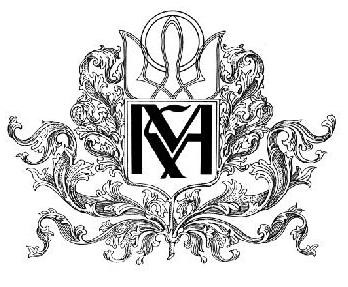
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ"

ФАКУЛЬТЕТ ІНФОРМАТИКИ

Кафедра інформатики



Самостійна робота з дисципліни:

«Бази даних та інформаційні системи»

**Розробка системи інформаційної підтримки мережі комп’ютерних майстернь**

Виконав:

Студент 2 групи  
Аугустин Марко

Камінецький Михайло

Викладач:

Ющенко Ю.О.

Київ – 2019

**Зміст**

[**1. Технічне завдання**](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.gjdgxs)

[***1.1. Мета розробки./Призначення АІС.***](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.30j0zll)

[***1.2. Опис предметної області*** (Вхідні дані).](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.1fob9te)

[1.2.1. Загальна інформація (Опис організації та її діяльності).](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.3znysh7)

1.2.2. [Збір фактів.](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.2et92p0)

[**2. Технічний проект**](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.tyjcwt)

[***2.1. Технічні вимоги***](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.3dy6vkm)

[2.1.1. Опис груп користувачів АІС.](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.1t3h5sf)

[2.1.2. Технічні вимоги користувачів до функціональності системи.](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.4d34og8)

[2.1.3. Специфікація вимог](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.2s8eyo1)

[***2.2. Інфологічне та даталогічне проектування*** (Інформаційне забезпечення)](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.17dp8vu)

[2.2.1. Концептуальна модель даних (ручне проектування)](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.3rdcrjn)

[2.2.2. Концептуальна модель даних (в CASE-системі)](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.26in1rg)

[2.2.3. Реляційна модель даних.](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.lnxbz9)

[**D. Висновки.**](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.1ksv4uv)

[**E. Додатки** (документи, які використовувались під час аналізу предметної області: внутрішні та зовнішні документи фірми, анкети (структура), анкети (заповнені), звітності тощо).](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.44sinio)

[**F. Перелік використаних джерел інформації та/або літератури.**](https://docs.google.com/document/d/1hrz9Qts0DQBrJ9nYDpTuiAPVF-xedAuPiFmfrmcNn2M/edit?copiedFromTrash#heading=h.2jxsxqh)

# 1. Технічне завдання

# 1.1. Мета розробки./Призначення АІС.

Автоматизована інформаційна система призначена для створення зручних умов та вдосконалення процесу роботи компанії, яка надає послуги в сфері ремонту та технічного обслуговування сучасної техніки. Метою розробки є спрощення процедури вибору та оформлення замовлення клієнтами, а також вдосконалення процесу роботи менеджерів, кур’єрів та майстрів. Таким чином застосунок повинен забезпечити автоматизацію наступних завдань:

* Автоматизація обробки, організації та збереження інформації, яка надалі буде використовуватись працівниками мережі.
* Можливість переглянути актуальні списки активних замовлень, доступних послуг, працівників мережі, клієнтську базу, тощо.
* Отримання інформації про клієнтів, наявні послуги чи активне замовлення.
* Можливість провести аналіз роботи як окремого відділення, так і всієї мережі за допомогою спеціальних звітів.

## 1.2. Опис предметної області (Вхідні дані).

## 1.2.1. Загальна інформація (Опис організації та її діяльності).

Компанія «Nerds Solution» являє собою мережу сучасних сервісних центрів, що появилась на ринку сервісних послуг у 2010 році. З часу заснування компанія відкрила 54 нових майстерень та продовжує розвиватись, завойовуючи довіру користувачів на території України. Головною задачею майстерень «Nerds Solution» є задоволення покупця та забезпечення якісних сервісних послуг, пов’язаних з комп’ютерною технікою та сучасними смарт-девайсами загалом.

Високі вимоги до співробітників та постійне вивчення нових технологій на ринку комп’ютерної техніки, дозволяє спеціалістам швидко і вміло виконувати такі завдання як заміна, чистка та ремонт комп’ютерного та смарт обладнання. Ключовою перевагою даної мережі багато клієнтів називають безплатну заміну їхньої техніки іншим девайсом тієї ж категорії(смартфон, ноутбук, ПК тощо), на період проведення сервісних робіт.

На сьогоднішній день з компанія щомісяця виконує замовлення від понад 6500 клієнтів. Однак процес оформлення замовлення, внесення його в базу даних та подальшої передачі інформації кур’єрам дуже складний і потребує спрощення.

В даний час робота відбувається наступним чином: клієнт дзвонить в офіс найближчого до нього філіалу, де йому відповідає секретар, що слідкує за телефоном та приймає нові замовлення, записуючи їх на аркуші паперу. Секретар передає занотовану інформацію менеджеру, який вносить отримані дані в базу Excel та здійснює необхідні підрахунки із врахуванням можливої знижки, після чого передзвонює клієнту для підтвердження замовлення. Наступним кроком, менеджер обирає вільного в даний момент кур’єра та надає йому необхідну інформацію про клієнта(адрес, контакти) та замовлення(тип девайсу, спосіб оплати тощо). Кур’єр на складі отримує товар яким користуватиметься клієнт на час ремонту. Після здійснення обміну технікою з клієнтом, кур’єр повертається до майстерні і віддає замовлення майстрам, які повідомлять по завершенню роботи. Оплата фіксована та проводиться на місці доставки, після того як кур’єр повертається вдруге із уже виконаним замовленням до клієнта. Останнім кроком кур1єр повертається до відділення та доповідає про завершення доставки, а менеджер доопрацьовує дане замовлення та формує звіт.

