**SQL ASSESSMENT**

CREATE DATABASE SCRIPT FOR EMPLOYEES DATABASE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DROP DATABASE IF EXISTS employees;

CREATE DATABASE IF NOT EXISTS employees; USE employees; SELECT 'CREATING DATABASE STRUCTURE' as 'INFO';

DROP TABLE IF EXISTS dept\_emp, dept\_manager, titles, salaries, employees, departments;

CREATE TABLE employees ( emp\_no INT NOT NULL, birth\_date DATE NOT NULL, first\_name VARCHAR(14) NOT NULL, last\_name VARCHAR(16) NOT NULL, gender ENUM ('M','F') NOT NULL, hire\_date DATE NOT NULL, PRIMARY KEY (emp\_no) );

CREATE TABLE departments ( dept\_no CHAR(4) NOT NULL, dept\_name VARCHAR(40) NOT NULL, PRIMARY KEY (dept\_no), UNIQUE KEY (dept\_name) );

CREATE TABLE dept\_manager ( emp\_no INT NOT NULL, dept\_no CHAR(4) NOT NULL, from\_date DATE NOT NULL, to\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, FOREIGN KEY (dept\_no) REFERENCES departments (dept\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no,dept\_no) );

CREATE TABLE dept\_emp ( emp\_no INT NOT NULL, dept\_no CHAR(4) NOT NULL, from\_date DATE NOT NULL, to\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, FOREIGN KEY (dept\_no) REFERENCES departments (dept\_no) ON DELETE CASCADE,PRIMARY KEY (emp\_no,dept\_no) );

CREATE TABLE titles ( emp\_no INT NOT NULL, title VARCHAR(50) NOT NULL, from\_date DATE NOT NULL, to\_date DATE, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no,title, from\_date) ) ;

CREATE TABLE salaries ( emp\_no INT NOT NULL, salary INT NOT NULL, from\_date DATE NOT NULL, to\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no, from\_date) ) ;

**Data**

INSERT INTO `departments` VALUES ('d001','Marketing'), ('d002','Finance'), ('d003','Human Resources'), ('d004','Production'),('d005','Development'), ('d006','Quality Management'),('d007','Sales'), ('d008','Research'),('d009','Customer Service'); INSERT INTO `employees` VALUES (10001,'1953-09-02','Georgi','Facello','M','1986-06-26'), (10002,'1964-06-02','Bezalel','Simmel','F','1985-11-21'), (10003,'1959-12-03','Parto','Bamford','M','1986-08-28'), (10004,'1954-05-01','Chirstian','Koblick','M','1986-12-01'), (10005,'1955-01-21','Kyoichi','Maliniak','M','1989-09-12'), (10006,'1953-04-20','Anneke','Preusig','F','1989-06-02'), (10007,'1957-05-23','Tzvetan','Zielinski','F','1989-02-10'), (10008,'1958-02-19','Saniya','Kalloufi','M','1994-09-15'), (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'), (10012,'1960-10-04','Patricio','Bridgland','M','1992-12-18'), (10013,'1963-06-07','Eberhardt','Terkki','M','1985-10-20'), (10014,'1956-02-12','Berni','Genin','M','1987-03-11'); INSERT INTO `dept\_emp` VALUES (10001,'d005','1986-06-26','9999-01-01'), (10002,'d007','1996-08-03','9999-01-01'), (10003,'d004','1995-12-03','9999-01-01'), (10004,'d004','1986-12-01','9999-01-01'), (10005,'d003','1989-09-12','9999-01-01'),(10006,'d005','1990-08-05','9999-01-01'), (10014,'d005','1993-12-29','9999-01-01');INSERT INTO `dept\_manager` VALUES (10013,'d001','1985-01-01','1991-10-01'), (10001,'d001','1991-10-01','9999-01-01'), (10002,'d002','1985-01-01','1989-12-17'), (10008,'d002','1989-12-17','9999-01-01'), (10012,'d003','1985-01-01','1992-03-21'), (10011,'d003','1992-03-21','9999-01-01'), (10014,'d004','1985-01-01','1988-09-09'), (10003,'d004','1988-09-09','1992-08-02'); INSERT INTO `salaries` VALUES (10001,60117,'1986-06-26','1987-06-26'), (10001,62102,'1987-06-26','1988-06-25'), (10002,66074,'1988-06-25','1989-06-25'), (10003,66596,'1989-06-25','1990-06-25'),(10004,66961,'1990-06-25','1991-06-25'), (10005,71046,'1991-06-25','1992-06-24'), (10006,74333,'1992-06-24','1993-06-24'), (10007,75286,'1993-06-24','1994-06-24'), (10008,75994,'1994-06-24','1995-06-24'); INSERT INTO `titles` VALUES (10001,'Senior Engineer','1986-06-26','9999-01-01'), (10002,'Staff','1996-08-03','9999-01-01'), (10003,'Senior Engineer','1995-12-03','9999-01-01'), (10004,'Engineer','1986-12-01','1995-12-01'), (10004,'Senior Engineer','1995-12-01','9999-01-01'), (10005,'Senior Staff','1996-09-12','9999-01-01'), (10005,'Staff','1989-09-12','1996-09-12'), (10006,'Senior Engineer','1990-08-05','9999-01-01'), (10007,'Senior Staff','1996-02-11','9999-01-01'), (10007,'Staff','1989-02-10','1996-02-11'), (10008,'Assistant Engineer','1998-03-11','2000-07-31');

