CCT College Dublin

Assessment Cover Page

Module Title:	Web Development/Data Base
Assessment Title:	CA2 (Cross Module CA with Web Dev)
Lecturer Name:	Mikhail Timofeev/ Aldana Louzan
Student Full Name:	FELIPE DA SILVA CUNHA
Student Number:	2022250
Assessment Due Date:	02/12/2022
Date of Submission:	04/12/2022

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.

Github:
FelipeCunha03/webdev_ca2

Summary

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. For the **Databases** module, you are required to write **SQL queries** that manipulate the data.
For the **Web Development** module, you are required to use a **JavaScript library** in an **HTML file with CSS** styling in order to output the results of each **SQL query** (see **Part 2**).

Part 1

1. List all attributes present in the departments relation.

```
select * from employees.departments;
```

Outpout:

dept_no	dept_name
d009	Customer Service
d005	Development
d002	Finance
d003	Human Resources
d001	Marketing
d004	Production
d006	Quality Management
d008	Research
d007	Sales
NULL	NULL

2. List all employee IDs of all past/current employees, their first and last names.(order by?)

```
select  emp_no, first_name, last_name  from employees.employees;
```

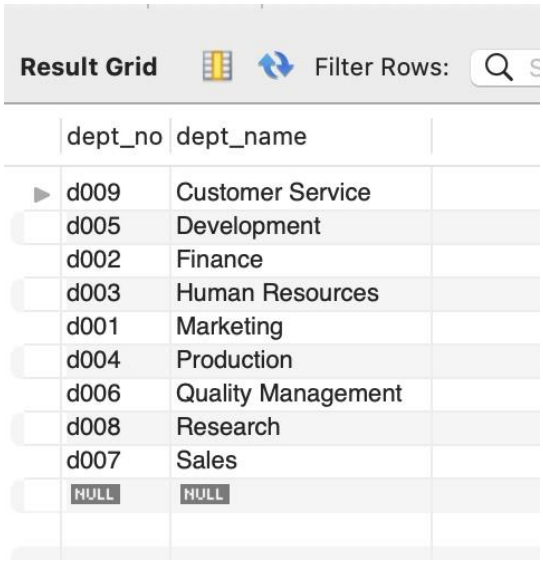
Output:

emp_no	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
10008	Saniya	Kalloufi
10009	Sumant	Peac
10010	Duangkaew	Piveteau
10011	Mary	Sluis
10012	Patricio	Bridgland
10013	Eberhardt	Terkki
10014	Berni	Genin

3. List all department titles present in the database.

```
select  * from employees departments
```

Output:



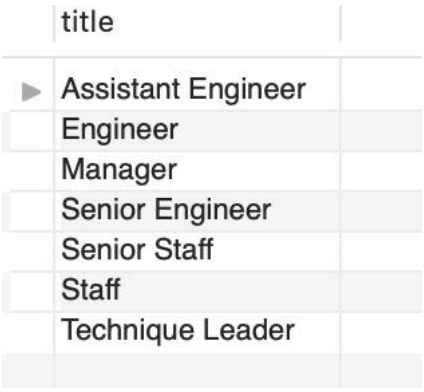
Result Grid Filter Rows:

dept_no	dept_name
d009	Customer Service
d005	Development
d002	Finance
d003	Human Resources
d001	Marketing
d004	Production
d006	Quality Management
d008	Research
d007	Sales
NULL	NULL

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

```
select distinct title from employees.titles
  order by title asc;
```

Output:



title
Assistant Engineer
Engineer
Manager
Senior Engineer
Senior Staff
Staff
Technique Leader

5. 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.employees
order by first_name, last_name asc;
```

Output:

first_name	last_name
Aamer	Anger
Aamer	Armand
Aamer	Azevdeo
Aamer	Azuma
Aamer	Baak
Aamer	Baaleh
Aamer	Baar
Aamer	Baba
Aamer	Bahl
Aamer	Bahl
Aamer	Bahr
Aamer	Basawa

Part 2

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

```
select count(hire_date) as amount_empl_started_1991_05_01 from employees.employees
where hire_date="1991-05-01";
```

Output:

amount_empl_started_1991_05_01
61

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