Розроблена АІС повинна пришвидшити і полегшити процес комунікації між кур’єром та менеджером, а також зменшить шанс похибки при оформленні заказу, оскільки спрощує ланцюг, викидаючи таку ланку, як секретар. Дозволить клієнтам слідкувати за актуальною інформацією про товари. Адміністратор зможе швидко отримувати та досліджувати дані про роботу всіх філіалів, а менеджерам вчасно реагувати на нові замовлення.

**1.2.2. Збір фактів.**

**1.2.2.1. Дослідження**

Було проаналізовано аналогічні АІС на ринку, зокрема:

• Комп’ютерний Дім   
• Gadgetarium  
• Solo Computers  
• Tech Team

**1.2.2.2. Спостереження**

В процесі спостереження за роботою організації було виявлено особливості роботи, не відображені в наданому керівництвом описі (п.1.2.1):

**-** звіти сформовані адміністраторами, також передаються аналітичному центру, який проводить глибший аналіз і будує стратегію пошуку клієнтів та заохочення їх до покупок.

- під час роботи виникають проблеми через секретаря, який може втратити нотатку з замовленням. очевидний висновок про неефективні методи роботи.

## 2. Технічний проект

## 2.1.Технічні вимоги

### **2.1.1.Опис груп користувачів АІС.**

Програмний застосунок призначений для обробки даних, що використовуються в процесі оформлення замовлення сервісних послуг клієнтами компанії. АІС має 4 групи користувачів: адміністратор, менеджери, кур’єри та клієнти.

Адміністратор не прив’язаний до жодного відділення і має доступ до перегляду всіх даних в БД, однак вносити зміни може лише в інформацію про послуги; додавати нові відділення, менеджерів. Щотижня формує звіт про діяльність кожного філіалу. на основі збору та дослідження інформації про замовлення, та передає керівництву компанії дані про статистику кожної майстерні, найбільш активних клієнтів, найпопулярніші товари, необхідні поповнення на складах тощо.

Кожна філія керується менеджерами. Менеджер виконує опрацювання нових замовлень: підтверджує у клієнта замовлення; обирає вільного в даний момент часу кур’єра, та надає йому всю потрібну інформацію(адреса, контакти, список замовлених товарів тощо); оновлює статус замовлення.

Кур’єри здійснюють доставку та проводять оплату. Вони можуть переглядати деталі отриманого замовлення та замовника.

Замовниками(клієнтами) виступають клієнти, з якими вже укладено договір про проведення сервісних робіт. Увійшовши в систему вони можуть лише переглядати та обирати потрібні їм послуги, додаючи їх до замовлення, та редагувати деякі персональні дані(наприклад, контакти представників).

**Опис вхідних даних:**

При реєстрації на сайті користувач повинен заповнити форму. Дані заносяться до користувацької бази. Приклад наведений в таблиці №1.

Таблиця №1. Форма реєстрації користувача мережі «Nerds Solution»

|  |  |
| --- | --- |
| Форма реєстрації користувача | |
| Ім’я |  |
| Прізвище |  |
| Email |  |
| Пароль |  |
| Найближче відділення |  |

Після занесення товарів до кошику, користувач оформлює замовлення, після якого заповнює спеціальну форму. Приклад форми в таблиці №2

Таблиця №2. Форма оформлення нового замовлення

|  |  |
| --- | --- |
| Форма оформлення замовлення | |
| Ім’я |  |
| Прізвище |  |
| Номер телефону |  |
| Коментар |  |
| E-mail |  |

До кожного клієнтського замовлення входить від одної до кількох послуг. Приклад форми редагування послуги продемонстрований в таблиці №3.

Таблиця №3. Форма створення послуги

|  |  |
| --- | --- |
| Послуга | |
| Назва |  |
| Вартість |  |
| Категорія |  |
| Наявність на складі |  |
| Новизна |  |
| Статус |  |

Кожна послуга належить певній категорії, яка має назву, порядковий номер і статус відображення. Приклад форми створення категорії в таблиці №4

Таблиця №4. Форма створення категорії

|  |  |
| --- | --- |
| Категорія | |
| Назва категорії |  |
| Порядковий номер |  |
| Статус відображення |  |

**2.1.2. Технічні вимоги користувачів до функціональності системи.**

АІС включає в себе 4 групи користувачів і для забезпечення функціональності потрібні такі запити:

*Клієнт:*

* Переглянути список доступних послуг
* Переглянути інформацію про окрему послугу
* Сформувати замовлення
* Додати послугу до замовлення
* Видалити послугу із замовлення
* Відправити замовлення
* Редагувати персональні дані
* Переглянути список покупок

*Адміністратор:*

* Переглянути список замовлень
* Переглянути список послуг
* Переглянути список категорій
* Переглянути список користувачів
* Додавати/видаляти/редагувати користувачів.
* Додавати/видаляти/редагувати замовлення.
* Додавати/видаляти/редагувати послуги.
* Додавати/видаляти/редагувати категорії.

*Майстер:*

* Переглянути список нових замовлень
* Взяти замовлення
* Видаляти/редагувати замовлення
* Переглянути інформацію про замовлення

*Кур'єр:*

* Переглянути список нових замовлень.
* Взяти замовлення.
* Переглянути інформацію про замовлення.
* Виконати замовлення.

**2.1.3. Специфікація вимог**

#### **2.1.3.1. Вимоги до даних.**

***Відділення:***

Мережа комп’ютерних майстерень «Nerds Solution» має багато відділень по Україні. Для позначення кожного відділення застосовуються унікальний ID, місто та адреса. Ідентифікатор, що присвоєно кожному відділенню, є унікальним в межах компанії.

