CCT College Dublin

Assessment Cover Page

Module Title:	Databases
Assessment Title:	CA1
Lecturer Name:	Aldana Louzan
Student Full Name:	Silvia Midori Shimabuko
Student Number:	2022199
Assessment Due Date:	04/12/2022
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.

Table of contents

1. Part 1 (20%)

1.1. List all attributes present in the department's relation. [4 marks]

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

1.2. List all employee IDs of all past/current employees, their first and last names. [4 marks]

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
10008	Saniya	Kalloufi
10009	Sumant	Peac

1.3. List all department titles present in the database. [4 marks]

SELECT dept_name FROM departments;

dept_name
Marketing
Finance
Human Resources
Production
Development
Quality Management
Sales
Research
Customer Service

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

SELECT DISTINCT title FROM titles ORDER BY title;

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

1.5. List all past/current employees' names ordered alphabetically in ascending order, i.e. first name and last name in alphabetical order. [4 marks]

SELECT first_name, last_name FROM employees ORDER BY first_name, last_name;

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

2. Part 2 (40%)

2.1. The number of all employees that started on 1991-05-01. [4 marks]

SELECT COUNT(*) AS quantity FROM employees WHERE hire_date = "1991-05-01";

quantity
61

2.2. 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]

SELECT t.emp_id, COUNT(t.emp_id) AS total_jobs FROM titles t GROUP BY t.emp_id HAVING COUNT(t.emp_id) > 2;

emp_id	total_jobs
10009	3
10066	3
10258	3
10451	3
10571	3
10612	3
10628	3
10634	3
11003	3

2.3. List female employees (past/current) together with all other relation attributes. [4 marks]

SELECT * FROM employees e WHERE e.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

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

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

2.5. 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]

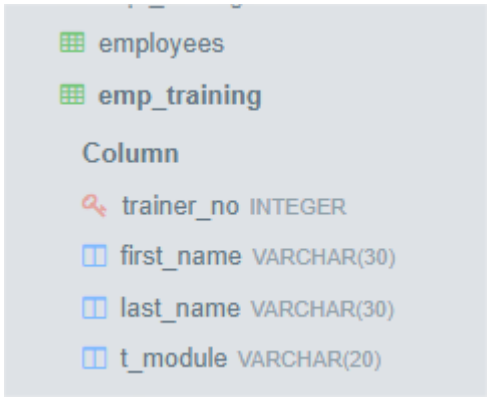
SELECT COUNT(*) AS total FROM employees WHERE last_name LIKE 'B%';

total
28794

2.6. 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]

```
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) NULL );
```



2.7. 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]

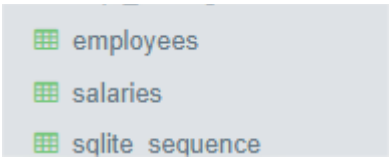
INSERT INTO 'emp_training' ('first_name', 'last_name', 't_module') VALUES ('Joe', 'Bloggs', 'Google Docs');

INSERT INTO 'emp_training' ('first_name', 'last_name', 't_module') VALUES ('Fred', 'Bloggs', 'Google Sheets');

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

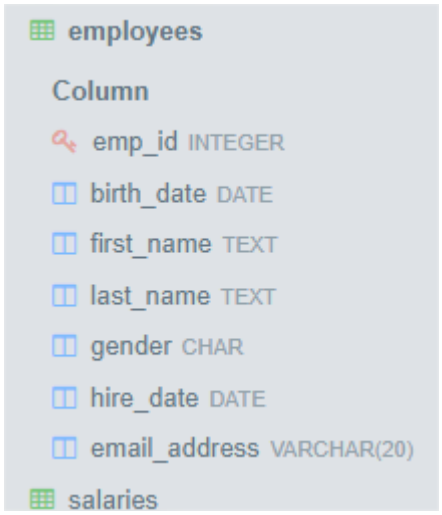
2.8. The organisation no longer wishes to record the employees training within the database. Therefore, delete the newly created emp_training table. [4 marks]