**Do not copy this data from here.**

**Table Join Model**

A diagram of a company

Description automatically generated

**Tasks**

**1. Create a SQL statement to list all managers and their titles. (revisit)**

**First attempt**

SELECT dm.emp\_no, e.first\_name, e.last\_name, t.title

FROM employees e

RIGHT JOIN titles t

ON e.emp\_no = t.emp\_no

RIGHT JOIN dept\_manager dm

ON e.emp\_no = dm. emp\_no;

**Second attempt**

SELECT dm.emp\_no, CONCAT(e.first\_name, ‘ ‘, e.last\_name) AS full\_name, t.title

FROM dept\_manager dm

LEFT JOIN employees e

ON dm.emp\_no = e.emp\_no

LEFT JOIN titles t

ON e.emp\_no = t.emp\_no

GROUP BY

dm.emp\_no,

e.first\_name,

e.last\_name,

t.title;

**Output**

A screenshot of a computer screen

Description automatically generated

**2. Create a SQL statement to show the salary of all employees and their department name.**

SELECT s.emp\_no, e.first\_name, e.last\_name, s.salary, d.dept\_name

FROM salaries s

INNER JOIN employees e

ON s.emp\_no = e.emp\_no

INNER JOIN dept\_emp de

ON e.emp\_no = de.emp\_no

INNER JOIN departments d

ON de.dept\_no = d.dept\_no;

**Second attempt**

Select dm.emp\_no, CONCAT(e.first\_name, ‘ ‘, e.last\_name) AS full\_name, s.salary, d.dept\_name, COALESCE(de.dept\_no, dm.dept\_no) AS dept\_no, s.from\_date, s.to\_date

From dept\_manager dm

LEFT JOIN departments d

ON d.dept\_no = dm.dept\_no

LEFT JOIN employees e

ON dm.emp\_no = e.emp\_no

LEFT JOIN salaries s

ON e.emp\_no = s.emp\_no

LEFT JOIN dept\_emp de

ON d.dept\_no = de.dept\_no

Union

SELECT s.emp\_no, CONCAT(e.first\_name, ‘ ‘, e.last\_name) AS full\_name, s.salary, d.dept\_name, COALESCE(de.dept\_no, dm.dept\_no) AS dept\_no, s.from\_date, s.to\_date

FROM employees e

LEFT JOIN salaries s

ON e.emp\_no = s.emp\_no

LEFT JOIN dept\_emp de

ON e.emp\_no = de.emp\_no

LEFT JOIN departments d

ON de.dept\_no = d.dept\_no

LEFT JOIN dept\_manager dm

ON dm.dept\_no = d.dept\_no;

**Outcome**

A screenshot of a computer screen

Description automatically generated

**3. Create a SQL statement to show the hire date and birth date who belongs to HR department**

SELECT e.emp\_no, e.hire\_date, e.birth\_date, d.dept\_name, de.to\_date

FROM employees e

LEFT JOIN dept\_emp de

ON e.emp\_no = de.emp\_no

LEFT JOIN departments d

ON de.dept\_no = d.dept\_no

WHERE d.dept\_name = ‘Human Resources’;

