Universitatea Tehnică a Moldovei

Facultatea - Calculatoare Informatică și Microelectronică

**Raport**

**la Baze de Date**

Lucrarea de laborator Nr.1

Tema: Crearea și întreținerea bazei de date

A efectuat: st.gr. SI-211 Chihai Adrian

A verificat: Rodica Bulai

Chișinău 2023

# 1 Sarcina lucrării

Instalarea crearea și întreținerea bazei de date

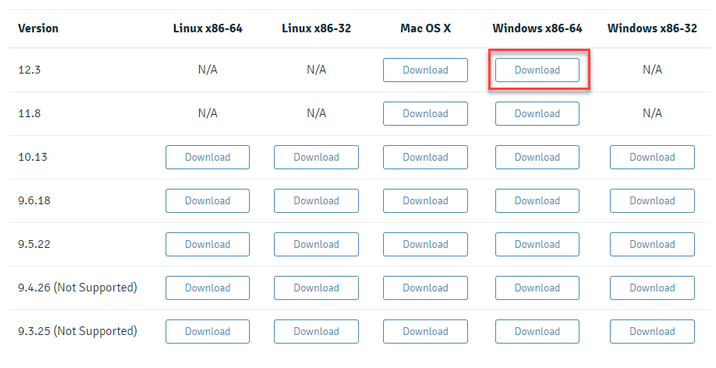
2 Obiective

Instalarea și configurarea unui sistem de gestionare a bazelor de date (de exemplu, PostgreSQL, Microsoft SQL Server) pe serverul desemnat. Crearea unei baze de date cu un nume și o structură adecvată (tabele, coloane, constrângeri etc.). Asigurarea disponibilității continue a bazei de date și a serviciului asociat. Realizarea backup-urilor periodice pentru a preveni pierderea datelor în caz de eșec sau incidente.

3 Desfășurarea lucrării de laborator

**3.1 Instalarea**

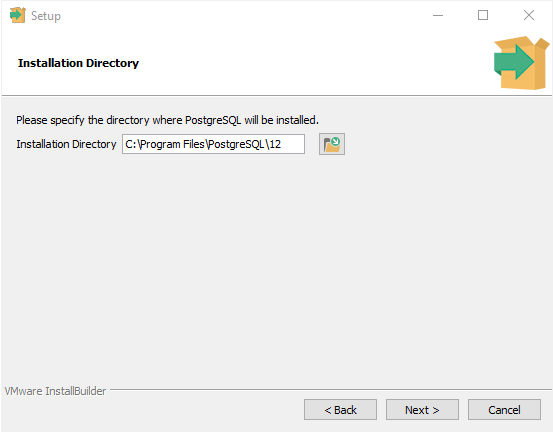
Descarcam instalatorul de postgres pentru windows

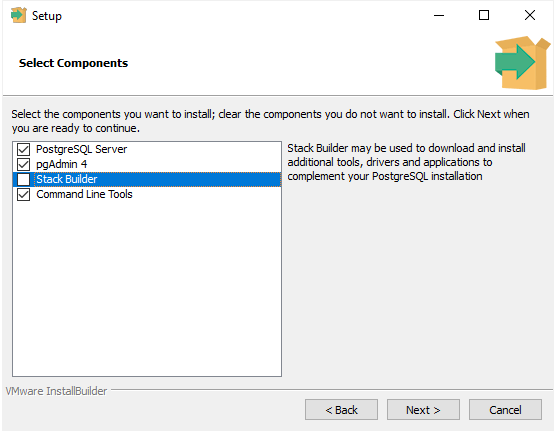


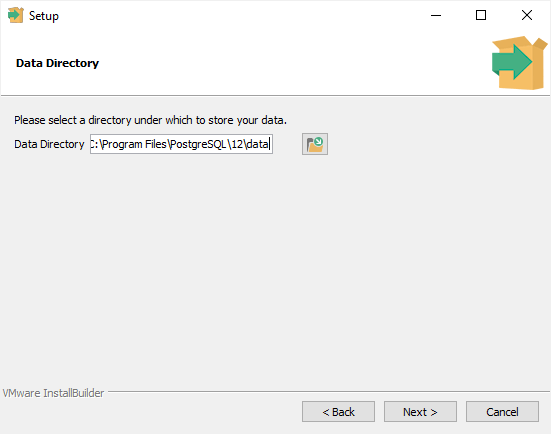
Instalam Postgresql pe windows pas dupa pas