***Адміністратор:***

Адміністратор може переглядати, редагувати та видаляти дані про користувачів, замовлення, послуги та категорії. Також він має право надавати користувачу статус робітника (кур’єр чи майстер).

***Майстер:***

Майстер може переглядати дані про замовлення, редагувати та видаляти їх. Також він приймає замовлення.

***Кур’єр:***

Кур’єр може переглядати дані про замовлення та приймати їх.

**2.1.3.2. Вимоги до роботи з даними**.

Введення:

* Ввести відомості про нового користувача
* Ввести відомості про нову замовлення
* Ввести відомості про нове послугу
* Ввести відомості про нову категорію

Редагування:

* Редагувати відомості про користувача
* Редагувати відомості про замовлення
* Редагувати відомості про послугу
* Редагувати відомості про категорію

Видалення:

* Видалити відомості про користувача.
* Видалити відомості про замовлення
* Видалити відомості про послугу
* Видалити відомості про категорії

Запити:

* Список всіх користувачів
* Список всіх працівників одного відділення
* Відділення, до якого належить даний працівник
* Список кур’єрів даного відділення
* Список усіх замовлень
* Список усіх категорій
* Список замовлень, створених клієнтом системи
* Список усіх послуг, що надаються майстернею
* Список усіх послуг, пов’язаних з поточним обслуговуванням
* Список усіх послуг, пов’язаних з заміною деталей
* Список усіх послуг, пов’язаних з ремонтом деталей
* Список усіх послуг, пов’язаних з чисткою комп’ютера
* Список усіх послуг, пов’язаних з налаштуванням софта
* Список усіх послуг, пов’язаних з доставкою замовлення

**2.2. Інфологічне та даталогічне проектування**

**(Інформаційне забезпечення)**

**2.2.1. Концептуальна модель даних (ручне проектування)**

**2.2.1.1. ER – модель даних**



**2.2.1.2. Загальні відомості**

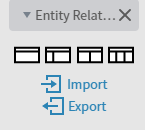
Концептуальна модель даних для АІС мережі гітарних майстерень «Hard Guitars» містить 7 сутностей: відділення, менеджери, майстри, кур’єри, клієнти, замовлення, послуги. Дана АІС містить зв’язки типу 1:n, m:n, 1: 0..n .

**2.2.2.Концептуальна модель даних(в обраній Case-системі)**

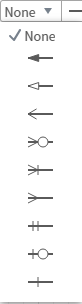
**2.2.2.1.Обрана Case-система  
2.2.2.1.1.Загальні відомості(назва, версія, адреса для завантаження)**

**LucidChart** — це орієнтоване на [веб](https://uk.wikipedia.org/wiki/%D0%A5%D0%BC%D0%B0%D1%80%D0%BD%D1%96_%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F) [програмне забезпечення](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F), призначене для створення та редагування [діаграм](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0). LucidChart дозволяє користувачам об’єднуватися та працювати разом над створенням [блоксхем](https://uk.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%BA-%D1%81%D1%85%D0%B5%D0%BC%D0%B0), організаційних діаграм, шаблонів веб-сайтів, [UML-діаграм](https://uk.wikipedia.org/wiki/Unified_Modeling_Language), [ментальних карт](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D0%BD%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%BA%D0%B0%D1%80%D1%82%D0%B0), прототипів програмного забезпечення та багатьох інших типів діаграм. Програма побудована на веб-стандартах [HTML5](https://uk.wikipedia.org/wiki/HTML5) та [Javascript](https://uk.wikipedia.org/wiki/Javascript), що підтримуються усіма новими браузерами, такими як [Google Chrome](https://uk.wikipedia.org/wiki/Google_Chrome), [Firefox](https://uk.wikipedia.org/wiki/Firefox), [Safari](https://uk.wikipedia.org/wiki/Safari_(web_browser)) та [Internet Explorer](https://uk.wikipedia.org/wiki/Internet_Explorer) 8+. Lucidchart – провідна програма для побудови діаграм та візуалізації в інтернеті. Остання версія: 21.2.9. Створювати необхідні схеми можна на сайті [www.lucidchart.com](http://www.lucidchart.com), створивши акаунт. Повний функціонал доступний лише користувачам, які купують платну підписку. Без підписки користувач може використовувати базові можливості, яких достатньо для повноцінної роботи.

**2.2.2.1.2.Позначення(опис всіх позначень, що використовуються)**

В системі є 4 типи контейнерів. Для побудови ER-моделі я використовую перший тип, який ділиться лінією на дві частини.

Всі ключі вказано у верхній частині контейнера, а решта атрибутів у нижній.

****В Lucidchart зручно працювати з будь-якими зв’язками.

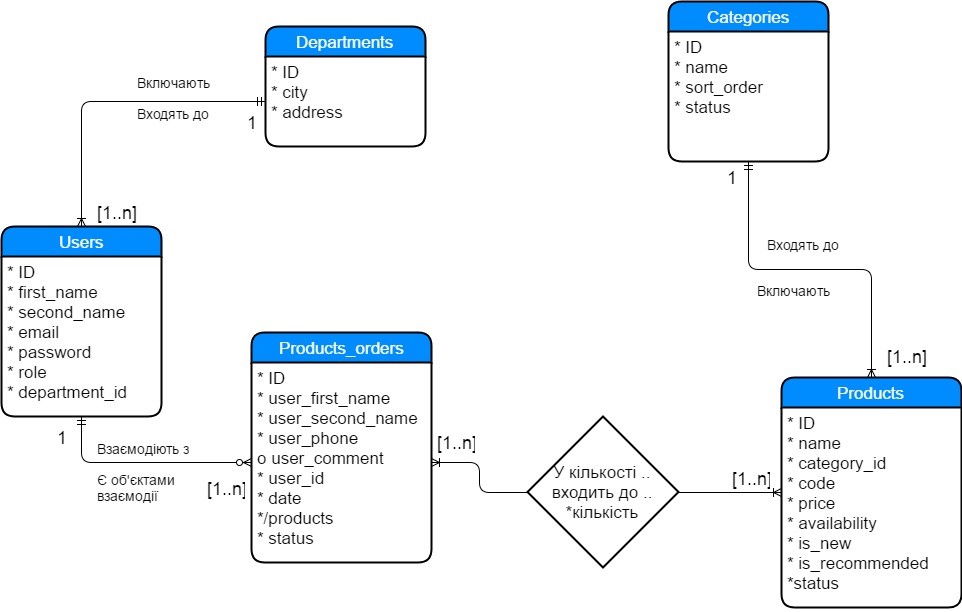
Можна змінювати товщину ліній та встановлювати необхідне позначення. Також можна поміняти місцями тип зв’язку між двома контейнерами.

