4.1.1 Особливості інсталяції та роботи з СУБД

CУБД - PostgreSQL — широко розповсюджена система керування базами даних з відкритим сирцевим кодом. Прототип був розроблений в Каліфорнійському університеті Берклі в 1987 році під назвою POSTGRES, після чого активно розвивався і доповнювався. В червні 1990 року з'явилась друга версія із переробленою системою правил маніпулювання та роботи з таблицями, у 1991 році — третя версія, із доданою підтримкою одночасної роботи кількох менеджерів збереження, покращеним механізмом запитів і доповненою системою внутрішніх правил.

Переваги PostgreSQL:

1. підтримка БД необмеженого розміру;
2. потужні і надійні механізми транзакцій і реплікації;
3. розширювана система вбудованих мов програмування і підтримка завантаження C-сумісних модулів;
4. спадкування;
5. легка розширюваність.
   * 1. Особливості створення структур даних

Нижче представлений код створення таблиць даних PostgreSQL:

CREATE TABLE hobby

(

id\_hobby SERIAL NOT NULL PRIMARY KEY

title character varying(50) NOT NULL,

);

CREATE TABLE thema

(

id\_thema NOT NULL SERIAL PRIMARY KEY,

thema\_title character varying(50) NOT NULL,

);

CREATE TABLE account\_status

( id\_account\_status NOT NULL SERIAL PRIMARY KEY,

account\_status\_title character varying(50) NOT NULL,

);

CREATE TABLE user

(

id\_user NOT NULL SERIAL PRIMARY KEY,

first\_name character varying(30) NOT NULL,

second\_name character varying(50) NOT NULL,

login character varying(50) NOT NULL,

date\_of\_birth date NOT NULL CHECK (date\_of\_birth > (current\_date – integer’5840’)),

e\_mail character varying(100) NOT NULL,

country character varying(50) NOT NULL,

city character varying(50) NOT NULL,

login character varying(30) NOT NULL,

password character varying(50) NOT NULL,

rating bigint DEFAULT 0 CHECK (rating < 101 AND rating > -101),

account\_status integer REFERENCES public.account\_status((id\_account\_status))

);

CREATE TABLE user\_contacts (

user\_id integer INTEGER NOT NULL REFERENCES public.user(id\_user),

receiver\_id integer INTEGER NOT NULL REFERENCES public.user(id\_user),

PRIMARY KEY (user\_id, receiver\_id)

);

CREATE TABLE messages (

id\_message SERIAL NOT NULL PRIMARY KEY,

from\_user INTEGER NOT NULL REFERENCES public.user(id\_user),

to\_user INTEGER NOT NULL REFERENCES public.user(id\_user),

datasent TIMESTAMP NOT NULL,

dataread TIMESTAMP NOT NULL,

mes\_content TEXT NOT NULL,

attached\_files INTEGER

);

CREATE TABLE message\_exchange (

id\_exchange SERIAL NOT NULL PRIMARY KEY,

message\_id INTEGER REFERENCES public.messages(id\_message),

to\_user INTEGER REFERENCES public.user(id\_user)

);

CREATE TABLE events(

id\_event SERIAL NOT NULL PRIMARY KEY,

title VARCHAR(100) NOT NULL,

place VARCHAR(150) NOT NULL,

beginning\_time TIMESTAMP NOT NULL,

discription TEXT NOT NULL

);

CREATE TABLE event\_themas(

event\_id INTEGER NOT NULL REFERENCES public.events(id\_event),

thema\_id INTEGER NOT NULL REFERENCES public.thema(id\_thema),

PRIMARY KEY (event\_id, thema\_id)

);

Дане середовище для управління реляційної бази даних допомагає услідкувати за багатьма недоліками, що можуть бути в результаті некоректної роботи компонентів програми.