

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Лабораторна робота №1
з дисципліни «Бази даних»

**«Проектування бази даних та ознайомлення з базовими
операціями СУБД PostgreSQL»**

Виконав студент групи: КВ-33

ПІБ: Поляков Єгор Олегович

Перевірив:

Київ 2025

Постановка задачі:

Метою роботи є здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

Завдання роботи полягає у наступному:

1. Розробити модель «сутність-зв'язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожен з таблиць засобами pgAdmin 4.

Завдання №1:

Опис предметної галузі

Тема моєї бази даних: **Електронний довідник для зберігання технічної документації** - це база даних, призначена для централізованого збереження, упорядкування та швидкого доступу до технічних матеріалів. Вона дозволить накопичувати, структурувати та оновлювати документацію, забезпечує зручний пошук і перегляд даних, а також сприяє ефективній організації роботи з технічною інформацією відповідно до вимог підприємства чи навчального закладу.

Розробка концептуальної моделі

Опис сутностей

При створенні даної бази даних я виділив такі сутності:

1. Працівник (Employee)

Атрибути: ім'я, прізвище, електронна пошта, відділ.

Призначення: зберігає дані про тих, хто працює з документами.

2. Документ (Document)

Атрибути: назва, опис, дата створення, дата оновлення, версія.

Призначення: зберігає основну інформацію про технічну документацію.

3. Категорія (Category)

Атрибути: назва, опис, тип.

Призначення: групує документи за темами чи напрямками.

4. Файл (File)

Атрибути: назва, формат, розмір, дата завантаження.

Призначення: зберігає електронні файли, прикріплені до документа.

5. Доступ (Access)

Атрибути: рівень доступу (читання / редагування / видалення), дата надання доступу, строк дії доступу.

Призначення: регулює права доступу працівників до документів.

Опис зв'язків між сутностями

Зв'язок «Працівник – Документ» є зв'язком 1:N. Один документ має лише одного автора, але один працівник може створювати багато документів.

Зв'язок «Категорія – Документ» є зв'язком 1:N. Один документ належить лише до однієї категорії, але одна категорія може містити багато документів.

Зв'язок «Документ – Файл» є зв'язком 1:1. Один файл належить лише одному документу, та один документ може мати один файл.

Зв'язок «Працівник – Доступ – Документ» є зв'язком N:M. Один працівник може мати доступ до багатьох документів, і один документ може бути доступний багатьом працівникам.

Графічне подання концептуальної моделі «Сутність-зв'язок» зображено на рис. 1.