DROP TABLE IF EXISTS emp_training;



2.9. Alter the employees table to include an email_address field of type varchar(20). [4 marks]

ALTER TABLE 'employees' ADD COLUMN 'email_address' VARCHAR(20) NULL;



2.10. 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;

emp_id	email_address
10001	gfacello@gmail.com

3. Part 3 (40%)

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. [4 marks]

```
SELECT e.gender, COUNT(*) AS num_empGender, d.dept_id AS dept_no
FROM employees e, dept_manager dm, departments d
WHERE dm.emp_id = e.emp_id AND dm.dept_id = d.dept_id
GROUP by e.gender, dm.dept_id
ORDER BY d.dept_id;
```

gender	num_empGender	dept_no
F	2	4
M	2	4
F	1	5
M	1	5
F	3	6
M	1	6

3.2. 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]

```
SELECT e.gender, AVG(s.salary) AS avg_salary, t.title
FROM employees e
INNER join titles t ON t.emp_id = e.emp_id
inner join salaries s ON s.emp_id = e.emp_id
WHERE t.title = 'Technique Leader'
group by gender;
```

gender	avg_salary	title
F	59238.58634267654	Technique Leader
M	59332.19594183215	Technique Leader

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

```
SELECT COUNT(s.emp_id) FROM employees e
INNER JOIN salaries s ON s.emp_id = e.emp_id
WHERE s.to_date = "9999-01-01"
AND s.salary BETWEEN 90000 AND 90040 ;
```

COUNT(s.emp_id)
98

3.4. 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]

```
SELECT e.last_name, e.first_name, e.emp_id, s.salary FROM employees e
INNER JOIN salaries s ON s.emp_id = e.emp_id
WHERE s.to_date = "9999-01-01"
AND s.salary > 90000
GROUP BY e.last_name, e.first_name
ORDER BY UPPER(e.last_name) DESC, UPPER(e.first_name) DESC;
```

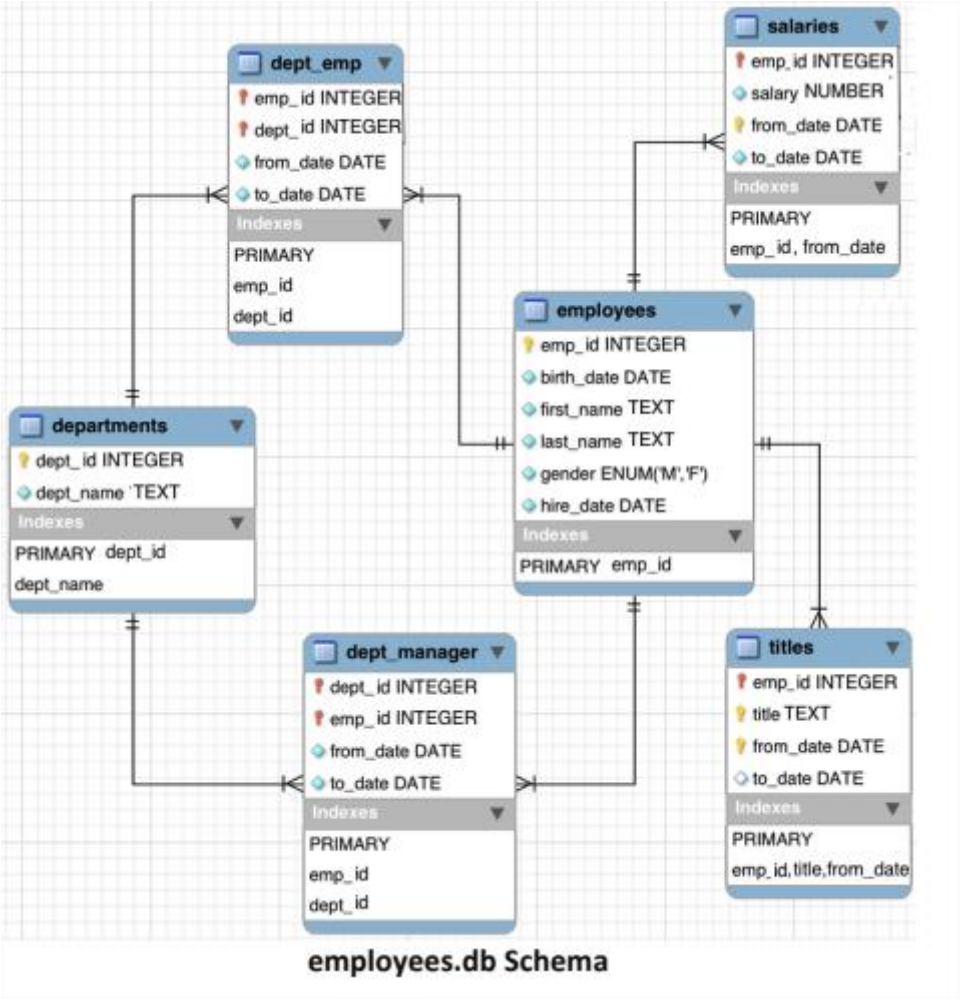
last_name	first_name	salary
Zykh	Zhongwei	99167
Zykh	Yongdong	93183
Zykh	Xumin	92320
Zykh	Toshimori	101401
Zykh	Seongbae	122466
Zykh	Rosalyn	120103