```
select a.emp_no,count(a.title) as num_of_titles from employees.titles a
group by a.emp_no
having count(a.title)>2;
```

Output;

emp_no	num_of_titles
12115	3
12069	3
12064	3
11951	3
11768	3
11760	3
11711	3
11543	3
11380	3
11371	3
11350	3
11322	3
11294	3
11106	3
11064	3
11027	3
11003	3

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

```
select * from employees.employees
where gender ="F";
```

Output:

emp_no	birth_date	first_name	last_name	gender	hire_date
10728	1958-11-17	Ugo	Maliniak	F	1989-12-09
10729	1961-05-24	Percy	Lorho	F	1986-04-20
10732	1962-04-22	Morris	Andreotta	F	1991-03-10
10733	1962-04-06	Sangeeta	Rodham	F	1996-02-20
10736	1962-07-09	Gurbir	Hellwagner	F	1991-09-26
10739	1961-02-11	Gor	Iacovou	F	1985-03-12
10741	1954-01-02	Sven	Huttel	F	1989-01-06
10745	1953-10-23	Poorav	Ranze	F	1987-05-26
10746	1958-12-29	Kolar	Usery	F	1990-08-31
10748	1964-04-12	Yuchang	Francisci	F	1990-08-14
10750	1953-06-14	Roddy	Demeyer	F	1991-08-04
10753	1958-09-30	Pradeep	Harbusch	F	1985-05-17
10758	1962-11-26	Fumiyo	Boyle	F	1988-02-06
10764	1953-01-12	Yunming	Beetstra	F	1989-11-27
10767	1954-02-28	Monique	Doering	F	1987-06-17

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

```
select * from employees.employees
 where hire_date < "1986-01-01"
 and last_name = "Simmel";
```

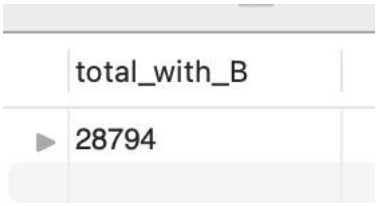
Output:

emp_no	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
237922	1959-04-12	Etsuo	Simmel	M	1985-03-16
246815	1964-08-08	Conrado	Simmel	F	1985-06-20
247973	1952-11-26	Xuejun	Simmel	F	1985-02-02

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

```
select  count(last_name) as total_with_B from  employees.employees
where last_name like  'B%';
```

Output:



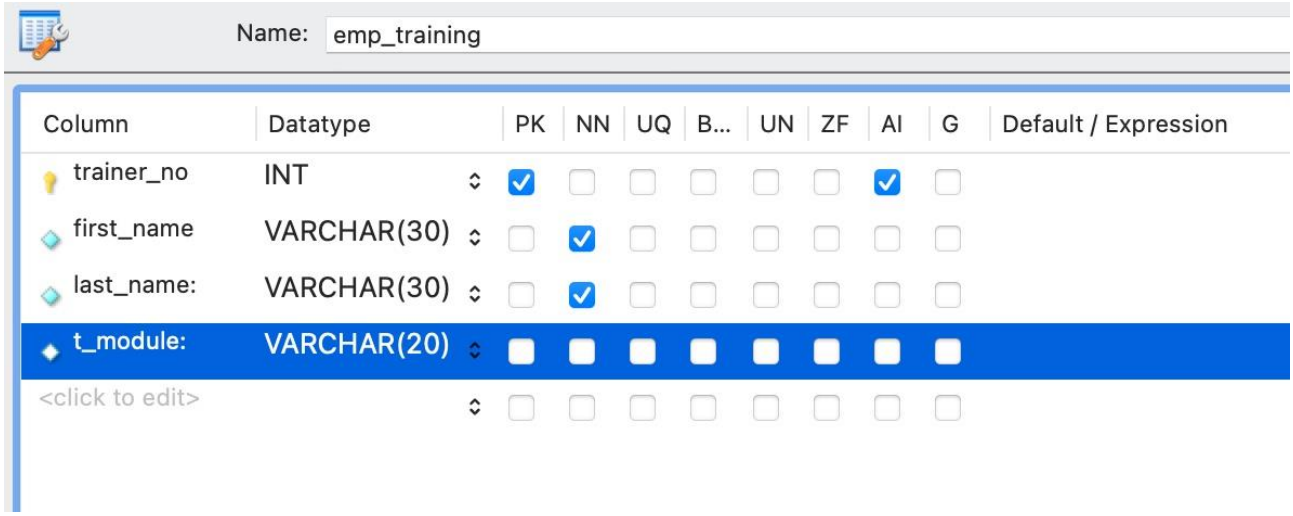
total_with_B
28794

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

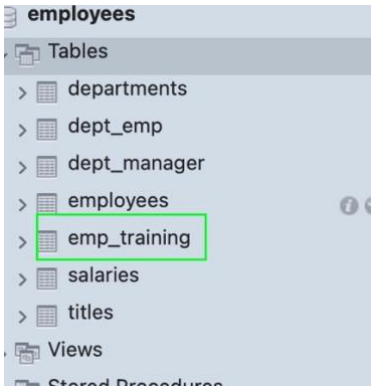
• **trainer_no**: this should be the primary key and is of type integer and is an auto-increment. • **first_name**: this data type is **varchar(30)** and should not be **NULL**

• **last_name**: this data type is **varchar(30)** and should not be **NULL**

• **t_module**: this data type is **varchar(20)**




