

Introducció a l'Enginyeria del Programari

Entorn desenvolupament per la pràctica

Curs 2023/24 Q1

El sistema a desenvolupar és una aplicació de consola que s'implementa fent servir el llenguatge de programació d'alt nivell orientat a objectes C++. Les dades es guarden fent servir el gestor de bases de dades relacional PostgreSQL. Per la connexió de l'aplicació amb el PostgreSQL es fa servir la llibreria libpqxx. En aquest document s'inclouen les instruccions per poder configurar les eines que necessiteu per aquest desenvolupament.

Una vegada tingueu configurat el vostre entorn de desenvolupament, podeu començar desenvolupant el programa principal (main). Al ser una aplicació de consola, farem servir menús textuais, on les diferents opcions tenen un número associat que serà el que l'usuari ha d'entrar per seleccionar la opció que vulgui seleccionar. L'usuari entrarà la informació de manera textual (no hi ha controls per seleccionar) i totes les dades resultants es presentaran de manera seqüencial.

Aquesta aplicació té dos menús principals depenent de si tenim un usuari loggejat o no. En el cas de no tenir cap usuari loggejat el menú conté les opcions:

1. Iniciar sessió
2. Registrar usuari
3. Sortir

Quan el sistema té un usuari que ja ha iniciat, el menú principal conté una opció per cada paquet de casos d'ús que tenim a l'especificació. Aquesta opció portarà a un menú que inclou les opcions que iniciaran cadascun dels casos d'ús corresponents al paquet. A continuació s'inclouen totes les opcions de menú i submenús que donen inici a cadascun dels casos d'ús:

- | | |
|----------------------------|--------------------------------|
| 1. Gestió usuari | 3. Consulta compres |
| 1. Registra usuari | 3. Consultes videojocs |
| 2. Consulta usuari | 1. Consulta videojoc |
| 3. Modifica usuari | 2. Consulta videojocs |
| 4. Esborra usuari | 3. Consulta videojocs per edat |
| 5. Tornar | 4. Consulta novetats |
| 2. Gestió compres | 5. Tornar |
| 1. Compra videojoc | 4. Sortir |
| 2. Compra paquet videojocs | |

També podeu començar a preparar el controlador façana per la capa de presentació (classe CapaPresentacio), on s'ha d'incloure una operació que controla cadascun dels casos d'ús als que dona accés cadascuna de les opcions del menú. Per aquesta primera implementació, podeu fer que cada operació de la capa de dades mostri el missatge "Funcionalitat <nom de la funcionalitat> en desenvolupament", que desenvolupareu seguint el disseny que publicarem.

Entorn de desenvolupament

Per desenvolupar el sistema es pot fer servir l'entorn de desenvolupament desitjat, en aquestes instruccions es fa servir Visual Studio.

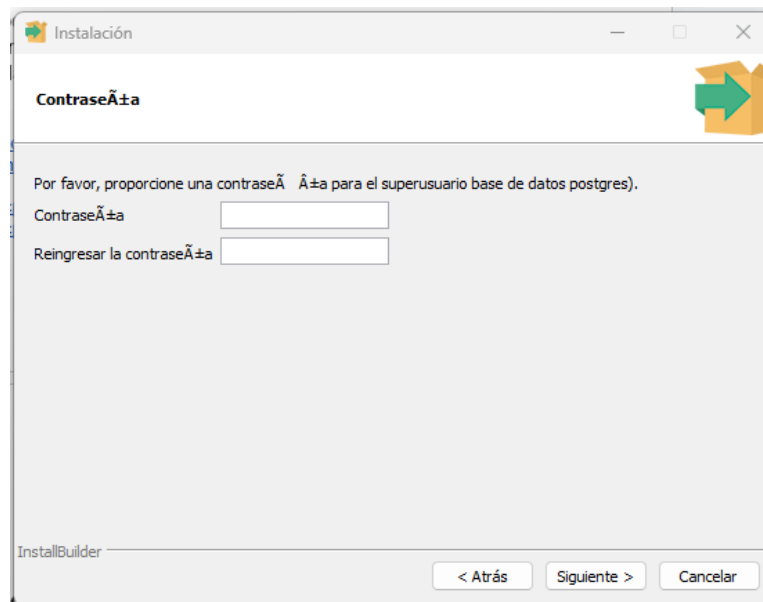
Necessitem instal·lar el Visual Studio 2022 edició Community¹ (la versió actual és la 2022). A la pàgina de la descàrrega podeu trobar informació explicant com fer servir aquest entorn de programació amb C/C++². En el nostre cas, necessitem l'extensió per desenvolupar sistemes amb C/C++.

Sistema de gestió de Base de Dades

Per aquesta pràctica farem servir postgresQL. En aquesta secció teniu les instruccions per instal·lar-ho i fer servir l'aplicació client per accedir a les dades.

Instal·lació PostgreSQL

S'ha de descarregar i instal·lar PostgreSQL³, al fer la instal·lació us demana una contrasenya que necessitem per després poder accedir al sistema.

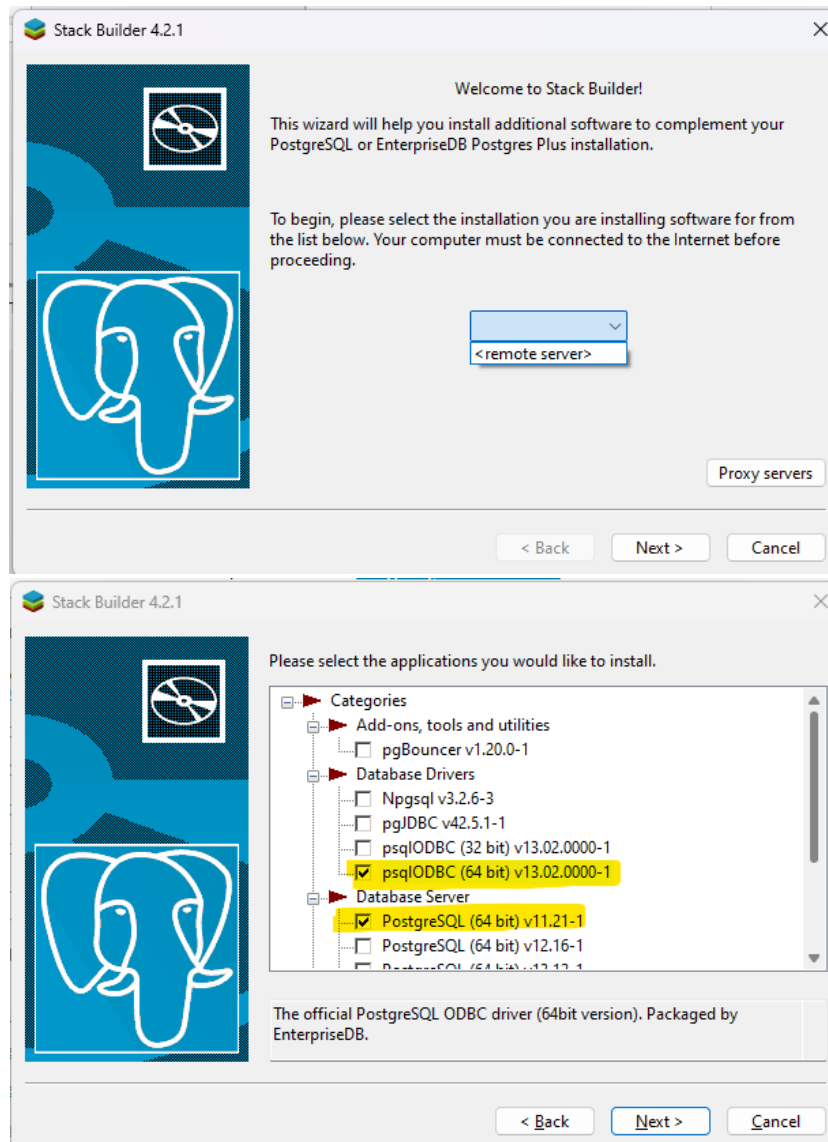


Si feu servir el paquet recomanat, al finalitzar la instal·lació del PostgreSQL, us suggereix instal·lar paquets addicionals, s'ha d'instal·lar al menys el servidor (Database server/PostgreSQL) i el driver per poder accedir (Database Drivers/psqlODBC).

