

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра інформаційних систем та технологій

Тема: Системний блок

Курсова робота  
З дисципліни «Бази даних»

Керівник

Ульяницька К. О.

«Допущений до захисту»

\_\_\_\_\_

(Особистий підпис керівника)

« » \_\_\_\_\_ 2021р.

Захищений з оцінкою

\_\_\_\_\_

(оцінка)

Члени комісії:

\_\_\_\_\_

(особистий підпис)

\_\_\_\_\_

(особистий підпис)

Виконавець

ст. Панченко В. П.

залікова книжка № ІА – 0323

гр. ІА-03

\_\_\_\_\_

(особистий підпис виконавця)

« » \_\_\_\_\_ 2021р.

\_\_\_\_\_

(розшифровка підпису)

\_\_\_\_\_

(розшифровка підпису)

Київ – 2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

(назва навчального закладу)

Кафедра ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Дисципліна «Бази даних»

Курс 2 Група ІА-03 Семестр 1

**ЗАВДАННЯ**

**на курсову роботу студента**

Панченка Віталія Павловича

(прізвище, ім'я, по батькові)

1. Тема роботи: «Системний блок»
2. Строк здачі студентом закінченої роботи : 22.12.21
3. Вихідні дані до роботи: предметна область «Системний блок», перелік питань, на які повинна відповідати база даних: Вивести інформацію про різні системні блоки, існуючі компоненти системних блоків, користувачів, продавців, запити користувачів, контракти, що укладались під час купівлю, склад окремих системних блоків.
4. Зміст розрахунково – пояснювальної записки (перелік питань, що підлягають розробці): постановка задачі, огляд існуючих рішень, визначення вимог до проекту бази даних, інфологічне проектування, визначення вимог до оперативної обстановки, вибір конкретної СУБД, даталогічне проектування, фізичне проектування.  
**Додатки:** Додаток А – Встановлення СУБД Oracle, Додаток Б – Скрипти для створення таблиць, послідовностей і т.д., Додаток В – Відпрацювання скриптів у СУБД, Додаток Г – Заповнення БД даними
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): ER-діаграма інфологічної моделі, даталогічна модель, фізична модель, встановлення СУБД, скрипти SQL, відпрацювання скриптів
6. Дата видачі завдання: 15.09.2021

## КАЛЕНДАРНИЙ ПЛАН

№, п/п	Назва етапів виконання курсової роботи	Строк виконання етапів роботи	Підписи або примітки
1.	Дата видачі завдання	15.09.21	
2.	Аналіз предметної області	15.10.21	
3.	Вступ, Огляд існуючих рішень	12.12.21	
4.	Визначення вимог до проекту баз даних	12.12.21	
5.	Інфологічне проектування	12.12.21	
6.	Визначення вимог до оперативної обстановки	12.12.21	
7.	Вибір конкретної СУБД	22.12.21	
8.	Даталогічне проектування	22.12.21	
9.	Фізичне проектування	22.12.21	
10.	Висновки	22.12.21	
11.	Список використаних джерел	22.12.21	
12.	Додатки	22.12.21	
13.	Захист курсової роботи	24.12.21	
14.			
15.			
16.			
17.			
18.			

Студент \_\_\_\_\_  
(підпис)

Панченко Віталій Павлович  
(прізвище, ім'я, по батькові)

Керівник \_\_\_\_\_  
(підпис)

Ульяницька Ксенія Олександрівна  
(прізвище, ім'я, по батькові)

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

## ЗМІСТ

ВСТУП.....	5
1. Постановка задачі.....	6
1.1 Аналіз предметної області.....	6
1.2 Побудова моделі предметної області .....	9
1.3 Визначення меж інформаційної підтримки.....	12
1.4 Зовнішні специфікації функцій .....	12
1.5 Групи користувачів бази даних .....	13
2. Огляд існуючих рішень .....	14
3. Визначення вимог до проекту бази даних .....	15
4. Інфологічне проектування .....	16
5. Визначення вимог до операційної обстановки .....	17
6. Вибір конкретної СУБД .....	18
6.1 Перетворення концептуальної моделі в схему БД .....	18
6.1.1 Формалізація зв'язків .....	20
6.1.2 Введення обмежень цілісності та визначення типів даних ....	21
6.1.3 Нормалізація бази даних .....	24
7. Даталогічне проектування .....	25
8. Фізичне проектування .....	26
8.1 Розробка скриптів під конкретну СУБД (DDL) .....	26
8.2 Відпрацювання скриптів (створення фізичної бази даних) .....	28
8.3 Фізична модель даних .....	30
8.4 Тестові дані (DML) .....	31
ВИСНОВКИ.....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	40
ДОДАТКИ.....	42
Додаток А.....	42
Додаток Б .....	44
Додаток В.....	49
Додаток Г .....	50

## ВСТУП

Протягом довгих років вчені досліджували мозок людини, навіть продовжують це робити зараз, і кожного разу вони «ламають голову» над тим, як же прекрасно і складно він побудований і скільки секретів він в собі тримає. Одне зрозуміло точно: всередині людини знаходиться надзвичайно потужний багатофункціональний комп'ютер. Завжди людина намагатиметься створити щось подібне, і, хоча, на мою думку, невдача заздалегідь передбачена, дослідження навряд припиняться. Тільки об'єм інформації, що може зберігатись у мозку однієї людини, - вже вражає. Поділ інформації на короткострокову і довгострокову, зберігання її в голові – це про людський надзвичайний невеликий контейнер із великим вмістом .

Проте людина, все-таки, не може зберігати всю інформацію, адже частина завжди відсіюється або банально «забувається», що, в принципі, значить те ж. Тому існує така дисципліна як «бази даних», яка є однією з найважливіших і, можна сказати, навіть базовою при роботі з даними. Люди працюють з безліччю цінної інформації щодня і цю інформацію бажано не забувати, тому цією задачею займаються різні електронні пристрої типу комп'ютерів. Для того, щоб дані зберігались не хаотично, а організовано, було винайдено поняття «баз даних», де дані логічно взаємопов'язані.

Згідно із даною темою роботи, наступна база даних буде зберігати інформацію, пов'язану із системними блоками.

# 1 ПОСТАНОВКА ЗАДАЧІ

## 1.1 Аналіз предметної області

База даних створюється для інформаційного обслуговування всіх тих, хто цікавиться системними блоками, цікавиться продуктивністю роботи конкретного комп'ютера та прагне зробити віртуальний «комп'ютер мрії». База даних повинна містити інформацію про запити користувачів, про існуючі комп'ютерні компоненти, про сумісність деталей, про базовий склад системного блоку, користувачів та готові «збірки» (тобто готовий системний блок, зібраний з різних компонентів), так само як і про запити і контракти.

Відповідно до предметної області система будується з врахуванням наступних особливостей:

- Деякі деталі можуть не підходити один одному (наприклад по габаритам);
- Дані по «саморобним збіркам» часом бувають приблизні, тому що не завжди тестувались вживу. Тобто користувач створює віртуальний комп'ютер і, часом, такий існує справді, а, часом, ні;
- Дані можуть змінюватись, враховуючи, що комп'ютери постійно тестуються новим програмним забезпеченням;
- Системний блок містить основні та другорядні компоненти.

Після аналізу предметної області можна виділити ряд сутностей, приступити до проектування інфологічної моделі.

Виділимо базові сутності цієї предметної області:

- **Компоненти** системного блоку. Атрибути компонентів: ID, ID Запитів, Тип, категорія (основний\другорядний), назва (модель), рік випуску, ціна, виробник, гарантійний строк, характеристичний опис, рейтинг компоненту, наявність компонента;

- **Готові «збірки».** Атрибути «збірок»: ID, ID Запитів, Назва, ціна, гарантійний строк, рік збірки, характеристичний опис, рейтинг «збірки», автор «збірки» (за наявності), наявність «збірки»;
- **Користувачі.** Атрибути користувачів: ID, прізвище, ім'я, по-батькові, вік;
- **Продавці.** Атрибути продавців: ID, назва, реквізити, контактні дані, рейтинг;
- **Запити.** Атрибути запитів: ID, ID користувача, Тип запиту (готові «збірки»\компоненти системного блоку), дата запиту;
- **Контактні дані.** Атрибути контактних даних: ID, ID користувача, електронна пошта, адреса;

Винесемо у окрему таблицю для зручності:

Таблиця 1.1 – Сутності та їх атрибути

Сутності	Атрибути
<i>Компоненти</i>	ID, ID Запитів, Тип, категорія (основний\другорядний), назва (модель), рік випуску, ціна, виробник, гарантійний строк, характеристичний опис, рейтинг компоненту, наявність компонента;
<i>Готові «збірки»</i>	ID, ID Запитів, Назва, ціна, гарантійний строк, рік збірки, характеристичний опис, рейтинг «збірки», автор «збірки» (за наявності), наявність «збірки»;
<i>Користувачі</i>	ID, прізвище, ім'я, по-батькові, вік;
<i>Продавці</i>	ID, назва, реквізити, контактні дані, рейтинг;
<i>Запити</i>	ID, ID користувача, Тип запиту (готові «збірки»\компоненти системного блоку), дата запиту;
<i>Контактні дані</i>	ID, ID користувача, електронна пошта, адреса.

Готові «збірки» поєднані з компонентами, адже складаються з цих компонентів і утворюють унікально «зібраний» системний блок. В свою чергу користувачів, продавців та запити поєднують контракти у випадку, якщо людина не просто зацікавлена в певній інформації, але і бажає придбати конкретну деталь чи «збірку».

Таблиця 1.2 – Зв'язки та їх сутності

Зв'язки	Сутності
<i>Склад готової «збірки»</i>	«Готові «збірки»», «Компоненти»
<i>Контракти</i>	«Користувачі», «Продавці», «Запити»

Атрибути складу: ID, ID «збірки», ID компонента.

Атрибути контрактів: ID, ID Користувача, ID Продавця, ID Запиту, Номер контракту.