**Outcome**

A black screen with white text

Description automatically generated

**Code including past employees**

**Second attempt**

SELECT e.emp\_no, CONCAT(e.first\_name, ‘ ‘, e.last\_name) AS full\_name, e.hire\_date, e.birth\_date, COALESCE(de.dept\_no, dm.dept\_no) AS dept\_no, d.dept\_name

FROM employees e

LEFT JOIN dept\_emp de

ON e.emp\_no = de.emp\_no

LEFT JOIN dept\_manager dm

ON e.emp\_no = dm.emp\_no

LEFT JOIN departments d

ON d.dept\_no = COALESCE(de.dept\_no, dm.dept\_no)

WHERE d.dept\_name = ‘Human Resources’

OR

COALESCE (de.dept\_no, dm.dept\_no) = ‘d003’;

**Output (Final answer)**

A black and white screen with numbers and a date

Description automatically generated

**4. Create a SQL statement to show all departments and their department’s managers.**

**First code**

SELECT COALESCE(de.dept\_no, dm.dept\_no) AS dept\_no, d.dept\_name, dm.emp\_no, e.first\_name, e.last\_name

FROM departments d

LEFT JOIN dept\_manager dm

ON d.dept\_no = dm.dept\_no

LEFT JOIN dept\_emp de

ON d.dept\_no = de.dept\_no

LEFT JOIN employees e

ON de.emp\_no = e.emp\_no;

**second code**

SELECT COALESCE(de.dept\_no, dm.dept\_no) AS dept\_no, d.dept\_name, dm.emp\_no, e.first\_name, e.last\_name

FROM dept\_manager dm

INNER JOIN departments d

ON dm.dept\_no = d.dept\_no

INNER JOIN dept\_emp de

ON d.dept\_no = de.dept\_no

INNER JOIN employees e

ON de.emp\_no = e.emp\_no;

**Third code**

SELECT d.dept\_no, d.dept\_name, dm.emp\_no, CONCAT(e.first\_name, ‘ ‘, e.last\_name) AS full\_name, dm. from\_date, dm.to\_date

FROM departments d

LEFT JOIN dept\_manager dm

ON dm.dept\_no = d.dept\_no

LEFT JOIN employees e

ON dm.emp\_no = e.emp\_no

ORDER BY d.dept\_no;

**Outcome**

**A screenshot of a computer screen

Description automatically generated**

**5. Create a SQL statement to show a list of HR’s employees who were hired after 1986**

SELECT e.emp\_no, e.first\_name, e.last\_name, e.hire\_date, COALESCE(de.dept\_no, dm.dept\_no) AS dept\_no, d.dept\_name

FROM employees e

LEFT JOIN dept\_emp de

ON e.emp\_no = de.emp\_no

LEFT JOIN dept\_manager dm

ON e.emp\_no = dm.emp\_no

LEFT JOIN departments d

ON d.dept\_no = COALESCE(de.dept\_no, dm.dept\_no)

WHERE d.dept\_name = ‘Human Resources’

OR

COALESCE (de.dept\_no, dm.dept\_no) = ‘d003’

AND e.hire\_date > ‘1986-01-01’;

**Outcome**

**A black screen with white text

Description automatically generated**

**6. Create a SQL statement to increase any employee’s salary up to 2%. Assume the employee has just phoned in with his/her last name.**

UPDATE salaries

SET salary = salary \* 1.02

WHERE emp\_no IN (

SELECT e.emp\_no

FROM employees e

WHERE e.last\_name = 'last\_name'

);

**Here you would swap the last name for the person you are increasing the salary for. This query will then update the salary by 2%**

**Original Data**

A screenshot of a computer screen

Description automatically generated

**Outcome (lets use Bezalel Simmel)**

UPDATE salaries

SET salary = salary \* 1.02

WHERE emp\_no IN (

SELECT e.emp\_no

FROM employees e

WHERE e.last\_name = 'Simmel'

);



**This query was tested on a duplicate database, called ‘employee\_test’ to ensure the original data is not altered.**

**7. Create a SQL statement to delete employee’s record who belongs to marketing department and name start with A**

DELETE FROM employees

WHERE emp\_no IN (SELECT e.emp\_no

FROM employees e

JOIN dept\_emp de ON e.emp\_no = de.emp\_no

JOIN departments d ON de.dept\_no = d.dept\_no

LEFT JOIN dept\_manager dm ON d.dept\_no = dm.dept\_no

WHERE (d.dept\_no = 'd001' OR d.dept\_name = 'Marketing')

