

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут автоматики, кібернетики та
обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри
комп'ютерних технологій та
економічної кібернетики
д. е. н., проф. П. М. Грицюк

« ____ » _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ «БАКАЛАВР»
«Інформаційна система підтримки діяльності хлібозаводу»

Виконав:

здобувач вищої освіти за ОПП
«Інформаційні системи та технології»
спеціальності 126 «Інформаційні системи та
технології», групи ІСТ-21інт

Федів Віталій Миколайович

Керівник:

ст. виклад. Шевченко І.М.

Рецензент:

к.е.н., доцент Волошин В.С.

Рівне – 2022

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 51 с., 20 рис., 3 табл., 20 літературних джерел.

Актуальність теми даної бакалаврської роботи полягає в тому, що Україна знаходиться у режимі воєнного стану і виникає логічна заборона на використання програмних продуктів країни-агресора. Саме тому розробка та впровадження вітчизняного програмного забезпечення для потреб бухгалтерського обліку є досить актуальною.

Об'єкт дослідження бакалаврської роботи – типове підприємство пекарської галузі України. **Предметною областю** роботи є інформаційна підтримка обліку сировини, закупок та виробництва хлібозаводу.

Метою дослідження бакалаврської роботи є: на основі отриманих теоретичних знань та практичних навичок, щодо розробки інформаційних систем та їхнього застосування до проблемних питань бакалаврської роботи, розробити ІС, яку можна було б впровадити до зазначеної предметної області об'єкта дослідження.

У бакалаврської роботи було: охарактеризовано предметну галузь та об'єкт дослідження, проаналізовано літературні джерела та практичний досвід, спроектовано логічну модель даних БД, розроблено інформаційну систему підтримки діяльності хлібозаводу.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, ПРОГРАМУВАННЯ, ХМАРНІ ТЕХНОЛОГІЇ, ХЛІБОЗАВОД, ВИРОБНИЧІ РЕСУРСИ, БАЗИ ДАНИХ, АВТОМАТИЗАЦІЯ ПРОЦЕСІВ.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	5
1.1. Характеристика предметної галузі та об'єкта дослідження	5
1.2. Аналіз літературних джерел та практичного досвіду	10
1.3. Актуальні інформаційні системи бухгалтерського обліку хлібозаводу	14
РОЗДІЛ 2. ІНФОРМАЦІЙНА СИСТЕМА ХЛІБОЗАВОДУ	24
2.1. Логічна модель даних	24
2.2. Вдосконалення функціональних можливостей бухгалтерської інформаційної системи хлібозаводу	28
2.3. Функціональні можливості ІС хлібозаводу	33
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
ДОДАТКИ	52

ВСТУП

Інформаційні системи та Web-додатки охопили усі сфери життя будь-якої людини у побуті та у взаємовідносинах з роботодавцями.

Актуальність теми даної бакалаврської роботи полягає в тому, що Україна знаходиться у режимі воєнного стану і виникає логічна заборона на використання програмних продуктів країни-агресора. Саме тому розробка та впровадження вітчизняного програмного забезпечення для потреб бухгалтерського обліку є досить актуальною.

Об'єкт дослідження бакалаврської роботи – типове підприємство пекарської галузі України. **Предметною областю** роботи є інформаційна підтримка обліку сировини, закупок та виробництва хлібозаводу.

Метою дослідження бакалаврської роботи є: на основі отриманих теоретичних знань та практичних навичок, щодо розробки інформаційних систем та їхнього застосування до проблемних питань бакалаврської роботи, розробити ІС, яку можна було б впровадити до зазначеної предметної області об'єкта дослідження.

Завданнями бакалаврської роботи є:

1. Охарактеризувати предметну галузь та об'єкт дослідження.
2. Проаналізувати літературні джерела та практичний досвід.
3. Дослідити інформаційні системи бухгалтерського обліку хлібозаводу.
4. Спроекувати логічну модель даних бази даних.
5. Розробити інформаційну систему підтримки діяльності хлібозаводу.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Характеристика предметної галузі та об'єкта дослідження

Харчова промисловість України є однією зі стратегічних галузей розвитку вітчизняної економіки. Вона покликана забезпечувати стабільне постачання населення необхідними якісними продуктами харчування і виступає важливою ланкою формування продовольчої безпеки держави й експортного потенціалу країни.

Першочергове місце у споживчому кошику займає хліб. Хліб був і залишається одним з основних продуктів харчування населення нашої країни. Кількість щоденного споживання хліба населення України дозволяє вважати його одним з найважливіших продуктів харчування.

У сумарному обсязі продукції всієї харчової промисловості України хлібопекарська галузь займає одне із провідних місць, а частка хлібопродуктів у раціоні населення України складає 15 %, що підтверджує їх статус як основного продукту харчування. Хлібобулочні вироби є важливим продуктом харчування для більшості населення України, а для найбільш вразливих верств – основним. Потреба в них притаманна людям будь-якого соціального статусу і за будь-якого рівня доходів [13].

Цій галузі традиційно приділяється менше уваги, ніж іншим, незважаючи на те, що позиції, які відбуваються на ринку хліба і хлібобулочних виробів, миттєво впливають на економічне та соціально-політичне життя країни.

Ринок хліба України на 99,9 % представлений продукцією вітчизняного виробництва. Однак протягом останнього десятиліття в хлібопекарській галузі зафіксовано спад виробництва [3].

Хліб – це харчовий продукт, що випікається з борошна, ціла група основних продуктів, які виробляються шляхом випічки або смаженням.

Хлібобулочні вироби – харчові продукти, які випікаються з борошна, дріжджів, солі, води та додаткової сировини [20].

Хлібопекарська галузь – одна з провідних галузей харчової промисловості України, призначена для забезпечення виробництва хліба, хлібобулочних та інших борошняних виробів у обсягах, які відповідають нормам державної продовольчої безпеки.

Концентрація виробництва – процес зосередження засобів виробництва, землі й трудових ресурсів на одному й тому самому підприємстві з метою збільшення обсягів продукції.

Хлібопекарська промисловість випускає великий асортимент хлібобулочних виробів. Крім того, виробляють національні сорти хліба: лаваш, маца, коровай та інші.

Важливими потребами подальшого розвитку ринку хлібобулочних виробів є суттєве поліпшення забезпечення потреб споживачів у якісному хлібі промислової випічки за прийнятною ціною, одержання на цій основі достатньої маси прибутку підприємствами-товаровиробниками і підвищення ефективності їх діяльності. Хоча існуючий асортимент хліба досить різноманітний, постійно ведеться велика робота з розширення та поліпшення асортименту хлібобулочних виробів, створюються нові дієтичні і лікувальні сорти хліба [11].

Поліпшення якості продукції, розвиток асортименту виробів, у тому числі дієтичного призначення, є актуальною проблемою для хлібопекарської промисловості.

Дуже важливими проблемами в сучасній харчовій промисловості України по виробництву хліба на сьогодні є: подальше вдосконалення технології з метою інтенсифікації хліба; регулювання його харчової цінності; виробництво нових дієтичних гатунків хліба та хлібобулочних виробів; широке використання упаковки для більш довгого зберігання свіжості хліба. На сьогоднішній день практично всі хлібзаводи приватизовані. В основному шляхом перетворення у відкриті та закриті акціонерні товариства [5].

Сьогодні більшість хлібзаводів мають потребу в проведенні реконструкції,

заміні технологічного устаткування на більш сучасне, енергозберігаюче. Значна частина великого хлібного бізнесу представлена приватними компаніями, у тому числі різними інтеграційними структурами. Адже загальнодержавні тенденції щодо злиття та поглинання компаній притаманні і ринку хлібопекарства. Тут також відбувається формування вертикально інтегрованих структур із замкнутим циклом виробництва – від виробництва борошна до випуску хлібопродуктів. Адже укрупнення і концентрація виробництва дає змогу оптимізувати логістику, збільшити асортимент, підвищити прибутковість і, в такий спосіб, вижити та залишитися гравцем ринку.

До складу таких міжрегіональних холдингів входять і хлібопекарські заводи, і борошномельні підприємства, і кондитерські виробництва [12, с. 118].

Серед потужних холдингових компаній варто відзначити ПАТ «Хліб Києва» близько 17 % вітчизняного ринку виробництва хлібопродуктів. Серед інших найбільших об'єднань можна назвати ВАТ «Коровай», що об'єднує підприємства Луганської області. Їхня частка у сумарному виробництві хліба і хлібобулочних виробів, за різними оцінками, складає близько 4 %. Якщо ж говорити про окремих виробників хліба, то лідерство серед українських хлібокомбінатів вже не перший рік утримує ВАТ – Одеський коровай – складає близько 3 % хліба і хлібобулочних виробів [11, с. 86].

Наприклад, лідер ринку ПАТ «Київхліб», переважно представлений в Києві і області, утримує близько 90 % столичного ринку, а також близько 14 % національного.

В Україні спостерігається негативна тенденція фінансування суб'єктів ринку. Перш за все, це пов'язано з браком власних коштів і складні умови отримання кредитних ресурсів. Банки вельми охоче кредитують лідерів всієї харчової галузі, і дуже неохоче ведуть діалог з дрібнішими гравцями ринку.

До 1990 року в Україні щороку випікалося близько 7 млн тонн хліба та хлібобулочних виробів. В подальші роки виробництво зазначеної продукції зменшилося більше, ніж у 4 рази [15].

Ряд ключових гравців ринку – це колишні регіональні об'єднання

(«Київхліб», «Донбасхліб», «Дніпрохліб», «Житомирхліб», «Миколаївхліб» та інші), які зберегли свою виробничу структуру. Останнім часом компанія налагоджує зв'язки з іншими країнами, зацікавленими в отриманні високоякісної продукції Київських хлібозаводів – німецькі партнери закупають пряники по всій Європі [12].

Конкурентами компанії є: «Хліб України», «Київський хліб», ПрАТ «Київхлібпродукт», «Хліб Союз», «Фастівське хлібоприймальне підприємство», ПАТ «Устимівське хлібоприймальне підприємство», ПАТ «Миронівське хлібоприймальне підприємство», ПрАТ «Переяслав-Хмельницький хлібзавод», ПАТ «Миронівський хлібпродукт» та інші.

Продукція ПАТ «Київхліб» на ринку хлібобулочних виробів міста Києва займає приблизно 90%, а решта 10% припадає на приватні підприємства та інші підприємства, що не входять до ПАТ «Київхліб». Тобто ПАТ «Київхліб» є монополістом на ринку хлібобулочних виробів міста Києва [3].

Конкурентоспроможність на ринку хлібобулочних виробів залежить від ряду таких чинників як: конкурентоспроможність товарів підприємств на зовнішньому та внутрішньому ринках, вид виробленого товару, місткість ринку, легкість доступу на ринок, однорідність ринку, конкурентні підприємства, що вже працюють на даному ринку, конкурентоспроможності галузі позиції, можливість технічних нововведень у галузі, конкурентоспроможність регіону і країни.

ДП ПАТ «Київхліб» постійно працює над вдосконаленням випуску продукції, поліпшенням її якості та розширення асортименту хлібобулочних виробів. Це стало незмінним правилом роботи підприємства [12].

За даними Державної служби статистики, у перші два місяці 2018 року в Україні було вироблено 239,36 тис. тонн хліба та хлібобулочних виробів, а у 2019 році в Україні було вироблено 220 тис. тонн хліба та хлібобулочних виробів, що на 8,09% менше, ніж у попередньому році [7].

За останні 3 роки спостерігається зменшення виробництва ХБВ (Рис. 1.1). Однією з головних причини зниження обсягів виробництва хліба та

хлібобулочних виробів є зростання цін на сировину та скорочення чисельності населення України. На 1 грудня 2019 року кількість наявного населення, за оцінкою Держстату, становила 37 млн 289 тис. осіб і зменшилася в порівнянні з аналогічною датою попереднього року на 4633,7 тис. осіб [10]. Також зміни в структурі харчування громадян України – переорієнтація споживачів на інші продукти харчування; збільшення виробництва хліба суб'єктами господарювання, що не звітують про обсяги перед статистичними органами – це міні-пекарні та супермаркетами, що виробляють хліб під власними торговими марками [14].

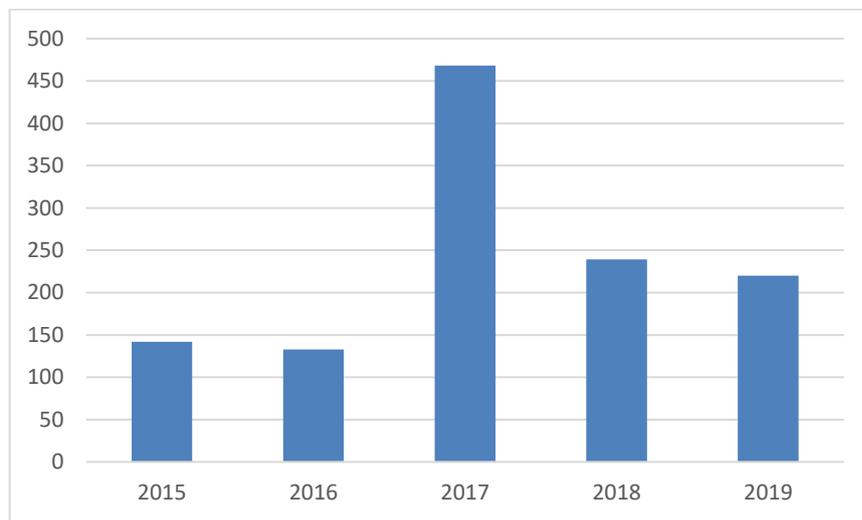


Рис. 1.1. Динаміка виробництва хліба та хлібобулочних виробів в Україні за січень-лютий 2015-2019 рр., тис. тонн

Відповідно до офіційної статистики, динаміка обсягів виробництва хліба та хлібобулочних виробів в Україні має стійку тенденцію до зниження. Останнім часом спостерігаються стрімкі темпи розвитку національного ринку хлібобулочних напівфабрикатів. Практично кожен великий виробник класичного хліба в Україні апробується у сегменті «заморозки».

Ринок характеризується тенденцією подорожчання хліба зі збільшенням цін на сировині складові: борошно, цукор, яйця, оліє жирову продукції, а також зростанням цін на енергоносії та паливно-мастильні матеріали.

Індекс споживчих цін на хліб росте протягом досліджуваного періоду. За

2019 рік ціни на хліб піднялися на 17,7% в порівнянні з 2018 роком, тоді як інфляція в Україні за рік склала 4,1%, тобто подорожчання так званого «хлібного кошика» відбулось більш ніж в 4 рази швидше зростання середнього рівня цін в країні.

За останні 4 роки спостерігається стабільне зменшення виробництва ХБВ (Рис. 1.2). Однією з головних проблем ринку ХБВ є наявність великого тіньового сегмента, який перешкоджає розвитку справедливої конкуренції на ринку [2].

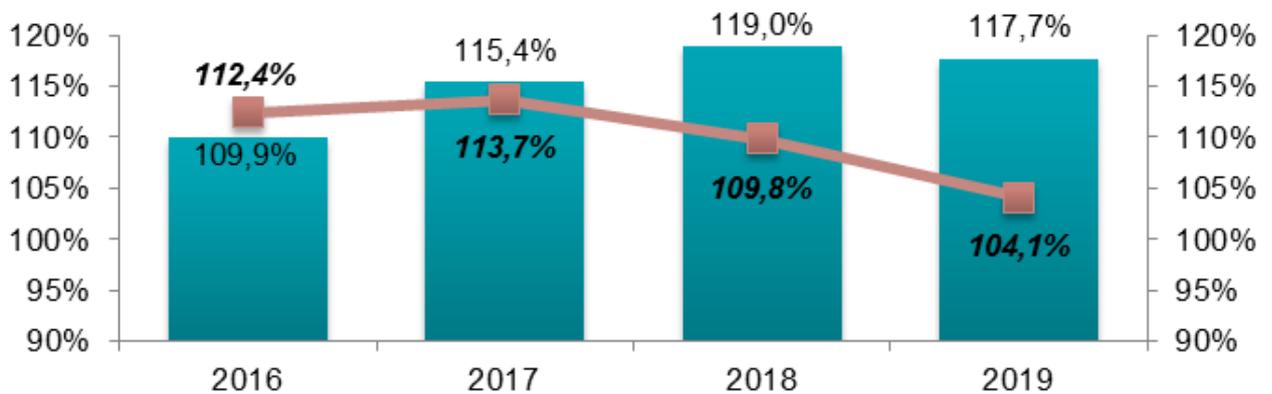


Рис. 1.2. Динаміка індексу споживчих цін на хліб та індексу інфляції в Україні за 2016-2019 рр.

Ринок практично повністю представлений продукцією вітчизняного виробництва, що пов'язано з короткими термінами зберігання і складністю транспортування. Експорт в сегменті зростає, основними споживачами українських ХБВ є США. Динаміка імпорту втричі перевищує темпи росту експорту, 99% усієї імпортової продукції поставляється в Україну з країн Європейського Союзу.

1.2. Аналіз літературних джерел та практичного досвіду

Сьогодні хлібопекарська промисловість України за виробничими потужностями, механізацією технологічних процесів, асортиментом,

спроможна забезпечити населення різними видами хлібобулочних виробів. Харчова цінність хлібобулочних виробів має важливе значення, оскільки вони забезпечують більше половини добової потреби людини в енергії і рослинного білку. Серед закордонних вчених, які досліджували технології виготовлення хліба, можна відмітити: С.М. Rosell, S. Cauvain, M. Seguchi, C. Wrigley, I. Batey та інші.

Питанням удосконалення процесів виробництва хлібопекарської промисловості часто зустрічається у роботах вітчизняних дослідників, зокрема Пугаченко О. Б. досліджував особливості хлібопекарного виробництва та їх вплив на склад і облік запасів та показав, що управління виробничими запасами на хлібопекарних підприємствах полягає у підтримці рівня виробничих запасів у мінімально необхідних розмірах, але таких, що забезпечують нормальну виробничо-комерційну діяльність підприємства.

Саченко С.І. досліджувала методики й організації обліку і внутрішньогосподарського контролю виробничих витрат на хлібопекарних підприємствах та розробила модель обліку витрат на виробництво для забезпечення можливості здійснення цього облікового процесу автоматизованим способом.

Стаднюк Т. проаналізувала операційну діяльність хлібопекарних підприємств Волинської області, дослідивши вплив на рівень рентабельності господарської діяльності суб'єктів господарювання. Лешун М.І. у своєму дослідженні проаналізувала облік, виробництво та збут в системі управління хлібопекарським підприємством та розробила практичні рекомендації з удосконалення процесів виробництва та збуту в системі управління на підприємствах хлібопекарської промисловості. Примуш Ю.С. та Севастьянова А.О. досліджували особливості обліку власного капіталу на прикладі ТДВ «Запорізький хлібозавод № 5» та запропонували різні підходи до визначення сутності власного капіталу.

Існує багато ІС у виробництві та найбільш практичною серед них є система «Галактика Харчова промисловість».

Система «Галактика Харчова промисловість» для автоматизації підприємств харчової промисловості враховує особливості бухгалтерського обліку в галузі, вирішує завдання постачання, збуту, дозволяє формувати виробничу програму і контролювати її виконання. Як правило, в першу чергу більшість підприємств впроваджує засоби автоматизації бухгалтерського і оперативного обліку, що відразу ж підвищує його оперативність і аналітичність [17].