## 1.2 Побудова моделі предметної області

Класифікуємо сутності, атрибути та зв'язки для полегшення розуміння цілої картини:

Таблиця 1.3 – Класифікація сутностей

Вид сутностей	Сутності
<i>Стержнева</i>	Компоненти, Готові «збірки», Користувачі, Продавці
<i>Асоціативна (слабка)</i>	Запити
<i>Характеристична (слабка)</i>	
<i>Позначуюча (слабка)</i>	

Таблиця 1.4 – Класифікація зв'язків

Зв'язок	Вид зв'язка
<i>Склад готової «збірки»</i>	Безумовний
<i>Контракти</i>	Умовний

Таблиця 1.5 – Класифікація атрибутів

Вид атрибутів	Атрибути
<i>Описовий</i>	Назва (модель) компонента, виробник, тип компонента, категорія компонента, ціна компонента, гарантійний строк компонента, габарити компонента, характеристичний опис компонента, рейтинг компонента, наявність компонента, назва «збірки», ціна «збірки», гарантійний строк «збірки», габарити «збірки», колір «збірки», рік «збірки», наявність «збірки», ПІБ користувача, електронна пошта користувача, вік користувача, адреса користувача, назва фірми-продавця, реквізити продавця, контактні дані продавця, рейтинг продавця, тип запиту, дата запиту
<i>Вказівний</i>	ID, автор «збірки», номер контракту
<i>Допоміжний</i>	ID користувача, ID «збірки», ID компонента, ID продавця, ID запиту

ER-діаграма (нотація Чена) інфологічної моделі з атрибутами:

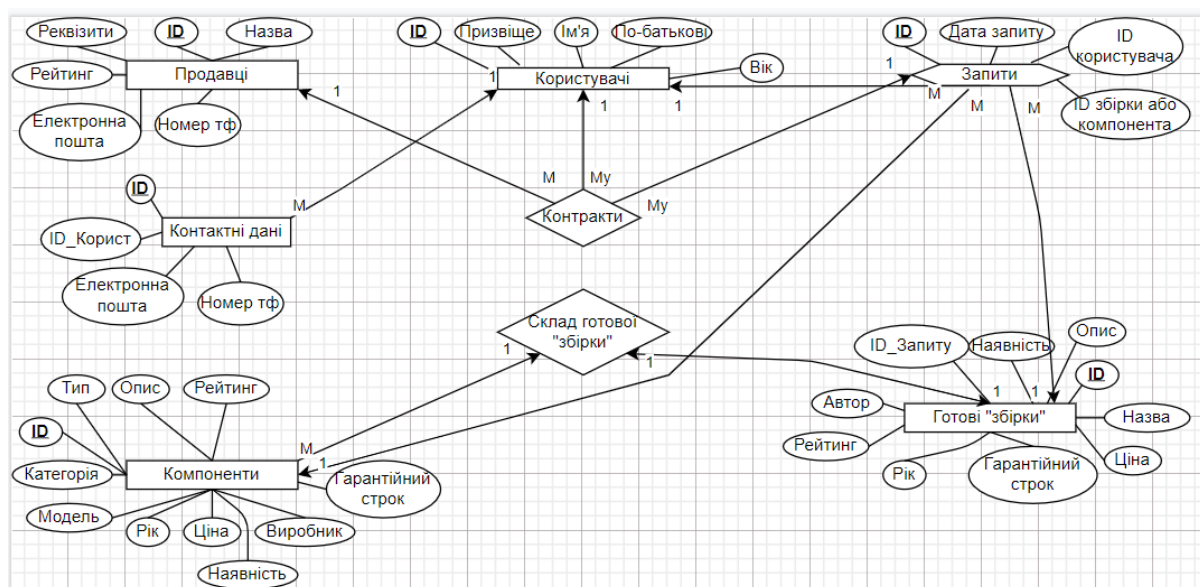


Рисунок 1.6 – ER-діаграма з атрибутами

### **1.3 Визначення меж інформаційної підтримки**

База даних створена на початку існування сайту, адже містить більшу частину необхідної інформації і є невід'ємною частиною фінальної картини. База може доповнюватись разом із появою на ринку нових компонентів, що особливо переплітається з відеокартами та процесорами, з появою нових «збірок», появою нового програмного забезпечення, що також означає нові комп'ютерні тестування.

Також матеріал може бути вилучений з бази даних, якщо інформація не є актуальною або цікавою користувачам. Може бути вилучена як ціла сутність, так і всі дані сутності, за наявності певної помилки, або лише певні рядки.

Є можливість переглянути інформацію про певний атрибут з певним значенням або відразу декілька атрибутів. Інформація може бути оновлювана як і в цілому, так і окремі її частини.

## 1.4 Зовнішні специфікації функцій

Перелік функцій для реалізації:

- Функції для роботи із розважальним контентом;
- Функції для роботи з контрактами;
- Функції для роботи із пізнавальним контентом.

Перелік питань, на які повинна відповідати база даних:

- Виведення інформації про користувачів;
- Виведення інформації про компоненти та цілісні системні блоки;
- Виведення інформації про продавців;
- Виведення інформації про укладені контракти;
- Виведення інформації про здійснені запити;
- Виведення інформації про конкретного користувача, конкретний компонент\збірку\продавця\контракт\запит, звертаючись до ідентифікатора чи будь-якого іншого атрибута сутності або ж за певною ознакою\умовою

### **1.5 Групи користувачів бази даних**

Ось невеликий перелік груп користувачів, що можуть бути зацікавлені в користуванні даною базою даних:

- Адміністратори;
- Зацікавлені в інформації користувачі;
- Потенційні покупці;
- Продавці;
- Розробники ПЗ та «заліза».

## 2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Головна задача – знайти інформаційний ресурс, що реалізує більшу частину функцій, що заплановано для реалізації в даній БД.

Найлегшим завданням було знайти ресурс, що надає інформацію про конкретні системні блоки та їх компоненти. В принципі, більшість інтернет-магазинів надають інформацію про системні блоки та їх технічні характеристики. Такими, наприклад, є популярні в Україні магазини Comfy та, наприклад, Rozetka. Проте такий інформаційний ресурс не може дати інформацію щодо самостійно створених моделей, навіть приблизних. Також не дає інформацію про основні деталі, з яких повинен складатись будь-який ПК.

Ресурс intel.ru, наприклад, надає інформацію як створити ігровий ПК, тобто основні компоненти ігрового ПК. Проте цей ресурс вже не містить інформацію попереднього.

Існують різні ресурси, що визначають приблизну продуктивність певного набору компонентів в сумісності з різним програмним забезпеченням, проте не містять інформацію попередніх ресурсів.

Отже, наша база даних допоможе створити унікальний сайт, що матиме функціональність попередніх і навіть трішки більше.

### **3 ВИЗНАЧЕННЯ ВИМОГ ДО БАЗИ ДАНИХ**

Вимоги до бази даних: Простота внесення, видалення, пошуку даних, зручність використання, нормалізованість. Суть створити не просто інформаційний сервіс, але зручний веб-сайт.

Перелік питань, на які повинна відповідати база даних:

- Виведення інформації про користувачів;
- Виведення інформації про компоненти та цілісні системні блоки;
- Виведення інформації про продавців;
- Виведення інформації про укладені контракти;
- Виведення інформації про здійснені запити;
- Виведення інформації про конкретного користувача, конкретний компонент\збірку\продавця\контракт\запит, звертаючись до ідентифікатора чи будь-якого іншого атрибута сутності або ж за певною ознакою\умовою



## 4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

Для зручності опрацювання даних в пункті 1.2 було наведено ER-діаграму (рис. 1.6), що представляє собою втілення інфологічної моделі на схемі, що показує зв'язки різних сутностей та їх атрибути. Використовується нотація Чена, на рисунку вказано види сутностей, види зв'язків, види зв'язних сутностей та види атрибутів.

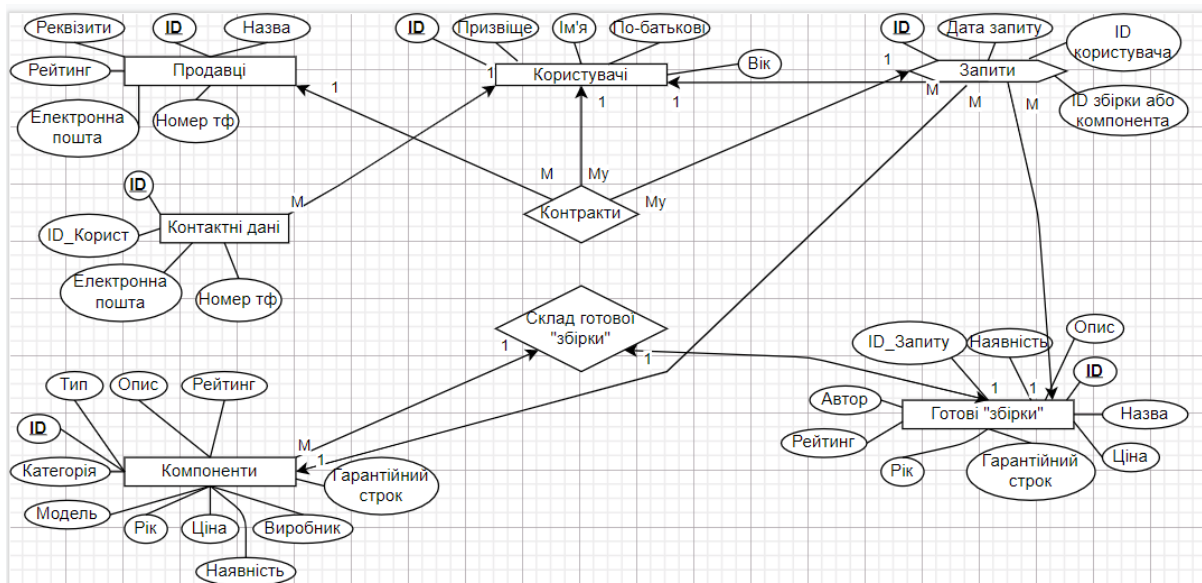


Рис. 1.6 ER-діаграма з атрибутами

## 5 ВИЗНАЧЕННЯ ВИМОГ ДО ОПЕРАТИВНОЇ ОБСТАНОВКИ

Особливих умов для користування бд немає. Наявна операційна система: ОС Windows. Наявний процесор: Intel core i5, 4 ядра. База даних oracle у відкритому стані займає приблизно 300 мб пам'яті + потрібне надлишкове місце, доступно місця: 278 гб. Для більшої швидкодії бажано мати SS-диск, проте в наявності лише HD-диск, хоча швидкодія є в адекватному стані.