AND e.first\_name LIKE ‘A%’);

**To check if delete has worked. You would use this query**

SELECT e.emp\_no, e.first\_name, e.last\_name, COALESCE(de.dept\_no, dm.dept\_no) AS dept\_no, d.dept\_name, COALESCE(de.from\_date, dm.from\_date) AS dept\_from\_date, COALESCE(de.to\_date, dm.to\_date) AS dept\_to\_date, s.salary, s.from\_date AS salary\_from\_date, s.to\_date AS salary\_to\_date

FROM employees e

LEFT JOIN dept\_emp de ON e.emp\_no = de.emp\_no

LEFT JOIN dept\_manager dm ON e.emp\_no = dm.emp\_no

LEFT JOIN departments d ON d.dept\_no = COALESCE(de.dept\_no, dm.dept\_no)

LEFT JOIN salaries s ON e.emp\_no = s.emp\_no

ORDER BY e.emp\_no, COALESCE(de.from\_date, dm.from\_date, s.from\_date), COALESCE(de.to\_date, dm.to\_date, s.to\_date);

**Outcome of this query**

A black screen with white text

Description automatically generated

**8. Create a database view to list the full names of all departments’ managers, and their salaries.**

CREATE VIEW department\_managers\_with\_salaries AS

SELECT e.emp\_no AS EmployeeNumber, CONCAT(e.first\_name, ' ', e.last\_name) AS FullName, d.dept\_name AS DepartmentName, s.salary AS CurrentSalary

FROM dept\_manager dm

INNER JOIN employees e ON e.emp\_no = dm.emp\_no

LEFT JOIN departments d ON dm.dept\_no = d.dept\_no

LEFT JOIN salaries s ON e.emp\_no = s.emp\_no;

**Outcome of managers who have salaries**

A screenshot of a computer screen

Description automatically generated

**9. Create a database view to list all departments and their department’s managers, who were hired between 1980 and 1990**

CREATE VIEW department\_managers\_hired AS

Select d.dept\_no, d.dept\_name, dm.emp\_no, dm.from\_date, dm.to\_date, CONCAT(e.first\_name, ' ', e.last\_name) AS FullName, e. hire\_date

FROM departments d

LEFT JOIN dept\_manager dm ON d.dept\_no = dm.dept\_no

LEFT JOIN employees e ON dm.emp\_no = e.emp\_no

WHERE e.hire\_date BETWEEN ‘1980-01-01’ AND ‘1990-01-01’;

A black screen with white text

Description automatically generated

**10. Create a SQL statement to increase salaries of all department’s managers up to 10% who are working since 1990.**

**Step 1**

**CREATE TEMPORARY TABLE manager\_inc AS**

SELECT d.dept\_no, d.dept\_name, dm.emp\_no, e.first\_name, e.last\_name, e.hire\_date, dm. from\_date, dm.to\_date, s.salary

FROM dept\_manager dm

LEFT JOIN departments d

ON dm.dept\_no = d.dept\_no

LEFT JOIN employees e

ON dm.emp\_no = e.emp\_no

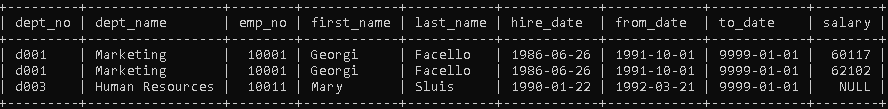
LEFT JOIN salaries s

ON e.emp\_no = s.emp\_no

WHERE YEAR(e.hire\_date) <= ‘1990’

AND YEAR(dm.to\_date) = ‘9999’;

**Outcome**



**Query to increase salary**

UPDATE manager\_inc

SET salary = salary \* 1.10

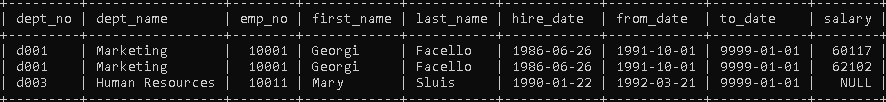
WHERE emp\_no IN (SELECT e.emp\_no

FROM employees e

WHERE YEAR(e.hire\_date) <= '1990’

AND YEAR(dm.to\_date) = ‘9999’

