

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних наук

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка сайту для продажів електроніки з можливістю адміністрування товарів

Виконав(ла): студент(ка) 4 курсу, групи СТс-42

спеціальності 126 Інформаційні системи

та технології

(шифр і назва спеціальності)

Прокопчук В.Д.

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

Гром'як Р.С.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Стадник М.А.

(прізвище та ініціали)

Тернопіль
2021

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«25» січня 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 126 Інформаційні системи та технології
(шифр і назва спеціальності)

студенту Прокопчук Віталій Дмитрович
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка сайту для продажів електроніки з
можливістю адміністрування товарів

Керівник роботи к.ф-м.н., доц. Гром'як Р.С.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «01» лютого 2021 року № 4/7-63

2. Термін подання студентом завершеної роботи 15 червня 2021 р.

3. Вихідні дані до роботи Опис предметної області

4. Зміст роботи (перелік питань, які потрібно розробити)

ВСТУП 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 1.1 Концептуальна і логічна модель бази даних 1.2
Загальна інформація про програму 1.3 Системні вимоги 1.4 Порядок інсталяції 1.5 З'єднання з
магазином 1.6 Використання розробленої програми 2 ПРАКТИЧНА РЕАЛІЗАЦІЯ 2.1 Архітектура
Інтернет-магазину 2.2 Принципи та засоби створення Web-сайтів 2.3 Концептуальне проектування
сайту 2.4 Фізична реалізація магазину 3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ
ПРАЦІ 3.1 Заходи по покращенню питань з охорони праці та техніки безпеки на підприємстві 3.2
Розрахунок вентиляції; ВИСНОВОК ; ПЕРЕЛІК ДЖЕРЕЛ; Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема. 2. Структурна схема сайту магазину. 3. Функціональна схема програмного модуля
адміністрування 4. Логічна структура бази даних. 5. Алгоритм імпорту даних. 6. Приклади інтерфейсу
Користувача. 7. Вигляд головної сторінки сайту. 8 Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., к.т.н., доц.		

7. Дата видачі завдання 25 січня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	25.01.21-27.01.21	<i>Виконано</i>
2.	Підбір джерел по темі роботи	28.01.21 – 01.04.21	<i>Виконано</i>
3.	Оформлення першого розділу	15.04.2021	<i>Виконано</i>
4.	Оформлення другого розділу	30.04.2021	<i>Виконано</i>
5.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	15.05.2021	<i>Виконано</i>
6.	Оформлення кваліфікаційної роботи	07.06.2021	<i>Виконано</i>
7.	Перевірка на плагіат	07.06.2021	<i>Виконано</i>
8.	Нормоконтроль	15.06.2021	<i>Виконано</i>
9.	Попередній захист кваліфікаційної роботи	18.06.2021	<i>Виконано</i>
10.	Захист кваліфікаційної роботи	26.06.2021	

Студент

_____ (підпис)

Прокопчук В.Д.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Гром'як Р.С.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка сайту для продажів електроніки з можливістю адміністрування товарів // Кваліфікаційна робота бакалавра // Прокопчук Віталій Дмитрович // Тернопільський національний технічний університет імені І. Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СТс-42 // – Тернопіль, 2021 // с. – 52, рис. – 23, табл. – 5, слайдів. – 8, додат. – 2, бібліогр. – 14.

Ключові слова: інтернет-магазин, адміністратор, користувач, інтернет, веб-програмування, база даних, mysql, цілісність посилань, реляційна база даних, адміністрування, користувач, товар, категорія.

Кваліфікаційна робота стосується розробки інтернет магазину електроніки загалом та програмного забезпечення для його адміністрування зокрема, а саме бази даних як ядра інформаційної системи для адміністрування електронного магазину, який може займатися продажем будь-яких товарів через мережу Інтернет.

При роботі над завданням даної роботи були враховані всі фактори для зручної роботи користувача з кінцевим програмним продуктом, такими як багатовіконний інтерфейс, можливість розташування вікон у зручному порядку, просте та піктографічне меню, допомога по програмі.

ANNOTATION

Site development for electronic goods sales with possible products administration
// Qualification work of the educational level "Bachelor" // Prokopchuk Vitalii // Ternopil
Ivan Puluj National Technical University, Faculty of Computer Information Systems and
Software Engineering, Department of Computer Science, Group CTc-42 // Ternopil, 2021
// p. – 52, fig. – 23, references – 14, posters – 8, applications – 2.

Keywords: online store, administrator, user, internet, web programming, database,
mysql, link integrity, relational database, administration, user, product, category.

Qualification work concerns the development of an online store in general and
software for its administration in particular, namely the database as the core of the
information system for the administration of an electronic store that can sell any goods
via the Internet.

When working on the task of this work, all factors were taken into account for the
user's convenient work with the final software product, such as multi-window interface,
the ability to arrange windows in a convenient order, simple and pictographic menu,
program help.

ЗМІСТ

ВСТУП	
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	
1.1 Концептуальна і логічна модель бази даних	
1.2 Загальна інформація про програму	
1.3 Системні вимоги	
1.4 Порядок інсталяції	
1.5 З'єднання з магазином	
1.6 Використання розробленої програми	
1.6.1 Керування каталогом	
1.6.2 Керування товарами	
1.6.3 Управління клієнтами та замовленнями	
1.6.4 Формування звітів	
2 ПРАКТИЧНА РЕАЛІЗАЦІЯ	
2.1 Архітектура Інтернет-магазину	
2.2 Принципи та засоби створення Web-сайтів	
2.2.1 Огляд технологій створення сайту	
2.2.2 Огляд інструментів для створення сайту	
2.3 Концептуальне проектування сайту	
2.4 Фізична реалізація магазину	
2.4.1 Вибір інструментів для створення дизайну сайту	
2.4.2 Опис роботи та інтерфейсу Інтернет-магазину	
2.4.3 Здійснення покупок	
3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ	
3.1 Заходи по покращенню питань з охорони праці та техніки безпеки на підприємстві	
3.2 Розрахунок вентиляції	
3.2.1 Природна вентиляція	
3.2.2 Штучна вентиляція	

3.3 Аналіз небезпеки і шкідливості при виконанні розробки	
ВИСНОВОК	
ПЕРЕЛІК ДЖЕРЕЛ	
Додатки	

ВСТУП

Останнім часом багато приватних осіб та компаній не уявляють своєї роботи без Інтернету: електронна пошта, FTP, і звичайно, WWW. Ця глобальна комп'ютерна мережа дозволяє оперативно взнати всю необхідну інформацію про діяльність підприємств, людей, про новини тощо. Зрозуміло, що вся інформація повинна оперативно поновлюватись.

Все частіше користувачі Інтернету хочуть бачити не тільки статичну інформацію, яку їм "викладає" на своїх сайтах та чи інша компанія, але і спостерігати за динамікою зміни даних, ставити запити та отримувати коректні відповіді на них. Тобто по суті зараз в Інтернеті є тенденція до відходу від простих статичних сторінок. На зміну їм приходять сайти, створені з широким використанням різного роду скриптів, CGI-програм тощо. А використання всіх можливостей цих нових засобів неможливе без баз даних.

З наведеного вище можна зробити висновок, що сучасний розробник програмного забезпечення для мережі Internet повинен бути широким спеціалістом з сучасних мов програмування, з баз даних та розуміти протоколи обміну даними в комп'ютерних мережах.

На сьогодні існує велика кількість засобів розробки програмного забезпечення взагалі та для Інтернету зокрема. Досить тільки згадати про C, C++, Pascal (універсальні потужні мови програмування високого рівня), Perl, PHP, ASP (мови для написання виконуваних на стороні web-сервера скриптів), Oracle, Informix, MS SQL Server, MySQL, Interbase (сервери баз даних), а також засоби для розробки програм будь-якої складності та призначення з використанням RAD-інструментів (rapid application design – швидка розробка програм) – Visual Studio, Delphi. Кожен із згаданих засобів та інструментів краще підходить для вирішення одних задач, а для вирішення інших можна використати інші засоби. Крім того, всі вони можуть працювати на різних платформах і операційних системах по різному.

Тому вибір засобів розробки залежить, по перше, від призначення розроблюваної програми, а, по друге, від того, наскільки розробник чи команда розробників добре володіють тими чи іншими засобами розробки.

Отже в даній роботі буде показано створення всі етапи створення віртуального магазину, використовуючи уміння і навички набуті під час навчання в університеті. Оскільки розробка цілого інтернет-магазину – задача непроста навіть для команди розробників, то в цій кваліфікаційній роботі більше уваги буде присвячено розробці програмного забезпечення для адміністрування електронного магазину – цієї області, до якої автор роботи має безпосереднє відношення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Концептуальна і логічна модель бази даних

Розробку концептуальної моделі почнемо з виділення всіх типів сутностей, інформацію про котрі потрібно тримати в базі даних.

В нашому випадку це наступні типи сутностей:

- замовники;
- замовлення;
- товари;
- постачальники;
- виробники;
- працівники.

Для вказаних типів сутностей встановлено таке рольове відношення:

- замовники роблять замовлення
- замовники володіють товарами;
- замовлення робляться на товари;
- замовлення приймають працівники;
- для виконання замовлень потрібні товари;
- постачальники поставляють товари;
- товари поставляються від виробників.

Окремої уваги заслуговують два зв'язки.

Перший – зв'язок між сутністю «замовлення» та «запчастини». Для кожного замовлення потрібно багато запчастин, але з другого боку кожна запчастина використовується в багатьох замовленнях. Тобто маємо кардинальність зв'язку «багато-до-багатьох».

Другий – зв'язок між сутностями «постачальники» та «виробники». Кожен постачальник займається продажем запчастин багатьох виробників. З другого боку – кожен виробник для забезпечення належного сервісу свого обладнання

постачає запчастини багатьом дистрибуторам. Тому кардинальність цього зв'язку теж «багато-до-багатьох».

Схема даних, описана вище, зображена н рисунку 1.1.

На цьому ж малюнку зображено атрибути для кожного з типів сутностей.

Для зв'язків з такою кардинальністю є проблеми з реалізацією засобами будь-якої сучасної СКБД. Access не є виключенням. Тому таких зв'язків позбуваються шляхом їх декомпозиції на два з кардинальністю «один-до-багатьох» кожен.

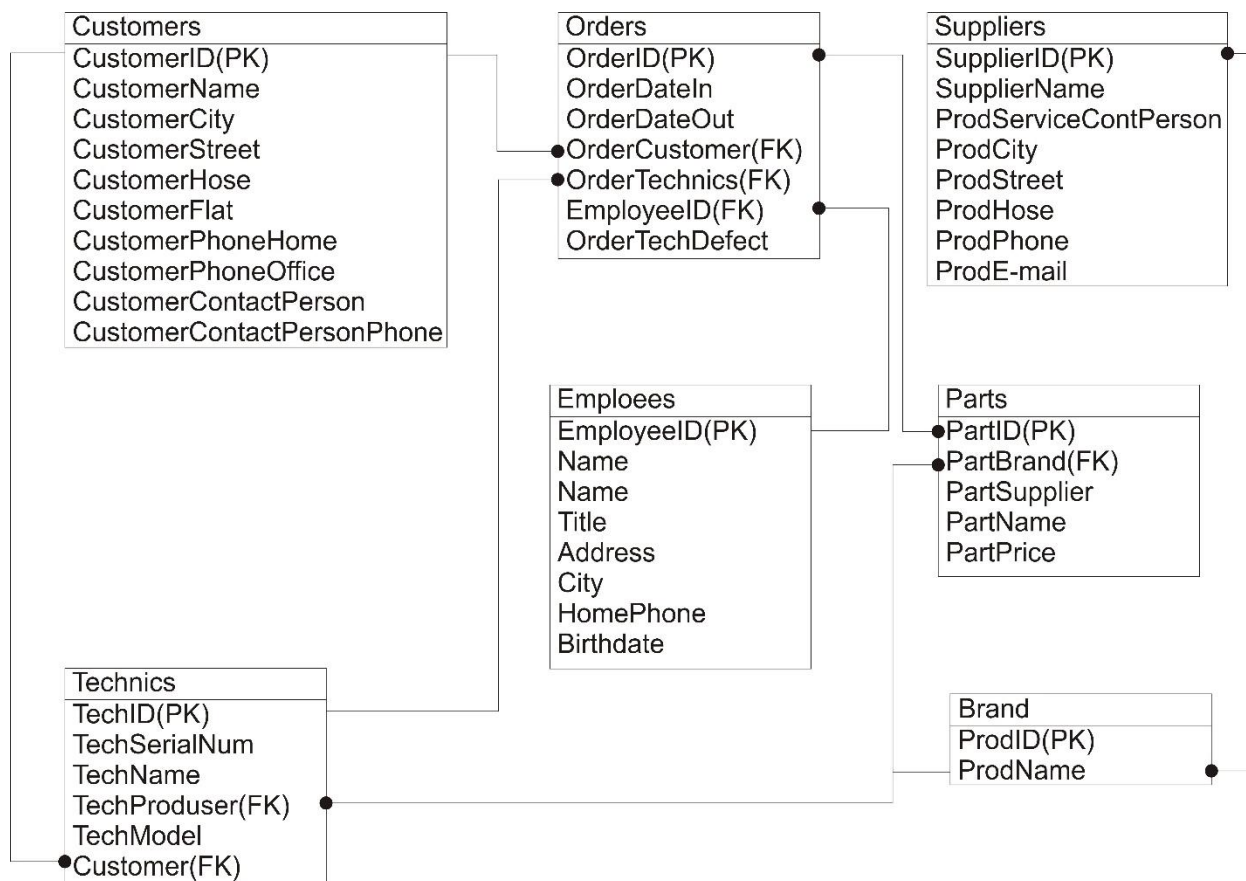


Рисунок 1.1 – Логічна структура розробленої бази даних

Для цього потрібно ввести проміжний тип сутності. Його можна утворити чисто штучним чином, взявши в якості атрибутів первинні ключі основних атрибутів. Але можна виділити і існуючий тип сутності, до складу якого входять в якості зовнішніх ключів первинні ключі основних сутностей.

В нашому випадку штучно утворювати проміжні сутності не потрібно. Шляхом більш детального аналізу предметної області було виділено ще два типи:

перший – «рахунки» і другий – «поставки». Сутність «рахунки» є зв’язуючою ланкою між сутностями «замовлення» та «запчастини». За змістом роботи даний тип сутності відображає виписаний рахунок на кожну деталь, використану при ремонті техніки, оформленої даним замовленням.

Таким чином, поставивши до бази даних запит, який читатиме інформацію з таблиці про рахунки, і передавши йому в якості параметра значення ідентифікатора замовлення, можна підрахувати його вартість. Саме таким чином працює частина розробленого проекту, яка реалізує виписку рахунків за надані послуги.

Сутність «поставки» стає зв’язуючою ланкою між сутностями «виробники техніки» та «постачальники». Завдяки наявності цієї сутності в базі даних з’являється можливість вести облік кожної операції поставки.

Вдосконалена схема бази даних показана на рисунку 1.2. Саме на основі цієї моделі в проекті розроблено схему бази даних, її таблиці, встановлено цілісність посилань (набір правил зміни значень зовнішніх ключів при зміні первинних ключів головної таблиці).

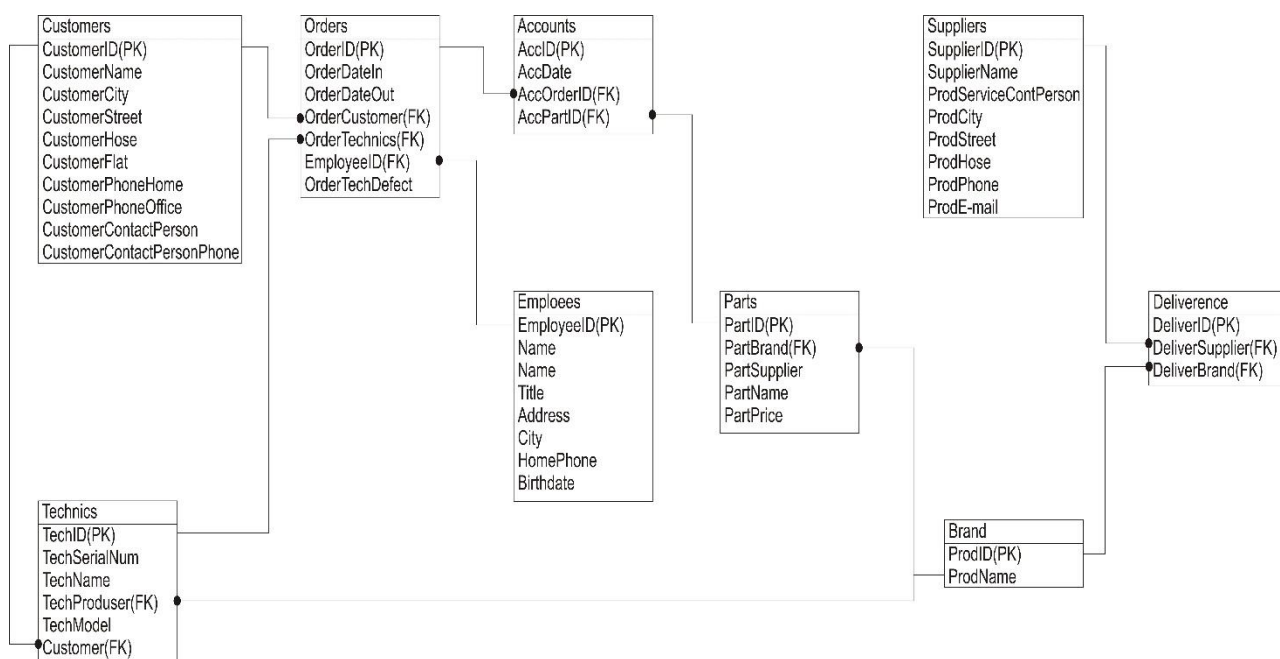


Рисунок 1.2 – Схема бази даних з усуненими зв’язками з кардинальністю «багато-до-багатьох»

Отримавши таку схему даних, її безпосередньо можна перенести на логічний рівень. На рівні логічної моделі кожен тип сутностей буде відображатись таблицею бази даних. Список таблиць та їх атрибутів з описами подано в таблиці 1.1

Таблиця 1.1 – Таблиці та їх атрибути розробленої бази даних

Таблиця	Атрибути	Опис
Customers (замовники)	CustomerID	Ідентифікатор замовника (первинний ключ)
	CustomerName	Ім'я замовника (первинний ключ)
	CustomerCity	Місто проживання замовника
	CustomerStreet	Вулиця замовника
	CustomerHose	Номер будинку замовника
	CustomerFlat	Номер квартири замовника
	CustomerPhoneHome	Номер домашнього телефону
	CustomerPhoneOffice	Номер робочого телефону
	CustomerContactPerson	Ім'я контактної особи
	CustomerContactPersonPhone	Телефон контактної особи
Technics (техніка)	TechID	Ідентифікатор техніки (первинний ключ)
	TechSerialNum	Серійний номер приладу
	TechName	Назва
	TechProducer	Виробник (марка)
	TechModel	Модель
	TechCustomer	Власник (замовник)
Orders (замовлення)	OrderID	Ідентифікатор замовлення (первинний ключ)
	OrderDateIn	Дата прийому на ремонт
	OrderDateOut	Дата видачі замовлення
	OrderCustomer	На кого замовлення (ім'я)
	EmployeeID	Хто прийняв замовлення (працівник)

Accounts (рахунки)	AccID	Ідентифікатор рахунку (первинний ключ)
	AccDate	Дата виписки рахунку
	AccPartID	Ідентифікатор використаної запчастини
	AccPartQuant	Кількість використаних запчастин
	AccOrderID	Ідентифікатор замовлення, для якого виписується рахунок
Employees (працівники)	EmployeeID	Ідентифікатор працівника (первинний ключ)
	Name	Ім'я працівника
	Title	Посада працівника
	Address	Адреса працівника
	City	Місто проживання працівника
	HomePhone	Номер домашнього телефону
	Birthdate	Дата народження
Parts (запчастини)	PartID	Ідентифікатор запчастини (первинний ключ)
	PartBrand	Для техніки якого виробника
	PartSupplier	Постачальник
	PartName	Назва запчастини
	PartPrice	Ціна за одиницю
Suppliers (постачальники)	SupplierID	Ідентифікатор постачальника (первинний ключ)
	SupplierName	Назва постачальника
	ProdServiceContPerson	Ім'я контактної особи
	ProdCity	Місто розміщення постачальника
	ProdStreet	Вулиця розміщення постачальника
	ProdHose	Номер будинку постачальника
	ProdPhone	Номер телефону постачальника
	ProdE-mail	Електронна пошта постачальника
Brand (виробники техніки)	ProdID	Ідентифікатор виробника (первинний ключ)
	ProdName	Назва виробника техніки
Deliverence (поставки)	DeliverID	Ідентифікатор поставки (первинний ключ)
	DeliverSupplier	Постачальник
	DeliverBrand	Що поставлено

Для кожного атрибуту всіх таблиць встановлено також типи даних, які в загальному випадку визначають домени атрибутів. Додаткові обмеження на значення певних атрибутів не встановлювались, окрім атрибуту «назва виробника», який означено як альтернативний ключ (унікальний індекс). Це зроблено для зручності організації зв'язку між таблицями, щоби у підлеглих таблицях користуватись не числовим ідентифікатором, а осмисленим іменем. Опис доменів атрибутів з вказанням типів даних і розмірів наведено в таблиці 1.2.

Таблиця 1.2 – Типи даних стовпців таблиць бази даних

Таблиця	Атрибути	Тип даних	Розмір
Customers (замовники)	CustomerID	AutoIncrement	Long Integer
	CustomerName	Text	25
	CustomerCity	Text	30
	CustomerStreet	Text	20
	CustomerHose	Text	6
	CustomerFlat	Text	4
	CustomerPhoneHome	Text	15
	CustomerPhoneOffice	Text	15
	CustomerContactPerson	Text	30
CustomerContactPersonPhone	Text	15	
Technics (техніка)	TechID	AutoIncrement	Long Integer
	TechSerialNum	Text	30
	TechName	Text	15
	TechProduser	Text	10
	TechModel	Text	10
Orders (замовлення)	TechCustomer	Number	Long Integer
	OrderID	AutoIncrement	Long Integer
	OrderDateIn	Date/Time	Short date
	OrderDateOut	Date/Time	Short date
	OrderCustomer	Number	Long Integer
	OrderTechnics	Number	Long Integer
	OrderTechDefect	Memo	
EmployeeID	Number	Long Integer	
Accounts (рахунки)	AccID	AutoIncrement	Long Integer
	AccDate	Date/Time	Short Date
	AccPartID	Number	Long Integer
	AccPartQuant	Number	Long Integer
	AccOrderID	Number	Long Integer

Employees (працівники)	EmployeeID	AutoIncrement	Long Integer
	Name	Text	30
	Title	Text	20
	Address	Text	40
	City	Text	15
	HomePhone	Text	15
	Birthdate	Date	Short Date
Parts (запчастини)	PartID	AutoIncrement	Long Integer
	PartBrand	Text	10
	PartSupplier	Number	Long Integer
	PartName	Text	30
	PartPrice	Number	Currency
Suppliers (постачальники)	SupplierID	AutoIncrement	Long Integer
	SupplierName	Text	20
	ProdServiceContPerson	Text	20
	ProdCity	Text	15
	ProdStreet	Text	15
	ProdHose	Text	6
	ProdPhone	Text	15
	ProdE-mail	Text	30
Brand (виробники техніки)	ProdID	AutoIncrement	Long Integer
	ProdName	Text	10
Deliverence (поставки)	DeliverID	AutoIncrement	Long Integer
	DeliverSupplier	Number	Long Integer
	DeliverBrand	Number	Long Integer

Як не раз було згадано у вступному розділі, в якостів середовища роботи розробленої бази даних вибрано MySQL. Вказання типів даних для усіх стовпців таблиць враховує особливості реалізації саме для MySQL версії 5.