¹ <https://visualstudio.microsoft.com/es/free-developer-offers/>

² <https://learn.microsoft.com/en-us/cpp/overview/visual-cpp-in-visual-studio?view=msvc-170#install-visual-studio-c-and-upgrade-from-earlier-versions>

³ <https://www.postgresql.org/>

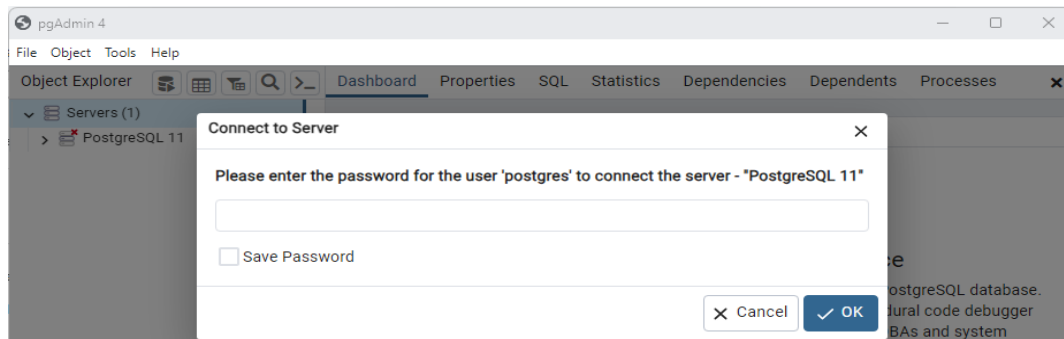


Una vegada acabada la instal·lació, podeu accedir a gestionar les vostres bases de dades fent servir l'aplicació pgAdmin4. Durant la instal·lació del pgAdmin4 us demana una contrasenya que es la que servirà per accedir a les vostres bases de dades (es crea el superusuari "postgres"), aquesta contrasenya és diferent a la contrasenya que li posem a cada una de les bases de dades.

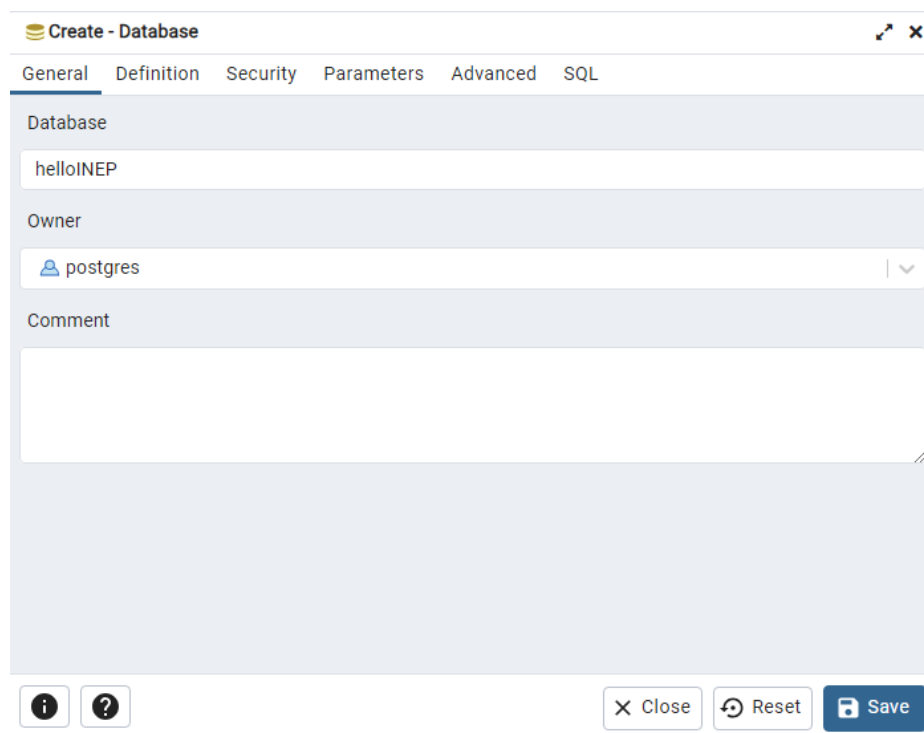
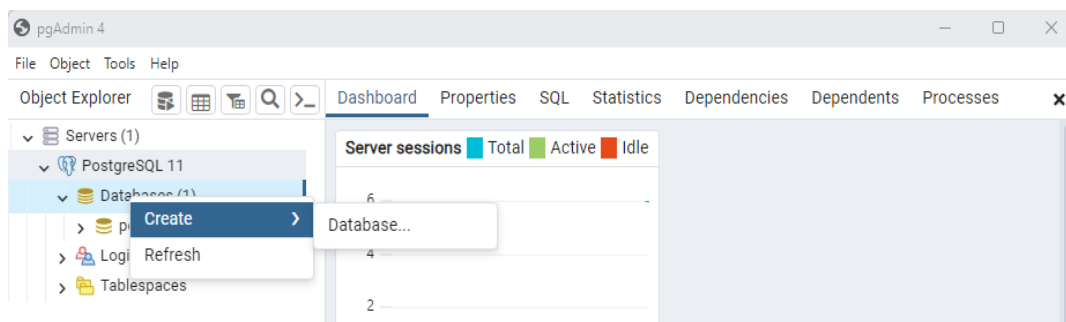
Quan acaba de fer aquesta segona instal·lació demana si es vol continuar instal·lant paquets addicionals, no cal.

Accés al servidor de bases de dades des de l'aplicació client

Per accedir al servidor, heu de fer servir l'aplicació client pgAdmin4, que s'ha instal·lat a l'instal·lar el PostgreSQL. Quan entreu en aquesta aplicació, a l'esquerra teniu la llista de servidors (només 1) i us demanarà la contrasenya del superusuari que heu definit durant el procés d'instal·lació.