Рішення «Галактика Харчова промисловість» є основою єдиного інформаційного простору підприємства і охоплює всю організаційну структуру підприємства харчової промисловості, надаючи кожному учаснику бізнесу зручний інструмент і ефективні важелі управління:

Генеральний директор отримує повну і актуальну інформацію про роботу підприємства, що дозволяє ефективніше управляти бізнесом.

Фінансовий директор набуває інструменти планування і аналізу стану підприємства, зручні засоби фінансового управління..

Головний інженер ефективно планує завантаження устаткування, автоматично формує плани проведення ремонтів устаткування.

Директор з економіки отримує інструмент планування господарської діяльності, аналізу результатів виконання бюджету підприємства.

Директор з виробництва може планувати виробничу програму, формувати змінні завдання, враховувати витрачається сировину, а також мати розширені можливості обліку якісних характеристик заміни в рецептурах.

Директор з якості отримує кошти ведення наскрізного партійного обліку.

Головний бухгалтер має можливість будувати багатопланову систему ведення бухгалтерії, інструментарій для управління податковими платежами та формування звітності перед інвесторами.

Керівник постачання знаходить потужний інструмент управління територіально-віддаленими складами, використання електронних ваг.

Керівник збуту за рахунок контролю разових і довгострокових рамкових контрактів організовує засобами галузевого рішення швидке обслуговування

клієнтів.

Керівник ІТ-відділу, маючи зручні засоби адміністрування, можливість простого і ефективного сполучення з іншими системами автоматизації.

Система «Галактика Харчова промисловість» дозволяє [19]:

- формувати плани і бюджети збуту, виробничу програму, оптимізувати платіжний баланс підприємства.
- враховувати особливості сировини, коригувати рецептурні журнали, формувати виробничі звіти, контролювати незавершене виробництво.
- здійснювати моніторинг фактичних витрат за різними методиками.
- збільшити оборотність коштів, вести інтенсивне цілодобове відвантаження.
- вести бухгалтерський та податковий облік, а також облік в міжнародних стандартах GAAP, ISA і інших для західних інвесторів.
- забезпечити «прозорість» руху матеріальних і фінансових ресурсів.

За допомогою рішення "Галактика Харчова промисловість" забезпечується:

- автоматизоване формування рецептур;
- підтримка характеристик якості сировини;
- управління якістю продукції;
- оптимізація програми виробництва з урахуванням виробничих потужностей;
- контроль термінів придатності сировини і готової продукції;
- управління транспортом і реалізація схем «центрозавозом»;
- контроль звернення тари;
- можливість формування галузевої звітності;
- створення територіально-розподілених виробництв і мереж дистрибуції.

Сьогодні жоден керівник не поспішає впроваджувати нові інформаційні технології без ретельного аналізу визначення їх економічної ефективності та доцільності. Аналіз статистики за раніше автоматизованими підприємствами системою «Галактика» показує наступні результати [9]:

- середній відсоток зниження рівня постійних витрат зазвичай не менше 5%;
- скорочення терміну оборотності оборотних коштів може досягати 12%;
- рівень неліквідних запасів на складі зазвичай вдається знизити на 10-12%;
- загальне зниження витрат може скласти до 15% річного обороту підприємства.

1.3. Актуальні інформаційні системи бухгалтерського обліку хлібозаводу

Існує багато інформаційних систем, призначених для автоматизації обліку хлібопекарської промисловості, але найбільш практичною серед них є інформаційна система «Дебет Плюс».

Система управління підприємством «ДЕБЕТ Плюс» має модульну структуру та дає можливість налаштувати функціональність, необхідну потребам підприємства [2]. Для комплексної автоматизації обліку на підприємствах хлібопекарської промисловості (хлібзаводи, хлібокомбінати, пекарні) використовується конфігурація «Хлібозавод, пекарня» системи «ДЕБЕТ Плюс» [Ошибка! Источник ссылки не найден.]. Дана конфігурація має такі функціональні можливості:

1. Робота із заявками на готову продукцію

- У системі є документ «Заявка», за допомогою якого збирається інформація від покупців про необхідну кількість готової продукції. Заявки ведуться в розрізі конкретних точок доставки (магазинів).
- На підставі заявок від покупців формується заявка у виробництво.

2. Відпуск готової продукції в експедиції

- По складу готової продукції ведеться позмінний облік.
- Накладні на відвантаження покупцям формуються автоматично на підставі прийнятих заявок.

- Відпускні ціни на готову продукцію для покупців автоматично беруться з прайс-листів.
- Є можливість використовувати різні прайс-листи для різних точок доставки одного і того ж покупця.
- При відпуску готової продукції в прайс-листі видно залишки на поточний момент.
- Є можливість надавати знижки (або робити надбавки).
- В системі передбачено ведення маршрутів доставки готової продукції.
- Система формує «перепустку» на кожну машину, яка вивозить готову продукцію.
- Передбачено надання послуг на відпустку давальницької продукції кінцевим споживачам.
- Система формує акти передачі зміни в експедиції.
- Передбачено формування звіту експедитора за зміну.

3. Облік сировини

- Борошно оприбутковується на склад і виписується у виробництво з якісними показниками (зокрема, зі значенням вологості).
- Запаси сировини у виробництві передаються з зміни в зміну на підставі актів передачі залишків.

4. Списання сировини у виробництво

- Є довідник рецептур , в якому задані норми списання сировини на готову продукцію.
- Є таблиця замін - список видів сировини, якими можна замінити інший вид сировини при його відсутності
- Формується «Ордер», в якому вказується продукція, передана на склад готової продукції, номер зміни і час випічки.
- По закінченні зміни формується акт передачі залишків сировини у виробництві наступного начальнику зміни.
- У кінці зміни формується звіт по роботі зміни, в якому зазначаються залишки сировини на початок зміни, кількість списаного сировини на

кожний конкретний вид продукції, кількостей прийнятого зі складу та переданого на склад сировини і залишки на кінець.

- Реалізована можливість переробки браку в санвідходи.
- Реалізовано контроль списання сировини - в кінці місяця формується звіт, в якому система на підставі кількості виготовленої продукції і середньої вологості списаної борошна розраховує, скільки необхідно було списати сировини на кожен вид готової продукції за відповідний період.
- Реалізовано звіт «Витрата сировини за асортиментом» - у кількісному і сумовому вираженні, призначений для контролю списання сировини і отримання сум прямих витрат на виготовлену продукцію.

5. Облік праці та заробітної плати

- У кінці кожної зміни начальник зміни заводить «Робочий аркуш», в якому зазначаються: ПІБ працівника, місце роботи, вид роботи - з довідника видів робіт; береться з відповідною розцінкою, час роботи. У звіт автоматично заноситься виготовлена за зміну продукція.
- В кінці місяця кожен майстер формує «Табель обліку робочого часу», в який для робітників обраної бригади система автоматично з робочих аркушів вибирає відпрацьований час, кількість вечірніх, нічних годин в розрізі видів робіт (розцінок).

Програмний продукт «1С: Бухгалтерія хлібобулочного і кондитерського підприємства» призначений для автоматизації обліку діяльності малих і середніх організацій, що займаються виробництвом, продажем хлібобулочних і кондитерських виробів. Конфігурація «Бухгалтерія хлібобулочного і кондитерського підприємства» має ряд додаткових функціональних можливостей, що дозволяють розширити спектр автоматизування завдань та керується основними бізнес процесами виробничої галузі (Рис. 1.3).



Рис. Ошибка! Текст указанного стиля в документе отсутствует..3. Бізнес процеси конфігурації «Бухгалтерія хлібобулочного і кондитерського підприємства»

Підсистема містить набір довідників і реєстрів відомостей для забезпечення роботи функціоналу хлібобулочного і кондитерського виробництва:

- Довідник «Магазини (ХБК)» призначений для зберігання інформації про точки оптової або роздрібної торгівлі контрагентів, в тому числі про зони доставки.
- Довідник «Види знижок (ХБК)» містить перелік видів знижок, що використовуються для формування набору використовуваних в організації знижок в розрізі контрагентів, договорів та магазинів.
- Довідник «Упаковки (ХБК)» призначений для зберігання додаткових одиниць виміру номенклатури.
- Регістр відомостей «Одиниці виміру номенклатури (ХБК)» призначений для призначення додаткової одиниці виміру номенклатури і вказівки параметрів базової одиниці виміру номенклатури.

- Регістр відомостей «Типові показники вологості номенклатури (ХБК)» призначений для фактичного відображення вологості номенклатури в розрізі складів.

- Регістр відомостей «Відповідність цін номенклатури (ХБК)» призначений для вказівки відповідності відсотка націнки для типів цін з націнкою на підставі базових типів цін.

- Довідник «Зони доставки (ХБК)» призначений для групування магазинів контрагента по різних районах доставки.

- Довідник «Транспортні засоби (ХБК)» призначений для зберігання в системі транспортних засобів, що використовуються для доставки номенклатури та інформації про вантажопідйомності і місткості.

- Довідник «Бригади (ХБК)» призначений для зберігання списку бригад, які працюють у виробничих підрозділах.

Ця підсистема призначена для автоматизації роботи користувачів по введенню нових замовлень клієнтів (магазинів), коригування замовлень, створення замовлень з зовнішніх систем. У підсистемі реалізована можливість роботи в двох режимах:

- З використанням змін столу замовлень.
- Без використання змін столу замовлень.

У підсистему хлібобулочного і кондитерського виробництва доданий документ «Замовлення покупця (ХБК)», який призначений для формування переліку номенклатури, який клієнт має намір придбати. На підставі даних замовлень формується потреба в виробництві продукції, що поставляється клієнтам продукції **[Ошибка! Источник ссылки не найден.]**.

Основним робочим елементом підсистеми управління замовленнями є обробка «АРМ: Управління замовленнями (ХБК)». Автоматизоване робоче місце надає функціонал роботи з замовленнями, дозволяючи в одному робочому місці використовувати функціонал завантаження замовлень з файлів Excel, працювати із завантаженням даних із зовнішніх систем при використанні

адаптера EDI, робити повернення продукції, зворотної тари і черстої продукції.

У даній підсистемі реалізовані наступні функціональні можливості:

- Використання ресурсної специфікації для виробленої номенклатури;
- Формування замовлень на виробництво;
- Диспетчеризація замовлень на виробництво;
- Розрахунок сировини і матеріалів за поточною вологості борошна і по необхідному для виробництва кількості;
- Формування дозволів на заміну сировини і матеріалів;
- Використання механізму виробничих змін.

Для забезпечення потреби у виробництві використовується механізм, заснований на використанні документа «Замовлення на виробництво (ХБК)». Замовлення на виробництво призначений для оформлення в системі потреби на виробництво продукції і напівфабрикатів, виконання робіт і управління виконанням цієї потреби [**Ошибка! Источник ссылки не найден.**].

Замовлення на виробництво може формуватися як вручну, так і на підставі потреби на замовлення покупців. Замовлення на виробництво може перебувати в статусах: «Створено», «До виробництва» і «Закрито».

В системі розроблений механізм диспетчеризації замовлень на виробництво. В дане робоче місце виводиться інформація по замовленнях на виробництво в статусі «До виробництва», відсортованих за датою потреби.

В рамках підсистеми оперативного обліку виробництва реалізований механізм розрахунку оперативних витрат сировини, необхідних для випуску хлібобулочної та кондитерської продукції. Механізм враховує фактичну вологість борошна і надає різні способи розрахунку витрат сировини і інших важливих параметрів (витрат води, відкиду). Механізм дозволяє вводити в розрахунок технологічні заміни і аналоги сировини. Розрахунок витрат сировини виконується за допомогою обробки «Розрахунок сировини і матеріалів (ХБК)».

Для використання змін виробництва використовується довідник «Зміни (ХБК)». У довіднику визначається час роботи зміни з можливістю уточнення до підрозділу. Для організації виробничого процесу в рамках робочої зміни призначені документи «Відкриття виробничої зміни (ХБК)» і «Закриття виробничої зміни (ХБК)».

Документ «Відкриття виробничої зміни (ХБК)» призначений для відкриття виробничої зміни підприємства і зберігання даних про неї в розрізі організації, підрозділу, зміни виробництва, бригади і бригадира.

За розрахованими даними по потребі в сировині і матеріалах, а також виходячи зі сформованого плану на виробництво, в документ підбирається перелік номенклатури, яку планується провести за зміну. За вказаною кількістю номенклатури до виробництва формується підбір необхідної сировини і матеріалів для виробництва. При проведенні документа за вказаною бригадою закріплюється виробничий склад, фіксуються передані матеріали на склад і залишки, якщо вони були до початку зміни. Бригадир є матеріально-відповідальною особою за цим складом [**Ошибка! Источник ссылки не найден.**].

Документ «Закриття виробничої зміни (ХБК)» призначений для:

- Реєстрації факту закінчення роботи зміни;
- Реєстрації факту випуску готової продукції та напівфабрикатів;
- Реєстрації фактичних даних по споживанню номенклатури у виробництві;
- Відпустки додаткових матеріалів і напівфабрикатів у виробництво.

У документі вказується факт випуску готової продукції за зміну, а також фактичні витрати сировини і матеріалів. Документ дозволяє провести інвентаризацію по виробничому складу і вказати фактичні залишки на складі.

При проведенні документа, в першу чергу, автоматично формується документ «Вимога-накладна» для відпустки додатково витрачених матеріалів у виробництво, потім автоматично формується документ «Звіт виробництва за зміну» для випуску напівфабрикатів етапу. Кількість документів відпустки

напівфабрикатів у виробництво буде дорівнює кількості етапів виробництва у виробленої номенклатури.

У підсистемі реалізації і доставки автоматизований процес доставки замовлень покупців, який здійснюється безпосередньо за адресами магазинів клієнтів. Доставка замовлень формується в оперативному порядку, у міру готовності замовлень до відвантаження.

Для оформлення замовлень на доставку в системі використовується автоматизоване робоче місце «АРМ: Доставка замовлень (ХБК)». Основним завданням менеджера по доставці є розподіл замовлень покупців, готових до доставки, за наявними в наявності транспортних засобів, на конкретну дату, з урахуванням зон доставки і вантажопідйомності транспортних засобів.

Для формування і контролю даних по доставці в підсистемі реалізації і доставки розроблений документ «Доставка замовлення (ХБК)». Замовлення на доставку може складатися з декількох замовлень покупця, відповідних одному магазину і одній даті відвантаження.

Основним способом відображення господарських операцій в обліку є введення документів програми, відповідних первинним бухгалтерським документам. Крім того, допускається безпосереднє введення окремих проводок.

По складах може вестися кількісний або кількісно-сумовий облік. У першому випадку оцінка товарів і матеріалів для цілей бухгалтерського і податкового обліку не залежить від того, з якого складу вони отримані. Складський облік може бути відключений, якщо в ньому немає необхідності **[Ошибка! Источник ссылки не найден.]**.

У програмі можна відобразити результати інвентаризації, які автоматично звіряються з даними обліку. На підставі інвентаризації відбивається виявлення надлишків і списання недостач.

Також у конфігурації автоматизований облік операцій надходження і реалізації товарів і послуг. При продажі товарів виписуються рахунки на оплату, оформляються накладні і рахунки-фактури. Всі операції по оптовій торгівлі враховуються в розрізі договорів з покупцями і постачальниками. Для

імпортних товарів враховуються дані про країну походження та номер митної декларації.

Для роздрібною торгівлі підтримується як оперативне відображення роздрібною реалізації, так і відображення продажів за результатами інвентаризації. Товари в роздробі можуть враховуватися за покупними або продажними цінами. Для роздрібних продажів підтримується оплата банківськими кредитами і використання платіжних карт.

У конфігурації підтримується використання декількох типів цін, наприклад: оптова, дрібнооптова, роздрібна, закупівельна... Це спрощує відображення операцій надходження і реалізації. Надано можливість вести облік артикулів номенклатури.

Реалізовано облік руху готівкових та безготівкових грошових коштів та валютних операцій. Підтримується введення і друк платіжних доручень, прибуткових і видаткових касових ордерів. Реалізовано облік грошових документів. На підставі касових документів формується касова книга встановленого зразка.

Автоматизовані операції по розрахунках з постачальниками, покупцями і підзвітними особами (включаючи перерахування грошових коштів на банківські карти співробітників або корпоративні банківські карти), внесення готівки на розрахунковий рахунок і отримання готівки по грошовому чеку, придбання і продаж іноземної валюти. При відображенні операцій суми платежів автоматично розбиваються на аванс і оплату [**Ошибка! Источник ссылки не найден.**].

Автоматизований розрахунок собівартості продукції і послуг, що випускаються основним і допоміжним виробництвом, облік переробки сировини, облік спецодягу, інвентарю та господарського приладдя. Протягом місяця облік випущеної готової продукції ведеться за плановою собівартістю. В кінці місяця розраховується фактична собівартість випущеної продукції і наданих послуг.

У конфігурації ведеться облік розрахунків з персоналом по заробітній платі. Автоматизовані наступні операції:

- відображення в обліку відрядної оплати праці;
- нарахування зарплати працівникам підприємства по окладу з можливістю вказати спосіб відображення в обліку окремо для кожного виду нарахування;
- облік утримань;
- ведення взаєморозрахунків з працівниками аж до виплати зарплати і перерахування зарплати на карткові рахунки працівників;
- депонування.

Конфігурація надає користувачеві набір стандартних звітів, які дозволяють аналізувати дані по залишкам, оборотам рахунків і по проводках в різних розрізах. При формуванні звітів є можливість налаштувати групування, відбір і сортування інформації, виведеної в звіт, виходячи із специфіки діяльності організації і виконуваних користувачем функцій **[Ошибка! Источник ссылки не найден.]**.

Експрес-перевірка ведення обліку допомагає користувачеві в будь-який момент отримати зведену і деталізовану інформацію про коректність даних. Всі перевірки об'єднані в наступні групи:

- положення облікової політики;
- аналіз стану бухгалтерського обліку;
- операції по касі;
- ведення книги продажів з ПДВ;
- ведення книги покупок з ПДВ.

Звіт про результати експрес-перевірки супроводжується коментарями до кожної виконаної перевірки, які містять: предмет контролю, результат перевірки, можливі причини помилок, рекомендації щодо усунення помилок.

Отже, існує багато інформаційних систем, призначених для автоматизації обліку хлібопекарської промисловості, але найбільш практичними серед них є інформаційна система «Дебет Плюс» та ERP модуль 1С «Хлібобулочне і

кондитерське виробництво».

РОЗДІЛ 2. ІНФОРМАЦІЙНА СИСТЕМА ХЛІБЗАВОДУ

2.1. Логічна модель даних

Логічна модель даних – модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання [1].

Виходячи з аналізу предметної галузі, під час проектування на інфологічному рівні була створена інформаційно-логічна модель АІС «Хлібзавод» (Рис. 2.1).