1.2 Загальна інформація про програму

Store Manager – прикладна програма для ОС Windows для швидкого та ефективного управління інтернет-магазином. Це незатратна, багатомовна програма. Store Manager – легке в роботі та інтуїтивно зрозуміле програмне забезпечення, яке може використовуватись навіть не ІТ-спеціалістами. Для

ефективного управління товарами, клієнтами, замовленнями, цінами та іншою інформацією магазину нічого іншого, крім програми не потрібно.

Основні функціональні можливості розробленого програмного продукту:

- управління багатьма магазинами;
- управління товарами, категоріями, атрибутами, виробниками та замовленнями;
- швидкий та легкий імпорту/експорт даних в/з програми;
- одночасні маніпуляції цінами багатьох продуктів;
- легка навігація, пошук товарів, розвинений механізм фільтрації даних;
- швидкий пошук товару, замовлення, клієнта тощо;
- розвинений механізм формування звітів, в т.ч. більше 20 стандартних звітів.

Багато додаткових можливостей також включено у Store Manager для спрощення як типових, так і складніших операцій. Всі внесені зміни відображаються в той же час, не використовуються синхронізація даних чи запис через встановлений час – лише "живі" дані. Store Manager для забезпечення управління багатьма магазинами має механізми швидкого перемикання між ними. Всі ці властивості доступні навіть для звичайного настільного комп'ютера.

При адмініструванні електронних магазинів часто доводиться робити висновок, що деякі речі можна було б реалізувати краще, ефективніше, ніж вони виконані в наявному ПЗ. Ось чому широкі можливості налаштувань – особливість розробленого ПЗ, націлена на туж саму мету: забезпечити максимум можливостей для користувача (адміністратора електронного магазину).

1.3 Системні вимоги

Windows 7/8/10 та завантажене і встановлене програмне забезпечення для "кошика" (Shopping Cart): osCommerce, Zen Cart or CRE.

1.4 Порядок інсталяції

Перед початком використання Store Manager потрібно купити ліцензію (окрема ліцензія на кожен комп'ютер) на це ПЗ. Після створення профілю на сайті виробника (а, отже, і оплати за корисмування) утановка відбувається за допомогою майстраустановки, як і для більшості програм для ОС Windows. Корстувач при утановці проходить наступні етапи:

- згода з умовами ліцензії;
- вибір місця на диску для встановлення ПЗ;
- вибір компонентів для утановки (повна утановка, компакта чи детальна);
- вибір папки для меню Start (для розміщення ярликів програми);
- виконання додаткових завдань.

Майстер видасть інформацію про успішну утановку. Після утановки Store Manager потребує реєстрації.

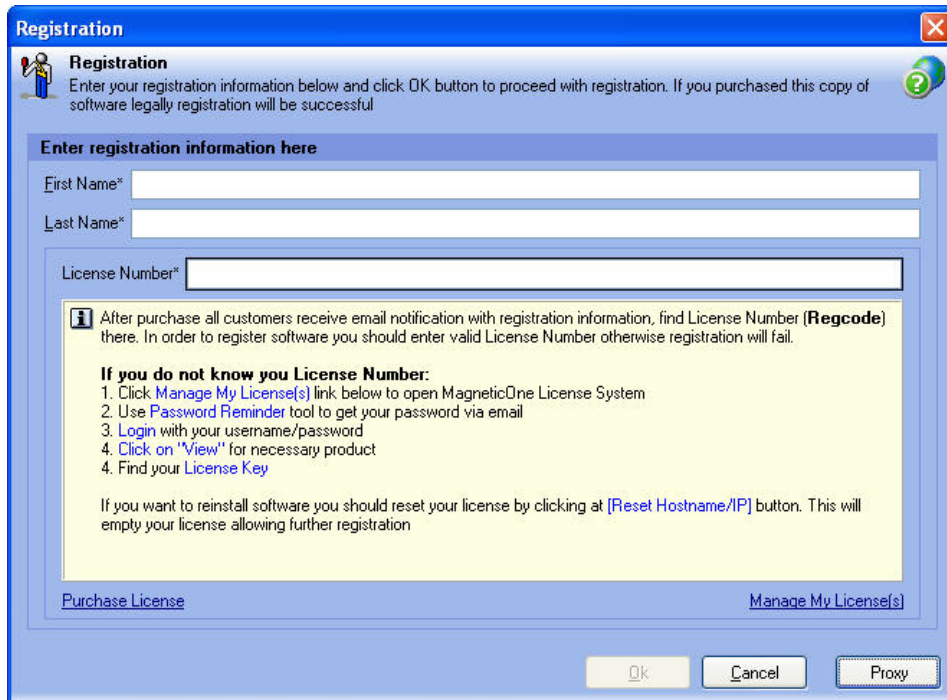


Рисунок 1.3 – Вікно реєстрації програми

Таким чином, при першому запуску програми потрібно ввести свої дані, які використовувались при створенні профілю на сайті виробника. Вікно реєстрації зображено на рисунку 1.3.

Програма запитає користувача, чи потрібно скачати з мережі Інтернет найостаннішу версію (у випадку її наявності), а також про оновлення ПЗ, яке виконується при запуску програми.

1.5 З'єднання з магазином

Для початку роботи з конкретним магазином потрібно налаштувати параметри з'єднання з ним. Ці параметри можуть бути встановлені при установці або пізніше – з головного меню програми Tools -> Preferences.

Вікно налаштувань дозволяє користувачу конфігурувати з'єднання з базою даних, та встановлювати інші параметри. Наприклад, створювати, редагувати та знищувати з'єднання як з локальними, так і з віддаленими базами даних. Докладно процес налаштування з'єднання описаний нижче.

Також меню налаштувань містить пункт для "тонких" налаштувань, а також налаштування повідомлень, які користувач може змінювати при бажанні.

Секція налаштувань Інтернет-з'єднання призначена для задання параметрів підключення (Proху та SMTP).

Секція меню Table Remapping дозволяє налаштувати Store Manager для роботи з переіменованими таблицями, якщо таке було зроблено.

База даних, котра встановлюється з програмою – тимчасова. Вона використовується з різною метою, але в основному для утворення мотсту з РНР MySQL. Тобто можна працювати з локальною копією віддаленої бази, а після внесення всіх змін – синхронізувати віддалену базу магазину з локальною її копією.

Можна створити довільну кількість потрібних баз даних. Для цього потрібно виконати заповнення полів з характеристиками бази, а саме:

- хост бази даних (доменне ім'я чи IP-адреса);
- номер порту (3306 – за замовчуванням);

- версія сервера MySQL;
- назва профілю та пароль користувача;
- ім'я бази даних.

Store Manager вимагає прямого підключення до СУБД MySQL. Більшість компаній-хостерів дозволяють таке з'єднання. Таким чином, використовуючи стандартні засоби управління сайтом, надані хостером, в список управління доступом потрібно додати IP-адреси комп'ютерів, з яких виконуватиметься адміністрування магазину та дозволити використання порта 3306. Можна перевірити параметри підключення, натиснувши кнопку "Test Connection".

FTP-підключення не є обов'язковим, тому при налаштуванні програми його параметри можна залишити без змін. FTP використовується для малюнків (зображення товарів), а саме – їх завантаження та відправка на сервер магазину.

Якщо пряме підключення до сервера з магазином не дозволяється, то можна використати безпечне SSH-з'єднання. SSH є видом тунелю між сервером та його адміністратором. Використовуючи цей тунель, може бути встановлено інше з'єднання. Наприклад для підключення до MySQL чи до FTP.

Слід зауважити, що SSH не означає автоматичної настройки MySQL. Тобто для підключення до бази даних потрібно це підключення створити. Якщо підключення до MySQL ні на пряму, ні через SSH-тунель неможливе (недозволено), то потрібно використовувати міст PHP MySQL.

Міст – ще один спосіб підключитись до бази даних на віддаленому сервері, для котрого пряме з'єднання не дозволено. Цей міст насправді ніякого з'єднання не встановлює. Він лише передає файли до/з віддаленої машини. А тому використання такого способу роботи з базою даних є повільним і в нормальних умовах не рекомендується.

За допомогою моста виконуються дві основні операції: отримання даних з віддаленої бази даних та направлення даних у базу.

1.6 Використання розробленої програми

Створення нового магазину не є складним процесом при умові застосування розробленої програми. Адже чисельні опції забезпечують виский рівень гнучкості та надання кожному магазину індивідуальності. Елементи інтерфейсу програми забезпечують керування каталогом товарів магазину, його користувачами (покупцями), формуванням звітів тощо.

1.6.1 Керування каталогом

Секція керування каталогом стосується всіх товарів та їх параметрів. Адже кожен товар належить до певної категорії. Тут відбувається управління товарами на рівні категорій.

Тому в цій секції можна задати назву категорії, додати нову категорію, видалити непотрібні категорії чи змінити параметри існуючих. Управління включає такі основні кроки.

1. Основні операції (add, edit, delete).

За замовчуванням встановлюється декілька стандартних категорій і їх можна поміняти через панель інструментів чи використовуючи контекстне меню категорії (рисунок 1.4).

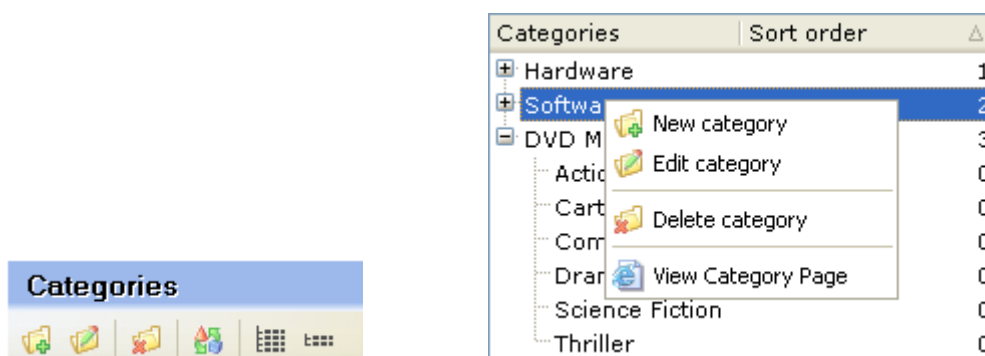


Рисунок 1.4 – Зміна категорії товару

При створенні нової категорії потрібно вказати: батьківську категорію (для кореневої це значення залишають порожнім), ввести назву категорії, вибрати

зображення для категорії, встановити порядок сортування товарів у категорії (по алфавіту за замовчуванням).

Після створення нова категорія може бути змінена чи видалена у будь-який час. Видалення категорії можливе трьома способами (рисунок 1.5).

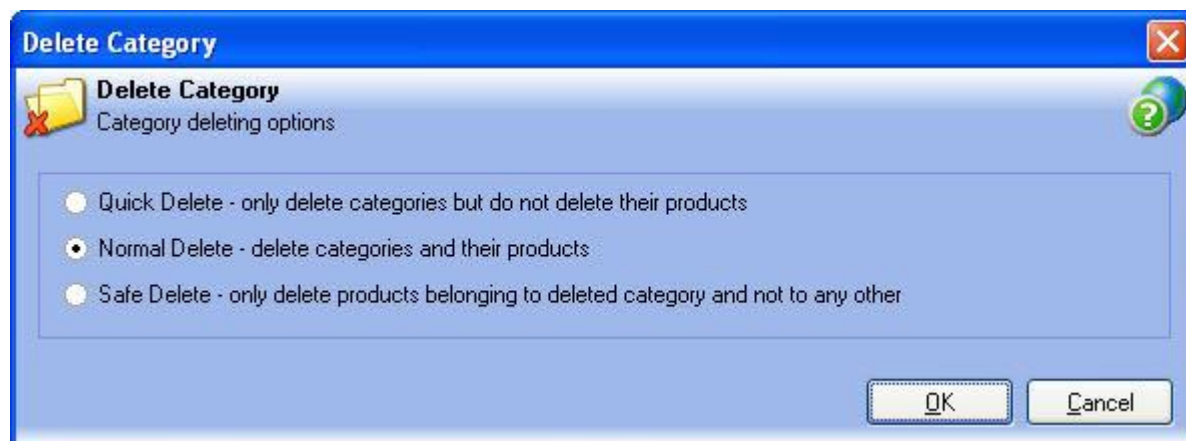



Рисунок 1.5 – Витирання категорії товарів

Вибір першого способу означає переміщення всіх продуктів до батьківської категорії. Друга опція дозволяє витерти категорію та всі її товари. Третя – безпечне видалення категорії. Використовується, коли є товари, що належать декільком категоріям. Означає, що коли товар належить декільком категоріям, то знищується лише категорія і товари з неї не зникають, а залишаються бути належними до інших.

2. Розширене управління (експорт та імпорт даних, менеджмент дерева категорій).

Операції експорту/імпорту виконуються за допомогою майстра, який викликається натисканням кнопки . Імпорт/експорт категорій та імпорт/експорт товарів відрізняються.

Імпорт категорій є досить складним процесом, тому рекомендується створити резервну копію бази та почати операцію імпорту. Імпорт відбувається з *.csv файлу, який вибирається на першій сторінці майстра. На наступній сторінці доступний попередній перегляд файлу з даними. Тут варто звернути увагу на

символи-розділювачі значень та символи для виділення кожного значення в рядку, які встановлюються на наступній торінці майстра (рисунок 1.6).

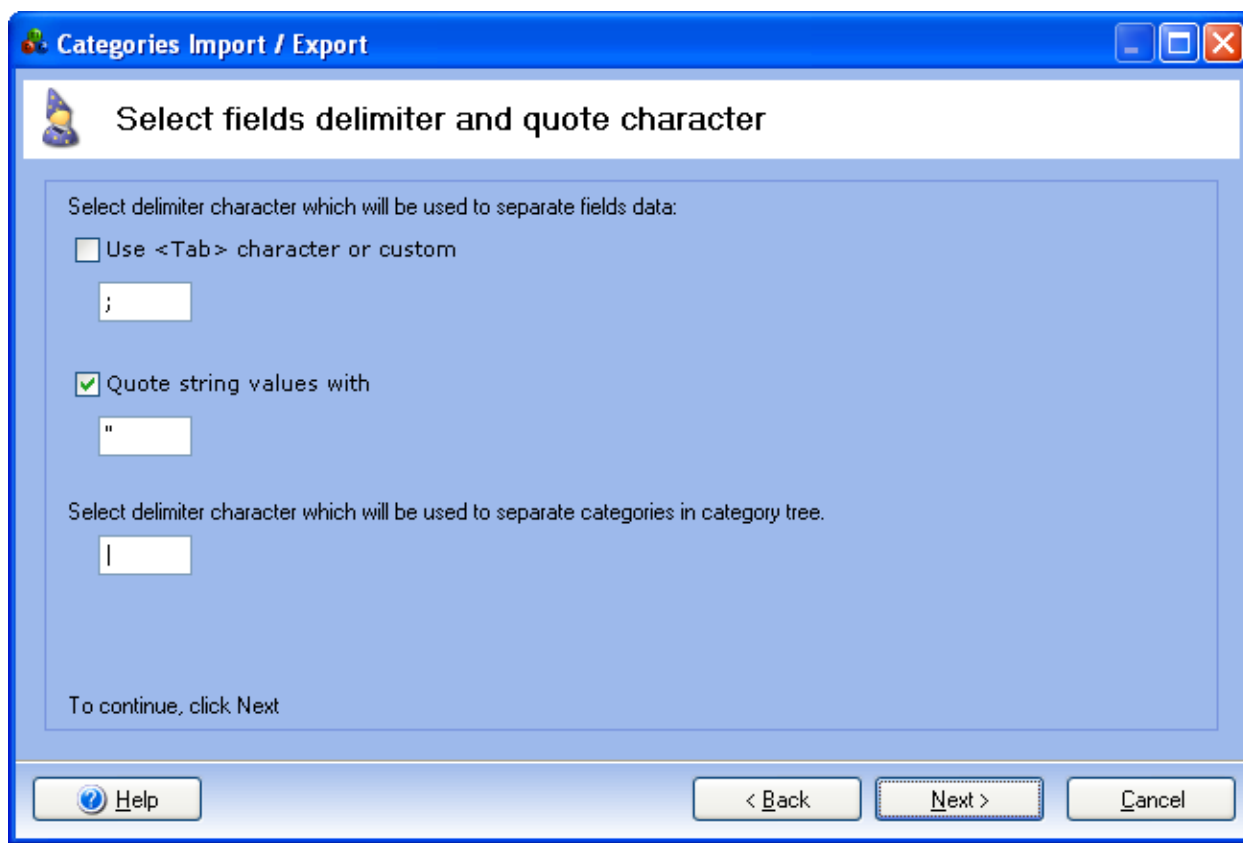


Рисунок 1.6 – Встановлення символів для розділення та виділення значень в *.csv-файлі

Перша опція – символ-розділювач. Найпоширенішим для цього є символ коми, крапки з комою чи символ табуляції. При встановці неправильного символу, очевидно, файл буде прочитано некоректно. Третій параметр – вибір символу для розділення категорій в їх дереві.

На наступній сторінці майстра встановлюється відповідність між стовпцями імпортованого файлу та полями бази даних (рисунок 1.7). Доступні поля вибираються зі списку зліва.

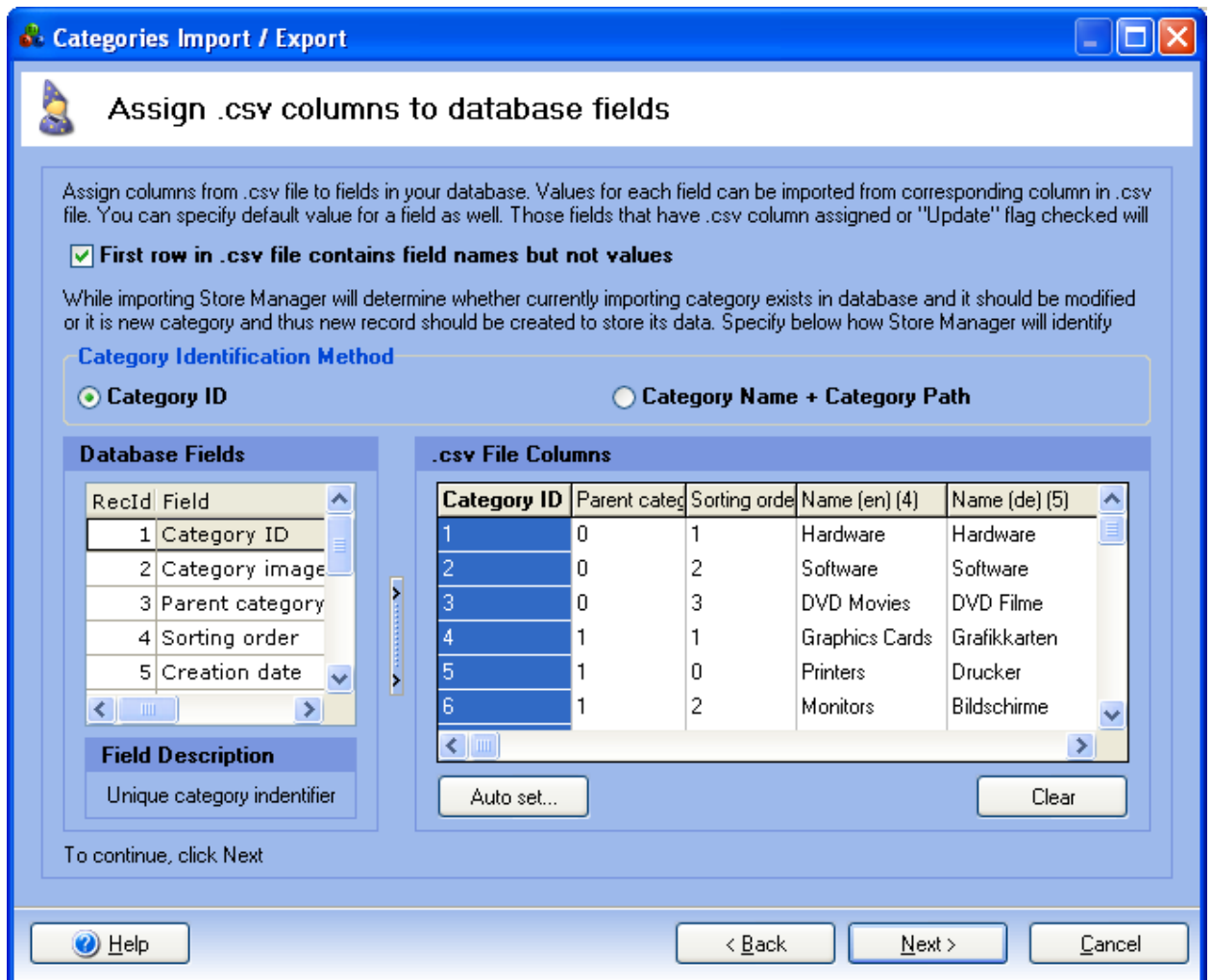




Рисунок 1.7 – Вікно встановлення відповідності між стовпцями файлу та полями бази даних

На наступній сторінці вибирається форматдати та часу. Після імпорту користувач може переглянути помилки, що виникли. Найчастіша їх причина – некоректне встановлення параметрів формату дати та часу, а, отже, і неправильний подальший імпорт цих значень.

Після виправлення помилок майстер імпорту завершує свою роботу.

Експорт виконується аналогічно до імпорту – за допомогою майстра. Кроки – аналогічні. Потрібно вказати об'єкти для експорту, файл призначення, символи-розділювачі та стовпці бази даних.

Експортований файл може бути відкритий засобами Excel.

Управління деревом категорій включає озкриття всього дерева натисканням кнопки  та згортання дерева  .

1.6.2 Керування товарами

Виконання основних операцій над товарами (створення, редагування, знищення) виконується через панель інструментів (рисунок 1.8).

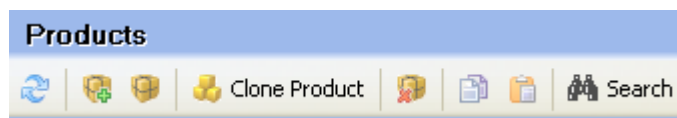


Рисунок 1.8 – Панель інструментів для створення товару

Процес створення та редагування товарів подібний. Вікно для цього показане на рисунку 1.9.

