Julfri Emmil O. Caguiat
CSMC 206 Database Management System
Assignment #2: SQL Exercise Report
December 20, 2022

1. **Mechanisms used to describe the facts and information about the company:**

   First, drafted the **Conceptual Model** based on the facts and information given. Second, designed the **Logical Relational Model** based that shows all the entities and their attributes relevant to the database to be built. Third, formed the **Relational Schema** derived from the Logical Model for the *employee, department*, *worksForDept*, and *manages* entities. Lastly, designed the **Physical Relational Model** showing all the entities and their attributes with the appropriate domains or data types.

   SQL instructions for Creating the Tables of the Database:

   > CREATE TABLE employee (emp_name varchar(30) PRIMARY KEY NOT NULL, email varchar(30), phone_num varchar(30));

   > CREATE TABLE department (dept_name varchar(30) PRIMARY KEY NOT NULL);

   > CREATE TABLE worksForDept (emp_name varchar(30), dept_name varchar(30), position varchar(30));

   > CREATE TABLE manages (emp_name varchar(30), manager_name varchar(30));

2. **Information are represented as follows:**

   - Employee's name is represented as attribute *emp_name* of type *varchar* in *employee*, **worksForDept**, and **manages tables**.
   - Email is represented as attribute *email* of type *varchar* in **employee table**.
   - Phone number is represented as attribute *phone_num* of type *varchar* in **employee table**.
   - Department's name is represented as attribute *dept_name* of type *varchar* in **department** and **worksForDept tables**.
   - Employee's role is represented as attribute *position* of type *varchar* in **worksForDept table**.

- Manager ‘s name is represented as attribute *manager_name* of type *varchar* attribute in **manages table**.

SQL instructions for Populating the Database:

- Populate **employee table**:

  > INSERT INTO employee (emp_name, email, phone_num) values ('Deneen Willmon', 'deneen.willmon@gmail.com', '0908 787 8889'), ('Lashay Dann', 'lashay.dann@gmail.com', '0804 197 1660'), ('Kallie Jolliff', 'kallie@ymail.com', '0917 128 7291'), ('Starla Priebe', 'starlap@yahoo.com', '0650 338 2292'), ('Jannette Bassnight', 'jannette012@hotmail.com', '0778 490 2817');

- Populate **department table**:

  > INSERT INTO department (dept_name) values ('Accounting'), ('Maintenance'), ('Development');

- Populate **worksForDept table**:

  > INSERT INTO worksForDept (emp_name, dept_name, position) values ('Deneen Willmon', 'Maintenance', 'engineer'), ('Kallie Jolliff', 'Maintenance', 'technician'), ('Starla Priebe', 'Development', 'coder'), ('Lashay Dann', 'Accounting', 'head accountant'), ('Jannette Bassnight', 'Development', 'unknown');

- Populate **manages table**:

  > INSERT INTO manages (emp_name, manager_name) values ('Kallie Jolliff', 'Deneen Willmon'), ('Starla Priebe', 'Deneen Willmon');

3. **Queries to obtain the following information:**

   a) List of employees and the department they work for.

   > SELECT * FROM worksForDept;

   b) List of employees and their manager (show the employee even if they don’t have a manager).

   /* insert remaining employees with no managers then query the **manages table** */

   > INSERT INTO manages (emp_name) values ('Deneen Willmon'), ('Lashay Dann'), ('Jannette Bassnight');

> SELECT * FROM manages;

c) List of employees having an **'a'** in their name.

> SELECT emp_name FROM employee WHERE emp_name LIKE '%a%';

4. **Update the database to account for the following facts:**

SQL instructions for Updating the Database:

a) Change **manages** to **IsManagedBy table**.

> ALTER TABLE manages RENAME TO IsManagedBy;

b) Deneen Willmon has married. Update her name to Deneen Crawley.

/* set PRAGMA FOREIGN KEYS to ON to activate Foreign Keys. DROP the **IsManagedBy** and **worksForDept tables** then recreate and set their *emp_name* attributes as Foreign Keys that reference **employee table**'s *emp_name* with ON UPDATE CASCADE and ON DELETE CASCADE constraints, so that, any changes on *emp_name* attribute in the **employee table** will also reflect on **IsManagedBy** and **worksForDept tables**. */

> PRAGMA FOREIGN KEYS = ON;

> DROP TABLE IsManagedBy;

> CREATE TABLE IsManagedBy (emp_name varchar(30), manager_name varchar(30), FOREIGN KEY (emp_name) REFERENCES employee (emp_name) ON UPDATE CASCADE ON DELETE CASCADE);

> DROP TABLE worksForDept;

> CREATE TABLE worksForDept (emp_name varchar(30), dept_name varchar(30), position varchar(30), FOREIGN KEY (emp_name) REFERENCES employee (emp_name) ON UPDATE CASCADE ON DELETE CASCADE);

/* repopulate IsManagedBy and worksForDept table */

> INSERT INTO IsManagedBy (emp_name, manager_name) values ('Kallie Jolliff', 'Deneen Willmon'), ('Starla Priebe', 'Deneen Willmon'), ('Deneen Willmon', NULL), ('Lashay Dann', NULL), ('Jannette Bassnight', NULL);

> INSERT INTO worksForDept (emp_name, dept_name, position) values ('Deneen Willmon', 'Maintenance', 'engineer'), ('Kallie Jolliff', 'Maintenance', 'technician'), ('Starla Priebe', 'Development', 'coder'), ('Lashay Dann', 'Accounting', 'head accountant'), ('Jannette Bassnight', 'Development', 'unknown');

/* update *emp_name* in the **employee table** from Deneen Willmon to Deneen Crawley; *emp_name* in **IsManagedBy** and **worksForDept** will automatically be updated. */

> UPDATE employee SET emp_name = 'Deneen Crawley' WHERE emp_name = 'Deneen Willmon';

/* update *manager_name* in the **IsManagedBy table** from Deneen Willmon to Deneen Crawley */

> UPDATE IsManagedBy SET manager_name = 'Deneen Crawley' WHERE manager_name = 'Deneen Willmon';

c) Add another information whether the employee is currently active (TRUE) or not (FALSE). All employees are currently active.

/* add *current_emp* attribute to **employee table** then set all values to 'TRUE' */

> ALTER table employee ADD current_emp varchar(10);

> UPDATE employee SET current_emp = 'TRUE' WHERE current_emp IS NULL;

d) Jannette has retired.

> UPDATE employee Set current_emp = 'FALSE' WHERE emp_name = 'Jannette Bassnight';