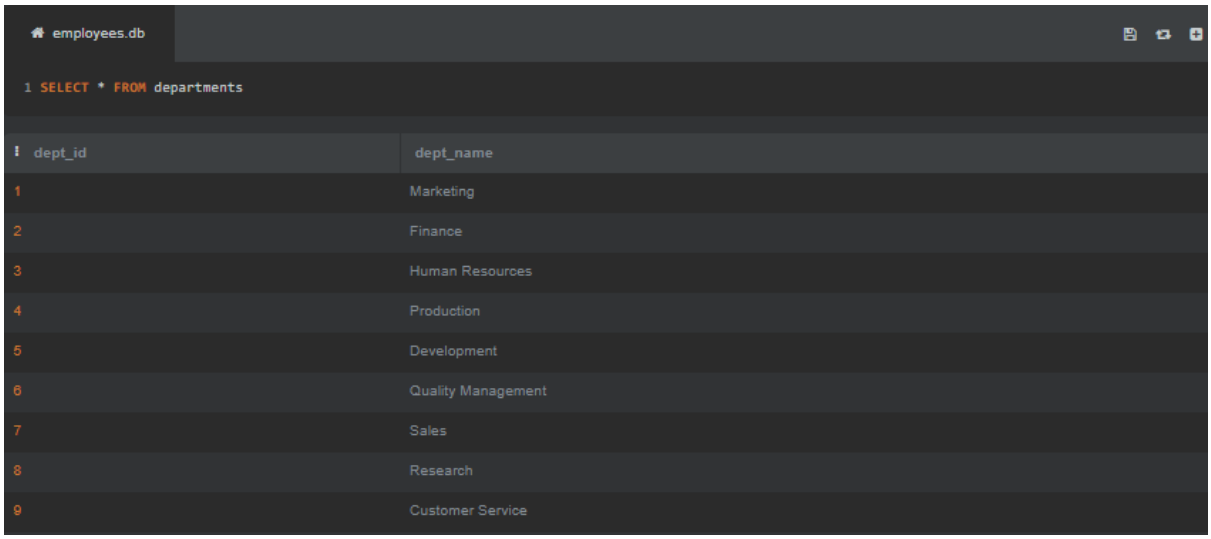
1.1 Databases CA Part 1

1. List all attributes present in the department's relation.

Query:

```
Select * From departments
```

Screenshot:



employees.db

```
1 SELECT * FROM departments
```

dept_id	dept_name
1	Marketing
2	Finance
3	Human Resources
4	Production
5	Development
6	Quality Management
7	Sales
8	Research
9	Customer Service

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:

employees.db

```
1 SELECT emp_id, first_name, last_name FROM employees
```

emp_id	first_name	last_name
10001	Georgi	Facello
10002	Bezalel	Simmel
10003	Parto	Bamford
10004	Chirstian	Koblick
10005	Kyoichi	Maliniak
10006	Anneke	Preusig
10007	Tzvetan	Zielinski

3. List all department titles present in the database.

Query:

```
SELECT dept_name FROM departments
```

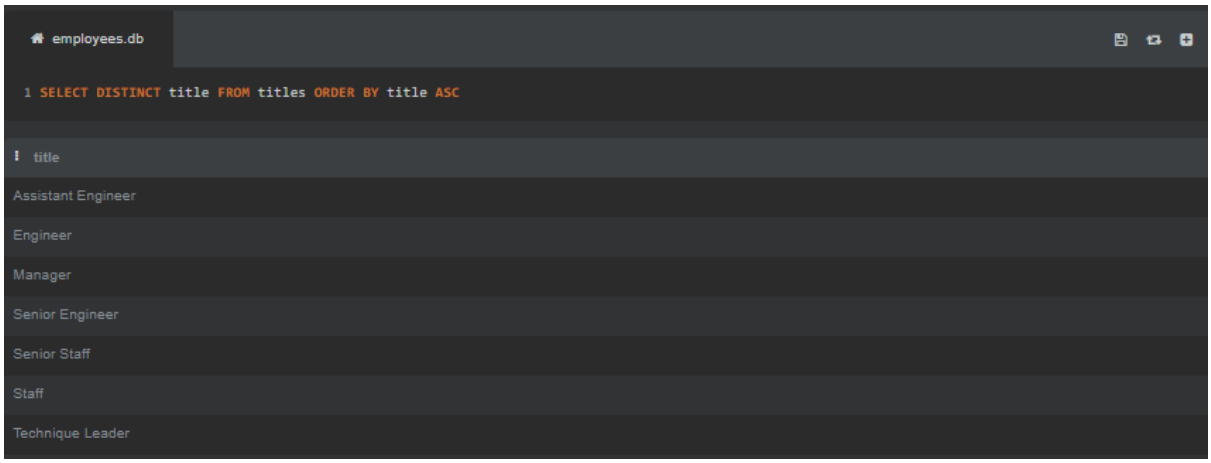
Screenshot:

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:

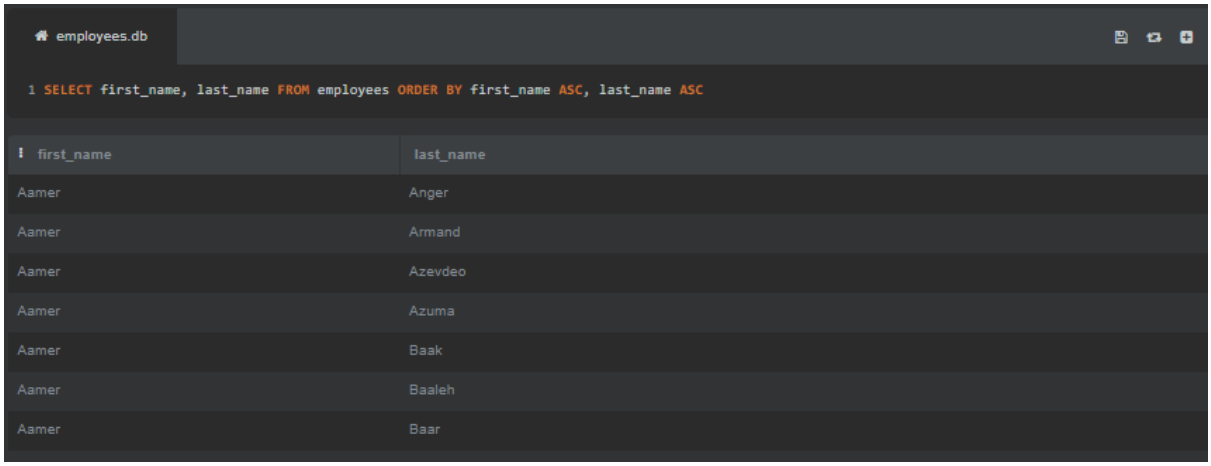


5. List all past/current employees' names ordered alphabetically in ascending order, i.e. first name and last name in alphabetical order.

Query:

```
SELECT first_name, last_name FROM employees ORDER BY first_name ASC, last_name ASC
```

Screenshot:



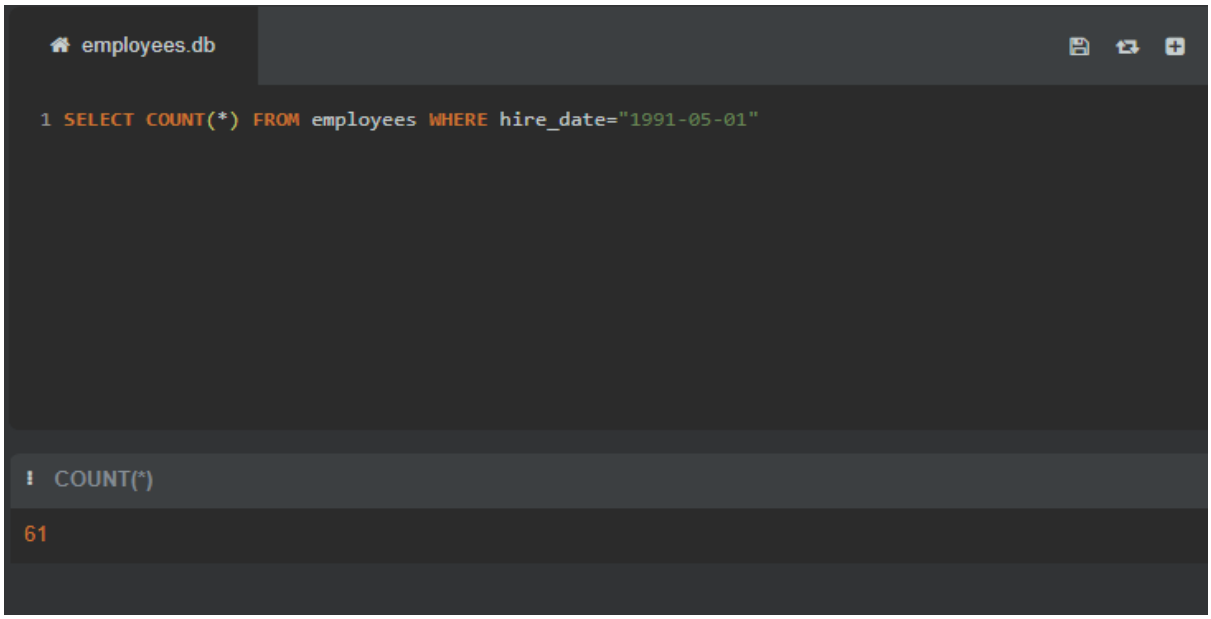
1.2 Database CA Part 2

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

Query:

```
SELECT COUNT(*) FROM employees WHERE hire_date="1991-05-01"
```

Screenshot:

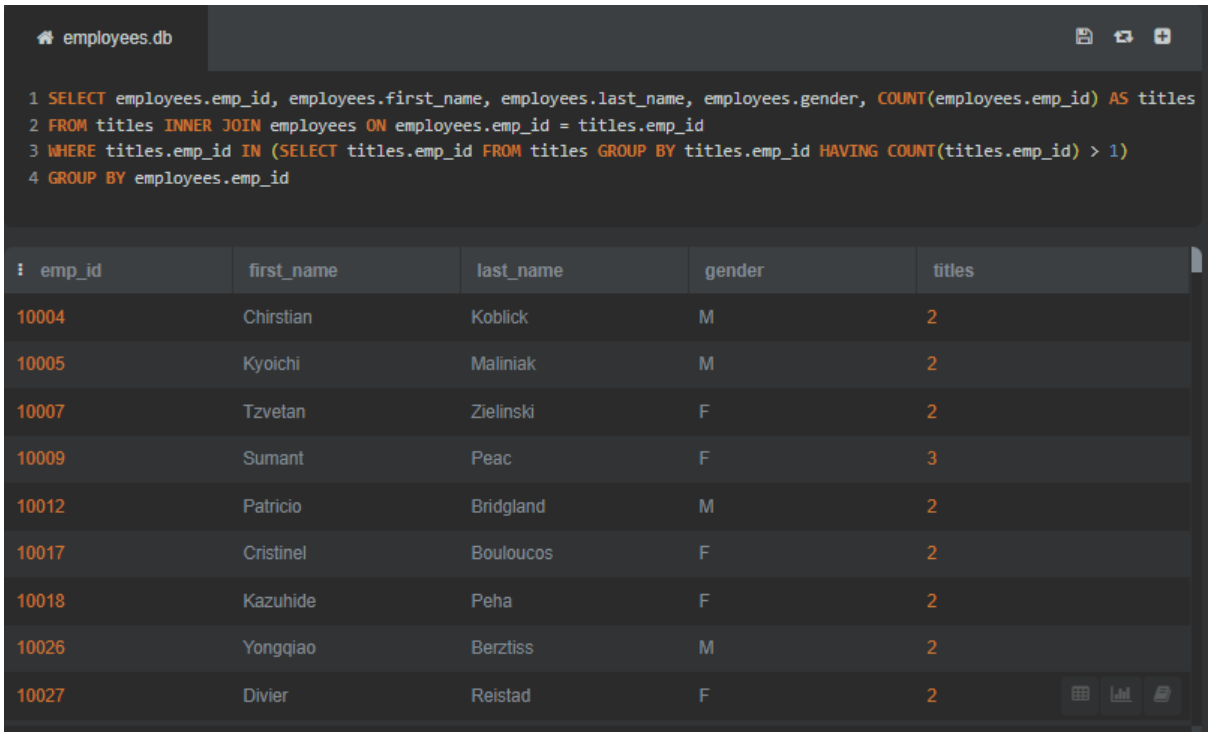


2. List all emp_no who have had strictly more than 2 titles and display the total number of the titles they have had.

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:



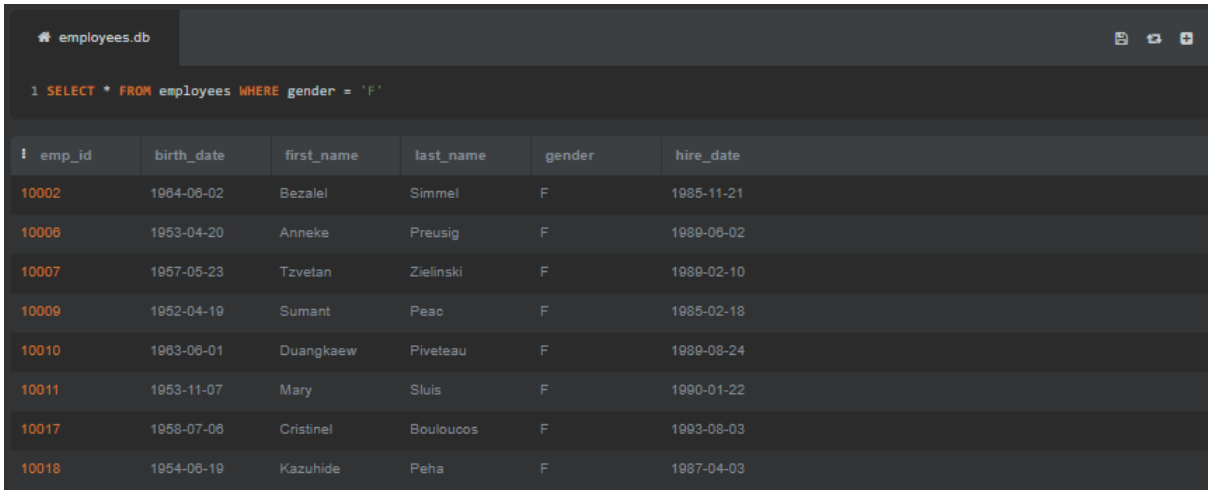
emp_id	first_name	last_name	gender	titles
10004	Chirstian	Koblick	M	2
10005	Kyoichi	Maliniak	M	2
10007	Tzvetan	Zielinski	F	2
10009	Sumant	Peac	F	3
10012	Patricio	Bridgland	M	2
10017	Cristinel	Bouloucos	F	2
10018	Kazuhide	Peha	F	2
10026	Yongqiao	Berztiss	M	2
10027	Divier	Reistad	F	2