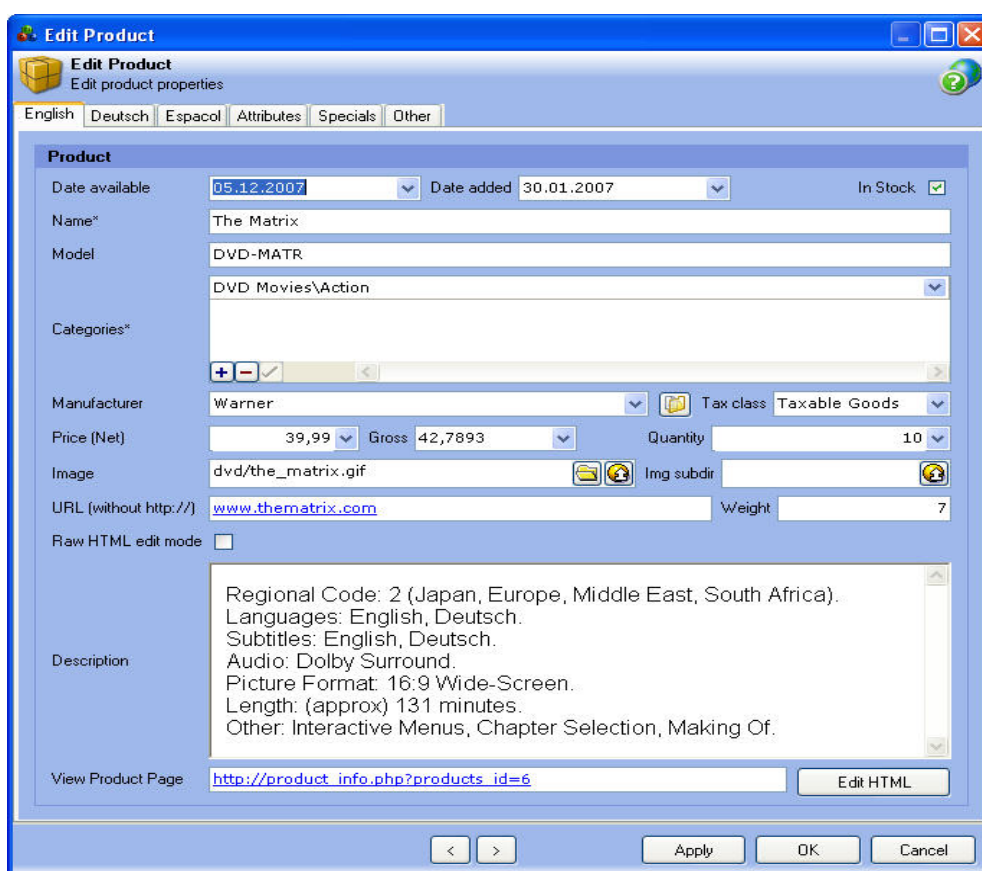
The screenshot shows the "Edit Product" window for "The Matrix". It includes fields for "Date available" (05.12.2007), "Date added" (30.01.2007), "Name" (The Matrix), "Model" (DVD-MATR), "Categories" (DVD Movies\Action), "Manufacturer" (Warner), "Tax class" (Taxable Goods), "Price (Net)" (39,99), "Gross" (42,7893), "Quantity" (10), "Image" (dvd/the_matrix.gif), "URL" (www.thematrix.com), and "Weight" (7). There is also a "Description" field with text about regional codes, languages, subtitles, audio, picture format, and length. The window has tabs for "English", "Deutsch", "Español", "Attributes", "Specials", and "Other".

Рисунок 1.9 – Вікно для редагування відомостей про товар

Для пошуку потрібного товару можна скористатись вікном пошуку, зображеним на рисунку 1.10.

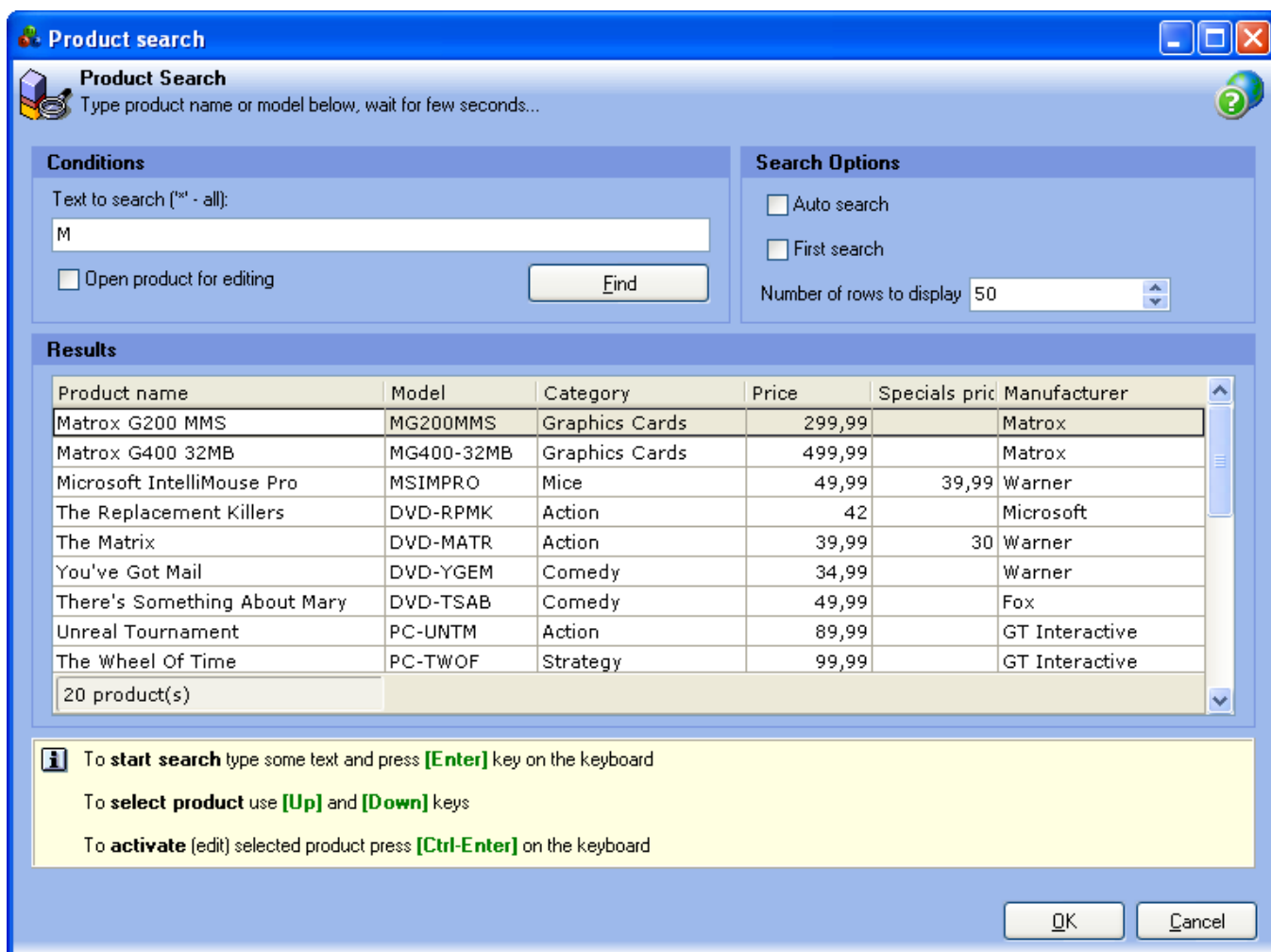


Рисунок 1.10 – Вікно пошуку

При перегляді відомостей про товари доступні групування та фільтрація (рисунок 1.9). Наводити детальнішу інформацію з цих питань тут неварто, оскільки елементи інтерфейсу користувача дозволяють легко виконати ці операції.

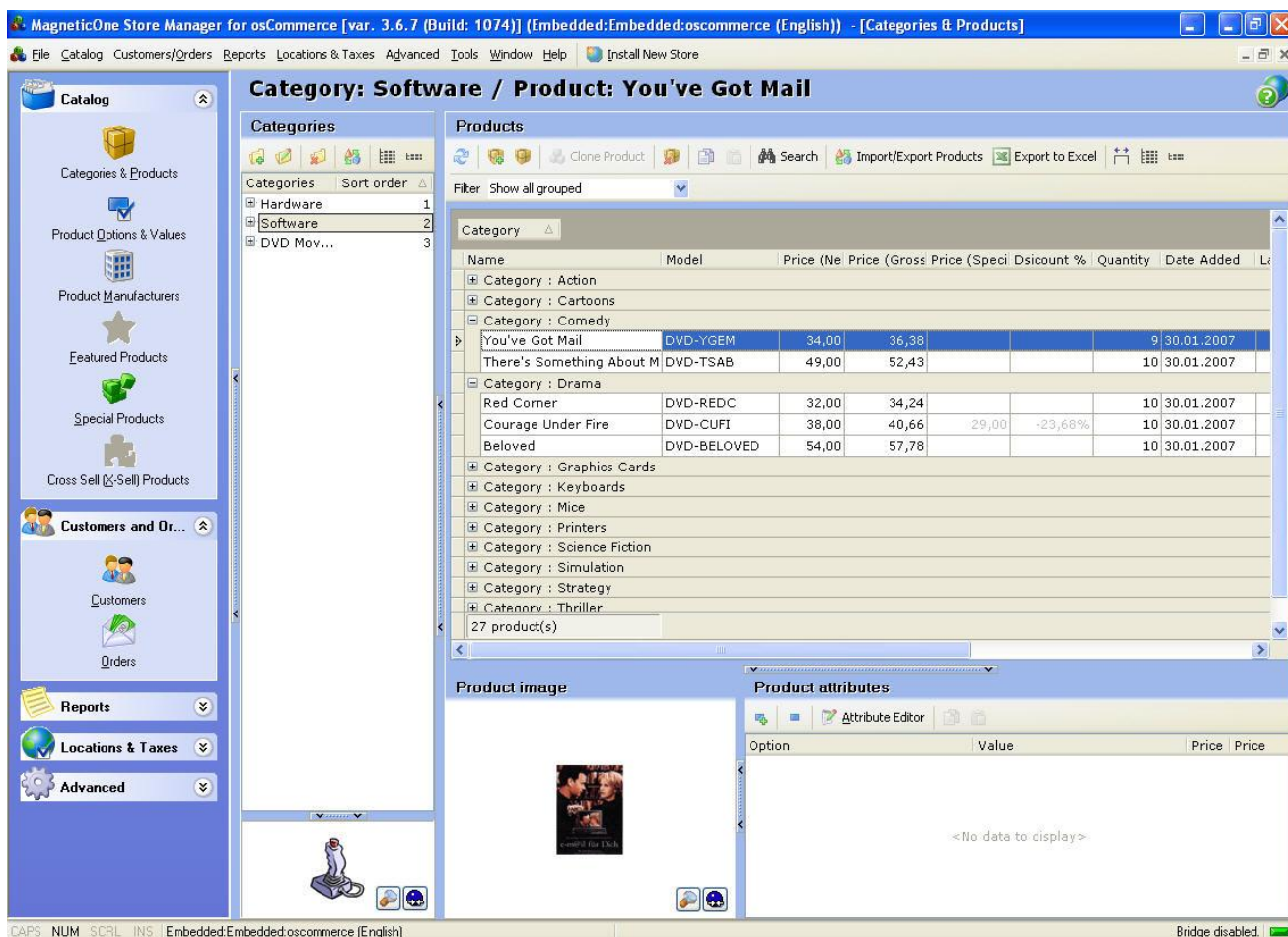


Рисунок 1.11 – Перегляда відомостей про товари

Кожна категорія товарів може бути повністю розгорнута чи згорнута.

1.6.3 Управління клієнтами та замовленнями

Загальний вигляд вікна для редагування відомостей про клієнтів (покупців) наведено на рисунку 1.12.

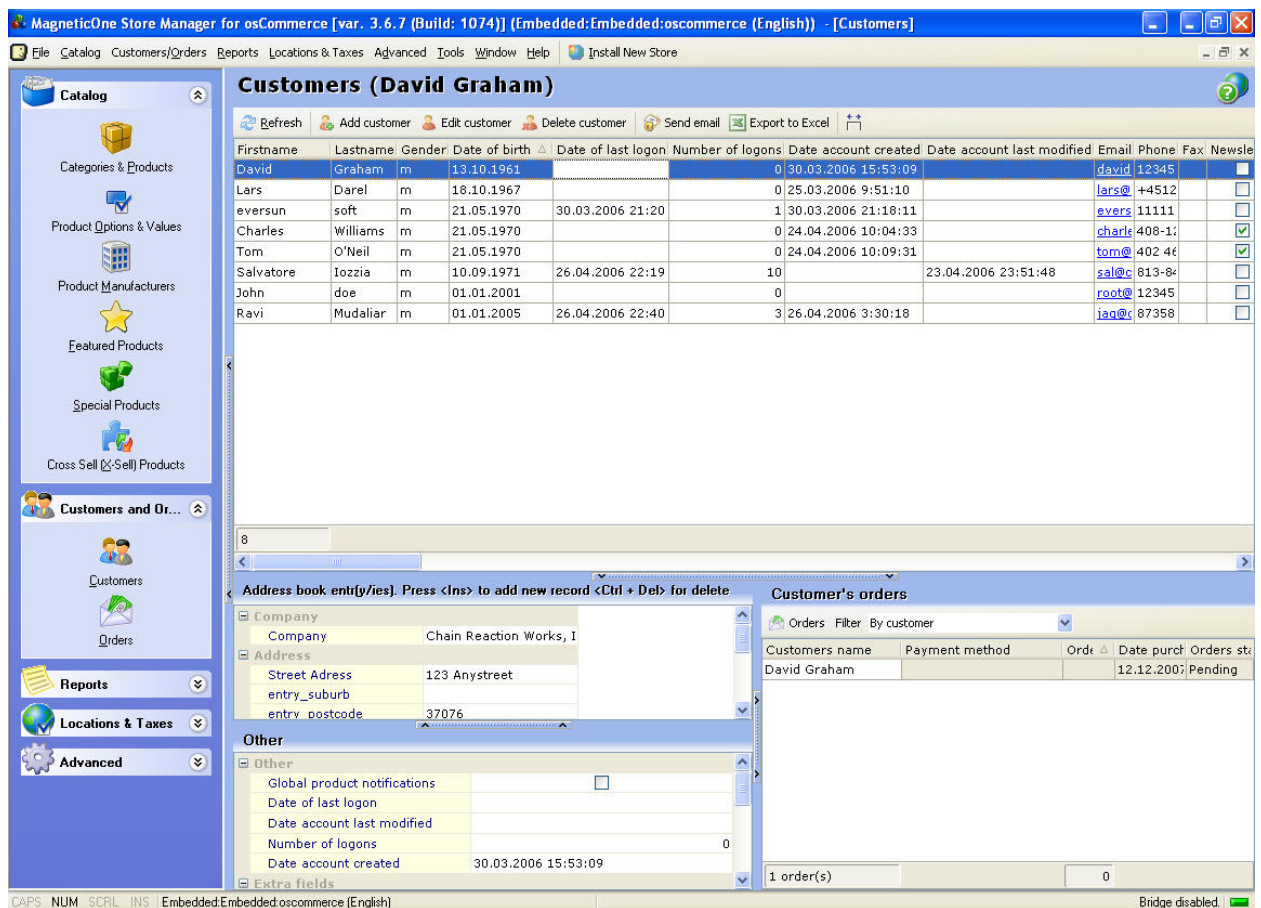


Рисунок 1.12 – Редагування відомостей про клієнта

Вибір замовника дає доступ до виконання операцій редагування та перегляду замовлень, які він зробив у магазині.

При редагуванні відомостей про замовлення (рисунок 1.13) використовується діалогове вікно з відомостями про покупця, спосіб оплати та на другій закладці – дані про замовлення (список товарів з вказанням ціни та кількості).

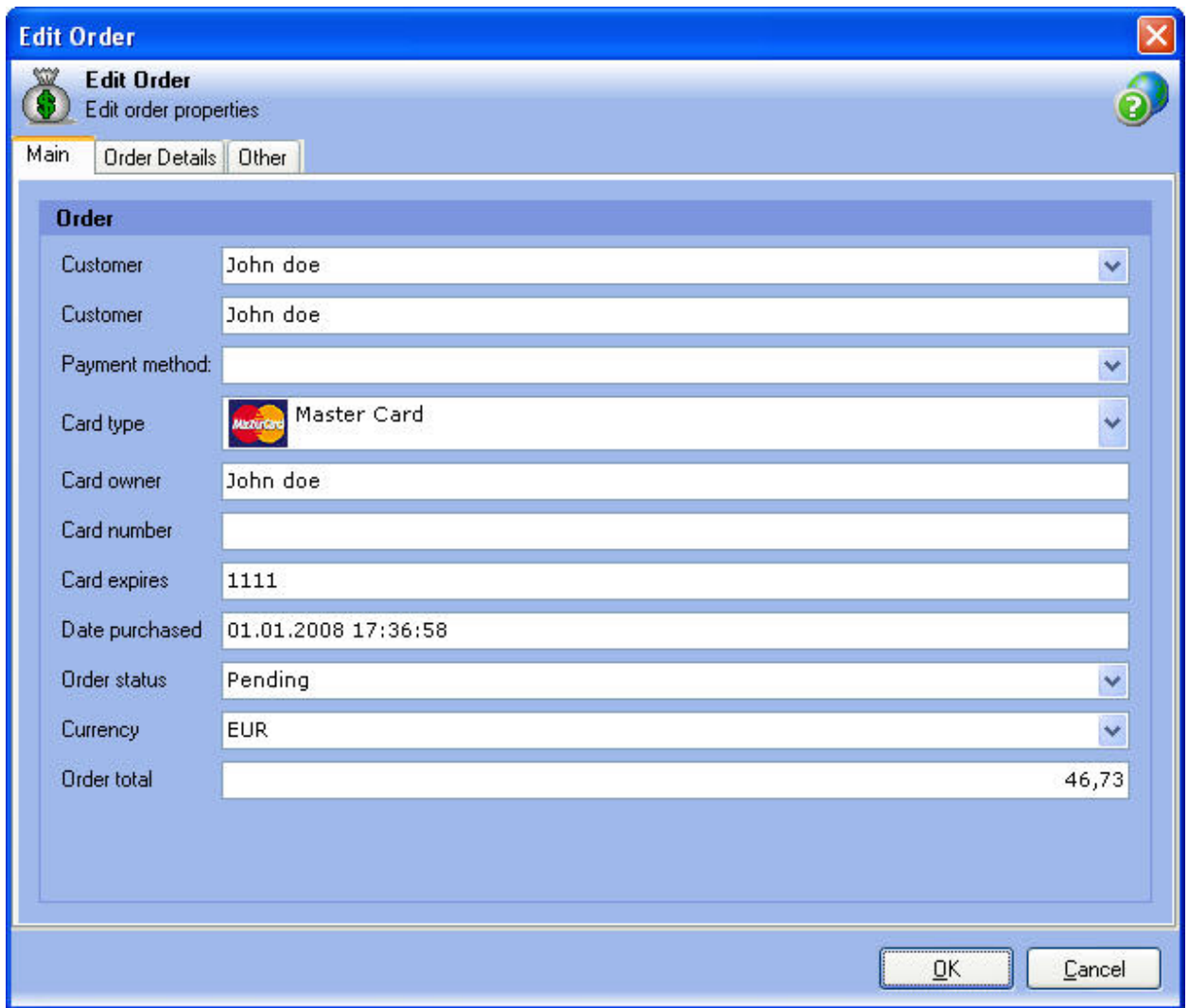


Рисунок 1.13 – Вікно редагування даних про

Для замовлення замовлень доступні операції пошуку, фільтрації сортування тощо по аналогії до набору доступних операцій для тоаврів та категорій.

1.6.4 Формування звітів

Секція формування звітів призначена для автоматичного формування звітних даних погодинно, щоденно, потижнево, помісячно та/або щорічно.

В програмі доступні два види звітів.

1. Стандартні.
2. Звіти користувача.

Ці типи майже повністю ідентичні, за винятком того, що звіти користувача можуть бути змінені засобами Store Manager.

Слід зауважити, що формування будь-якого звіту виконується з даних, отриманих шляхом виконання SQL-запитів, котрі розміщені в спеціальному каталозі програми. Тому створення нового звіту означає створення нового запиту.

2 ПРАКТИЧНА РЕАЛІЗАЦІЯ

2.1 Архітектура Інтернет-магазину

Архітектура типового Інтернет-магазину складається з трьох частин:

- програмна;
- адміністраторська;
- клієнтська.

Програмна архітектура продукту розглядається як взаємозв'язок клієнтської і серверної частини.

У клієнтській частині створюється сам сайт, як набір інструментів інтерфейсу користувача , а також доступ до бази даних.

Серверна частина передбачає розгортання програмного забезпечення на стороні хостинг-провайдера.

2.2 Принципи та засоби створення Web-сайтів

2.2.1 Огляд технологій створення сайту

Існує декілька технологій створення сайту: статичні html-сторінки, SSI-технологія, CGI-технологія, «Site engine» і технології використовуючі XML, PHP+SQL. Технології застосовуються в залежності від призначення, від особливостей функціонування сайту. Відповідно до цього і вибирається технологія реалізації проекту.

Якщо використовувати технологію статичних html-сторінок, то можна створити за тиждень в програмі FrontPage, DreamWeaver, Word або веб-редакторів хостингу. Створюється десяток сторінок, одна з яких називається Головною, а децю інших є зміст основних рубрик проекту. Потім всі вони зв'язуються мережею гіперпосилань, на них безладно вішаються картинки і кнопки, вибирається фон - і сайт готовий до запуску - без єдиної строчки коду, написаної самим оператором.

Як правило за допомогою цієї технології створюються особисті. домашні сайти, які створюються недосвідченими користувачами.

Недоліком такої технології є те, що при необхідності відновити сайт або додати декілька сторінок доводиться наново переробляти важ сайт. Доводиться набрати сторінку в html-форматі, помістити її на сервер, дати анотоване посилання із змісту рубрики, в якій сторінка публікується, повідомити про появу матеріалу на Головній сторінці, в розділі «Новини», змінити інші зв'язані сторінки (встановивши перехресні посилання), при цьому всі сторінки потрібно знову закатати на сервер, всі посилання перевірити на правильність, а при знаходженні помилки – виправити обидва варіанти сторінки, і той, що на сервері, і той, який на домашньому комп'ютері. Публікації і оновлення матеріалів, при використанні такої технології, дуже ускладнені. Ці ж кроки доводиться робити і при зміні дизайну сайту.

На допомогу приходить технологія з використанням SSI, розшифровується як Server Side Include, тобто включення на стороні сервера, і дозволяє збирати сторінку з окремих шматочків, підключаючи до одних файлів інші. Тобто будь-яка сторінка на сайті є «листочковим пирогом» - текст матеріалу виявляється затиснутий між верхнім і нижнім колонтитулом, причому останні однакові на всіх сторінках і по цьому їх можна достатньо легко винести в окремі файли, а потім підключати SSI-інструкціями. За допомогою текстового редактора (візуальні вже непридатні, оскільки вони вважають за краще працювати з цілісними html-документами, а не з «SSI-нарізкою»), розрізають сторінки сайту, замінюється код, що відноситься до дизайну (ті самі колонтитули), на команди виду `<-include virtual=«/top.txt»->`. Реалізується ідея розділення змісту і форми: у файлах залишається тільки контент, а дизайн настроюється через зміну SSI-вставок. Таким чином легко змінити оформлення і навігацію на всьому сайті. І верстка сторінок стає помітно полегшується - не доводиться плутатися в коді таблиць і осередків, досить замінити в звичайному txt-файлі символи нового рядка на тег `<p>` і додати в початок і кінець файлу інструкції на вставку верхніх і нижніх колонтитулів.

Недолік технології в тому, що при публікації нових матеріалів все одно половину сайту доводиться міняти уручну і перезакатувати на сервер.