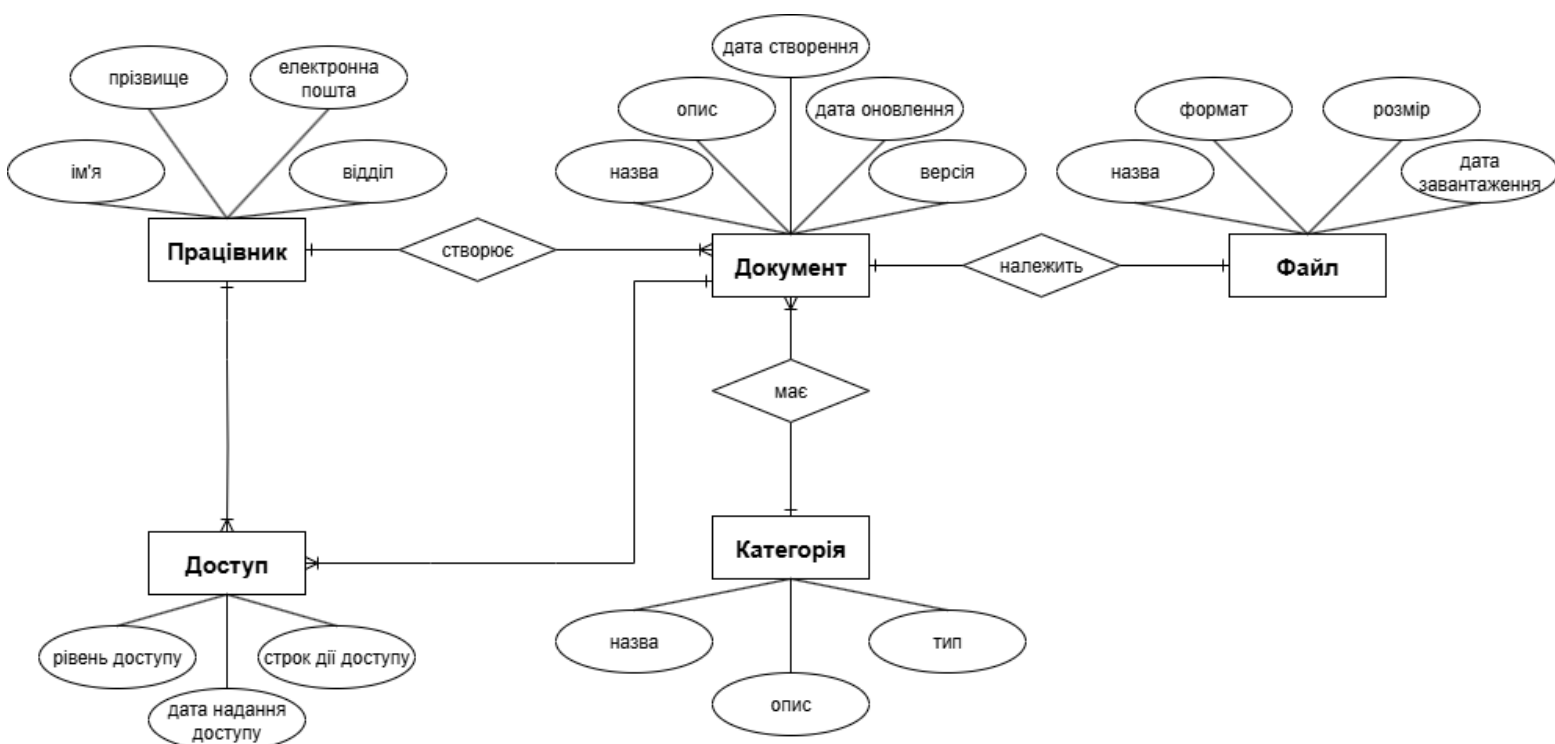


Рис. 1 – ER-діаграма побудована за нотацією “Пташина лапка”

Завдання №2:

Перетворення концептуальної моделі у логічну модель та схему бази даних

Сутність «Працівник» перетворено в таблицю Employee з первинним ключем employee_id та атрибутами first_name, last_name, email, department.

Сутність «Категорія» перетворено в таблицю Category з первинним ключем category_id та атрибутами category_name, description, type.

Сутність «Документ» перетворено в таблицю Document з первинним ключем document_id та атрибутами title, description, date_created, date_updated, version. У таблиці є зовнішні ключі employee_id та category_id, які пов'язані з таблицями Employee та Category відповідно.

Сутність «Файл» перетворено в таблицю File з первинним ключем file_id та атрибутами file_name, format, size, upload_date. У таблиці є зовнішній ключ document_id, який пов’язаний з таблицею Document.

Сутність «Доступ» перетворено в таблицю Access з первинним ключем access_id та атрибутами access_level, date_granted, expiry_date. У таблиці є зовнішні ключі employee_id та document_id, які пов’язані з таблицями Employee та Document відповідно.

Зв’язок 1:1 реляційна модель не підтримує як окремий, унікальний механізм, тому для його реалізації використовується наступний прийом: зв’язок між сутностями «Документ» та «Файл» будується як зв’язок 1:N (один-до-багатьох). При цьому логіка системи та бізнес-правила гарантують, що в результаті один файл належить лише одному документу, і один документ може мати лише один файл.

Графічне подання логічної моделі «Сутність-зв’язок» зображено на рис. 2.

