

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

Звіт до лабораторної роботи №4 з дисципліни

«Бази даних»

Прийняв:
Викладач кафедри ІІІ
Марченко О. І.
21 листопада 2021 року

Виконав
Студент групи ІТ-01
Бардін В. Д.

Лабораторна робота №3

Тема: Побудова простих запитів

Мета:

- Вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць, групування та сортування даних
- Навчитись використовувати вбудовані функції в запитах
- Вивчити призначення представлень (view) баз даних, синтаксису та семантики команд SQL для їх створення, зміни та видалення, системних збережених процедур для отримання інформації про представлення

Завдання: Програмне забезпечення «Діяльність фірми з розробки програмних продуктів». Підприємства, о 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

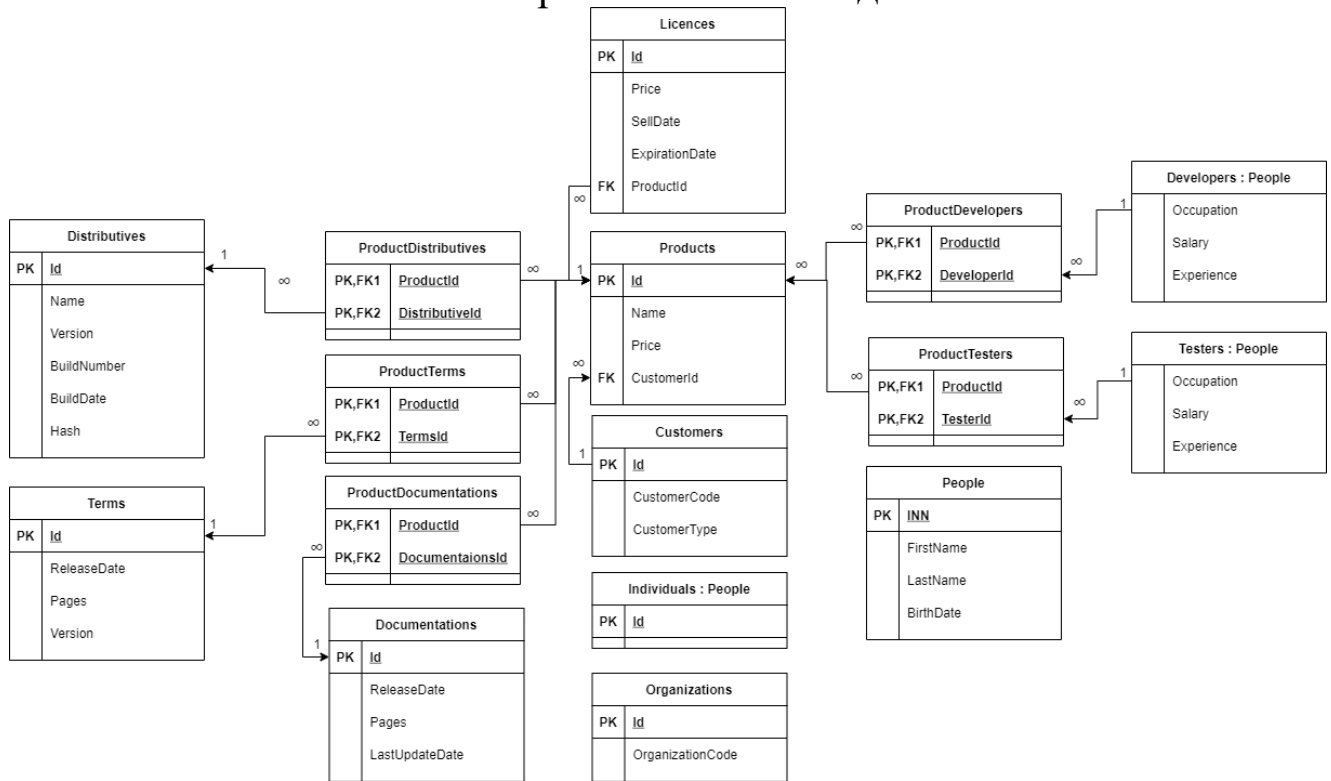


Рис. 1 — Схема БД

Для створення запитів буде використано базу попередніх лабораторних робіт. Схему БД наведено на рис. 1.