Щоб на сайт додати інтерактивності, то доцільніше використовувати технологію CGI-скриптів. Скрипти є програми, які запускаються і працюють на стороні сервера, взаємодіючи з користувачем через браузер. Інформація вводиться в такий скрипт шляхом заповнення веб-форми, обробляється їм, а потім результат виводиться відвідувачеві у вигляді динамічної сторінки. Це досить гнучкий і універсальний інструмент, за допомогою якого автоматизуються рутинні дії. За допомогою Perl і пишеться програма, яка запрошує через веб-форму параметри матеріалу: рубрику, заголовок, ім'я автора, сам текст, анотацію і т. д., потім записує введену інформацію у вигляді, що вже верстає, в новий файл, проставляє на нього посилання в інших файлах і робить решту всієї рутинної роботи. З допомогою CGI можна робити ще багато хороших речей - починаючи від вставки в сторінки, наприклад, випадкового посилання з колекції і закінчуючи гостьовою книгою або навіть банерною мережею. Зв'язка «SSI+CGI» може працювати довго і успішно.

Недолік такої технології: будь-яка зміна, окрім публікації, не автоматизована, оскільки зробити це з технологією зберігання інформації в статичних html-файлах принципово неможливо.

Технологія «Site engine», або, у вітчизняній інтерпретації, «движок сайту». У цій технології змінюється внутрішнє представлення інформації. Тут інформація представляється у вигляді файлової системи. Працює вона швидше, ніж будь-яка інша, до того ж її деревовидна структура підходить для проекту як не можна краще: рубрика асоціюється з якимсь каталогом, опубліковані матеріали - з файлами, що містяться в нім, а дрібніші розділи - з підкаталогами. Кожен файл може мати простий формат: поля розділені яким-небудь символом, - їх легко змінювати в текстовому редакторі. Сам текст пишеться як завжди. Використовується скрипт з ім'ям engine.cgi. За допомогою його з файлів такого формату формуються звичайні веб-сторінки - але з автоматичною навігацією по рубриках/каталогах, з гарантовано правильними посиланнями, заданим оформленням і інтерактивними вставками. При цьому оновлення сайту буде швидше. Для зручності управління сайтом в цій технології використовується веб файл-менеджер, відповідний під конкретний движок, управляти сайтом при використанні такої технології легше.

Недолік даної технології виявляється, коли потрібно в черговий раз змінити дизайн. Причому змінити дрібний, локальний дизайн, який на відміну від глобального «декору», глибоко зашитий в код (оператори виведення даних дуже тісно переплітаються з іншими), тому простіше переписати скрипт наново, чим змінювати його. При появі нових вимог (наприклад, треба щоб на головній сторінці відображався початок останнього опублікованого матеріалу і найпопулярніших тим форуму) потрібно сортувати всі файли (ігноруючи їх розділення по каталогах) по даті останнього оновлення або кількості коментарів до повідомлення. Скрипт стає все більш складним і повільним. В результаті завантаження процесора сервера швидко збільшується.

Для вирішення проблеми зміни дизайну, незалежно від контенту сайту використовується технологія XML (eXtensible Markup Language, розширювана мова розмітки) спільно з XSLT (eXtensible Stylesheet Language for Transformations, розширювана мова стилів для трансформацій). Перший є універсальним засобом запису структурованої інформації. По вигляду він вельми схожий на HTML, а в реальності споріднені зв'язки між цими мовами досить заплутані, хоча є і загальний предок - SGML. Ну а XSLT – це мова перетворень документів з одного «діалекту» XML в іншій. На виході скрипта движка виходить XML-текст, що не містить ніяких конкретних інструкцій по його оформленню, а потім, використовуючи XSLT-заміни, перетвориться в звичайний html.

Не дивлячись на те що технології, які використовуються в цьому випадку, дозволяють майже повністю контролювати і змінювати зовнішній вигляд документа, не модифікуючи сам скрипт движка (у набагато ширших межах, ніж шаблонний підхід), він має істотний недолік: або дизайнера треба учити XSLT, або писати відповідні інструкції по перетворенню XML в HTML. І те і інше задоволене проблематично.

Технологія використання зв'язки «PHP+SQL». Недолік цієї технології в тому, що швидкість виконання запитів зазвичай не дуже велика, доводиться використовувати методи оптимізації – кешувати отриману інформацію в локальних файлах. При цьому технологія «PHP+MySQL» вимоглива до ресурсів хостингу.

Вибір технології залежить від масштабу проекту - для створення домашньої сторінки достатньо звичайних статичних файлів, а для невеликого контент-ресурса підійде зв'язка «CGI+SSI». Якщо є потреба в зберіганні і аналізі даних краще скористатися зв'язкою «PHP+SQL», при цьому можна суміщати декілька технологій при створенні сайту.

2.2.2 Огляд інструментів для створення сайту

Існує велика кількість методів та інструментів для створення Інтернет-сторінок. Можна почати з готового шаблону сайту або створити його за допомогою візуального редактора. Це найпростіший вихід. Треба тільки витратити небагато час на виконання елементарних правил настройки. Безкоштовні хостинги надають готові шаблони сайтів. Як правило такі шаблони не дуже зручні в настройці і у них готовий дизайн не завжди хорошої якості, і може не відповідати задуму. Є і довершені, сучасні шаблони, що включають масу сервісів, аж до власних сайтів. Там багато настройок і параметрів можна змінювати, відповідно до своїх потреб.

Але є і недоліки. По-перше, сайт винен обов'язково знаходитися на платному хостингу, тому що для роботи шаблону необхідна підтримка серверних технологій. Більшість безкоштовних хостингів цього не підтримують. У других, необхідно мати солідний досвід установки і настройки складного шаблону.

Таким чином шаблони годяться для тих, кому достатньо простенької домашньої сторінки, або для тих, хто вже набув певного досвіду в сайтостроєнні.

Ще один метод створення сайту за допомогою редакторів. До одних з найбільш поширених відносяться візуальні редактори FrontPage і DreamWeaver.

Створювати сайт з їх допомогою досить просто. Для цього не обов'язкове знання мов програмування. Перевага цього методу (в порівнянні з використанням готового шаблону) в тому, що сторінку створюєте самі, від початку до кінця. Проте кількість тегів більше, чим при верстці в ручну. Через це підвищується час завантаження сторінки, що безпосередньо позначається на відвідуваності сайту.

Недоліком їх є, в першу чергу, величезна надмірність кодів, внаслідок чого виходять великі сторінки, що поволі завантажуються.

Для простих веб-сторінок написання коду вручну займає стільки ж часу, скільки вивчення веб-редакторів. Для написання HTML коду можна користуватися і «Блокнотом», тоді код виходить оптимальним, таким, що не містить зайві теги.

Існують програми, які дозволяють підсвічувати код, вставляти заготовлені стандартні блоки коду, дозволяють проглянути сторінку в готовому вигляді, одна з таких програм HtmlPad Fisher Man.

Також необхідно відзначити програми для оптимізації HTML коду. Ці програми видаляють з коду зайві пропуски, теги, замінюють їх на простіші, що значно зменшує розмір веб-сторінки. Прикладом таких програм можуть служити HTML-Compress, HTML Optimizer. Після обробки HTML коду однієї з цих програм розмір сторінки зменшується в середньому на 25%, тільки HTML-код читати стає не дуже зручно.

В результаті огляду засобів можна зробити висновок, що для створення сайту краще всього скористатися Html редактором Html Pad Fisher Man, а для

розробки графіки можна скористатися декількома редакторами: Photoshop, GIF Animator і ін.

2.3 Концептуальне проектування сайту

Основною метою створення віртуальних магазинів є продаж товарів і отримання прибутку, тому структура, дизайн та наповнення ресурсу мають бути такими, щоб зацікавити потенційного покупця, полегшити вибір необхідного йому товару.

На основі викладених міркувань створюємо структуру сайту, яку показано у вигляді структурної схеми на рисунку 2.1.

Окрім візуальної частини ресурсу та постійних інформаційних даних, для збереження і обробки інформації про товари і трансакції необхідна наявність бази даних. Дана БД має містити дані про товари, користувачів та замовлення, оскільки вони відносяться до змінної інформації ресурсу.

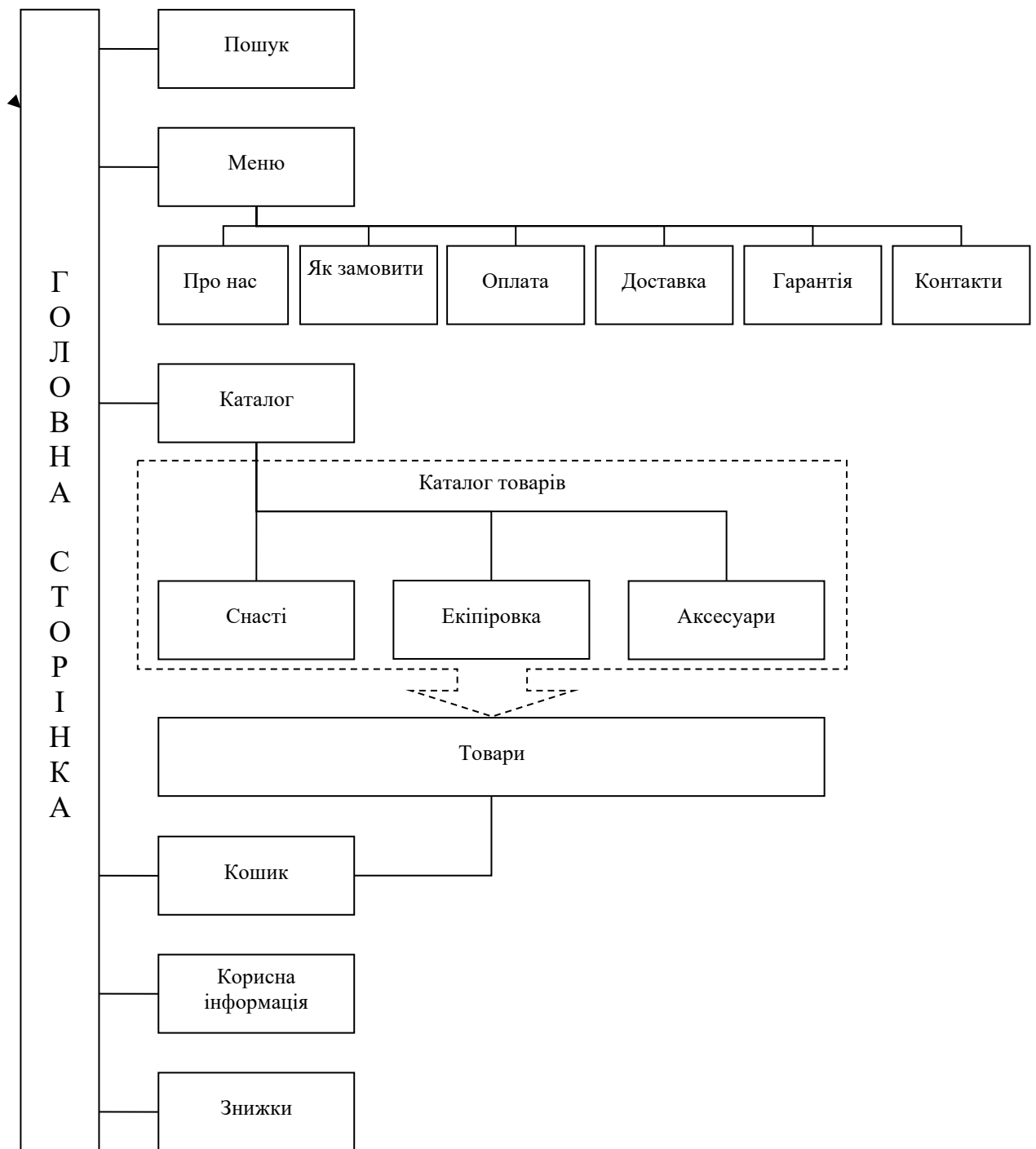


Рисунок 2.1 – Структурна схема Інтернет-магазину

2.4 Фізична реалізація магазину

2.4.1 Вибір інструментів для створення дизайну сайту

Візуальні редактори (візуальні середовища розробки) дозволяють програмісту працювати не з програмним кодом, а безпосередньо з візуальними

образами елементів інтерфейсу. Для позначення технології, яка основана на використанні візуальних редакторів, використовується термін WYSIWYG (What You See Is What You Get – „що ви бачите те і отримуєте”).

Для розробки фізичної реалізації використовувалися декілька редакторів для створення Web-документів, але основним інструментом був редактор DreamWeaver, оскільки він дозволяє одночасно працювати як з кодом сторінки так і з її візуальним представленням, що є досить зручним при розробці різних елементів сайту.

Графічні елементи розроблялися за допомогою редактора Adobe Photoshop, який призначено для редагування і створення растрової графіки. Програма використовується для роботи з фотографіями, слайдами та малюнками для Web-сторінок.

2.4.2 Опис роботи та інтерфейсу Інтернет-магазину

Після відкриття початкового файлу `index.php` відкривається головна сторінка сайту, показана на рисунку 2.2.

Подібну структуру мають і решта сторінок.

У верхній частині сторінки міститься „шапка” сайту. В ній, поряд з графічними елементами дизайну, відображено назву магазину, пошук та сторінку входу користувача, якщо авторизації користувача не було, або кнопку виходу з профілю, якщо користувача авторизовано.

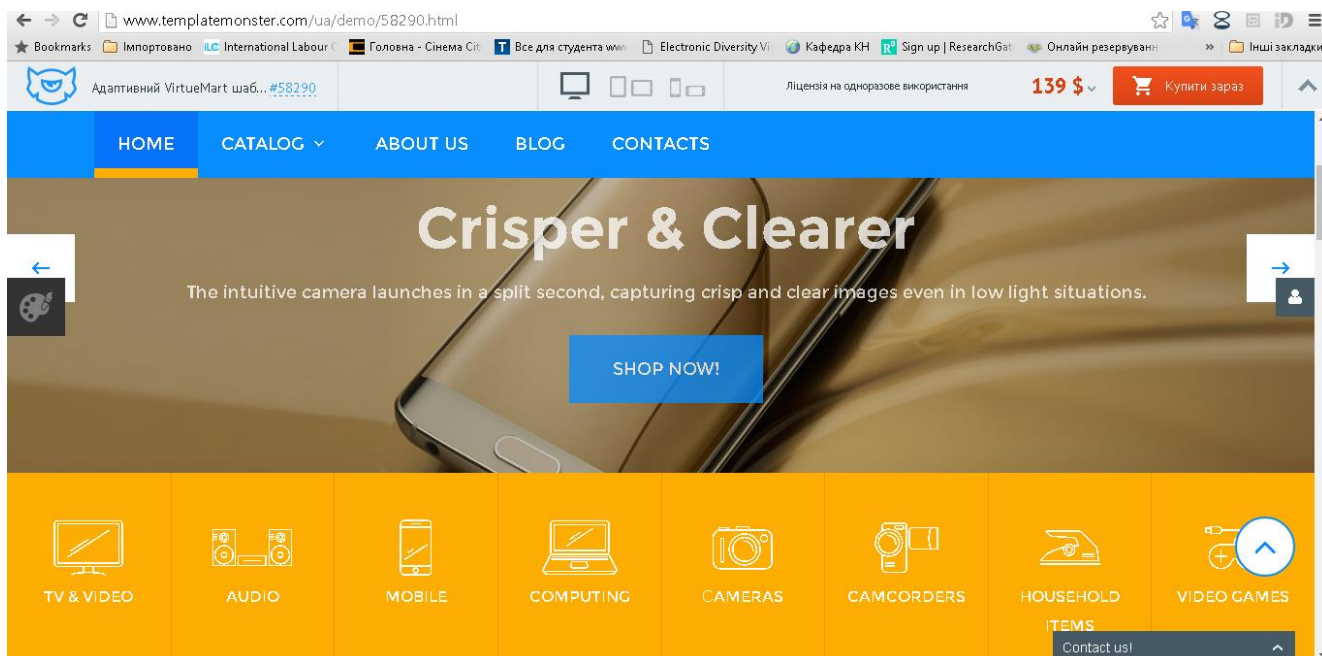


Рисунок 2.2 – Головна сторінка сайту

Нижче розміщено меню сайту згідно із структурною схемою. Типовий вигляд інформативного вікна категорії показано на рисунку 2.2. По боках сторінки розміщено бокові панелі навігації, а в центрі інформаційна частина сторінок сайту. На головній сторінці тут розміщуються випадкові товари з бази даних.

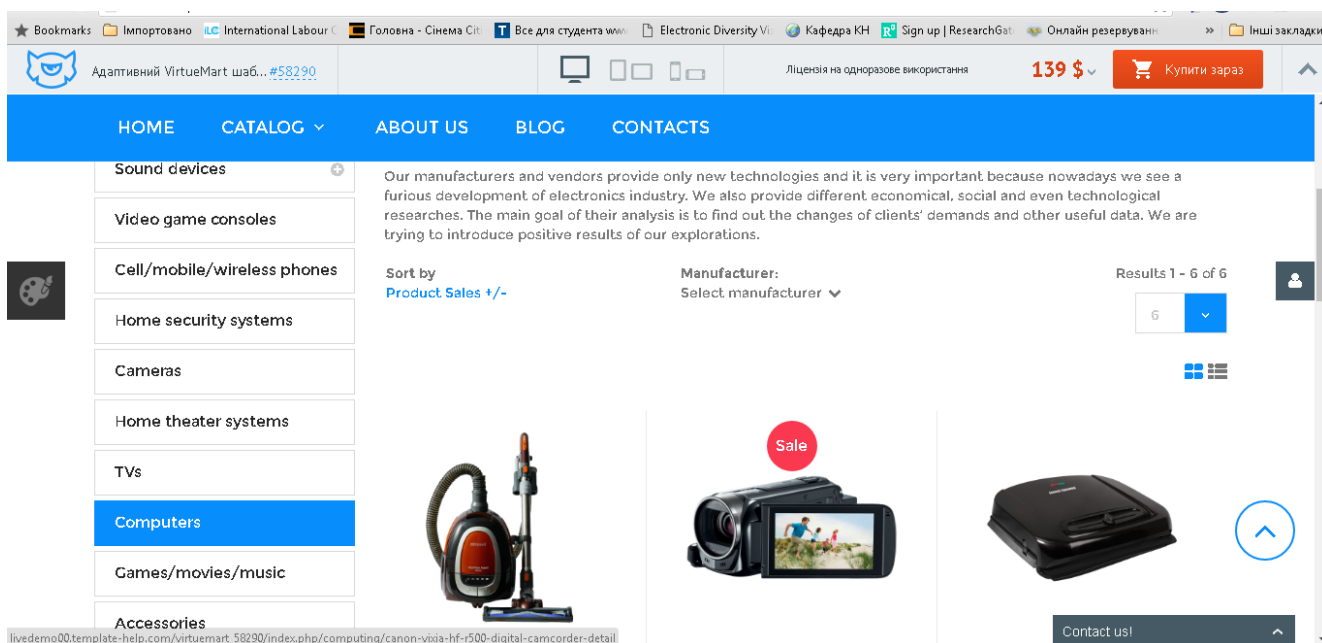


Рисунок 2.3 – Сторінка категорії товарів

Ліва панель навігації розпочинається, посиланнями на розділи та категорії каталогу товарів віртуального магазину. При виборі категорії відкривається сторінка показана на рисунку 2.3. Тут в інформаційній частині відображаються товари обраної категорії, з їх зображенням, та ціною, присутня також можливість сортування товарів за назвою, наявністю, ціною та виробником, якого можна обрати за допомогою випадального списку.

В верхній частині сайту розміщене відображення стану кошику покупця. В ньому відображаються назви обраних товарів. Якщо користувач не авторизований або товари не обрано, то тут міститься надпис про те, що кошик порожній.

Під кошиком розміщено вікно «Корисна інформація» із посиланнями із різноманітними посиланнями. Нижче розміщено вікно, де випадковим відображується один із товар з бази даних, який має позначку «Акція».

2.4.3 Здійснення покупок

Представлення інформації про товар має вигляд, показаний на рисунку 2.4.

На даній сторінці відображається запит із БД про конкретну позицію каталогу: зображення товару, його характеристики ціну. Також користувач може вибрати закладку „Відгуки”, те зможе переглянути та лишити власні оцінки та відгуки про обраний товар. Нові відгуки заносяться у відповідну таблицю БД і перевіряються адміністратором в режимі постмодерації.

Якщо інформація про товар задовольняє користувача і він хоче його придбати, то йому необхідно натиснути кнопку „Додати у кошик”, після чого товар заносяться в кошик користувача.

Слід зауважити, що при відправленні товару в кошик користувачеві буде запропоновано пройти авторизацію чи реєстрацію на сайті, якщо це не зроблено раніше через відповідну форму входу у верхній частині сторінки.

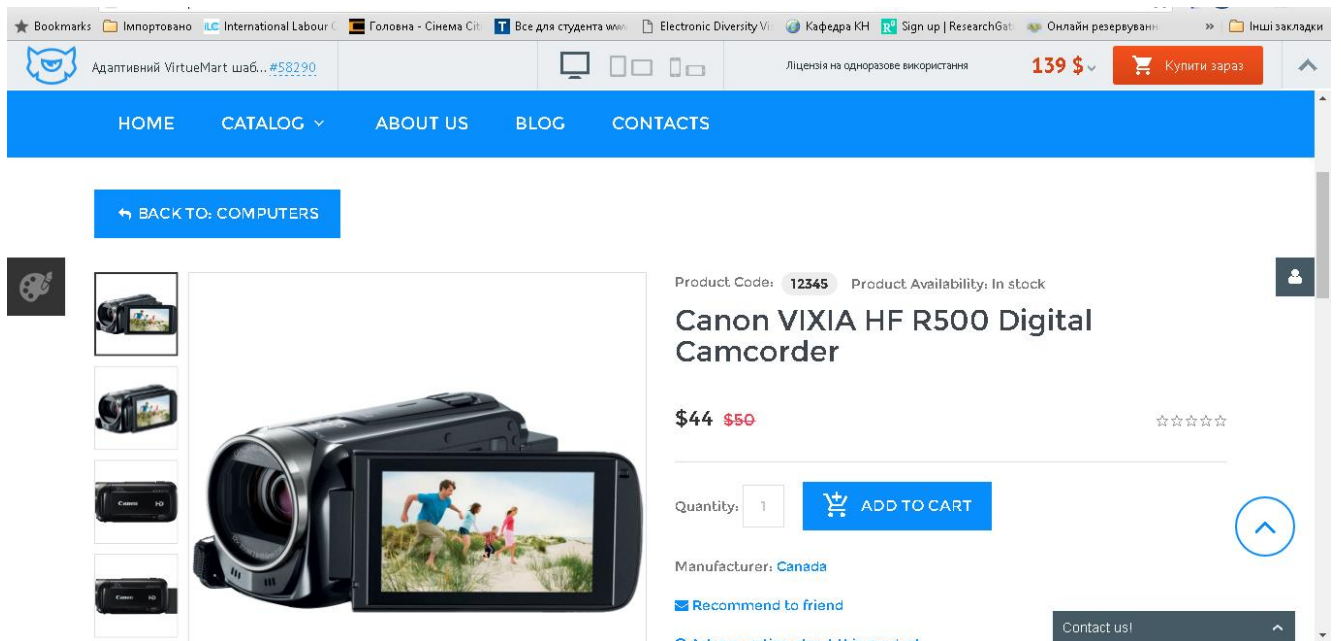


Рисунок 2.4 – Представлення інформації про товар

Вказані реєстраційні дані будуть використані при замовленні товару.