Будь-який зв’язок можна підписати. Контейнери можна розмалювати в будь-який колір для кращого візуального сприйняття.

**2.2.2.1.3.Опис наявних обмежень та недоліків**

Основний недолік Case-системи полягає у відсутності багатьох необхідних позначень для атрибутів, зокрема: похідних, багатозначних, обов’язкових/необов’язкових). Для позначення необов’язкових атрибутів використано символ °.

**2.2.2.2.Побудова ER-моделі в заданій CASE-системі**

****

**2.2.3. Реляційна модель даних**

**2.2.3.1. Опис реляційної моделі.**

***Опис Сутностей***

|  |  |
| --- | --- |
| Сутність | Значення |
| Користувачі | * Користувачі системи |
| Послуги | * Пропонуються майстерньою, заносяться до замовлення |
| Категорії | * Включають до себе послуги |
| Замовлення | * Створюються клієнтами, виконуються майстрами, доставляються кур'єрами |
| Відділення | * Окремі заклади, що належать мережі |

***Опис зв’язків:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сутність №1 | Сутність №2 | Ім’я зв’язку | Тип зв’язку | Клас належності |
| *Категорії* | *Послуги* | Включають/входять до | 1:n | Обов’язк./Обов’язк. |
| *Відділення* | *Користувачі* | Включають/відносяться до | 1:n | Обов’язк./Обов’язк. |
| *Користувачі* | *Замовлення* | Взаємодіє з / є об’єктом взаємодії | 1:n | Обов’язк./Необов’язк. |
| *Замовлення* | *Послуги* | “У кількості … входить до …” | m:n | Обов’язк./Обов’язк. |

***Реляційна модель:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Таблиця №1* Відношення «Відділення» | | | | | |
| № | Назва сутності | Тип | Обов'язковість | Цілісність | Примітки |
| 1 | PK\_ID Відділення | counter | NOT NULL |  |  |
| 2 | Місто | String(50) | NOT NULL |  |  |
| 3 | Вулиця | String(50) | NOT NULL |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Таблиця №2* Відношення «Користувачі» | | | | | |
| № | Назва сутності | Тип | Обов'язковість | Цілісність | Примітки |
| 1 | PK\_ID Користувача | counter | NOT NULL |  |  |
| 2 | Ім'я | String(50) | NOT NULL |  |  |
| 3 | Прізвище | String(50) | NOT NULL |  |  |
| 4 | Email | String(30) | NOT NULL |  |  |
| 5 | Пароль | Integer(22) | NOT NULL |  |  |
| 6 | Роль | String(20) | NOT NULL |  |  |
| 7 | FK ID відділення | Integer(12) | NOT NULL |  | Зв’язок (відносяться до) [1..n] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Таблиця №3* Відношення «Послуги» | | | | | |
| № | Назва сутності | Тип | Обов'язковість | Цілісність | Примітки |
| 1 | PK\_ID послуги | counter | NOT NULL |  |  |
| 2 | Назва | String(40) | NOT NULL |  |  |
| 3 | Ціна | integer | NOT NULL |  |  |
| 4 | Наявність | Integer(11) | NOT NULL |  |  |
| 5 | Новизна | Integer(11) | NOT NULL |  |  |
| 6 | Рекомендації | Integer(11) | NOT NULL |  |  |
| 7 | Статус | Integer(11) | NOT NULL |  |  |
| 8 | FK ID Категорії | Integer(11) | NOT NULL |  | Зв'язок (відносяться) [1..n] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Таблиця №4* Відношення «Категорії» | | | | | |
| № | Назва сутності | Тип | Обов'язковість | Цілісність | Примітки |
| 1 | PK ID Категорії | counter | NOT NULL |  |  |
| 2 | Назва | String(50) | NOT NULL |  |  |
| 3 | Порядок в списку | Integer(11) | NOT NULL |  |  |
| 4 | Статус | Integer(11) | NOT NULL |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Таблиця №5* Відношення «Замовлення» | | | | | |
| № | Назва сутності | Тип | Обов'язковість | Цілісність | Примітки |
| 1 | PK\_ID Замовлення | counter | NOT NULL |  |  |
| 2 | Ім'я клієнта | String(30) | NOT NULL |  |  |
| 3 | Прізвище клієнта | String(30) | NOT NULL |  |  |
| 4 | Номер телефону клієнта | Integer(12) | NOT NULL |  |  |
| 5 | Комментарій | String(140) | NULL |  |  |
| 6 | Дата замовлення | date | NOT NULL |  |  |
| 7 | Статус | Integer(11) | NOT NULL |  |  |
| 8 | FK ID клієнта | counter | NOT NULL |  | Зв’язок(оформляє/оформляється) [1..n] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Таблиця №6* Відношення «Замовлення-послуги» | | | | | |
| № | Назва сутності | Тип | Обов'язковість | Цілісність | Примітки |
| 1 | FK\_PK\_ID замовлення | counter | NOT NULL | ON DELETE CASCADE  ON UPDATE CASCADE | Зв'язок 1..n |
| 2 | FK\_PK\_ID послуги | counter | NOT NULL | ON DELETE CASCADE  ON UPDATE CASCADE | Зв'язок 1..n |
| 3 | Кількість | Integer | NOT NULL |  |  |

* + - 1. **Визначення обмежень цілісності.**
         1. **. Обмеження домену**

*Обмеження домену* – набір допустимих значень одного або кількох атрибутів.

