**C#**

Proszę skorzystać z rozwiązania zawartego w tym repozytorium.

Zadania znajdują się w pliku głównym rozwiązania: **Program.cs**

Odpowiedzi proszę umieszczać w formie komentarza bezpośrednio w pliku programu.

**SQL**

Proszę korzystać z <https://www.w3schools.com/sql/trysql.asp?filename=trysql_op_in>

* 1. Czy są towary w kategorii Seafood droższe od towaru „Northwoods Cranberry Sauce”
  2. Ile w bazie jest zamówień od klientów z Niemiec?
  3. Klient prosi o podniesienie cen wszystkich towarów w bazie SQL wg zasady:

         towary z kategorii „Condiments” +7%

         pozostałe towary                    +5%

**ODPOWIEDZI** (wypisz używane kwerendy sql)

1. ODP:

SQL:

1. ODP:

SQL:

1. ODP:

SLQ:

**(C/AL)**

Wyjaśnić działanie funkcji (C/AL) – co jest jej głównym zadaniem?:

DB\_DRIVER := '{Firebird/InterBase(r) driver}';

DB\_UID := 'SYSDBA';

DB\_PWD := 'masterkey';

DB\_NAME := '10.1.0.4/3050:E:\tfs\base\DB\_SMM.FDB';

DB\_CHARSET := 'WIN1250';

conn := conn.OdbcConnection('Driver='+DB\_DRIVER+';UID='+DB\_UID+';PWD='+DB\_PWD+';DBNAME='+DB\_NAME+';Charset='+DB\_CHARSET+';DIALECT=3;');

conn.ConnectionTimeout := 10;

conn.Open;

WHILE conn\_reader.Read DO BEGIN

Actions.SETFILTER(No,FORMAT(conn\_reader.GetString(conn\_reader.GetOrdinal('No'))));

IF (NOT Actions.FINDFIRST) THEN BEGIN

Actions.No := conn\_reader.GetString(conn\_reader.GetOrdinal('No'));

Cases.RESET;

Cases.SETFILTER(No,Actions.No);

IF Cases.FINDFIRST THEN Actions.SettlementMethodCode := Cases.SettlementMethodCode;

Actions.Code := NoSeriesMgt.GetNextNo('ACTION-NO',TODAY,FALSE);

Actions.Typ\_Cases := conn\_reader.GetString(conn\_reader.GetOrdinal('Case\_Code'));

Actions.No := conn\_reader.GetString(conn\_reader.GetOrdinal('No'));

Actions.Name := conn\_reader.GetString(conn\_reader.GetOrdinal('Name'));

Actions.Customer\_Name := conn\_reader.GetString(conn\_reader.GetOrdinal('Customer\_Name'));

Actions.Description := COPYSTR(conn\_reader.GetString(conn\_reader.GetOrdinal('Description')),1,245);

IF (STRLEN(Actions.Description) = 245) THEN Actions.Description += '[...]';

Actions.Group := conn\_reader.GetString(conn\_reader.GetOrdinal('Group'));

CASE conn\_reader.GetString(conn\_reader.GetOrdinal('Status')) OF

'0': BEGIN Actions.Status\_TFK := Actions.Status\_TFK::Otwarta; Actions.Status\_NAV := Actions.Status\_NAV::Otwarte; END;

'1': BEGIN Actions.Status\_TFK := Actions.Status\_TFK::Zatwierdzona; Actions.Status\_NAV := Actions.Status\_NAV::Otwarte; END;

'2': BEGIN Actions.Status\_TFK := Actions.Status\_TFK::Zafakturowana; Actions.Status\_NAV := Actions.Status\_NAV::Zafakturowane; END;

'100': BEGIN Actions.Status\_TFK := Actions.Status\_TFK::Usunięta; Actions.Status\_NAV := Actions.Status\_NAV::Niefakturowane; END;

'101': BEGIN Actions.Status\_TFK := Actions.Status\_TFK::Spisana; Actions.Status\_NAV := Actions.Status\_NAV::Niefakturowane; END;

ELSE

Actions.Status\_TFK := Actions.Status\_TFK::Nieznana;

END;

Actions.Date := DT2DATE(conn\_reader.GetDateTime(conn\_reader.GetOrdinal('DATE')));

Actions.Date\_Utworzenia := DT2DATE(conn\_reader.GetDateTime(conn\_reader.GetOrdinal('Date\_i')));

Actions.Worker := conn\_reader.GetString(conn\_reader.GetOrdinal('Worker'));

Actions.Worker\_Code := conn\_reader.GetString(conn\_reader.GetOrdinal('worker\_code'));;

Actions.Description := conn\_reader.GetString(conn\_reader.GetOrdinal('Description2'));

Actions.Name\_Projektu := conn\_reader.GetString(conn\_reader.GetOrdinal('projekt\_Name'));

Actions.Entered\_Time := COPYSTR(CONVERTSTR(conn\_reader.GetString(conn\_reader.GetOrdinal('Entered\_H')),'.',','),1,4);

Actions.Invoiced\_Time := COPYSTR(CONVERTSTR(conn\_reader.GetString(conn\_reader.GetOrdinal('Invoiced\_H')),'.',','),1,4);

IF Actions.INSERT(TRUE) THEN NoSeriesMgt.SaveNoSeries();

END