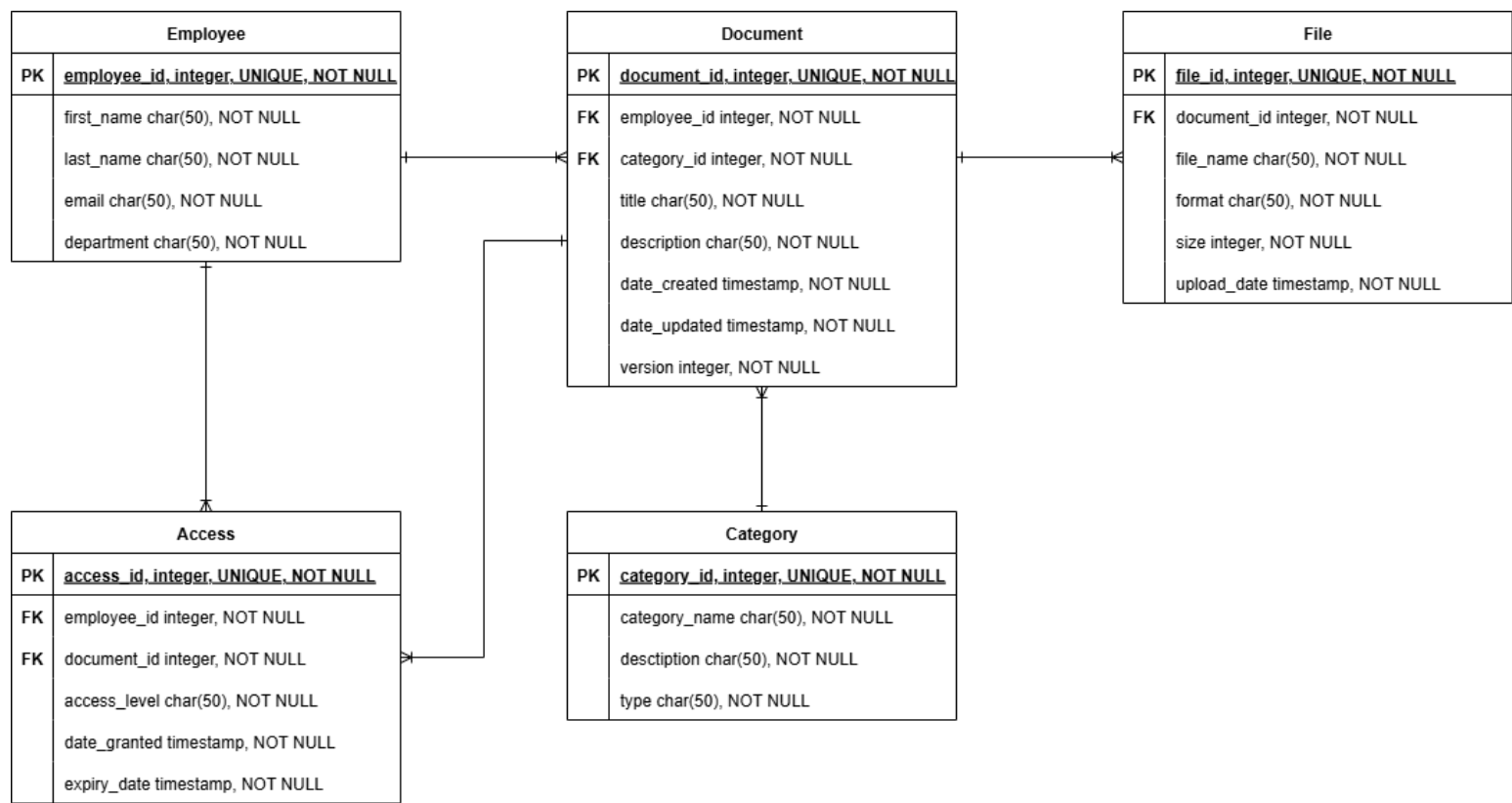


Рис. 2 – Схема бази даних

Завдання №3:

Нормалізація схеми бази даних до третьої нормальної форми (3НФ)

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої:

Сутність	Атрибут	Тип атрибуту
Employee – містить дані	employee_id – унікальний ідентифікатор працівника	integer (числовий)
	first_name – ім’я працівника	character varying (рядок)

