**Popis databázy:**

Nižšie sú zobrazené tri tabuľky, ktoré sú používané na evidenciu zamestnancov pre štruktúry firiem.

Tabuľka **Employee** ukladá údaje o zamestnancoch. Každý pridávaný zamestnanec dostáva automaticky priradené employeeID. Všetky údaje okrem titulu sú vyžadované(pokiaľ zamestnanec nemá titul tak ostáva null).

V tabuľke **Structure** sú uložené informácie o štruktúre firmy. StructureCode slúži na identifikáciu štruktúry. Táto hodnota sa pridáva automaticky pri vkladaní. Stĺpec type určuje o aký typ štruktúry ide(1-firma, 2-divízia, 3-projekt, 4-oddelenie). BossID odkazuje na nejakého zamestnanca v tabuľke Employee, ktorý je vedúci danej štruktúry. UpperStructureID určuje pod akú štruktúru spadá(ak nespadá pod žiadnu, tak má hodnotu null).

|  |
| --- |
| **Structure** |
| **structureCode** (**PK**)(int, NOT NULL) |
| **name** (varchar(50), NOT NULL) |
| **type** (int, NOT NULL) |
| **bossID (FK)**(int, NOT NULL) |
| **upperStructureID (FK)**(varchar(20), NULL) |

Posledná tabuľka **BelongsTo** špecifikuje pod akú/é štruktúru/y zamestnanec patrí. Určuje sa to pomocou identifikátorov zamestnanca(employeeID) a štruktúry(structureID).

|  |
| --- |
| **Employee** |
| **employeeID** (**PK**)(int, NOT NULL) |
| **degree** (varchar(10), NULL) |
| **name** (varchar(20), NOT NULL) |
| **surname** (varchar(20), NOT NULL) |
| **cellPhone** (varchar(20), NOT NULL) |
| **email** (varchar(20), NOT NULL) |

|  |
| --- |
| **BelongsTo** |
| **id** (**PK**)(int, NOT NULL) |
| **employeeID (FK)**(int, NOT NULL) |
| **structureID (FK)**(int, NOT NULL) |

**Rozbehanie projektu:**

Projekt z gitu sa dá jednoducho otvoriť vo Visual Studiu. Vytvorenie databázy a všetkých potrebných tabuliek sa vykonáva pomocou priloženého vygenerovaného skriptu(DB\_script). Následne je v projekte v triede StartUp vhodné overiť správnosť connection stringu k databáze. Na testovanie API sa môže použiť Postman klient.

**Popis operácií nad tabuľkami:**

Nad všetkými troma tabuľkami boli vytvorené tieto operácie:

* **Get[Employees/Structures/BelongsTos]:** vypíše všetky údaje pre zvolenú tabuľku. Pristupuje sa pomocou HTTTPGET s koncovkou **/api/[employees/structures/belongstos]** napr. /api/structures
* **Get[Employee/Structure/BelongsTo]:** vypíše pre tabuľku podľa zvoleného identifikátora. Využíva sa HTTPGET kde sa za posledné lomítko určí id: **/api/[employees/structures/belongstos]/[id]** napr. /api/structures/4
* **Create[Employee/Structure/BelongsTo]:** vytvorí nový údaj do tabuľky. Na odoslanie požiadavky sa používa HTTPPOST s koncovou adresou **/api/[employees/structures/belongstos]** napr. /api/structures a do tela sa vložia údaje.
* **Update[Employee/Structure/BelongsTo]:** slúži na upravovanie údajov. Používa sa HTTPPUT s koncovou adresou **/api/[employees/structures/belongstos]/[id].** K potrebnému údaju pristupujeme pomocou jeho id. Napr. /api/structures/4 a v tele pošleme upravené údaje.
* **Delete[Employee/Structure/BelongsTo]:** pomocou tejto operácie mažeme požadovaný údaj z tabuľky. Na mazanie sa používa HTTPDELETE s adresou **/api/[employees/structures/belongstos]/[id].** K požadovanému údaju sa dostaneme pomocou zadávaného id. Napr. /api/structures/4.