Переглянути вміст кошику у скороченому вигляді можна у відповідному вікні правої верхньої панелі. Повну інформацію про обрані товари можна переглянути на сторінці „Кошик”, зображеній на рисунку 2.5, натиснувши відповідну кнопку в віконця.

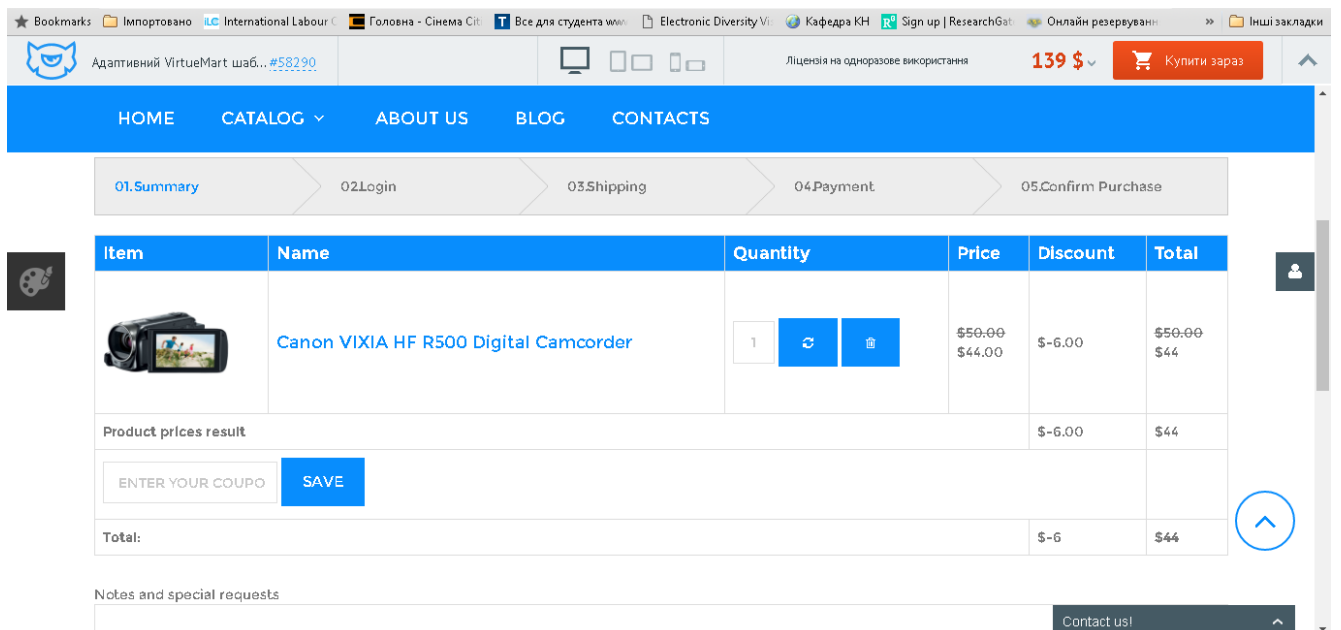


Рисунок 2.5 – Форма реєстрації користувача

На даній сторінці розміщено зображення обраних товарів, їх назви та ціни. Нижче наведено повну вартість можливого замовлення. Для того, щоб видалити товар з кошику необхідно поставити маркер, що розміщено біля кожної позиції кошику та натиснути кнопку «Видалити». Також у замовлення можна вносити зміни стосовно кількості кожної одиниці обраного товару, після чого необхідно натиснути кнопку «перерахувати замовлення».

Для здійснення замовлення треба натиснути кнопку „Оформити замовлення” на сторінці кошику.

Після перевірки і підтвердження замовлення додається в БД (слід зауважити що запис в БД створюється окремо для кожного товару в кошику), а на електронну адресу покупця, вказану при реєстрації відсилається лист з повною інформацією про замовлення. Це повідомлення є підтвердженням того, що замовлення клієнта прийняте на обробку і підготовку, а також доказом укладення договору між продавцем і покупцем про продаж товарів по зразках на даному сайті. У відправленому листі присутні посилання для підтвердження замовлення або відмові від нього.

Зручнішим і швидшим способом доставки є доставка кур'єром. У деяких містах покупець може отримати своє замовлення в офісі представництва кур'єрської служби і заощадити на вартості доставки кур'єром по місту.

3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Заходи по покращенню питань з охорони праці та техніки безпеки на підприємстві

Успішна профілактика виробничого травматизму та професійної захворюваності можлива лише за умови ретельного вивчення причин їх виникнення. Для полегшення цього завдання прийнято поділяти причини виробничого травматизму і професійної захворюваності на наступні основні групи: організаційні, технічні, санітарно-гігієнічні, психофізіологічні.

Організаційні причини: відсутність або неякісне проведення навчання з питань охорони праці; відсутність контролю; порушення вимог інструкцій, правил, норм, стандартів; невиконання заходів щодо охорони праці; порушення технологічних регламентів, правил експлуатації устаткування, транспортних засобів, інструменту; порушення норм і правил планово-попереджувального ремонту устаткування; недостатній технічний нагляд за небезпечними роботами; використання устаткування, механізмів та інструменту не за призначенням.

Технічні причини: несправність виробничого устаткування, механізмів, інструменту; недосконалість технологічних процесів; конструктивні недоліки устаткування, недосконалість або відсутність захисних загороджень, запобіжних пристроїв, засобів сигналізації та блокування.

Санітарно-гігієнічні причини: підвищений (вище ГДК) вміст в повітрі робочих зон шкідливих речовин; недостатнє чи нераціональне освітлення; підвищені рівні шуму, вібрації; незадовільні мікрокліматичні умови; наявність різноманітних випромінювань вище допустимих значень; порушення правил особистої гігієни.

Психофізіологічні причини: помилкові дії внаслідок втоми працівника через надмірну важкість і напруженість роботи; монотонність праці; хворобливий стан працівника; необережність; невідповідність психофізіологічних чи

антропометричних даних працівника використовуваній техніці чи виконуваній роботі.

Основні заходи щодо попередження та усунення причин виробничого травматизму і професійної захворюваності поділяються на технічні та організаційні.

До технічних заходів належать заходи з виробничої санітарії та техніки безпеки.

Заходи з виробничої санітарії передбачають організаційні, гігієнічні та санітарно-технічні заходи та засоби, що запобігають дії на працюючих шкідливих виробничих факторів. Це створення комфортного мікроклімату шляхом влаштування відповідних систем опалення, вентиляції, кондиціонування повітря; теплоізоляція конструкцій будівлі та технологічного устаткування; заміна шкідливих речовин та матеріалів нешкідливими; герметизація шкідливих процесів; зниження рівнів шуму та вібрації; встановлення раціонального освітлення; забезпечення необхідного режиму праці та відпочинку, санітарного та побутового обслуговування.

Заходи з техніки безпеки передбачають систему організаційних та технічних заходів та засобів, що запобігають дії на працюючих небезпечних виробничих факторів. До них належать: розроблення та впровадження безпечного устаткування; механізація та автоматизація технологічних процесів; використання запобіжних пристосувань, автоматичних блокуючих засобів; правильне та зручне розташування органів керування устаткуванням; розроблення та впровадження систем автоматичного регулювання, контролю та керування технологічними процесами, принципово нових нешкідливих та безпечних технологічних процесів.

До організаційних заходів належать: правильна організація роботи, навчання, контролю та нагляду з охорони праці; дотримання трудового законодавства, міжгалузевих та галузевих нормативних актів про охорону праці; впровадження безпечних методів та наукової організації праці; проведення оглядів, лекційної та наочної агітації і пропаганди з питань охорони праці; організація планово-

попереджувального ремонту устаткування, технічних оглядів та випробувань техніки.

3.2 Розрахунок вентиляції

Під вентиляцією розуміють сукупність заходів та засобів призначених для забезпечення на постійних робочих місцях та зонах обслуговування виробничих приміщень метеорологічних умов та чистоти повітряного середовища, що відповідають гігієнічним та технічним вимогам. Основне завдання вентиляції – вилучити із приміщення забруднене або нагріте повітря та подати свіже.

3.2.1 Природна вентиляція

Природна вентиляція відбувається в результаті теплового та вітрового напору. Тепловий напір обумовлений різницею температур, а значить і густини внутрішнього і зовнішнього повітря. Вітровий напір обумовлений тим, що при обдуванні вітром будівлі, з її навітряної сторони утворюється підвищений тиск, а підвітряної – розрідження (див. рис. 3.1).

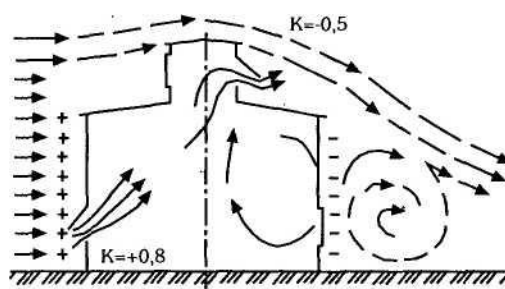


Рисунок 3.1 – Розподіл тиску в будівлі при дії вітру

Перевагою природної вентиляції є її дешевизна та простота експлуатації. Основний її недолік в тому, що повітря надходить в приміщення без попереднього очищення, а видалене відпрацьоване повітря також не очищується і забруднює довкілля.

3.2.2 Штучна вентиляція

Загальнообмінна вентиляція забезпечує створення необхідного мікроклімату та чистоти повітряного середовища у всьому об'ємі робочої зони приміщення. Вона застосовується для видалення надлишкового тепла при відсутності токсичних виділень, а також у випадках, коли характер технологічного процесу та особливості виробничого устаткування виключають можливість використання місцевої витяжної вентиляції.

Розрізняють чотири основні схеми організації повітрообміну при загальнообмінній вентиляції: зверху вниз, зверху вгору, знизу вгору, знизу вниз, як зображено на рисунку 3.2.

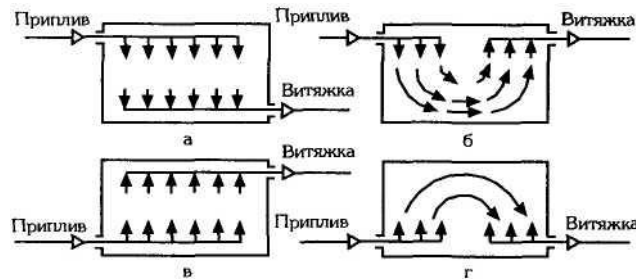


Рисунок 3.2 – Схема організації повітрообміну при загальнообмінній вентиляції

Припливна вентиляція. Схема припливної механічної вентиляції (див. рис. 3.3) включає: повітрязбірний пристрій 1; фільтр для очищення повітря 2; повітрянагрівач (калорифер) 3; вентилятор 5; мережу повітроводів 4 та припливні патрубки з насадками 6. Якщо немає необхідності підігрівати припливне повітря, то його пропускають безпосередньо у виробничі приміщення через обвідний канал 7.

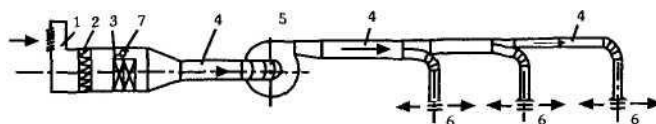


Рисунок 3.3 – Схема припливної вентиляції

Припливне повітря подається в приміщення, як правило, розсіяним потоком для чого використовуються спеціальні насадки.

Витяжна та припливно-витяжна вентиляція. Витяжна вентиляція (див. рис. 3.4) складається із очисного пристрою 1, вентилятора 2, центрального 3 та відсмоктуючих повітроводів 4.

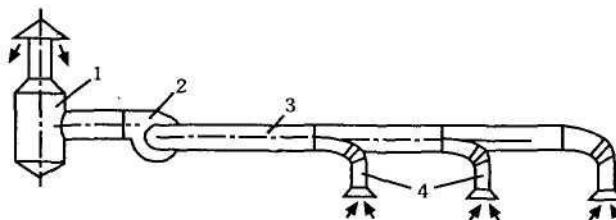


Рисунок 3.4 – Схема витяжної вентиляції

В умовах промислового виробництва найбільш розповсюджена припливно-витяжна система вентиляції із загальним припливом в робочу зону та місцевою витяжкою шкідливих речовин безпосередньо з місць їх утворення.

Місцева витяжна вентиляція здійснюється за допомогою місцевих витяжних зонтів, всмоктуючих панелей, витяжних шаф, бортових відсмоктувачів (див. рис. 3.5).

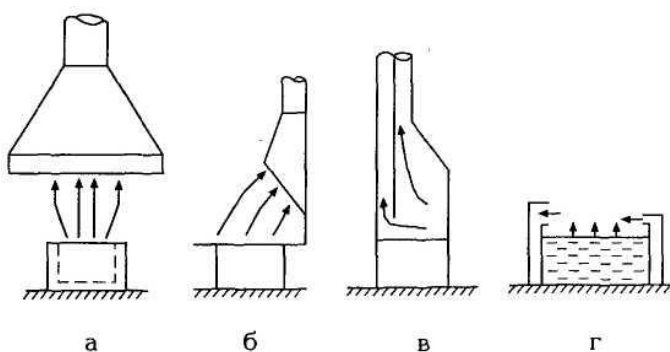


Рисунок 3.5 – Приклади місцевої витяжної вентиляції: а – витяжний зонтик, б – всмоктувальна панель, в – витяжна шафа з комбінованою витяжкою, г – бортовий відсмоктувач з передувом

Основне завдання розрахунку загальнообмінних систем штучної вентиляції – визначити кількість повітря, що необхідно подати і вилучити з приміщення. При розрахунку вентиляції в цехах, повітрообмін, як правило, визначають розрахунковим шляхом за конкретними даними про кількість шкідливих виділень (теплоти, вологи, парів, газів).

Для приміщень, де немає шкідливих виділень (або кількість їх незначна) приплив (витяжку) повітря можна визначити за кратністю повітрообміну (k) – відношенням об'єму вентиляційного повітря L ($\text{м}^3/\text{год}$) до об'єму приміщення V (м^3):

$$k = \frac{L}{V_n} \quad (3.6)$$

Для нашого випадку при розмірах приміщення 3м x 4,5м x 3 м маємо об'єм приміщення $V_n=40,5 \text{ м}^3$. Об'єм повітря на одного працівника при трьох працюючих становить $13,5 \text{ м}^3$.

Для приміщень, де не утворюються шкідливі виділення та надлишкове тепло і немає необхідності у створенні метеорологічного комфорту можна використати формулу:

$$L = l \times n, \quad (4.7)$$

де l – мінімальне подання повітря на одного працівника відповідно до санітарних норм (при об'ємі приміщення, що припадає на одного працівника, до 20 м^3 – $30 \text{ м}^3/\text{год}$, а при об'ємі більше 20 м^3 – $20 \text{ м}^3/\text{год}$);

n – кількість працівників в приміщенні.

Отже, для двох працюючих в приміщенні згідно (3.7) $L=30*2 = 60\text{м}^3$.

Тепер кратність повітрообміну згідно (3.6) становить $60/40,5=1,48$.

Використовуючи спеціальну довідникову літературу, можна встановити для даного значення кратності повітрообміну об'єм припливного повітря.

Для забезпечення кращих і стабільних кліматичних умов можна використати кондиціонер. Пропонується для вказаного приміщення встановити спліт-систему кондиціонування з характеристиками, наведеними в таблиці 3.3.

Таблиця 3.3 – Характеристики обраного кондиціонера

Характеристика	Значення
Кондиціонер Fujitsu ABY18U / AOY18U	
З обігрівом:	так
Висота зовнішнього блоку:	830 см
Ширина зовнішнього блоку:	650 см
Глибина зовнішнього блоку:	320 см
Маса зовнішнього блоку:	52 кг
Висота внутрішнього блоку:	990 см
Ширина внутрішнього блоку:	655 см
Глибина внутрішнього блоку:	199 см
Маса внутрішнього блоку:	28 кг
Продуктивність охолодження:	5400 Вт
Продуктивність нагрівання:	5800 Вт
Споживана потужність - охолодження:	1900 Вт
Споживана потужність - нагрівання:	1940 Вт
Швидкість циркуляції (Куб/хв):	13 Куб/хв
Рівень шуму кімнатний:	36 дБ
Рівень шуму зовнішній:	44 дБ
Пульт ДК:	так
Таймер:	так

Отже, в розділі виконано огляд питань організації заходів з охорони праці під час розробки програмного забезпечення та виконано розрахунок нагнітальної вентиляції.

ВИСНОВОК

В даній роботі проведено розробку Інтернет-магазину по продажу електроніки, використовуючи уміння і навички, набуті під час навчання. Підводячи підсумки виконаної роботи можна зробити наступні висновки. Автором роботи, який був членом команди з розробки інтернет-магазину, було розроблено комплекс програмних модулів, які разом становлять один програмний комплекс, призначений для адміністрування інтернет-магазину і підтримки роботи цього магазину.

Цей комплекс дозволяє адміністратору виконувати весь потрібний набір операцій що управління роботою магазину. Питання адміністрування досить важливе для реальних умов застосування програми. Це основне для Internet-магазину. Дана програма використовується в реальних умовах.

Отже, результаті виконання даної роботи створено програмне забезпечення для віддаленого адміністрування інтернет-магазину. Робота програми забезпечує:

- управління багатьма магазинами;
- управління товарами, категоріями, атрибутами, виробниками та замовленнями;
- швидкий та легкий імпорт/експорт даних в/з програми;
- одночасні маніпуляції цінами багатьох продуктів;
- легка навігація, пошуктоварів, розвинений механізм фільтрації даних;
- швидкий пошук товару, замовлення, клієнта тощо;
- розвинений механізм формування звітів, в т.ч. більше 20 стандартних звітів.

Виконавши даний проект, автор набув досвіду практичного створення подібного програмного забезпечення, створення баз даних під керуванням СКБД MySQL, побудови SQL-запитів на вибірку даних з БД.

В основній частині роботи наведено поняття, класифікацію та організацію діяльності віртуальних магазинів, описано архітектуру Інтернет-магазину,

розглянуто принципи, технології, інструменти та вимоги створення Web-сайтів. Виконано концептуальне проектування сайту, обґрунтовано вибір програмних засобів реалізації роботи, проведено проектування бази даних, виконано фізичну реалізацію віртуального магазину, описано інтерфейс сайту, процес здійснення покупок та проведено тестування сайту.

Також висвітлено питання безпеки під час експлуатації, обслуговування, ремонту та налагодження ЕОМ.

ПЕРЕЛІК ДЖЕРЕЛ

1. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н. Прохоренок. – 3-е изд. – СПб. : БХВ-Петербург, 2010. – 912 с. – (Профессиональное программирование).
2. PHP : справочник / Р.Аллен Вайк, Майкл Дж. Уокер, Роберт Кокс. – К. : ДиаСофт, 2001. – 448 с.
3. PHP 5. Практика создания Web-сайтов / М. Кузнецов, И. Симдянов, С. Голышев. – СПб. : БХВ-Петербург, 2007. – 960 с. эл. опт. диск (CD-ROM).
4. Web-программирование для всех / В. Дунаев. – СПб. : БХВ-Петербург, 2008. – 560 с. – ISBN 978-5-9775-0197-2
5. Архитектура, методы и средства Интернет-технологий / Е. Д. Вязилов. – М. : URSS : КРАСАНД, 2009. – 512 с.
6. Безпека життєдіяльності: Навч. посібник./ За ред. В.Г. Цапка. – 4-те вид., перероб. і доп. – К.: Знання, 2006. – 397 с.
7. Дейт К. Дж. Введение в системы баз данных. –К.; М.; СПб.: Изд. дом "Вильямс", 1999. –848 с.
8. Електронна комерція : підручник / В. Л. Плєскач, Т. Г. Затонацька. – К. : Знання, 2007. – 535 с. – ISBN 966-346-241-8
9. Житецький В.Ц. Охорона праці користувачів комп'ютерів. Навчальний посібник. – Вид. 2-ге, доп. – Львів: Афіша, 2000. – 176с.
10. Конноли Томас, Бегг Каротин, Страчан Анна. Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд.: Пер. с англ. : уч. пос. – М.: Издательский дом "Вильямс", 2000. – 1120 с.
11. Разработка Web-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – 3-е изд. – М. : Вильямс, 2007. – 880 с. эл. опт. диск (CD-ROM).
12. Разработка, дизайн, программирование, тестирование и раскрутка Web-сайта / М. Кэмпбел ; Пер. с англ. Грищук Т. В., Тимаков А.А. – М. : Триумф, 2007. – 480 с. : ил.

13. Самоучитель PHP 5 / Д.Н. Колисниченко. – 2-е изд. – СПб. : Наука и техника, 2005. – 567 с.

14. Техника Web-дизайна для студента / Ю. Едомский. – СПб. : БХВ-Петербург, 2005. – 400 с. : ил.