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

```
SELECT  e.first_name, e.last_name, s.FROM_date, s.to_date, s.salary
FROM employees e
INNER JOIN salaries s ON s.emp_id = e.emp_id
WHERE e.emp_id = 10012;
```

first_name	last_name	from_date	to_date	salary
Patricio	Bridgland	1992-12-18	1993-12-18	40000
Patricio	Bridgland	1993-12-18	1994-12-18	41867
Patricio	Bridgland	1994-12-18	1995-12-18	42318
Patricio	Bridgland	1995-12-18	1996-12-17	44195
Patricio	Bridgland	1996-12-17	1997-12-17	46460
Patricio	Bridgland	1997-12-17	1998-12-17	46485

3.6. In relation to the table named salaries. Answer in text:



a) What is the degree of this table?

A: The degree of this table is 4.

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

A: This table has a composite key, made by the combination of two columns emp_id and from_date.

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

A: In this table, emp_id is a foreign key

3.7. 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]

Table	Columns	Reason
dept_emp	emp_id / dept_id	This employee can belong to one or more departments and the departments can have one or more employees, but the same employee can't be registered more than one time for the same department.

dept_manager	**emp_id / dept_id**	**One or more employees can manage the department, but the employee can't be registered to manage the same department twice.**
salaries	**emp_id / from_date**	**The salaries table has the history of all salaries from a specific employee in different periods of time.**
Titles	**emp_id / title / from_date**	**One employee can have more than one title since the title and/or from_date are different from the prior registered**

References

W3Schools.(not specified) The SQL HAVING Clause. Available at https://www.w3schools.com/sql/sql_having.asp (28/11/2022)

W3Schools.(not specified) SQL INNER JOIN Keyword. Available at https://www.w3schools.com/sql/sql_join_inner.asp (28/11/2022)

W3Schools.(not specified) SQL Working With Dates. Available at https://www.w3schools.com/sql/sql_dates.asp (29/11/2022)

Explore database.(2017) What is degree of a relation in dbms. Available at https://www.exploredatabase.com/2017/02/what-is-degree-of-relation-table-in-db (02/12/2022)

Freecodecamp.(2022) Crow's Foot Notation – Relationship Symbols And How to Read Diagrams. Available at https://www.freecodecamp.org/news/crows-foot-notation-relationship-symbols-and-how-to-read-diagrams/ (02/12/2022)

Geeks for geeks.(2021). Composite Key in SQL. Available at https://www.geeksforgeeks.org/composite-key-in-sql/ (02/12/2022)