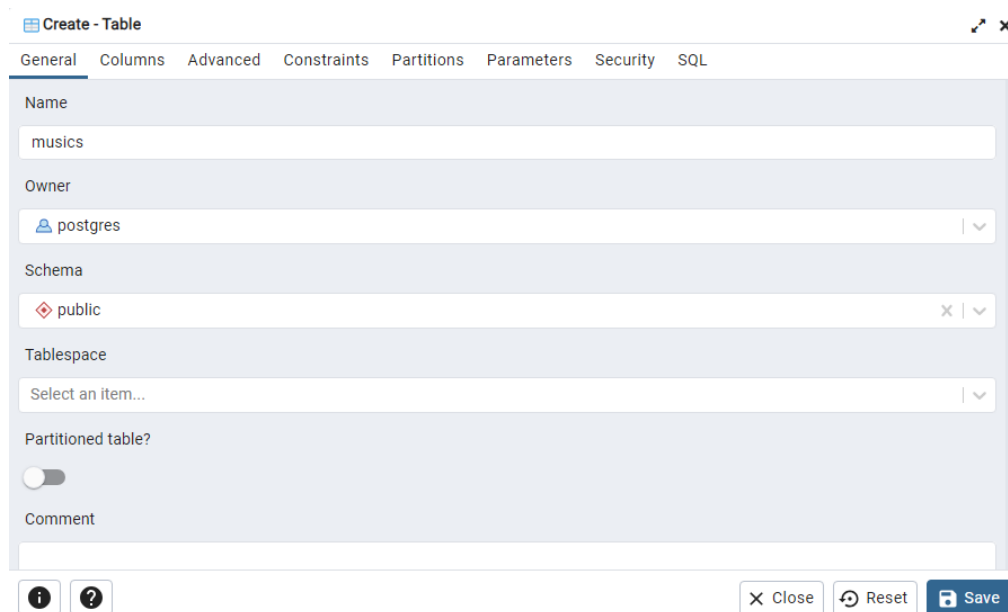
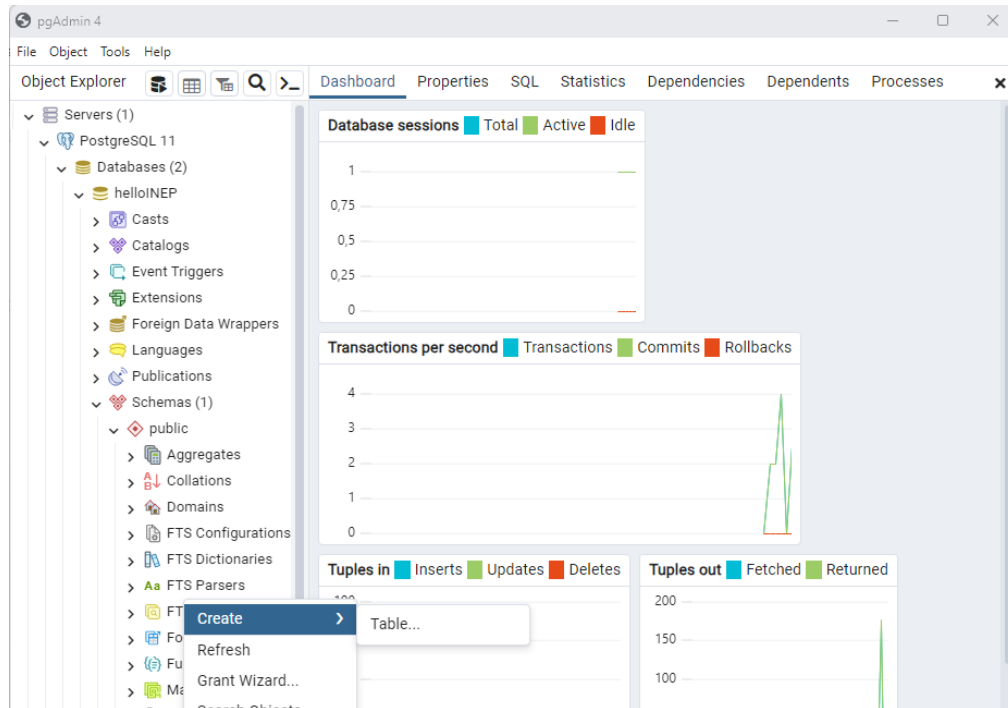


Una vegada dins, creareu la vostra primera base de dades amb el nom de "helloINEP".

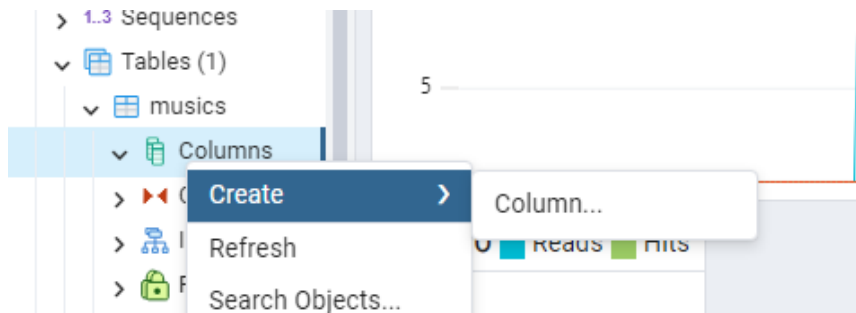


Crear i accedir al contingut d'una taula

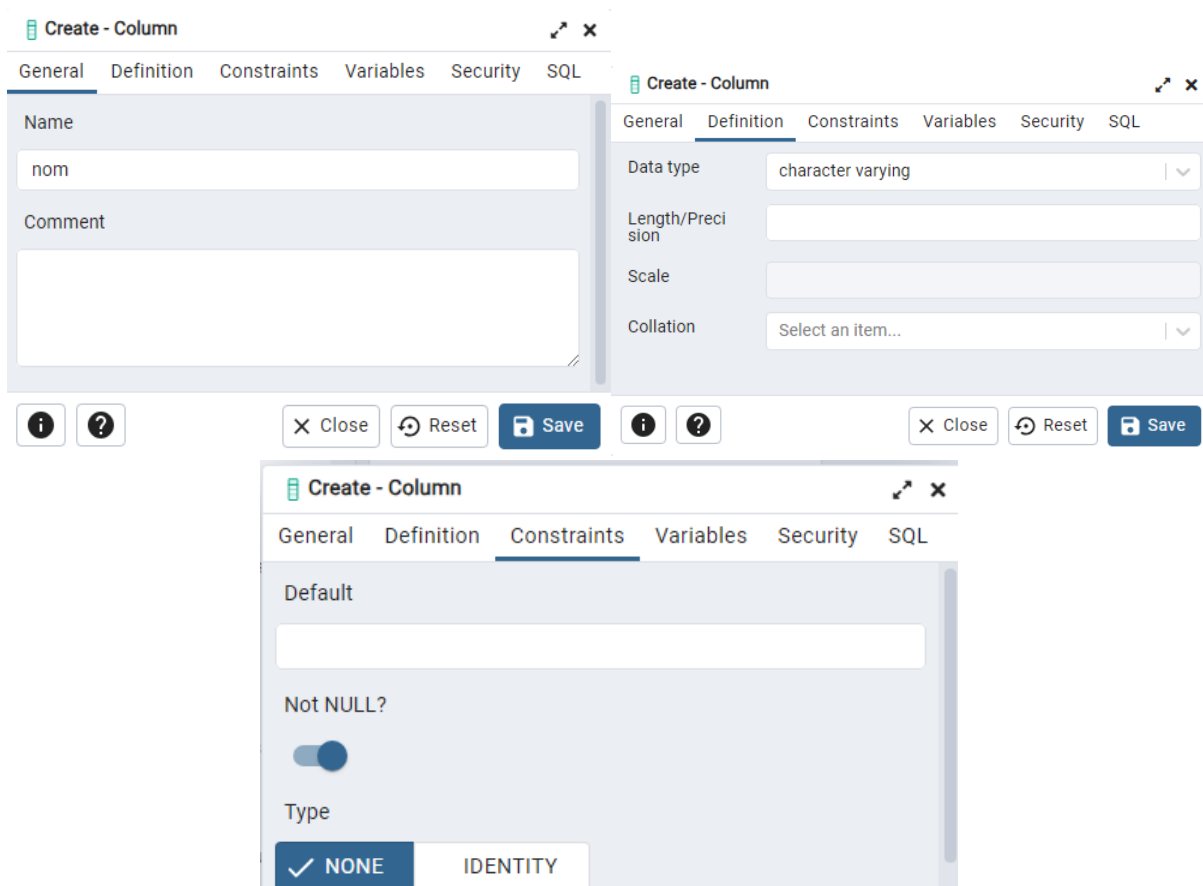
Des de l'aplicació pgAdmin4 es poden crear les taules d'una base de dades. Crearem la taula "musics" dins la base de dades "helloINEP".



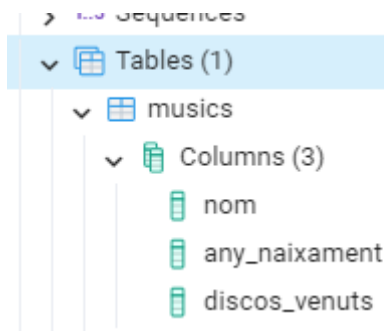
Una vegada creada la taula, li afegirem les columnes “nom”, “any_naixement” i “discos_venuts”.