ДОДАТКИ

Приклади запитів для створення звітів

```

SELECT
    T1.customers_name,
    T1.orders_id,
    SUM(T2.value) AS TotalPurchased,
    T3.symbol_left
FROM
    /*PREFIX*/orders T1
    INNER JOIN /*PREFIX*/orders_total T2 ON (T1.orders_id = T2.orders_id and
T2.class LIKE "%\_total")
    LEFT JOIN /*PREFIX*/currencies T3 ON (T1.currency = T3.code)
GROUP BY
    T1.customers_name
ORDER BY
    TotalPurchased DESC

```

```

SELECT Count(Distinct(T1.customers_name)) as CountCustomers
FROM /*PREFIX*/orders T1
INNER JOIN /*PREFIX*/orders_products T2 ON (T1.orders_id = T2.orders_id)
LEFT JOIN /*PREFIX*/currencies T3 ON (T1.currency = T3.code)

```

```

SELECT
    T1.products_id,
    T1.orders_products_id,
    T1.products_name,
    T1.products_price,
    T1.final_price,
    SUM(T1.products_quantity) AS sum_purchased,
    T1.products_model
FROM /*PREFIX*/orders_products T1
GROUP BY T1.products_id
ORDER BY sum_purchased DESC

```

```

SELECT (Count(DISTINCT products_id) + 0.0) as CountProducts
FROM /*PREFIX*/orders_products

```

```

SELECT
    products_tax,
    products_quantity,
    final_price,
    products_price,
    products_name,
    products_model,
    products_id,
    orders_id,
    orders_products_id
FROM
    /*PREFIX*/orders_products
where
orders_id IN (%s)

```

```

SELECT
    T1.orders_id,
    T1.customers_id,
    T1.customers_name,
    T1.customers_company,
    T1.customers_street_address,
    T1.customers_city,

```

```

T1.customers_postcode,
T1.customers_state,
T1.customers_country,
T1.customers_telephone,
T1.customers_email_address,
T1.date_purchased,
T1.delivery_name,
T1.delivery_company,
T1.delivery_street_address,
T1.delivery_city,
T1.delivery_postcode,
T1.delivery_state,
T1.delivery_country,
T1.billing_name,
T1.billing_company,
T1.billing_street_address,
T1.billing_city,
T1.billing_postcode,
T1.billing_state,
T1.billing_country,
T1.payment_method,
T1.orders_status,
T1.cc_type,
T1.cc_number,
T1.cc_owner,
T1.cc_expires,
T1.currency,
T1.currency_value,
T2.orders_status_name,
T3.symbol_left
FROM
  /*PREFIX*/orders T1
  LEFT JOIN /*PREFIX*/orders_status T2 ON (T1.orders_status = T2.orders_status_id)
And (T2.language_id = :language_id)
  LEFT JOIN /*PREFIX*/currencies T3 ON (T1.currency = T3.code)
where T1.orders_id in (%s)
Order By T1.orders_id

SELECT
  T1.orders_id,
  T1.date_added,
  T1.customer_notified,
  T1.comments,
  T2.orders_status_name,
  T2.language_id
FROM
  /*PREFIX*/orders_status_history T1
  INNER JOIN /*PREFIX*/orders_status T2 ON (T1.orders_status_id =
T2.orders_status_id)
WHERE T1.orders_id IN (%s) and T2.language_id = :language_id

SELECT
  T1.products_model,
  T3.products_name,
  T3.products_viewed,
  T3.products_url,
  T1.products_price,
  T1.products_quantity,
  T4.manufacturers_name,
  Sum(T2.products_quantity) as CountProducts
FROM /*PREFIX*/products T1
LEFT JOIN /*PREFIX*/orders_products T2 ON (T1.products_id = T2.products_id)
INNER JOIN /*PREFIX*/products_description T3 ON ((T1.products_id = T3.products_id)

```



```

and (T3.language_id = :language_id))
LEFT JOIN /*PREFIX*/manufacturers T4 ON (T1.manufacturers_id =
T4.manufacturers_id)
/*Where ((IsNull(T2.products_quantity) = 1) Or (T2.products_quantity = 0))*/
Group by T1.products_id
Order by CountProducts, T3.products_viewed, T1.products_price Desc

```

```

SELECT
  T1.products_model,
  T3.products_name,
  T3.products_viewed,
  T3.products_url,
  T1.products_price,
  T1.products_quantity,
  T4.manufacturers_name,
  Sum(T2.products_quantity) as CountProducts
FROM /*PREFIX*/products T1
LEFT JOIN /*PREFIX*/orders_products T2 ON (T1.products_id = T2.products_id)
INNER JOIN /*PREFIX*/products_description T3 ON ((T1.products_id = T3.products_id)
and (T3.language_id = :language_id))
LEFT JOIN /*PREFIX*/manufacturers T4 ON (T1.manufacturers_id =
T4.manufacturers_id)
/*Where ((IsNull(T2.products_quantity) = 1) Or (T2.products_quantity = 0))*/
Group by T1.products_id
Order by CountProducts, T3.products_viewed, T1.products_price Desc

```

```

SELECT
  T1.products_model,
  T3.products_name,
  T3.products_viewed,
  T3.products_url,
  T2.products_price,
  T1.products_quantity,
  T4.manufacturers_name,
  Sum(T2.products_quantity) as CountProducts
FROM /*PREFIX*/products T1
LEFT JOIN /*PREFIX*/orders_products T2 ON (T1.products_id = T2.products_id)
INNER JOIN /*PREFIX*/orders T5 ON ((T5.orders_id = T2.orders_id) /*And
(T2.orders_status = 3)*/)
INNER JOIN /*PREFIX*/products_description T3 ON ((T1.products_id = T3.products_id)
and (T3.language_id = :language_id))
LEFT JOIN /*PREFIX*/manufacturers T4 ON (T1.manufacturers_id =
T4.manufacturers_id)
/*Where ((IsNull(T2.products_quantity) = 1) Or (T2.products_quantity = 0))*/
WHERE (T5.date_purchased >= :StartDate or :IgnoreStartDate = 1) And
      (T5.date_purchased <= :FinishDate Or :IgnoreFinishDate = 1)
Group by T1.products_id, T2.products_price
Order by CountProducts Desc, T3.products_viewed Desc, T1.products_price

```

Програмний код інтерфейсу головного модуля програми

```

unit dmMain;

interface

uses
  SysUtils, Classes, ZSqlUpdate, DB, ZAbstractRODataset, ZAbstractDataset,
  ZDataset, ZConnection, Variants, uTypes, cxClasses, cxStyles, Controls, Forms,
  Dialogs,
  cxGridBandedTableView, cxContainer, cxEdit, dxLayoutLookAndFeels, Windows,
  IdBaseComponent, IdComponent, IdTCPConnection, IdTCPClient, IdFTP,
  ActnList, ImgList, ShellApi, ExtCtrls, ZSqlProcessor, ExtDlgs,
  ZAbstractTable, uEditCategory, uEditProductOption, uEditProductOptionValue,
  cxGridTableView,
  dxmdaset, uEditProduct, IdExplicitTLSClientServerBase, ZSqlMonitor, cxDBTL,
  uStringListCollection,
  uTimePeriodSelectDialog, StrUtils, IdHTTP, IniFiles, DKLang,
  uEditProductManufactures,
  uOrdersEditForm, uEditCustomer, uEditProductAttributes, uFastProductsSearch,
  uAttributesEdtDialog, Ole2,
  Decal, uProgressDialog, uCommonConstants, AdwordsCommon, uAdwordsRelatedClasses;

  {$I personal_project_def.inc}

const
  def_LanguageIDParamName = 'language_id';

type
  TdmMainData = class(TDataModule)
    ZConnection1: TZConnection;
    qProducts: TZQuery;
    uProducts: TZUpdateSQL;
    dsProducts: TDataSource;
    qCategories: TZQuery;
    uCategories: TZUpdateSQL;
    dsCategories: TDataSource;
    qProductAttributes: TZQuery;
    uProductAttributes: TZUpdateSQL;
    dsProductAttributes: TDataSource;
    qProductDescription: TZQuery;
    uProductDescription: TZUpdateSQL;
    dsProductDescription: TDataSource;
    qProductsOptions: TZQuery;
    uProductsOptions: TZUpdateSQL;
    dsProductsOptions: TDataSource;
    qProductAttributesproducts_id: TIntegerField;
    qProductAttributesproducts_attributes_id: TIntegerField;
    qProductAttributesoptions_id: TIntegerField;
    qProductAttributesprice_prefix: TStringField;
    qProductAttributesoptions_values_id: TIntegerField;
    qProductsOptionsValues: TZQuery;
    uProductsOptionsValues: TZUpdateSQL;
    dsProductsOptionsValues: TDataSource;
    qProductsOptionsValuesproducts_options_values_to_products_options_id:
    TIntegerField;
    qProductsOptionsValuesproducts_options_id: TIntegerField;
    qProductsOptionsValuesproducts_options_values_id: TIntegerField;
    qProductsOptionsValueslanguage_id: TIntegerField;
    qProductsOptionsValuesproducts_options_values_name: TStringField;
    qProductsproducts_id: TIntegerField;

```

```

qProductsproducts_image: TStringField;
qProductsproducts_date_added: TDateTimeField;
qProductsproducts_last_modified: TDateTimeField;
qProductsproducts_date_available: TDateTimeField;
qProductsproducts_status: TSmallintField;
qProductsmanufacturers_id: TIntegerField;
qProductscategories_id: TIntegerField;
qCategoriesList: TZQuery;
uCategoriesList: TZUpdateSQL;
dsCategoriesList: TDataSource;
qCategoriesList1: TZQuery;
uCategoriesList1: TZUpdateSQL;
qProductsOptionsproducts_options_id: TIntegerField;
qProductsOptionslanguage_id: TIntegerField;
qProductsOptionsproducts_options_name: TStringField;
ilMainLarge: TImageList;
ilMainSmall: TImageList;
ActionList1: TActionList;
acApplyProductChanges: TAction;
acCancelProductChanges: TAction;
acRefreshProducts: TAction;
acNewCategory: TAction;
acEditCategory: TAction;
acDeleteCategory: TAction;
acNewProduct: TAction;
acEditProduct: TAction;
acDeleteProduct: TAction;
acAddProductAttribute: TAction;
acEditProductAttribute: TAction;
acDeleteProductAttribute: TAction;
qManufacturers: TZQuery;
uManufacturers: TZUpdateSQL;
dsManufacturers: TDataSource;
acAddManufacturer: TAction;
acDeleteManufacturer: TAction;
acRefreshManufacturers: TAction;
acApplyManufacturerChanges: TAction;
acCancelManufacturerChanges: TAction;
qManufacturersInfo: TZQuery;
uManufacturersInfo: TZUpdateSQL;
dsManufacturersInfo: TDataSource;
qManufacturersInfolanguages_id: TIntegerField;
qManufacturersInfomanufacturers_url: TStringField;
qManufacturersInfourl_clicked: TIntegerField;
qManufacturersInfodate_last_click: TDateTimeField;
qManufacturersInfomanufacturers_id: TIntegerField;
qManufacturersmanufacturers_id: TIntegerField;
qManufacturersmanufacturers_name: TStringField;
qManufacturersmanufacturers_image: TStringField;
qTaxClass: TZQuery;
uTaxClass: TZUpdateSQL;
dsTaxClass: TDataSource;
qTaxClasstax_class_id: TIntegerField;
qTaxClasstax_class_title: TStringField;
qTaxClasstax_class_description: TStringField;
qTaxClasslast_modified: TDateTimeField;
qTaxClassdate_added: TDateTimeField;
qProductsproducts_tax_class_id: TIntegerField;
qProductsproducts_price_gross: TFloatField;
qProducts1products_price_gross: TFloatField;
qTaxRates: TZQuery;
uTaxRates: TZUpdateSQL;
dsTaxRates: TDataSource;

```

```

qTaxRatestax_rates_id: TIntegerField;
qTaxRatestax_zone_id: TIntegerField;
qTaxRatestax_class_id: TIntegerField;
qTaxRatestax_priority: TIntegerField;
qTaxRatestax_description: TStringField;
qTaxRateslast_modified: TDateTimeField;
qTaxRatesdate_added: TDateTimeField;
qZones: TZQuery;
uZones: TZUpdateSQL;
dsZones: TDataSource;
qProducts1TaxRate: TFloatField;
qProductDescriptionproducts_id: TIntegerField;
qProductDescriptionlanguage_id: TIntegerField;
qProductDescriptionproducts_description: TMemoField;
qProductDescriptionproducts_url: TStringField;
qProductDescriptionproducts_viewed: TIntegerField;
qProductslanguage_id: TIntegerField;
acNewProductOption: TAction;
acEditProductOption: TAction;
acDeleteProductOption: TAction;
acApplyProductOptionsChanges: TAction;
acCancelProductOptionsChanges: TAction;
acRefreshProductOptions: TAction;
acNewProductOptionValue: TAction;
acEditProductOptionValue: TAction;
acDeleteProductOptionValue: TAction;
qCustomers: TZQuery;
uCustomersFull: TZUpdateSQL;
dsCustomers: TDataSource;
qOrders: TZQuery;
uOrdersFull: TZUpdateSQL;
dsOrders: TDataSource;
acApplyCustomerChanges: TAction;
acCancelCustomerChanges: TAction;
acRefreshCustomers: TAction;
qCategoriescategories_id: TIntegerField;
qCategoriescategories_image: TStringField;
qCategoriesparent_id: TIntegerField;
qCategoriessort_order: TIntegerField;
qCategoriesdate_added: TDateTimeField;
qCategorieslast_modified: TDateTimeField;
qCategorieslanguage_id: TIntegerField;
qCategoriescategories_name: TStringField;
acEditCustomer: TAction;
acDeleteCustomer: TAction;
qCountries: TZQuery;
uCountries: TZUpdateSQL;
dsCountries: TDataSource;
qLanguages: TZQuery;
uLanguages: TZUpdateSQL;
dsLanguages: TDataSource;
qLanguageslanguages_id: TIntegerField;
qLanguagesname: TStringField;
qLanguagescode: TStringField;
qLanguagesimage: TStringField;
qLanguagesdirectory: TStringField;
qLanguagessort_order: TIntegerField;
qOrdersProducts: TZQuery;
uOrdersProducts: TZUpdateSQL;
dsOrdersProducts: TDataSource;
acRefreshOrders: TAction;
qOrdersProductsAttributes: TZQuery;
uOrdersProductsAttributes: TZUpdateSQL;

```

```

dsOrdersProductsAttributes: TDataSource;
qOrdersProductsorders_products_id: TIntegerField;
qOrdersProductsproducts_model: TStringField;
qOrdersProductsproducts_id: TIntegerField;
qOrdersProductsorders_id: TIntegerField;
qOrdersProductsAttributesorders_products_attributes_id: TIntegerField;
qOrdersProductsAttributesorders_id: TIntegerField;
qOrdersProductsAttributesorders_products_id: TIntegerField;
qOrdersProductsAttributesproducts_options: TStringField;
qOrdersProductsAttributesprice_prefix: TStringField;
qOrdersTotal: TZQuery;
uOrdersTotal: TZUpdateSQL;
dsOrdersTotal: TDataSource;
qOrdersTotalorders_id: TIntegerField;
qOrdersTotaltitle: TStringField;
qOrdersTotaltext: TStringField;
qOrdersTotalclass: TStringField;
qOrdersTotalsort_order: TIntegerField;
qOrdersTotalorders_total_id: TIntegerField;
qProductsproducts_url: TStringField;
qCategoriesListcategories_id: TIntegerField;
qCategoriesListlanguage_id: TIntegerField;
qCategoriesListcategories_name: TStringField;
qCategoriesList1categories_id: TIntegerField;
qCategoriesList1categories_image: TStringField;
qCategoriesList1parent_id: TIntegerField;
qCategoriesList1sort_order: TIntegerField;
qCategoriesList1date_added: DateTimeField;
qCategoriesList1last_modified: DateTimeField;
qCategoriesList1language_id: TIntegerField;
qCategoriesList1categories_name: TStringField;
qCategoriesList1ParentName: TStringField;
InsertProductsDescriptionZSQLProcessor: TZSQLProcessor;
tProductsRefresh: TTimer;
acEditManufactures: TAction;
OpenPictureDialog: TOpenPictureDialog;
SelOrderStatusSQL: TZQuery;
SelOrderStatusSQLorders_status_id: TIntegerField;
SelOrderStatusSQLorders_status_name: TStringField;
SelOrderStatusDS: TDataSource;
acAttributeEditor: TAction;
AllAvailProductOptionValuesSQL: TZQuery;
AllAvailProductOptionValuesSQLproducts_options_values_id: TIntegerField;
AllAvailProductOptionValuesSQLlanguage_id: TIntegerField;
AllAvailProductOptionValuesSQLproducts_options_values_name: TStringField;
AllAvailProductOptionValuesSQLproducts_options_id: TIntegerField;
AllAvailProductOptionValuesSQLlanguage_id_1: TIntegerField;
AllAvailProductOptionValuesSQLproducts_options_name: TStringField;
AllAvailProductOptionValuesSQLproducts_options_values_to_products_options_id:
TIntegerField;
AllAvailProductOptionValuesSQLDS: TDataSource;
AllAvailProductOptionValuesSQLAttributeChecked: TStringField;
AllAvailProductOptionValuesSQLAttributeValueSign: TStringField;
qProductsOptionsValues1: TZQuery;
qProductsOptionsValues1UpdateSQL: TZUpdateSQL;
qProductsOptionsValues1DS: TDataSource;
qProductsOptionsValues1products_options_values_to_products_options_id:
TIntegerField;
qProductsOptionsValues1products_options_id: TIntegerField;
qProductsOptionsValues1products_options_values_id: TIntegerField;
qProductsOptionsValues1language_id: TIntegerField;
qProductsOptionsValues1products_options_values_name: TStringField;
CanDeleteProductsOptionSQL: TZQuery;

```

```

CanDeleteProductsOptionNameSQL: TZQuery;
ConfigurationTable: TZTable;
ConfigurationTableDS: TDataSource;
ConfigurationTableconfiguration_id: TIntegerField;
ConfigurationTableconfiguration_key: TStringField;
ConfigurationTableconfiguration_group_id: TIntegerField;
ConfigurationTablesort_order: TIntegerField;
ConfigurationTablelast_modified: TDateTimeField;
ConfigurationTabledate_added: TDateTimeField;
qProductAttributesproducts_attributes_filename: TStringField;
qProductAttributesproducts_attributes_maxdays: TIntegerField;
qProductAttributesproducts_attributes_maxcount: TIntegerField;
GetConfigurationValueSQL: TZQuery;
FoundAttributeDownloadsSQL: TZQuery;
acRefreshConfiguration: TAction;
ConfigurationTableInfo: TStringField;
acBackupRestore: TAction;
qProductsmanufacturers_name: TStringField;
qProducts1: TZQuery;
qProducts1products_id: TIntegerField;
qProducts1products_image: TStringField;
qProducts1products_date_added: TDateTimeField;
qProducts1products_last_modified: TDateTimeField;
qProducts1products_date_available: TDateTimeField;
qProducts1products_status: TSmallintField;
qProducts1products_tax_class_id: TIntegerField;
qProducts1manufacturers_id: TIntegerField;
qProducts1products_url: TStringField;
qProducts1language_id: TIntegerField;
qProducts1products_description: TMemoField;
uProducts1: TZUpdateSQL;
dsProducts1: TDataSource;
qProductsCategory: TZQuery;
qProductsCategoryproducts_id: TIntegerField;
qProductsCategorycategories_id: TIntegerField;
uProductsCategory: TZUpdateSQL;
dsProductsCategory: TDataSource;
qProductsCategoryCategoriesName: TStringField;
InsertManufacturesInfoSQL: TZQuery;
qProductAttributesproducts_options_name: TStringField;
GetCountSubCategoriesAndProductsSQL: TZQuery;
FTP: TIdFTP;
qProductsCategoryFullCategoryname: TStringField;
CopyPasteProductsMemData: TdxMemData;
CopyPasteProductsAttributesMemData: TdxMemData;
qProductAttributesproducts_options_values_name: TStringField;
qFeatured: TZQuery;
uFeatured: TZUpdateSQL;
dsFeatured: TDataSource;
qFeaturedfeatured_id: TIntegerField;
qFeaturedproducts_id: TIntegerField;
qFeaturedfeatured_date_added: TDateTimeField;
qFeaturedfeatured_last_modified: TDateTimeField;
qFeaturedexpires_date: TDateTimeField;
qFeatureddate_status_change: TDateTimeField;
qFeaturedstatus: TIntegerField;
qSpecials: TZQuery;
uSpecials: TZUpdateSQL;
dsSpecials: TDataSource;
qSpecialsspecials_id: TIntegerField;
qSpecialsproducts_id: TIntegerField;
qSpecialsspecials_date_added: TDateTimeField;
qSpecialsspecials_last_modified: TDateTimeField;

```

```

qSpecialsexpires_date: TDateTimeField;
qSpecialsdate_status_change: TDateTimeField;
qSpecialsstatus: TIntegerField;
GetTablesListSQL1: TZQuery;
GetTablesListSQL1DS: TDataSource;
acAddCustomer: TAction;
qZoneszone_id: TIntegerField;
qZoneszone_country_id: TIntegerField;
qZoneszone_code: TStringField;
qZoneszone_name: TStringField;
qCountriescountries_id: TIntegerField;
qCountriescountries_name: TStringField;
qCountriescountries_iso_code_2: TStringField;
qCountriescountries_iso_code_3: TStringField;
qCountriesaddress_format_id: TIntegerField;
qCustomers1: TZQuery;
qCustomers1customers_id: TIntegerField;
qCustomers1customers_gender: TStringField;
qCustomers1customers_firstname: TStringField;
qCustomers1customers_lastname: TStringField;
qCustomers1FullName: TStringField;
dsCustomers1: TDataSource;
qCustomers1customers_telephone: TStringField;
qCustomers1customers_email_address: TStringField;
qCurrencies: TZQuery;
qCurrenciescurrencies_id: TIntegerField;
qCurrenciestitle: TStringField;
qCurrenciescode: TStringField;
qCurrenciessymbol_left: TStringField;
qCurrenciessymbol_right: TStringField;
qCurrenciesdecimal_point: TStringField;
qCurrenciesthousands_point: TStringField;
qCurrenciesdecimal_places: TStringField;
qCurrenciesvalue: TFloatField;
qCurrencieslast_updated: TDateTimeField;
dsCurrencies: TDataSource;
qOrdersProductsAttributes1: TZQuery;
qOrdersProductsAttributes1orders_products_attributes_id: TIntegerField;
qOrdersProductsAttributes1orders_id: TIntegerField;
qOrdersProductsAttributes1orders_products_id: TIntegerField;
qOrdersProductsAttributes1products_options: TStringField;
qOrdersProductsAttributes1price_prefix: TStringField;
qProductAttributes1: TZQuery;
qProductAttributes1products_id1: TIntegerField;
qProductAttributes1products_attributes_id: TIntegerField;
qProductAttributes1products_id: TIntegerField;
qProductAttributes1options_id: TIntegerField;
qProductAttributes1price_prefix: TStringField;
qProductAttributes1options_values_id: TIntegerField;
qProductAttributes1products_options_id: TIntegerField;
qProductAttributes1products_options_name: TStringField;
qProductAttributes1products_options_values_name: TStringField;
qProductAttributes1language_id: TIntegerField;
qProductAttributes1products_attributes_id_1: TIntegerField;
qProductAttributes1products_attributes_filename: TStringField;
qProductAttributes1products_attributes_maxdays: TIntegerField;
qProductAttributes1products_attributes_maxcount: TIntegerField;
qProductAttributes1AttributeChecked : TStringField;
dsProductAttributes1 : TDataSource;
CalcSubTotalsSQL : TZQuery;
CalcSubTotalsSQLSumFinalPrice : TFloatField;
RetAdditionalFieldsListSQL: TZQuery;
qCategoriesList2: TZQuery;

```

```

uCategoriesList2: TZUpdateSQL;
dsCategoriesList2: TDataSource;
qCategoriesList2categories_id: TIntegerField;
qCategoriesList2categories_image: TStringField;
qCategoriesList2parent_id: TIntegerField;
qCategoriesList2sort_order: TIntegerField;
qCategoriesList2date_added: TDateTimeField;
qCategoriesList2last_modified: TDateTimeField;
qCategoriesList2language_id: TIntegerField;
qCategoriesList2categories_name: TStringField;
SendAllEmails: TAction;
SelAllEmailsSQL: TZQuery;
SelAllEmailsSQLcustomers_email_address: TStringField;
SelAllEmailsSQLcustomers_firstname: TStringField;
SelAllEmailsSQLcustomers_lastname: TStringField;
ReplaceProductsIDSQLErrorProcessor: TZSQLProcessor;
ReplaceOrdersIDSQLErrorProcessor: TZSQLProcessor;
ZSQLMonitor: TZSQLMonitor;
AllAvailProductOptionValuesSQLAttributeSortOrder: TIntegerField;
AllAvailProductOptionValuesSQLproducts_attributes_filename: TStringField;
AllAvailProductOptionValuesSQLproducts_attributes_maxdays: TIntegerField;
AllAvailProductOptionValuesSQLproducts_attributes_maxcount: TIntegerField;
qProducts1specials_date_added: TDateTimeField;
qProducts1specials_last_modified: TDateTimeField;
qProducts1expires_date: TDateTimeField;
qProducts1date_status_change: TDateTimeField;
qProducts1status: TIntegerField;
GetSubCategoriesIDSQLErrorProcessor: TZQuery;
GetSubCategoriesIDSQLErrorProcessorcategories_id: TIntegerField;
GetProductsIDSQLErrorProcessor: TZQuery;
GetProductsIDSQLErrorProcessorproducts_id: TIntegerField;
DeleteProductsSQL: TZSQLProcessor;
DeleteCategoriesSQL: TZSQLProcessor;
acClearProductOption: TAction;
acClearProductsOptionsValues: TAction;
ReCheckConnectionsTimer: TTimer;
CopyPasteLanguagesMemData: TdxMemData;
CopyPasteLanguagesMemDataLanguage_id: TIntegerField;
CopyPasteLanguagesMemDataName: TStringField;
CopyPasteLanguagesMemDataRel_id: TIntegerField;
SaveDialogXLS: TSaveDialog;
SaveDialogHTML: TSaveDialog;
SaveDialogXML: TSaveDialog;
HTTP: TIdHTTP;
qCategories1: TZQuery;
qCategories1categories_id: TIntegerField;
qCategories1parent_id: TIntegerField;
qCategories1sort_order: TIntegerField;
CalcSubTotalSQLSumPrice: TFloatField;
OrderStatusQuery: TZQuery;
OrderStatusQueryorders_id: TIntegerField;
OrderStatusQuerydate_added: TDateTimeField;
OrderStatusQuerycustomer_notified: TIntegerField;
OrderStatusQuerycomments: TMemoField;
OrderStatusQueryorders_status_name: TStringField;
OrderStatusQuerylanguage_id: TIntegerField;
OrderStatusQuerycustomer_notified_str: TStringField;
OrderStatusQueryDS: TDataSource;
qOrdersorders_id: TIntegerField;
qOrderscustomers_id: TIntegerField;
qOrderscustomers_name: TStringField;
qOrderspayment_method: TStringField;
qOrderscc_type: TStringField;