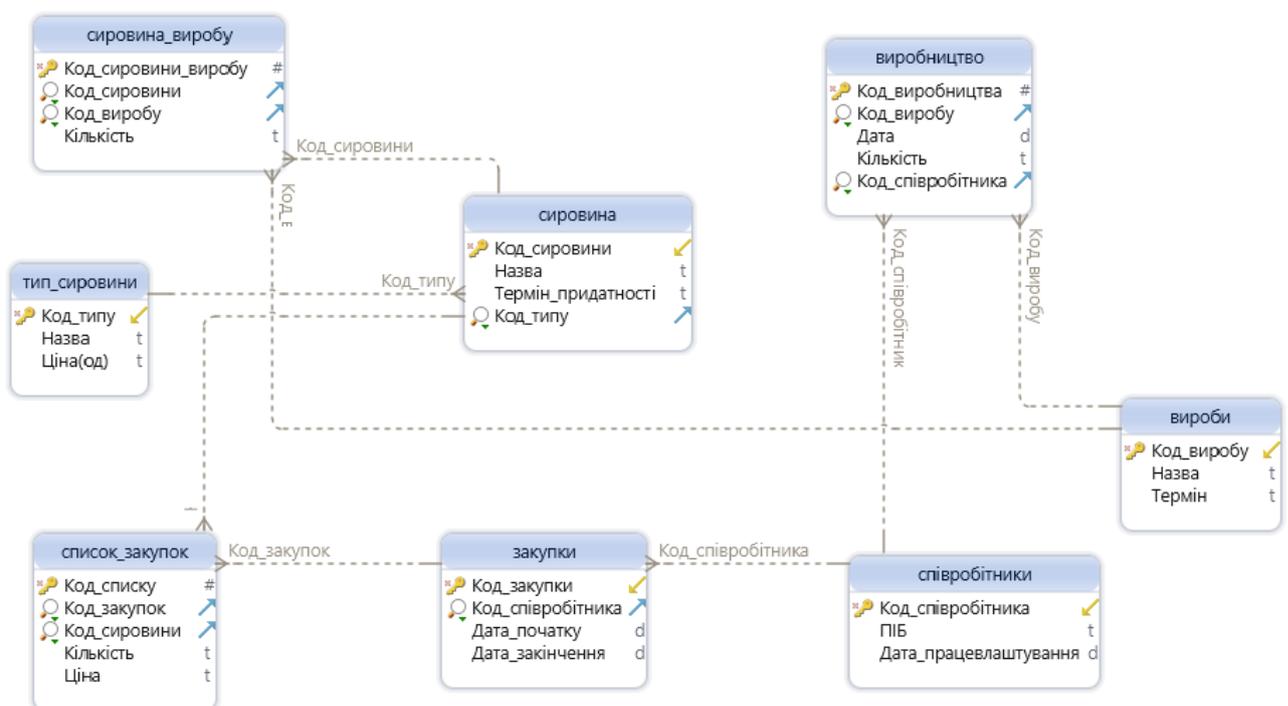


Рис. 2.1. Логічна модель даних АІС «Хлібзавод»

Як видно з рисунка головними сутностями моделі є: Сировина, Закупки, Вироби, Виробництво.

Для створення бази даних використовуємо додаток MySQL-Front. Після завантаження даного програмного забезпечення створюємо нову базу даних «**vurobnuztvo**», для цього використовуємо пункт головного меню «База даних» – «Создать» – «База данных...». SQL запит на створення БД:

```
CREATE DATABASE `vurobnuztvo`
```

Після цього створюємо таблицю «**vurobnuztvo**» (Рис.).

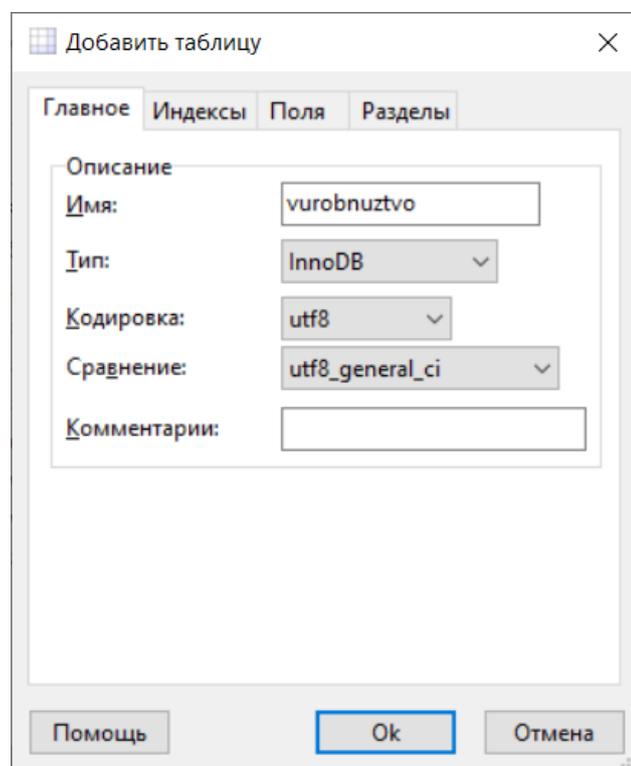


Рис. 2.2. Вікно додавання таблиці «**vurobnuztvo**» у середовищі MySQL-Front

Далі у таблиці «**vurobnuztvo**» створюємо поля за допомогою наступного SQL запиту:

```
CREATE TABLE `vurobnuztvo` (  
  `Id_vurobnuztva` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `Id_vuroby` int(11) unsigned DEFAULT NULL,  
  `Data` date DEFAULT NULL,
```

```

`Kilkist` int(3) DEFAULT NULL,
`Id_spivrobotnika` int(11) unsigned DEFAULT NULL,
PRIMARY KEY (`Id_vurobnuztva`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;

```

Відповідно до наведеної схеми даних (див. Рис. 2.1) створюємо ще сім таблиць з необхідними полями.

Після створення таблиць, їх структура буде наступною (Рис. 3):

Поля (3)			
 Id_vuroby	int(11) unsigned	Нет	<auto_increment>
 Nazva	varchar(50)	Да	<NULL>
 Termin	int(3)	Да	<NULL>
Поля (4)			
 Id_zakupok	int(11) unsigned	Нет	<auto_increment>
 Id_spivrobotnika	int(11) unsigned	Да	<NULL>
 Data_poch	date	Да	<NULL>
 Data_kin	date	Да	<NULL>
Поля (5)			
 Id_spusky	int(11) unsigned	Нет	<auto_increment>
 Id_zakupok	int(11) unsigned	Да	<NULL>
 Id_surovunu	int(11) unsigned	Да	<NULL>
 Kilkist	int(11)	Да	<NULL>
 Zina	int(3)	Да	<NULL>
Поля (3)			
 Id_spivrobotnika	int(11) unsigned	Нет	<auto_increment>
 PIB	varchar(255)	Да	<NULL>
 Data_praz	date	Да	<NULL>
Поля (4)			
 Id_surovunu	int(11) unsigned	Нет	<auto_increment>
 Nazva	varchar(100)	Да	<NULL>
 Termin_prud	int(3)	Да	<NULL>
 Id_tupy	int(11) unsigned	Да	<NULL>
Поля (3)			
 Id_tupy	int(11) unsigned	Нет	<auto_increment>
 Nazva	varchar(255)	Да	<NULL>
 Zina(od)	int(3)	Да	<NULL>
Поля (4)			
 Id_surovunu_vuroby	int(11) unsigned	Нет	<auto_increment>
 Id_surovunu	int(11) unsigned	Да	<NULL>
 Id_vuroby	int(11) unsigned	Да	<NULL>
 Kilkist	int(11)	Да	<NULL>

Рис. 2.3. Структура таблиць інформаційної системи «Хлібзавод»

Повний текст програмного коду SQL копії бази даних наведено у додатку А.

Для заповнення таблиць даними використовуємо програмний продукт Navicat. Заповнюємо дані таблиці «**spivrobotniku**» наступними значеннями (Рис.).

Id_spivrobotnika	PIB	Data_praz
1	Рибак Риба Рибович	2020-10-14
2	Сомач Сом Сомич	2020-10-14
3	Лободняк Максим Петрович	2020-06-19
4	Петрук Олег Олегович	2019-05-16
5	Петро Петро Петрович	2019-04-23

Рис. 2.4. Дані таблиці «**spivrobotniku**»

Відповідно до логічної моделі даних АІС «Хлібзавод» (Рис. 2.), створюємо зв'язок між таблицями «**zakupku**» та «**spivrobotniku**».

Для цього відкриваємо властивості таблиці «**zakupku**» (пункт контекстного меню «**Design Table**») та переходимо на вкладку «**Foreign Keys**». У вікні, що з'явилося вводимо наступні параметри (Рис.).

Name	Field Names	Reference Databa	Reference Table	Foreign Field Names	On Delete	On Update
zakup	Id_spivrobotnika	vurobnuztvo	spivrobotniku	Id_spivrobotnika	RESTRICT	CASCADE

Рис. 2.5. Зв'язок між таблицями «**zakupku**» та «**spivrobotniku**»

Далі відкриваємо таблицю «**zakupku**» та заповнюємо дані про десять закупок співробітниками (Рис.). У полі «**Id_spivrobotnika**» вибираємо потрібного співробітника із списку натиснувши .

Id_zakupok	Id_spivrobotnika	Data_poch	Data_kin
1	10	2020-10-01	2020-10-02
2	9	2020-10-01	2020-10-04
3	8	2020-09-23	2020-09-30
4	7	2020-10-01	2020-10-14
5	6	2020-09-24	2020-09-29
6	5	2020-09-21	2020-09-30
7	4	2020-09-08	2020-09-29
8	3	2020-10-22	2020-10-27
9	2	2020-10-20	2020-10-27

Рис. 2.6. Дані таблиці «**zakupku**»

Відповідно до схеми даних АІС «Хлібзавод» (*Ошибка! Источник ссылки не найден.*) створюємо аналогічні зв'язки між іншими таблицями (Рис.).

▶ spusok_zak1	Id_zakypok	vurobnuztvo	zakypku	Id_zakypok	RESTRICT	CASCADE
spusok_zak2	Id_surovunu	vurobnuztvo	surovuna	Id_surovunu	RESTRICT	CASCADE
▶ vurob1	Id_vuroby	vurobnuztvo	vurobu	Id_vuroby	RESTRICT	CASCADE
vurob2	Id_spivrobotnuka	vurobnuztvo	spivrobotniku	Id_spivrobotnuka	RESTRICT	CASCADE
▶ surov	Id_tupy	vurobnuztvo	tup_surovunu	Id_tupy	RESTRICT	CASCADE
▶ surov_vur1	Id_surovunu	vurobnuztvo	surovuna	Id_surovunu	RESTRICT	CASCADE
surov_vur2	Id_vuroby	vurobnuztvo	vurobu	Id_vuroby	RESTRICT	CASCADE

Рис. 2.7. Зв'язки між таблицями інформаційної системи «Хлібзавод»

Отже, під час моделювання бази даних ІС «Хлібзавод» було створено вісім таблиць, з'єднані між собою зв'язками, схему даних та логічну модель даних автоматизованої інформаційної системи.

2.2. Вдосконалення функціональних можливостей бухгалтерської інформаційної системи хлібозаводу

Виходячи з аналізу предметної галузі, в середовищі ІС була спроектована схема даних інформаційної системи хлібозаводу (Рис. 2.8).

Для створення довідника «Магазини» необхідно натиснути правою кнопкою миші по пункту Довідники дерева конфігурації ІС: підприємство та в контекстному меню вибрати пункт Додати. Відкриється вікно створення нового довідника. Довідник «Магазини» буде використовуватися в підсистемі Маркетинг та Продажі. Для визначення списку реквізитів довідника «Магазини» потрібно перейти на вкладку Дані. Кожен довідник, навіть при відсутності реквізитів, вже має два реквізити по замовчуванню: код та найменування. Щоб додати реквізит необхідно натиснути на кнопку «Додати» та заповнити основні параметри (Таблиця 2.1).

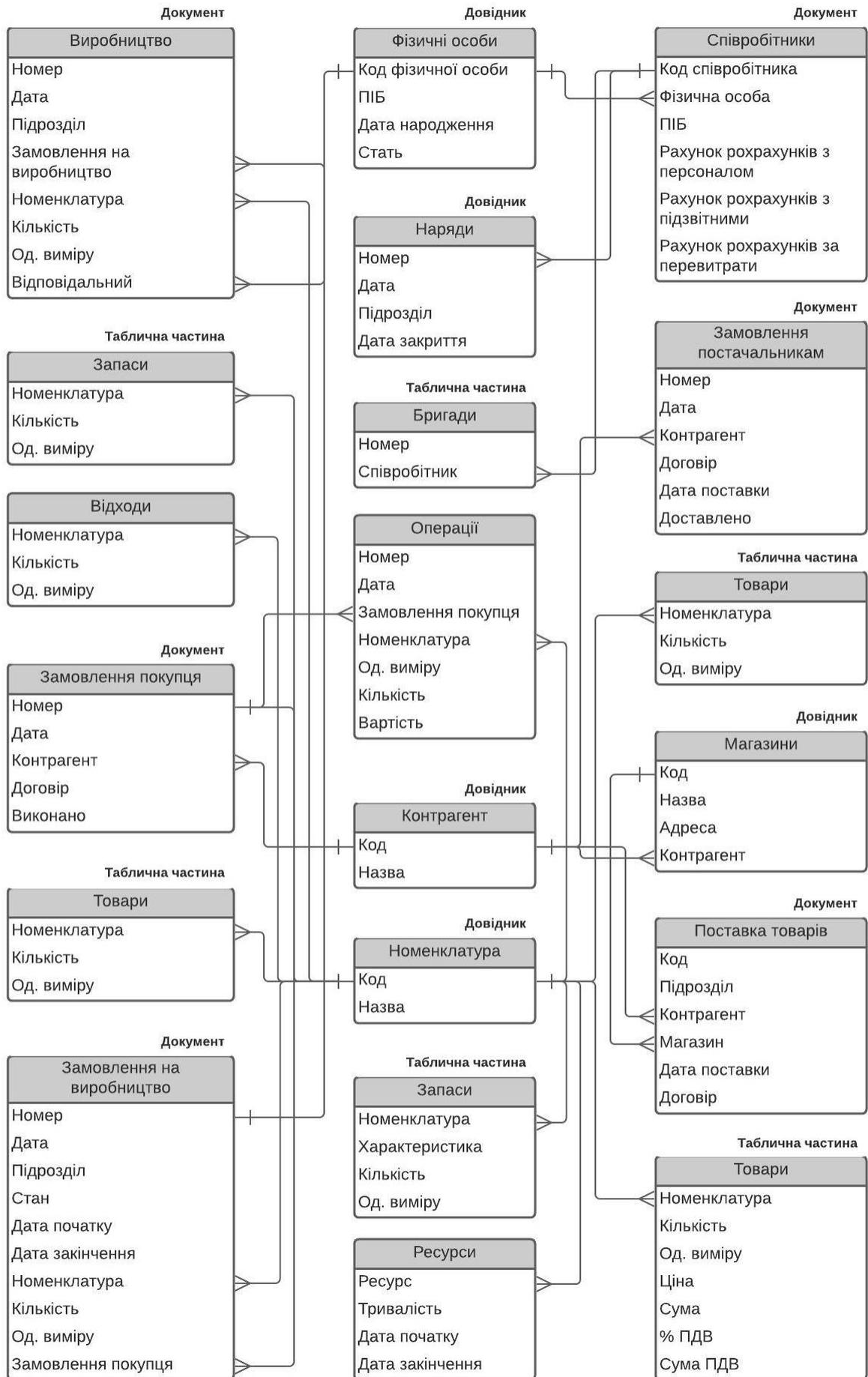


Рис. Ошибка! Текст указанного стиля в документе отсутствует..8. Схема даних інформаційної системи хлібозаводу в середовищі 1С

Таблица Ошибка! Текст указанного стиля в документе отсутствует..1

Реквізити довідника «Магазини»

Ім'я реквізиту	Тип	Довжина
Адреса	Строка	150
Контрагент	СправочникСсылка.Контрагенты	

У довідниках існує дві основні форми: форма елемента та форми списку довідника. Щоб створити форму елемента довідника потрібно натиснути на кнопку «Відкрити» навпроти тексту «Елемента». Елементи форми можна розташувати на свій розсуд, перетягуючи їх мишею по формі. Аналогічним чином створюється форма списку довідника. У формі елемента користувач буде вносити дані по кожному новому запису довідника, а в формі списку зможе переглядати відразу всі ці записи в тому вигляді, який визначився при конструюванні форми. Щоб побачити зміни оновлюємо конфігурацію БД: «Конфігурація» – «Оновити конфігурацію БД» (F7). В режимі 1С додаємо декілька значень у створений довідник «Магазини».

Для створення нового документу «Поставка товарів» потрібно у дереві конфігурації вибрати: «Документи» – «Додати». Відкриється вікно створення нового документа (Рис. 2.9).

Рис. Ошибка! Текст указанного стиля в документе отсутствует..9.

Створення документа «Поставка товарів» в середовищі 1С

Документ «Поставка товарів» буде використовуватися в підсистемі аналогічній довідникам – Маркетинг та Продажі. Для визначення списку реквізитів документа «Поставка товарів» потрібно перейти на вкладку **Дані**. Кожен документ також має два реквізити по замовчуванню: *код* та *найменування*. Щоб додати реквізит необхідно натиснути на кнопку «Додати» та заповнити основні параметри (*Таблиця Ошибка! Текст указанного стиля в документе отсутствует..2*).

Таблиця Ошибка! Текст указанного стиля в документе отсутствует..2

Реквізити документа «Поставка товарів»

Ім'я реквізиту	Тип
Підрозділ	СправочникСсылка.СтруктурныеЕдиницы
Контрагент	СправочникСсылка.Контрагенты
Магазин	СправочникСсылка.Магазини
Дата поставки	Дата
Договір	СправочникСсылка.ДоговорыКонтрагентов

Створимо для документа «Поставка товарів» табличну частину, в яку автоматично буде переноситися список номенклатури, кількість та сума, з

іменем Товари та реквізитами (Таблиця *Ошибка! Текст указанного стиля в документе отсутствует..3*).

Таблиця Ошибка! Текст указанного стиля в документе отсутствует..3

Реквізити табличної частини документа «Поставка товарів»

Ім'я реквізиту	Тип	Довжина
Номенклатура	СправочникСсылка.Номенклатура	
Кількість	Число	12
ОдиниціВиміру	СправочникСсылка.ЕдиницыИзмерения	
Ціна	Число	12
Сума	Число	12
Відсоток ПДВ	Число	10
Сума ПДВ	Число	12

Для зберігання документів на вкладці **Журнали** відмічаємо журнал *ПоставкаВМагазин*. Аналогічно до довідників, створюємо форму документа, натиснувши на кнопку «Відкрити» навпроти тексту «Документа». Аналогічним чином створюється форма списку документа. У формі списку користувач зможе переглядати відразу всі записи.