Тип Autoincrement використаний для нумерації позицій товарів в списку.

Атрибути Deliver\_Date i Reg\_Date мають тип даних DateTime, що був користаний для встановлення дати і часу замовлення.

VarChar(n) використаний для атрибутів з варіативною кількістю символів.

А Char(n) для типізації атрибутів, в яких кількість символів стала та відома. В основному це унікальні ідентифікатори визначені законодавчими актами.

Обов’язковість наявності значення кожного атрибуту можна подивитись в попередній табличці в колонці «*NULL/NOT NULL*».

* + - * 1. **Цілісність сутностей (дати визначення та посилання на відповідні комірки в таблиці).**

Кожна сутність ER-моделі представлена реляцією з аналогічними атрибутами, у яких вказана обов’язковісь, описана формула обчислення у похідних атрибутах(колонка «*Пояснення (перехід від ER-моделі)*»), визначено ключі для реляції(колонка «*Ключ*»).

Відповідність між відношеннями реляційної моделі та сутностями ER-моделі можна подивитись в табличці.

* + - * 1. **Цілісність посилань (дати визначення та посилання на відповідні комірки в таблиці).**

Цілісність посилань можна подивитись в табличці у колонці «*FK: ON DELETE, ON UPDATE*»

* + - * 1. **Корпоративні (семантичні) обмеження цілісності.**
* Дата оформлення замовлення та дата доставки повинна бути однієї, час поставки може відрізнятись.
* Кур'єр не може отримати нове завдання від менеджера, поки не завершить активне замовлення.
* Дата та час оформлення замовлення клієнта повинен бути пізніше ніж дата та час реєстрації клієнта (при введенні/уточненні).
* Дата виконання замовлення не може бути раніше ніж дата оформлення замовлення (при введенні/уточненні)
* Дата оформлення замовлення та дата доставки повинні бути однаковими,час доставки повинен відрізнятись.
* Початкове значення атрибуту «Реалізований дохід» дорівнює 0.00 грн

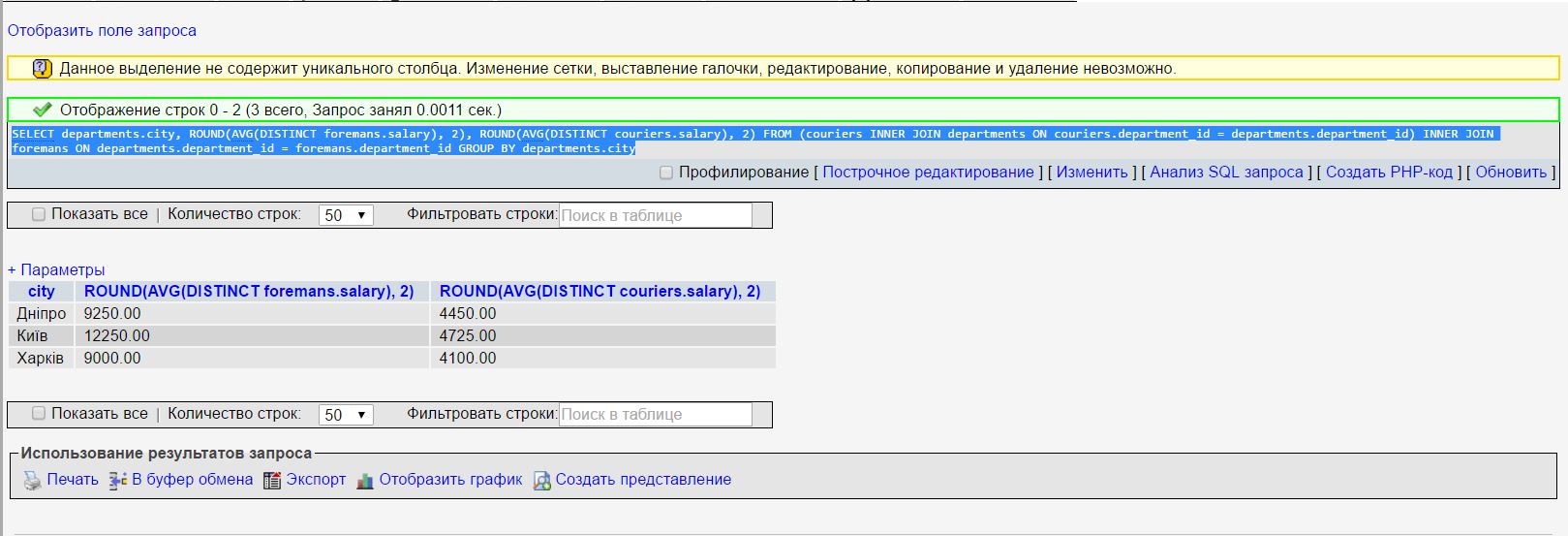
**2.2.4 Запити**

*1)* Вивести коди замовлень та код клієнтів, що замовляли заміну екрану

Код:

SELECT order\_id, client\_id  
FROM orders  
WHERE order\_id IN(

SELECT order\_id  
FROM orders\_services  
WHERE service\_id IN(

SELECT service\_id  
FROM services  
WHERE services.name = "Заміна екрану"));

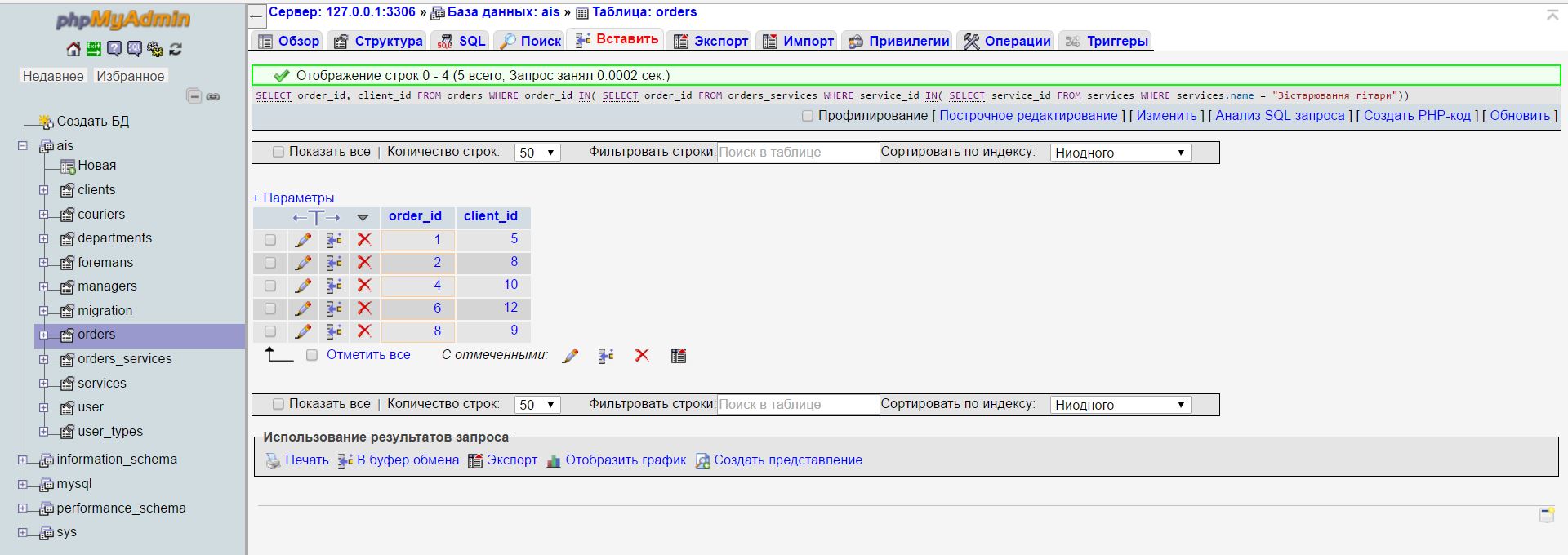
**2)Вивести для кожного міста середні значення зарплат майстрів та кур’єрів, що працюють в цих містах**

*Код:*

SELECT departments.city, ROUND(AVG(DISTINCT foremans.salary), 2), ROUND(AVG(DISTINCT couriers.salary), 2)

FROM (couriers INNER JOIN departments ON couriers.department\_id = departments.department\_id)

INNER JOIN foremans ON departments.department\_id = foremans.department\_id

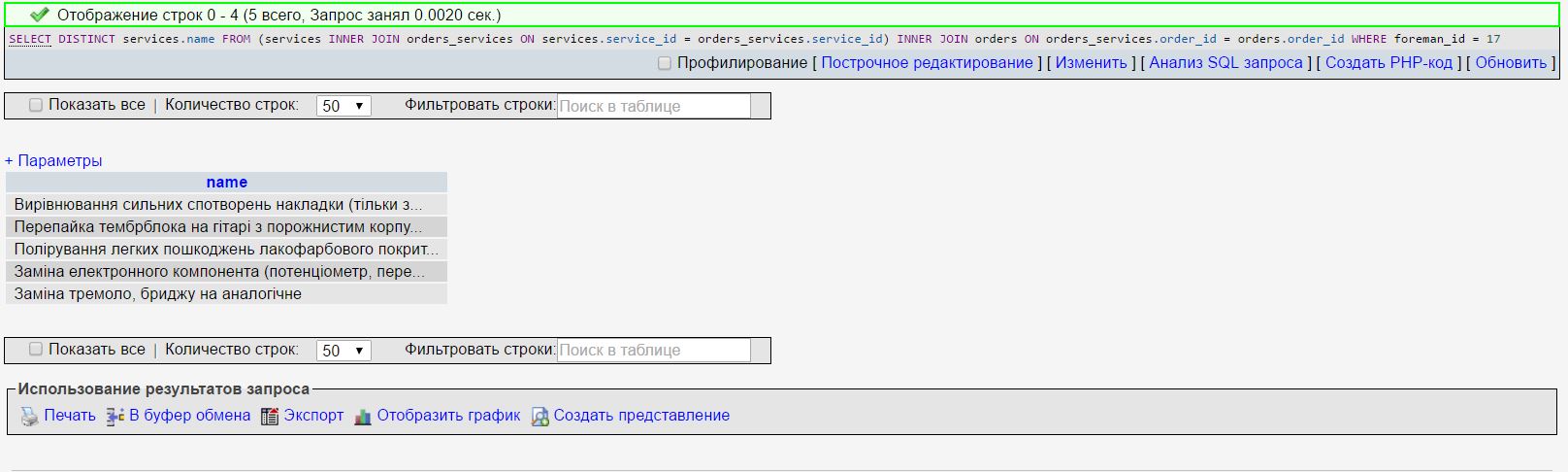
****GROUP BY departments.city;

