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

Звіт до лабораторної роботи №3 з дисципліни

«Бази даних»

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

Виконав
Студент групи ІТ-01
Бардін В. Д.

Київ – 2021

Лабораторна робота №3

Тема: Побудова простих запитів

Мета:

– Вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць

– Вивчити команди SQL для створення запитів з використанням під запитів та з’єднань

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

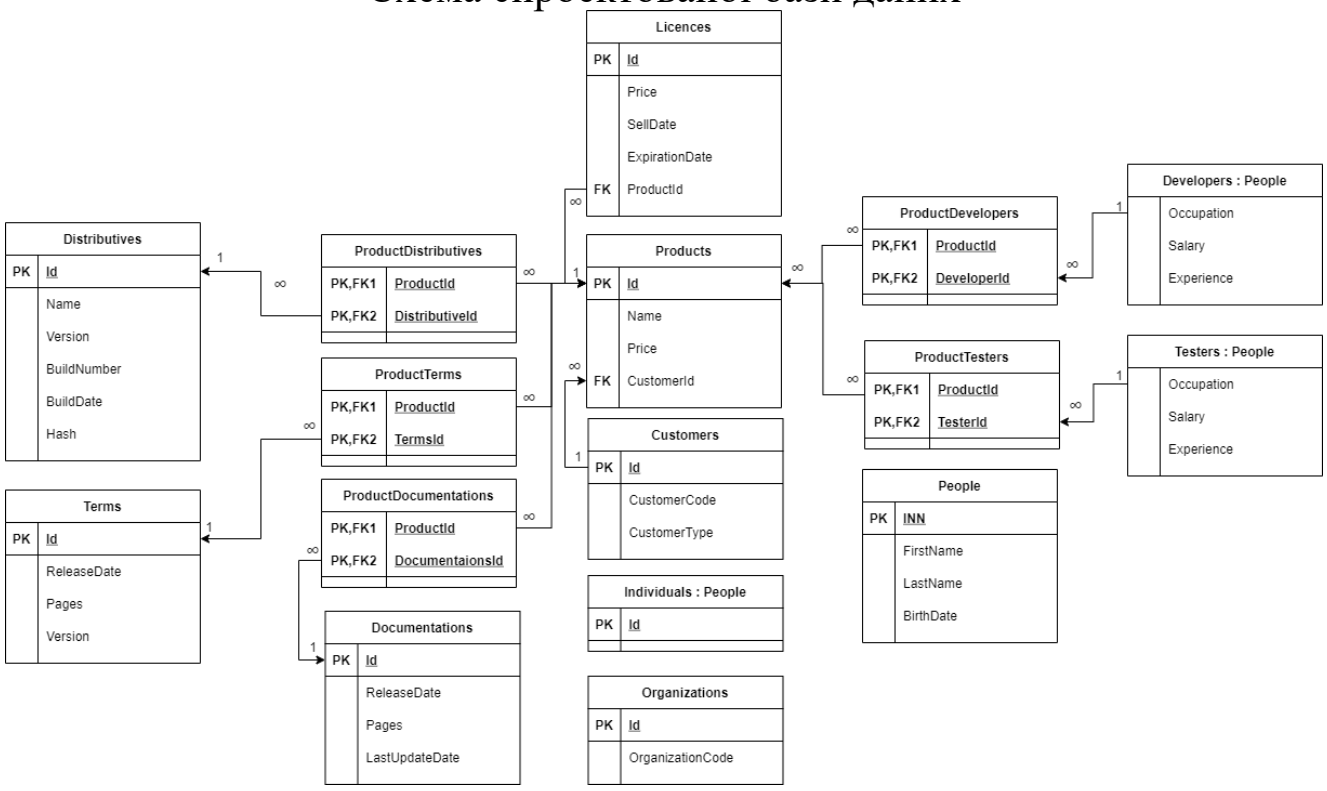

Рис. 1 — Схема БД

Для створення запитів буде використано базу, створену в 2-й лабораторній роботі, схему БД наведено на рис. 1.

