CCT College Dublin

Assessment Cover Page

Module Title:	Databases
Assessment Title:	Data Manipulation and Validation
Lecturer Name:	Aldana Louzan
Student Full Name:	Halil UGUR
Student Number:	2022389
Assessment Due Date:	04/12/2022 @23:59
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.

Contents

Abstract ***4***

Part 1 ..... ***5***

Part 2 ..... ***8***

Part 3 ..... ***12***

GitHub Repository ***15***

References ***15***

Pictures

Picture 1: Department's relations query result 5

Picture 2: Employees information query result 5

Picture 3: Department titles query result 6

Picture 4: Job titles query result 6

Picture 5: List of employees sorted as name and last name 7

Picture 6: Employee search size result 8

Picture 7: List of employees that have more than two more titles 8

Picture 8: List all tables 9

Picture 9: List employees that surname is "Simmel" 9

Picture 10: Employee size that surname start with B 9

Picture 11: Created employee training table 10

Picture 12: Added new data into training table 10

Picture 13: Employees training table was deleted 11

Picture 14: email_address column added to employees table 11

Picture 15: Added new email address into email_address column for 10001 employee 11

Picture 16: Manager size by gender on departments 12

Picture 17: Salary amount by gender 12

Picture 18: Size of employees by salary range 13

Picture 19: List of employees group by last name and first name 13

Picture 20: Salary list of the person whose employee id is 10012 14

Abstract

The Employee Sample Database (created by Fusheng Wang and Carlo Zaniolo at Siemens Corporate Research) is an extensive database of four million records spread across six tables, created for system testing purposes. This article was written to share the results of queries made on the employees database. The queries consist of three parts, and each is presented under a separate heading. In the first part, the aim is to query the existing tables and to obtain new, understandable data from the available data. In the second part, the aim is to add a new column to the existing table or to add a new table to the database. In the third part, data analysis is aimed at by running complex queries on tables.

Part 1

1. List all attributes present in the departments relation.

```
SELECT *
FROM departments AS d
        JOIN dept_manager AS dm ON d.dept_id = dm.dept_id
        JOIN dept_emp AS de ON d.dept_id = de.dept_id;
```

! dept_id	dept_name	dept_id	emp_id	from_date	to_date	emp_id	dept_id	from_date	to_date
1	Marketing	1	110022	1985-01-01	1991-10-01	10017	1	1993-08-03	9999-01-01
1	Marketing	1	110022	1985-01-01	1991-10-01	10055	1	1992-04-27	1995-07-22
1	Marketing	1	110022	1985-01-01	1991-10-01	10058	1	1988-04-25	9999-01-01
1	Marketing	1	110022	1985-01-01	1991-10-01	10108	1	1999-12-06	2001-10-20
1	Marketing	1	110022	1985-01-01	1991-10-01	10140	1	1991-03-14	9999-01-01
1	Marketing	1	110022	1985-01-01	1991-10-01	10175	1	1988-09-24	1995-05-24
1	Marketing	1	110022	1985-01-01	1991-10-01	10208	1	1995-02-05	1999-05-15
1	Marketing	1	110022	1985-01-01	1991-10-01	10228	1	1993-01-28	9999-01-01
1	Marketing	1	110022	1985-01-01	1991-10-01	10239	1	1996-05-04	9999-01-01

Picture 1: Department's relations query result

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

```
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
10010	Duangkaew	Piveteau

Picture 2: Employees information query result

3. List all department titles present in the database.

```
SELECT dept_name
FROM departments;
```

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

Picture 3: Department titles query result

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

```
SELECT DISTINCT title
FROM titles
ORDER BY title ASC;
```

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

Picture 4: Job titles query result

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
ORDER BY first_name ASC, last_name ASC;
```

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

Picture 5: List of employees sorted as name and last name

Part 2

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

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

! COUNT(*)
61

Picture 6: Employee search size result

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 e.emp_id, COUNT(e.emp_id) AS title_size
FROM employees AS e
         JOIN titles AS t ON e.emp_id = t.emp_id
GROUP BY e.emp_id
HAVING COUNT(e.emp_id) > 2;
```

! emp_id	title_size
10009	3
10066	3
10258	3
10451	3
10571	3
10612	3
10628	3
10634	3
11003	3
11027	3

Picture 7: List of employees that have more than two more titles

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

```
SELECT *
FROM employees AS e
         JOIN titles AS t ON e.emp_id = t.emp_id
         JOIN dept_emp AS de ON de.emp_id = e.emp_id
         JOIN dept_manager AS dm ON dm.emp_id = e.emp_id
         JOIN departments AS d ON d.dept_id = de.dept_id OR d.dept_id =
dm.dept_id
         JOIN salaries AS s ON s.emp_id = e.emp_id
WHERE gender = "F";
```

! emp...	bir...	fir...	las...	gen...	hir...	e...	title	fro...	to...	e...	de...	fro...	to...	de...	e...	fro...	to...	de...	de...	e...	sal...	fro...	to...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	52070	198...	198...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	55297	198...	198...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	55767	198...	198...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	58800	198...	198...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	62669	198...	199...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	62593	199...	199...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	63447	199...	199...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	67322	199...	199...
110114	195...	Isamu	Leg...	F	198...	110...	Ma...	198...	999...	110...	2	198...	999...	2	110...	198...	999...	2	Fin...	110...	67761	199...	199...

Picture 8: List all tables

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