```
CREATE TABLE `employees`.`emp_training` (
trainer_no INT   AUTO_INCREMENT,
first_name VARCHAR(30) NOT NULL,
last_name VARCHAR(20) NOT NULL,
t_module VARCHAR(20) NULL,
PRIMARY KEY (trainer_no));
```



7. Insert 2 new rows into the **emp_training** table:

Row 1: fname: **Joe**

lname: **Bloggs** module: **Google Docs**

Row 2:

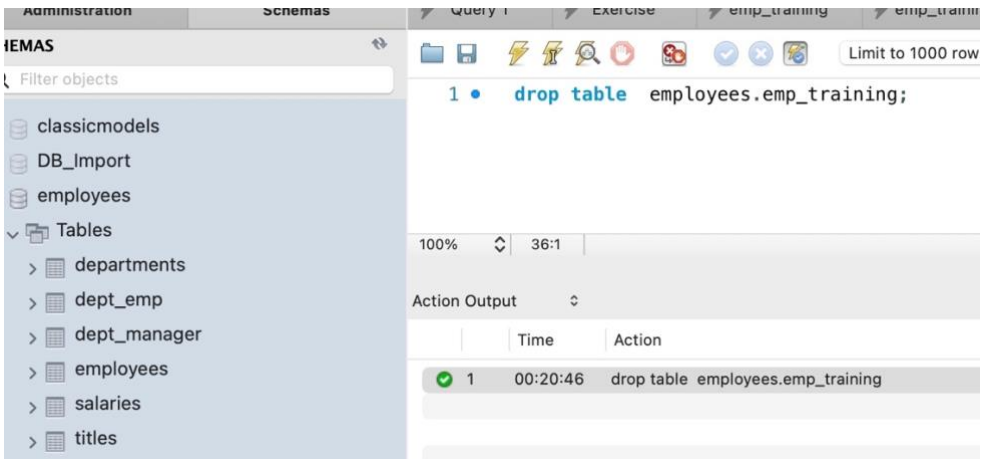
fname: **Fred**
lname: **Bloggs**
module: **Google Sheets**