Для створення нового звіту потрібно у дереві конфігурації вибрати: «Звіти» – «Додати». Задаємо звіту ім'я «Реалізація товарів» та натискаємо кнопку «Відкрити схему компонування даних». Далі необхідно натиснути правою кнопкою миші по пункту «**Набір даних**» – «**Додати набір даних – Запит**». Після цього формуємо запит на реалізацію виготовлених товарів, за допомогою кнопки «**Конструктор запиту**», що має наступний код:

ВИБРАТЬ

ПоставкаТоварів.Магазин,

ПоставкаТоварів.Товари.(

Номенклатура,

*(Ціна * Кількість) КАК Сума,*

ВідсотокПДВ,

Кількість,

Ціна,

*(Ціна * Кількість * ВідсотокПДВ/100) КАК СумаПДВ*

),

ПоставкаТоварів.Підрозділ,

ПоставкаТоварів.Контрагент

ИЗ

Документ.ПоставкаТоварів КАК ПоставкаТоварів

Для перегляду даного запиту необхідно перейти в ІС: **«Маркетинг та продажі»** – **«Реалізація товарів»** та натиснути **«Сформувати»**.

На завершення потрібно зробити висновок, що для систематизації даних підтримки прийняття управлінських рішень, керівниками структурних підрозділів хлібозаводу потрібно розробити систему звітів для адміністрування та обліку інформації, що стосується реалізації товарів.

2.3. Функціональні можливості ІС хлібозаводу

Для розробки програмного забезпечення інформаційної системи створюємо новий проект інформаційної системи за допомогою пункту головного меню **«Файл»** – **«Создать»** – **«Проект...»** – **«Приложение Windows Forms»** програмного забезпечення Microsoft Visual Studio.

Створюємо нову форму проекту **«Form2»** за допомогою пункту головного меню **«Проект»** – **«Добавить форму»**, яка буде відображати дані про виробу. У властивостях форми задаємо наступні параметри:

- **Text** – Вироби (встановлює ім'я форми);
- **MinimizeBox** – False (вказує, чи буде доступна кнопка мінімізації вікна);
- **MaximizeBox** – False (вказує, чи буде доступна кнопка максимізації вікна в заголовку форми);
- **StartPosition** – CenterScreen (вказує на початкову позицію, з якою форма з'являється на екрані – по центру);

- **BackColor** – Honeydew (фоновий колір форми).

На створеній формі розміщуємо наступні компоненти з панелі інструментів
(*Рис. Ошибка! Текст указанного стиля в документе отсутствует..10*).

Для відображення даних про вироби у вигляді таблиці, до форми «**Form2**» додаємо наступний програмний код:

```
public partial class Form2 : Form
{
    MySqlCommand cmd;
    string query;
    Boolean insert_mode;
    public Form2()
    {
        InitializeComponent();
        query = @"SELECT vurobu.Id_vuroby, vurobu.Nazva_vuroby, vurobu.Termin FROM
vurobu
ORDER BY vurobu.Nazva_vuroby ASC";
        MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
        DataSet ds = new DataSet();
        da.Fill(ds, "vurobu");
        dataGridView1.DataSource = ds;
        dataGridView1.DataMember = "vurobu";
        dataGridView1.Columns[0].HeaderText = "Код виробу";
        dataGridView1.Columns[1].HeaderText = "Назва виробу";
        dataGridView1.Columns[2].HeaderText = "Придатність";
        dataGridView1.Columns[0].Width = 90;
        dataGridView1.Columns[1].Width = 160;
        dataGridView1.Columns[2].Width = 80;
        dataGridView1.Refresh();
    }
}
```

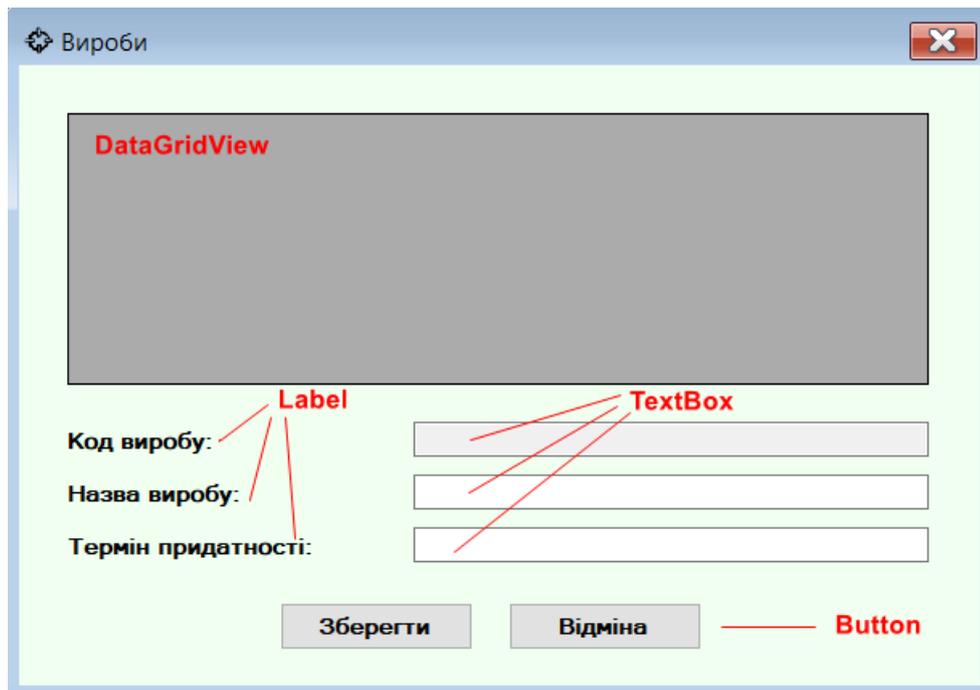


Рис. Ошибка! Текст указанного стиля в документе отсутствует..10.

Форма «Вироби» АІС «Хлібзавод»

DataGridView – даний елемент управління дозволяє організувати дані у вигляді таблиці. Label – елемент управління призначений для створення підписів до інших елементів управління. TextBox – елемент управління призначений для введення користувачем текстових даних. Button – кнопка.

На початку коду було створено глобальні змінні «cmd», «query» та «insert_mode». Відповідно «cmd» - змінна для робота з базою даних MySQL, «query» - змінна для тексту запиту, «insert_mode» - змінна для переключення між режимами редагування даних та додавання нових записів. Даний код містить SQL запит для відбору всіх значень із таблиці «vuroby» з сортуванням по назві виробу.

Для функціонування кнопки «Зберегти», додаємо на подію «Click» наступний програмний код:

```
private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO vurobu (Nazva_vuroby, Termin) VALUES (";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
    }
}
```

```

query2 = query2 + "" + Convert.ToString(textBox3.Text) + """;
}
else
{
query2 = "UPDATE `vurobu` SET";
query2 = query2 + "`Nazva_vuroby`=" + Convert.ToString(textBox2.Text) + ",";
query2 = query2 + "`Termin`=" + Convert.ToString(textBox3.Text) + "" ";
query2 = query2 + "WHERE `Id_vuroby`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
}
cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "vurobu");
dataGridView1.DataSource = ds;
dataGridView1.Refresh();
for (int i = 0; i < dataGridView1.RowCount; i++)
{
dataGridView1.Rows[i].Selected = false;
if (dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox2.Text))
{
dataGridView1.Rows[i].Selected = true;
break;
}
}
dataGridView1.Enabled = true;
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
insert_mode = true;
}

```

Після натиснення на кнопку «Зберегти», додається новий запис у таблицю форми «Вироби».

Для функціонування кнопки «Відміна», додаємо на подію «Click» наступний програмний код:

```

private void button2_Click(object sender, EventArgs e)
{
dataGridView1.Enabled = true;
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
insert_mode = true;
}

```

Після натиснення на кнопку «Відміна», всі дані, які були у компонентах форми «Вироби» очищуються.

Аналогічно до попередньої форми, створюємо «Form3» (відобразатиме дані про тип сировини), «Form4» (відобразатиме дані про співробітників), «Form5» (відобразатиме дані про сировину), «Form6» (відобразатиме дані про закупки сировини), «Form9» (відобразатиме дані про виробництво продукції).

Для створення підключення до бази даних MySQL виконаємо пункт головного меню «Средства» – «Подключиться к базе данных...» програмного забезпечення Microsoft Visual Studio. По етапам вибираємо «База даних», далі «Набір даних». Створюємо нове підключення вказавши джерело даних **MySQL Database**. Далі вказуємо параметри підключення до БД (**server name** та **user name**) та перевіряємо підключення.

Створюємо нову форму проекту «Form6», яка буде відобразити дані про закупки сировини. Задаємо відповідні параметри у властивостях форми та розміщуємо компоненти з панелі інструментів.

Для створення інформаційної системи з БД MySQL потрібно імпортувати дві бібліотеки **MySql.Data.MySqlClient** (представляє відкрите підключення до бази даних MySQL) та **MySql.Data.Types** (визначає конкретний тип даних).

Створюємо глобальні змінні «cmd» (змінна для робота з базою даних MySQL), «query» (змінна для тексту запиту) та «insert_mode» (змінна для переключення між режимами редагування даних та додавання нових записів).

Створюємо програмний код на подію «Shown» форми «Закупки» («Form6») для відображення списку співробітників та сировини у відповідних компонентах «ComboBox»:

```
private void Form6_Shown(object sender, EventArgs e)
{
    insert_mode = true;
    DateTime data1;
    data1 = DateTime.Today;
    dateTimePicker1.Value = data1;
    dateTimePicker2.Value = data1.AddDays(7);
    dateTimePicker3.Value = data1;
    dateTimePicker4.Value = data1;
}
```

```

string query2 = @"SELECT spivrobotnuku.Id_spivrobotnuka, spivrobotnuku.PIB
FROM spivrobotnuku ORDER BY spivrobotnuku.PIB ASC";
using (MySqlDataAdapter da_sur = new MySqlDataAdapter(query2, Form1.con))
{
    DataTable table = new DataTable();
    table.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_sur.Fill(table);
    comboBox1.DataSource = table;
    comboBox1.DisplayMember = table.Columns["PIB"].ColumnName;
    comboBox1.ValueMember = table.Columns["Id_spivrobotnuka"].ColumnName;
    comboBox1.Text = "";
}
string query3 = @"SELECT surovuna.Id_surovunu, surovuna.Nazva_surovunu
FROM surovuna GROUP BY surovuna.Nazva_surovunu";
using (MySqlDataAdapter da_tup = new MySqlDataAdapter(query3, Form1.con))
{
    DataTable table2 = new DataTable();
    table2.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_tup.Fill(table2);
    comboBox2.DataSource = table2;
    comboBox2.DisplayMember = table2.Columns["Nazva_surovunu"].ColumnName;
    comboBox2.ValueMember = table2.Columns["Id_surovunu"].ColumnName;
    comboBox2.Text = "";
}
}
}

```

Відповідно до коду, при запуску форми у компонентах «**dateTimePicker**» буде відображатися поточна дата.

Для відображення у таблиці компоненту «**dataGridView1**» закупок сировини за обраний період, на подію «**Click**» кнопки «**Показати**» додаємо наступний програмний код:

```

private void button1_Click(object sender, EventArgs e)
{
    if (dateTimePicker2.Value < dateTimePicker1.Value)
    {
        MessageBox.Show("Невірно вказані дати", "Увага", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        return;
    }
    string data_p, data_z;
    DateTime date1 = dateTimePicker1.Value;
    DateTime date2 = dateTimePicker2.Value;
    data_p = date1.ToString("yyyy-MM-dd");
    data_z = date2.ToString("yyyy-MM-dd");
    query = @"SELECT zakypku.Id_zakypok, spivrobotnuku.PIB, zakypku.Data_poch,
    zakypku.Data_kin, (zakypku.Data_kin - zakypku.Data_poch) AS dni_zakypok,
    zakypku.Id_spivrobotnuka FROM zakypku
    Inner Join spivrobotnuku ON zakypku.Id_spivrobotnuka = spivrobotnuka

```

```

WHERE zakypku.Data_poch >= "" + data_p + @" AND
zakypku.Data_kin <= "" + data_z + """;
MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "zakypku");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "zakypku";
dataGridView1.Columns[0].HeaderText = "Код закупки";
dataGridView1.Columns[1].HeaderText = "ПІБ";
dataGridView1.Columns[2].HeaderText = "Початок";
dataGridView1.Columns[3].HeaderText = "Закінчення";
dataGridView1.Columns[4].HeaderText = "Кількість днів";
dataGridView1.Columns[5].Visible = false;
dataGridView1.Columns[0].Width = 50;
dataGridView1.Columns[1].Width = 200;
dataGridView1.Columns[2].Width = 70;
dataGridView1.Columns[3].Width = 70;
dataGridView1.Columns[4].Width = 70;
dataGridView1.Columns[5].Width = 80;
dataGridView1.Refresh();
}

```

Після запуску коду з'явиться таблиця з наступними даними: код закупки, співробітники, які відповідають за закупки сировини, дати початку та закінчення, кількість днів.

Для функціонування кнопки «Зберегти» таблиці компоненту «**dataGridView1**», додаємо на подію «**Click**» наступний програмний код:

```

private void button2_Click(object sender, EventArgs e)
{
string query2;
if (insert_mode == true)
{
query2 = "INSERT INTO `zakypku` (`Id_spivrobotnika`, `Data_poch`, `Data_kin`) VALUES
(";
query2 = query2 + "" + Convert.ToString(comboBox1.SelectedValue) + """;
query2 = query2 + "" + dateTimePicker3.Value.ToString("yyyy-MM-dd") + """;
query2 = query2 + "" + dateTimePicker4.Value.ToString("yyyy-MM-dd") + """;
}
else
{
query2 = "UPDATE `zakypku` SET";
query2 = query2 + "`Id_spivrobotnika` = "" + Convert.ToString(comboBox1.SelectedValue)
+ """;
query2 = query2 + "`Data_poch` = "" + dateTimePicker3.Value.ToString("yyyy-MM-dd") +
""";
query2 = query2 + "`Data_kin` = "" + dateTimePicker4.Value.ToString("yyyy-MM-dd") + ""
";
}
}

```

```

query2 = query2 + "WHERE `Id_zakypok`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
}
cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "zakypku");
dataGridView1.DataSource = ds;
dataGridView1.Refresh();
try
{
for (int i = 0; i < dataGridView1.RowCount; i++)
{
dataGridView1.Rows[i].Selected = false;
if (dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(comboBox1.Text))
{
dataGridView1.Rows[i].Selected = true;
break;
}
}
}
catch
{
dataGridView1.Rows[0].Selected = true;
}
comboBox1.Enabled = true;
dataGridView1.Enabled = true;
dataGridView2.Enabled = true;
textBox2.Text = "";
comboBox1.Text = "";
insert_mode = true;
}

```

Після натиснення на кнопку «Зберегти», додається новий запис у таблицю «**dataGridView1**» форми «Закупки».

Для відображення сировини відповідно до закупки у другій таблиці компоненту «**dataGridView2**» форми «Закупки», додаємо на подію «Click» першої таблиці компоненту «**dataGridView1**» наступний програмний код:

```

private void dataGridView1_Click(object sender, EventArgs e)
{
string query3;
{
query3 = @"SELECT spusok_zakypok.Id_spusky, surovuna.Nazva_surovunu,
spusok_zakypok.Kilkist, spusok_zakypok.Zina, spusok_zakypok.Id_surovunu,

```

```

zakypku.Id_zakypok FROM surovuna , spusok_zakypok , zakypku
WHERE surovuna.Id_surovunu = spusok_zakypok.Id_surovunu AND
spusok_zakypok.Id_zakypok = " +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
query3 = query3 + @" GROUP BY surovuna.Nazva_surovunu";
MySqlDataAdapter da = new MySqlDataAdapter(query3, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "spusok_zakypok");
dataGridView2.DataSource = ds;
dataGridView2.DataMember = "spusok_zakypok";
dataGridView2.Columns[0].HeaderText = "Код списку";
dataGridView2.Columns[1].HeaderText = "Сировина";
dataGridView2.Columns[2].HeaderText = "Кількість";
dataGridView2.Columns[3].HeaderText = "Ціна";
dataGridView2.Columns[4].Visible = false;
dataGridView2.Columns[5].Visible = false;
dataGridView2.Columns[0].Width = 100;
dataGridView2.Columns[1].Width = 200;
dataGridView2.Columns[2].Width = 90;
dataGridView2.Columns[3].Width = 90;
dataGridView2.Refresh();
}
}

```

Після запуску коду у таблиці компоненту «**dataGridView2**» з'явиться запис відповідно до обраної закупки таблиці «**dataGridView1**» з наступними даними: код списку, сировина, яку закупає співробітник, кількість та ціна сировини.

Для функціонування кнопки «**Зберегти**» таблиці компоненту «**dataGridView2**», додаємо на подію «**Click**» наступний програмний код:

```

private void button4_Click(object sender, EventArgs e)
{
string query2;
if (insert_mode == true)
{
query2 = "INSERT INTO `spusok_zakypok` (`Id_surovunu`, `Kilkist`, `Zina`) VALUES (";
query2 = query2 + "" + Convert.ToString(comboBox2.SelectedValue) + """;
query2 = query2 + "" + Convert.ToString(textBox2.Text) + """;
query2 = query2 + "" + Convert.ToString(textBox3.Text) + """;
}
else
{
query2 = "UPDATE `spusok_zakypok` SET";
query2 = query2 + "`Id_surovunu`=" + Convert.ToString(comboBox2.SelectedValue) +
""";
query2 = query2 + "`Kilkist`=" + Convert.ToString(textBox2.Text) + """;
query2 = query2 + "`Zina`=" + Convert.ToString(textBox3.Text) + """;
query2 = query2 + "WHERE `Id_spusky`=" +
Convert.ToString(dataGridView2.CurrentRow.Cells[0].Value);
}
}

```

```

}
cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "spusok_zakupok");
dataGridView1.DataSource = ds;
dataGridView2.Refresh();
dataGridView1_Click(this, e);
dataGridView1.Enabled = true;
dataGridView2.Enabled = true;
textBox2.Text = "";
comboBox2.Text = "";
insert_mode = true;
}

```

Після натиснення на кнопку «Зберегти», додається новий запис у таблицю «**dataGridView2**» форми «Закупки» відповідно до обраної закупки таблиці «**dataGridView1**».

Для функціонування кнопки «Відміна», додаємо на подію «Click» наступний програмний код:

```

private void button5_Click(object sender, EventArgs e)
{
comboBox2.Enabled = true;
dataGridView1.Enabled = true;
dataGridView2.Enabled = true;
textBox2.Text = "";
insert_mode = true;
}

```

Після натиснення на кнопку «Відміна», всі дані, які були у компонентах форми «Закупки» очищуються.

Більш детальний програмний код ІС хлібозаводу наведений у додатку Б.

Отже, під час розробки ІС «Хлібзавод» було створено підключення даної ІС до бази даних MySQL, що дозволяє організувати дані у вигляді таблиці та розглянуто детальне створення форм на прикладі форми «Закупки».

За допомогою інформаційної системи можна сформувати накладну на закупку сировини для кожного співробітника. Для цього потрібно у головному меню інформаційної системи вибрати пункт «Довідники» – «Співробітники» та натиснути на кнопку «Накладна». Дана кнопка відкриває документ Word, в

якому знаходиться автоматично згенерована накладна, в якій містяться дані про постачальника, товар, кількість товару, ціна без ПДВ та сума без ПДВ (*Рис. **Ошибка! Текст указанного стиля в документе отсутствует..***).

Видаткова накладна № 1 від 12.10.2020 р.

Постачальник: ТОВ «Підприємство»

Покупець: ПАТ «Хлібзавод»

Договір: купівлі-продажу № 1 від 10.10.2020 р.