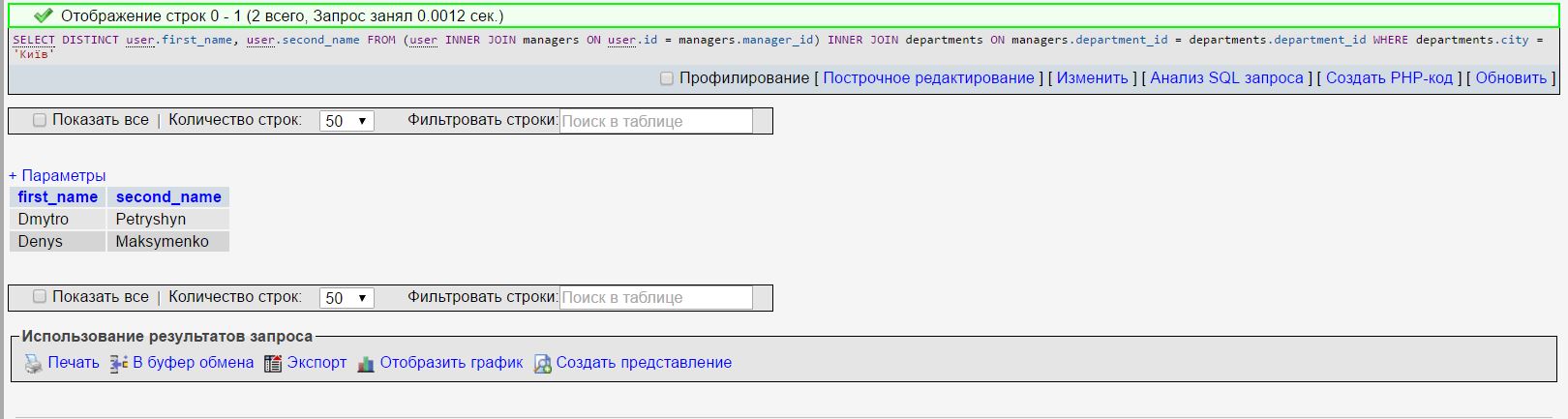
3)Вивести всі послуги, які є у всіх тих замовленнях, що і послуги `Чистка ноутбука`

Код:

SELECT \* FROM services  
WHERE NOT EXISTS (  
 SELECT \* FROM orders   
 WHERE NOT EXISTS (  
 SELECT \* FROM orders\_services  
 WHERE orders\_services.service\_id = services.service\_id AND orders\_services.order\_id = orders.order\_id) AND orders.order\_id IN ( SELECT orders\_services.order\_id

FROM services INNER JOIN orders\_services ON services.service\_id = orders\_services.service\_id

 WHERE services.name = 'Чистка ноутбука'))

4. Вивести назви послуг, які виконував певний майстер

SELECT DISTINCT services.name  
FROM (services INNER JOIN orders\_services ON services.service\_id = orders\_services.service\_id)   
INNER JOIN orders ON orders\_services.order\_id = orders.order\_id  
WHERE foreman\_id = [Put foremans id];

5. Вивести ім’я та прізвище менеджерів, що живуть в певному місті

SELECT DISTINCT user.first\_name, user.second\_name

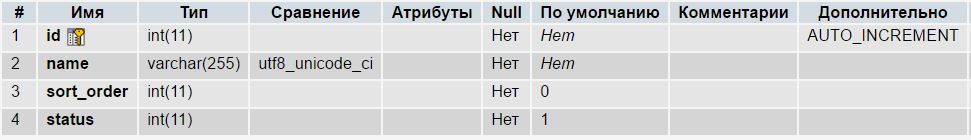
FROM (user INNER JOIN managers ON user.id = managers.manager\_id)

INNER JOIN departments ON managers.department\_id = departments.department\_id

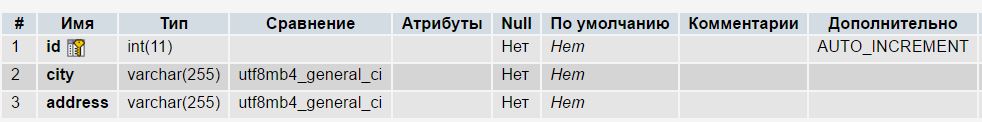
WHERE departments.city = [Put the city];

**2.2.5 Нормалізація**

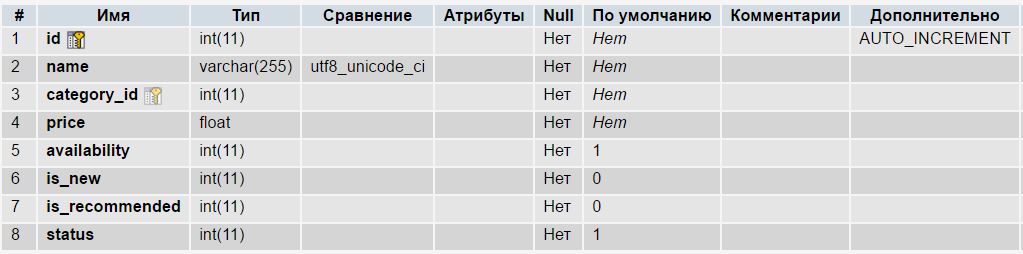
**2.2.5.1. Перевірка на 3НФ**

1) Реляція categories

Квазі ключем є id. Дана реляція має наступну функціональну залежність: {id -> name, id -> sort\_order, id -> status }. Всі атрибути реляції є атомарним, а тому вона знаходяться у 1НФ. Реляція знаходиться у 2НФ, бо кожен не первинний атрибут функціонально повно залежить від квазі ключа. Реляція не має функціональних залежностей не від ключа, а тому вона містить лише нетранзитивні залежності,. Тому реляція categories знаходиться в 3НФ.

