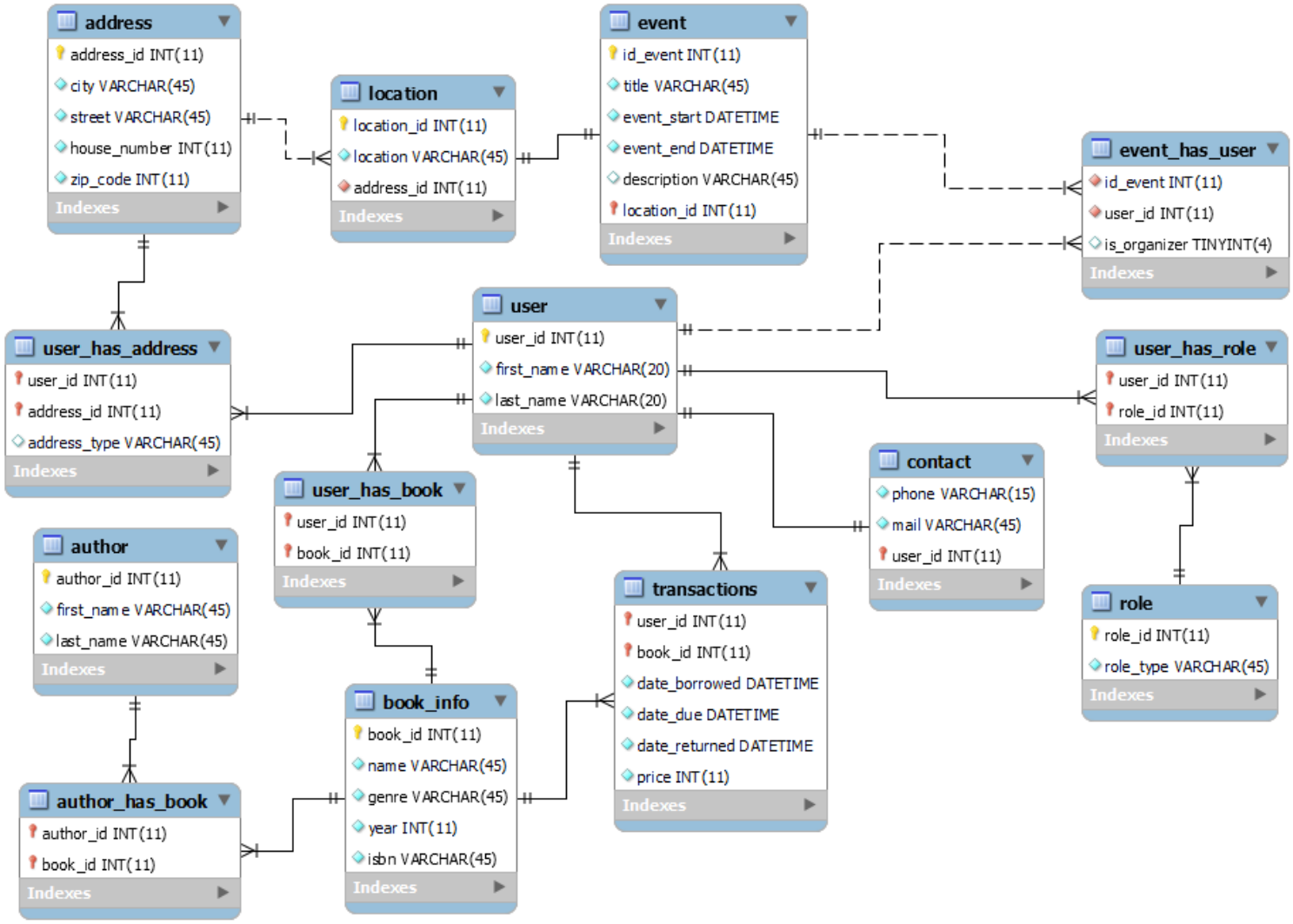
Bezpečnost Databázových Systémů – Databáze Knihovny

Autoři: **Michal Žernovič (230923)**, **Vojtěch Vaculík (230338)**

[GitHub] - Skupinový repositář

Schéma databáze



Databáze Knihovny

Jako téma naší databáze jsme si zvolili library management systém. Jde o databázi obsahující mimo jiné seznam knih a jejich vlastnosti, seznam lidí využívající knihovnické služby, jejich transakce, členství atd. Vzhledem k tomu, že digitální databáze nahradili knižní záznamy, jsme si zvolili právě tento systém, protože jde o případ z reálného světa. Tuto databázi jsme se pokusili navrhnout co nejrobustnější abychom předešli neustálým změnám. Mimo základní informace obsahuje také i členství, speciální workshopy a dárcovství knih. Tímto jsme se pokusili vybrat speciální možnosti, které mnoho knihoven nabízí a snažili se je implementovat přímo do našeho návrhu.

Samotný problém knihovního systému by se dal řešit různými způsoby a mohl by určitě obsahovat i více tabulek. Náš návrh však spočívá ve své jednoduchosti a blbuvzdornosti, tudíž je velmi přehledný a jednoduchý na použití.

Jako inspiraci pro tento systém, jsme použili inspiraci z několika reálných knihoven a jejich systémů, které jsou používány každý den. Díky těmto informacím jsme vyfiltrovali data, které jsou vhodné pro použití v databázi a které nejsou.