про працівників	<i>last_name</i> – прізвище працівника	character varying (рядок)
	<i>email</i> – електронна пошта працівника	character varying (рядок)
	<i>department</i> – відділ, де працює працівник	character varying (рядок)
Category – містить інформацію про категорії документів	<i>category_id</i> – унікальний ідентифікатор категорії	integer (числовий)
	<i>category_name</i> – назва категорії	character varying (рядок)
	<i>description</i> – опис категорії	character varying (рядок)
	<i>type</i> – тип категорії	character varying (рядок)
Document – містить дані про документи	<i>document_id</i> – унікальний ідентифікатор документа	integer (числовий)
	<i>employee_id</i> – ідентифікатор автора(працівника) документа	integer (зовнішній ключ до Employee)
	<i>category_id</i> – ідентифікатор категорії	integer (зовнішній ключ до Category)
	<i>title</i> – назва документа	character varying (рядок)
	<i>description</i> – опис документа	character varying (рядок)
	<i>date_created</i> – дата створення документа	timestamp (дата та час)
	<i>date_updated</i> – дата останнього оновлення	timestamp (дата та час)
	<i>version</i> – версія документа	integer (числовий)
File – містить інформацію про файли документа	<i>file_id</i> – унікальний ідентифікатор файлу	integer (числовий)
	<i>document_id</i> – ідентифікатор документа	integer (зовнішній ключ до Document)
	<i>file_name</i> – назва файлу	character varying (рядок)
	<i>format</i> – формат файлу	character varying (рядок)
	<i>size</i> – розмір файлу	integer (числовий)
	<i>upload_date</i> – дата завантаження файлу	timestamp (дата та час)
Access – регулює права доступу працівників до документів	<i>access_id</i> – унікальний ідентифікатор доступу	integer (числовий)
	<i>employee_id</i> – ідентифікатор працівника	integer (зовнішній ключ до Employee)
	<i>document_id</i> – ідентифікатор документа	integer (зовнішній ключ до Document)
	<i>access_level</i> – рівень доступу (читання, редагування, видалення)	character varying (рядок)
	<i>date_granted</i> – дата надання доступу	timestamp (дата та час)
	<i>expiry_date</i> – строк дії доступу	timestamp (дата та час)

Функціональні залежності для кожної таблиці

1. Employee:

employee_id -> first_name, last_name, email, department

2. Category:

category_id -> category_name, description, type

3. Document:

document_id -> employee_id, category_id, title, description, date_created, date_updated, version,

employee_id -> category_id, title, description, date_created, date_updated, version

category_id -> employee_id, title, description, date_created, date_updated, version

4. File:

file_id -> document_id, file_name, format, size, upload_date

document_id -> file_name, format, size, upload_date

5. Access:

access_id -> employee_id, document_id, access_level, date_granted, expiry_date

employee_id -> document_id, access_level, date_granted, expiry_date

document_id -> employee_id, access_level, date_granted, expiry_date

Ці функціональні залежності відображають взаємозв'язки між атрибутами в таблицях бази даних, показуючи, які з них однозначно визначають інші. Це відіграє ключову роль у процесі нормалізації та побудові ефективної структури бази даних, адже коректне визначення залежностей допомагає уникати дублювання інформації та підтримувати цілісність даних.

Транзитивна функціональна залежність виникає тоді, коли один атрибут визначає інший опосередковано — через третій атрибут. Наприклад, якщо атрибут A -> B, а також B -> C, то можна сказати, що існує транзитивна залежність A -> C. Наявність таких залежностей зазвичай свідчить про приховану надмірність і потребу у подальшій нормалізації (зокрема переході до третьої нормальної форми).

У даній моделі транзитивних функціональних залежностей не виявлено. Це означає, що структура бази даних побудована грамотно: надмірність відсутня, атрибути логічно розділені між таблицями, а дані зберігаються у цілісному й оптимальному вигляді.

Відповідність схеми нормальним формам

1. Перша нормальна форма (1НФ)

Щоб задовольнити умови 1НФ, усі атрибути в таблицях мають бути атомарними, а записи — унікальними.

У даній моделі всі атрибути є неподільними (наприклад, ім'я та прізвище працівника зберігаються окремо), а ідентифікація здійснюється за допомогою первинних ключів (employee_id, document_id тощо). Це гарантує унікальність кожного запису. Таким чином, схема **відповідає вимогам 1НФ**.

2. Друга нормальна форма (2НФ)

Щоб схема перебувала у 2НФ, необхідно, щоб:

- Схема перебуває в 1НФ;
- Кожний неключовий атрибут функціонально залежить від цілого ключа, а не його частини.