№	Товар	Кількість	Ціна без ПДВ	Сума без ПДВ
1	борошно	80	160	12800

Всього: 12800

Сума без ПДВ: 2560

Всього з ПДВ: 15360

Від постачальника: Садчиков В. С.

Отримав(ла): Редька Ігор Леонідович

*Рис. **Ошибка! Текст указанного стиля в документе отсутствует..11.***

Приклад автоматично згенерованої накладної на товар

Створена інформаційна система може сформувати наступні звіти:

- **Залишок сировини.** Для отримання звіту потрібно у головному меню інформаційної системи вибрати пункт «Звіт» – «Залишок сировини». Звіт забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, в якій відображається залишок по кожній сировині (*Рис. **Ошибка! Текст указанного стиля в документе отсутствует..12.***).

	A	B	C	D	E
1					
2	№ з/п	Сировина	Тип сировини	Залишок	
3	1	сіль	йодована	-2	
4	2	дріжджі	компонент закваски	-6	
5	3	борошно	пшеничне	55	
6	4	вода	рідкий інгредієнт	73	
7	5	кефір	рідкий інгредієнт	58	
8	6	дріжджі	сухі дріжджі	5	
9	7	цукор	тростниковий	8	
10					

*Рис. **Ошибка! Текст указанного стиля в документе отсутствует..12.***

Приклад автоматично згенерованої

таблиці звіту по залишку сировини

• **Витрати на сировину.** Для отримання звіту потрібно у головному меню інформаційної системи вибрати пункт «Звіт» – «Витрати на сировину». Звіт забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, в якій відображаються витрати по кожній сировині (*Рис. Ошибка! Текст указанного стиля в документе отсутствует..13*).

	A	B	C	D	E	F	G
1							
2	№ з/п	Сировина	Тип сировини	Ціна	Кількість	Витрати	
3	1	борошно	житнє	23	4	92	
4	2	крохмаль	картопляний	54	1	54	
5	3	цукор	пальмовий	12	1	12	
6	4	борошно	пшеничне	13	80	1040	
7	5	вода	рідкий інгредієнт	5	12	60	
8	6	цукор	тростниковий	56	10	560	
9							

Рис. Ошибка! Текст указанного стиля в документе отсутствует..13.

*Приклад автоматично згенерованої
таблиці звіту витрат на сировину*

• **Виробництво.** Для отримання звіту потрібно у головному меню інформаційної системи вибрати пункт «Звіт» – «Виробництво». Звіт забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, в якій відображаються виготовлені вироби, дата виготовлення та кількість (*Рис. Ошибка! Текст указанного стиля в документе отсутствует..14*).

	A	B	C	D	E
1					
2	№ з/п	Вироби	Дата	Кількість	
3	1	Пиріжки	01.10.2020	80	
4	2	Пироги	01.10.2020	60	
5	3	Соломка	01.10.2020	85	
6	4	Хлібні палички	02.10.2020	100	
7	5	Хлібні хрусти	01.10.2020	90	
8					

Рис. Ошибка! Текст указанного стиля в документе отсутствует..14.

*Приклад автоматично згенерованої
таблиці звіту по виробництву*

• **Закупки.** Для отримання звіту потрібно у головному меню інформаційної системи вибрати пункт «Звіт» – «Закупки». Звіт забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, в якій відображаються закупки співробітниками сировини, дата, кількість, ціна закупок та витрачена сума коштів (*Рис. Ошибка! Текст указанного стиля в документе отсутствует..15*).

	A	B	C	D	E	F	G	H
1								
2	№ з/п	ПІБ	Сировина	Кількість	Ціна	Початок	Кінець	Витрачено (грн)
3	1	Бондар Олег Александрович	кефір	70	140	01.10.2020	14.10.2020	9800,00
4	2	Жук Олег Рибович	дріжджі	10	189	01.10.2020	04.10.2020	1890,00
5	3	Лободняк Максим Петрович	цукор	12	234	22.10.2020	27.10.2020	2808,00
6	4	Петро Петро Петрович	дріжджі	1	20	21.09.2020	30.09.2020	20,00
7	5	Петрук Олег Олегович	сіль	1	20	08.09.2020	29.09.2020	20,00
8	6	Петрук Петро Петрович	вода	88	200	23.09.2020	30.09.2020	17600,00
9	7	Петрушка Сом Віталійович	вода	88	250	24.09.2020	29.09.2020	22000,00
10	8	Редька Ігор Леонідович	борошно	80	160	01.10.2020	02.10.2020	12800,00
11							Всього:	66938,00

Рис. Ошибка! Текст указанного стиля в документе отсутствует..15.

*Приклад автоматично згенерованої
таблиці звіту по закупках*

• **Реалізація виробів.** Для отримання звіту потрібно у головному меню інформаційної системи вибрати пункт «Звіт» – «Реалізація виробів». Звіт забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, в якій відображається продаж виробів, кількість та ціна виробів, дата продажу, а також сума отриманих коштів (*Рис. Ошибка! Текст указанного стиля в документе отсутствует..*).

	A	B	C	D	E	F	G
1							
2	№ з/п	Виріб	Кількість	Ціна	Дата	Отримано (грн)	
3	1	Пиріжки	80	189	01.10.2020	15120,00	
4	2	Пирогои	60	250	01.10.2020	15000,00	
5	3	Соломка	85	200	01.10.2020	17000,00	
6	4	Хлібні палички	100	160	02.10.2020	16000,00	
7	5	Хлібні хрусти	90	140	01.10.2020	12600,00	
8					Всього:	75720,00	
9							

Рис. Ошибка! Текст указанного стиля в документе отсутствует..16.

*Приклад автоматично згенерованої
таблиці звіту з реалізації виробів*

Інтерфейс користувача представлений на формі «**Form1**». Для головного меню, використовується компонент «**MenuStrip**» з панелі інструментів та додаємо пункти головного меню АІС «Хлібзавод» (*Рис. Ошибка! Текст указанного стиля в документе отсутствует..17*).

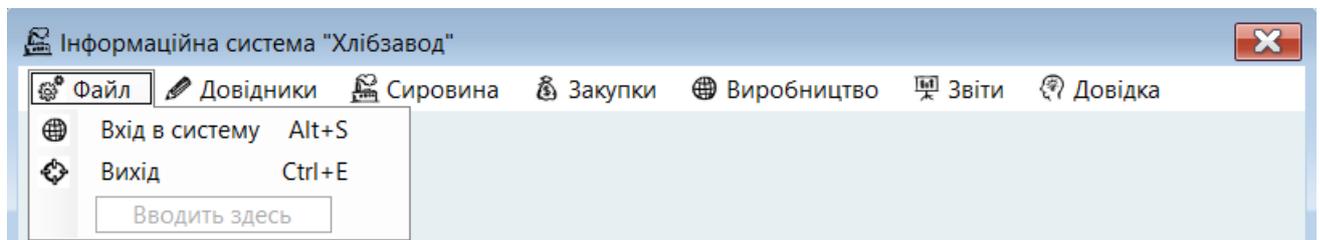


Рис. Ошибка! Текст указанного стиля в документе отсутствует..17.

Головне меню АІС «Хлібзавод»

Головне меню складається з наступних пунктів:

1) «**Файл**» включає в себе два підпункти:

- *Вхід в систему* – забезпечує відкриття форми «**Вхід в систему**» у вигляді модального вікна. Форма входу дозволяє ввести ім'я користувача, пароль та сервер для підключення до БД. При натисненні на кнопку «**ОК**» виконується наступний програмний код, за допомогою якого відбувається перевірка на правильність введення даних та підключення до бази даних «**vurobnuztvo**»:

```
private void button1_Click(object sender, EventArgs e)
{
    Form1.con.Close();
}
```

```

string user_name = textBox1.Text;
string user_password = textBox2.Text;
string user_connection_string;
string user_server = textBox3.Text;
if (user_password == "")
{
user_connection_string = "server=" + user_server + ";User Id=" + user_name +
";Persist Security Info=True; database=vurobnuztvo";
}
else
{
user_connection_string = "server=" + user_server + ";User Id=" + user_name +
";password=" + user_password + ";Persist Security Info=True;
database=vurobnuztvo";
}
try
{
Form1.con.ConnectionString = user_connection_string;
Form1.con.Open();
MessageBox.Show("Підключення відбулось успішно", "Підключення",
MessageBoxButtons.OK, MessageBoxIcon.Information);
this.Close();
}
catch
{
MessageBox.Show("Помилка при підключенні", "Увага",
MessageBoxButtons.OK, MessageBoxIcon.Question);
}
}

```

- *Вихід* – забезпечує закриття програмного забезпечення.

2) «Довідники» включає в себе три підпункти:

- *Вироби* – забезпечує відкриття форми «**Вироби**» у вигляді модального вікна. Форма дозволяє переглядати, додавати, редагувати, видаляти дані про вироби.
- *Тип сировини* – забезпечує відкриття форми «**Тип сировини**» у вигляді модального вікна. Форма дозволяє переглядати, додавати, редагувати, видаляти дані про тип сировини.
- *Співробітники* – забезпечує відкриття форми «**Співробітники**» у вигляді модального вікна. Форма дозволяє переглядати, додавати, редагувати, видаляти дані про співробітників.

- 3) **«Сировина»** – забезпечує відкриття форми **«Сировина»** у вигляді модального вікна. Форма дозволяє переглядати, додавати, редагувати, видаляти та сортувати дані про сировину.
 - 4) **«Закупки»** – забезпечує відкриття форми **«Закупки»** у вигляді модального вікна. Форма дозволяє переглядати, додавати, редагувати, видаляти та сортувати по даті дані про закупки. За допомогою форми можна бачити відповідальних співробітників за закупки сировини, кількість днів, які витрачаються при закупках, кількість та ціну сировини.
 - 5) **«Виробництво»** – забезпечує відкриття форми **«Виробництво»** у вигляді модального вікна. Форма дозволяє переглядати, додавати, редагувати, видаляти та сортувати дані про вироби. За допомогою форми можна бачити відповідальних співробітників за виготовлення виробів, дату виготовлення виробів та їх кількість.
- б) **«Звіти»** включає в себе підпункти:
- Залишок сировини.
 - Витрати на сировину.
 - Виробництво.
 - Закупки.
 - Реалізація виробів.

Отже, під час розробки інтерфейсу ІС «Хлібзавод» було створено головне меню, за допомогою якого можна швидко отримати доступ до форм із даними.

ВИСНОВКИ

Проаналізувавши хлібопекарську галузь та створивши інформаційну систему підтримки діяльності хлібозаводу, можна зробити наступні висновки:

1. Ринок хлібобулочних виробів сьогодні практично повністю представлений продукцією вітчизняного виробництва, що пов'язано з короткими термінами зберігання і складністю транспортування; експорт в сегменті зростає, основними споживачами українських ХБВ є США; динаміка імпорту втричі перевищує темпи росту експорту, 99% усієї імпортової продукції поставляється в Україну з країн Європейського Союзу.

2. Серед закордонних вчених, які досліджували технології виготовлення хліба, можна відмітити: С.М. Rosell, S. Cauvain, M. Seguchi, C. Wrigley, I. Batey та інші; сьогодні жоден керівник не поспішає впроваджувати нові інформаційні технології без ретельного аналізу визначення їх економічної ефективності та доцільності; аналіз статистики за раніше автоматизованими підприємствами системою «Галактика» показує наступні результати: середній відсоток зниження рівня постійних витрат зазвичай не менше 5%, скорочення терміну оборотності оборотних коштів може досягати 12%, рівень неліквідних запасів на складі зазвичай вдається знизити на 10-12%, загальне зниження витрат може скласти до 15% річного обороту підприємства.

3. Існує багато інформаційних систем, призначених для автоматизації обліку хлібопекарської промисловості, але найбільш практичними серед них є інформаційна система «Дебет Плюс» та ERP модуль 1С «Хлібобулочне і кондитерське виробництво».

4. Головними сутностями логічної моделі бази даних інформаційної системи є: Сировина, Закупки, Вироби, Виробництво.

5. Розроблена інформаційна система обліку хлібобулочних виробів для хлібозаводу складається з бази даних та програмного додатку – клієнтської частини. Для зберігання інформації про хлібобулочні вироби використана база даних MySQL. Розроблено інформаційно-логічну і даталогічну моделі та створено саму базу даних. Клієнтську частину розроблено в середовищі програмування Microsoft Visual Studio, мова програмування C#. Для зв'язку з базою даних використано джерело даних MySQL Database.

Таким чином, створена інформаційна система повинна значно полегшити та прискорити роботу з обліку хлібобулочних виробів, а також полегшити створення відповідної звітності. Перспективою подальших розвідок у даному напрямку є перевірка програмного продукту на відповідність потребам хлібозаводу, його розвиток та вдосконалення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Conceptual schema: веб-сайт. URL: https://en.wikipedia.org/wiki/Conceptual_schema (дата звернення: 03.11.2021).
2. Аналіз ринку хлібобулочних виробів України 2020 рік: веб-сайт. URL: <https://pro-consulting.ua/ua/issledovanie-rynka/analiz-rynka-hlebobulochnyh-izdelij-2020> (дата звернення: 03.11.2021).
3. Брайковська А.М., Яковенко І.П. Роль електронної комерції в забезпеченні конкурентоспроможності підприємств на ринку хліба та хлібобулочних виробів. *Економіка і управління*. 2015. № 33. С. 219–227.
4. В Україні зменшилося виробництво хліба: веб-сайт. URL: <https://agravery.com/uk/posts/show/v-ukraini-majze-na-11-zmensilosa-virobnictvo-hliba> (дата звернення: 03.11.2021).
5. Випікання хліба в Україні: веб-сайт. URL: https://espreso.tv/news/2019/04/24/v_ukrayini_staly_vupikaty_menshe_khliba_ta_khlibobulochnukh_vyrobiv (дата звернення: 03.11.2021).
6. Виробництво борошна в Україні: веб-сайт. URL: <http://www.ukrmillers.com/muka/virobnitstvo-boroshna-v-ukrajini-za-pidsumkami-2019-roku-skorocheno-na-3> (дата звернення: 03.11.2021).
7. Виробництво хліба в Україні. Держстат: веб-сайт. URL: <https://agropolit.com/news/9262-v-ukrayini-na-10-skorotilosya-virobnitstvo-hliba--derjstat> (дата звернення: 03.11.2021).
8. Виробництво хлібобулочних виробів: веб-сайт. URL: <https://ukranews.com/ua/news/541729-u-2018-roci-v-ukraini-zrostut-ciny-na-khlib-i-znyzysya-yogo-vyrobnyctvo> (дата звернення: 03.11.2021).

9. Галактика. Харчова промисловість. Рішення на базі Галактика ERP: веб-сайт. URL: <https://topsoft.by/products/erp/industry-solutions/pishhevaya-promyshlennost/> (дата звернення: 03.11.2021).
10. Держстат: Оцінка чисельності наявного населення України: веб-сайт. URL: <https://www.kmu.gov.ua/> (дата звернення: 03.11.2021).
11. Єрмков О. Ю. Напрями та ефективність підприємств хлібопродуктового комплексу. *Економіка АПК*. 2006. № 7. С. 86–90.
12. Колотуша М.М. Методологічні засади аналізу фінансового стану хлібопекарської підгалузі харчової промисловості України. *Економіст*. 2007. № 5. С. 15–27.
13. Макаренко В. В. Вся правді про хліб. *Агро Перспектива*. 2007. № 6, 7. С. 24–27.
14. Офіційний сайт Державної служби статистики України: веб-сайт. URL: <http://ukrstat.gov.ua/> (дата звернення: 03.11.2021).
15. Підсумки роботи підприємств хлібопекарської галузі, виконавчого апарату: веб-сайт. URL: http://ukrplibprom.org.ua/ua/novini/28_sichnja_2014_roku.shtml (дата звернення: 03.11.2021).
16. Проектування та розробка інформаційної системи обліку контингенту дітей дошкільного та шкільного віку / Жукова В.М. *Вісник СНУ імені Володимира Даля*. 2014. № 5. С. 212.
17. Розвиток Галактика ERP: веб-сайт. URL: https://www.tadviser.ru/index.php/Продукт?cache=no&rtype=proj_onbase (дата звернення: 03.11.2021).
18. Статистичний збірник «Природний рух населення України» за 2019 рік: веб-сайт. URL: http://database.ukrcensus.gov.ua/PXWEB2007/ukr/publ_new1/index.asp (дата звернення: 03.11.2021).
19. Харчова промисловість. Галактика: веб-сайт. URL: <https://galaktika.ua/blog/xarchova-promislovist.html?lang=uk> (дата звернення: 03.11.2021).

20. Хлібопекарське виробництво. Терміни та визначення: веб-сайт. URL: <http://normativ.com.ua/sanpin/book29.php> (дата звернення: 03.11.2021).