Per crear la columna, s'ha d'omplir dades a les seccions “General” (Name: “nom”), Definition (Data type: “character varying”) i Constrains (Not null: activat només per la columna nom). Per les columnes any_naixement i discos_venuts, escollir el tipus de dades “bigint” i no activar el not null.

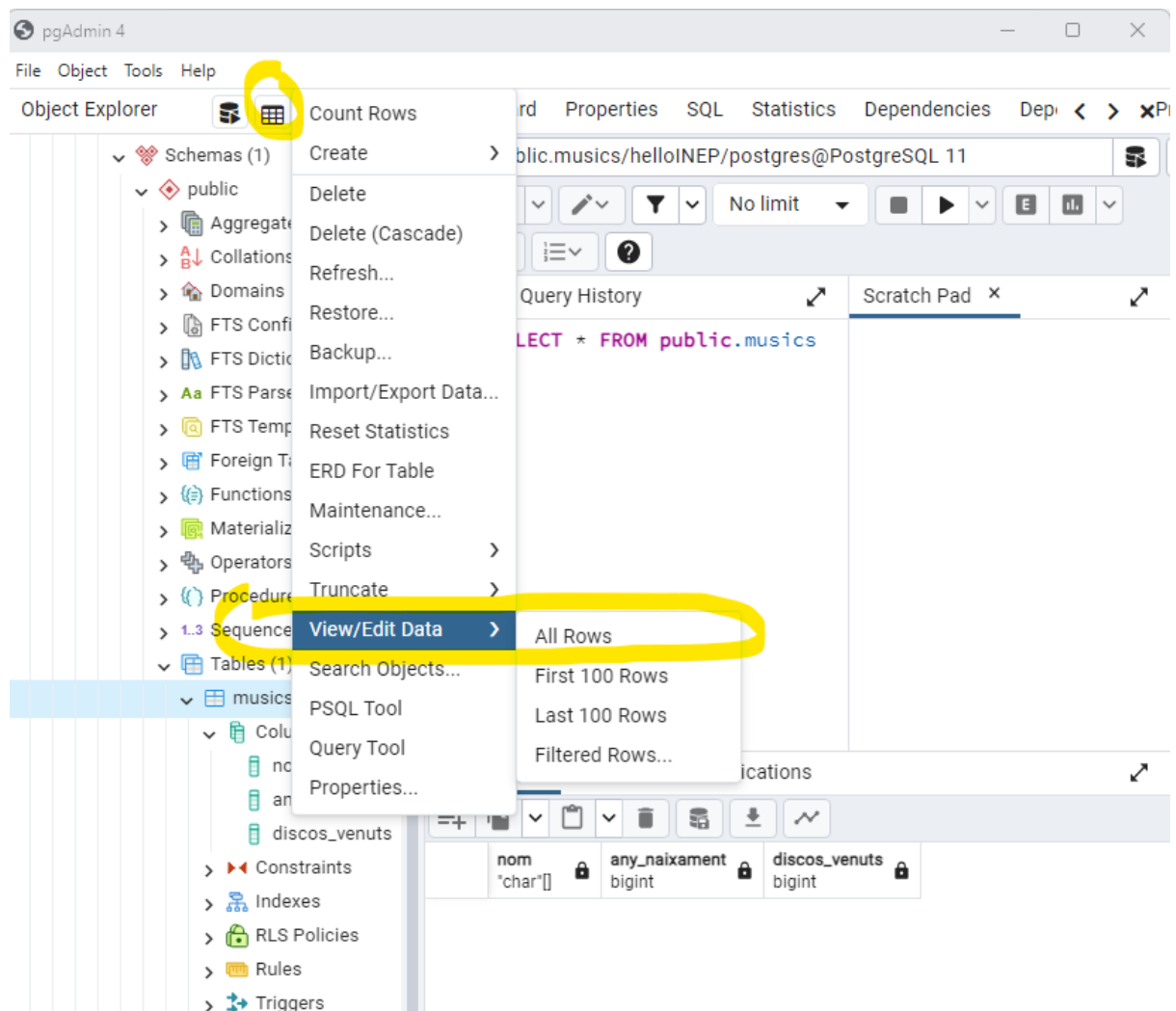


La definició de la taula s'ha de veure com a la figura següent.