```
insert into employees.emp_training (first_name,last_name,t_module) value ('Fred','Bloggs','Google Sheets'), ('Joe','Bloggs','Google Docs');
```

trainer_no	first_name	last_name	t_module
1	Joe	Bloggs	Google Docs
2	Fred	Bloggs	Google Sheets
NULL	NULL	NULL	NULL

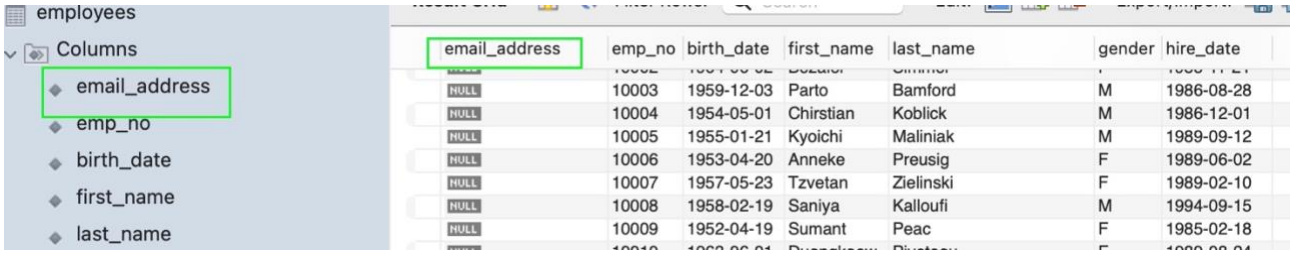
8. The organisation no longer wishes to record the employees training within the database. Therefore, delete the newly created emp_training table.

```
drop table  employees.emp_training;
```



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

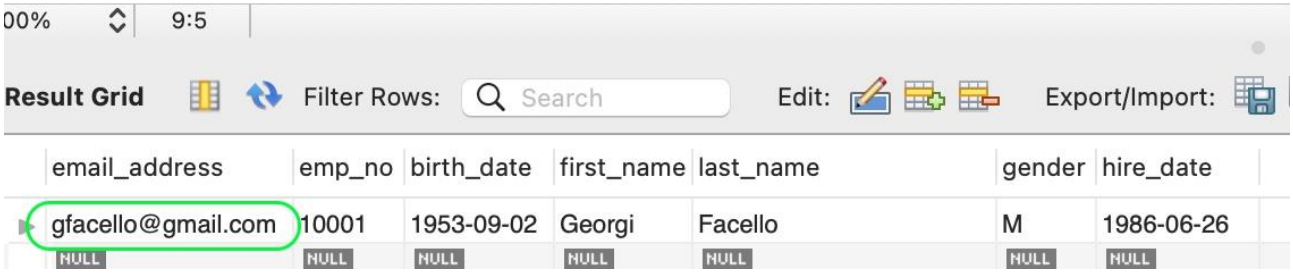
```
alter table  employees.employees add email_address varchar(20)FIRST;
```



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

```
update  employees.employees
set email_address='gfacello@gmail.com'
where emp_no=10001;
```

```
5 •  select * from   employees.employees
6    where emp_no=10001;
7
```



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 count(a.emp_no) as num_empGender, a.gender,b.dept_no from employees.employees a
inner join employees.dept_manager b
ON a.emp_no = b.emp_no
group by a.gender,b.dept_no
order by b.dept_no asc;
```

Output:

num_empGender	gender	dept_no
2	M	d001
1	M	d002
1	F	d002
2	F	d003
2	F	d004
2	M	d004
1	M	d005
1	F	d005
3	F	d006
1	M	d006
2	M	d007
1	M	d008
1	F	d008
3	F	d009
1	M	d009

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.

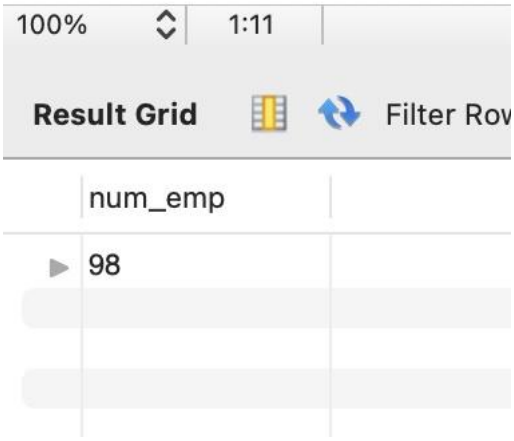
```
select  avg(c.salary) as avg_salary ,b.title, a.gender from employees.employees a
inner join employees.titles b
on a.emp_no = b.emp_no
inner join employees.salaries c
on c.emp_no = b.emp_no
where b.title = 'Technique Leader'
group by a.gender;
```

Result Grid Filter Rows: Search

avg_salary	title	gender
59332.1959	Technique Leader	M
59238.5863	Technique Leader	F

