A black and white drawing of a building

Description automatically generated

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського» ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

# Кафедра системного програмування та спеціалізованих комп’ютерних систем

**Лабораторна робота №1**

з дисципліни **Бази даних і засоби управління**

*на тему: “Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL”*

Виконав:

студент ІІI курсу

групи КВ-12

Прокопенко С.П.

Перевірив:

Павловский В. І.

Київ – 2023

**Мета:** здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

**Завдання:**

1. Розробити модель «сутність зв’язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».

2. Перетворити створену модель у схему бази даних (таблиці) PostgreSQL.

3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).

4. Ознайомитись із інструментарієм PosgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

**Опис предметної області**

Дана предметна область — онлайн-майданчик для оренди спорядження. Онлайн-майданчик для оренди спорядження це програмний продукт, що забезпечує можливість знаходити, орендувати та використовувати певні види спорядження.

**Опис сутностей**

1. Користувач (User)

Атрибути: UserID, Name, Email, Number

Призначення: зберігає інформацію про користувача

1. Оренда (Rental)

Атрибути: RentalID, Duration, RentalDate, ReturnDate, UserID, Equipmen

Призначення: зберігає інформацію про оренду спорядження

1. Спорядження (Equipment)

Атрибути: EquipmentID, Name, RentalCost

Призначення: зберігає інформацію про спорядження.

**Опис зв’язків між сутностями**

Зв’язок “User” – “Rental” 1:N, де “Rental” — сутність, оскільки кожен користувач може укладати декілька оренд спорядження.

Зв’язок “Rental” — “Equipment” 1:1, так як кожна оренда має відношення лише до однієї одиниці спорядження.

Графічне подання концептуальної моделі «Сутність-зв’язок» за нотацією Чена зображено на рисунку 1.

Изображение выглядит как диаграмма, рисунок, шаблон, оригами

Автоматически созданное описание

Рисунок 1 — ER-діаграма, побудована за нотацією Чена

**Перетворення концептуальної моделі у логічну модель та схему бази даних**

Сутність User перетворено в таблицю User з первинним ключем UserID та атрибутами Email, Name, Number.

Слабку сутність Rental перетворено в таблицю Rental з первинним ключем RentalID та зовнішніми ключами UserID та EquipmentID. Атрибути: Duration, RentalDate, ReturnDate.

Сутність Equipment перетворено на таблицю Equipment з первинним ключем EquipmentID та атрибутами Name, RentalCost, Description.

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

Изображение выглядит как текст, линия, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 — Схема бази даних

Таблиця 1 демонструє детальний перехід від однієї моделі до іншої

Таблиця 1 — опис об’єктів бази даних

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| User – містить інформацію про користувача | UserID – унікальний ідентифікатор користувача | integer |
| Name – ім’я користувача | character varying |
| Email – електронна пошта користувача | character varying |
| Number – номер  телефону користувача | integer |
| Rental – містить інформацію про оренду спорядження | RentalID – унікальний ідентифікатор оренди | integer |
| EquipmentID – ідентифікатор орендованого спорядження | integer |
| UserID – ідентифікатор користувача | integer |
| RentalDate – дата початку оренди спорядження | date |
| ReturnDate – дата, коли спорядження необхідно повернути | date |
|  | Duration – тривалість оренди | integer |
| Equipment – містить інформацію про спорядження | EquipmentID – унікальний ідентифікатор спорядження | Integer |
| Name – назва спорядження | character variable |
| RentalCost – вартість оренди спорядження | integer |
| Description – опис спорядження | character variable |

**Функціональні залежності для кожної таблиці**

1. User

UserID -> {Name, Email, Number}

1. Rental

RentalID, UserID (FK) -> {EquipmentID, UserID, RentalDate, ReturnDate, Duration}

1. Equipment -> {Name, RentalCost, Description}

**Відповідність схеми нормальним формам**

1. Щоб задовільнити умови 1НФ кожен атрибут в таблиці має бути атомарним, тобто:

* Кожна клітинка містить єдине значення;
* Кожен запис є унікальним.

Дана схема відповідає 1НФ.

1. Щоб схема відповідала 2НФ повинні виконуватись умови:

* Схема перебуває в 1НФ;
* Кожний неключовий атрибут функціонально залежить від цілого ключа.

У даній схемі кожна таблиця має власний унікальний ідентифікатор (ключ). Кожний атрибут у кожній таблиці залежить від цього унікального ідентифікатора. Тобто, схема також в НФ2.

1. Щоб схема відповідала 3НФ повинні виконуватись умови:

* Схема перебуває в 2НФ;
* Кожен неключовий атрибут нетранзитивно залежить від кожного ключа.

Оскільки дана схема в НФ2 та неключові атрибути не транзитивно залежать від інших неключових атрибутів, схема також в НФ3.

Висновок:

Схема бази даних відповідає нормальним формам НФ1, НФ2 та НФ3. Вона добре структурована і нормалізована, що сприяє ефективному та надійному зберіганню та обробці даних.

**Таблиці бази даних у pgAdmin4**

**UserID**

Изображение выглядит как текст, число, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

**Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание**

**Equipment**

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, линия, число

Автоматически созданное описание

**Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание**

**Rental**

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

**Изображение выглядит как текст, линия, Шрифт, число

Автоматически созданное описание**

**Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

**UserID**

-- Table: public.User

-- DROP TABLE IF EXISTS public."User";

CREATE TABLE IF NOT EXISTS public."User"

(

"UserID" integer NOT NULL,

"Name" character varying(50) COLLATE pg\_catalog."default" NOT NULL,

"Email" character varying(50) COLLATE pg\_catalog."default" NOT NULL,

"Number" integer NOT NULL,

CONSTRAINT "User\_pkey" PRIMARY KEY ("UserID")

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."User"

OWNER to postgres;

**Equipment**

-- Table: public.Equipment

-- DROP TABLE IF EXISTS public."Equipment";

CREATE TABLE IF NOT EXISTS public."Equipment"

(

"EquipmentID" integer NOT NULL,

"Name" character varying(50) COLLATE pg\_catalog."default" NOT NULL,

"RentalCost" integer NOT NULL,

"Description" character varying(500) COLLATE pg\_catalog."default",

CONSTRAINT "Equipment\_pkey" PRIMARY KEY ("EquipmentID")

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Equipment"

OWNER to postgres;

**Rental**

-- Table: public.Rental

-- DROP TABLE IF EXISTS public."Rental";

CREATE TABLE IF NOT EXISTS public."Rental"

(

"RentalID" integer NOT NULL,

"EquipmenID" integer NOT NULL,

"UserID" integer NOT NULL,

"RentalDate" time with time zone NOT NULL,

"ReturnDate" time with time zone NOT NULL,

"Duration" integer NOT NULL,

CONSTRAINT "Rental\_pkey" PRIMARY KEY ("RentalID"),

CONSTRAINT "EquipmentID" FOREIGN KEY ("EquipmenID")

REFERENCES public."Equipment" ("EquipmentID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "Rental\_UserID\_fkey" FOREIGN KEY ("UserID")

REFERENCES public."User" ("UserID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Rental"

OWNER to postgres;