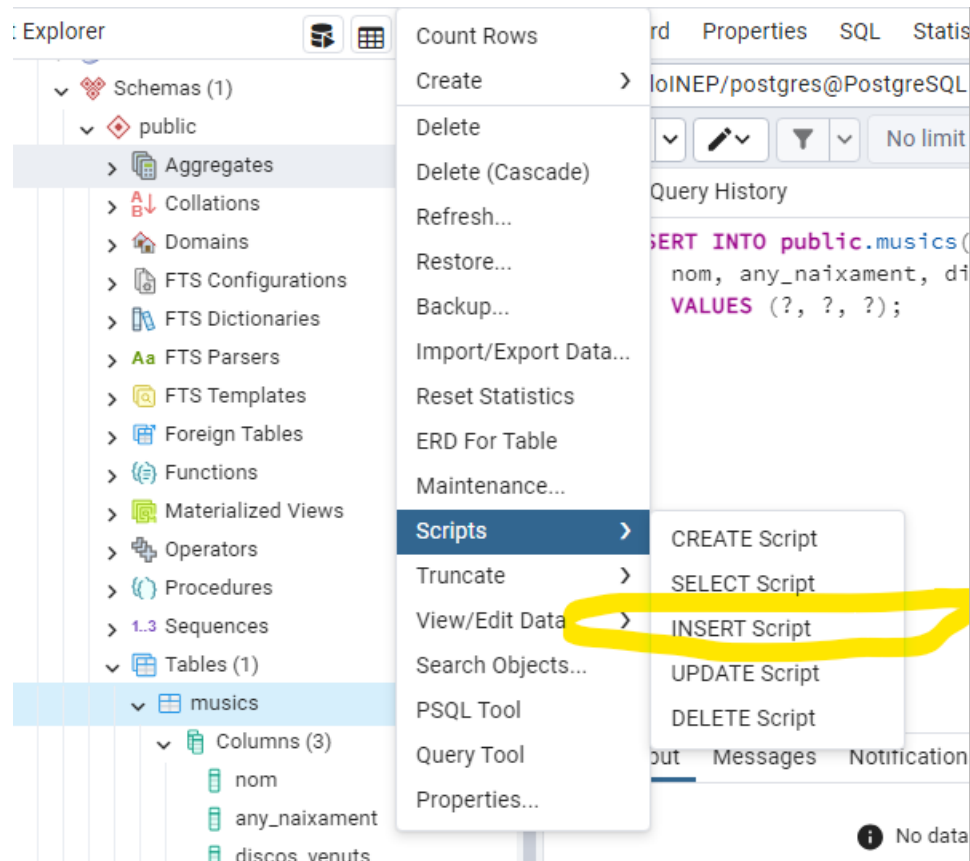


Inserir i consultar dades a una taula de la base de dades

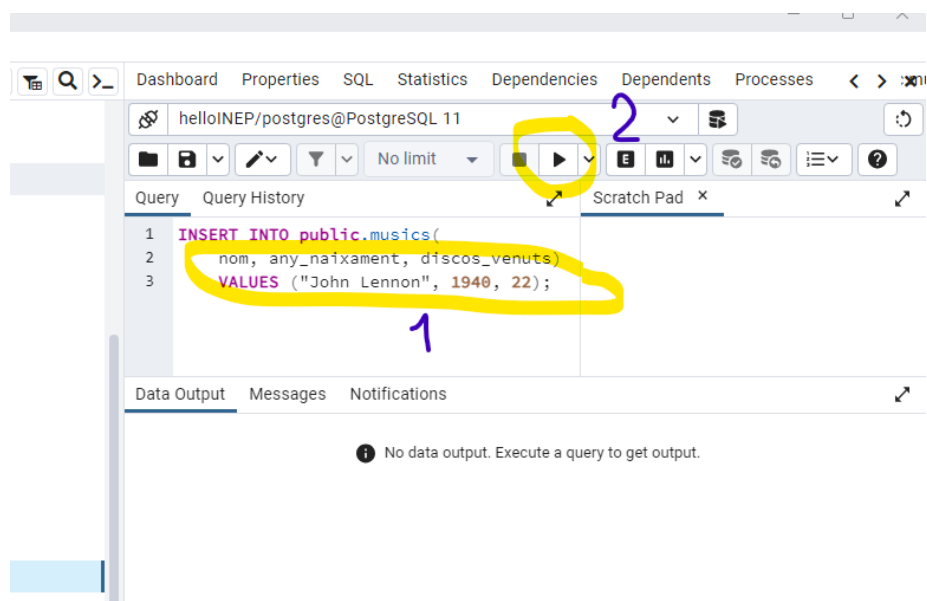
Per consultar les files d'una taula, s'ha d'accedir al menú "View/Edit Data" sobre el nom de la taula i seleccionar l'opció corresponent (e.g. "All Rows"). A la dreta es mostra la consulta que es fa amb el llenguatge SQL (a la part superior de la pantalla) i el resultat de la consulta a la part inferior. A la següent figura mostra el que es veu quan la taula està buida.



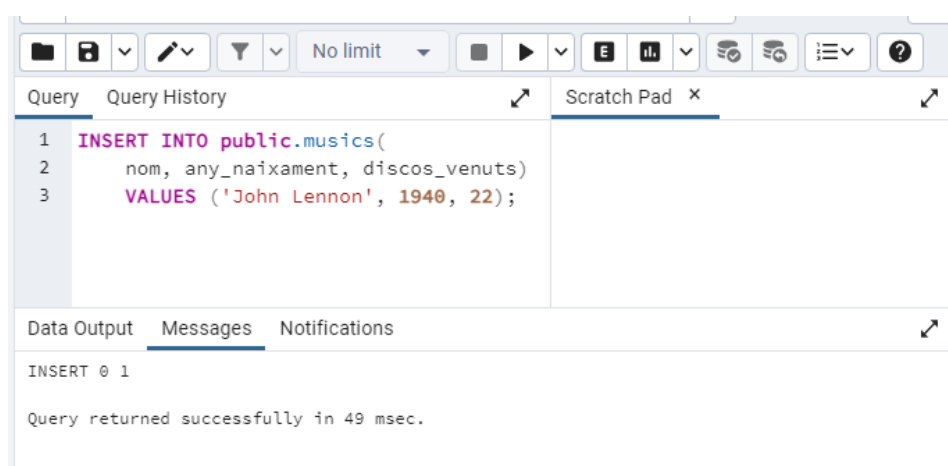
Per inserir files, s'ha d'accedir a l' menú *Scripts* i escollir la opció *INSERT* a la taula *musics*.



Afegirem la informació (1) de Jonh Lennon (any de naixament 1940 i 22 - millions - a discos venuts) i (2) clicarem el botó per executar la instrucció.



Amb el resultat "Query returned successfully" a la pantalla inferior com es mostra a la següent figura.



Si consultem la taula després d'inserir John Lennon i Rosalia, hauria de sortir el que mostra la següent figura. Per consultar la taula escollir la opció Scripts/SELECT script al menú sobre el nom de la taula.

The screenshot shows the same database IDE interface. The 'Query' pane now contains the following SQL statement:

```
1 SELECT * FROM public.musics  
2
```

The 'Data Output' pane displays the results of the query in a table format. The table has four columns: 'nom', 'any_naixament', and 'discos_venuts'. The first column is labeled 'nom' with a data type of 'character varying'. The second column is labeled 'any_naixament' with a data type of 'bigint'. The third column is labeled 'discos_venuts' with a data type of 'bigint'. The table contains two rows of data:

	nom character varying	any_naixament bigint	discos_venuts bigint
1	John Lennon	1940	22
2	Rosalia	1992	3

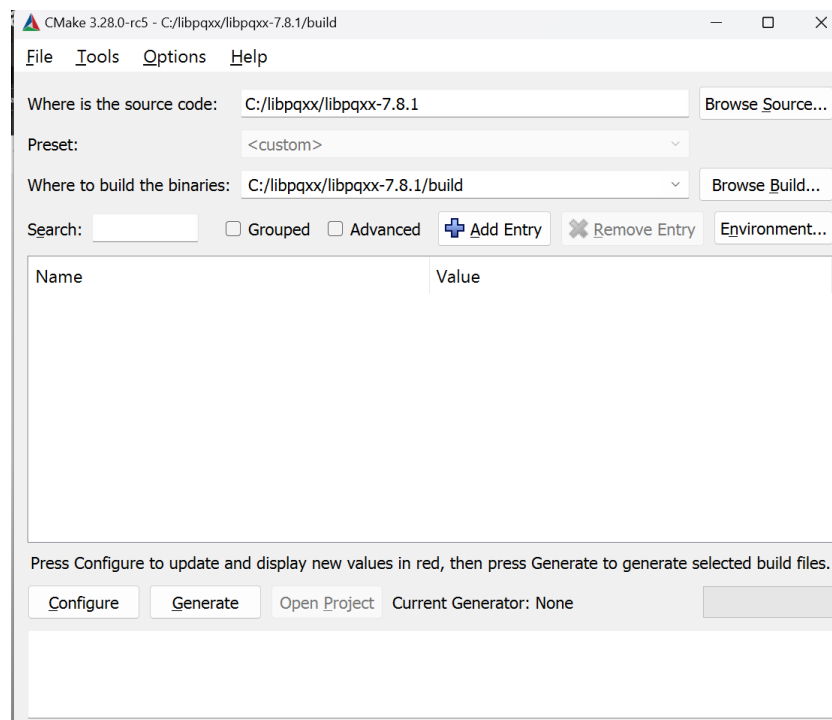
Accés al gestor de bases de dades des de l'aplicació

Per connectar-nos al gestor de bases de dades, des de la nostra aplicació, farem servir la llibreria libpqxx⁴ que s'ha d'instal·lar abans de poder-la fer servir en el nostre sistema.

Instal·lar llibreria libpqxx

Per poder fer servir la llibreria libpqxx en un entorn Windows s'ha de compilar i instal·lar. El codi font de la llibreria s'ha de baixar del GitHub⁵ i s'ha de compilar. Per compilar la llibreria podeu fer servir el CMake⁶ (recomanació: incloure el CMake al path del sistema, durant la instal·lació es dóna aquesta opció) per generar els arxius necessaris per la compilació i després fer servir el Visual Studio 2022 per compilar-la i instal·lar-la.

Per generar els arxius per després compilar, obrim l'aplicació CMake i omplim la informació que ens demana referent a la localització del codi font i on volem que es generin els arxius.



Ens demana si volem crear el directori que hem indicat pel <build> si no el tenim.

⁴ <https://pqxx.org/libpqxx/>

⁵ <https://github.com/jtv/libpqxx>

⁶ <https://cmake.org/download/>

Specify the generator for this project

Visual Studio 17 2022

Optional platform for generator(if empty, generator uses: x64)

x64

Optional toolset to use (argument to -T)

☒ Use default native compilers
☐ Specify native compilers
☐ Specify toolchain file for cross-compiling
☐ Specify options for cross-compiling

Finish Cancel

També es poden configurar algunes opcions, no cal que genereu el BUILD_DOC ni el BUILD_TEST, referent a les configuracions amb Release en teniu prou.