Deschidem instalatia

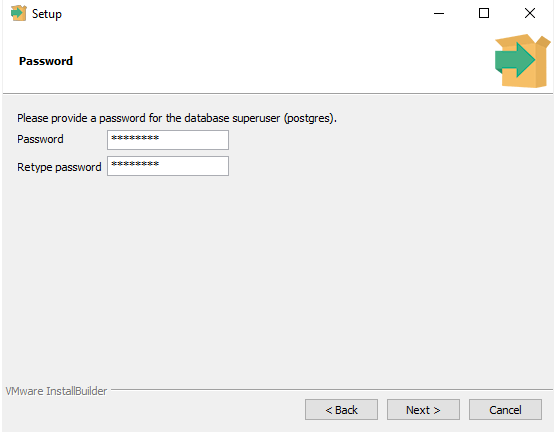


Specificam unde dorim sa instalam   


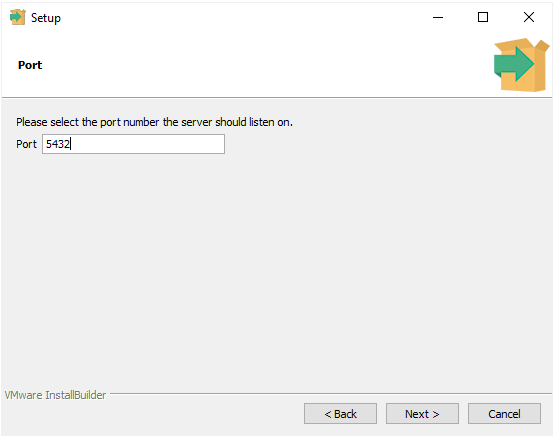
Alegem componentele necesare   


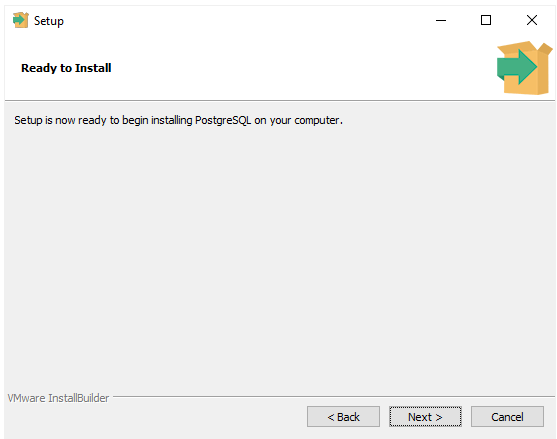
Alegem folderul de stocare a datelor   


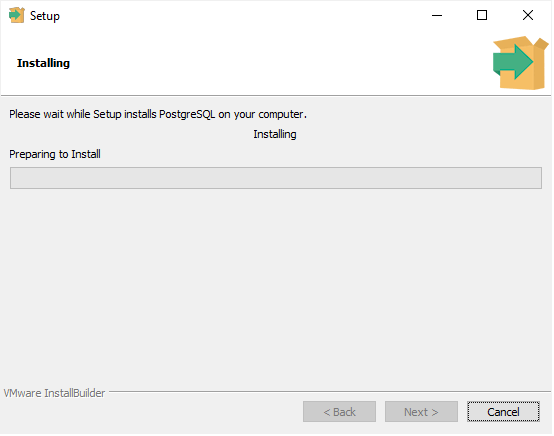
Punem parola pentru userul postgres care se socoate root user



Specificam portul (portul implicit pentru postgres este 5432)



Configurarile pentur instalare au fost introduce acum putem apasa butonul next pentru instalare  


Instalarea  


Finisarea instalarii



Crearea nbazei de date university

-- Create the database

CREATE DATABASE your\_database\_name;

CREATE TABLE discipline (

id\_disciplina SERIAL PRIMARY KEY,

disciplina VARCHAR(100),

nr\_ore\_plan\_disciplina INT

);

CREATE TABLE grupe (

id\_grupa SERIAL PRIMARY KEY,

cod\_grupa VARCHAR(10),

specialitatea VARCHAR(100),

facultatea VARCHAR(100)

);

