**9.** How the resulting salaries if every employee working on the ‘Research’ Departments is given a 10 percent raise.

= UPDATE EMPLOYEE

SET Salary = Salary \* 1.10

WHERE DNo = 1;

|  |
| --- |
| 9. Retrieve the name of employees who born in the year 1990  = SELECT Name  FROM EMPLOYEE  WHERE YEAR(Birthdate) = 1990;  **10**. How the resulting salaries if every employee working on the ‘Research’  Departments is given a 10 percent raise.  = UPDATE EMPLOYEE  SET Salary = Salary \* 1.10  WHERE DNo = 1;  10. Retrieve the name of employees and their dept name.  = SELECT EMPLOYEE.Name AS EmployeeName, DEPARTMENT.DName AS DepartmentName  FROM EMPLOYEE  JOIN DEPARTMENT ON EMPLOYEE.DNo = DEPARTMENT.DNo;  **11**. Find the names of all branches in the loan relation.  = SELECT DISTINCT branch\_name FROM loan;  11. Find all customers having a loan, an account or both at the bank  = SELECT DISTINCT customer\_name FROM borrower  UNION  SELECT DISTINCT customer\_name FROM depositor;  11. Delete all account tuples in the Perryridge branch  = DELETE FROM account WHERE branch\_name = 'Perryridge';  11. Find the average account balance at each branch whose average balance is greater  than 1200  = SELECT branch\_name, AVG(balance) AS avg\_balance  FROM account  GROUP BY branch\_name  HAVING AVG(balance) > 1200;  11. Find average account balance at each branch.  = SELECT branch\_name, AVG(balance) AS avg\_balance  FROM account  GROUP BY branch\_name;  **12.** Find all loan numbers for loans made at the Perryridge branch with loan amount  greater than $1200.  = SELECT loan\_number  FROM loan  WHERE branch\_name = 'Perryridge' AND amount > 1200;  12. v) Find all customers of the bank who have a loan but not an account:  = SELECT c.customer\_name  FROM customer c  JOIN borrower b ON c.customer\_name = b.customer\_name  LEFT JOIN account a ON c.customer\_name = a.customer\_name  WHERE a.customer\_name IS NULL;  12. vi) Find the average account balance at each branch whose average balance is greater than 1200:  = SELECT branch\_name, AVG(balance) AS avg\_balance  FROM account  GROUP BY branch\_name  HAVING AVG(balance) > 1200;  12. vii) Find the number of depositors for each branch:  = SELECT branch\_name, COUNT(DISTINCT customer\_name) AS num\_depositors  FROM depositor  GROUP BY branch\_name;  12. viii) Delete all account tuples at every branch located in Needham:  = DELETE FROM account  WHERE branch\_name IN (SELECT branch\_name FROM branch WHERE branch\_city = 'Needham');  **13.** iv) Find all customers who have both a loan and account at the bank:  = SELECT DISTINCT c.customer\_name  FROM customer c  JOIN borrower b ON c.customer\_name = b.customer\_name  JOIN account a ON c.customer\_name = a.customer\_name;  13. v) For all customers who have a loan from the bank, find their names, loan numbers, and loan amount:  = SELECT c.customer\_name, b.loan\_number, l.amount  FROM customer c  JOIN borrower b ON c.customer\_name = b.customer\_name  JOIN loan l ON b.loan\_number = l.loan\_number;  13. vi) Delete all loans with loan amounts between $1300 and $1500:  = DELETE FROM loan  WHERE amount BETWEEN 1300 AND 1500;  13. vii) Find the average account balance at each branch whose average balance is greater than 1200:  = SELECT branch\_name, AVG(balance) AS avg\_balance  FROM account  GROUP BY branch\_name  HAVING AVG(balance) > 1200;  13. viii) Find the names of all customers whose street address includes the substring 'main':  = SELECT customer\_name  FROM customer  WHERE customer\_street LIKE '%main%';  **15**. iv) Find all loan numbers for loans made at the Perryridge branch with loan amount greater than $1200:  =SELECT loan\_number  FROM loan  WHERE branch\_name = 'Perryridge' AND amount > 1200;  15. vi) Find the average account balance at the Needham branch:  = SELECT AVG(balance) AS avg\_balance  FROM account  WHERE branch\_name = 'Needham';  **16**.iv) Find the average account balance at the Dhaka branch:  = SELECT AVG(balance) AS avg\_balance  FROM account  WHERE branch\_name = 'Dhaka';  16. v) Find all customers of the bank who have an account but not a loan:  = SELECT DISTINCT c.customer\_name  FROM customer c  JOIN depositor d ON c.customer\_name = d.customer\_name  LEFT JOIN borrower b ON c.customer\_name = b.customer\_name  WHERE b.customer\_name IS NULL;  16. vi) Find average account balance at each branch:  = SELECT branch\_name, AVG(balance) AS avg\_balance  FROM account  GROUP BY branch\_name;  **17**. iv) Modify the database so that Jones now lives in Newtown:  = UPDATE employee  SET city = 'Newtown'  WHERE person\_name = 'Jones';  17. v) Give all employees of First Bank Corporation a 10 percent salary raise:  = UPDATE works  SET salary = salary \* 1.10  WHERE company\_name = 'First Bank Corporation';  17. vi) Give all managers in this database a 10 percent salary raise:  = UPDATE works  SET salary = salary \* 1.10  WHERE person\_name IN (SELECT person\_name FROM manages);  **17.** vii) Give all managers in this database a 10 percent salary raise, unless the salary would be greater than $100,000. In such cases, give only a 3 percent raise:  = UPDATE works  SET salary = CASE      WHEN salary \* 1.10 <= 100000 THEN salary \* 1.10      ELSE salary \* 1.03  END  WHERE person\_name IN (SELECT person\_name FROM manages);  17. viii) Delete all tuples in the works relation for employees of Small Bank Corporation:  = DELETE FROM works  WHERE company\_name = 'Small Bank Corporation';  **18.**iv) Modify the database so that Johnson now lives in New York:  = UPDATE employee  SET city = 'New York'  WHERE person\_name = 'Johnson';  18. v. Find the company with the most employees:  = SELECT company\_name, COUNT(person\_name) AS num\_employees  FROM works  GROUP BY company\_name  ORDER BY num\_employees DESC  LIMIT 1;  18. vi. Find the company with the smallest payroll:  = SELECT company\_name, SUM(salary) AS total\_payroll  FROM works  GROUP BY company\_name  ORDER BY total\_payroll ASC  LIMIT 1;  18. vii. Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation:  = SELECT company\_name  FROM works  GROUP BY company\_name  HAVING AVG(salary) > (SELECT AVG(salary) FROM works WHERE company\_name = 'First Bank Corporation'); |