

1. Create a stored procedure that adds the missing data to the «Bank scheme» database.
2. Create a stored procedure that changes the *datetime* data type to *date* for all the corresponding columns of the «Bank scheme».
3. Create a stored procedure that counts the number of accounts for each bank customer and returns either 'None', '1', '2' or '3+'. The result set should include the customer identification number, the customer type and the number of accounts.
4. Create a stored procedure that uses two CASE expressions to generate two output columns, one to show whether the customer has any checking accounts and the other to show whether the customer has any savings accounts. If the customer has the account, print 'Y', otherwise print 'N'. The result set should include the following information: the customer ID, their home address, the existence of checking accounts and the existence of savings accounts.
5. Create a stored procedure that declares a variable and set it to the count of all PRODUCT\_TYPE\_CD in the Product\_Type table. If the count is greater than or equal to 3, the stored procedure should display a message that says, "The number of PRODUCT\_TYPE\_CD is greater than or equal to 3". Otherwise, it should say, "The number of PRODUCT\_TYPE\_CD is less than 3".
6. Create a stored procedure that uses two variables to store:
  - a) the count of all of the customers in the Customer table;
  - b) the average avail balance for each customer.

If the customers count is greater than or equal to 13, the stored procedure should display a result set that displays the values of both variables. Otherwise, the procedure should display a result set that displays a message that says, "The number of customers is less than 13".

7. Create a stored procedure that calculates the common factors between 15 and 30. This procedure should display a string that displays the common factors in this form:  
Common factors of 15 and 30: 1 3 5 15
8. Create a stored procedure that shows all numeric characters from the entire string. You can use the ADDRESS columns in the «Bank scheme» database or any row of your choice.
9. Create a stored procedure for the «Bank scheme» database of your choice. Condition: the procedure must be encrypted.
10. Create two stored procedures for the «Bank scheme» database. Condition: one procedure must call another.