Проте при обслуговуванні великої кількості користувачів і за наявності великої кількості інформації, може знадобитись більш потужний сервер, що матиме більше ядер і більш потужний процесор.

Так як сучасні технології досягли меж швидкодії одного ядра, саме вкладання в кількість ядер може стати у нагоді, адже процеси розподілятимуться по декільком ядрам і не так навантажуватимуть систему.

## 6 ВИБІР КОНКРЕТНОЇ СУБД

### 6.1 Перетворення концептуальної моделі в схему БД

Перед тим як перейти до перетворення концептуальної моделі в схему БД, вкажемо обрану СУБД. Для даної роботи будемо використовувати Oracle Database Server 11g XE через його надійність, доступність та вміння працювати з ним. Також будемо застосовувати sql developer.

Так як дана субд не буде займати більше одного ядра процесора та багато пам'яті вона вигідна у плані навантаження системи. Встановлюємо:

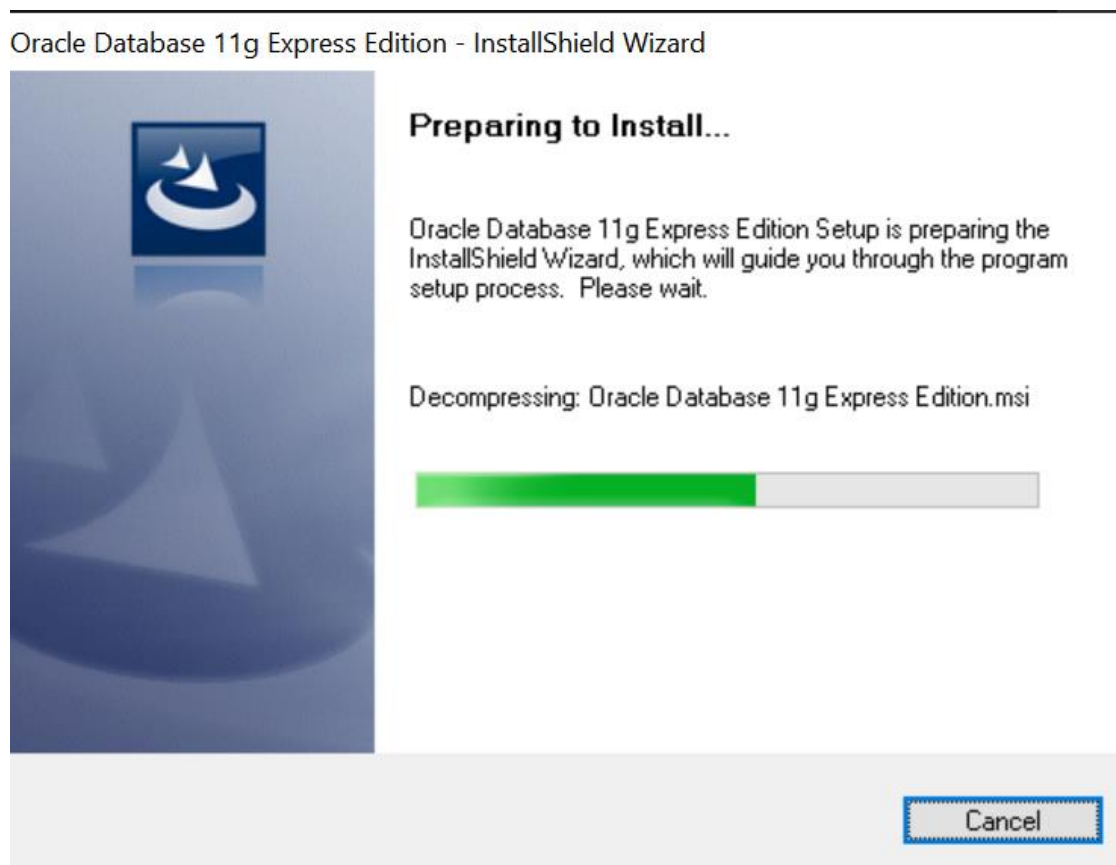


Рисунок 6.1 – Встановлення СУБД

Створимо користувача:

```
SQL> create user user2 identified by 2;  
User created.
```

### Рисунок 6.2 – Створення користувача

Надамо доступ до прав та з'єднання. Приєднаємо користувача:

```
SQL> grant connect to user2;  
Grant succeeded.  
  
SQL> grant resource to user2;  
Grant succeeded.  
  
SQL> connect user2/2;  
Connected.
```

### Рисунок 6.3 – Надання прав та можливості з'єднання

Наведемо приклад створення тимчасового табличного простору:

```
SQL> create tablespace ts  
2 datafile 'ts.dbf'  
3 size 1m;  
Tablespace created.
```

### Рисунок 6.4 – Створення тимчасового табличного простору

### 6.1.1 Формалізація зв'язків

В цьому підрозділі класифікуємо зв'язки. Наведемо існуючі зв'язки та їх формалізацію у таблиці:

Таблиця 6.5 – Формалізація зв'язків

Зв'язок	Опис
Контактні дані-користувачі	1:N, формалізується додатковим атрибутом в сутності «Контактні дані»
Користувачі-контракти	1:N, формалізується додатковим атрибутом в сутності «Контракти»
Користувачі-запити	1:N, формалізується додатковим атрибутом в сутності «Запити»
Продавці-контракти	1:N, формалізується додатковим атрибутом в сутності «Контракти»
Контракти-запити	1:N, формалізується додатковим атрибутом в сутності «Контракти»
Збірки-запити	1:N, формалізується додатковим атрибутом в сутності «Збірки»
Компоненти-запити	1:N, формалізується додатковим атрибутом в сутності «Компоненти»
Склад «збірки»-компоненти	1:N, формалізується додатковим атрибутом в сутності «Склад збірки»
Склад «збірки»-збірки	1:N, формалізується додатковим атрибутом в сутності «Склад збірки»

### 6.1.2 Введення обмежень цілісності та визначення типів даних

Задля зручності, я об'єднав два розділи – «Обмеження цілісності» та «Визначення типів даних» - в один. Опишемо атрибути сутностей, а також вкажемо на їх обмеження цілісності і тип даних:

Таблиця 6.6 – Сутність Компоненти

Компоненти		
PK	ID	Number (15) not null
FK	ID Запитів	Number (15)
	Тип	Varchar2 (30) not null
	Категорія	Varchar2 (8) check
	Модель	Varchar2 (30) not null
	Рік випуску	Number (4) not null
	Ціна	Number (10) not null
	Виробник	Varchar2 (40) not null
	Гарантійний строк	Number (2)
	Характеристичний опис	Varchar2 (255)
	Рейтинг	Number (1)
	Наявність	Char (1) check

Таблиця 6.7 – Сутність Готові «збірки»

Готові «збірки»		
PK	ID	Number (15) not null
FK	ID Запитів	Number (15)
	Назва	Varchar2 (30) not null
	Ціна	Number (10) not null
	Гарантійний строк	Number (2)
	Рік збірки	Number (4) not null
	Характеристичний опис	Varchar2 (255) not null
	Рейтинг	Number (1)

	Автор	Varchar2 (30)
	Наявність	Char (1) check

Таблиця 6.8 – Сутність Користувачі

Користувачі		
PK	ID	Number (15) not null
	Прізвище	Varchar2 (20) not null
	Ім'я	Varchar2 (20) not null
	По-батькові	Varchar2 (20)
	Вік	Varchar2 (3) check

Таблиця 6.9 – Сутність Продавці

Продавці		
PK	ID	Number (15) not null
	Назва	Varchar2 (40) not null
	Реквізити	Varchar2 (100) not null
	Електронна адреса	Varchar2 (30) not null
	Номер телефону	Varchar2 (15) not null
	Рейтинг	Number (1)

Таблиця 6.10 – Сутність Контактні дані

Контактні дані		
PK	ID	Number (15) not null
FK	ID Користувача	Number (15) not null
	Електронна адреса	Varchar2 (30) not null
	Номер телефону	Varchar2 (15) not null

Таблиця 6.11 – Сутність Запити

Запити		
PK	ID	Number (15) not null
FK	ID Користувача	Number (15) not null
	Тип запиту	Varchar (10) check
	Дата запиту	Date

Таблиця 6.12 – Сутність Склад «збірки»

Склад «збірки»		
PK	ID	Number (15) not null
FK	ID «збірки»	Number (15) not null
FK	ID Компонента	Number (15) not null

Таблиця 6.13 – Сутність Контракти

Контракти		
PK	ID	Number (15) not null
FK	ID Користувача	Number (15) not null
FK	ID Продавця	Number (15) not null
FK	ID Запиту	Number (15) not null
	Номер контракту	Number (15) not null



### 6.1.3 Нормалізація бази даних

Так як базу даних попередньо було нормалізовано, наведемо приблизні дані, що були на початку:

Користувачі: атрибут «ПІБ», атрибути «Номер телефону» та «Електронна адреса». Проблема 1: може виникнути плутанина при записі ПІБ, адже користувач може це зробити в будь-якому порядку або не вказати потрібної інформації. Тому тепер це три окремих відповідних атрибути. Проблема 2: користувач може мати більше одного номеру телефону і також більше одної електронної адреси, тому після внесення даних певну інформацію буде складно знайти або відбуватиметься дублювання даних, тому для цих двох атрибутів виділено окрему сутність – «Контактні дані», що мають за один з ключів – id Користувача.