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

Query:

```
SELECT * FROM employees WHERE gender = 'F'
```

Screenshot:



employees.db

```
1 SELECT * FROM employees WHERE gender = 'F'
```

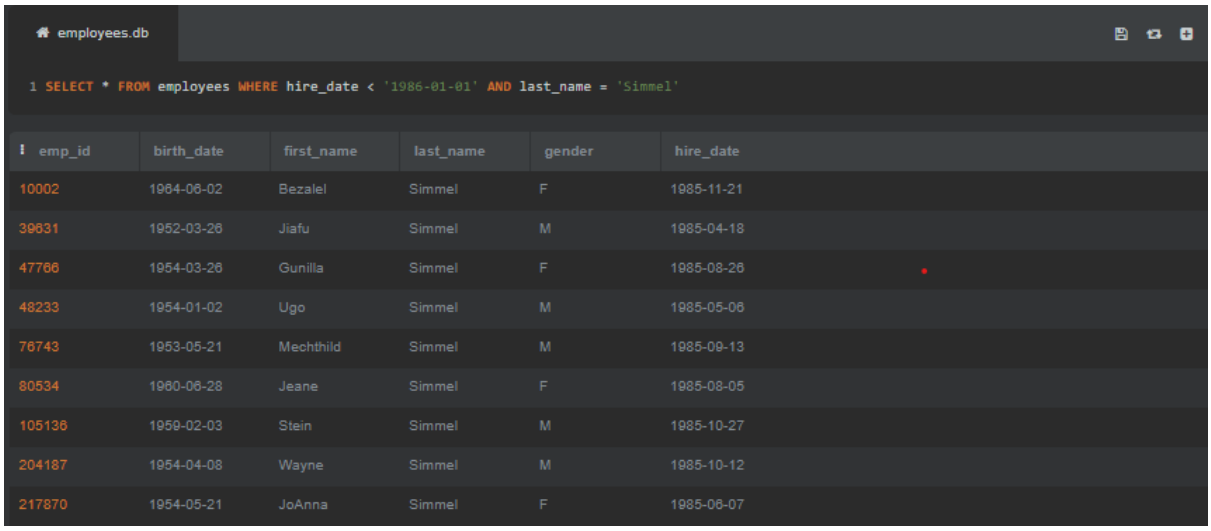
emp_id	birth_date	first_name	last_name	gender	hire_date
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
10006	1953-04-20	Anneke	Preusig	F	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
10009	1952-04-19	Sumant	Peac	F	1985-02-18
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24
10011	1953-11-07	Mary	Sluis	F	1990-01-22
10017	1958-07-06	Cristinel	Bouloucos	F	1993-08-03
10018	1954-06-19	Kazuhide	Peha	F	1987-04-03

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:



employees.db

```
1 SELECT * FROM employees WHERE hire_date < '1986-01-01' AND last_name = 'Simmel'
```

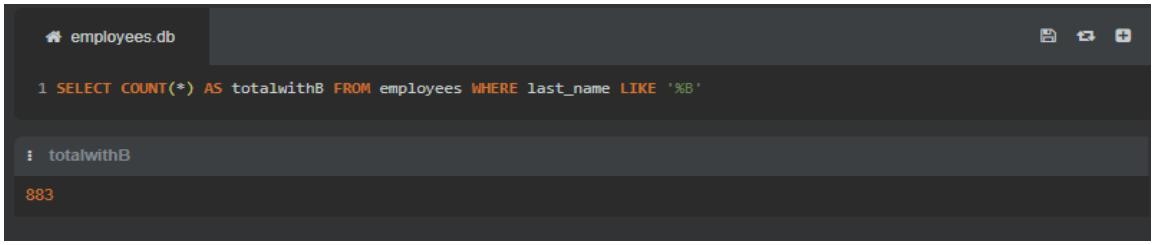
emp_id	birth_date	first_name	last_name	gender	hire_date
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
39631	1952-03-26	Jiafu	Simmel	M	1985-04-18
47766	1954-03-26	Gunilla	Simmel	F	1985-08-26
48233	1954-01-02	Ugo	Simmel	M	1985-05-06
76743	1953-05-21	Mechthild	Simmel	M	1985-09-13
80534	1960-06-28	Jeane	Simmel	F	1985-08-05
105136	1959-02-03	Stein	Simmel	M	1985-10-27
204187	1954-04-08	Wayne	Simmel	M	1985-10-12
217870	1954-05-21	JoAnna	Simmel	F	1985-06-07

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:

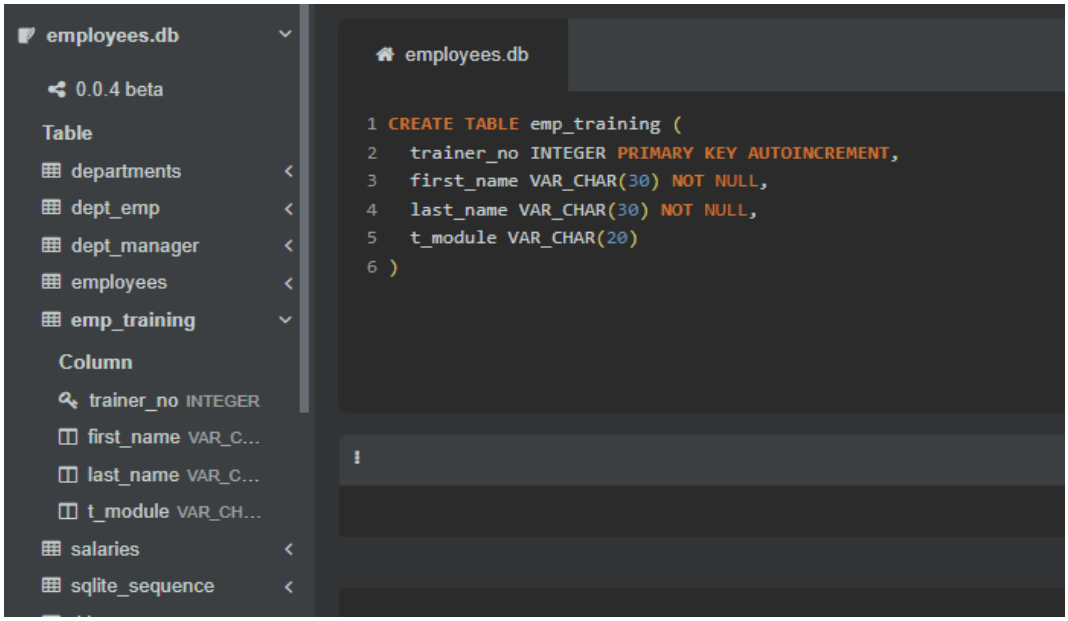


6. Create a new table called emp_training with 3 columns.

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')
```

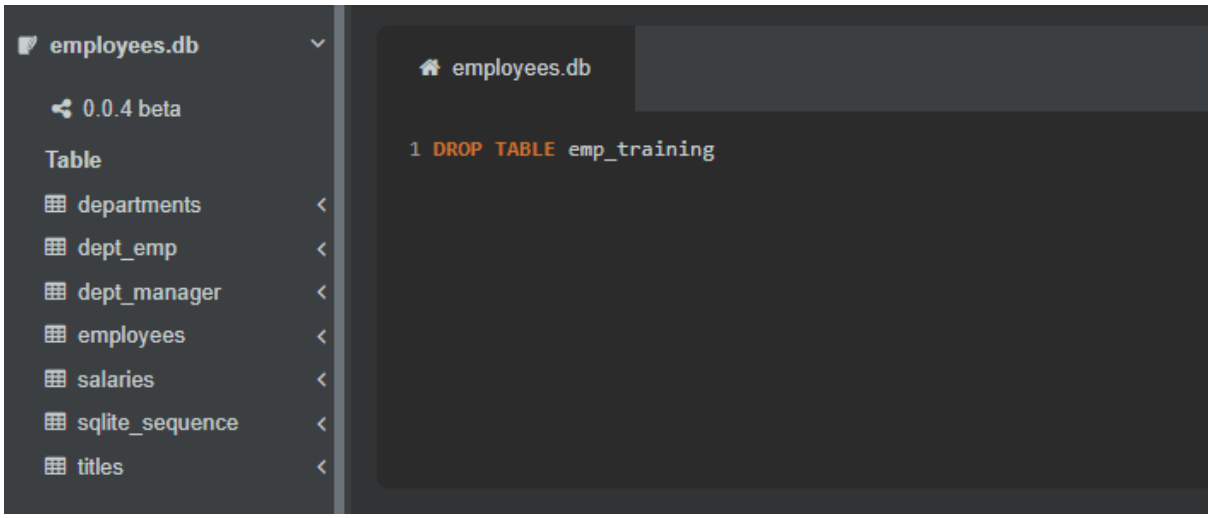
Screenshot:

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:

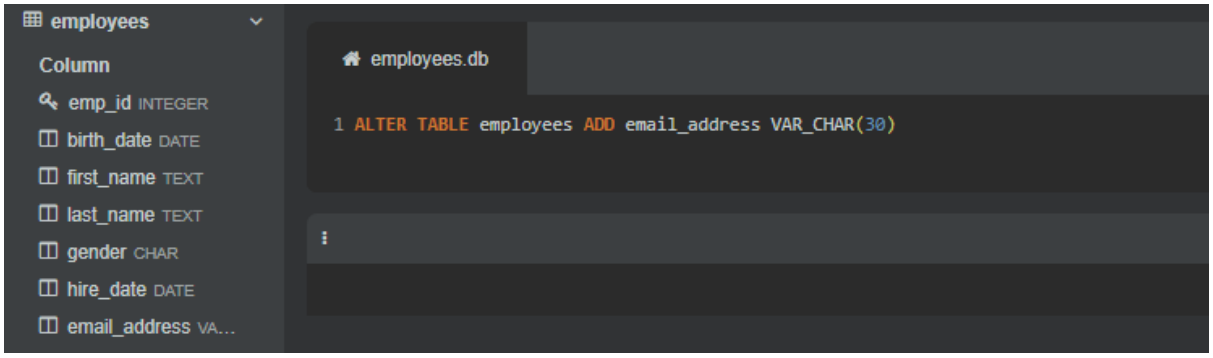


9. Alter the employees table to include an email_address field of type varchar(20).

Query:

```
ALTER TABLE employees ADD email_address VAR_CHAR(30)
```

Screenshot:

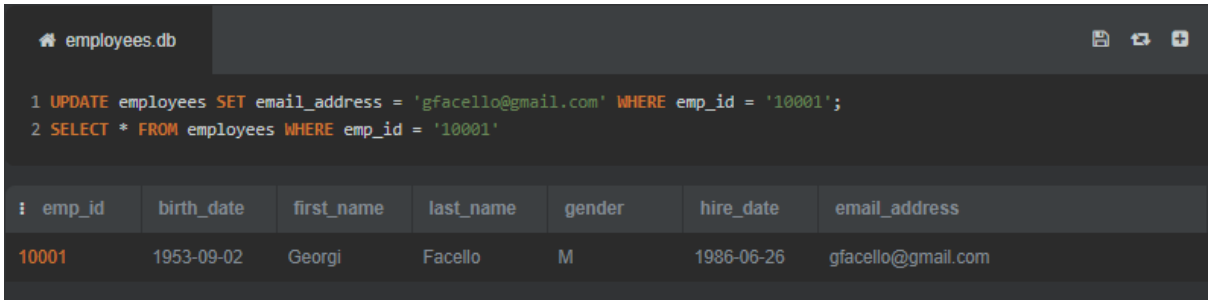


10. Update the email address of Georgi Facello to gfacello@gmail.com, where emp_no equals to 10001.

Query:

```
UPDATE employees SET email_address = ‘gfacello@gmail.com’ WHERE
emp_id = ‘10001’
```

Screenshot:



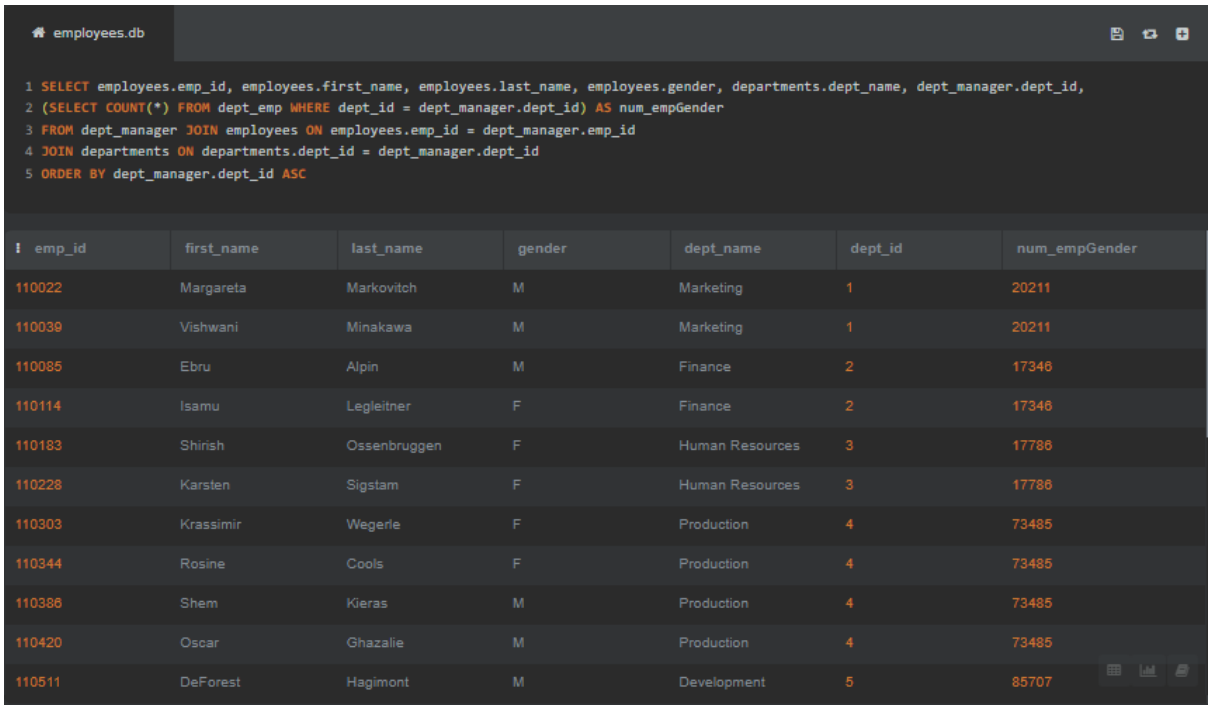
1.3 Database CA Part 3

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:



employees.db

