

**Generovanie dát**

Alena Martinková, MAR0702

DAIS 2

# **Generovanie dát**

Na generovanie dát som využila viacero spôsobov. Na základné tabuľky, bez komplikovanejších väzieb, som použila SQL Data Generator. Na vygenerovanie dát väzobnej tabuľky s viacerými cudzími kľúčmi som využila jazyk Python s knižnicou. V jednom prípade som použila možnosť vloženia jednoduchého Python skriptu priamo do SQL Data Generatora – toto vkladanie bolo jedno z najpomalších aj keď nemalo toľko dát – z dôvodu použitia Python skriptu priamo v generátore.

## **Pravidlá generovania**

Hodnoty predstavujúce cudzie kľúče boli generované tak, aby pomer ich použitia zhruba odpovedal realite. Taktiež boli generované tak, aby spĺňali potrebné integritné obmedzenia.

### **League**

|  |  |  |
| --- | --- | --- |
| **Názov** | **Dátový typ** | **Popis** |
| leagueID | INTEGER | Inkrementácia + 1 |
| name | VARCHAR(100) | Reťazec - použitie csv vloženého do generátora |
| category | INTEGER | Číslo |

### **Team**

|  |  |  |
| --- | --- | --- |
| **Názov** | **Dátový typ** | **Popis** |
| teamID | INTEGER | Inkrementácia + 1 |
| leagueID | INTEGER | **Cudzí kľúč**, ktorý sa opakoval vždy 10x, tak aby **rank** vždy sedel |
| name | VARCHAR(100) | Reťazec - použitie csv vloženého do generátora |
| rank | INTEGER | Číslo od 1 do 10, opakujúce sa v pravidelných intervaloch tak, aby sedelo v rámci cudzieho kľúča **leagueID** |
| covid | INTEGER | Číslo 0 alebo 1 |
| quarantinedFrom | DATE/NULL | Vygenerované na hodnotu NULL |

* + 1. **Player**

|  |  |  |
| --- | --- | --- |
| **Názov** | **Dátový typ** | **Popis** |
| playerID | INTEGER | Inkrementácia + 1 |
| teamID | INTEGER | **Cudzí kľúč**, ktorý sa opakoval vždy 18x - 20x, tak aby mal každý tím +- rovnaký počet hráčov |
| email | VARCHAR(100) | Reťazec |
| dateOfBirth | DATE | Dátum od roku 1980 do roku 2003 |
| firstName | VARCHAR(50) | Reťazec |
| lastName | VARCHAR(50) | Reťazec |
| stick | CHAR(1) / NULL | L,R alebo NULL - značí Left, Right alebo NULL = brankár |
| covid | INTEGER | Číslo 0 alebo 1 |

### **Statistic**

|  |  |  |
| --- | --- | --- |
| **Názov** | **Dátový typ** | **Popis** |
| statisticID | INTEGER | Inkrementácia + 1 |
| playerID | INTEGER | **Cudzí kľúč,** každý použitý práve raz nahratý z csv súboru |
| teamID | INTEGER | **Cudzí kľúč,** použitý z csv |
| goals | INTEGER | Číslo medzi 0 a 200 |
| assists | INTEGER | Číslo medzi 0 a 200 |

Na vygenerovanie dát tejto tabuľky som okrem samotného generátora použila taktiež csv súbor, ktorý som si vygenerovala pomocou skriptu v Pythone. Do tohto csv súboru som vždy zapisovala kombináciu playerID – teamID, tak aby korektne reprezentovala dáta z tabuľky Team.

## **PlayerTransferHistory**

|  |  |  |
| --- | --- | --- |
| **Názov** | **Dátový typ** | **Popis** |
| playerTransferID | INTEGER | Inkrementácia + 1 |
| playerID | INTEGER | **Cudzí kľúč,** každý použitý práve raz |
| oldTeamID | INTEGER | **Cudzí kľúč** |
| newTeamID | INTEGER | **Cudzí kľúč** |
| date | DATE | Dátum od roku 2018 do 30.9.2021 |

Na vygenerovanie tejto tabuľky som okrem samotného generátora použila priamo v ňom taktiež Python skript a csv súbor zhodný zo súborom použitým v tabuľke Statistic. Stĺpce playerID a newTeamID sa vyplnili z csv a Python skript zabezpečil, že oldTeamID nie je zhodné s newTeamID.