CMake 3.28.0-rc5 - C:/libpqxx/libpqxx-7.8.1/build

File Tools Options Help

Where is the source code: C:/libpqxx/libpqxx-7.8.1 Browse Source...

Preset: <custom>

Where to build the binaries: C:/libpqxx/libpqxx-7.8.1/build Browse Build...

Search: ☐ Grouped ☐ Advanced

Name	Value
BUILD_DOC	<input type="checkbox"/>
BUILD_TEST	<input type="checkbox"/>
CMAKE_CONFIGURATION_TYPES	Debug;Release
CMAKE_INSTALL_PREFIX	C:/Program Files/libpqxx

Press Configure to update and display new values in red, then press Generate to generate selected build files.

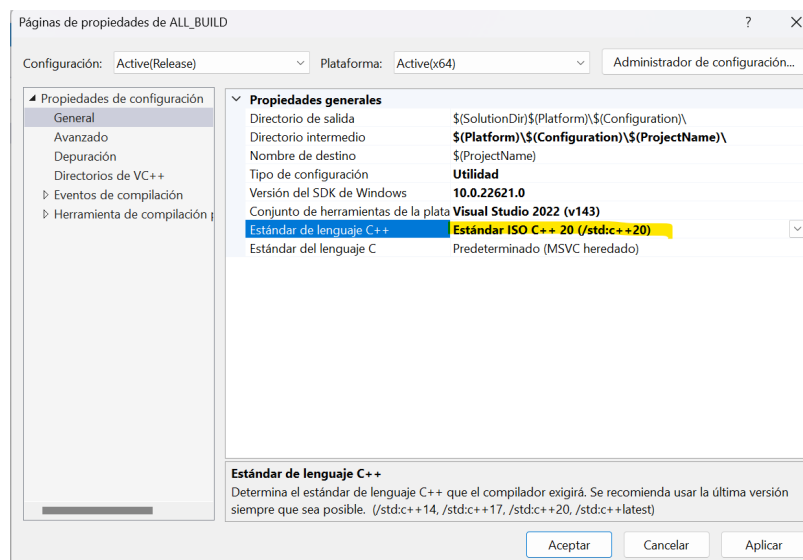
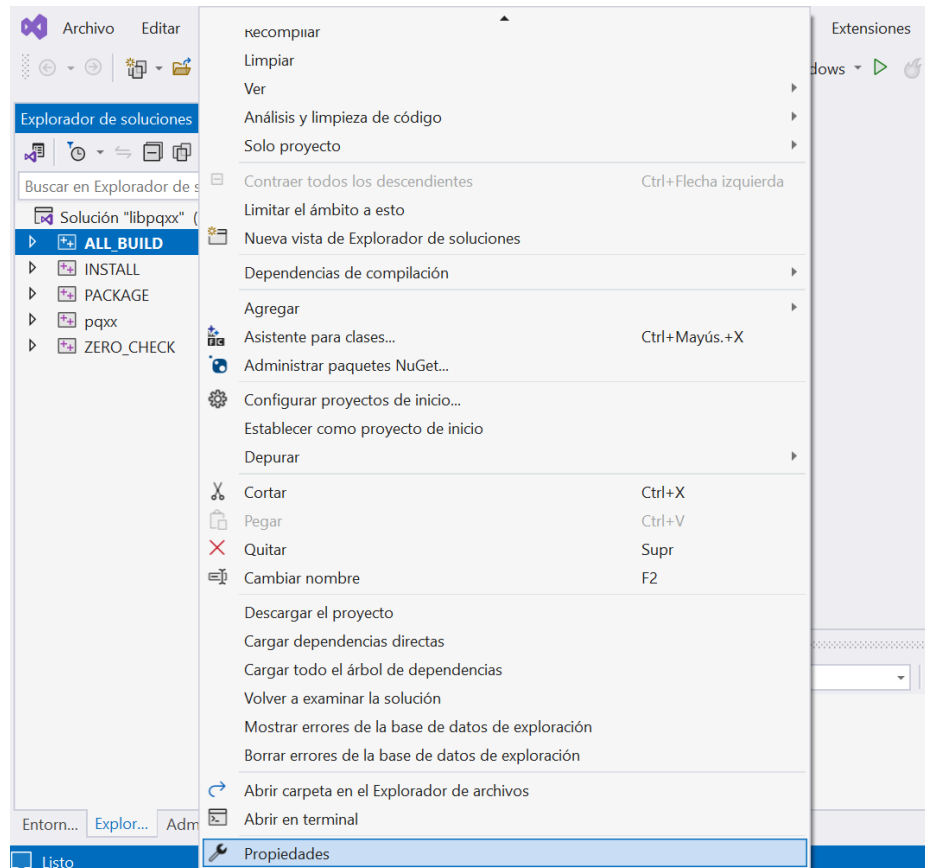
Current Generator: Visual Studio 17 2022

Generating config.h
 Generating config.h - done
 Configuring done (38.1s)

Una vegada seleccioneu la configuració, feu servir el botó <Generate>.

Amb aquestes accions es genera el codi que hem d'obrir amb el Visual Studio 2022 per compilar i instal·lar, obriu l'arxiu .sln que s'ha generat al directori corresponent (al directori indicat al camp "Where to build the binaries"). Per compilar s'ha d'executar el Visual Studio com a administrador (**IMPORTANT**: si no s'executa el Visual Studio com administrador no funciona).

Abans de passar a compilar el projecte, heu de confirmar que estigui ben configurat la versió que es fa servir de l'estàndard de C++ a les propietats del projecte (opció Propietats en el menú sobre els paquets <ALL_BUILDS> i <pqxx> a l'explorador de solucions) per evitar errors de compilació.



Una vegada estigui tot ben configurat, es compila (opció <Compilar> sobre ALL_BUILD).

```

Mostrar salida de: Compilación
3>----- Operación compilar iniciada: Proyecto: ALL_BUILD, Configuración: Release x64 ----
3>Building Custom Rule C:/libpqxx/libpqxx-7.8.1/CMakeLists.txt
===== Compilación: 3 correcto, 0 erróneo, 0 actualizado, 0 omitido =====
===== Compilar completado a las 18:15 y tardó 35,441 segundos =====