Системні блоки (готові збірки): атрибут id компонента. Виникає бажання скласти список компонентів, з яких складається певний системний блок, проте знову-таки у нас є багато компонентів і тому виникне дублювання даних при внесенні більш ніж одного компонента до сутності «збірок». Задля цього було створено окрему сутність «Склад збірки», що містить id компонента і блоку.

Також усім сутностям було надано атрибут id, якого не було на початку, що призводить їх до 2 НФ, враховуючи, що всі сутності тепер є у 1 НФ, тому що атрибут можуть мати тільки одне значення. Більш того, всі сутності є в 3 НФ, тому що немає транзитивних залежностей.

## 7 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

Отже, в минулому розділі ми описали типи формалізували та нормалізували нашу базу даних. Ми описали типи даних атрибутів даної БД і тепер можливо скласти цілковиту картину. Так виглядає повна модель даних, оформлена через інтернет сервіс [diagrams.net](http://diagrams.net), що дозволяє будувати ер-діаграми та готові даталогічні моделі. Початкову версію було дещо виправлено, тому частково вона виглядає неприродно.

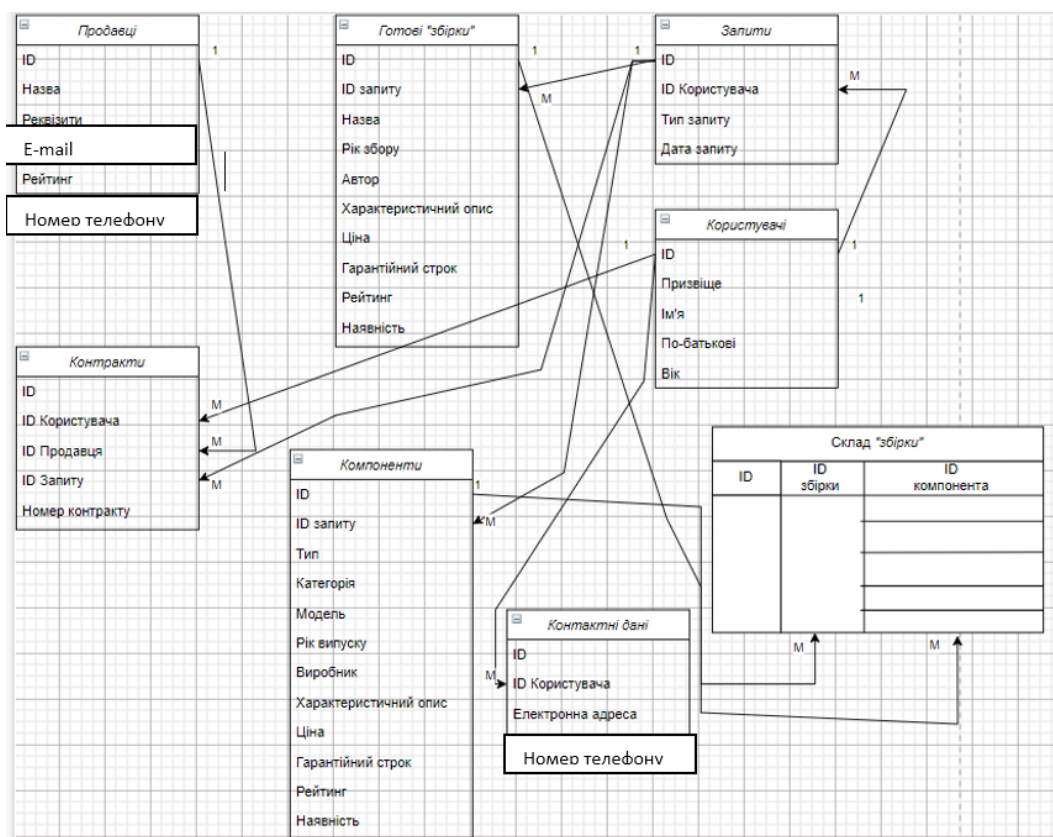


Рисунок 7.1 – Даталогічна модель

## 8 ФІЗИЧНЕ ПРОЕКТУВАННЯ

### 8.1 Розробка скриптів під конкретну СУБД (DDL)

На останньому етапі проектування, попередні моделі ми перетворюємо в DDL-код. Будемо створювати безпосередньо таблиці, а також первинні, зовнішні ключі, тригери та послідовності.

Приклад створення таблиць Components та Composition:

```
CREATE TABLE COMPONENTS
(
  ID NUMBER(15, 0) NOT NULL
, ID_REQ NUMBER(15, 0)
, TYPE VARCHAR2(30 BYTE) NOT NULL
, CATEGORY VARCHAR2(8 BYTE) NOT NULL
, MODEL VARCHAR2(30 BYTE) NOT NULL
, YEAR NUMBER(4, 0) NOT NULL
, PRICE NUMBER(10, 0) NOT NULL
, PRODUCER VARCHAR2(40 BYTE) NOT NULL
, WARRANTY NUMBER(2, 0)
, DESCRIPTION VARCHAR2(255 BYTE)
, RATING NUMBER(1, 0)
, AVAILABLE CHAR(1 BYTE) NOT NULL
, CONSTRAINT COMPONENTS_PK PRIMARY KEY
  (
    ID
  )
)
```

Рисунок 8.1 – Створення таблиці Components

```

alter table components
add constraint ID_Req_Comp
foreign key (ID_Req) references Requests (ID);

create sequence comp_id_seq;

create trigger Comp_BI
before insert on Components
for each row
begin
    select comp_id_seq.nextval
    into :new.ID
    from dual;
End;

```

Рисунок 8.2 – Створення триггеру, послідовності та зовнішніх ключів  
таблиці Компоненти

```

CREATE TABLE COMPOSITION
(
    ID NUMBER(15, 0) NOT NULL
, ID_UNIT NUMBER(15, 0) NOT NULL
, ID_COMP NUMBER(15, 0) NOT NULL
, CONSTRAINT COMPOSITION_PK PRIMARY KEY
(
    ID
)

```

Рисунок 8.3 – Створення таблиці Composition

```

alter table composition
add constraint ID_Unit_Compos
foreign key (ID_Unit) references System_unit (ID);

create sequence compos_id_seq;

create trigger Compos_BI
before insert on Composition
for each row
begin
    select compos_id_seq.nextval
    into :new.ID
    from dual;
End;

alter table composition
add constraint ID_Comp_Compos
foreign key (ID_Comp) references Components (ID);

```

Рисунок 8.4 – Створення триггеру, послідовності та зовнішніх ключів  
таблиці Склад «збірки»

## 8.2 Відпрацювання скриптів (створення фізичної бази даних)

Отже, в минулому підрозділі було наведено приклад SQL-скриптів, в цьому розділі покажемо результат, а саме – створені таблиці:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER(15,0)	No	(null)	1 (null)	
2 ID REQ	NUMBER(15,0)	No	(null)	2 (null)	
3 TYPE	VARCHAR2(30 BYTE)	No	(null)	3 (null)	
4 CATEGORY	VARCHAR2(8 BYTE)	No	(null)	4 (null)	
5 MODEL	VARCHAR2(30 BYTE)	No	(null)	5 (null)	
6 COLUMN2	NUMBER(4,0)	No	(null)	6 (null)	
7 PRICE	NUMBER(10,0)	No	(null)	7 (null)	
8 PRODUCER	VARCHAR2(40 BYTE)	No	(null)	8 (null)	
9 WARRANTY	NUMBER(2,0)	Yes	(null)	9 (null)	
10 DESCRIPTION	VARCHAR2(255 BYTE)	Yes	(null)	10 (null)	
11 RATING	NUMBER(1,0)	Yes	(null)	11 (null)	
12 AVAILABLE	CHAR(1 BYTE)	No	(null)	12 (null)	

Рисунок 8.5 – Готова таблиця Components (Компоненти)

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER(15,0)	No	(null)	1 (null)	
2 ID UNIT	NUMBER(15,0)	No	(null)	2 (null)	
3 ID COMP	NUMBER(15,0)	No	(null)	3 (null)	

Рисунок 8.6 – Готова таблиця Composition (Склад «збірки»)

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER(15,0)	No	(null)	1 (null)	
2 ID USER	NUMBER(15,0)	No	(null)	2 (null)	
3 EMAIL	VARCHAR2(30 BYTE)	No	(null)	3 (null)	
4 PHONE	VARCHAR2(15 BYTE)	Yes	(null)	4 (null)	

Рисунок 8.7 – Готова таблиця Contact\_Details (Контактні дані)

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER(15,0)	No	(null)	1 (null)	
2 ID REQ	NUMBER(15,0)	No	(null)	2 (null)	
3 ID USER	NUMBER(15,0)	No	(null)	3 (null)	
4 ID SELL	NUMBER(15,0)	No	(null)	4 (null)	
5 CONTRACT	NUMBER(15,0)	No	(null)	5 (null)	

Рисунок 8.8 – Готова таблиця Contracts (Контракти)

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ID	NUMBER(15,0)	No	(null)	1	(null)
2	ID USER	NUMBER(15,0)	No	(null)	2	(null)
3	TYPE	VARCHAR2(12 BYTE)	No	(null)	3	(null)
4	DATE REQ	DATE	No	(null)	4	(null)

Рисунок 8.9 – Готова таблиця Requests (Запити)

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ID	NUMBER(15,0)	No	(null)	1	(null)
2	NAME	VARCHAR2(40 BYTE)	No	(null)	2	(null)
3	PROPS	VARCHAR2(100 BYTE)	No	(null)	3	(null)
4	EMAIL	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5	PHONE	VARCHAR2(15 BYTE)	No	(null)	5	(null)
6	RATING	NUMBER(1,0)	Yes	(null)	6	(null)

Рисунок 8.10 – Готова таблиця Sellers (Продавці)