### **Pitch**

|  |  |  |
| --- | --- | --- |
| **Názov** | **Dátový typ** | **Popis** |
| pitchID | INTEGER | Inkrementácia + 1 |
| capacity | INTEGER | Číslo od 50 do 500 |
| name | VARCHAR(100) | Reťazec - použitie csv vloženého do generátora |

### **TeamMatch**

|  |  |  |
| --- | --- | --- |
| **Názov** | **Dátový typ** | **Popis** |
| teamMatchID | INTEGER | Inkrementácia + 1 |
| firstTeamID | INTEGER | **Cudzí kľúč** |
| secondTeamID | INTEGER | **Cudzí kľúč** |
| pitchID | INTEGER | **Cudzí kľúč** |
| firstTeamGoals | INTEGER | Číslo od 0 do 20 |
| secondTeamGoals | INTEGER | Číslo od 0 do 20 |
| postponed | INTEGER | Číslo 0 alebo 1 |
| date | DATE | Dátum |

Dáta do tejto tabuľky boli všetky vygenerované pomocou Python skriptu. Najprv som si vytvorila slovník so všetkými tímami a im priradenou hodnotou 0 – tá označovala počet zápasov, ktoré boli vytvorené. Následne som si vytvorila druhý slovník, ktorý udržiaval dáta o tom aké tímy hrajú v daný deň zápas. V cykle sa vždy vybrali dva náhodné tímy zo slovníka a postupovalo sa takto:

1. Dokým boli rovnaké vyberali sa
2. Skontrolovalo sa či v deň, ktorý je momentálne nastavený už nemajú zápas -> ak áno pregeneroval sa tím, ktorý zápas má a znova sa skontroloval bod I. a taktiež II.
3. Keď bolo všetko v poriadku, vygenerovalo sa náhodné skóre (firstTeamGoals, secondTeamGoals), taktiež hodnota postponed, hodnota pitchID sa vždy pridávala sekvenčne
4. V slovníku s hodnotami počtu zápasov daného tímu sa pre dané tímy hodnota zväčšila o jedna – ak bola presiahnutá nastavená hranica (mnou nastavená na 250 tak, aby všetky tímy mali +- rovnaký počet zápasov) bol tím zo slovníka odstránený a tým pádom nemohol byť ďalej náhodne vybratý
5. Po každom 1000. zápase sa dátum zvýšil o jedna
6. Po dokončení cyklu sa hodnoty zapísali do csv súboru

### **Ticket**

|  |  |  |
| --- | --- | --- |
| **Názov** | **Dátový typ** | **Popis** |
| ticketID | INTEGER | Inkrementácia + 1 |
| teamMatchID | INTEGER | **Cudzí kľúč**, ktorý sa opakoval vždy 3x – 5x, tak aby mal každý zápas priradený nejaký lístok |
| firstName | VARCHAR(50) | Reťazec |
| lastName | VARCHAR(50) | Reťazec |
| price | SQLMONEY | Peniaze |
| storno | INTEGER | Číslo 0 alebo 1 |
| email | VARCHAR(100) | Reťazec |

## **Vkladanie dát do databázy**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabuľka** | **Počet záznamov** | **Čas** | **Spôsob** |
| League | 1000 | < 1s | Generátor |
| Team | 10 000 | < 1s | Generátor |
| Player | 200 000 | 15s | Generátor |
| Statistic | 100 000 | 10s | Generátor + csv vygenerované v Pythone |
| PlayerTransferHistory | 100 000 | 2m 40s | Generátor + Python skript + csv vygenerované v Pythone |
| TeamMatch | 1 199 999 | 18s | Csv vygenerované v Pythone + C# BulkInsert |
| Ticket | 3 500 000 | 5m | Generátor |
| Pitch | 100 000 | 7s | Generátor |
| **Celkovo:** | **5 210 999** | **8m 32s** | - |

Dáta boli vkladané buď pomocou spomínaného SQL Data Generatora za použitia csv súborov alebo Python skriptu a v jednom prípade pomocou jazyka C# a triedy SqlBulkCopy. Vkladanie pomocou jazyka C# prebiehalo z vygenerovaného csv súboru.

Vkladanie trvalo najdlhšie pre tabuľky *Ticket* a *PlayerTransferHistory*. Všeobecným problémom mohol byť prenos po sieti - internetové pripojenie doma nemám ideálne, spolu s faktom, že prenos bol spustený cez VPN. Okrem toho taktiež fakt, že obidve tabuľky boli vkladané cez generátor, ktorý vo všeobecnosti vkladal pomaly.

Pre tabuľku *Ticket* je ďalším dôvodom vkladanie veľkého počtu záznamov cez generátor spolu s podmienkou na cudzí kľúč – každý zápas má 3 – 5 lístkov, kde sa samozrejme kontroluje správna hodnota kľúča. Pozrela som sa aj na výslednú veľkosť tabuľky, ktorá je zhruba 250 MB – čo by taktiež vysvetľovalo dĺžku nahrávania.

Pre tabuľku *PlayerTransferHistory* spôsobilo spomalenie pridanie externého zdroju dát – csv súboru, spolu so skriptom spusteným priamo v generátore – skript v generátore výrazne spomalil chod generovania. Okrem toho sa musia v tabuľke kontrolovať tri cudzie kľúče, tzn. tri clustered index seek.