```
SELECT *
FROM employees
WHERE last_name = "Simmel"
  AND hire_date < "1986-01-01";
```

! 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

Picture 9: List employees that surname is "Simmel"

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.

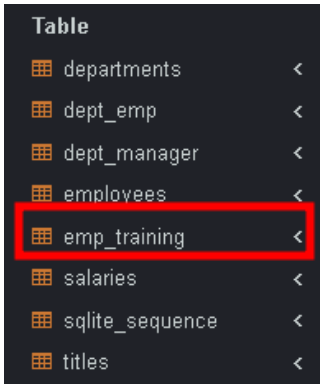
```
SELECT COUNT(*) AS "total with B"
FROM employees
WHERE last_name LIKE "B%";
```

! total with B
28794

Picture 10: Employee size that surname start with B

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)

```
DROP TABLE IF EXISTS emp_training;
CREATE TABLE IF NOT EXISTS emp_training
(
    trainer_no INTEGER PRIMARY KEY AUTOINCREMENT,
    first_name TEXT NOT NULL,
    last_name TEXT NOT NULL,
    t_module TEXT
);
```



Picture 11: Created employee training table

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

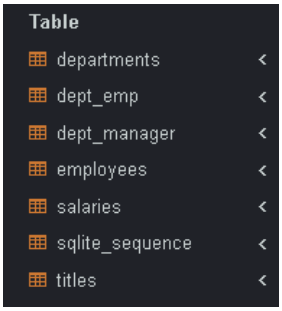
```
INSERT INTO emp_training (first_name, last_name, t_module)
VALUES ('Joe', 'Bloggs', 'Google Docs');

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

Picture 12: Added new data into training table

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

```
DROP TABLE IF EXISTS emp_training;
```



Picture 13: Employees training table was deleted

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

```
ALTER TABLE employees
    ADD COLUMN email_address VARCHAR(20);
```

emp_id	birth_date	first_name	last_name	gender	hire_date	email_address
10001	1953-09-02	Georgi	Facello	M	1986-06-26	NULL
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	NULL
10003	1959-12-03	Parto	Bamford	M	1986-08-28	NULL
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01	NULL
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12	NULL

Picture 14: email_address column added to employees table

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	birth_date	first_name	last_name	gender	hire_date	email_address
10001	1953-09-02	Georgi	Facello	M	1986-06-26	gfacello@gmail.com

Picture 15: Added new email address into email_address column for 10001 employee

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

! dept_id	gender	num_empGender
1	M	2
2	F	1
2	M	1
3	F	2
4	F	2
4	M	2
5	F	1
5	M	1
6	F	3
6	M	1
7	M	2
8	F	1
8	M	1
9	F	3
9	M	1

Picture 16: Manager size by gender on departments

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 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;
```

! gender	avg_salary	title
F	59238.58634267654	Technique Leader
M	59332.19594183215	Technique Leader

Picture 17: Salary amount by gender

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(*)
FROM employees AS e
         JOIN salaries AS s ON e.emp_id = s.emp_id
WHERE CURRENT_DATE < s.to_date
  AND s.salary BETWEEN 90000 AND 90040;
```

COUNT(*)
98

Picture 18: Size of employees by salary range

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 displaying their current salaries (using the INNER JOIN method).

```
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;
```

last_name	first_name	salary
dAstous	Yongmin	104031
dAstous	Qunsheng	95222
dAstous	Quingbo	103345
dAstous	Nahum	95111
dAstous	Make	103354
dAstous	Lunjin	97798
dAstous	Kristen	96297
dAstous	Kaijung	105257
dAstous	Huai	90939
dAstous	Honglan	110039
dAstous	Hatsukazu	94203
dAstous	Etsuo	95371
dAstous	Elgin	96769
dAstous	Chrisa	107145
dAstous	Bogdan	113940
dAstous	Basil	92484
dAstous	Adel	95810
Zykh	Zhongwei	99167
Zykh	Yongdong	93183

Picture 19: List of employees group by last name and first name

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

```
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;
```

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
Patricio	Bridgland	1998-12-17	1999-12-17	47364
Patricio	Bridgland	1999-12-17	2000-12-16	51122
Patricio	Bridgland	2000-12-16	2001-12-16	54794
Patricio	Bridgland	2001-12-16	9999-01-01	54423

Picture 20: Salary list of the person whose employee id is 10012

6. In relation to the table named salaries in Figure 1 above. Answer in text:
   - What is the degree of this table?
   - What column(s), if any, make(s) up the primary key?
   - What column(s), if any, make(s) up the foreign key?

Answer:

- The table of degree is 4. Because there are 4 attributes (columns) in the salaries table.
- There is no primary key. But there is composite key.
- emp_id is a foreign key for the salaries table

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:

A composite key is two or more columns in a table that are combined to uniquely identify each row in the table. The columns must be combined to create a unique identifier, but individually they do not guarantee uniqueness [1].

For dept_manager and dept_emp tables: There are emp_id and dept_id columns. These columns are likely to repeat the same data, but when combined they form a unique key.

For salaries and titlies tables: In this table, we need to consider the emp_id and dates. Because the employee may have different duties or salaries on different dates.

GitHub Repository

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

References

1) geeksforgeeks (2021). *Composite Key in SQL*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/composite-key-in-sql/

2) javatpoint (n.d.). *Learn SQL Tutorial - Javatpoint*. [online] www.javatpoint.com. Available at: https://www.javatpoint.com/sql-tutorial

3) sqltutorial (2020). *SQL Tutorial - Essential SQL for the Beginners*. [online] SQL Tutorial. Available at: https://www.sqltutorial.org/

4) w3schools (2019). *SQL Tutorial*. [online] W3schools.com. Available at: https://www.w3schools.com/