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ID	NUMBER(15,0)	No	(null)	1	(null)
2	ID REQ	NUMBER(15,0)	Yes	(null)	2	(null)
3	NAME	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4	PRICE	NUMBER(10,0)	No	(null)	4	(null)
5	WARRANTY	NUMBER(2,0)	Yes	(null)	5	(null)
6	YEAR	NUMBER(4,0)	No	(null)	6	(null)
7	DESCRIPTION	VARCHAR2(255 BYTE)	No	(null)	7	(null)
8	RATING	NUMBER(1,0)	Yes	(null)	8	(null)
9	AUTHOR	VARCHAR2(30 BYTE)	Yes	(null)	9	(null)
10	AVAILABLE	CHAR(1 BYTE)	No	(null)	10	(null)

Рисунок 8.11 – Готова таблиця System\_Units (Готові «збірки»)

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ID	NUMBER(15,0)	No	(null)	1	(null)
2	SNAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	NAME	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4	PATRONOMIC	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	AGE	NUMBER(3,0)	No	(null)	5	(null)

Рисунок 8.12 – Готова таблиця Users (Користувачі)

### 8.3 Фізична модель даних

Отже, маємо інфологічну, даталогічну моделі, маємо приклади SQL-скриптів і готові таблиці. Все, що нам залишилось, це поєднати все це в одну фізичну модель. На щастя, sql developer надає можливість створення таких схем відношень, тому модель даної БД виглядає так:

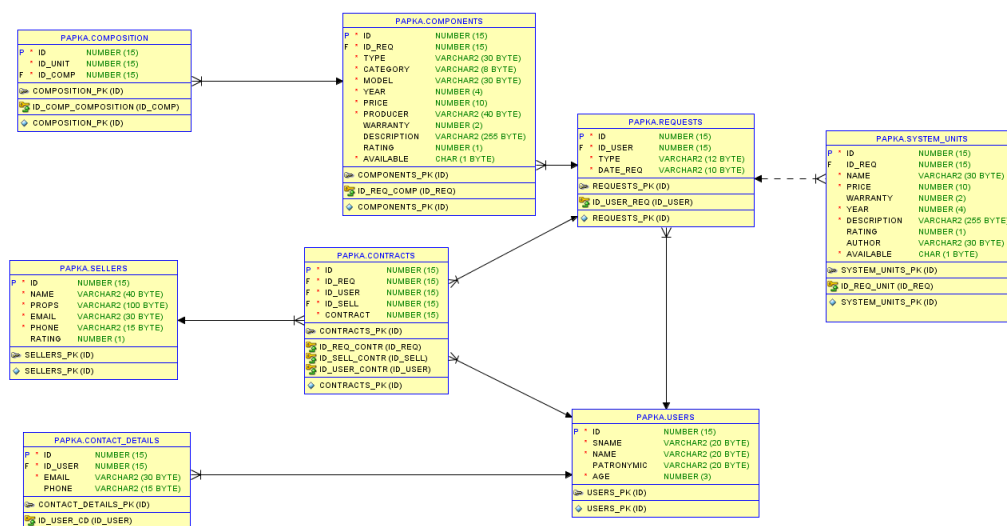


Рисунок 8.13 – Фізична модель даних (Відсутній зв'язок Склад збірки-готові збірки через технічну помилку)

## 8.4 Тестові дані (DML)

Кінець проектування – це реалізація базою даних деяких функцій типу вставлення, видалення, оновлення інформації, а також вибірка.

Перелік функцій та запитів по функціям:

### 1. Вставити дані в таблицю Users (Рядок 10)

*insert into users (id, sname, name, patronymic, age) values (null, 'Blaskovitz', 'William', 'Jacob', 25);*

ID	SNAME	NAME	PATRONYMIC	AGE
1	1 sname	name	patro	20
2	2 sname	name	patr	19
3	3 sname	name	pat	18
4	4 sname	name	pa	17
5	5 sname	name	patro	18
6	6 sname	name	pat	19
7	7 sname	name	patron	20
8	8 sname	name	patrony	21
9	9 sname	name	patronym	22
10	10 Blaskovitz	William Jacob		25

Рисунок 8.14 – Вставлення в таблицю Users

### 2. Вставити дані в таблицю Requests (Рядок 11)

*insert into requests (id, id\_user, type, date\_req) values (null, 10, 'Components', '22-12-2021');*

ID	ID_USER	TYPE	DATE_REQ
1	1	1 type	22-12-2021
2	2	1 type	21-12-2021
3	3	2 typ	21-12-2021
4	4	3 ty	21-12-2021
5	5	4 t	21-12-2021
6	6	5 typet	21-12-2021
7	7	6 typety	21-12-2021
8	8	7 typety	22-12-2021
9	9	8 typetype	22-12-2021
10	10	9 typee	22-12-2021
11	11	10 Components	22-12-2021

Рисунок 8.15 – Вставлення в таблицю Requests



### 3. Вставити дані в таблицю Components (Рядок 9)

Перед тим, як вставляти дані, додамо перевірку на наявність компоненту:

```
alter table components
```

```
add constraint ava_check check (available in('y', 'n'));
```

```
insert into components (id, ID_req, type, category, model, year, price, producer,  
warranty, description, rating, available)
```

```
values (null, 10, 'type', 'category', 'model', 21, 10, 'producer', 24, 'description', 5,  
'y');
```

	ID	ID_REQ	TYPE	CATEGORY	MODEL	YEAR	PRICE	PRODUCER	WARRANTY	DESCRIPTION	RATING	AVAILABLE
1	1	1	type	category	model	3	1	producer	0	description	5	y
2	2	2	type	categor	model	5	2	produce	0	descriptio	5	n
3	3	3	type	catego	model	6	3	produc	0	descripti	3	n
4	4	4	type	cateq	model	9	4	produ	0	descr	2	n
5	5	5	type	cate	mode	11	5	prod	0	desc	4	n
6	6	6	typ	cat	mod	13	6	pro	0	des	4	y
7	7	7	ty	ca	mo	16	7	pr	3	de	5	n
8	8	9	type	category	model	20	9	producer	12	description	5	y
9	9	10	type	category	model	21	10	producer	24	description	5	y

Рисунок 8.16 – Вставлення в таблицю Components

### 4. Вставити дані в таблицю Sellers (Рядок 10)

```
insert into sellers (id, name, props, email, phone, rating)
```

```
values (null, 'Rozetka', '5168244432358012', 'rozetkahelp@gmail.com',  
'+380938881212', 5);
```

ID	NAME	PROPS	EMAIL	PHONE	RATING
1	1 company	some props	some email	some phone	5
2	2 compan	some prop	some emai	some phon	4
3	3 compa	some pro	some ema	some pho	3
4	4 comp	some pr	some em	some ph	2
5	5 com	some p	some e	some p	1
6	6 co	some	some	some	2
7	7 com	someops	somail	someone	3
8	8 company	props	email	phone	4
9	9 name	ops	mail	number	5
10	10 Rozetka	5168244432358012	rozetkahelp@gmail.com	+380938881212	5

Рисунок 8.17 – Вставлення в таблицю Sellers

5. Видалити останній рядок в таблиці contact\_details (Рядок з id 11 попередньо було видалено)

*delete from contact\_details where id = 12;*

ID	ID_USER	EMAIL	PHONE
1	1	1 email	phone
2	2	2 emai	phon
3	3	3 emal	phoe
4	4	4 emil	phne
5	5	5 eail	pone
6	6	6 mail	hone
7	7	7 emai	phon
8	8	8 ema	pho
9	9	9 em	ph
10	10	10 user1@gmail.com	+380731223067

Рисунок 8.18 – Видалення з таблиці Contact\_Details

6. Видалити рядок в таблиці contracts з id=5 (Рядок 5 з id = 6)

*delete from contracts where id = 5;*

ID	ID_REQ	ID_USER	ID_SELL	CONTRACT
1	1	1	1	1 2312432
2	2	2	2	2 12312433
3	3	3	3	3 12312434
4	4	4	4	4 12312435
5	6	6	6	6 12312437
6	7	7	7	7 12312438
7	8	8	8	8 12312439
8	9	9	9	9 12312440
9	10	10	10	10 12312441
10	11	9	9	9 12312442

Рисунок 8.19 – Видалення з таблиці Contracts

7. Видалити рядки в таблиці composition з id\_comp = 6 (Зникли рядки з id = 5, 12)

*delete from contracts where id\_comp = 6;*

	ID	ID_UNIT	ID_COMP
1	1	1	1
2	2	1	2
3	3	1	3
4	4	1	4
5	5	1	5
6	7	2	1
7	8	2	2
8	9	2	3
9	10	2	4
10	11	2	5

Рисунок 8.20 – Видалення з таблиці Composition

8. Видалити рядок в таблиці system\_units з warranty = 12 (Зникли рядки з id = 8,9)

*delete from system\_units where warranty = 12;*

	ID	ID_REQ	NAME	PRICE	WARRANTY	YEAR	DESCRIPTION	RATING	AUTHOR	AVAILABLE
1	1	1	name	6	3	12	description	3	abscent	y
2	2	2	name	5	3	13	description	4	abscent	y
3	3	3	name	7	3	14	description	5	abscent	y
4	4	4	name	9	6	15	description	4	abscent	y
5	5	5	name	8	6	16	description	3	abscent	y
6	6	6	name	8	4	17	description	5	abscent	y
7	7	7	name	7	9	18	description	4	abscent	y
8	10	10	name	10	24	21	description	5	abscent	y

Рисунок 8.21 – Видалення з таблиці System\_Units

9. Змінити id\_unit з певним id в таблиці composition

*update composition*

*set id\_unit = 3 where id = 1*

ID	ID_UNIT	ID_COMP
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	7	2
7	8	2
8	9	2
9	10	2
10	11	2

ID	ID_UNIT	ID_COMP
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	7	2
7	8	2
8	9	2
9	10	2
10	11	2

Рисунок 8.22 – Оновлення даних таблиці Composition

10. Змінити ім'я, прізвище та по-батькові в таблиці users де age = 17

update users

set sname = 'Pjantkovski', name = 'Oleg', patronymic = 'Grygorovych' where age = 17;