CREATE TABLE studenti (

id\_student SERIAL PRIMARY KEY,

nume\_student VARCHAR(50),

prenume\_student VARCHAR(50),

data\_nasterii\_student DATE,

adresa\_postala\_student VARCHAR(200)

);

CREATE TABLE profesori (

id\_profesor SERIAL PRIMARY KEY,

nume\_profesor VARCHAR(50),

prenume\_profesor VARCHAR(50),

adresa\_postala\_profesor VARCHAR(200)

);

CREATE TABLE studenti\_reusita (

id\_student INT,

id\_disciplina INT,

id\_profesor INT,

id\_grupa INT,

tip\_evaluare VARCHAR(50),

nota INT,

data\_evaluare DATE,

FOREIGN KEY (id\_student) REFERENCES studenti(id\_student),

FOREIGN KEY (id\_disciplina) REFERENCES discipline(id\_disciplina),

FOREIGN KEY (id\_profesor) REFERENCES profesori(id\_profesor),

FOREIGN KEY (id\_grupa) REFERENCES grupe(id\_grupa)

);

Crearea grupei cu roluri si a unui user

Am create group\_standart care are acces la selectarea inserarea si updatarea datelor din baza de date university si crearea unui user care apartine acestei grupe



Prin toatel privilegiile se numera

1. **SELECT**
2. **INSERT**
3. **UPDATE**
4. **DELETE**
5. **CREATE**
6. **DROP**
7. **USAGE**
8. **GRANT OPTION**
9. **ALL PRIVILEGES**
10. **EXECUTE**
11. **CONNECT**
12. **TEMPORARY**
13. **TRIGGER**
14. **REFERENCES**
15. **CREATE TEMP TABLESPACE**
16. **USAGE ON SCHEMA**
17. **SEQUENCE**

Efectuarea întreținerii în PostgreSQL este importantă pentru a asigura sănătatea și performanța bazei de date. Mai jos sunt prezentate câteva comenzi și sarcini de întreținere folosite frecvent:

VACUUM:

VACUUM este folosit pentru a recupera spațiul de pe disc ocupat de tuplele moarte (înregistrările șterse sau actualizate).

Formă de bază: VACUUM [FULL] [ANALYZE] [VERBOSE] nume\_tabel;

Exemplu: VACUUM FULL ANALYZE my\_table;

ANALYZE:

ANALYZE colectează statistici despre distribuția datelor în tabele și indexuri. Acest lucru ajută optimizerul de interogare să ia decizii mai bune.

Formă de bază: ANALYZE [VERBOSE] [nume\_tabel];

Exemplu: ANALYZE my\_table;

REINDEX:

REINDEX reconstruiește unul sau mai multe indexuri.

Formă de bază: REINDEX [VERBOSE] [nume\_tabel];

Exemplu: REINDEX my\_index;

CLUSTER:

CLUSTER reorganizează fizic datele unei tabele bazate pe indexul specificat. Acest lucru poate îmbunătăți performanța interogărilor.

Formă de bază: CLUSTER nume\_tabel USING nume\_index;

Exemplu: CLUSTER my\_table USING my\_index;

VACUUM FULL:

VACUUM FULL rescrie întreaga tabelă și este o formă mai agresivă a VACUUM. Necesită un blocaj exclusiv pe tabel.

Formă de bază: VACUUM FULL [VERBOSE] nume\_tabel;

Exemplu: VACUUM FULL my\_table;

pg\_dump:

pg\_dump este o utilitate în linie de comandă pentru realizarea unui backup al unei baze de date PostgreSQL.

Exemplu: pg\_dump -U username -d database\_name -f backup\_file.sql

pg\_restore:

pg\_restore este folosit pentru a restaura o bază de date PostgreSQL dintr-un arhivă creată de pg\_dump.

Exemplu: pg\_restore -U username -d database\_name backup\_file.sql

CHECKPOINT:

Declanșează manual un checkpoint. Un checkpoint curăță toate paginile murdare din cache-ul de memorie către disc.

Exemplu: CHECKPOINT;

RESET:

Resetează unul sau mai mulți parametri de configurare la valorile lor implicite.

Exemplu: RESET nume\_parametru;

SET:

Schimbă valoarea unui parametru de configurare pentru sesiunea curentă.

Exemplu: SET nume\_parametru = valoare;