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

```
select count(a.emp_no) as num_emp  from employees.employees a
inner join employees.salaries b
ON a.emp_no = b.emp_no
where b.salary between 90000 and 90040
and b.to_date=  '9999-01-01';
```



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).

```
select  a.first_name, a.last_name, b.salary, b.to_date from employees.employees a
inner join employees.salaries b
on a.emp_no = b.emp_no
where b.to_date = '9999-01-01'
and b.salary > 90000
group by a.first_name,a.last_name
order by a.last_name desc ,a.first_name desc;
```

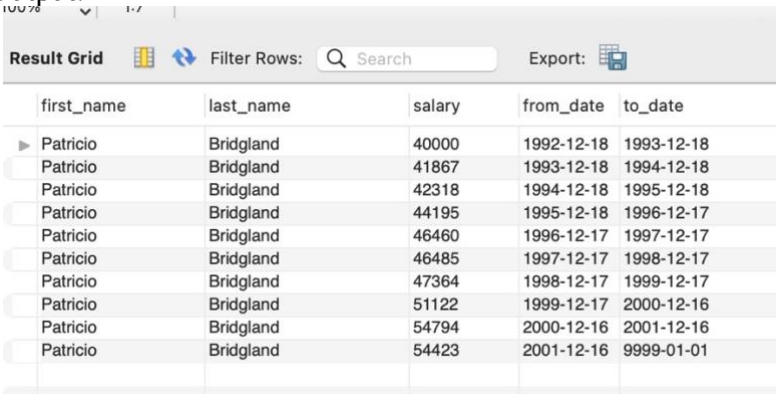
Output:

first_name	last_name	salary	to_date
Zvonko	Aamodt	51385	9999-01-01
Ziya	Aamodt	81522	9999-01-01
Zhiwei	Aamodt	66892	9999-01-01
Youpyo	Aamodt	71957	9999-01-01
Younwoo	Aamodt	56364	9999-01-01
Yoshimitsu	Aamodt	63733	9999-01-01
Yolla	Aamodt	80226	9999-01-01
Yishay	Aamodt	83354	9999-01-01
Yakkov	Aamodt	99713	9999-01-01
Woody	Aamodt	69521	9999-01-01
Weiyi	Aamodt	58550	9999-01-01
Vugranam	Aamodt	81932	9999-01-01
Vidar	Aamodt	91272	9999-01-01
Vasilii	Aamodt	62317	9999-01-01
Valeri	Aamodt	50585	9999-01-01
Ung	Aamodt	42189	9999-01-01
Uinam	Aamodt	58052	9999-01-01

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

```
select  a.first_name, a.last_name, b.salary, b.from_date, b.to_date from employees.employees a
inner join employees.salaries b
on b.emp_no = a.emp_no
where a.emp_no = 10012;
```

Output:



Result Grid Filter Rows: Search Export:

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

   Four.

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

   - **dept_emp**:**emp_id** and **dept_id** are PK together.
   - **salaries: emp_id** and **from_date** are PK together.
   - **employees:** emp_id;
   - **departaments: dept_id** and **dept_name** are PK together.
   - **dept_manager: emp_id** and **dept_id** are PK together.
   - **titles: emp_id**, **title**,**from_date** are PK together.

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

   - **dept_emp**: **emp_id** is FK but is PK with **dept_id together**;
   - **salaries**: **emp_id** is FK but is PK as well **with from_date** together;
   - **Employees:** no FK;
   - **departaments**: no FK;
   - **dept_manager**: **dept_id** and **emp_id** are FK but they are PK as well together.
   - **title**: **emp_id** is FK but It is PK as well with **from_date** together.

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

   dept_emp : emp_id and dept_id are PK uniquely identify together that because we cannot have the same employee working in more then one department.

   dept_manager: emp_id and **dept_id** are PK uniquely identify together because each employee belong just one department.

   salaries: emp_id and **from_date** are PK uniquely identify together because **from_date** can belong just one employee,

   titles: emp_id ,title and from_date are PK uniquely identify together because one title can be belong just one employee and just one period of time, in this case the title of employee can change that why from_date is PK as well.

References

https://www.tutorialspoint.com/sql/sql-order-by.htm

https://www.w3schools.com/sql/sql_count_avg_sum.asp

All the slides from class as well.