ID	SNAME	NAME	PATRONYMIC	AGE
1	1 sname	name	patro	20
2	2 snam	name	patr	19
3	3 sna	nam	pat	18
4	4 Pjantkovski	Oleg	Grygorovych	17
5	5 sname	nam	patro	18
6	6 s	name	pat	19
7	7 snames	namen	patron	20
8	8 snamesn	namenam	patrony	21
9	9 snamesna	namenam	patronym	22
10	10 Blaskovitz	William	Jacob	25

Рисунок 8.23 – Оновлення даних таблиці Users

11. Змінити назву компанії, реквізити, електронну пошту, рейтинг та контактний телефонний номер в таблиці sellers, де Rating < 2

*update sellers*

*set name = 'Eldorado', props = 5375123400003232, email = 'eldoradotech@gmail.com', phone = '+380441226458', rating = 5 where rating < 2;*

ID	NAME	PROPS	EMAIL	PHONE	RATING
1	1 company	some props	some email	some phone	5
2	2 compan	some prop	some emai	some phon	4
3	3 compa	some pro	some ema	some pho	3
4	4 comp	some pr	some em	some ph	2
5	5 Eldorado	5375123400003232	eldoradotech@gmail.com	+380441226458	5
6	6 co	some	some	some	2
7	7 com	someops	somail	someone	3
8	8 company	props	email	phone	4
9	9 name	ops	mail	number	5
10	10 Rozetka	5168244432358012	rozetkahelp@gmail.com	+380938881212	5

Рисунок 8.24 – Оновлення даних таблиці Sellers

12. Змінити електронну пошту в таблиці contact\_details з email = 'user1@gmail.com'

*update contact\_details*

*set email = 'user2@gmail.com' where email = 'user1@gmail.com';*

ID	ID_USER	EMAIL	PHONE
1	1	1 email	phone
2	2	2 emai	phon
3	3	3 emal	phoe
4	4	4 Emil	phne
5	5	5 eail	pone
6	6	6 mail	hone
7	7	7 emai	phon
8	8	8 ema	pho
9	9	9 em	ph
10	10	10 user2@gmail.com	+380731223067

Рисунок 8.25 – Оновлення даних таблиці Contact\_Details

13. Вибірка з таблиць contracts, sellers, що покаже id контракту, назву компанії та номер контракту

```
select c.id_sell, c.contract, s.name
```

```
from contracts c
```

```
left join sellers s on c.id_sell = s.id
```

	ID_SELL	CONTRACT	NAME
1	1	12312432	company
2	2	12312433	compan
3	3	12312434	compa
4	4	12312435	comp
5	6	12312437	co
6	7	12312438	com
7	8	12312439	company
8	9	12312440	name
9	10	12312441	Rozetka
10	9	12312442	name

Рисунок 8.26 – Вибірка з таблиць contracts, sellers

14. Вибірка з таблиць contact\_details, users, що покаже id користувача, телефон користувача, id контактних даних та фамілію користувача

```
select cd.id_user, cd.phone, u.id, u.sname
```

```
from contact_details cd
```

```
left join users u on cd.id_user = u.id
```

	ID_USER	PHONE	ID	SNAME
1		1 phone	1	sname
2		2 phon	2	sname
3		3 phoe	3	sname
4		4 phne	4	Pjantkovski
5		5 pone	5	sname
6		6 hone	6	s
7		7 phon	7	sname
8		8 pho	8	sname
9		9 ph	9	sname
10		10 +380731223067	10	Blaskovitz

Рисунок 8.27 – Вибірка з таблиць Contact\_Details, Users

15. Вибірка з таблиць composition, components, system\_units, що покаже id компонента, id збірки, назву компонента та назву збірки

```
select c0.id_comp, c0.id_unit, c.model, su.name
```

```
from composition c0
```

```
left join components c on c0.id_comp = c.id
```

```
left join system_units su on c0.id_unit = su.id
```

	ID_COMP	ID_UNIT	MODEL	NAME
1	1	3	model	name
2	2	1	model	name
3	3	1	model	name
4	4	1	model	name
5	5	1	mode	name
6	1	2	model	name
7	2	2	model	name
8	3	2	model	name
9	4	2	model	name
10	5	2	mode	name

Рисунок 8.28 - Вибірка з таблиць Composition, Components, System\_Units

## ВИСНОВКИ

Отже, ця робота містить повну інформацію про проектування даних. При проектуванні все, спершу, поділяється на етапи, а потім реалізовується.

Спочатку обирається та аналізується предметна область. Створюються теорії та плани щодо виконуваних базою даних в майбутньому функцій. Осмислюється сфера обслуговування бд.

Наступним етапом є створення сутностей, їх атрибутів та зв'язків. Після опису вони поєднуються в одній діаграмі відношень із зазначенням нотації та типу зв'язків.

Попередній етап називався інфологічним проектуванням, за ним слідує даталогічне проектування. На цьому етапі відбувається нормалізація та представлення даних не на рівні абстракції, але як цілісний зв'язок таблиць з первинними та зовнішніми ключами.

За даталогічним проектуванням слідує обрання конкретної СУБД для подальшого фізичного проектування. В цій роботі було використано СУБД від компанії Oracle.

Фізичне проектування включає в себе втілення початкового задуму і перенесення таблиць і схем в код, створення самої бази даних і внесення тестових даних.

Не зважаючи на складність і довготривалість процесу проектування, він є найважливішим, адже від того як буде продумано БД буде залежати продуктивність і зручність її використання. Тому в цій роботі було детально продемонстровано всі вищезазначені етапи.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сайт для створення схем та діаграм –

<https://www.diagrams.net/>

2. Сайти (приклад), що містять інформацію про техніку –

Rozetka - <https://rozetka.com.ua/>

Comfy - <https://comfy.ua/>

3. Oracle СУБД –

<https://www.oracle.com/ru/database/technologies/xe-downloads.html>

4. Типи даних –

[https://docs.oracle.com/cd/B28359\\_01/server.111/b28286/sql\\_elements001.htm](https://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements001.htm)

5. Створення таблиць –

[https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_7002.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7002.htm)

6. Select –

[https://docs.oracle.com/cd/B28359\\_01/server.111/b28286/statements\\_10002.htm](https://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_10002.htm)

7. Insert into –

[https://docs.oracle.com/cd/B28359\\_01/server.111/b28286/statements\\_9014.htm](https://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9014.htm)

8. Update –

[https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_10007.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_10007.htm)

9. Delete –

[https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_8005.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8005.htm)

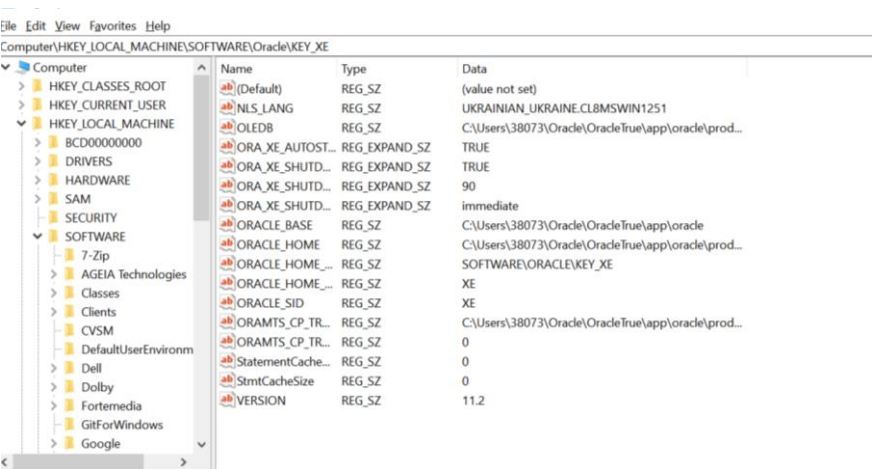
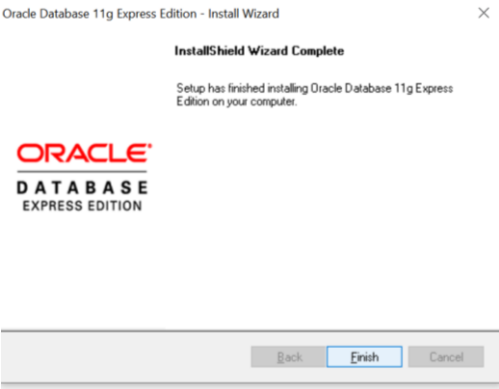
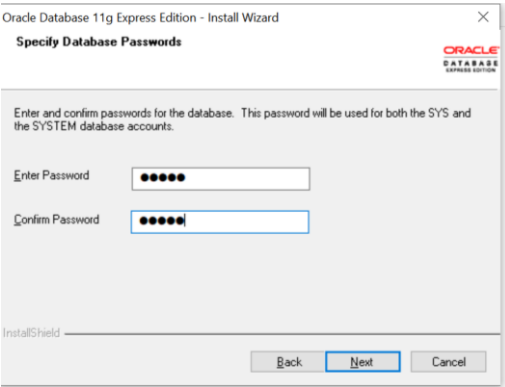
10. Join –

<https://stackoverflow.com/questions/17238038/joining-tables-in-oracle-sql-developer>