```



```

qOrderscc_owner: TStringField;
qOrderscc_expires: TStringField;
qOrdersdate_purchased: TDateTimeField;
qOrdersorders_status: TIntegerField;
qOrderscurrency: TStringField;
qOrdersorders_status_name: TStringField;
qOrdersvalue: TFloatField;
ReplaceFeaturedIDSQLErrorProcessor: TZSQLProcessor;
RecheckConnectionSQL: TZQuery;
RecheckConnectionSQLBenchmarkValue: TLargeintField;
qOrdersForCustomer: TZQuery;
dsOrdersForCustomer: TDataSource;
qOrdersForCustomerorders_id: TIntegerField;
qOrdersForCustomercustomers_id: TIntegerField;
qOrdersForCustomercustomers_name: TStringField;
qOrdersForCustomerpayment_method: TStringField;
qOrdersForCustomercc_type: TStringField;
qOrdersForCustomercc_owner: TStringField;
qOrdersForCustomercc_expires: TStringField;
qOrdersForCustomerdate_purchased: TDateTimeField;
qOrdersForCustomerorders_status: TIntegerField;
qOrdersForCustomercurrency: TStringField;
qOrdersForCustomerorders_status_name: TStringField;
qOrdersForCustomervalue: TFloatField;
EditOrderAC: TAction;
qOrdersFull: TZQuery;
qOrdersFullorders_id: TIntegerField;
qOrdersFullcustomers_id: TIntegerField;
qOrdersFullcustomers_name: TStringField;
qOrdersFullcustomers_company: TStringField;
qOrdersFullcustomers_street_address: TStringField;
qOrdersFullcustomers_suburb: TStringField;
qOrdersFullcustomers_city: TStringField;
qOrdersFullcustomers_postcode: TStringField;
qOrdersFullcustomers_state: TStringField;
qOrdersFullcustomers_country: TStringField;
qOrdersFullcustomers_telephone: TStringField;
qOrdersFullcustomers_email_address: TStringField;
qOrdersFullcustomers_address_format_id: TIntegerField;
qOrdersFulldelivery_name: TStringField;
qOrdersFulldelivery_company: TStringField;
qOrdersFulldelivery_street_address: TStringField;
qOrdersFulldelivery_suburb: TStringField;
qOrdersFulldelivery_city: TStringField;
qOrdersFulldelivery_postcode: TStringField;
qOrdersFulldelivery_state: TStringField;
qOrdersFulldelivery_country: TStringField;
qOrdersFulldelivery_address_format_id: TIntegerField;
qOrdersFullbilling_name: TStringField;
qOrdersFullbilling_company: TStringField;
qOrdersFullbilling_street_address: TStringField;
qOrdersFullbilling_suburb: TStringField;
qOrdersFullbilling_city: TStringField;
qOrdersFullbilling_postcode: TStringField;
qOrdersFullbilling_state: TStringField;
qOrdersFullbilling_country: TStringField;
qOrdersFullbilling_address_format_id: TIntegerField;
qOrdersFullpayment_method: TStringField;
qOrdersFullcc_type: TStringField;
qOrdersFullcc_owner: TStringField;
qOrdersFullcc_expires: TStringField;
qOrdersFulllast_modified: TDateTimeField;
qOrdersFulldate_purchased: TDateTimeField;

```

```

qOrdersFullorders_status: TIntegerField;
qOrdersFullorders_date_finished: TDateTimeField;
qOrdersFullcurrency: TStringField;
qOrdersFullorders_status_name: TStringField;
qOrdersFullvalue: TFloatField;
dsOrdersFull: TDataSource;
AddOrderAC: TAction;
qCustomersFull: TZQuery;
qCustomersFullcustomers_id: TIntegerField;
qCustomersFullcustomers_gender: TStringField;
qCustomersFullcustomers_firstname: TStringField;
qCustomersFullcustomers_lastname: TStringField;
qCustomersFullcustomers_email_address: TStringField;
qCustomersFullcustomers_default_address_id: TIntegerField;
qCustomersFullcustomers_telephone: TStringField;
qCustomersFullcustomers_fax: TStringField;
qCustomersFullcustomers_password: TStringField;
qCustomersFullcustomers_info_date_of_last_logon: TDateTimeField;
qCustomersFullcustomers_info_number_of_logons: TIntegerField;
qCustomersFullcustomers_info_date_account_created: TDateTimeField;
qCustomersFullcustomers_info_date_account_last_modified: TDateTimeField;
qCustomersFullglobal_product_notifications: TIntegerField;
qCustomerscustomers_id: TIntegerField;
qCustomerscustomers_gender: TStringField;
qCustomerscustomers_firstname: TStringField;
qCustomerscustomers_lastname: TStringField;
qCustomerscustomers_email_address: TStringField;
qCustomerscustomers_default_address_id: TIntegerField;
qCustomerscustomers_telephone: TStringField;
qCustomerscustomers_fax: TStringField;
qCustomerscustomers_password: TStringField;
dsCustomersFull: TDataSource;
uOrderStatusQuery: TZUpdateSQL;
OrderStatusQueryorders_status_history_id: TIntegerField;
uCustomers: TZUpdateSQL;
uOrders: TZUpdateSQL;
qAddressBook: TZQuery;
qAddressBookaddress_book_id: TIntegerField;
qAddressBookentry_gender: TStringField;
qAddressBookentry_company: TStringField;
qAddressBookentry_firstname: TStringField;
qAddressBookentry_lastname: TStringField;
qAddressBookentry_street_address: TStringField;
qAddressBookentry_suburb: TStringField;
qAddressBookentry_postcode: TStringField;
qAddressBookentry_city: TStringField;
qAddressBookentry_state: TStringField;
qAddressBookentry_country_id: TIntegerField;
qAddressBookentry_zone_id: TIntegerField;
dsAddressBook: TDataSource;
uAddressBook: TZUpdateSQL;
qAddressBookCountryName: TStringField;
qAddressBookcustomers_id: TIntegerField;
qCustomerslcustomers_default_address_id: TIntegerField;
qAddressBook1: TZQuery;
qAddressBook1address_book_id: TIntegerField;
qAddressBook1customers_id: TIntegerField;
qAddressBook1entry_gender: TStringField;
qAddressBook1entry_company: TStringField;
qAddressBook1entry_firstname: TStringField;
qAddressBook1entry_lastname: TStringField;
qAddressBook1entry_street_address: TStringField;
qAddressBook1entry_suburb: TStringField;

```

```

qAddressBooklentry_postcode: TStringField;
qAddressBooklentry_city: TStringField;
qAddressBooklentry_state: TStringField;
qAddressBooklentry_country_id: TIntegerField;
qAddressBooklentry_zone_id: TIntegerField;
qAddressBooklcountries_id: TIntegerField;
qAddressBooklcountries_name: TStringField;
qAddressBooklcountries_iso_code_2: TStringField;
qAddressBooklcountries_iso_code_3: TStringField;
qAddressBookladdress_format_id: TIntegerField;
ZSQLProcessor: TZSQLProcessor;
qFamilies: TZQuery;
qFamiliesfamily_id: TIntegerField;
qFamiliesfamily_name: TStringField;
dsFamilies: TDataSource;
qProductsFamilies: TZQuery;
dsProductsFamilies: TDataSource;
qProductsFamiliesproducts_id: TIntegerField;
qProductsFamiliesfamily_id: TIntegerField;
qFamilies1: TZQuery;
dsFamilies1: TDataSource;
qFamilies1family_id: TIntegerField;
qFamilies1family_name: TStringField;
OrderStatusQueryorders_status_id: TIntegerField;
CanDeleteProductsOptionNameSQLCount1: TIntegerField;
CanDeleteProductsOptionSQLCount1: TIntegerField;
FoundAttributeDownloadsSQLCount1: TIntegerField;
GetCountSubCategoriesAndProductsSQLCount1: TIntegerField;
CalcSubTotalSQLSumPriceTax: TFloatField;
qOrdersProducts1: TZQuery;
qProductsXSell: TZQuery;
qProductsXSellID: TIntegerField;
qProductsXSellproducts_id: TIntegerField;
qProductsXSellxsell_id: TIntegerField;
qProductsXSellsort_order: TIntegerField;
uProductsXSell: TZUpdateSQL;
dsProductsXSell: TDataSource;
dsProductsOptions1: TDataSource;
dsProductsOptionsValues1: TDataSource;
qProductstax_rate: TFloatField;
qProductsspecials_status: TIntegerField;
qOrderscustomers_company: TStringField;
qOrderscustomers_street_address: TStringField;
qOrderscustomers_suburb: TStringField;
qOrderscustomers_city: TStringField;
qOrderscustomers_postcode: TStringField;
qOrderscustomers_state: TStringField;
qOrderscustomers_country: TStringField;
qOrderscustomers_telephone: TStringField;
qOrderscustomers_email_address: TStringField;
qOrderscustomers_address_format_id: TIntegerField;
qOrdersdelivery_name: TStringField;
qOrdersdelivery_company: TStringField;
qOrdersdelivery_street_address: TStringField;
qOrdersdelivery_suburb: TStringField;
qOrdersdelivery_city: TStringField;
qOrdersdelivery_postcode: TStringField;
qOrdersdelivery_state: TStringField;
qOrdersdelivery_address_format_id: TIntegerField;
qOrdersbilling_name: TStringField;
qOrdersbilling_company: TStringField;
qOrdersbilling_street_address: TStringField;
qOrdersbilling_suburb: TStringField;

```

```

qOrdersbilling_city: TStringField;
qOrdersbilling_postcode: TStringField;
qOrdersbilling_state: TStringField;
qOrdersbilling_country: TStringField;
qOrdersbilling_address_format_id: TIntegerField;
qOrderslast_modified: TDateTimeField;
qOrdersorders_date_finished: TDateTimeField;
qCategoriesEdt: TZQuery;
uCategoriesEdt: TZUpdateSQL;
dsCategoriesEdt: TDataSource;
qCategoriesEdtcategories_id: TIntegerField;
qCategoriesEdtcategories_image: TStringField;
qCategoriesEdtparent_id: TIntegerField;
qCategoriesEdtsort_order: TIntegerField;
qCategoriesEdtdate_added: TDateTimeField;
qCategoriesEdtlast_modified: TDateTimeField;
qCategoriesEdtlanguage_id: TIntegerField;
qCategoriesEdtcategories_name: TStringField;
qCategoriesEdtParentName: TStringField;
qProductscategories_name: TStringField;
qTaxClasstmp_tax_rates_id: TIntegerField;
qTaxClasstax_rate: TFloatField;
InsertCategoriesDescriptionsSQL: TZSQLProcessor;
dsOptionsTypeMDS: TDataSource;
OptionsTypeMDS: TdxMemData;
OptionsTypeMDSid: TIntegerField;
OptionsTypeMDSname: TStringField;
qCustomerscustomers_info_date_of_last_logon: TDateTimeField;
qCustomerscustomers_info_number_of_logons: TIntegerField;
qCustomerscustomers_info_date_account_created: TDateTimeField;
qCustomerscustomers_info_date_account_last_modified: TDateTimeField;
qProductspercentage_discount: TFloatField;
qCustomerslcustomers_newsletter: TStringField;
qProductslmanufacturers_name: TStringField;
qOrdersProductslorders_products_id: TIntegerField;
qOrdersProductslproducts_id: TIntegerField;
qOrdersProductslorders_id: TIntegerField;
qMetaTagsCategories: TZQuery;
uMetaTagsCategories: TZUpdateSQL;
dsMetaTagsCategories: TDataSource;
qMetaTagsCategoriescategories_id: TIntegerField;
qMetaTagsCategorieslanguage_id: TIntegerField;
qMetaTagsCategoriesmetatags_title: TStringField;
qMetaTagsCategoriesmetatags_keywords: TMemoField;
qMetaTagsCategoriesmetatags_description: TMemoField;
qMetaTagsProducts: TZQuery;
uMetaTagsProducts: TZUpdateSQL;
dsMetaTagsProducts: TDataSource;
qMetaTagsProductsproducts_id: TIntegerField;
qMetaTagsProductslanguage_id: TIntegerField;
qMetaTagsProductsmetatags_title: TStringField;
qMetaTagsProductsmetatags_keywords: TMemoField;
qMetaTagsProductsmetatags_description: TMemoField;
qProductslproducts_viewed: TIntegerField;
qProductsproducts_viewed: TIntegerField;
SavePictureDialog: TSavePictureDialog;
uExtraFields: TZUpdateSQL;
dsExtraFields: TDataSource;
qExtraFields: TZQuery;
qExtraFieldsproducts_extra_fields_id: TIntegerField;
qExtraFieldsproducts_extra_fields_order: TIntegerField;
qExtraFieldsproducts_extra_fields_status: TSmallintField;
qExtraFieldslanguages_id: TIntegerField;

```

```

qProductsExtraFields: TZQuery;
qProductsExtraFieldsproducts_extra_fields_id: TIntegerField;
qProductsExtraFieldsproducts_extra_fields_order: TIntegerField;
qProductsExtraFieldsproducts_extra_fields_status: TSmallintField;
qProductsExtraFieldslanguages_id: TIntegerField;
qProductsExtraFieldsproducts_id: TIntegerField;
uProductsExtraFields: TZUpdateSQL;
dsProductsExtraFields: TDataSource;
ReplaceExtraFieldsIDSQLErrorProcessor: TZSQLProcessor;
CopyPasteProductsExtraFieldsData: TdxMemData;
CopyPasteProductsExtraFieldsDataproducts_id: TIntegerField;
CopyPasteProductsExtraFieldsDataproducts_extra_fields_name: TStringField;
CopyPasteProductsExtraFieldsDataproducts_extra_fields_value: TStringField;
qSelAllProducts: TZQuery;
qSelAllCategories: TZQuery;
uFamilies: TZUpdateSQL;
dsOrdersProductsDownloads: TDataSource;
uOrdersProductsDownloads: TZUpdateSQL;
qOrdersProductsDownloads: TZQuery;
qOrdersProductsDownloadsorders_products_download_id: TIntegerField;
qOrdersProductsDownloadsorders_id: TIntegerField;
qOrdersProductsDownloadsorders_products_id: TIntegerField;
qOrdersProductsDownloadsorders_products_filename: TStringField;
qOrdersProductsDownloadsdownload_maxdays: TIntegerField;
qOrdersProductsDownloadsdownload_count: TIntegerField;
procedure acApplyProductChangesExecute(Sender: TObject);
procedure acCancelProductChangesExecute(Sender: TObject);
procedure acApplyProductChangesUpdate(Sender: TObject);
procedure acCancelProductChangesUpdate(Sender: TObject);
procedure acCancelCategoryChangesUpdate(Sender: TObject);
procedure acAcceptCategoryChangesUpdate(Sender: TObject);
procedure acRefreshProductsExecute(Sender: TObject);
procedure acNewCategoryExecute(Sender: TObject);
procedure acEditCategoryExecute(Sender: TObject);
procedure acDeleteCategoryExecute(Sender: TObject);
procedure acEditCategoryUpdate(Sender: TObject);
procedure acNewCategoryUpdate(Sender: TObject);
procedure acAddProductAttributeExecute(Sender: TObject);
procedure acEditProductAttributeExecute(Sender: TObject);
procedure acDeleteProductAttributeExecute(Sender: TObject);
procedure DataModuleCreate(Sender: TObject);
procedure DataModuleDestroy(Sender: TObject);
procedure qProductAttributesBeforePost(DataSet: TDataSet);
procedure qProductsBeforePost(DataSet: TDataSet);
procedure qCategoriesEdtBeforePost(DataSet: TDataSet);
procedure qCategoriesListBeforePost(DataSet: TDataSet);
procedure ZConnection1BeforeConnect(Sender: TObject);
procedure qProductsBeforeOpen(DataSet: TDataSet);
procedure acNewProductExecute(Sender: TObject);
procedure acEditProductExecute(Sender: TObject);
procedure acAddManufacturerExecute(Sender: TObject);
procedure acDeleteManufacturerExecute(Sender: TObject);
procedure acRefreshManufacturersExecute(Sender: TObject);
procedure acApplyManufacturerChangesExecute(Sender: TObject);
procedure acCancelManufacturerChangesExecute(Sender: TObject);
procedure acApplyManufacturerChangesUpdate(Sender: TObject);
procedure acCancelManufacturerChangesUpdate(Sender: TObject);
procedure qManufacturersBeforeOpen(DataSet: TDataSet);
procedure qManufacturersAfterOpen(DataSet: TDataSet);
procedure qManufacturersBeforePost(DataSet: TDataSet);
procedure qManufacturersInfoBeforePost(DataSet: TDataSet);
procedure acEditProductUpdate(Sender: TObject);
procedure acDeleteProductUpdate(Sender: TObject);

```

```

procedure acDeleteManufacturerUpdate(Sender: TObject);
procedure acAddManufacturerUpdate(Sender: TObject);
procedure qProductDescriptionBeforePost(DataSet: TDataSet);
procedure acNewProductOptionExecute(Sender: TObject);
procedure acNewProductOptionValueExecute(Sender: TObject);
procedure acEditProductOptionExecute(Sender: TObject);
procedure acDeleteProductOptionValueExecute(Sender: TObject);
procedure acApplyProductOptionsChangesExecute(Sender: TObject);
procedure acCancelProductOptionsChangesExecute(Sender: TObject);
procedure acCancelProductOptionsChangesUpdate(Sender: TObject);
procedure acApplyProductOptionsChangesUpdate(Sender: TObject);
procedure qProductsOptionsValuesBeforePost(DataSet: TDataSet);
procedure acDeleteProductOptionExecute(Sender: TObject);
procedure acEditProductOptionValueExecute(Sender: TObject);
procedure acEditProductOptionValueUpdate(Sender: TObject);
procedure acDeleteProductOptionValueUpdate(Sender: TObject);
procedure acNewProductOptionValueUpdate(Sender: TObject);
procedure acDeleteProductOptionUpdate(Sender: TObject);
procedure acEditProductOptionUpdate(Sender: TObject);
procedure acNewProductOptionUpdate(Sender: TObject);
procedure acApplyCustomerChangesExecute(Sender: TObject);
procedure acCancelCustomerChangesExecute(Sender: TObject);
procedure acApplyCustomerChangesUpdate(Sender: TObject);
procedure acCancelCustomerChangesUpdate(Sender: TObject);
procedure acRefreshCustomersExecute(Sender: TObject);
procedure acDeleteCustomerExecute(Sender: TObject);
procedure acDeleteCustomerUpdate(Sender: TObject);
procedure acEditCustomerUpdate(Sender: TObject);
procedure ZConnection1AfterConnect(Sender: TObject);
procedure qCategoriesBeforeOpen(DataSet: TDataSet);
procedure qProductsOptionsBeforeOpen(DataSet: TDataSet);
procedure qProductsOptionsValuesBeforeOpen(DataSet: TDataSet);
procedure acRefreshProductOptionsExecute(Sender: TObject);
procedure qManufacturersInfoBeforeOpen(DataSet: TDataSet);
procedure acRefreshOrdersExecute(Sender: TObject);
procedure qOrdersTotaltextGetText(Sender: TField; var Text: String;
DisplayText: Boolean);
procedure ZConnection1BeforeDisconnect(Sender: TObject);
procedure acDeleteProductExecute(Sender: TObject);
procedure qCategoriesList1BeforeOpen(DataSet: TDataSet);
procedure qCategoriesListBeforeOpen(DataSet: TDataSet);
procedure qProductsNewRecord(DataSet: TDataSet);
procedure qProductsAfterPost(DataSet: TDataSet);
procedure tProductsRefreshTimer(Sender: TObject);
procedure acEditManufacturesExecute(Sender: TObject);
procedure qManufacturersInfoNewRecord(DataSet: TDataSet);
procedure qManufacturersNewRecord(DataSet: TDataSet);
procedure acAddProductAttributeUpdate(Sender: TObject);
procedure acDeleteProductAttributeUpdate(Sender: TObject);
procedure acAttributeEditorExecute(Sender: TObject);
procedure acEditProductAttributeUpdate(Sender: TObject);
procedure acAttributeEditorUpdate(Sender: TObject);
procedure AllAvailProductOptionValuesSQLAfterPost(DataSet: TDataSet);
procedure qProductsOptionsValues1BeforePost(DataSet: TDataSet);
procedure qProductAttributesoptions_idChange(Sender: TField);
procedure dsProductsDataChange(Sender: TObject; Field: TField);
procedure qProductsOptionsBeforeDelete(DataSet: TDataSet);
procedure qProductsOptionsValues1BeforeDelete(DataSet: TDataSet);
procedure qProductsOptionsValuesBeforeDelete(DataSet: TDataSet);
procedure acRefreshConfigurationExecute(Sender: TObject);
procedure ConfigurationTableCalcFields(DataSet: TDataSet);
procedure acBackupRestoreExecute(Sender: TObject);
procedure acBackupRestoreUpdate(Sender: TObject);

