

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт до комп'ютерного практикума №1 з дисципліни

«Бази даних»

Прийняла:
Викладач кафедри ІІІ
Марченко О. І.
29 вересня 2021 року

Виконав:
Студент групи ІТ-01
Бардін В. Д.

Київ – 2021

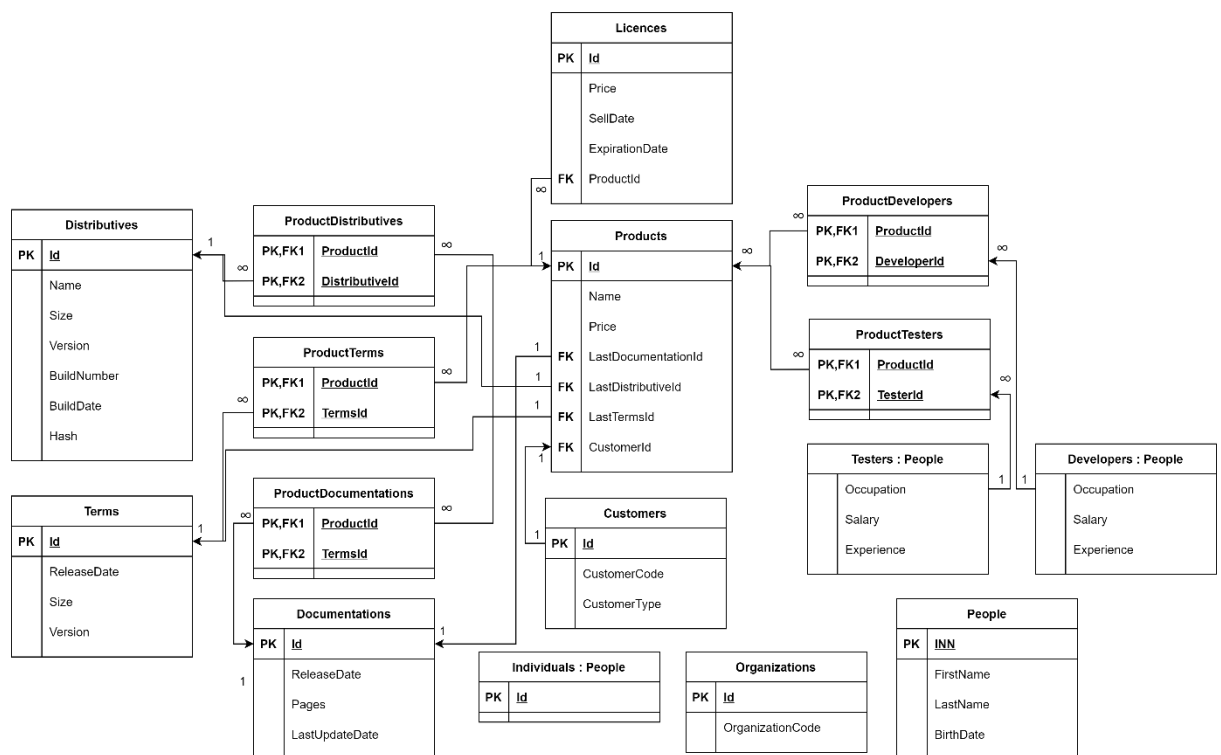
Комп'ютерний практикум №1

Тема: Створення програм на асемблері

Завдання:

Програмне забезпечення «Діяльність фірми з розробки програмних продуктів». Підприємства, що розробляють ПЗ, зазвичай мають декілька відділів, а саме: дирекція, бухгалтерія, маркетинговий відділ, відділ розробки ПЗ, відділ тестування ПЗ, відділ супроводження тощо. ПЗ, котре поставляється Замовнику, має назву, список розробників (внутрішній список тестувальників, котрий Замовнику не надається), вартість, документацію, дистрибутив, правила використання. Замовниками можуть бути як фізичні так і юридичні особи. Кожний Замовник має можливість замовити декілька ПП, на кожний з яких він отримує ліцензію, в якій вказано назву продукту, дату продажу, вартість, терміни апгрейдів.

Схема спроектованої бази даних



Під час декомпозиції задачі на об'єкти, було виявлено такі **доменні сутності**: Ліцензія, Продукт, Документація, Дистрибутив, Правила користування програмним застосунком. А також актори: Замовник, Розробник та Тестувальник.

Головною сутністю(коренем агрегації) є Продукт. Ця сутність містить інформацію таку інформацію про продукт: назва, ціна та посилання на таблиці документів, дистрибутивів, умов використання програми, та таблицю замовників. А також інформацію про розробників та тестувальників.

У таблиці продукту, Foreign Key вказують на останні версії дистрибутиву, документації та умови використання. Але зважаючи на те, що у кожного продукту може бути декілька версій — було створено таблиці ProductDistributives, ProductTerms та ProductDocumentations. Це дозволяє зберігати не лише найновіші версії, а й більш старі.

Так, як СУБД яка буде використовуватися для подальших робіт, **PostgreSQL**, підтримує наслідування таблиць схема проектувалась таким чином, щоб використовувати ці можливості та максимально зменшувати дублювання коду. Тому таблиці розробників, індивідуальних замовників та тестувальників наслідують таблицю People.

Таблиця ліцензій містить інформацію про продану ліцензію, а саме: ціну, дату продажу, дату завершення та посилання на продукт, до якого вона відноситься.

Таблиця замовників містить код замовника, це буде ІНН для фізичної особи або ж ЄДРПОУ для юридичної. А також тип замовлення за допомогою якого можна буде зрозуміти з якої додаткової таблиці брати дані.

Зв'язки таблиць

Розглянемо продукт, як корінь агрегації, тоді: кожен продукт буде мати лише одного замовника, безліч дистрибутивів, версій документації та правил користування. Отже, таблиця продуктів буде мати зв'язок 1 до 1, а з дистрибутивами, документацією та правилами використання 1(продукт) до 1, але буде посилатись на **останню версію**. Також створені 3 допоміжні таблиці ProductDistributives, ProductTerms та ProductDocumentations, які будуть мати зв'язок 1 до 1 між відповідними сутностями та дозволять робити версіонування.

Також є таблиця ліцензій, яка має зв'язок багато(ліцензій) до 1(продукту). Таблиці продукту і розробників/тестувальників мають зв'язок багато до багатьох, через те, що продукт може розробляти багато розробників/тестувальників і в одно час кожен з них може приймати участь в розробці декількох продуктів.

Допоміжні таблиці

Так як за умовою задачі може бути 2 типи замовники таблиця Customers містить код замовника та його тип. Використовуючи дані з таблиці замовника можна буде отримати інформацію про замовника з таблиці індивідуальних замовників чи компаній. А також для того, щоб зробити зв'язок багато до багатьох були створені ще 5 додаткових таблиць: ProductDistributives, ProductTerms та ProductDocumentations, ProductDevelopers та ProductTesters.

Висновок

В результаті виконання лабораторної роботи 2 було створено схему бази даних компанії, що займається розробкою ПО. Ця схема буде використовуватись в подальших лабораторних роботах, як сховище даних домених сутностей.