COMP2013J Databases and Information Systems

Week 8 In-Class Exam Preparation

DIFFERENCES FROM IN-CLASS TEST

Week 8

Today, there is a database (called “lunches.db”) available on Moodle. You can download this and load it into MySQL to help solve today’s questions.

Today, you can also discuss questions and solutions with classmates, the TAs and the lecturer.

Week 9

The test in week 9 is a **written test** so you will not use a computer.

It is an individual test so you will not be allowed to discuss with classmates.

You are asked to operate on a database called “Lunches”, which records employees of a company, their managers and the lunches they eat (which have a supplier and a type of food associated with them).

The tables are created with the following statements:

```
CREATE TABLE departments
  (
     dept_code CHAR(3) PRIMARY KEY,
     dept_name VARCHAR(20) NOT NULL
  );

CREATE TABLE employees
  (
     id           CHAR(3) PRIMARY KEY,
     first_name   VARCHAR(20) NOT NULL,
     last_name    VARCHAR(20) NOT NULL,
     dept_code    CHAR(3) NOT NULL,
     hire_date    DATE,
     credit_limit DOUBLE(6, 2),
     phone_ext    CHAR(4) DEFAULT '0000',
     manager_id   CHAR(3),
     FOREIGN KEY (dept_code) REFERENCES departments(dept_code)
        ON UPDATE CASCADE ON DELETE RESTRICT,
     FOREIGN KEY (manager_id) REFERENCES employees(id)
        ON DELETE SET NULL ON UPDATE CASCADE
  );

CREATE TABLE suppliers
  (
     supplier_id      VARCHAR(3) PRIMARY KEY,
     supplier_name    VARCHAR(20) NOT NULL,
     supplier_address VARCHAR(30)
  );

CREATE TABLE foods
  (
     food_id     VARCHAR(3) PRIMARY KEY,
     description VARCHAR(20),
     price       DOUBLE(4, 2) NOT NULL,
     supplier_id VARCHAR(3),
     FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id)
        ON UPDATE CASCADE ON DELETE RESTRICT
  );

CREATE TABLE lunches
  (
     lunch_date DATE,
     id         VARCHAR(3),
     food_id    VARCHAR(3),
     FOREIGN KEY (id) REFERENCES employees(id),
     FOREIGN KEY (food_id) REFERENCES foods(food_id)
  );
```

Running a SELECT query for all the data in each table shows the following:

```
mysql> SELECT * FROM departments; SELECT * FROM employees; SELECT * FROM foods; SELECT * FROM
lunches; SELECT * FROM suppliers;
+-----------+------------+
| dept_code | dept_name  |
+-----------+------------+
| ACT       | Accounting |
| EXE       | Executive  |
| MKT       | Marketing  |
| PER       | Personnel  |
| SAL       | Sales      |
| SHP       | Shipping   |
+-----------+------------+
6 rows in set (0.00 sec)

+-----+------------+-----------+-----------+------------+--------------+-----------+----------+
| id  | first_name | last_name | dept_code | hire_date  | credit_limit | phone_ext | manager_id
|
+-----+------------+-----------+-----------+------------+--------------+-----------+----------+
| 201 | Susan      | Brown     | EXE       | 2012-06-01 |        30.00 | 3484      | NULL       |
| 202 | Jim        | Kern      | SAL       | 2103-08-16 |        25.00 | 8722      | 201        |
| 203 | Martha     | Woods     | SHP       | 2018-02-02 |        25.00 | 7591      | 201        |
| 204 | Ellen      | Owens     | SAL       | 2017-07-01 |        15.00 | 8683      | 202        |
| 205 | Henry      | Perkins   | SAL       | 2014-03-01 |        25.00 | 7528      | 202        |
| 206 | Carol      | Rose      | ACT       | NULL       |         0.00 | 0000      | NULL       |
| 207 | Dan        | Smith     | SHP       | 2018-12-01 |        25.00 | 2259      | 203        |
| 208 | Fred       | Campbell  | SHP       | 2017-04-01 |        25.00 | 1752      | 203        |
| 209 | Paula      | Jacobs    | MKT       | 2013-03-17 |        15.00 | 3357      | 201        |
| 210 | Nancy      | Hoffman   | SAL       | 2018-02-16 |        25.00 | 2974      | 203        |
+-----+------------+-----------+-----------+------------+--------------+-----------+----------+
10 rows in set (0.00 sec)

+---------+-------------------+-------+-------------+
| food_id | description       | price | supplier_id |
+---------+-------------------+-------+-------------+
| 301     | Steak Dinner      | 35.00 | 101         |
| 302     | Fish Supper       | 25.00 | 104         |
| 303     | Vegetarian Dinner | 20.00 | 103         |
| 304     | Cajun Treats      | 33.00 | 105         |
| 305     | Wine and Cheese   | 55.00 | 103         |
+---------+-------------------+-------+-------------+
5 rows in set (0.00 sec)

+------------+------+---------+
| lunch_date | id   | food_id |
+------------+------+---------+
| 2018-03-04 | 202  | 301     |
| 2017-02-28 | 206  | 302     |
| 2017-10-07 | 207  | 301     |
| 2018-08-22 | 204  | 303     |
| NULL       | 208  | 302     |
| 2019-01-15 | 209  | 301     |
| NULL       | 204  | 304     |
+------------+------+---------+
7 rows in set (0.00 sec)
```

```
+-------------+---------------------+------------------+
| supplier_id | supplier_name       | supplier_address |
+-------------+---------------------+------------------+
| 101         | Campbell Catering   | Dublin           |
| 102         | Tesco Supermarket   | Kildare          |
| 103         | Nolan Foods         | Waterford        |
| 104         | O Sullivan Supplies | Wexford          |
| 105         | Cajun Treats        | Dublin           |
+-------------+---------------------+------------------+
5 rows in set (0.00 sec)
```

QUESTIONS

For each question below, write an SQL query that can be used to perform the task required.

1. List the food ID and the description of all foods that were supplied by supplier 103.
2. Add your own personal details to the employees table. Put yourself in the shipping department with employee ID of 211, manager_id of 204, a hire date of today's date and whatever you like for the other fields.
3. List all the distinct supplier_address values in the suppliers table and sort these in descending alphabetical order.
4. For each manager, list the description of the food they had for lunch each day. Sort this list by the manager's name, in ascending alphabetical order (a manager is an employee who is the manager of any other employee).
5. List those foods that are not supplied by supplier 103 and do not cost more than €30.
6. List the first name and last names of all employees who work in either the Personnel or Shipping departments and who were hired in 2017.
7. Find the price of the least expensive and the most expensive foods in the foods table.
8. List the names of any suppliers who do not supply any foods.
9. For every employee in the employees table, list their full name, the name of their manager and the name of the department they work in.
10. Change Jim Kern's record in the employees table so that his credit limit is 95.00 and his phone extension is 0114.

What is the effect of the following queries (for SELECT queries, you can show the data that would be selected)? For each question, assume you are operating on the original database.

11. DELETE FROM employees WHERE id='201';
12. SELECT employees.first_name, employees.last_name, managers.first_name, managers.last_name FROM employees LEFT JOIN employees AS managers ON employees.manager_id=managers.id WHERE employees.dept_code='ACT';
13. SELECT employees.first_name, employees.last_name, managers.first_name, managers.last_name FROM employees JOIN employees AS managers ON employees.manager_id=managers.id WHERE employees.dept_code='ACT';
14. SELECT dept_code, MAX(credit_limit) FROM employees GROUP BY dept_code;
15. UPDATE foods SET price= price * 1.1 WHERE supplier_id='103';