Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 70%. We keep your highest score.

Next item →

1. Which of the following queries will return the data for employees who belong to the department with the highest value of department ID. **1 / 1 point**

- (•) SELECT * FROM EMPLOYEES WHERE DEP_ID =
  (SELECT MAX(DEPT_ID_DEP) FROM DEPARTMENTS)
- ( ) SELECT * FROM EMPLOYEES WHERE DEP_ID =
  (SELECT DEPT_ID_DEP FROM DEPARTMENTS WHERE DEPT_ID_DEP IS MAX)
- ( ) SELECT * FROM EMPLOYEES WHERE DEPT_ID_DEP =
  MAX (SELECT DEPT_ID_DEP FROM DEPARTMENTS)
- ( ) SELECT * FROM EMPLOYEES WHERE DEP_ID = MAX(DEP_ID)

Correct
Correct. This uses subqueries and functions.

2. A DEPARTMENTS table contains DEP_NAME, and DEPT_ID_DEP columns and an EMPLOYEES table contains columns called F_NAME and DEP_ID. We want to retrieve the Department Name for each Employee. Which of the following queries will correctly accomplish this? **1 / 1 point**

- ( ) SELECT E.F_NAME, D.DEP_NAME FROM EMPLOYEES, DEPARTMENTS
- (•) SELECT F_NAME, DEP_NAME FROM EMPLOYEES, DEPARTMENTS WHERE DEPT_ID_DEP = DEP_ID
- ( ) SELECT D.F_NAME, E.DEP_NAME FROM EMPLOYEES E, DEPARTMENTS D WHERE D.DEPT_ID_DEP = E.DEP_ID
- ( ) SELECT F_NAME, DEP_NAME FROM EMPLOYEES E, DEPARTMENTS D WHERE E.DEPT_ID_DEP = D.DEP_ID

Correct
Correct! This is a correct way to use multiple tables using an implicit join.

3. You are writing a query that will give you the total cost to the Pet Rescue organization of rescuing animals. The cost of each rescue is stored in the Cost column. You want the result column to be called “Total_Cost”. Which of the following SQL queries is correct? **1 / 1 point**

- ( ) SELECT SUM(Cost) FROM PetRescue
- (•) SELECT SUM(Cost) AS Total_Cost FROM PetRescue
- ( ) SELECT SUM(Total_Cost) From PetRescue
- ( ) SELECT Total_Cost FROM PetRescue

Correct
Correct. The SUM(Cost) function will give the total cost, and the AS Total_Cost clause will give the result column an alias of Total_Cost.

4. Which of the following queries correctly calculates the total number of days an employee has lived, using their date of birth ('DOB') and the current date, in MySQL? Assume the 'DOB' column exists in the 'Employees' table. **1 / 1 point**

- ( ) SELECT (CURRENT_DATE – DOB) FROM Employees
- (•) SELECT DATEDIFF(CURRENT_DATE, DOB) FROM Employees
- ( ) SELECT FROM_DAYS(DATEDIFF(CURRENT_DATE, DOB)) FROM Employees
- ( ) SELECT FROM_DAYS(DATE_SUB(CURRENT_DATE, DOB) FROM Employees

Correct
Correct. DATEDIFF calculates the total number of days between two dates.

5. You have a record of a set of medicines called ‘MEDS’. Their date of expiry is exactly 1 year after their date of manufacturing. The name of the medicines is available as ‘NAME’ and their date of manufacturing is available as a column ‘DOM’. Which of the commands will generate an output that contains name of the medicines and also displays their date of expiry as a column ‘DOE’? Assume use of MySQL. **1 / 1 point**

- (•) SELECT NAME, DATE_ADD(DOM, INTERVAL 1 YEAR) AS DOE FROM MEDS
- ( ) SELECT NAME, DATE_ADD(DOM, INTERVAL 1 YEARS) AS DOE FROM MEDS
- ( ) SELECT NAME, DATEADD(DOM, INTERVAL 1 YEAR) FROM MEDS
- ( ) SELECT NAME, DATEADD(DOM, INTERVAL 1 YEAR) AS DOE FROM MEDS

Correct
Correct. Use DATE_ADD for adding 1 year and represent at DOE.