```
1 SELECT employees.emp_id, employees.first_name, employees.last_name, employees.gender, departments.dept_name, dept_manager.dept_id,
2 (SELECT COUNT(*) FROM dept_emp WHERE dept_id = dept_manager.dept_id) AS num_empGender
3 FROM dept_manager JOIN employees ON employees.emp_id = dept_manager.emp_id
4 JOIN departments ON departments.dept_id = dept_manager.dept_id
5 ORDER BY dept_manager.dept_id ASC
```

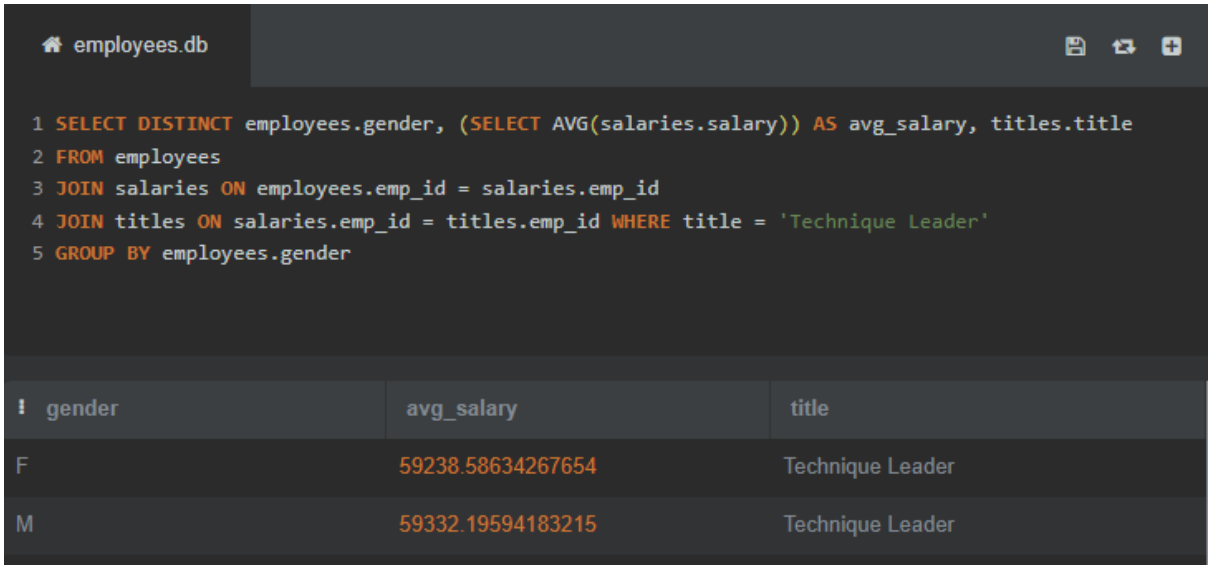
emp_id	first_name	last_name	gender	dept_name	dept_id	num_empGender
110022	Margareta	Markovitch	M	Marketing	1	20211
110039	Vishwani	Minakawa	M	Marketing	1	20211
110085	Ebru	Alpin	M	Finance	2	17346
110114	Isamu	Legleitner	F	Finance	2	17346
110183	Shirish	Ossenbruggen	F	Human Resources	3	17786
110228	Karsten	Sigstam	F	Human Resources	3	17786
110303	Krassimir	Wegerle	F	Production	4	73485
110344	Rosine	Cools	F	Production	4	73485
110386	Shem	Kieras	M	Production	4	73485
110420	Oscar	Ghazalie	M	Production	4	73485
110511	DeForest	Hagimont	M	Development	5	85707

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:



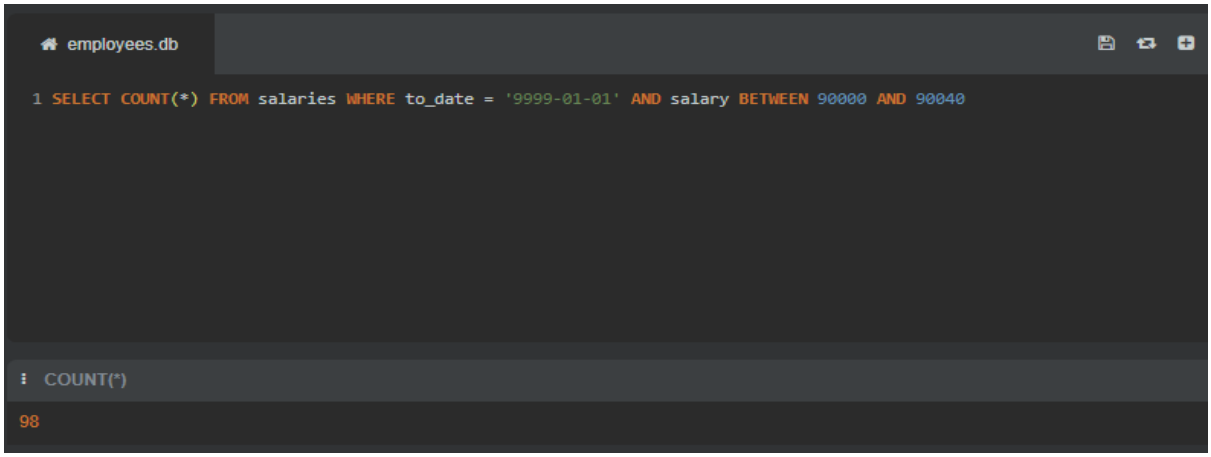
gender	avg_salary	title
F	59238.58634267654	Technique Leader
M	59332.19594183215	Technique Leader

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:

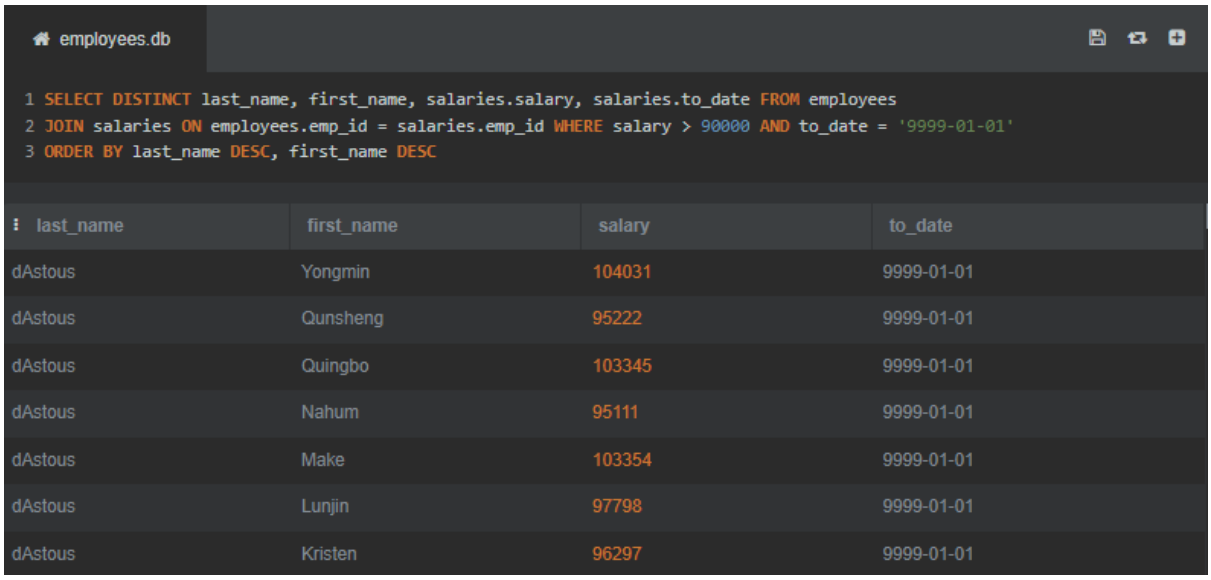