Частина 1: Запити з функціями агрегації

-- 1.a Count total amount of products for individual customers

```

SELECT CONCAT(INDS."FirstName", ' ', INDS."LastName") AS "Customer Name"
      , COUNT(PRODS."CustomerId") AS "Projects Amount"
FROM "Individuals" AS INDS
      JOIN "Customers" AS custs ON (custs."CustomerType" = 'Individual'
      AND custs."CustomerCode" = INDS."INN")
      JOIN "Products" AS PRODS ON custs."Id" = PRODS."CustomerId"
GROUP BY "Customer Name";
    
```

-- 1.b Summed price for all organizations customers price

```

SELECT ORGS."CompanyName"
      , SUM(prods."Price") AS "Total Price"
FROM "Organizations" AS ORGS
      JOIN "Customers" AS custs ON (custs."CustomerType" = 'Organization'
      AND custs."CustomerCode" = ORGS."OrganizationCode")
      JOIN "Products" AS prods ON custs."Id" = prods."CustomerId"
GROUP BY ORGS."CompanyName";
    
```

-- 1.c Select developers for products (with lower function)

```

SELECT prods."Id" AS "ProductId"
      , devs."INN" AS "DevTIN"
      , CONCAT(devs."FirstName", ' ', devs."LastName") AS "DevName"
      , devs."Occupation"
      , devs."Salary"
      , AGE(LOWER(devs."Experience")) AS "Experience"
FROM "Products" AS prods
      JOIN "ProductDevelopers" AS prodDevs ON prods."Id" = prodDevs."ProductId"
      FULL OUTER JOIN "Developers" AS devs ON prodDevs."DeveloperId" = devs."INN"
ORDER BY prods."Id";
    
```

-- 1.d Select documentations released during current month and that contains more than 500 pages

```
SELECT DOCS."Id" AS "DocsId"
      , DOCS."Pages"
      , DOCS."ReleaseDate"
      , DOCS."LastUpdateDate"
      , C."CustomerCode" AS "Customer Id"
FROM "Documentations" AS DOCS
      JOIN "ProductDocumentations" PD on DOCS."Id" = PD."DocumentationId"
      JOIN "Products" P on P."Id" = PD."ProductId"
      JOIN "Customers" C on P."CustomerId" = C."Id"
WHERE DOCS."Pages" > 500
      AND (date_part('year', now()) = date_part('year', DOCS."ReleaseDate"))
      AND date_trunc('month', now()) = date_trunc('month', DOCS."ReleaseDate");
```

-- 1.e Price for individual customers

```
SELECT CONCAT(INDS."FirstName", ' ', INDS."LastName") AS "Customer Name"
      , SUM(PRODS."Price") AS "Total Price"
FROM "Individuals" AS INDS
      JOIN "Customers" AS CUSTS ON (CUSTS."CustomerType" = 'Individual'
      AND CUSTS."CustomerCode" = INDS."INN")
      JOIN "Products" AS PRODS ON CUSTS."Id" = PRODS."CustomerId"
GROUP BY "Customer Name";
```

-- 1.f Price for individual customers

```
SELECT CONCAT(INDS."FirstName", ' ', INDS."LastName") AS "Customer Name"
      , COUNT(PRODS."CustomerId") AS "Products For Customer"
      , SUM(PRODS."Price") AS "Total Price"
FROM "Individuals" AS INDS
      JOIN "Customers" AS CUSTS ON (CUSTS."CustomerType" = 'Individual'
      AND CUSTS."CustomerCode" = INDS."INN")
      JOIN "Products" AS PRODS ON CUSTS."Id" = PRODS."CustomerId"
GROUP BY "Customer Name", PRODS."CustomerId"
HAVING SUM(PRODS."Price") > 120000;
```