```

```

procedure ConfigurationTableBeforePost (DataSet: TDataSet);
procedure ZConnection1AfterDisconnect (Sender: TObject);
procedure ConfigurationTableAfterOpen (DataSet: TDataSet);
procedure qProducts1CalcFields (DataSet: TDataSet);
procedure qProductsCategoryBeforeDelete (DataSet: TDataSet);
procedure qProductsCategoryNewRecord (DataSet: TDataSet);
procedure qProducts1NewRecord (DataSet: TDataSet);
procedure qCustomersFullAfterOpen (DataSet: TDataSet);
procedure AllAvailProductOptionValuesSQLAfterOpen (DataSet: TDataSet);
procedure qOrdersTotalAfterOpen (DataSet: TDataSet);
procedure qOrdersProductsAttributesAfterOpen (DataSet: TDataSet);
procedure qOrdersProductsAfterOpen (DataSet: TDataSet);
procedure qTaxRatesAfterOpen (DataSet: TDataSet);
procedure qProductAttributesAfterOpen (DataSet: TDataSet);
procedure qProducts1AfterOpen (DataSet: TDataSet);
procedure qProductsAfterOpen (DataSet: TDataSet);
procedure dsManufacturersDataChange (Sender: TObject; Field: TField);
procedure qCategoriesNewRecord (DataSet: TDataSet);
procedure qProductsOptionsValues1NewRecord (DataSet: TDataSet);
procedure qProductsOptionsBeforePost (DataSet: TDataSet);
procedure qProductsOptionsAfterPost (DataSet: TDataSet);
procedure qProductsOptionsValues1AfterPost (DataSet: TDataSet);
procedure qProductsOptionsValuesAfterPost (DataSet: TDataSet);
procedure qCategoriesBeforeDelete (DataSet: TDataSet);
procedure qCategoriesAfterPost (DataSet: TDataSet);
procedure qProductAttributesBeforeEdit (DataSet: TDataSet);
procedure qProductAttributesBeforeInsert (DataSet: TDataSet);
procedure qProductsCategoryBeforePost (DataSet: TDataSet);
procedure qProductsCategoryBeforeClose (DataSet: TDataSet);
procedure qProductsCategoryAfterOpen (DataSet: TDataSet);
procedure qProductsCategoryCalcFields (DataSet: TDataSet);
procedure qProductsCategoryBeforeInsert (DataSet: TDataSet);
procedure qProductsCategoryBeforeEdit (DataSet: TDataSet);
procedure qProductsOptionsNewRecord (DataSet: TDataSet);
procedure qProductsOptionsValuesNewRecord (DataSet: TDataSet);
procedure qFeaturedNewRecord (DataSet: TDataSet);
procedure qFeaturedBeforePost (DataSet: TDataSet);
procedure qSpecialsBeforePost (DataSet: TDataSet);
procedure qSpecialsNewRecord (DataSet: TDataSet);
procedure qCustomersFullBeforePost (DataSet: TDataSet);
procedure acAddCustomerExecute (Sender: TObject);
procedure qCustomersFullBeforeInsert (DataSet: TDataSet);
procedure qCustomersFullBeforeEdit (DataSet: TDataSet);
procedure qCustomers1CalcFields (DataSet: TDataSet);
procedure qOrdersProductsNewRecord (DataSet: TDataSet);
procedure qProductAttributes1AfterPost (DataSet: TDataSet);
procedure qOrdersProductsAttributesBeforePost (DataSet: TDataSet);
procedure qOrdersTotalBeforePost (DataSet: TDataSet);
procedure qProductAttributesNewRecord (DataSet: TDataSet);
procedure qCustomersFullNewRecord (DataSet: TDataSet);
procedure SendAllEmailsExecute (Sender: TObject);
procedure ReplaceProductsIDSQLError (Processor: TZSQLProcessor;
  StatementIndex: Integer; E: Exception;
  var ErrorHandleAction: TZErrorHandleAction);
procedure ZSQLMonitorTrace (Sender: TObject; Event: TZLoggingEvent;
  var LogTrace: Boolean);
//Diff
procedure qProductAttributesBeforeOpen (DataSet: TDataSet);
//End diff
procedure ReCheckConnectionsTimerTimer (Sender: TObject);
procedure DeleteProductsSQLError (Processor: TZSQLProcessor;
  StatementIndex: Integer; E: Exception;
  var ErrorHandleAction: TZErrorHandleAction);

```

```

procedure acClearProductOptionExecute(Sender: TObject);
procedure acClearProductOptionUpdate(Sender: TObject);
procedure acClearProductsOptionsValuesExecute(Sender: TObject);
procedure acClearProductsOptionsValuesUpdate(Sender: TObject);
procedure qOrdersProductsBeforeEdit(DataSet: TDataSet);
procedure qOrdersProductsAfterPost(DataSet: TDataSet);
procedure acNewProductUpdate(Sender: TObject);
procedure InsertProductsDescriptionZSQLProcessorBeforeExecute(
  Processor: TZSQLProcessor; StatementIndex: Integer);
procedure OrderStatusQueryCalcFields(DataSet: TDataSet);
procedure EditOrderACUpdate(Sender: TObject);
procedure EditOrderACEExecute(Sender: TObject);
procedure qOrdersFullAfterOpen(DataSet: TDataSet);
procedure qOrdersFullPostError(DataSet: TDataSet; E: EDatabaseError;
  var Action: TDataAction);
procedure qOrdersFullNewRecord(DataSet: TDataSet);
procedure qOrdersFullBeforePost(DataSet: TDataSet);
procedure qOrdersFullBeforeInsert(DataSet: TDataSet);
procedure qOrdersFullBeforeEdit(DataSet: TDataSet);
procedure qOrdersFullAfterPost(DataSet: TDataSet);
procedure qOrdersFullAfterCancel(DataSet: TDataSet);
procedure AddOrderACEExecute(Sender: TObject);
procedure acEditCustomerExecute(Sender: TObject);
procedure ConfigurationTableBeforeOpen(DataSet: TDataSet);
procedure qAddressBookBeforePost(DataSet: TDataSet);
procedure dsCustomersDataChange(Sender: TObject; Field: TField);
procedure qAddressBookNewRecord(DataSet: TDataSet);
procedure qAddressBookBeforeDelete(DataSet: TDataSet);
procedure ZSQLProcessorError(Processor: TZSQLProcessor;
  StatementIndex: Integer; E: Exception;
  var ErrorHandleAction: TZErrorHandleAction);
procedure qProductsFamiliesNewRecord(DataSet: TDataSet);
procedure qProductAttributesBeforeDelete(DataSet: TDataSet);
procedure qProductsBeforeDelete(DataSet: TDataSet);
procedure qProducts1BeforeDelete(DataSet: TDataSet);
procedure qSpecialsBeforeDelete(DataSet: TDataSet);
procedure qFeaturedBeforeDelete(DataSet: TDataSet);
procedure qManufacturersBeforeDelete(DataSet: TDataSet);
procedure qManufacturersInfoBeforeDelete(DataSet: TDataSet);
procedure qCustomersBeforePost(DataSet: TDataSet);
procedure qOrdersForCustomerBeforePost(DataSet: TDataSet);
procedure qOrdersForCustomerBeforeDelete(DataSet: TDataSet);
procedure qCustomersFullBeforeDelete(DataSet: TDataSet);
procedure qCustomersBeforeDelete(DataSet: TDataSet);
procedure acAddCustomerUpdate(Sender: TObject);
procedure qOrdersBeforePost(DataSet: TDataSet);
procedure qOrdersBeforeDelete(DataSet: TDataSet);
procedure qOrdersProductsBeforePost(DataSet: TDataSet);
procedure qOrdersProductsBeforeDelete(DataSet: TDataSet);
procedure qOrdersProductsAttributesBeforeDelete(DataSet: TDataSet);
procedure qOrdersTotalBeforeDelete(DataSet: TDataSet);
procedure qOrdersFullBeforeDelete(DataSet: TDataSet);
procedure OrderStatusQueryBeforePost(DataSet: TDataSet);
procedure OrderStatusQueryBeforeDelete(DataSet: TDataSet);
procedure ConfigurationTableBeforeDelete(DataSet: TDataSet);
procedure qOrdersAfterOpen(DataSet: TDataSet);
procedure qCustomersAfterOpen(DataSet: TDataSet);
procedure InsertProductsDescriptionZSQLProcessorError(
  Processor: TZSQLProcessor; StatementIndex: Integer; E: Exception;
  var ErrorHandleAction: TZErrorHandleAction);
procedure ReplaceFeaturedIDSQLError(Processor: TZSQLProcessor;
  StatementIndex: Integer; E: Exception;
  var ErrorHandleAction: TZErrorHandleAction);

```



```

procedure AllAvailProductOptionValuesSQLBeforePost (DataSet: TDataSet);
procedure qCategoriesAfterOpen (DataSet: TDataSet);
procedure OrderStatusQueryNewRecord (DataSet: TDataSet);
procedure qProductsXSellAfterOpen (DataSet: TDataSet);
procedure qProductsXSellBeforePost (DataSet: TDataSet);
procedure qProductsXSellNewRecord (DataSet: TDataSet);
procedure qOrdersForCustomerAfterOpen (DataSet: TDataSet);
procedure qOrdersNewRecord (DataSet: TDataSet);
procedure qOrdersBeforeEdit (DataSet: TDataSet);
procedure qCategoriesEdtNewRecord (DataSet: TDataSet);
procedure InsertCategoriesDescriptionsSQLError (Processor: TZSQLProcessor;
StatementIndex: Integer;
    E: Exception; var ErrorHandlerAction: TZErrorHandlerAction);
procedure qProductsOptionsAfterOpen (DataSet: TDataSet);
procedure qProductspercentage_discountChange (Sender: TField);
procedure qProductsspecials_new_products_priceChange (Sender: TField);
procedure qSpecialsAfterOpen (DataSet: TDataSet);
procedure qMetaTagsCategoriesNewRecord (DataSet: TDataSet);
procedure qMetaTagsProductsNewRecord (DataSet: TDataSet);
procedure qOrdersProducts1AfterOpen (DataSet: TDataSet);
procedure qCategoriesBeforePost (DataSet: TDataSet);
procedure qProductsFilterRecord (DataSet: TDataSet; var Accept: Boolean);
procedure qExtraFieldsAfterOpen (DataSet: TDataSet);
procedure qExtraFieldsAfterPost (DataSet: TDataSet);
procedure qExtraFieldsBeforeDelete (DataSet: TDataSet);
procedure qExtraFieldsBeforePost (DataSet: TDataSet);
procedure qProductsExtraFieldsAfterOpen (DataSet: TDataSet);
procedure qProductsExtraFieldsAfterPost (DataSet: TDataSet);
procedure qProductsExtraFieldsBeforePost (DataSet: TDataSet);
procedure qOrdersFilterRecord (DataSet: TDataSet; var Accept: Boolean);
procedure qOrdersAfterRefresh (DataSet: TDataSet);
procedure qProductDescriptionAfterOpen (DataSet: TDataSet);
procedure qCustomersFilterRecord (DataSet: TDataSet; var Accept: Boolean);
procedure qFamiliesBeforePost (DataSet: TDataSet);
procedure qOrdersProductsAfterDelete (DataSet: TDataSet);
procedure qOrdersProductsDownloadsBeforeDelete (DataSet: TDataSet);
procedure qOrdersProductsDownloadsBeforePost (DataSet: TDataSet);
private
{ Private declarations }
StoredDataSetsState : TList;
StoredDataSetsPosition : TStringList;

NeedForceReconnect : Boolean;
SSHRemoteHostName : String;
SSHRemotePort : WORD;
LastManufacturersId : Integer;

qProductsOptionsInsertStatus,
qProductsOptionsValues1InsertStatus,
qProductsOptionsValuesInsertStatus,
qProductsCategoryInsertStatus,
qExtraFieldsInsertStatus : Boolean;

fmEditProductOptionValueDialogActive : Boolean;
fmEditProductOptionDialogActive : Boolean;
fmEditCategoryDialogActive : Boolean;

fmEditCategory : TfmEditCategory;
fmEditProductOption : TfmEditProductOption;
fmEditProductOptionValue : TfmEditProductOptionValue;
fmEditProduct : TfmEditProduct;
fmEditProductManufactures : TfmEditProductManufactures;
fmOrderEdit : TfmOrderEdit;

```

```

fmEditCustomer          : TfmEditCustomer;
fmEditAttributesExt     : TfmEditAttributesExt;

qProductsCategoryIdentMap      : DMap;
qProductsCategorycategories_idOldVal : Integer;
SaveqCustomerscustomers_password : String;

SilentErrorMode          : Boolean;
{$IFDEF DEMO}DemoOrdersAddCounter : Integer;{$ENDIF}
{$IFDEF DEMO}DemoCustomersAddCounter : Integer;{$ENDIF}
SavedTemporaryProductsId,
SavedTemporaryOrdersId      : Integer;

StringListCollection : TStringListCollection;
NewCategoryParent      : Integer;
DontChangeSpecialsAndFeaturedTables : Boolean;
FOrdersProductsDelta  : Double;
FUpdateOrdersProductsQuantity : Boolean;
//-----
// Diff private
//-----
QProductsAttributesForceAscSorting : Boolean;
CurrAttributesSortingOrder         : Boolean;

SilentDeleteCategories          : Boolean;
CountDeletedSubCategoriesAndProducts : Integer;
SkipTdmMainDataqProducts1AfterOpen,
SkipTdmMainDataqProductsAfterOpen,
SkipTdmMainDataqCustomersAfterOpen,
SkipdmMainDataqProductAttributesAfterOpen,
SkipdmMainDataAllAvailProductOptionValuesSQLAfterOpen,
SkipdmMainDataqOrdersAfterOpen,
SkipqCustomersAfterOpen,
SkipTdmMainDataqCategoriesAfterOpen,
SkipdmMainDataqOrdersFullAfterOpen : Boolean;
SkipqProductsOptionsAfterOpen : Integer;
Skip_TdmMainData_ZSQLMonitorTrace : Integer;

LastRecheckConnectionTime : DWORD;

AfterConnectScriptErrorStr : String;
CurrentXSellSortOrder      : Integer;
CurrentXSellMaxId         : Integer;

SaveOrderStatus, SaveOrderStatus1 : Integer;
LastProductID: Integer;

LastAttributesPrefix : String;

qProductsSQLTemplate : String;
LockqProductspercentage_discountChange : Integer;

dmProgressForm          : TProgressForm;
CountDbOperationProcessed : Integer;
LastProgressFormUpdateTime : DWORD;
FConnectComplete        : Boolean;
SilentShowPreferencess  : Boolean;

NeedShowChoiceCategoryDeleteMethodForm : Boolean;

FFoundFeaturedTable,
FFoundFamiliesTables,
FFoundXSellTable,

```

```

FFoundOptionsTypeField,
FFoundOptionsSortOrderField,
FFoundOptionsLengthField,
FFoundExtraImages,
FFoundZenCardTables,
FFoundExtraFieldsTables      : Boolean;

procedure AfterChangeDBLanguage();
procedure GetCountProductsAndSubCategories(var CountSubCategories : Integer;
var CountProducts : Integer);
procedure GetProductsAndSubCategoriesID1(SubCategoriesIdList, ProductsIdList :
TStringList; TargetCategoryID : Integer);
procedure AddAdditionalIndexes();
procedure ModifyDBFields();
function AddUnknownTypeFields( DataSet : TDataSet; FieldsArray : Variant;
NeedPostReopen : Boolean = True ) : Boolean; inline;
function NeedRunSSHClient() : Boolean;
procedure RunOrRestartSSHClient();
procedure RunAfterDBConnect();
procedure FastSyncProductsQuery();
procedure FillOptionsTypesDataSet();

public
AllAvailProductOptionValuesSQLUpdated : Boolean;
qProductAttributes1Updated           : Boolean;
SavedAttributeFileUrlOrFileName      : String;
ProductsGridTableController,
AttributesGridTableController,
CustomersGridTableController        : TcxGridTableController;
CategoriesTreeList                   : TcxDBTreeList;
SavedLastProductsCategoryIdForPaste  : Integer;
SavedLastProductsFamiliesIdForPaste : Integer;
PasteProductsRecordsFromStreamMsgStr,
PasteAttributesFromStreamMsgStr      : String;

OnPasteNeedOnlyCopyToFamilies       : Boolean;

OptionsTypeFieldName,
OptionsSortOrderFieldName,
OptionsLengthFieldName : String;
ExcludeOptionsAdditionalFieldsList  : TStringList;

AdditionalProductsFieldsList,
AdditionalProductsDescriptionFieldsList,
AdditionalCustomersFieldsList,
AdditionalProductsAttributesFieldsList,
AdditionalOrdersFieldsList,
AdditionalCategoriesFieldsList,
AdditionalProductsOptionsFieldsList,
AdditionalCategoriesDescriptionFieldsList      : TStringList;
AllProductsSelectionSQL,
OnlyCategoriesProductsSelectionSQL : String;

TemporaryAppHandle           : DWORD;
PasteAttributesFromStreamWithoutPreparing : Boolean;

ActiveDBConnection          : TZConnection;

ForcePasteInCleanCopy       : Boolean;
PastedProductsIdList       : TList;

AlreadyAutoAddedFieldsTablesList : TStringList;

```

```

CannotDisconnect          : Boolean;

fmFastProductsSearch      : TfmFastProductsSearch;
fmAttributeEdtDialog      : TAttributeEdtDialog;
DeleteProductIfEditProductDialogCancel : Boolean;

CategoryDeletingMethod : TDeleteCategoryTypeEnum;
NeedFullRefreshProducts  : Boolean;
FilteredRecordsMap : DMap;

//-----
//Dynamic assigned fields
qProductsproducts_name,
qProductsproducts_price,
qProducts1products_name,
qOrdersProductsproducts_name,
qProductsXSellproducts_name : TField;
//-----

EditDescriptionOnly : Boolean;

constructor Create(AOwner: TComponent); override;
destructor Destroy(); override;

function GetValues(Fieldname: String; List: TStringList; TableName: String;
idFieldName: String = '');
    const SortOrderFieldName : String = '') : Boolean; overload;
function GetValues(Fieldname: string; List: TStringList; TableName: string;
    ConditionFieldNames: array of string; ConditionFieldValues: Variant):
Boolean; overload;
function RefreshQuery(AQuery: TZQuery; AParams: Variant; RestoreCursorPos :
Boolean = True;
    ForceRefresh : Boolean = False): Boolean; inline;
function RefreshProducts(ForceRefresh : Boolean = False;
WithoutProductsTableRefresh : Boolean = False): Boolean;
function RefreshProductOptions(): Boolean;
function RefreshManufacturers(): Boolean;
function RefreshCustomers(ForceRefresh : Boolean = False): Boolean;
function RefreshProductsOptions1(ForceRefresh : Boolean = False): Boolean;
function RefreshProductsOptions(ForceRefresh : Boolean = False): Boolean;
function RefreshOrders(ForceRefresh : Boolean = True) : Boolean;
function GetMaxID(TableName, IDFieldName: string; AppendPrefix : Boolean =
True): Integer;
function SetDefaultIdValue(Query: TDataSet; const FieldName: string; const
TableName : String;
    AdditionalIncrement : Integer = 0; AppendPrefix : Boolean = True): Boolean;
procedure ProcessAdditionalFieldsList(DataSet: TDataSet; FieldsList:
TStringList);
//function GetRecordCount(ASQL: string): Integer;

//-----
procedure SaveActiveDataSets();
procedure RestoreSavedActiveDataSets();
procedure RefreshActiveDataSets();
//-----

function ExecCustomSQL(const SQL_STR : String; ZConnection : TZConnection =
Nil) : Integer; inline;
function GetCustomSQLResult(const SQL_STR : String; ZConnection : TZConnection
= Nil) : Variant; inline;
function GetCustomSQLResultArray(const SQL_STR : String; ZConnection :
TZConnection = Nil) : Variant; inline;

```

```

function RetValueForConfigKey(Key : String) : String;
function DataBaseConnectionParamsAreChanged() : Boolean;
function GetFullCategoryPath(CategoryId : Integer; const DelimChar : String) :
String;

procedure CommonReconnectDataModule();
//-----
function CopySelectedProductsRecordsToClipboard() : Boolean;
function PasteProductsRecordsFromClipboard() : Boolean;
function isValidProductsClipboardContent() : Boolean;

function CopySelectedProductsToStream(CopyStream : TStream) : Boolean;
function PasteProductsRecordsFromStream(PasteStream : TStream) : Boolean;
//-----
function CopySelectedAttributesToClipboard() : Boolean;
function PasteAttributesFromClipboard(GridTableController:
TcxGridTableController = nil;
    ShowProgress : Boolean = False) : Boolean;
function isValidAttributesClipboardContent() : Boolean;

function CopySelectedAttributesToStream(CopyStream : TStream) : Boolean;
function PasteAttributesFromStream(PasteStream : TStream) : Boolean;
//-----
function RetAdditionalFieldsList(EtalonList : TStringList; TableName : String)
: TStringList;
function AdditionalProductsFieldsPresent() : Boolean;
function AdditionalCategoriesFieldsPresent() : Boolean;
//-----
procedure DeactivateMonitor();
procedure ActivateMonitor();
//-----
procedure SilentReconnectDB();
procedure OnExternalDBFormDestroy(EForm : TComponent);

procedure GetProductsAndSubCategoriesID(SubCategoriesIdList, ProductsIdList :
TStringList);
procedure FastSyncCategorieQuery();
function Create_GetAdwordsDataClassInstance() : TBaseGetAdwordsDataClass;
inline;
property ConnectComplete : Boolean read FConnectComplete;

property FoundFeaturedTable : Boolean read FFoundFeaturedTable;
property FoundFamiliesTables : Boolean read FFoundFamiliesTables;
property FoundXSellTable : Boolean read FFoundXSellTable;
property FoundOptionsTypeField : Boolean read FFoundOptionsTypeField;
property FoundOptionsSortOrderField : Boolean read
FFoundOptionsSortOrderField;
property FoundOptionsLengthField : Boolean read FFoundOptionsLengthField;
property FoundExtaImages : Boolean read FFoundExtaImages;
property FoundZenCardTables : Boolean read FFoundZenCardTables;
property FoundExtraFieldsTables : Boolean read FFoundExtraFieldsTables;
property OrdersProductsDelta : Double read FOrdersProductsDelta;
property UpdateOrdersProductsQuantity : Boolean read
FUpdateOrdersProductsQuantity write FUpdateOrdersProductsQuantity;
procedure UpdateDmProgressForm();
end;

var
dmMainData: TdmMainData;
ConnectionError: Boolean = True;
DoManufacturersAfterScroll: Boolean = False;
RefreshProductsAfterCategoryScroll: Boolean = True;
DorefreshOrdersProductsAfterOrdersScroll: Boolean = True;

```

```

CurrentLanguageID : Integer = -1; //active language
CurrentLanguageName: String;
ShowConnectionSplash: Boolean = True;
_PastingProducts: Boolean = False;
{Demo}
_ProductsAddedSession      : Integer = 0;
_ProductsEditSession       : Integer = 0;
_PasteProductActionCount   : Integer = 0;
_PasteAttributesActionCount : Integer = 0;
LastSQL : TStringList;

const
  _ProductsAddLimit: Integer = 1;
  _ProductsEditLimit: Integer = 1;
  _PasteProductActionLimit: Integer = 1;
  _PasteAttributesActionLimit: Integer = 1;

  SSHClientAppName = 'plink.exe';
  SSHParamStr      = '| -ssh -2 -C -batch -l %s -pw %s -P %d -L %d:127.0.0.1:%d
%s';

//-----
//Diff consts
//-----

function FTPConnectionPresent() : Boolean; forward;
procedure NormalizedDxmemDataSet(dxMemDataSet : TdxMemData); forward;

implementation

uses
  uSettings, uConstants, uUtils, uUtils_1, uConnectingSplash, uPreferences,
  uImageCache, uGetFTPImageThread, QStrings, uBackupRestoreWizard,
  uCategoriesDescriptionFrame, uTestDataCompatibilityDialog, Messages, WinSock,
  MutexUn,
  uProductsDescriptionFrame, uOptionsDescriptionFrame,
uOptionValuesDescriptionFrame, Clipbrd,
  uChoiceCopyProductsMethod, uChoiceDBLang, ReBranding, Math,
  ZDbcLogging, udbUtils, dmControls, uConfigListSaver, CommonDM,
  CommonReportsDM, uUserAccess, uLoginForm, uChoiceCopyProductsMethod_1,
  PJMessageDialog, cxTL,
  uChoiceCategoryDeletingMethodUn, JclDebug;

{$R *.dfm}

```