**Original data**



**Updated**

**Table of employee details, salary, department name, department number**

SELECT s.emp\_no, e.first\_name, e.last\_name, s.salary, d.dept\_name, COALESCE(de.dept\_no, dm.dept\_no) AS dept\_no

FROM employees e

LEFT JOIN salaries s

ON e.emp\_no = s.emp\_no

LEFT JOIN dept\_emp de

ON e.emp\_no = de.emp\_no

LEFT JOIN departments d

ON de.dept\_no = d.dept\_no

LEFT JOIN dept\_manager dm

ON dm.dept\_no = d.dept\_no;

**Duplicate database to test queries**

CREATE DATABASE IF NOT EXISTS employees test

USE employees test; SELECT 'CREATING DATABASE STRUCTURE' as 'INFO'; DROP TABLE IF EXISTS dept\_emp, dept\_manager, titles,

salaries, employees, departments; CREATE TABLE employees ( emp\_no INT NOT NULL, birth\_date DATE NOT NULL, first\_name VARCHAR(14) NOT NULL, last\_name VARCHAR(16) NOT NULL, gender ENUM ('M','F') NOT NULL, hire\_date DATE NOT NULL, PRIMARY KEY (emp\_no) ); CREATE TABLE departments ( dept\_no CHAR(4) NOT NULL, dept\_name VARCHAR(40) NOT NULL, PRIMARY KEY (dept\_no), UNIQUE KEY (dept\_name) ); CREATE TABLE dept\_manager ( emp\_no INT NOT NULL, dept\_no CHAR(4) NOT NULL, from\_date DATE NOT NULL,

to\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, FOREIGN KEY (dept\_no) REFERENCES departments (dept\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no,dept\_no) ); CREATE TABLEdept\_emp ( emp\_no INT NOT NULL, dept\_no CHAR(4) NOT NULL, from\_date DATE NOT NULL, to\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, FOREIGN KEY (dept\_no) REFERENCES departments (dept\_no) ON DELETE CASCADE,PRIMARY KEY (emp\_no,dept\_no) ); CREATE TABLE titles ( emp\_no INT NOT NULL, title VARCHAR(50) NOT NULL, from\_date DATE NOT NULL, to\_date DATE, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no,title, from\_date) ) ; CREATE TABLE salaries ( emp\_no INT NOT NULL, salary INT NOT NULL, from\_date DATE NOT NULL, to\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no, from\_date) ) ;

CREATE DATABASE IF NOT EXISTS employees test

USE employees test; SELECT 'CREATING DATABASE STRUCTURE' as 'INFO'; DROP TABLE IF EXISTS dept\_emp, dept\_manager, titles, salaries, employees, departments; CREATE TABLE employees ( emp\_no INT NOT NULL, birth\_date DATE NOT NULL, first\_name VARCHAR(14) NOT NULL, last\_name VARCHAR(16) NOT NULL, gender ENUM ('M','F') NOT NULL, hire\_date DATE NOT NULL, PRIMARY KEY (emp\_no) ); CREATE TABLE departments ( dept\_no CHAR(4) NOT NULL, dept\_name VARCHAR(40) NOT NULL, PRIMARY KEY (dept\_no), UNIQUE KEY (dept\_name) ); CREATE TABLE dept\_manager ( emp\_no INT NOT NULL, dept\_no CHAR(4) NOT NULL, from\_date DATE NOT NULL, USto\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, FOREIGN KEY (dept\_no) REFERENCES departments (dept\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no,dept\_no) ); CREATE TABLEdept\_emp ( emp\_no INT NOT NULL, dept\_no CHAR(4) NOT NULL, from\_date DATE NOT NULL, to\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, FOREIGN KEY (dept\_no) REFERENCES departments (dept\_no) ON DELETE CASCADE,PRIMARY KEY (emp\_no,dept\_no) ); CREATE TABLE titles ( emp\_no INT NOT NULL, title VARCHAR(50) NOT NULL, from\_date DATE NOT NULL, to\_date DATE, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no,title, from\_date) ) ; CREATE TABLE salaries ( emp\_no INT NOT NULL, salary INT NOT NULL, from\_date DATE NOT NULL, to\_date DATE NOT NULL, FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE, PRIMARY KEY (emp\_no, from\_date) ) ;

fix

Question 2

6 – temporay table, stored procedure

7 – temporary table

8 – georgy had no salary in marketing

10 – redo.