Частина 1: Прості запити для вибірки

```
-- 1. Show all developers
SELECT devs."INN" AS "DevTIN"
     , concat(devs."FirstName", ' ', devs."LastName") AS "DevName"
     , devs."Occupation"
     , devs."Salary"
     , age(lower(devs."Experience")) AS "Experience"
FROM "Developers" AS devs;

-- 2. Select developer with TIN "1"
SELECT devs."INN" AS "DevTIN"
     , concat(devs."FirstName", ' ', devs."LastName") AS "DevName"
     , devs."Occupation"
     , devs."Salary"
     , age(lower(devs."Experience")) AS "Experience"
FROM "Developers" AS devs
WHERE devs."INN" = 1;

-- 3. Select developer by its first and last name
SELECT devs."INN" AS "DevTIN"
     , concat(devs."FirstName", ' ', devs."LastName") AS "DevName"
     , devs."Occupation"
     , devs."Salary"
     , age(lower(devs."Experience")) AS "Experience"
FROM "Developers" AS devs
WHERE devs."FirstName" = 'Vladislav' AND devs."LastName" = 'Bardin';

-- 4. Select developers by names
SELECT devs."INN" AS "DevTIN"
     , concat(devs."FirstName", ' ', devs."LastName") AS "DevName"
     , devs."Occupation"
```

```
     , devs."Salary"
     , age(lower(devs."Experience")) AS "Experience"
FROM "Developers" AS devs
WHERE devs."FirstName" = 'Vladislav' OR devs."FirstName" = 'Ivan';

-- 5. Select developers whose names not like a specified one
SELECT devs."INN" AS "DevTIN"
     , concat(devs."FirstName", ' ', devs."LastName") AS "DevName"
     , devs."Occupation"
     , devs."Salary"
     , age(lower(devs."Experience")) AS "Experience"
FROM "Developers" AS devs
WHERE devs."FirstName" NOT LIKE 'Vladislav';

-- 6. Select developers whose salary is higher than 700 and less then 1500
SELECT devs."INN" AS "DevTIN"
     , concat(devs."FirstName", ' ', devs."LastName") AS "DevName"
     , devs."Occupation"
     , devs."Salary"
     , age(lower(devs."Experience")) AS "Experience"
FROM "Developers" AS devs
WHERE '[700,1500]'::numrange @> devs."Salary";

-- 7. Select all testers who is older than 18
SELECT QAs."INN" AS "DevTIN"
     , concat(QAs."FirstName", ' ', QAs."LastName") AS "DevName"
     , age(QAs."BirthDate") AS "Age"
     , QAs."Occupation"
     , QAs."Salary"
     , age(lower(QAs."Experience")) AS "Experience"
FROM "Testers" AS QAs
WHERE extract(YEAR FROM age(QAs."BirthDate"::date)) > 18;

-- 8. Select testers whose birthday is less than in month
SELECT concat(QAs."FirstName", ' ', QAs."LastName") AS "Name"
FROM "Testers" AS QAs
WHERE to_char(QAs."BirthDate", 'MM-DD') <= to_char(now() + '1 month'::interval, 'MM-
DD');

-- 9. Select all distributives released at 2021
SELECT distrs."Id" AS "DistributiveId"
     , distrs."Name" AS "Name"
     , distrs."BuildDate"
     , distrs."Version"
     , distrs."Hash"
FROM "Distributives" AS distrs
WHERE date_part('year', distrs."BuildDate") = 2021;

-- 10. Select documentations released during current month and that contains more
that 500 pages
SELECT docs."Id" AS "DocsId"
     , docs."Pages"
     , docs."ReleaseDate"
     , docs."LastUpdateDate"
FROM "Documentations" AS docs
WHERE docs."Pages" > 500 AND (date_part('year', now()) = date_part('year',
docs."ReleaseDate")
  AND date_trunc('month', now()) = date_trunc('month', docs."ReleaseDate"));

-- 11. Select developers whose first + last name ILIKE 'Vladislav Bardin'
SELECT devs."INN" AS "DevTIN"
     , concat(devs."FirstName", ' ', devs."LastName") AS "DevName"
     , devs."Occupation"
     , devs."Salary"
     , age(lower(devs."Experience")) AS "Experience"
FROM "Developers" AS devs
```

```
WHERE concat(devs."FirstName", ' ', devs."LastName") ILIKE 'Vladislav Bardin';

-- 12. Select licenses that costs more than 500$ and will ends in a month or costs
more than 100$ and 'll ends in a week
SELECT licenses."ProductId"
     , licenses."Id" AS "LicenseId"
     , licenses."SellDate"
     , licenses."ExpirationDate"
FROM "Licences" AS licenses
WHERE licenses."Price" > 500 AND licenses."ExpirationDate" <= now() + '1
month'::interval
   OR licenses."Price" > 100 AND licenses."ExpirationDate" <= now() + '1
week'::interval;

-- 13. Select all distributives that match 'X-%' pattern
SELECT distrs."Id" AS "DistributiveId"
     , distrs."Name" AS "Name"
     , distrs."BuildDate"
     , distrs."Version"
     , distrs."Hash"
FROM "Distributives" AS distrs
WHERE distrs."Name" LIKE 'X-%';

-- 14. Select terms that consists of [300 - 450] pages
SELECT terms."Id"
     , terms."Pages"
     , terms."ReleaseDate"
FROM "Terms" AS terms
WHERE '[300,450]'::int4range @> terms."Pages";

-- 15. Select organizations that contains 'Star' in its name
SELECT orgs."Id" AS "OrganizationId"
     , orgs."OrganizationCode"
     , orgs."CompanyName"
FROM "Organizations" AS orgs
WHERE "CompanyName" ILIKE '%star%';
```