ДОДАТКИ

Додаток А

SQL код бази даних інформаційної системи підтримки діяльності хлібозаводу

```

SET FOREIGN_KEY_CHECKS=0;
-----
-- Table structure for `spivrobotniku`
-----
DROP TABLE IF EXISTS `spivrobotniku`;
CREATE TABLE `spivrobotniku` (
  `Id_spivrobotnika` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `PIB` varchar(255) DEFAULT NULL,
  `Data_praz` date DEFAULT NULL,
  PRIMARY KEY (`Id_spivrobotnika`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

-----
-- Records of spivrobotniku
-----
INSERT INTO `spivrobotniku` VALUES ('1', 'Шевченко Ігор Дмитрович', '2010-09-05');
INSERT INTO `spivrobotniku` VALUES ('2', 'Павлов Роман Ігорович', '2015-06-10');
INSERT INTO `spivrobotniku` VALUES ('3', 'Цимбал Віктор Петрович', '2018-04-25');
INSERT INTO `spivrobotniku` VALUES ('4', 'Корнійчук Тарас Дмитрович', '2020-02-03');

-----
-- Table structure for `spusok_zakypok`
-----
DROP TABLE IF EXISTS `spusok_zakypok`;
CREATE TABLE `spusok_zakypok` (
  `Id_spusky` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_zakypok` int(11) unsigned NOT NULL,
  `Id_surovunu` int(11) unsigned NOT NULL,
  `Kilkist` int(11) DEFAULT NULL,
  `Zina` int(3) DEFAULT NULL,

```

```

PRIMARY KEY (`Id_spusky`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

```

```

-----
-- Records of spusok_zakypok
-----

```

```

INSERT INTO `spusok_zakypok` VALUES ('1', '1', '1', '800', '16');
INSERT INTO `spusok_zakypok` VALUES ('2', '1', '2', '900', '12');
INSERT INTO `spusok_zakypok` VALUES ('3', '1', '3', '50', '1');

```

```

-----
-- Table structure for `surovuna`
-----

```

```

DROP TABLE IF EXISTS `surovuna`;
CREATE TABLE `surovuna` (
  `Id_surovunu` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Nazva_surovunu` varchar(255) DEFAULT NULL,
  `Termin_prud` int(3) DEFAULT NULL,
  `Id_tupy` int(11) unsigned DEFAULT NULL,
  PRIMARY KEY (`Id_surovunu`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;

```

```

-----
-- Records of surovuna
-----

```

```

INSERT INTO `surovuna` VALUES ('1', 'Пшеничне борошно', '12', '1');
INSERT INTO `surovuna` VALUES ('2', 'Вівсяне борошно', '12', '1');
INSERT INTO `surovuna` VALUES ('3', 'Дріжді', '3', '2');
INSERT INTO `surovuna` VALUES ('4', 'Рисове борошно', '12', '1');
INSERT INTO `surovuna` VALUES ('5', 'Кукурудзяне борошно', '9', '1');
INSERT INTO `surovuna` VALUES ('6', 'Житнє борошно', '8', '1');
INSERT INTO `surovuna` VALUES ('7', 'Ляне борошно', '10', '1');
INSERT INTO `surovuna` VALUES ('8', 'Сіль йодована', '10', '3');

```

```

-----
-- Table structure for `surovuna_vuroby`
-----

```

```

DROP TABLE IF EXISTS `surovuna_vuroby`;
CREATE TABLE `surovuna_vuroby` (
  `Id_surovunu_vuroby` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_surovunu` int(11) unsigned NOT NULL,
  `Id_vuroby` int(11) unsigned NOT NULL,
  `Kilkist` int(11) DEFAULT NULL,
  PRIMARY KEY (`Id_surovunu_vuroby`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

```

```

-----
-- Records of surovuna_vuroby
-----

```

```

INSERT INTO `surovuna_vuroby` VALUES ('1', '1', '1', '500');
INSERT INTO `surovuna_vuroby` VALUES ('2', '3', '1', '10');
INSERT INTO `surovuna_vuroby` VALUES ('3', '2', '2', '200');
INSERT INTO `surovuna_vuroby` VALUES ('4', '3', '2', '5');

```

```

-----
-- Table structure for `tup_surovunu`
-----
DROP TABLE IF EXISTS `tup_surovunu`;
CREATE TABLE `tup_surovunu` (
  `Id_tupy` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Nazva_tupy_surovunu` varchar(255) DEFAULT NULL,
  `Zina(od)` int(3) DEFAULT NULL,
  PRIMARY KEY (`Id_tupy`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

-----
-- Records of tup_surovunu
-----
INSERT INTO `tup_surovunu` VALUES ('1', 'Борошно', '18');
INSERT INTO `tup_surovunu` VALUES ('2', 'Дріжді', '5');
INSERT INTO `tup_surovunu` VALUES ('3', 'Сіль', '10');

-----
-- Table structure for `vurobnuztvo`
-----
DROP TABLE IF EXISTS `vurobnuztvo`;
CREATE TABLE `vurobnuztvo` (
  `Id_vurobnuztva` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_vuroby` int(11) unsigned DEFAULT NULL,
  `Data` date DEFAULT NULL,
  `Kilkist` int(3) DEFAULT NULL,
  `Id_spivrobotnika` int(11) unsigned DEFAULT NULL,
  PRIMARY KEY (`Id_vurobnuztva`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;

-----
-- Records of vurobnuztvo
-----
INSERT INTO `vurobnuztvo` VALUES ('1', '1', '2021-03-01', '100', '1');
INSERT INTO `vurobnuztvo` VALUES ('2', '1', '2021-03-03', '80', '1');

-----
-- Table structure for `vurobu`
-----
DROP TABLE IF EXISTS `vurobu`;
CREATE TABLE `vurobu` (
  `Id_vuroby` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Nazva_vuroby` varchar(50) DEFAULT NULL,
  `Termin` int(3) DEFAULT NULL,
  PRIMARY KEY (`Id_vuroby`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;

-----
-- Records of vurobu
-----
INSERT INTO `vurobu` VALUES ('1', 'Хлібні хрусти', '30');

```

```
INSERT INTO `vurobu` VALUES ('2', 'Хлібні палички', '24');
INSERT INTO `vurobu` VALUES ('3', 'Соломка', '30');
INSERT INTO `vurobu` VALUES ('4', 'Пирози', '2');
INSERT INTO `vurobu` VALUES ('5', 'Пирожки', '2');
INSERT INTO `vurobu` VALUES ('6', 'Батон', '2');

-----
-- Table structure for `zakypku`
-----
DROP TABLE IF EXISTS `zakypku`;
CREATE TABLE `zakypku` (
  `Id_zakypok` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_spivrobotnika` int(11) unsigned DEFAULT NULL,
  `Data_poch` date DEFAULT NULL,
  `Data_kin` date DEFAULT NULL,
  PRIMARY KEY (`Id_zakypok`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

-----
-- Records of zakypku
-----
INSERT INTO `zakypku` VALUES ('1', '1', '2021-12-01', '2021-12-02');
INSERT INTO `zakypku` VALUES ('2', '1', '2021-09-01', '2021-09-02');
INSERT INTO `zakypku` VALUES ('3', '2', '2021-04-01', '2021-04-03');
```

**Основний програмний код на мові С# інформаційної системи
підтримки діяльності хлібозаводу**

```
using Microsoft.CSharp.RuntimeBinder;
using Microsoft.Office.Interop.Excel;
using MySql.Data.MySqlClient;
using System;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Globalization;
using System.IO;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using System.Windows.Forms;
using Vurobnuztvo.Properties;

namespace Vurobnuztvo
{
    public class Form1 : Form
    {
        public static MySqlConnection con = new MySqlConnection();
        private MySqlCommand cmd;
        private IContainer components = (IContainer) null;
        private MenuStrip menuStrip1;
        private ToolStripMenuItem файлToolStripMenuItem;
        private ToolStripMenuItem вхідВСистемуToolStripMenuItem;
        private ToolStripMenuItem журналToolStripMenuItem;
        private ToolStripMenuItem виробуToolStripMenuItem;
        private ToolStripMenuItem тунСировиниToolStripMenuItem;
        private ToolStripMenuItem співробітникиToolStripMenuItem;
        private ToolStripMenuItem сировинаToolStripMenuItem;
        private ToolStripMenuItem закупкиToolStripMenuItem;
        private ToolStripMenuItem звітуToolStripMenuItem;
        private ToolStripMenuItem звітПоЗакупкахToolStripMenuItem;
        private ToolStripMenuItem звітРеалізаціїТоварівToolStripMenuItem;
        private ToolStripMenuItem довідкаToolStripMenuItem;
        private ToolStripMenuItem проСистемуToolStripMenuItem;
        private ToolStripMenuItem вихідToolStripMenuItem;
        private ToolStripMenuItem облікToolStripMenuItem;
    }
}
```

```

private ToolStripMenuItem виробництвоToolStripMenuItem;
private ToolStripMenuItem закупкиToolStripMenuItem;
private ToolStripMenuItem реалізаціяВиробівToolStripMenuItem;

public Form1() => this.InitializeComponent();

private void бухідToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Ви впевнені, що хочете завершити роботу?", "Бухід",
        MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk) != DialogResult.Yes)
        return;
    this.Close();
}

private void звимПоЗакупкахToolStripMenuItem_Click(object sender, EventArgs e)
{
    string path2 = "zalushok.xlsx";
    string path1_1 = Environment.CurrentDirectory + "\\rep_copy";
    string path1_2 = Environment.CurrentDirectory + "\\rep";
    File.Copy(Path.Combine(path1_1, path2), Path.Combine(path1_2, path2), true);
    // ISSUE: variable of a compiler-generated type
    Microsoft.Office.Interop.Excel.Application instance =
    (Microsoft.Office.Interop.Excel.Application)
    Activator.CreateInstance(Marshal.GetTypeFromCLSID(new Guid("00024500-0000-0000-C000-
    000000000046")));
    instance.Visible = false;
    string Filename = Environment.CurrentDirectory + "\\rep\\zalushok.xlsx";
    // ISSUE: reference to a compiler-generated method
    // ISSUE: variable of a compiler-generated type
    Workbook workbook = instance.Workbooks.Open(Filename, System.Type.Missing,
    System.Type.Missing, System.Type.Missing, System.Type.Missing,
    System.Type.Missing, System.Type.Missing, System.Type.Missing,
    System.Type.Missing, System.Type.Missing, System.Type.Missing,
    System.Type.Missing);
    // ISSUE: reference to a compiler-generated field
    if (Form1.\u003C\u003Eo__11.\u003C\u003Eep__0 == null)
    {
        // ISSUE: reference to a compiler-generated field
        Form1.\u003C\u003Eo__11.\u003C\u003Eep__0 = CallSite<Func<CallSite, object,
        Worksheet>>.Create(Binder.Convert(CSharpBinderFlags.ConvertExplicit, typeof(Worksheet),
        typeof(Form1)));
    }
    // ISSUE: reference to a compiler-generated field
    // ISSUE: reference to a compiler-generated field
    // ISSUE: variable of a compiler-generated type
    Worksheet worksheet = Form1.\u003C\u003Eo__11.\u003C\u003Eep__0.Target((CallSite)
    Form1.\u003C\u003Eo__11.\u003C\u003Eep__0, workbook.Worksheets[(object) 1]);
    string selectCommandText = "SELECT\r\n        surovuna.Nazva_surovunu,\r\n
    tup_surovunu.Nazva_tupy_surovunu,\r\n        (spusok_zakypok.Kilkist - surovuna_vuroby.Kilkist)
    AS zalushok\r\n        FROM\r\n        surovuna\r\n        Inner Join tup_surovunu ON
    tup_surovunu.Id_tupy = surovuna.Id_tupy\r\n        Inner Join surovuna_vuroby ON
    surovuna_vuroby.Id_surovunu = surovuna.Id_surovunu\r\n        Inner Join spusok_zakypok ON
    spusok_zakypok.Id_surovunu = surovuna_vuroby.Id_surovunu\r\n        GROUP BY\r\n

```

```

tup_surovunu.Nazva_tupy_surovunu,\r\n      surovuna.Nazva_surovunu\r\n      ORDER
BY\r\n      tup_surovunu.Nazva_tupy_surovunu ASC";
    int num = 3;
    using (MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(selectCommandText,
Form1.con))
    {
        DataTable dataTable = new DataTable();
        dataTable.Locale = CultureInfo.InvariantCulture;
        mySqlDataAdapter.Fill(dataTable);
        foreach (DataRow row in (InternalDataCollectionBase) dataTable.Rows)
        {
            worksheet.Cells[(object) num, (object) 2] = (object) row["Nazva_surovunu"].ToString();
            worksheet.Cells[(object) num, (object) 3] = (object)
row["Nazva_tupy_surovunu"].ToString();
            worksheet.Cells[(object) num, (object) 4] = (object) row["zalushok"].ToString();
            ++num;
        }
    }
    instance.Visible = true;
}

```

```

private void звимРеализацijiТоваривToolStripMenuItem_Click(object sender, EventArgs e)
{
    string path2 = "vutratu_surovunu.xlsx";
    string path1_1 = Environment.CurrentDirectory + "\\rep_copy";
    string path1_2 = Environment.CurrentDirectory + "\\rep";
    File.Copy(Path.Combine(path1_1, path2), Path.Combine(path1_2, path2), true);
    // ISSUE: variable of a compiler-generated type
    Microsoft.Office.Interop.Excel.Application instance =
(Microsoft.Office.Interop.Excel.Application)
Activator.CreateInstance(Marshal.GetTypeFromCLSID(new Guid("00024500-0000-0000-C000-
000000000046")));
    instance.Visible = false;
    string Filename = Environment.CurrentDirectory + "\\rep\\vutratu_surovunu.xlsx";
    // ISSUE: reference to a compiler-generated method
    // ISSUE: variable of a compiler-generated type
    Workbook workbook = instance.Workbooks.Open(Filename, System.Type.Missing,
System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
System.Type.Missing);
    // ISSUE: reference to a compiler-generated field
    if (Form1.\u003C\u003Eo__12.\u003C\u003Eep__0 == null)
    {
        // ISSUE: reference to a compiler-generated field
        Form1.\u003C\u003Eo__12.\u003C\u003Eep__0 = CallSite<Func<CallSite, object,
Worksheet>>.Create(Binder.Convert(CSharpBinderFlags.ConvertExplicit, typeof(Worksheet),
typeof(Form1)));
    }
    // ISSUE: reference to a compiler-generated field
    // ISSUE: reference to a compiler-generated field
    // ISSUE: variable of a compiler-generated type

```

```

Worksheet worksheet = Form1.\u003C\u003Eo__12.\u003C\u003Ep__0.Target((CallSite)
Form1.\u003C\u003Eo__12.\u003C\u003Ep__0, workbook.Worksheets[(object) 1]);
string selectCommandText = "SELECT\r\n        surovuna.Nazva_surovunu,\r\n
tup_surovunu.Nazva_tupy_surovunu,\r\n        tup_surovunu.`Zina(od)`,\r\n
spusok_zakypok.Kilkist,\r\n        (tup_surovunu.`Zina(od)` * spusok_zakypok.Kilkist) AS
vutratu_surovuna\r\n        FROM\r\n        tup_surovunu\r\n        Inner Join spusok_zakypok
ON spusok_zakypok.Id_surovunu = tup_surovunu.Id_tupy\r\n        Inner Join surovuna ON
surovuna.Id_tupy = tup_surovunu.Id_tupy\r\n        GROUP BY\r\n
tup_surovunu.Nazva_tupy_surovunu\r\n        ORDER BY\r\n
tup_surovunu.Nazva_tupy_surovunu ASC";
int num = 3;
using (MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(selectCommandText,
Form1.con))
{
    DataTable dataTable = new DataTable();
    dataTable.Locale = CultureInfo.InvariantCulture;
    mySqlDataAdapter.Fill(dataTable);
    foreach (DataRow row in (InternalDataCollectionBase) dataTable.Rows)
    {
        worksheet.Cells[(object) num, (object) 2] = (object) row["Nazva_surovunu"].ToString();
        worksheet.Cells[(object) num, (object) 3] = (object)
row["Nazva_tupy_surovunu"].ToString();
        worksheet.Cells[(object) num, (object) 4] = (object) row["Zina(od)"].ToString();
        worksheet.Cells[(object) num, (object) 5] = (object) row["Kilkist"].ToString();
        worksheet.Cells[(object) num, (object) 6] = (object) row["vutratu_surovuna"].ToString();
        ++num;
    }
}
instance.Visible = true;
}

private void облікToolStripMenuItem_Click(object sender, EventArgs e)
{
    int num = (int) new Form9().ShowDialog();
}

private void виробництвоToolStripMenuItem_Click(object sender, EventArgs e)
{
    string path2 = "vurobnuztvo.xlsx";
    string path1_1 = Environment.CurrentDirectory + "\\rep_copy";
    string path1_2 = Environment.CurrentDirectory + "\\rep";
    File.Copy(Path.Combine(path1_1, path2), Path.Combine(path1_2, path2), true);
    // ISSUE: variable of a compiler-generated type
    Microsoft.Office.Interop.Excel.Application instance =
(Microsoft.Office.Interop.Excel.Application)
Activator.CreateInstance(Marshal.GetTypeFromCLSID(new Guid("00024500-0000-0000-C000-
000000000046")));
    instance.Visible = false;
    string Filename = Environment.CurrentDirectory + "\\rep\\vurobnuztvo.xlsx";
    // ISSUE: reference to a compiler-generated method
    // ISSUE: variable of a compiler-generated type
    Workbook workbook = instance.Workbooks.Open(Filename, System.Type.Missing,
System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,

```

```

System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
System.Type.Missing);
    // ISSUE: reference to a compiler-generated field
    if (Form1.\u003C\u003Eo__14.\u003C\u003Ep__0 == null)
    {
        // ISSUE: reference to a compiler-generated field
        Form1.\u003C\u003Eo__14.\u003C\u003Ep__0 = CallSite<Func<CallSite, object,
Worksheet>>.Create(Binder.Convert(CSharpBinderFlags.ConvertExplicit, typeof(Worksheet),
typeof(Form1)));
    }
    // ISSUE: reference to a compiler-generated field
    // ISSUE: reference to a compiler-generated field
    // ISSUE: variable of a compiler-generated type
    Worksheet worksheet = Form1.\u003C\u003Eo__14.\u003C\u003Ep__0.Target((CallSite)
Form1.\u003C\u003Eo__14.\u003C\u003Ep__0, workbook.Worksheets[(object) 1]);
    string selectCommandText = "SELECT\r\n        vurobu.Nazva_vuroby,\r\n
vurobnuztvo.Data,\r\n        vurobnuztvo.Kilkist\r\n        FROM\r\n        vurobnuztvo\r\n
Inner Join vurobu ON vurobu.Id_vuroby = vurobnuztvo.Id_vuroby\r\n        ORDER BY\r\n
vurobu.Nazva_vuroby ASC";
    int num = 3;
    using (MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(selectCommandText,
Form1.con))
    {
        DataTable dataTable = new DataTable();
        dataTable.Locale = CultureInfo.InvariantCulture;
        mySqlDataAdapter.Fill(dataTable);
        foreach (DataRow row in (InternalDataCollectionBase) dataTable.Rows)
        {
            worksheet.Cells[(object) num, (object) 2] = (object) row["Nazva_vuroby"].ToString();
            worksheet.Cells[(object) num, (object) 3] = (object) row["Data"].ToString();
            worksheet.Cells[(object) num, (object) 4] = (object) row["Kilkist"].ToString();
            ++num;
        }
    }
    instance.Visible = true;
}

private void zakynkuToolStripMenuItem1_Click(object sender, EventArgs e)
{
    string path2 = "zakypka.xlsx";
    string path1_1 = Environment.CurrentDirectory + "\\rep_copy";
    string path1_2 = Environment.CurrentDirectory + "\\rep";
    File.Copy(Path.Combine(path1_1, path2), Path.Combine(path1_2, path2), true);
    // ISSUE: variable of a compiler-generated type
    Microsoft.Office.Interop.Excel.Application instance =
(Microsoft.Office.Interop.Excel.Application)
Activator.CreateInstance(Marshal.GetTypeFromCLSID(new Guid("00024500-0000-0000-C000-
000000000046")));
    instance.Visible = false;
    string Filename = Environment.CurrentDirectory + "\\rep\\zakypka.xlsx";
    // ISSUE: reference to a compiler-generated method
    // ISSUE: variable of a compiler-generated type