У нашій моделі ці умови виконуються: усі атрибути залежать від первинних ключів таблиць. Наприклад, у таблиці **Employee** атрибути first_name, last_name, email, department залежать саме від employee_id. Часткових залежностей немає. Отже, схема **відповідає 2НФ**.

3. Третя нормальна форма (3НФ)

Щоб відповідати 3НФ необхідно:

- Схема перебуває в 2НФ;
- Кожен неключовий атрибут має бути не залежним від будь-якого іншого ключового атрибута.

Оскільки схема в НФ2 та моделі транзитивних залежностей не виявлено. Наприклад, у таблиці **Document** атрибути (title, description, date_created, date_updated, version) залежать безпосередньо від document_id, а не через інші неключові атрибути. Таким чином, схема також **відповідає 3НФ**.

Висновок:

Схема бази даних відповідає нормальним формам НФ1, НФ2 та НФ3. Вона є логічно структурованою, уникає надмірності та забезпечує цілісність і ефективне управління даними.

Завдання №4:

Інструментарій PostgreSQL та pgAdmin 4

Схема бази даних у pgAdmin 4 зображено на рисунку 3.

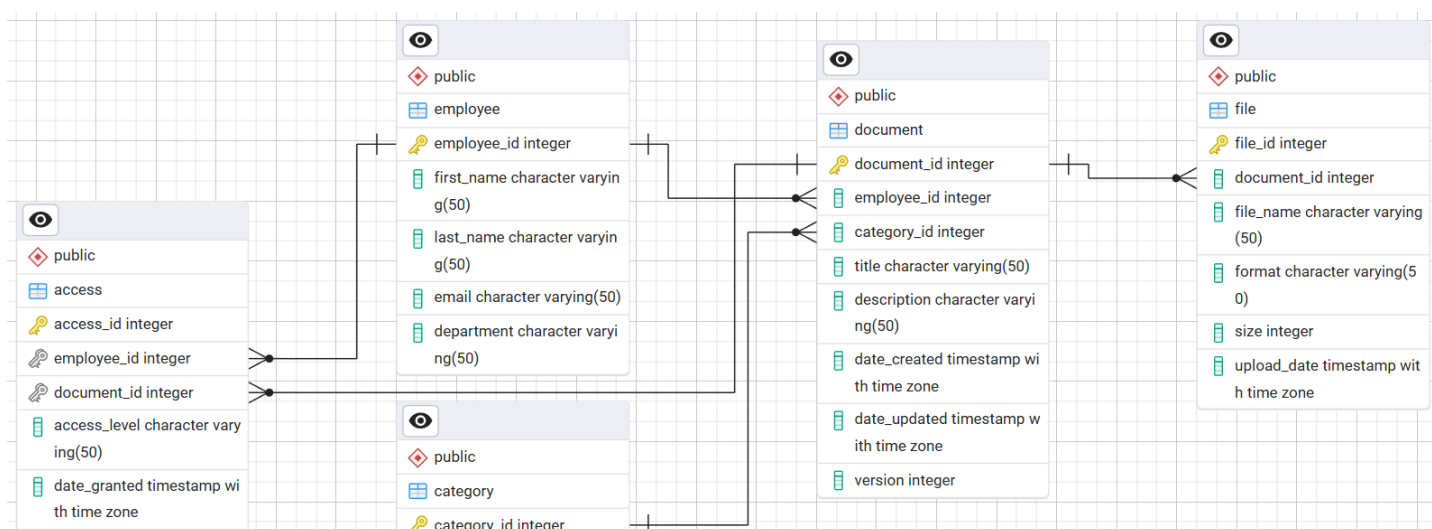


Рисунок 3 - Схема бази даних у pgAdmin 4

Таблиці бази даних у pgAdmin4

- Tables (5)
 - access
 - document
 - employee
 - file
 - category

Employee

Table: access (public)

×

General Columns Advanced Constraints

Columns

+

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
⋮	✎	access_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
⋮	✎	employee_id	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
⋮	✎	document_id	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
⋮	✎	access_level	character varying	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
⋮	✎	date_granted	timestamp with ti			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
⋮	✎	expiry_date	timestamp with ti			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Close