```

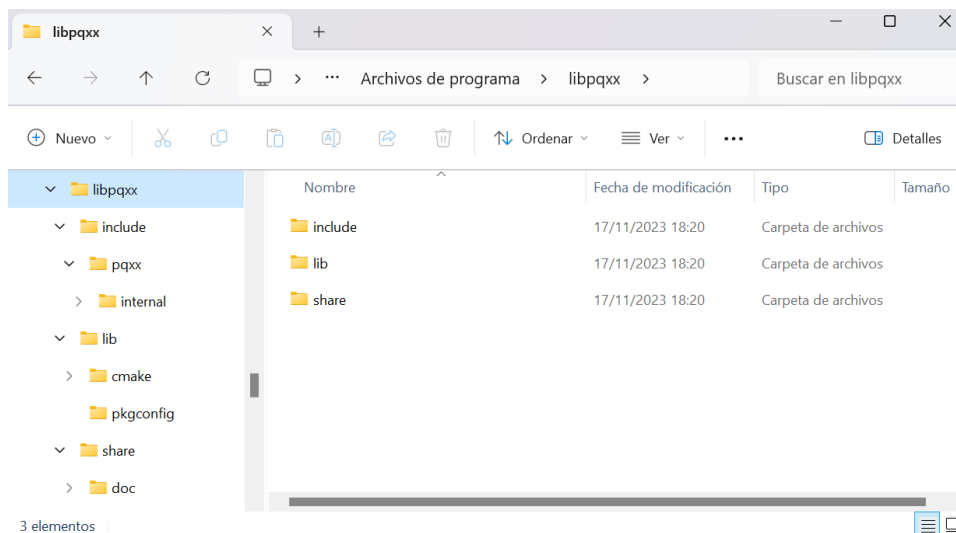
Una vegada compilat (ALL_BUILD), l'hem d'instal·lar, fent servir l'opció <Compilar> sobre el paquet <INSTALL> a l'explorador de solucions.

```

Mostrar salida de: Compilación
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/datatypes.md
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/escaping.md
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/getting-started.md
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/mainpage.md
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/parameters.md
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/performance.md
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/prepared-statement.md
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/streams.md
1>-- Installing: C:/Program Files/libpqxx/share/doc/libpqxx/thread-safety.md
===== Compilación: 1 correcto, 0 erróneo, 3 actualizado, 0 omitido =====
===== Compilar completado a las 18:20 y tardó 00,580 segundos =====

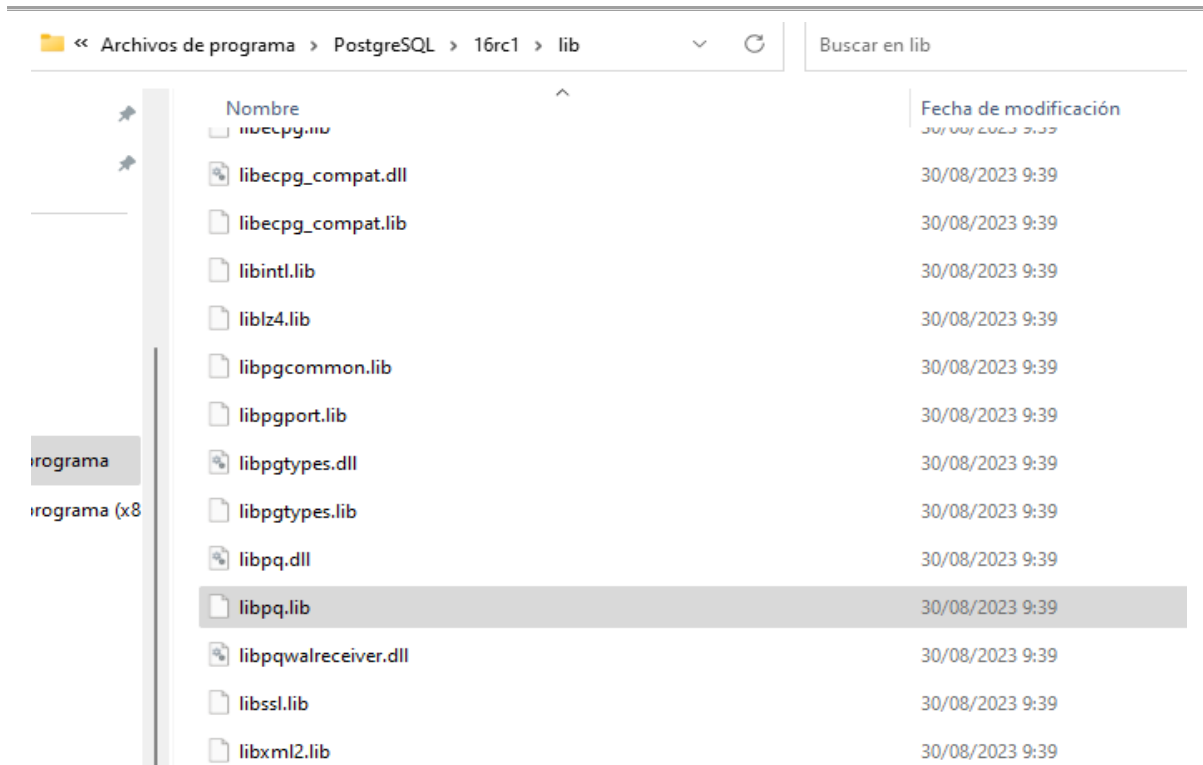
```

Una vegada acabada la instal·lació, tindreu un directori <libpqxx> al vostre directori d'Arxius de Programa amb els directoris que es veuen a l'esquerra de la imatge (include, lib, share).



Si heu fet servir les instruccions anteriors, l'arxiu .lib s'ha instal·lat juntament amb el PostgreSQL (la trobareu al directori <C:\Program Files\PostgreSQL<Versió>\lib>). Per més informació podeu accedir a la documentació de PostgreSQL⁷.

⁷ https://www.tutorialspoint.com/postgresql/postgresql_c_cpp.htm

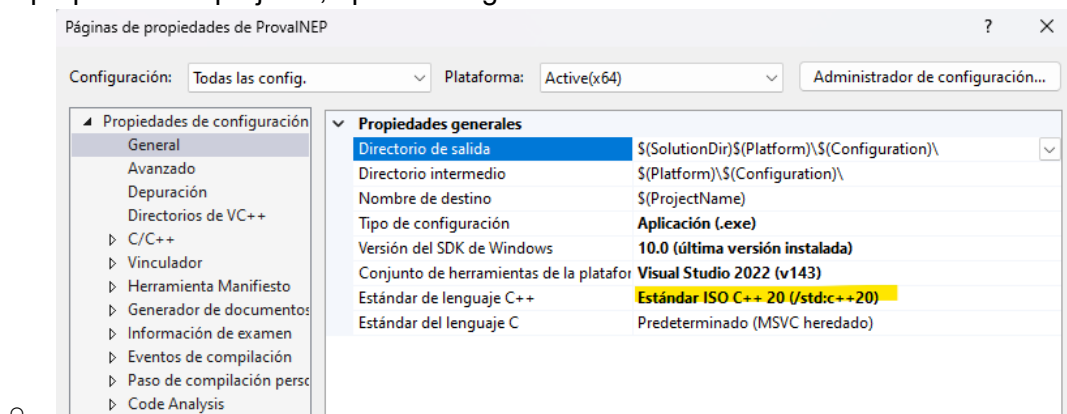


Es important que tingueu aquests arxius localitzats perquè necessitareu configurar l'entorn de treball de la vostra aplicació indicant aquests directoris com indica la següent secció.

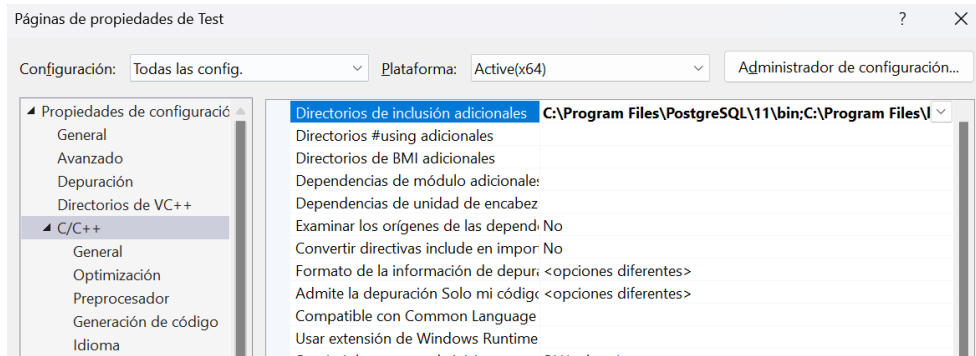
Configurar l'aplicació per accedir a la base de dades

Per poder fer servir la llibreria necessitem tenir ben configurat l'entorn de programació. Primer de tot necessitem crea el nostre projecte (aplicació de consola). Una vegada tenim el nostre projecte, hem de configurar alguns aspectes a la pantalla de <Propietats> del projecte perquè pugui fer servir la llibreria libpqxx que acabem d'instal·lar:

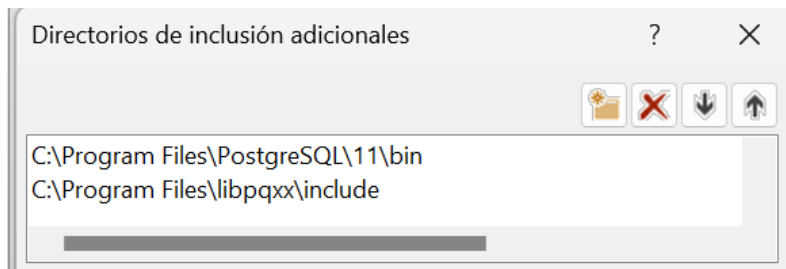
- Fa servir el mateix estàndard de C++ que hem fet servir per compilar la llibreria libpqxx, a les propietats del projecte, opció Configuració/General



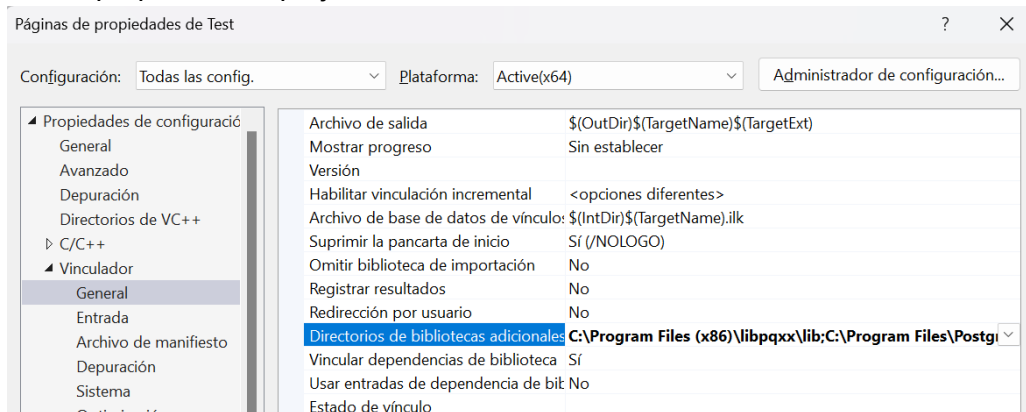
- Que tenim el path del directori <include> de la llibreria libpqxx als <Directorios de inclusión adicionales> de la opció <General> del grup <C/C++> de les propietats del projecte.



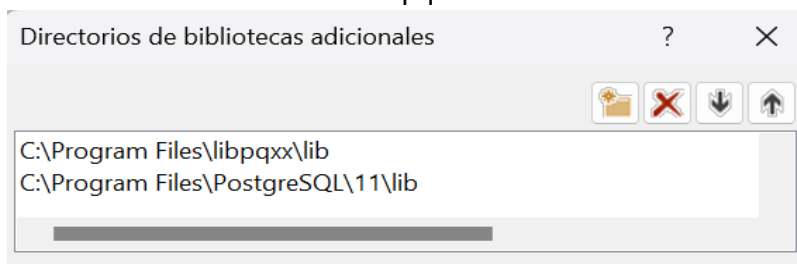
- En aquest camp també heu de tenir el directori <bin> del PostgreSQL que haureu instal·lat anteriorment.



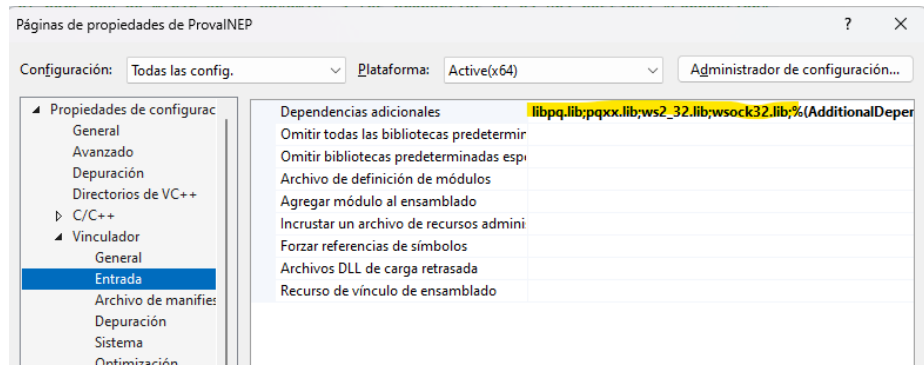
- Que tenim el path del directori <lib> del PostgreSQL (on està l'arxiu libpq.lib) als <Directorios de inclusión adicionales> de la opció <General> del grup <Vinculador> (linker) de les propietats del projecte.



- ■ necessitem la llibreria: libpq.lib
- En aquest camp també heu de tenir el directori <lib> de la llibreria libpqxx
- ■ necessitem la llibreria: pqxx.lib



- A la mateix grup <Vinculador>, a la opció <Entrada> (Input), al camp <Dependencias adicionales> afegim: libpq.lib;pqxx.lib;ws2_32.lib;wssock32.lib



Exemple de connexió a la base de dades per consultar informació

El següent codi, que correspon a l'arxiu .cpp on tingueu el main, accedeix a la base de dades helloINEP, que heu creat anteriorment, en concret mostra la informació que hi ha a la taula musics. Haureu de fer servir la contrasenya que heu configurat pel superusuari a l'instal·lar el progreSQL.

```
#include <iostream>
#include <pqxx/pqxx>

int main() {
    try {
        // Per establir la connexió amb la base de dades, es crea una
        pqxx::connection
        // per confirmar el port heu de mirar en el pgAdmin, a les propietats hi ha
        una pestanya "Connection"
        // l'usuari postgres és el superusuari que s'ha creat durant la
        instal·lació del postgresQL
        // heu de posar la contrasenya que heu escollit durant el procés
        d'instal·lació
        pqxx::connection conn("dbname=helloINEP user=postgres password=inep
        hostaddr=127.0.0.1 port=5433");

        if (conn.is_open()) {
            std::cout << "Connexió exitosa amb la base de dades." << std::endl <<
            std::endl;
        }
        else {
            std::cerr << "Error de connexió amb la base de dades." << std::endl <<
            std::endl;
            return 1;
        }

        // Per realitzar una consulta...
        // 1.- es crea un pqxx::work amb la connexió que hem creat
        pqxx::work txn(conn);
        // 2.- s'executa una comanda en SQL que correspon a la consulta
        pqxx::result result = txn.exec("SELECT * FROM musics");
        // si a l'executar us dona el missatge que no troba la taula, potser
        necessiteu posar:
        //      "SELECT * FROM public.musics" o "SELECT * FROM
        public.\"MUSICS\"
        // podeu confirmar la sintaxi de la comanda en el pgAdmin, sobre la taula
        escollir la opció Scripts\SELECT Script

        // Per mostrar el resultat de la consulta, hem de recórrer les files
        (primer for) ...
        std::cout << "-----" << std::endl;
```



```
std::cout << "Opció de recórrer el resultat amb dos loops" << std::endl;
std::cout << "-----" << std::endl;
for (const auto& row : result) {
    // ... i per cada fila, es recorren els camps (l'ordre és el que es veu
a les columnes al pgAdmin
    for (const auto& field : row) {
        std::cout << field.name() << ": " << field.c_str() << "\t";
    }
    std::cout << std::endl;
}

// també es poden recórrer els resultats com si fosin una matriu de files i
columnes (amb indexos començant per 0)
std::cout << "-----" << std::endl;
std::cout << "Opció de recórrer el resultat com una matriu" << std::endl;
std::cout << "-----" << std::endl;
for (size_t i = 0; i < result.size(); ++i) {
    for (size_t j = 0; j < result[i].size(); ++j) {
        std::cout << result[i][j].name() << ": " << result[i][j].c_str() <<
"\t";
    }
    std::cout << std::endl;
}

// Finalment, s'ha de confirmar la transacció
txn.commit();

// La connexió es tanca automàticament al sortir del try
}
catch (const std::exception& e) {
    std::cerr << "Error: " << e.what() << std::endl;
    return 1;
}

return 0;
}
```