ДОДАТКИ

Додаток А

Встановлення СУБД Oracle



Файл Параметри Перегляд

Процеси Продуктивність Банк програм Автозавантаження Користувачі Докладно Служби

Ім'я	Іденти...	Стан	Ім'я корист...	ЦП	Пам'ять (ак...	Віртуалізація U...
MsMpEng.exe	3964	Виконується	SYSTEM	00	262 180 K	Не дозволено
NisSrv.exe	12772	Виконується	LOCAL SER...	00	2 112 K	Не дозволено
nvcontainer.exe	3448	Виконується	SYSTEM	00	4 856 K	Не дозволено
nvcontainer.exe	13296	Виконується	38073	00	14 980 K	Вимкнено
NVDisplay.Container...	1976	Виконується	SYSTEM	00	2 664 K	Не дозволено
NVDisplay.Container...	17324	Виконується	SYSTEM	00	7 892 K	Не дозволено
NVIDIA Web Helper.e...	11836	Виконується	38073	00	2 500 K	Вимкнено
OneDrive.exe	10768	Виконується	38073	00	9 748 K	Вимкнено
oracle.exe	16504	Виконується	SYSTEM	00	215 764 K	Не дозволено
PresentationFontCac...	4720	Виконується	LOCAL SER...	00	108 K	Не дозволено
regedit.exe	17400	Виконується	38073	00	1 908 K	Не дозволено
Registry	124	Виконується	SYSTEM	00	8 144 K	Не дозволено
RtkAudUService64.exe	3324	Виконується	SYSTEM	00	772 K	Не дозволено
RtkAudUService64.exe	6476	Виконується	38073	00	704 K	Вимкнено
RuntimeBroker.exe	9780	Виконується	38073	00	2 112 K	Вимкнено
RuntimeBroker.exe	16380	Виконується	38073	00	8 720 K	Вимкнено
RuntimeBroker.exe	15736	Виконується	38073	00	1 380 K	Вимкнено
RuntimeBroker.exe	2068	Виконується	38073	00	1 108 K	Вимкнено
RuntimeBroker.exe	13116	Виконується	38073	00	1 136 K	Вимкнено
RuntimeBroker.exe	11744	Виконується	38073	00	2 440 K	Вимкнено
SearchApp.exe	6548	Призупинено	38073	00	0 K	Вимкнено
SearchIndexer.exe	5464	Виконується	SYSTEM	00	31 912 K	Не дозволено

Список Завершити завдання

Диспетчер завдань

Файл Параметри Перегляд

Процеси Продуктивність Банк програм Автозавантаження Користувачі Докладно Служби

Ім'я	Іденти...	Опис	Стан	Група
NcbService	1540	Network Connection Broker	Виконується	LocalSystemNe...
NcdAutoSetup		Network Connected Devices Auto-Set...	Зупинено	LocalServiceNo...
Netlogon		Netlogon	Зупинено	
Netman		Мережні підключення	Зупинено	LocalSystemNe...
netprofm	2112	Служба списку мереж	Виконується	LocalService
NetSetupSvc		Служба налаштування мережі	Зупинено	netsvcs
NetTcpPortSharing		Net.Tcp Port Sharing Service	Зупинено	
NgcCntrSvc	4788	Контейнер Microsoft Passport	Виконується	LocalServiceNet...
NgcSvc	14348	Microsoft Passport	Виконується	LocalSystemNe...
NlaSvc	1696	Network Location Awareness	Виконується	NetworkService
nsi	2004	Network Store Interface Service	Виконується	LocalService
NvContainer.LocalSystem	3448	NVIDIA LocalSystem Container	Виконується	
NVDisplay.ContainerLocalSy...	1976	NVIDIA Display Container LS	Виконується	
OneSyncSvc		Sync Host	Зупинено	UnistackSvcGro...
OneSyncSvc_3390233	2932	Sync Host_3390233	Виконується	UnistackSvcGro...
OracleJobSchedulerXE		OracleJobSchedulerXE	Зупинено	
OracleMTSRecoveryService		OracleMTSRecoveryService	Зупинено	
OracleServiceXE	16504	OracleServiceXE	Виконується	
OracleXEClrAgent		OracleXEClrAgent	Зупинено	
OracleXETNSListener	9744	OracleXETNSListener	Виконується	
ose		Office Source Engine	Зупинено	
p2pimsvc		Peer Networking Identity Manager	Зупинено	LocalServicePee...

admin	12.11.2021 19:00	Папка файлів
diag	12.11.2021 19:00	Папка файлів
fast_recovery_area	12.11.2021 19:01	Папка файлів
oradata	12.11.2021 19:00	Папка файлів
product	12.11.2021 18:58	Папка файлів

## Додаток Б

### SQL-скрипти по створенню таблиць, послідовностей, триггерів, ключів

#### Components

```
CREATE TABLE COMPONENTS
(
  ID NUMBER(15, 0) NOT NULL
, ID_REQ NUMBER(15, 0)
, TYPE VARCHAR2(30 BYTE) NOT NULL
, CATEGORY VARCHAR2(8 BYTE) NOT NULL
, MODEL VARCHAR2(30 BYTE) NOT NULL
, YEAR NUMBER(4, 0) NOT NULL
, PRICE NUMBER(10, 0) NOT NULL
, PRODUCER VARCHAR2(40 BYTE) NOT NULL
, WARRANTY NUMBER(2, 0)
, DESCRIPTION VARCHAR2(255 BYTE)
, RATING NUMBER(1, 0)
, AVAILABLE CHAR(1 BYTE) NOT NULL
, CONSTRAINT COMPONENTS_PK PRIMARY KEY
  (
    ID
  )
)

alter table components
add constraint ID_Req_Comp
foreign key (ID_Req) references Requests (ID);

create sequence comp_id_seq;

create trigger Comp_BI
before insert on Components
for each row
begin
  select comp_id_seq.nextval
  into :new.ID
  from dual;
End;
```

## Composition (Склад «збірки»):

```
CREATE TABLE COMPOSITION
(
    ID NUMBER(15, 0) NOT NULL
, ID_UNIT NUMBER(15, 0) NOT NULL
, ID_COMP NUMBER(15, 0) NOT NULL
, CONSTRAINT COMPOSITION_PK PRIMARY KEY
(
    ID
)
)
```

```
alter table composition
add constraint ID_Unit_Compos
foreign key (ID_Unit) references System_unit (ID);
```

```
create sequence compos_id_seq;
```

```
create trigger Compos_BI
before insert on Composition
for each row
begin
    select compos_id_seq.nextval
    into :new.ID
    from dual;
End;
```

```
alter table composition
add constraint ID_Comp_Compos
foreign key (ID_Comp) references Components (ID);
```

## Contact\_Details:

```
CREATE TABLE CONTACT_DETAILS
(
    ID NUMBER(15, 0) NOT NULL
, ID_USER NUMBER(15, 0) NOT NULL
, EMAIL VARCHAR2(30 BYTE) NOT NULL
, PHONE VARCHAR2(15 BYTE)
, CONSTRAINT CONTACT_DETAILS_PK PRIMARY KEY
(
    ID
)
)
```

```
alter table contact_details
add constraint ID_User_Cont
foreign key (ID_User) references Users (ID);
```

```
create sequence cont_id_seq;
```

```
create trigger Cont_BI
before insert on Contact_details
for each row
begin
    select cont_id_seq.nextval
    into :new.ID
    from dual;
End;
```

## Contracts:

```
CREATE TABLE CONTRACTS
(
    ID NUMBER(15, 0) NOT NULL
, ID_REQ NUMBER(15, 0) NOT NULL
, ID_USER NUMBER(15, 0) NOT NULL
, ID_SELL NUMBER(15, 0) NOT NULL
, CONTRACT NUMBER(15, 0) NOT NULL
, CONSTRAINT CONTRACTS_PK PRIMARY KEY
(
    ID
)
)

alter table contracts
add constraint ID_Req_Contr
foreign key (ID_Req) references Requests (ID)
add constraint ID_User_Contr
foreign key (ID_User) references Users (ID)
add constraint ID_Sell_Contr
foreign key (ID_Sell) references Sellers (ID);

create sequence contr_id_seq;

create trigger Contr_BI
before insert on Contracts
for each row
begin
    select contr_id_seq.nextval
    into :new.ID
    from dual;
End;
```

## Requests:

```
CREATE TABLE REQUESTS
(
    ID NUMBER(15, 0) NOT NULL
, ID_USER NUMBER(15, 0) NOT NULL
, TYPE VARCHAR2(12 BYTE) NOT NULL
, DATE_REQ DATE NOT NULL
, CONSTRAINT REQUESTS_PK PRIMARY KEY
(
    ID
)
)

alter table requests
add constraint ID_User_Req
foreign key (ID_User) references Users (ID);

create sequence req_id_seq;

create trigger req_BI
before insert on requests
for each row
begin
    select req_id_seq.nextval
    into :new.ID
    from dual;
End;
```

Sellers:

```
CREATE TABLE SELLERS
(
  ID NUMBER(15, 0) NOT NULL
, NAME VARCHAR2(40 BYTE) NOT NULL
, PROPS VARCHAR2(100 BYTE) NOT NULL
, EMAIL VARCHAR2(30 BYTE) NOT NULL
, PHONE VARCHAR2(15 BYTE) NOT NULL
, RATING NUMBER(1, 0)
, CONSTRAINT SELLERS_PK PRIMARY KEY
(
  ID
)
)
```

```
create sequence sell_id_seq;

create trigger sell_BI
before insert on sellers
for each row
begin
  select sell_id_seq.nextval
  into :new.ID
  from dual;
End;
```

System\_Units (Готові «збірки»):

```
CREATE TABLE SYSTEM_UNITS
(
  ID NUMBER(15, 0) NOT NULL
, ID_REQ NUMBER(15, 0)
, NAME VARCHAR2(30 BYTE) NOT NULL
, PRICE NUMBER(10, 0) NOT NULL
, WARRANTY NUMBER(2, 0)
, YEAR NUMBER(4, 0) NOT NULL
, DESCRIPTION VARCHAR2(255 BYTE) NOT NULL
, RATING NUMBER(1, 0)
, AUTHOR VARCHAR2(30 BYTE)
, AVAILABLE CHAR(1 BYTE) NOT NULL
, CONSTRAINT SYSTEM_UNITS_PK PRIMARY KEY
(
  ID
)
)
```

```
alter table system_units
add constraint ID_Req_Sys
foreign key (ID_Req) references system_units (ID);

create sequence sys_id_seq;

create trigger sys_BI
before insert on system_units
for each row
begin
  select sys_id_seq.nextval
  into :new.ID
  from dual;
End;
```