Reset

Save

Table: employee (public)

×

General Columns Advanced Constraints

Primary Key Foreign Key Unique

	Name	Columns
✎	employee_pkey	employee_id

Close

Reset

Save

У цієї таблиці немає зовнішніх ключів (foreign key)



	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (50)	department character varying (50)
1	1	Іван	Петренко	ivan.petrenko@example.com	Відділ розробки
2	2	Олена	Коваленко	olena.kovalenko@example.com	Відділ аналітики

Access

Table: access (public)



General Columns Advanced Constraints

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
⋮	✎	access_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
⋮	✎	employee_id	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
⋮	✎	document_id	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
⋮	✎	access_level	character varying	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
⋮	✎	date_granted	timestamp with time zone			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
⋮	✎	expiry_date	timestamp with time zone			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Close

Reset

Save

Table: access (public)



General Columns Advanced Constraints

Primary Key Foreign Key Unique

Primary Key

	Name	Columns
✎	access_pkey	access_id

Close

Reset

Save

Table: access (public)



General Columns Advanced Constraints

Primary Key Foreign Key Unique

Foreign Key

	Name	Columns	Referenced Table
✎	document_ace_fkey	(document_id) -> (document_id)	document
✎	employee_ace_fkey	(employee_id) -> (employee_id)	employee

Close

Reset

Save

Data OutputMessagesNotifications

Showing rows: 1 to 2Page No: 1 of 1

	access_id [PK] integer	employee_id integer	document_id integer	access_level character varying (50)	date_granted timestamp with time zone	expiry_date timestamp with time zone
1	301	1	101	редагування	2023-10-01 13:00:00+03	2033-10-01 13:00:00+03
2	302	2	101	читання	2023-10-20 14:00:00+03	2024-10-20 14:00:00+03

Document

Table: document (public)

×

GeneralColumnsAdvancedConstraints

Columns

+

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
<div></div>	<div></div>	document_id	integer			<div></div>	<div></div>	
<div></div>	<div></div>	employee_id	integer			<div></div>	<div></div>	
<div></div>	<div></div>	category_id	integer			<div></div>	<div></div>	
<div></div>	<div></div>	title	character varying	50		<div></div>	<div></div>	
<div></div>	<div></div>	description	character varying	50		<div></div>	<div></div>	
<div></div>	<div></div>	date_created	timestamp with time zone			<div></div>	<div></div>	
<div></div>	<div></div>	date_updated	timestamp with time zone			<div></div>	<div></div>	
<div></div>	<div></div>	version	integer			<div></div>	<div></div>	

×

 Close

↺

 Reset

💾

 Save

Table: document (public)

×

GeneralColumnsAdvancedConstraints

Primary KeyForeign KeyUnique

+

	Name	Columns
<div></div>	document_pkey	document_id

×

 Close

↺

 Reset

💾

 Save

Table: document (public)



General Columns Advanced Constraints

Primary Key Foreign Key Unique

	Name	Columns	Referenced Table	
		employee_doc_fkey	(employee_id) -> (employee_id)	employee
		category_doc_fkey	(category_id) -> (category_id)	category

Close Reset Save

Data Output Messages Notifications

Showing rows: 1 to 2 Page No: 1 of 1

	document_id [PK] integer	employee_id integer	category_id integer	title character varying (50)	description character varying (50)	date_created timestamp with time zone	date_updated timestamp with time zone	version integer
1	101	1	10	Специфікація API для модуля авторизації	Опис методів та кінцевих точок API v2.1	2023-10-01 13:00:00+03	2023-10-15 17:30:00+03	2
2	102	2	11	Інструкція з налаштування 2FA	Гайд щодо підвищення безпеки акаун...	2023-11-05 11:00:00+02	2023-11-05 11:00:00+02	1

File

Table: file (public)



General Columns Advanced Constraints

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
			file_id	integer		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
			document_id	integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
			file_name	character varying	50	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
			format	character varying	50	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
			size	integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
			upload_date	timestamp with tii		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Close Reset Save

Table: file (public)