2) Реляція departments

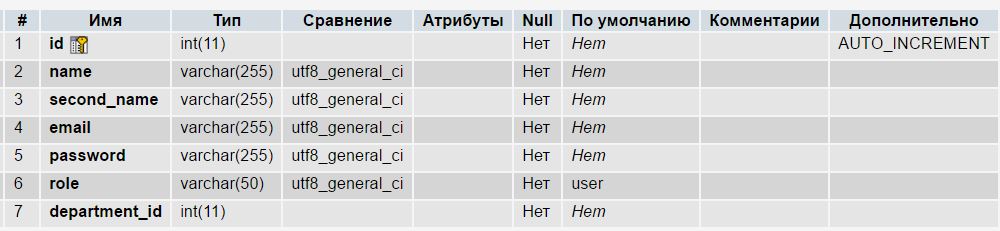
Квазі ключем є id. Дана реляція має наступну функціональну залежність: {id -> city, id -> address }. Всі атрибути реляції є атомарним, а тому вона знаходяться у 1НФ. Реляція знаходиться у 2НФ, бо кожен не первинний атрибут функціонально повно залежить від квазі ключа. Реляція не має функціональних залежностей не від ключа, а тому вона містить лише нетранзитивні залежності,. Тому реляція departments знаходиться в 3НФ.

3) Реляція products

Квазі ключем є id. Дана реляція має наступну функціональну залежність: {id -> name, id -> category\_id, id -> price, id ->availability, id->is\_new, id->is\_recommended, id->status }. Всі атрибути реляції є атомарним, а тому вона знаходяться у 1НФ. Реляція знаходиться у 2НФ, бо кожен не первинний атрибут функціонально повно залежить від квазі ключа. Реляція не має функціональних залежностей не від ключа, а тому вона містить лише нетранзитивні залежності,. Тому реляція departments знаходиться в 3НФ.

4) Реляція products\_orders

Квазі ключем є id. Дана реляція має наступну функціональну залежність: {id -> user\_name, id -> user\_second\_name, id->user\_phone, id->user\_comment, id->user\_id, id->foremen\_id, id->courier\_id, id->date, id->products, id->status}. Всі атрибути реляції є атомарним, а тому вона знаходяться у 1НФ. Реляція знаходиться у 2НФ, бо кожен не первинний атрибут функціонально повно залежить від квазі ключа. Реляція не має функціональних залежностей не від ключа, а тому вона містить лише нетранзитивні залежності,. Тому реляція products\_users знаходиться в 3НФ.

5) Реляція users

Квазі ключем є id. Дана реляція має наступну функціональну залежність: {id -> name, id -> second\_name, id->email, id->password, id->role, id->department\_id }. Всі атрибути реляції є атомарним, а тому вона знаходяться у 1НФ. Реляція знаходиться у 2НФ, бо кожен не первинний атрибут функціонально повно залежить від квазі ключа. Реляція не має функціональних залежностей не від ключа, а тому вона містить лише нетранзитивні залежності,. Тому реляція users знаходиться в 3НФ.

**D.Висновки**

Спостереження за роботою компанії та вивчення документації дало змогу оцінити становише та зробити висновки про те, яким чином можна покращити роботу компанії, а точніше зробити ефективнішим процес від оформлення замовлення клієнтом та його подальше опрацювання співробітниками компанії Саме цю можливість повинна забезпечувати АІС. Однак не включає дані про прийом товарів на складах, оскільки зі слів генерального директора на складах підтримується стала кількість продукції, кількість якої залежить від статистики продаж, і нові поставки відбуваються кожного тижня.

З ключових моментів, які найбільш тривожили персонал, стало очевидним що компанія працює по застарілій схемі оптового продажу мастильної продукції. Побудована ER-модель спрощує процес оформлення та опрацювання персоналом замовлень. База даних повинна містити мінімум надлишкової інформації. Передбачається що кожна філія зберігає дані про замовлення протягом тижня, після чого вони видаляються, а завдяки синхронізації всіх баз даних компанії, їхня копія зберігатиметься у базі звітів.

Під час виконання роботи було побудовано декілька запитів, які демонструють маніпуляцію даних з бази даних. Також було перевірено та виявлено, що кожна реляція знаходиться у 3НФ.

**E. Додатки**

1.Фінансові звіти компанії за 2012-2016

|  |  |  |
| --- | --- | --- |
| **Рік** | **Виручка, грн.** | **Державні закупівлі, грн.** |
| 17 |  | 5 425 102 |
| 2016 | 150 000 000 - 200 000 000 | 11 212 260 |
| 2015 | 150 000 000 - 200 000 000 | 6 382 871 |
| 2014 | 55 000 000 - 60 000 000 | 1 555 081 |
| 2011 | до 100 000 |  |

**F. Література**

1. Ульман Дж. Основы системы баз данных. Финансы и статистика. 1983.
2. Гайдаржи В.І., Дацюк О.А. Основи проектування та використання баз даних: Навчальний посібник. К. 2004.
3. Райордан Р. Основы реляционных баз данных. М.: Русская редакция. 2001.
4. Дейт К. Дж. Введение в системы баз данных, 8-е издание. Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил. ISBN 5-8459-0788-8 .
5. Date C.J. / Дейт К.Дж. - SQL and Relational Theory / SQL и Теория реляционных баз данных . 2-е издание, 2011.
6. «Еnterprise Architectures and Digital Administration: Planning, Design and Assessment» Автор: Ambrose Goikoetxea
7. «Information Technology Auditing» Автор: James A. Hall