-- 1.g Developers with 2 or more projects

```
SELECT CONCAT(DEVS."FirstName", ' ', DEVS."LastName") AS "Dev Name"
      , COUNT(PD."DeveloperId") AS "Projects Amount"
      , SUM(DEVS."Salary")
FROM "Developers" AS DEVS
      JOIN "ProductDevelopers" PD on DEVS."INN" = PD."DeveloperId"
GROUP BY "Dev Name"
HAVING COUNT(PD."DeveloperId") >= 2;
```

-- 1.h Get min terms pages amount that is more than 100

```
SELECT MIN(terms."Pages")
FROM "Terms" AS terms
      JOIN "ProductTerms" AS prodTerms ON terms."Id" = prodTerms."TermsId"
      FULL JOIN "Products" AS prods ON prodTerms."ProductId" = prods."Id"
HAVING MIN(terms."Pages") > 100;
```

-- 1.i Select all products terms by developer's id

```
SELECT concat(devs."FirstName", ' ', devs."LastName") AS "DevName"
      , prods."Id" AS "ProductId"
      , terms."Id" AS "TermsId"
      , terms."Pages"
      , terms."ReleaseDate"
      , terms."Version"
FROM "Terms" AS terms
      JOIN "Developers" AS devs ON devs."INN" = 1
      JOIN "ProductDevelopers" AS prodDevs ON prodDevs."DeveloperId" = 1
      JOIN "Products" AS prods ON prodDevs."ProductId" = prods."Id"
      JOIN "ProductTerms" AS prodTerms ON prods."Id" = prodTerms."ProductId"
      AND terms."Id" = prodTerms."TermsId"
ORDER BY terms."Id";
```

Частина 2: Робота з представленнями

-- 2.a

```
CREATE VIEW "Docs Released at December 2021" AS
SELECT DOCS."Id" AS "DocsId"
      , DOCS."Pages"
      , DOCS."ReleaseDate"
      , DOCS."LastUpdateDate"
      , C."CustomerCode" AS "Customer Id"
FROM "Documentations" AS DOCS
      JOIN "ProductDocumentations" PD on DOCS."Id" = PD."DocumentationId"
      JOIN "Products" P on P."Id" = PD."ProductId"
      JOIN "Customers" C on P."CustomerId" = C."Id"
WHERE DOCS."Pages" > 500
      AND (date_part('year', now()) = date_part('year', DOCS."ReleaseDate"))
      AND date_trunc('month', now()) = date_trunc('month', DOCS."ReleaseDate");
```

-- 2.b Select products which has docs that released in this month

```
CREATE VIEW "Products With Docs Released In December 2021" AS
SELECT P."Id" AS "Product Id"
      , "DOCS_2021".*
FROM "Documentations" AS DOCS
      JOIN "Docs Released at December 2021" "DOCS_2021" on DOCS."Id" =
"DOCS_2021"."DocsId"
      FULL OUTER JOIN "ProductDocumentations" PD on DOCS."Id" =
PD."DocumentationId"
      JOIN "Products" P ON PD."ProductId" = P."Id"
WHERE DOCS."Pages" > 100
      AND (date_part('year', now()) = date_part('year', DOCS."ReleaseDate"))
      AND date_trunc('month', now()) = date_trunc('month', DOCS."ReleaseDate");
```

-- 2.c Alter view

```
ALTER VIEW IF EXISTS "Products With Docs Released In December 2021"
      RENAME TO "Docs Released At Dec.2021";
```

-- 2.d View info

```
SELECT pg_describe_object(refclassid, refobjid, refobjsubid)
FROM pg_depend
WHERE objid = (SELECT objid FROM "SCompany-Main-DB".pg_catalog.pg_views WHERE
viewname = 'Docs Released at December 2021');
```

Висновок:

В результаті виконання даної лабораторної роботи було створено запити з використанням ф-цій агрегації, відображення, а також системних збережених процедур.