General Columns Advanced Constraints

Primary Key Foreign Key Unique



	Name	Columns	
		file_pkey	file_id

Close Reset Save

Table: file (public) ✕










GeneralColumnsAdvancedConstraints

Primary KeyForeign KeyUnique







	Name	Columns	Referenced Table
 	document_fil_fkey	(document_id) -> (document_id)	document

✕ Close↺ Reset💾 Save

Data OutputMessagesNotifications



Sh

















	file_id [PK] integer 	document_id integer 	file_name character varying (50) 	format character varying (50) 	size integer 	upload_date timestamp with time zone 
1	201	101	auth_api_v2.1	PDF	1024	2023-10-01 13:05:00+03
2	202	102	2fa_guide	DOCX	512	2023-11-05 11:10:00+02

Category

Table: category (public) ✕

GeneralColumnsAdvancedConstraints

Columns



	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
  	category_id	integer 			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text"/>
  	category_name	character varying 	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
  	description	character varying 	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
  	type	character varying 	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>

✕ Close↺ Reset💾 Save

Table: category (public) ✕

GeneralColumnsAdvancedConstraints

Primary KeyForeign KeyUnique

	Name	Columns
 	category_pkey	category_id

✕ Close↺ Reset💾 Save

У цієї таблиці немає зовнішніх ключів (foreign key)

Data Output

Messages

Notifications

	category_id [PK] integer	category_name character varying (50)	description character varying (50)	type character varying (50)
1	10	Технічні специфікації	Детальні описи продуктів та архітектурних рішень	Внутрішня
2	11	Посібники користувача	Інструкції та гайди для кінцевих користувачів	Публічна

Таблиці в коді SQL

```
BEGIN;
```

```
CREATE TABLE IF NOT EXISTS public.employee
(
    employee_id integer NOT NULL,
    first_name character varying(50) NOT NULL,
    last_name character varying(50) NOT NULL,
    email character varying(50) NOT NULL,
    department character varying(50) NOT NULL,
    PRIMARY KEY ("employee_id")
);
```

```
CREATE TABLE IF NOT EXISTS public."category"
(
    category_id integer NOT NULL,
    category_name character varying(50) NOT NULL,
    description character varying(50) NOT NULL,
    type character varying(50) NOT NULL,
    PRIMARY KEY (category_id)
);
```

```
CREATE TABLE IF NOT EXISTS public.document
(
    document_id integer NOT NULL,
    employee_id integer NOT NULL,
    category_id integer NOT NULL,
    title character varying(50) NOT NULL,
    description character varying(50) NOT NULL,
    date_created timestamp with time zone NOT NULL,
    date_updated timestamp with time zone NOT NULL,
    version integer NOT NULL,
    PRIMARY KEY (document_id)
);
```

```
CREATE TABLE IF NOT EXISTS public.file
(
    file_id integer NOT NULL,
    document_id integer NOT NULL,
    file_name character varying(50) NOT NULL,
    format character varying(50) NOT NULL,
    size integer NOT NULL,
    upload_date timestamp with time zone NOT NULL,
    PRIMARY KEY (file_id)
);
```

```
CREATE TABLE IF NOT EXISTS public.access
(
    access_id integer NOT NULL,
    employee_id integer NOT NULL,
    document_id integer NOT NULL,
    access_level character varying(50) NOT NULL,
    date_granted timestamp with time zone NOT NULL,
    expiry_date timestamp with time zone NOT NULL,
```

```

        PRIMARY KEY ("access_id")
    );

ALTER TABLE IF EXISTS public.document
    ADD FOREIGN KEY ("employee_id")
    REFERENCES public.employee (employee_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;

ALTER TABLE IF EXISTS public.document
    ADD FOREIGN KEY ("category_id")
    REFERENCES public."category" (category_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;

ALTER TABLE IF EXISTS public.file
    ADD FOREIGN KEY ("document_id")
    REFERENCES public.document (document_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;

ALTER TABLE IF EXISTS public.access
    ADD FOREIGN KEY (document_id)
    REFERENCES public.document (document_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;

ALTER TABLE IF EXISTS public.access
    ADD FOREIGN KEY (employee_id)
    REFERENCES public.employee (employee_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;

END;
```

Контакти:

Penozumopiï GIT: https://github.com/PixelPal/Databases_course.git

Telegram: @justz_egor