4. List all unique employees' last and first names (using GROUP BY method) that have a current salary (i.e., to_date equals to 9999-01-01) greater than 90000, outputting both names in descending order (sort by the last name first and then the first name) and also displaying their current salaries (using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



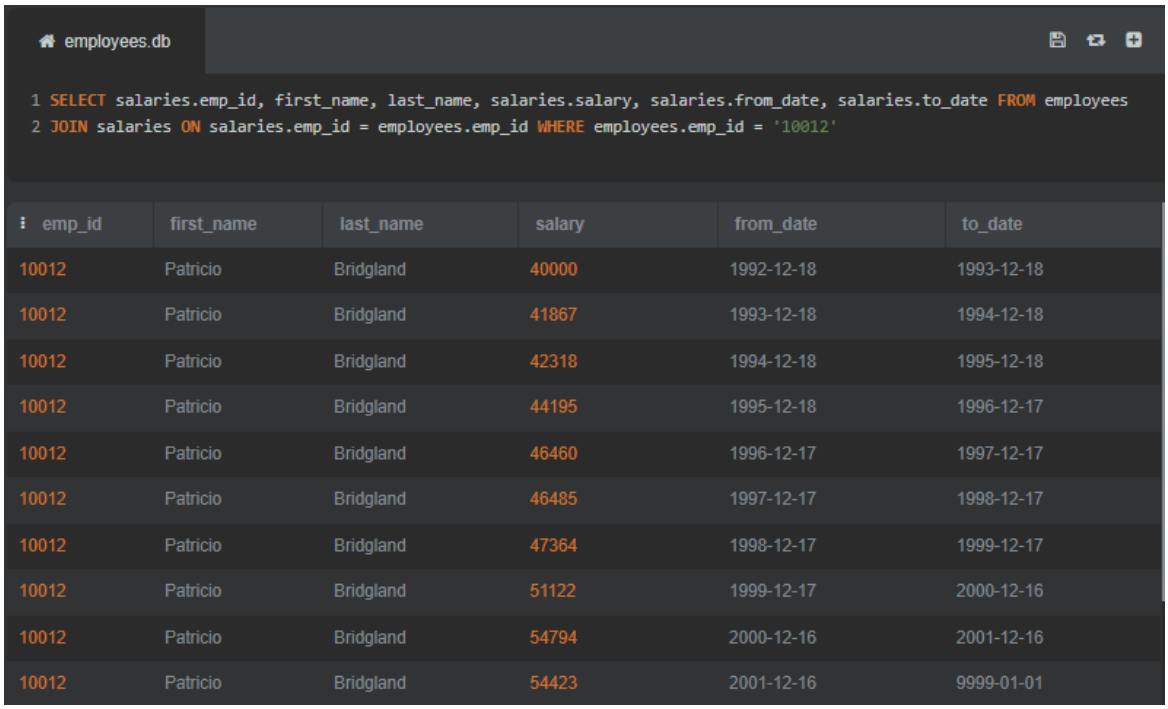
last_name	first_name	salary	to_date
dAstous	Yongmin	104031	9999-01-01
dAstous	Qunsheng	95222	9999-01-01
dAstous	Quingbo	103345	9999-01-01
dAstous	Nahum	95111	9999-01-01
dAstous	Make	103354	9999-01-01
dAstous	Lunjin	97798	9999-01-01
dAstous	Kristen	96297	9999-01-01

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:



emp_id	first_name	last_name	salary	from_date	to_date
10012	Patricio	Bridgland	40000	1992-12-18	1993-12-18
10012	Patricio	Bridgland	41867	1993-12-18	1994-12-18
10012	Patricio	Bridgland	42318	1994-12-18	1995-12-18
10012	Patricio	Bridgland	44195	1995-12-18	1996-12-17
10012	Patricio	Bridgland	46460	1996-12-17	1997-12-17
10012	Patricio	Bridgland	46485	1997-12-17	1998-12-17
10012	Patricio	Bridgland	47364	1998-12-17	1999-12-17
10012	Patricio	Bridgland	51122	1999-12-17	2000-12-16
10012	Patricio	Bridgland	54794	2000-12-16	2001-12-16
10012	Patricio	Bridgland	54423	2001-12-16	9999-01-01

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.