Частина 2: запити з використанням під запитів та з’єднань

```
-- 1. Select all product's developers
SELECT concat(devs."FirstName", ' ', devs."LastName") AS "Name"
     , "Occupation"
     , "Salary"
     , prodDevs."ProductId" AS "ProjectId"
FROM "Developers" AS devs
LEFT JOIN "ProductDevelopers" AS prodDevs ON (devs."INN" = prodDevs."DeveloperId"
                                              AND prodDevs."ProductId" =
'4a276c2e-ec71-416c-afc0-a28e24ad7b0c')
ORDER BY devs."INN";

-- 2. Select testers for product
SELECT concat(QAs."FirstName", ' ', QAs."LastName") AS "Name"
     , QAs."Occupation"
     , QAs."Salary"
     , age(lower(QAs."Experience")) AS "Experience"
FROM "Testers" AS QAs
WHERE "INN" = (
    SELECT "TesterId"
    FROM "ProductTesters"
    WHERE "ProductId" = '4a276c2e-ec71-416c-afc0-a28e24ad7b0c'
          )
ORDER BY "FirstName", "LastName";

-- 3. Select all products distributives by developer's id
SELECT concat(devs."FirstName", ' ', devs."LastName")
     , prods."Id" AS "ProductId"
```

```
     , distrs."Id" AS "TermsId"
     , distrs."BuildDate"
     , distrs."Version"
     , distrs."Hash"
FROM "Distributives" AS distrs
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 "ProductDistributives" AS prodDistr ON prods."Id" = prodDistr."ProductId"
                                            AND distrs."Id" =
prodDistr."DistributiveId";

-- 4. 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";

-- 5. Select developers for products
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";

-- 6. Select licenses for products
SELECT licenses."ProductId"
     , licenses."Id" AS "LicenseId"
     , licenses."SellDate"
     , licenses."ExpirationDate"
FROM "Licences" AS licenses
JOIN "Products" AS prods ON licenses."ProductId" = prods."Id"
ORDER BY licenses."Id";

-- 7. Select all individuals customers
SELECT inds."Id" AS "CustomerId"
     , inds."INN" AS "TIN"
     , concat(inds."FirstName", ' ', inds."LastName")
FROM "Individuals" AS inds
JOIN "Customers" AS custs ON ("CustomerType" = 'Individual'
                              AND custs."CustomerCode" = inds."INN");

-- 8. Select documentations for products
SELECT prods."Id"
     , docs."Id"
     , docs."ReleaseDate"
     , docs."Pages"
     , docs."LastUpdateDate"
FROM "Products" AS prods
JOIN "ProductDocumentations" AS prodDocs ON prods."Id" = prodDocs."ProductId"
LEFT JOIN "Documentations" AS docs on prodDocs."DocumentationId" = docs."Id"
ORDER BY prods."Id";
```

```sql
-- 9. Select terms for projects
SELECT prods."Id"
     , terms."Id"
     , terms."Pages"
     , terms."ReleaseDate"
FROM "Terms" AS terms
JOIN "ProductTerms" AS prodTerms on terms."Id" = prodTerms."TermsId"
RIGHT JOIN "Products" AS prods on prods."Id" = prodTerms."ProductId"
ORDER BY prods."Id";

-- 10. Show all possible pairs Developer-Tester
SELECT concat(devs."FirstName", ' ', devs."LastName") AS dev
     , concat(QAs."FirstName", ' ', QAs."LastName") AS QA
FROM "Developers" AS devs
CROSS JOIN "Testers" AS QAs;

-- 11. Get terms that contains from 300 to 450 pages for products
SELECT prods."Id"
     , terms."Id"
     , 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"
WHERE '[300,450]'::int4range @> terms."Pages";

-- 12. Select developers for projects
SELECT prods."Id"
     , devs."FirstName"
     , devs."LastName"
     , devs."Occupation"
     , devs."Salary"
FROM "Developers" AS devs
JOIN "ProductDevelopers" AS prodDevs on devs."INN" = prodDevs."DeveloperId"
FULL OUTER JOIN "Products" AS prods ON prodDevs."ProductId" = prods."Id"
ORDER BY prods."Id", devs."INN";

-- 13. Select all testers for customer
SELECT prods."Id"
     , custs."Id"
     , QAs."FirstName"
     , QAs."LastName"
     , QAs."Occupation"
     , QAs."Salary"
FROM "Testers" AS QAs
JOIN "ProductTesters" AS prodQAs ON QAs."INN" = prodQAs."TesterId"
JOIN "Products" AS prods ON prodQAs."ProductId" = prods."Id"
FULL JOIN "Customers" AS custs ON prods."CustomerId" = custs."Id"
ORDER BY prods."Id", QAs."INN";

-- 14. Select products for individual customers
SELECT concat(inds."FirstName", ' ', inds."LastName") AS "CustomerName"
     , prods."Id" AS "ProjectId"
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"
ORDER BY prods."Id";

-- 15. Select products for organization customers
SELECT orgs."CompanyName"
     , prods."Id" AS "ProjectId"
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"
ORDER BY prods."Id";
```

Висновок:

В результаті виконання даної лабораторної роботи було створено 30 запитів, для вибору даних, з БД. Розглянуто різні види об`єднань, а також об’єднань таблиць.