```

```

Workbook workbook = instance.Workbooks.Open(Filename, System.Type.Missing,
System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
System.Type.Missing);
// ISSUE: reference to a compiler-generated field
if (Form1.\u003C\u003Eo__15.\u003C\u003Ep__0 == null)
{
// ISSUE: reference to a compiler-generated field
Form1.\u003C\u003Eo__15.\u003C\u003Ep__0 = CallSite<Func<CallSite, object,
Worksheet>>.Create(Binder.Convert(CSharpBinderFlags.ConvertExplicit, typeof (Worksheet),
typeof (Form1)));
}
// ISSUE: reference to a compiler-generated field
// ISSUE: reference to a compiler-generated field
// ISSUE: variable of a compiler-generated type
Worksheet worksheet = Form1.\u003C\u003Eo__15.\u003C\u003Ep__0.Target((CallSite)
Form1.\u003C\u003Eo__15.\u003C\u003Ep__0, workbook.Worksheets[(object) 1]);
string selectCommandText = "SELECT\r\n      spivrobotniku.PIB,\r\n
surovuna.Nazva_surovunu,\r\n      spusok_zakypok.Kilkist,\r\n      spusok_zakypok.Zina,\r\n
zakypku.Data_poch,\r\n      zakypku.Data_kin\r\n      FROM\r\n      zakypku\r\n
Inner Join spivrobotniku ON spivrobotniku.Id_spivrobotnika = zakypku.Id_spivrobotnika\r\n
Inner Join spusok_zakypok ON zakypku.Id_zakypok = spusok_zakypok.Id_zakypok\r\n      Inner
Join surovuna ON surovuna.Id_surovunu = spusok_zakypok.Id_surovunu\r\n      ORDER BY\r\n
spivrobotniku.PIB ASC";
int num = 3;
using (MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(selectCommandText,
Form1.con))
{
DataTable dataTable = new DataTable();
dataTable.Locale = CultureInfo.InvariantCulture;
mySqlDataAdapter.Fill(dataTable);
foreach (DataRow row in (InternalDataCollectionBase) dataTable.Rows)
{
worksheet.Cells[(object) num, (object) 2] = (object) row["PIB"].ToString();
worksheet.Cells[(object) num, (object) 3] = (object) row["Nazva_surovunu"].ToString();
worksheet.Cells[(object) num, (object) 4] = (object) row["Kilkist"].ToString();
worksheet.Cells[(object) num, (object) 5] = (object) row["Zina"].ToString();
worksheet.Cells[(object) num, (object) 6] = (object) row["Data_poch"].ToString();
worksheet.Cells[(object) num, (object) 7] = (object) row["Data_kin"].ToString();
++num;
}
}
instance.Visible = true;
}

private void реалізаціяВиробівToolStripMenuItem_Click(object sender, EventArgs e)
{
string path2 = "realiz.xlsx";
string path1_1 = Environment.CurrentDirectory + "\\rep_copy";
string path1_2 = Environment.CurrentDirectory + "\\rep";
File.Copy(Path.Combine(path1_1, path2), Path.Combine(path1_2, path2), true);
// ISSUE: variable of a compiler-generated type

```

```

    Microsoft.Office.Interop.Excel.Application instance =
    (Microsoft.Office.Interop.Excel.Application)
    Activator.CreateInstance(Marshal.GetTypeFromCLSID(new Guid("00024500-0000-0000-C000-
    000000000046")));
    instance.Visible = false;
    string Filename = Environment.CurrentDirectory + "\\rep\\realiz.xlsx";
    // ISSUE: reference to a compiler-generated method
    // ISSUE: variable of a compiler-generated type
    Workbook workbook = instance.Workbooks.Open(Filename, System.Type.Missing,
    System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
    System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
    System.Type.Missing, System.Type.Missing, System.Type.Missing, System.Type.Missing,
    System.Type.Missing);
    // ISSUE: reference to a compiler-generated field
    if (Form1.\u003C\u003Eo__16.\u003C\u003Eep__0 == null)
    {
        // ISSUE: reference to a compiler-generated field
        Form1.\u003C\u003Eo__16.\u003C\u003Eep__0 = CallSite<Func<CallSite, object,
    Worksheet>>.Create(Binder.Convert(CSharpBinderFlags.ConvertExplicit, typeof(Worksheet),
    typeof(Form1)));
    }
    // ISSUE: reference to a compiler-generated field
    // ISSUE: reference to a compiler-generated field
    // ISSUE: variable of a compiler-generated type
    Worksheet worksheet = Form1.\u003C\u003Eo__16.\u003C\u003Eep__0.Target((CallSite)
    Form1.\u003C\u003Eo__16.\u003C\u003Eep__0, workbook.Worksheets[(object) 1]);
    string selectCommandText = "SELECT\r\n        vurobu.Nazva_vuroby,\r\n
    vurobnuztvo.Kilkist,\r\n        spusok_zakypok.Zina,\r\n        vurobnuztvo.Data\r\n
    FROM\r\n        vurobu ,\r\n        spusok_zakypok ,\r\n        vurobnuztvo\r\n
    WHERE\r\n        vurobnuztvo.Id_vuroby = vurobu.Id_vuroby AND\r\n
    spusok_zakypok.Id_zakypok = vurobnuztvo.Id_vurobnuztva\r\n        ORDER BY\r\n
    vurobu.Nazva_vuroby ASC";
    int num = 3;
    using (MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(selectCommandText,
    Form1.con))
    {
        DataTable dataTable = new DataTable();
        dataTable.Locale = CultureInfo.InvariantCulture;
        mySqlDataAdapter.Fill(dataTable);
        foreach (DataRow row in (InternalDataCollectionBase) dataTable.Rows)
        {
            worksheet.Cells[(object) num, (object) 2] = (object) row["Nazva_vuroby"].ToString();
            worksheet.Cells[(object) num, (object) 3] = (object) row["Kilkist"].ToString();
            worksheet.Cells[(object) num, (object) 4] = (object) row["Zina"].ToString();
            worksheet.Cells[(object) num, (object) 5] = (object) row["Data"].ToString();
            ++num;
        }
    }
    instance.Visible = true;
}

protected override void Dispose(bool disposing)
{

```

```

    if (disposing && this.components != null)
        this.components.Dispose();
    base.Dispose(disposing);
}

```

```

using Microsoft.CSharp.RuntimeBinder;
using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Interop.Word;
using MySql.Data.MySqlClient;
using System;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Globalization;
using System.IO;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using System.Windows.Forms;

```

```

namespace Vurobnuztvo
{
    public class Form6 : Form
    {
        private MySqlCommand cmd;
        private string query;
        private bool insert_mode;
        private IContainer components = (IContainer) null;
        private DateTimePicker dateTimePicker1;
        private DateTimePicker dateTimePicker2;
        private DateTimePicker dateTimePicker3;
        private DateTimePicker dateTimePicker4;
        private Label label1;
        private Label label2;
        private Button button1;
        private Label label3;
        private DataGridView dataGridView1;
        private ContextMenuStrip contextMenuStrip1;
        private ToolStripMenuItem pedazyvamuToolStripMenuItem;
        private ToolStripMenuItem vudalumuToolStripMenuItem;
        private DataGridView dataGridView2;
        private ContextMenuStrip contextMenuStrip2;
        private ToolStripMenuItem pedazyvamuToolStripMenuItem1;
        private ToolStripMenuItem vudalumuToolStripMenuItem1;
        private Label label4;
        private Label label5;
        private TextBox textBox1;
        private Button button2;
        private Button button3;
        private Button button4;
        private Button button5;
        private ComboBox comboBox1;
        private ComboBox comboBox2;
    }
}

```

```

private Label label6;
private Label label7;
private Label label8;
private TextBox textBox2;
private TextBox textBox3;
private Label label9;
private Button button7;
private Button button6;

public Form6() => this.InitializeComponent();

private void Form6_Shown(object sender, EventArgs e)
{
    this.insert_mode = true;
    DateTime today = DateTime.Today;
    this.dateTimePicker1.Value = today;
    this.dateTimePicker2.Value = today.AddDays(7.0);
    this.dateTimePicker3.Value = today;
    this.dateTimePicker4.Value = today;
    using (MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter("SELECT\r\n
spivrobotnuku.Id_spivrobotnuka,\r\n      spivrobotnuku.PIB\r\n      FROM\r\n
spivrobotnuku\r\n      ORDER BY\r\n      spivrobotnuku.PIB ASC", Form1.con))
    {
        DataTable dataTable = new DataTable();
        dataTable.Locale = CultureInfo.InvariantCulture;
        mySqlDataAdapter.Fill(dataTable);
        this.comboBox1.DataSource = (object) dataTable;
        this.comboBox1.DisplayMember = dataTable.Columns["PIB"].ColumnName;
        this.comboBox1.ValueMember = dataTable.Columns["Id_spivrobotnuka"].ColumnName;
        this.comboBox1.Text = "";
    }
    using (MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter("SELECT\r\n
surovuna.Id_surovunu,\r\n      surovuna.Nazva_surovunu\r\n      FROM\r\n
surovuna\r\n      GROUP BY\r\n      surovuna.Nazva_surovunu", Form1.con))
    {
        DataTable dataTable = new DataTable();
        dataTable.Locale = CultureInfo.InvariantCulture;
        mySqlDataAdapter.Fill(dataTable);
        this.comboBox2.DataSource = (object) dataTable;
        this.comboBox2.DisplayMember = dataTable.Columns["Nazva_surovunu"].ColumnName;
        this.comboBox2.ValueMember = dataTable.Columns["Id_surovunu"].ColumnName;
        this.comboBox2.Text = "";
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (this.dateTimePicker2.Value < this.dateTimePicker1.Value)
    {
        int num = (int) MessageBox.Show("Невірно вказані дати", "Увага",
        MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
    }
    else

```

```

{
    DateTime dateTime1 = this.dateTimePicker1.Value;
    DateTime dateTime2 = this.dateTimePicker2.Value;
    this.query = "SELECT\r\n        zakypku.Id_zakypok,\r\n        spivrobotnuku.PIB,\r\n
zakypku.Data_poch,\r\n        zakypku.Data_kin,\r\n        (zakypku.Data_kin -
zakypku.Data_poch) AS dni_zakupok,\r\n        zakypku.Id_spivrobotnuka\r\n        FROM\r\n
zakypku\r\n        Inner Join spivrobotnuku ON zakypku.Id_spivrobotnuka =
spivrobotnuku.Id_spivrobotnuka\r\n        WHERE\r\n        zakypku.Data_poch >= '" +
dateTime1.ToString("yyyy-MM-dd") + "' AND\r\n        zakypku.Data_kin <= '" +
dateTime2.ToString("yyyy-MM-dd") + "'";
    MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(this.query, Form1.con);
    DataSet dataSet = new DataSet();
    mySqlDataAdapter.Fill(dataSet, "zakypku");
    this.dataGridView1.DataSource = (object) dataSet;
    this.dataGridView1.DataMember = "zakypku";
    this.dataGridView1.Columns[0].HeaderText = "Код закупки";
    this.dataGridView1.Columns[1].HeaderText = "ПІБ";
    this.dataGridView1.Columns[2].HeaderText = "Початок";
    this.dataGridView1.Columns[3].HeaderText = "Закінчення";
    this.dataGridView1.Columns[4].HeaderText = "Кількість днів";
    this.dataGridView1.Columns[5].Visible = false;
    this.dataGridView1.Refresh();
}
}

private void button2_Click(object sender, EventArgs e)
{
    string str1;
    if (this.insert_mode)
    {
        string str2 = "INSERT INTO `zakypku` (`Id_spivrobotnuka`, `Data_poch`, `Data_kin`)
VALUES (" + "" + Convert.ToString(this.comboBox1.SelectedValue) + ", ";
        DateTime dateTime = this.dateTimePicker3.Value;
        string str3 = dateTime.ToString("yyyy-MM-dd");
        string str4 = str2 + "" + str3 + ", ";
        dateTime = this.dateTimePicker4.Value;
        string str5 = dateTime.ToString("yyyy-MM-dd");
        str1 = str4 + "" + str5 + ")";
    }
    else
    {
        string str6 = "UPDATE `zakypku` SET" + "`Id_spivrobotnuka`=" +
Convert.ToString(this.comboBox1.SelectedValue) + ", ";
        DateTime dateTime = this.dateTimePicker3.Value;
        string str7 = dateTime.ToString("yyyy-MM-dd");
        string str8 = str6 + "`Data_poch`=" + str7 + ", ";
        dateTime = this.dateTimePicker4.Value;
        string str9 = dateTime.ToString("yyyy-MM-dd");
        str1 = str8 + "`Data_kin`=" + str9 + " " + "WHERE `Id_zakypok`=" +
Convert.ToString(this.dataGridView1.CurrentRow.Cells[0].Value);
    }
    this.cmd = new MySqlCommand();
    this.cmd.Connection = Form1.con;
}

```

```

this.cmd.CommandText = str1;
this.cmd.CommandType = CommandType.Text;
this.cmd.ExecuteNonQuery();
MySQLDataAdapter mySqlDataAdapter = new MySQLDataAdapter(this.query, Form1.con);
DataSet dataSet = new DataSet();
mySqlDataAdapter.Fill(dataSet, "zakypku");
this.dataGridView1.DataSource = (object) dataSet;
this.dataGridView1.Refresh();
try
{
    for (int index = 0; index < this.dataGridView1.RowCount; ++index)
    {
        this.dataGridView1.Rows[index].Selected = false;
        if
(this.dataGridView1.Rows[index].Cells[1].Value.ToString().Contains(this.comboBox1.Text))
        {
            this.dataGridView1.Rows[index].Selected = true;
            break;
        }
    }
}
catch
{
    this.dataGridView1.Rows[0].Selected = true;
}
this.comboBox1.Enabled = true;
this.dataGridView1.Enabled = true;
this.dataGridView2.Enabled = true;
this.textBox2.Text = "";
this.comboBox1.Text = "";
this.insert_mode = true;
}

private void button3_Click(object sender, EventArgs e)
{
    this.comboBox1.Enabled = true;
    this.dataGridView1.Enabled = true;
    this.dataGridView2.Enabled = true;
    this.comboBox1.Text = "";
    this.insert_mode = true;
}

private void pedazySamuToolStripMenuItem_Click(object sender, EventArgs e)
{
    if ((uint) this.dataGridView1.RowCount <= 0U)
        return;
    this.comboBox1.SelectedValue = this.dataGridView1.CurrentRow.Cells[5].Value;
    this.comboBox1.Text = Convert.ToString(this.dataGridView1.CurrentRow.Cells[1].Value);
    this.dateTimePicker3.Value =
Convert.ToDateTime(this.dataGridView1.CurrentRow.Cells[2].Value);
    this.dateTimePicker4.Value =
Convert.ToDateTime(this.dataGridView1.CurrentRow.Cells[3].Value);
    this.dataGridView1.Enabled = false;
}

```

```

    this.dataGridView2.Enabled = false;
    this.insert_mode = false;
}

private void dataGridView1_DoubleClick(object sender, EventArgs e)
{
    if ((uint) this.dataGridView1.RowCount <= 0U)
        return;
    this.comboBox1.SelectedValue = this.dataGridView1.CurrentRow.Cells[5].Value;
    this.comboBox1.Text = Convert.ToString(this.dataGridView1.CurrentRow.Cells[1].Value);
    this.dateTimePicker3.Value =
Convert.ToDateTime(this.dataGridView1.CurrentRow.Cells[2].Value);
    this.dateTimePicker4.Value =
Convert.ToDateTime(this.dataGridView1.CurrentRow.Cells[3].Value);
    this.dataGridView1.Enabled = false;
    this.dataGridView2.Enabled = false;
    this.insert_mode = false;
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Ви впевнені, що хочете видалити запис?", "Видалення",
MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk) != DialogResult.Yes)
        return;
    try
    {
        string str = "DELETE FROM `zakypku` WHERE `Id_zakypok` = " +
Convert.ToString(this.dataGridView1.CurrentRow.Cells[0].Value);
        this.cmd = new MySqlCommand();
        this.cmd.Connection = Form1.con;
        this.cmd.CommandText = str;
        this.cmd.CommandType = CommandType.Text;
        this.cmd.ExecuteNonQuery();
        MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(this.query, Form1.con);
        DataSet dataSet = new DataSet();
        mySqlDataAdapter.Fill(dataSet, "zakypku");
        this.dataGridView1.DataSource = (object) dataSet;
        this.dataGridView1.Refresh();
        this.dataGridView1.Enabled = true;
        this.dataGridView2.Enabled = true;
        this.comboBox1.Text = "";
        this.insert_mode = true;
    }
    catch
    {
        int num = (int) MessageBox.Show("Видалення неможливо. Закупки має пов'язані записи",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Hand);
    }
}

private void dataGridView1_Click(object sender, EventArgs e)
{

```

```

        MySqlConnection mySqlConnection = new MySqlConnection("Server=" +
        Convert.ToString(this.textBox1.Text) + ";Database=" +
        Convert.ToString(this.textBox2.Text) + ";User Id=" +
        Convert.ToString(this.textBox3.Text) + ";Password=" +
        Convert.ToString(this.textBox4.Text));
        MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter("SELECT\r\n
        spusok_zakypok.Id_spusky,\r\n          surovuna.Nazva_surovunu,\r\n
        spusok_zakypok.Kilkist,\r\n          spusok_zakypok.Zina,\r\n
        spusok_zakypok.Id_surovunu,\r\n          zakypku.Id_zakypok\r\n          FROM\r\n
        surovuna ,\r\n          spusok_zakypok ,\r\n          zakypku\r\n          WHERE\r\n
        surovuna.Id_surovunu = spusok_zakypok.Id_surovunu AND\r\n
        spusok_zakypok.Id_zakypok = " +
        Convert.ToString(this.dataGridView1.CurrentRow.Cells[0].Value) + " GROUP BY
        surovuna.Nazva_surovunu", Form1.con);
        DataSet dataSet = new DataSet();
        mySqlDataAdapter.Fill(dataSet, "spusok_zakypok");
        this.dataGridView2.DataSource = (object) dataSet;
        this.dataGridView2.DataMember = "spusok_zakypok";
        this.dataGridView2.Columns[0].HeaderText = "Код спуску";
        this.dataGridView2.Columns[1].HeaderText = "Сировина";
        this.dataGridView2.Columns[2].HeaderText = "Кількість";
        this.dataGridView2.Columns[3].HeaderText = "Ціна";
        this.dataGridView2.Columns[4].Visible = false;
        this.dataGridView2.Columns[5].Visible = false;
        this.dataGridView2.Columns[0].Width = 100;
        this.dataGridView2.Columns[1].Width = 200;
        this.dataGridView2.Columns[2].Width = 90;
        this.dataGridView2.Columns[3].Width = 90;
        this.dataGridView2.Refresh();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        string str = !this.insert_mode ? "UPDATE `spusok_zakypok` SET" + "`Id_surovunu`=" +
        Convert.ToString(this.comboBox2.SelectedValue) + "," + "`Kilkist`=" +
        Convert.ToString(this.textBox2.Text) + "," + "`Zina`=" + Convert.ToString(this.textBox3.Text) +
        "" + "WHERE `Id_spusky`=" +
        Convert.ToString(this.dataGridView2.CurrentRow.Cells[0].Value) : "INSERT INTO
        `spusok_zakypok` (`Id_surovunu`, `Kilkist`, `Zina`) VALUES (" + "" +
        Convert.ToString(this.comboBox2.SelectedValue) + "," + "" +
        Convert.ToString(this.textBox2.Text) + "," + "" + Convert.ToString(this.textBox3.Text) + ")";
        this.cmd = new MySqlCommand();
        this.cmd.Connection = Form1.con;
        this.cmd.CommandText = str;
        this.cmd.CommandType = CommandType.Text;
        this.cmd.ExecuteNonQuery();
        MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(this.query, Form1.con);
        DataSet dataSet = new DataSet();
        mySqlDataAdapter.Fill(dataSet, "spusok_zakypok");
        this.dataGridView1.DataSource = (object) dataSet;
        this.dataGridView2.Refresh();
        this.dataGridView1_Click((object) this, e);
        this.dataGridView1.Enabled = true;
        this.dataGridView2.Enabled = true;
        this.textBox2.Text = "";
        this.comboBox2.Text = "";
        this.insert_mode = true;
    }