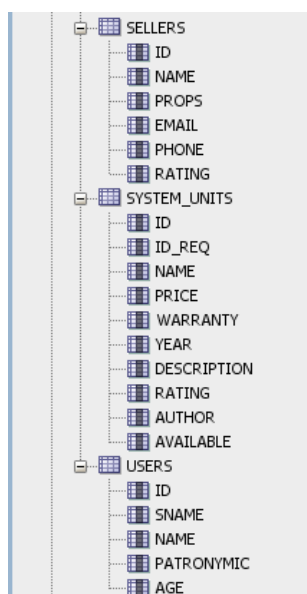
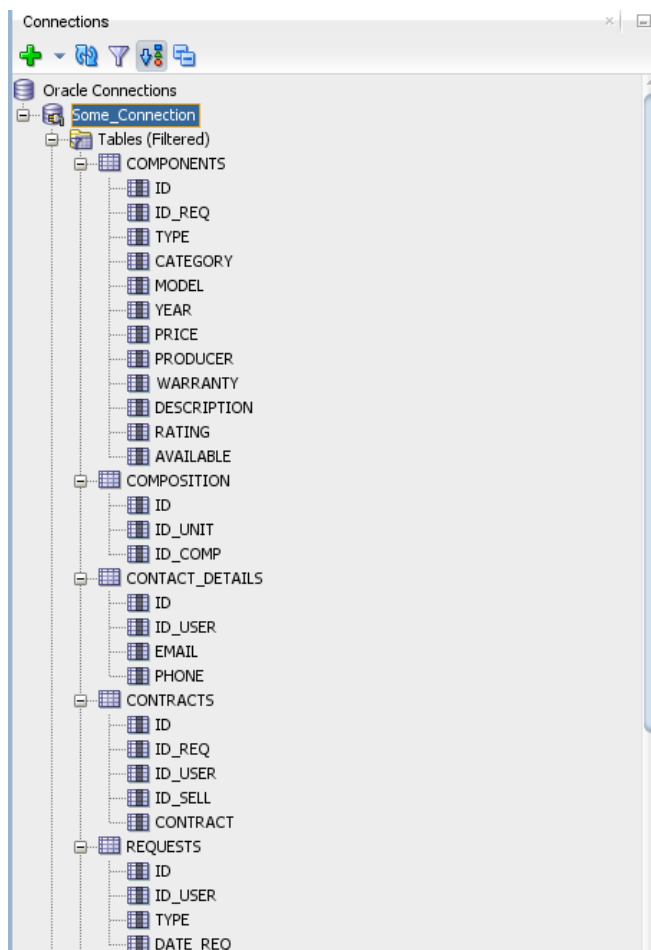
## Users:

```
CREATE TABLE USERS
(
  ID NUMBER(15, 0) NOT NULL
, SNAME VARCHAR2(20 BYTE) NOT NULL
, NAME VARCHAR2(20 BYTE) NOT NULL
, PATRONYMIC VARCHAR2(20 BYTE)
, AGE NUMBER(3, 0) NOT NULL
, CONSTRAINT USERS_PK PRIMARY KEY
  (
    ID
  )
)
```

```
create sequence users_id_seq;

create trigger users_BI
before insert on users
for each row
begin
  select users_id_seq.nextval
  into :new.ID
  from dual;
End;
```

## Відпрацювання скриптів у СУБД



## Додаток Г

Скрипти для заповнення бази даних

```
insert into users (id, sname, name, patronymic, age) values (null, 'sname',  
'name', 'patro', 20);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 'snam', 'name',  
'patr', 19);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 'sna', 'nam',  
'pat', 18);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 'sn', 'na', 'pa',  
17);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 'sname', 'nam',  
'patro', 18);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 's', 'name',  
'pat', 19);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 'snames',  
'namen', 'patron', 20);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 'snamesn',  
'namena', 'patrony', 21);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 'snamesna',  
'namenam', 'patronym', 22);
```

```
insert into users (id, sname, name, patronymic, age) values (null, 'Blaskovitz',  
'William', 'Jacob', 25);
```

*insert into requests (id, id\_user, type, req\_dat) values (null, 1, 'type', '21-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 2, 'typ', '21-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 3, 'ty', '21-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 4, 't', '21-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 5, 'typet', '21-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 6, 'typety', '21-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 7, 'typetyp', '22-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 8, 'typetype', '22-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 9, 'typeepyt', '22-12-2021');*

*insert into requests (id, id\_user, type, req\_dat) values (null, 10, 'Components', '22-12-2021');*

*alter table components*

*add constraint ava\_check check (available in('y', 'n'));*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 1, 'type', 'category', 'model', 03, 1, 'producer', 0, 'description', 5, 'y');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 2, 'type', 'categor', 'model', 05, 2, 'produce', 0, 'descriptio', 5, 'n');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 3, 'type', 'catego', 'model', 06, 3, 'produc', 0, 'descripti', 3, 'n');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 4, 'type', 'categ', 'model', 09, 4, 'produ', 0, 'descr', 2, 'n');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 5, 'type', 'cate', 'mode', 11, 5, 'prod', 0, 'desc', 4, 'n');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 6, 'typ', 'cat', 'mod', 13, 6, 'pro', 0, 'des', 4, 'y');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 7, 'ty', 'ca', 'mo', 16, 7, 'pr', 3, 'de', 5, 'n');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 8, 'typ', 'catory', 'mel', 19, 8, 'procer', 6, 4, 'rating', 'y');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 9, 'type', 'category', 'model', 20, 9, 'producer', 12, 'description', 5, 'y');*

*insert into components (id, ID\_req, type, category, model, year, price, producer, warranty, description, rating, available)*

*values (null, 10, 'type', 'category', 'model', 21, 10, 'producer', 24, 'description', 5, 'y');*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'company', 'some props', 'some email', 'some phone', 5);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'compan', 'some prop', 'some emai', 'some phon', 4);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'compa', 'some pro', 'some ema', 'some pho', 3);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'comp', 'some pr', 'some em', 'some ph', 2);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'com', 'some p', 'some e', 'some p', 1);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'co', 'some', 'some', 'some', 2);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'com', 'someops', 'somail', 'someone', 3);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'company', 'props', 'email', 'phone', 4);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'name', 'ops', 'mail', 'number', 5);*

*insert into sellers (id, name, props, email, phone, rating)*

*values (null, 'Rozetka', '5168244432358012', 'rozetkahelp@gmail.com', '+380938881212', 5);*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 1, 'name', 6, 3, 12, 'description', 3, 'abscent', 'y');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 2, 'name', 5, 3, 13, 'description', 4, 'abscent', 'y');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 3, 'name', 7, 3, 14, 'description', 5, 'abscent', 'y');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 4, 'name', 9, 6, 15, 'description', 4, 'abscent', 'y');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 5, 'name', 8, 6, 16, 'description', 3, 'abscent', 'y');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 6, 'name', 8, 4, 17, 'description', 5, 'abscent', 'y');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 7, 'name', 7, 9, 18, 'description', 4, 'abscent', 'y');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 8, 'name', 6, 12, 19, 'description', 5, 'abscent', 'n');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 9, 'name', 9, 12, 20, 'description', 5, 'abscent', 'n');*

*insert into system\_units (id, id\_req, name, price, warranty, year, description, rating, author, available)*

*values (null, 10, 'name', 10, 24, 21, 'description', 5, 'abscent', 'y');*

*insert into composition (id, id\_unit, id\_comp) values (1, 1, 1);*

*insert into composition (id, id\_unit, id\_comp) values (1, 1, 2);*

*insert into composition (id, id\_unit, id\_comp) values (1, 1, 3);*

*insert into composition (id, id\_unit, id\_comp) values (1, 1, 4);*

*insert into composition (id, id\_unit, id\_comp) values (1, 1, 5);*

*insert into composition (id, id\_unit, id\_comp) values (1, 1, 6);*

*insert into composition (id, id\_unit, id\_comp) values (2, 2, 1);*

*insert into composition (id, id\_unit, id\_comp) values (2, 2, 2);*

*insert into composition (id, id\_unit, id\_comp) values (2, 2, 3);*

*insert into composition (id, id\_unit, id\_comp) values (2, 2, 4);*

*insert into composition (id, id\_unit, id\_comp) values (2, 2, 5);*



*insert into composition (id, id\_unit, id\_comp) values (2, 2, 6);*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 1, 'email', 'phone');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 2, 'emai', 'phon');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 3, 'emal', 'phoe');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 4, 'emil', 'phne');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 5, 'eail', 'pone');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 6, 'mail', 'hone');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 7, 'emai', 'phon');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 8, 'ema', 'pho');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 9, 'em', 'ph');*

*insert into contact\_Details (id, id\_user, email, phone)*  
*values (null, 10, 'user1@gmail.com', '+380731223067');*

*insert into contact\_Details (id, id\_user, email, phone)*

*values (null, 9, 'email', 'phone');*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 1, 1, 1, 12312432);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 2, 2, 2, 12312433);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 3, 3, 3, 12312434);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 4, 4, 4, 12312435);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 5, 5, 5, 12312436);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 6, 6, 6, 12312437);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 7, 7, 7, 12312438);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 8, 8, 8, 12312439);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 9, 9, 9, 12312440);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 10, 10, 10, 12312441);*

*insert into contracts (id, id\_req, id\_user, id\_sell, contract)*

*values (null, 9, 9, 9, 12312442);*

*update composition*

*set id\_unit = 3 where id = 1;*

*update users*

*set sname = 'Pjantkovski', name = 'Oleg', patronymic = 'Grygorovych' where  
age = 17;*

*update sellers*

*set name = 'Eldorado', props = 5375123400003232, email =  
'eldoradotech@gmail.com', rating = 5 where rating < 2;*

*update sellers*

*set phone = '+380441226458' where name = 'Eldorado';*

*update contact\_details*

*set email = 'user2@gmail.com' where email = 'user1@gmail.com';*