```

```

private void button5_Click(object sender, EventArgs e)
{
    this.comboBox2.Enabled = true;
    this.dataGridView1.Enabled = true;
    this.dataGridView2.Enabled = true;
    this.textBox2.Text = "";
    this.insert_mode = true;
}

private void pedazyvamuToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if ((uint) this.dataGridView2.RowCount <= 0U)
        return;
    this.comboBox2.SelectedValue = this.dataGridView2.CurrentRow.Cells[4].Value;
    this.comboBox2.Text = Convert.ToString(this.dataGridView2.CurrentRow.Cells[1].Value);
    this.textBox1.Text = Convert.ToString(this.dataGridView2.CurrentRow.Cells[0].Value);
    this.textBox2.Text = Convert.ToString(this.dataGridView2.CurrentRow.Cells[2].Value);
    this.textBox3.Text = Convert.ToString(this.dataGridView2.CurrentRow.Cells[3].Value);
    this.dataGridView2.Enabled = false;
    this.dataGridView1.Enabled = false;
    this.comboBox1.Enabled = false;
    this.insert_mode = false;
}

private void видалитиToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Ви впевнені, що хочете видалити запис?", "Видалення",
        MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk) != DialogResult.Yes)
        return;
    try
    {
        string str = "DELETE FROM `spusok_zakypok` WHERE `Id_spusky` = " +
        Convert.ToString(this.dataGridView2.CurrentRow.Cells[0].Value);
        this.cmd = new MySqlCommand();
        this.cmd.Connection = Form1.con;
        this.cmd.CommandText = str;
        this.cmd.CommandType = CommandType.Text;
        this.cmd.ExecuteNonQuery();
        MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(this.query, Form1.con);
        DataSet dataSet = new DataSet();
        mySqlDataAdapter.Fill(dataSet, "spusok_zakypok");
        this.dataGridView2.DataSource = (object) dataSet;
        this.dataGridView2.Refresh();
        this.comboBox1.Enabled = true;
        this.dataGridView1.Enabled = true;
        this.dataGridView2.Enabled = true;
        this.textBox1.Text = "";
        this.textBox2.Text = "";
        this.comboBox1.Text = "";
        this.insert_mode = true;
    }
    catch

```

```

    {
        int num = (int) MessageBox.Show("Видалення неможливо. Список закупок має пов'язані записи", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Hand);
    }
}

private void button6_Click(object sender, EventArgs e)
{
    string path2_1 = "nakladna.docx";
    string path2_2 = "nakladna " +
Convert.ToString(this.dataGridView1.CurrentRow.Cells[1].Value) + ".docx";
    string path1_1 = Environment.CurrentDirectory + "\\rep_copy";
    string path1_2 = Environment.CurrentDirectory + "\\rep";
    File.Copy(Path.Combine(path1_1, path2_1), Path.Combine(path1_2, path2_2), true);
    string str = Environment.CurrentDirectory + "\\rep\\" + path2_2;
    if (Form6.\u003C\u003Eo__17.\u003C\u003Eр__0 == null)
    {
        for (int index1 = 0; index1 < this.dataGridView1.SelectedCells.Count; ++index1)
        {
            for (int index2 = 0; index2 < this.dataGridView1.ColumnCount; ++index2)
                worksheet.Cells[(object) (index1 + 2), (object) (index2 + 1)] = (object)
this.dataGridView1.Rows[index1].Cells[index2].Value.ToString();
        }
        for (int index3 = 0; index3 < this.dataGridView2.SelectedCells.Count; ++index3)
        {
            for (int index4 = 0; index4 < this.dataGridView2.ColumnCount; ++index4)
                worksheet.Cells[(object) (index3 + 2), (object) (index4 + 7)] = (object)
this.dataGridView2.Rows[index3].Cells[index4].Value.ToString();
        }
        instance.Visible = true;
    }
}

protected override void Dispose(bool disposing)
{
    if (disposing && this.components != null)
        this.components.Dispose();
    base.Dispose(disposing);
}

private void InitializeComponent()
{
    this.components = (IContainer) new Container();
    DataGridViewCellStyle gridViewCellStyle1 = new DataGridViewCellStyle();
    DataGridViewCellStyle gridViewCellStyle2 = new DataGridViewCellStyle();
    DataGridViewCellStyle gridViewCellStyle3 = new DataGridViewCellStyle();
    DataGridViewCellStyle gridViewCellStyle4 = new DataGridViewCellStyle();
    ComponentResourceManager componentResourceManager = new
ComponentResourceManager(typeof (Form6));
    this.dateTimePicker1 = new DateTimePicker();
    this.dateTimePicker2 = new DateTimePicker();
    this.dateTimePicker3 = new DateTimePicker();
    this.dateTimePicker4 = new DateTimePicker();
    this.label1 = new Label();
}

```

```

this.label2 = new Label();
this.button1 = new Button();
this.label3 = new Label();
this.dataGridView1 = new DataGridView();
this.contextMenuStrip1 = new ContextMenuStrip(this.components);
this.педагогамуToolStripMenuItem = new ToolStripMenuItem();
this.вудалумуToolStripMenuItem = new ToolStripMenuItem();
this.dataGridView2 = new DataGridView();
this.contextMenuStrip2 = new ContextMenuStrip(this.components);
this.педагогамуToolStripMenuItem1 = new ToolStripMenuItem();
this.вудалумуToolStripMenuItem1 = new ToolStripMenuItem();
this.label4 = new Label();
this.label5 = new Label();
this.textBox1 = new TextBox();
this.button2 = new Button();
this.button3 = new Button();
this.button4 = new Button();
this.button5 = new Button();
this.comboBox1 = new ComboBox();
this.comboBox2 = new ComboBox();
this.label6 = new Label();
this.label7 = new Label();
this.label8 = new Label();
this.textBox2 = new TextBox();
this.textBox3 = new TextBox();
this.label9 = new Label();
this.button7 = new Button();
this.button6 = new Button();
((ISupportInitialize) this.dataGridView1).BeginInit();
this.contextMenuStrip1.SuspendLayout();
((ISupportInitialize) this.dataGridView2).BeginInit();
this.contextMenuStrip2.SuspendLayout();
this.SuspendLayout();
this.dateTimePicker1.Location = new Point(242, 24);
this.dateTimePicker1.Name = "dateTimePicker1";
this.dateTimePicker1.Size = new Size(182, 22);
this.dateTimePicker1.TabIndex = 0;
this.dateTimePicker2.Location = new Point(502, 24);
this.dateTimePicker2.Name = "dateTimePicker2";
this.dateTimePicker2.Size = new Size(182, 22);
this.dateTimePicker2.TabIndex = 0;
this.dateTimePicker3.Location = new Point(137, 315);
this.dateTimePicker3.Name = "dateTimePicker3";
this.dateTimePicker3.Size = new Size(182, 22);
this.dateTimePicker3.TabIndex = 0;
this.dateTimePicker4.Location = new Point(566, 315);
this.dateTimePicker4.Name = "dateTimePicker4";
this.dateTimePicker4.Size = new Size(182, 22);
this.dateTimePicker4.TabIndex = 0;
this.label1.AutoSize = true;
this.label1.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.label1.Location = new Point(50, 24);

```

```

this.label1.Name = "label1";
this.label1.Size = new Size(163, 17);
this.label1.TabIndex = 1;
this.label1.Text = "Закунку за період 3:";
this.label2.AutoSize = true;
this.label2.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.label2.Location = new Point(451, 24);
this.label2.Name = "label2";
this.label2.Size = new Size(31, 17);
this.label2.TabIndex = 1;
this.label2.Text = "no.";
this.button1.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.button1.Location = new Point(337, 65);
this.button1.Name = "button1";
this.button1.Size = new Size(120, 30);
this.button1.TabIndex = 2;
this.button1.Text = "Показати";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new EventHandler(this.button1_Click);
this.label3.AutoSize = true;
this.label3.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.label3.Location = new Point(27, 278);
this.label3.Name = "label3";
this.label3.Size = new Size(208, 17);
this.label3.TabIndex = 1;
this.label3.Text = "Прізвище Ім'я По батькові.";
gridViewCellStyle1.Alignment = DataGridViewContentAlignment.MiddleCenter;
gridViewCellStyle1.BackColor = SystemColors.Control;
gridViewCellStyle1.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Regular,
GraphicsUnit.Point, (byte) 204);
gridViewCellStyle1.ForeColor = SystemColors.WindowText;
gridViewCellStyle1.SelectionBackColor = SystemColors.Highlight;
gridViewCellStyle1.SelectionForeColor = SystemColors.HighlightText;
gridViewCellStyle1.WrapMode = DataGridViewTriState.True;
this.dataGridView1.ColumnHeadersDefaultCellStyle = gridViewCellStyle1;
this.dataGridView1.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dataGridView1.ContextMenuStrip = this.contextMenuStrip1;
gridViewCellStyle2.Alignment = DataGridViewContentAlignment.MiddleCenter;
gridViewCellStyle2.BackColor = SystemColors.Window;
gridViewCellStyle2.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Regular,
GraphicsUnit.Point, (byte) 204);
gridViewCellStyle2.ForeColor = SystemColors.ControlText;
gridViewCellStyle2.SelectionBackColor = SystemColors.Highlight;
gridViewCellStyle2.SelectionForeColor = SystemColors.HighlightText;
gridViewCellStyle2.WrapMode = DataGridViewTriState.False;
this.dataGridView1.DefaultCellStyle = gridViewCellStyle2;
this.dataGridView1.Location = new Point(30, 112);
this.dataGridView1.Name = "dataGridView1";
this.dataGridView1.RowHeadersWidth = 51;

```

```

this.dataGridView1.RowTemplate.Height = 24;
this.dataGridView1.Size = new Size(718, 139);
this.dataGridView1.TabIndex = 3;
this.dataGridView1.Click += new EventHandler(this.dataGridView1_Click);
this.dataGridView1.DoubleClick += new EventHandler(this.dataGridView1_DoubleClick);
this.contextMenuStrip1.ImageScalingSize = new Size(20, 20);
this.contextMenuStrip1.Items.AddRange(new ToolStripItem[2]
{
    (ToolStripItem) this.педагугамуToolStripMenuItem,
    (ToolStripItem) this.вудагугамуToolStripMenuItem
});
this.contextMenuStrip1.Name = "contextMenuStrip1";
this.contextMenuStrip1.Size = new Size(155, 52);
this.педагугамуToolStripMenuItem.Name = "педагугамуToolStripMenuItem";
this.педагугамуToolStripMenuItem.Size = new Size(154, 24);
this.педагугамуToolStripMenuItem.Text = "Педагугаму";
this.педагугамуToolStripMenuItem.Click += new
EventHandler(this.педагугамуToolStripMenuItem_Click);
this.вудагугамуToolStripMenuItem.Name = "вудагугамуToolStripMenuItem";
this.вудагугамуToolStripMenuItem.Size = new Size(154, 24);
this.вудагугамуToolStripMenuItem.Text = "Вудагугаму";
this.вудагугамуToolStripMenuItem.Click += new
EventHandler(this.вудагугамуToolStripMenuItem_Click);
gridViewCellStyle3.Alignment = DataGridViewContentAlignment.MiddleCenter;
gridViewCellStyle3.BackColor = SystemColors.Control;
gridViewCellStyle3.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Regular,
GraphicsUnit.Point, (byte) 204);
gridViewCellStyle3.ForeColor = SystemColors.WindowText;
gridViewCellStyle3.SelectionBackColor = SystemColors.Highlight;
gridViewCellStyle3.SelectionForeColor = SystemColors.HighlightText;
gridViewCellStyle3.WrapMode = DataGridViewTriState.True;
this.dataGridView2.ColumnHeadersDefaultCellStyle = gridViewCellStyle3;
this.dataGridView2.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dataGridView2.ContextMenuStrip = this.contextMenuStrip2;
gridViewCellStyle4.Alignment = DataGridViewContentAlignment.MiddleCenter;
gridViewCellStyle4.BackColor = SystemColors.Window;
gridViewCellStyle4.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Regular,
GraphicsUnit.Point, (byte) 204);
gridViewCellStyle4.ForeColor = SystemColors.ControlText;
gridViewCellStyle4.SelectionBackColor = SystemColors.Highlight;
gridViewCellStyle4.SelectionForeColor = SystemColors.HighlightText;
gridViewCellStyle4.WrapMode = DataGridViewTriState.False;
this.dataGridView2.DefaultCellStyle = gridViewCellStyle4;
this.dataGridView2.Location = new Point(30, 405);
this.dataGridView2.Name = "dataGridView2";
this.dataGridView2.RowHeadersWidth = 51;
this.dataGridView2.RowTemplate.Height = 24;
this.dataGridView2.Size = new Size(718, 120);
this.dataGridView2.TabIndex = 4;
this.contextMenuStrip2.ImageScalingSize = new Size(20, 20);
this.contextMenuStrip2.Items.AddRange(new ToolStripItem[2]
{

```

```

    (ToolStripItem) this.педагогумуToolStripMenuItem1,
    (ToolStripItem) this.видалумуToolStripMenuItem1
});
this.contextMenuStrip2.Name = "contextMenuStrip2";
this.contextMenuStrip2.Size = new Size(155, 52);
this.педагогумуToolStripMenuItem1.Name = "педагогумуToolStripMenuItem1";
this.педагогумуToolStripMenuItem1.Size = new Size(154, 24);
this.педагогумуToolStripMenuItem1.Text = "Педагогуму";
this.педагогумуToolStripMenuItem1.Click += new
EventHandler(this.педагогумуToolStripMenuItem1_Click);
this.видалумуToolStripMenuItem1.Name = "видалумуToolStripMenuItem1";
this.видалумуToolStripMenuItem1.Size = new Size(154, 24);
this.видалумуToolStripMenuItem1.Text = "Видалуму";
this.видалумуToolStripMenuItem1.Click += new
EventHandler(this.видалумуToolStripMenuItem1_Click);
this.label4.AutoSize = true;
this.label4.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.label4.Location = new Point(27, 315);
this.label4.Name = "label4";
this.label4.Size = new Size(76, 17);
this.label4.TabIndex = 1;
this.label4.Text = "Початок:";
this.label5.AutoSize = true;
this.label5.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.label5.Location = new Point(437, 315);
this.label5.Name = "label5";
this.label5.Size = new Size(98, 17);
this.label5.TabIndex = 1;
this.label5.Text = "Закінчення:";
this.textBox1.Location = new Point(157, 551);
this.textBox1.Name = "textBox1";
this.textBox1.ReadOnly = true;
this.textBox1.Size = new Size(162, 22);
this.textBox1.TabIndex = 7;
this.button2.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.button2.Location = new Point(266, 356);
this.button2.Name = "button2";
this.button2.Size = new Size(120, 30);
this.button2.TabIndex = 11;
this.button2.Text = "Зберегти";
this.button2.UseVisualStyleBackColor = true;
this.button2.Click += new EventHandler(this.button2_Click);
this.button3.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.button3.Location = new Point(408, 356);
this.button3.Name = "button3";
this.button3.Size = new Size(120, 30);
this.button3.TabIndex = 12;
this.button3.Text = "Відміна";
this.button3.UseVisualStyleBackColor = true;

```

```

    this.button3.Click += new EventHandler(this.button3_Click);
    this.button4.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
    this.button4.Location = new Point(266, 632);
    this.button4.Name = "button4";
    this.button4.Size = new Size(120, 30);
    this.button4.TabIndex = 13;
    this.button4.Text = "Зберегти";
    this.button4.UseVisualStyleBackColor = true;
    this.button4.Click += new EventHandler(this.button4_Click);
    this.button5.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
    this.button5.Location = new Point(408, 632);
    this.button5.Name = "button5";
    this.button5.Size = new Size(120, 30);
    this.button5.TabIndex = 14;
    this.button5.Text = "Відміна";
    this.button5.UseVisualStyleBackColor = true;
    this.button5.Click += new EventHandler(this.button5_Click);
    this.comboBox1.FormattingEnabled = true;
    this.comboBox1.Location = new Point(309, 275);
    this.comboBox1.Name = "comboBox1";
    this.comboBox1.Size = new Size(439, 24);
    this.comboBox1.TabIndex = 16;
    this.comboBox2.FormattingEnabled = true;
    this.comboBox2.Location = new Point(515, 551);
    this.comboBox2.Name = "comboBox2";
    this.comboBox2.Size = new Size(233, 24);
    this.comboBox2.TabIndex = 17;
    this.label6.AutoSize = true;
    this.label6.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
    this.label6.Location = new Point(386, 554);
    this.label6.Name = "label6";
    this.label6.Size = new Size(85, 17);
    this.label6.TabIndex = 1;
    this.label6.Text = "Сировина:";
    this.label7.AutoSize = true;
    this.label7.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
    this.label7.Location = new Point(424, 591);
    this.label7.Name = "label7";
    this.label7.Size = new Size(47, 17);
    this.label7.TabIndex = 1;
    this.label7.Text = "Ціна:";
    this.label8.AutoSize = true;
    this.label8.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
    this.label8.Location = new Point(27, 591);
    this.label8.Name = "label8";
    this.label8.Size = new Size(80, 17);
    this.label8.TabIndex = 1;
    this.label8.Text = "Кількість:";

```

```

this.textBox2.Location = new Point(157, 591);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new Size(162, 22);
this.textBox2.TabIndex = 7;
this.textBox3.Location = new Point(515, 591);
this.textBox3.Name = "textBox3";
this.textBox3.Size = new Size(233, 22);
this.textBox3.TabIndex = 7;
this.label9.AutoSize = true;
this.label9.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.label9.Location = new Point(27, 551);
this.label9.Name = "label9";
this.label9.Size = new Size(96, 17);
this.label9.TabIndex = 1;
this.label9.Text = "Код снукку:";
this.button7.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.button7.Location = new Point(30, 65);
this.button7.Name = "button7";
this.button7.Size = new Size(200, 30);
this.button7.TabIndex = 19;
this.button7.Text = "Звим по снівробітнику";
this.button7.UseVisualStyleBackColor = true;
this.button7.Click += new EventHandler(this.button7_Click);
this.button6.Font = new Font("Microsoft Sans Serif", 7.8f, FontStyle.Bold, GraphicsUnit.Point,
(byte) 204);
this.button6.Location = new Point(548, 65);
this.button6.Name = "button6";
this.button6.Size = new Size(200, 30);
this.button6.TabIndex = 18;
this.button6.Text = "Накладна";
this.button6.UseVisualStyleBackColor = true;
this.button6.Click += new EventHandler(this.button6_Click);
this.AutoScaleDimensions = new.SizeF(8f, 16f);
this.AutoScaleMode = AutoScaleMode.Font;
this.BackColor = Color.Honeydew;
this.ClientSize = new Size(782, 678);
this.Controls.Add((Control) this.button);
this.Controls.Add((Control) this.textBox);
this.Controls.Add((Control) this.label);
this.Controls.Add((Control) this.dateTimePicker);
this.Icon = (Icon) componentResourceManager.GetObject($"$this.Icon");
this.MaximizeBox = false;
this.MinimizeBox = false;
this.Name = nameof (Form6);
this.StartPosition = FormStartPosition.CenterScreen;
this.Text = "Закупку";
this.Shown += new EventHandler(this.Form6_Shown);
((ISupportInitialize) this.dataGridView1).EndInit();
this.contextMenuStrip1.ResumeLayout(false);
((ISupportInitialize) this.dataGridView2).EndInit();
this.contextMenuStrip2.ResumeLayout(false);

```

```
this.ResumeLayout(false);  
this.PerformLayout();  
}  
}  
}
```