Popis tabulek

- **USER**
  Hlavní tabulka, má několik atributů, vážou se na ni všechny ostatní tabulky. Obsahuje informácie o používatelovi – **user_id** (BIGSERIAL lebo potrebujeme uložiť číslo, potenciálne veľké), **first_name** (VARCHAR lebo potrebujeme uložiť text) a **last_name** (VARCHAR lebo potrebujeme uložiť text).
- **ROLE**
  Tabulka určující, jakou roli má uživatel, uživatel může mít různé role např.: Admin, Manager, Employee, Donator, Customer.
  role_id (BIGSERIAL lebo potrebujeme uložiť číslo, potenciálne veľké), **role_type** (VARCHAR, potrebujeme textom nazvať rolu).
  Uživatel může mít více rolí.
    - **USER_HAS_ROLE**
      Mezi tabulka, která propojuje tabulku USER a ROLE, klíčové parametry jsou **user_id** a **role_id**. Oboje BIGSERIAL kvôli uloženiu čísla.
- **ADDRESS**
  Tabulka, která drží hodnoty různých adres. Uživatel může mít více než jednu adresu. Jedna adresa môže mať aj viac uživateľov. **address_id** (BIGSERIAL kvôli tomu že je to id). **city** (VARCHAR, potrebujeme napísať názov města). **street** (VARCHAR, potrebujeme názov ulice). **house_number** (INT, stačí nám číslo domu). **zip_code** (INT, ZIP sa označuje číslom)

    - **USER_HAS_ADDRESS**
      Mezi tabulka propojující tabulku USER a ADDRESS, klíčové parametry jsou zde **user_id** a **address_id**. Oboje BIGSERIAL kvôli uloženiu čísla.
- **LOCATION**
  Tato tabulka je provázána s tabulkou ADDRESS. Propojuje je společně s tabulkou EVENT, kde
  používá parametry **location_id** a **address_id.** Opisuje miesto kde sa koná EVENT. Při oboch sú použité BIGSERIAL kvôli číslu. Taktiež tam patrí **location** jako VARCHAR, ktorá slúži na slovné opísanie lokácie.
- **EVENT**
  Tabulka event určuje ony workshopy, je propojena s USER speciální mezi tabulkou a také je spojena s tabulkou ADDRESS za pomoci LOCATION, která udává lokaci daného EVENTU. **id_event, location_id** sú BIGSERIAL kvôli číslu, **title** je názov eventu, preto VARCHAR. **event_start, event_end** sú TIMESTAMP kvôli presnému dátumu a času eventu, čo je najvhodnejší dátový typ na takéto informácie. **description** je VARCHAR lebo ide o textový opis eventu.
    - **EVENT_HAS_USER**
      Mezi tabulka propojující USER s různými akcemi (eventy), jako klíčové parametry používá **id_event** a **user_id**. Oboje BIGSERIAL kvôli uloženiu čísla.
- **TRANSACTIONS**
  Tabulka transakcí drží historii všech transakcí, které kdy USER udělal a je propojena s tabulkou BOOK_INFO. Obsahuje aj čas požičania, termín vrátenia a samotné vrátenie. **dser_id** a **book_id** sú BIGSERIAL kvôli uloženiu čísla. **date_borrowed, date_due** a **date_returned** sú TIMESTAMP, kvôli presnému dátumu a času. **price** je INT kvôli tomu že ukladáme cenu ako číslovku.
- **BOOK_INFO**
  Drží všechny informace o knihách a jejich vlastnostech a je propojena s tabulkami AUTHOR a TRANSACTIONS. **book_id** je BIGSERIAL kvôli uloženiu čísla. **name, genre** a **isbn** sú VARCHAR, lebo ukladáme text (pri ISBN je to kombinácia čísel a pomlčiek). **year** je INT lebo ukladáme číslo.
    - **USER_HAS_BOOK**
      Mezitabulka přiřazující danou knihu k USER, klíčové parametry jsou zde **user_id** a **book_id.** Oboje BIGSERIAL kvôli uloženiu čísla.
- **AUTHOR**
  Obsahuje všechny informace o daném autorovi a propojuje je se všemi jeho knihami (Jeden autor môže mať viac kníh, jedna kniha môže mať viac autorov.) **author_id** je BIGSERIAL pretože ukladáme číslo, **first_name** a **last_name** sú text, preto volíme VARCHAR.
    - **AUTHOR_HAS_BOOK**
      Mezi tabulka propojující AUTHOR s jeho knihami, klíčové parametry jsou **author_id** a **book_id.** Oboje BIGSERIAL kvôli uloženiu čísla.

- **CONTACT**
  Obsahuje kontaktné informácie USER-a. One-to-one relationship. u**ser_id** je BIGSERIAL pretože ide o číslo id, **mail** je VARCHAR pretože ukladáme textový názov mailu, **phone** je VARCHAR pretože pridávame aj + pred predvoľbu.

3tí norm. forma (3rd normal form)

Spĺňa 1st normal form, pretože:

- Všetky riadky sú unikátne (máme ID's)
- Každá bunka obsahuje jednu hodnotu
- Nedá sa už ďalej rozdeliť

Spĺňa 2nd normal form, pretože:

- Spĺňa 1st normal form
- Žiadne čiastočné závislosti, sú kvôli tomu vytvorené extra tabuľky

Spĺňa 3rd normal form, pretože:

- Spĺňa 1st NF a 2nd NF
- Všetky tranzitívne závislosti sú odstránené, nekľúčové atribúty sú umiestnené vo vlastnej tabuľke

DDL script – PostgreSQL


db_scripts_PostgreSQL.sql


postgres-queries-final.txt

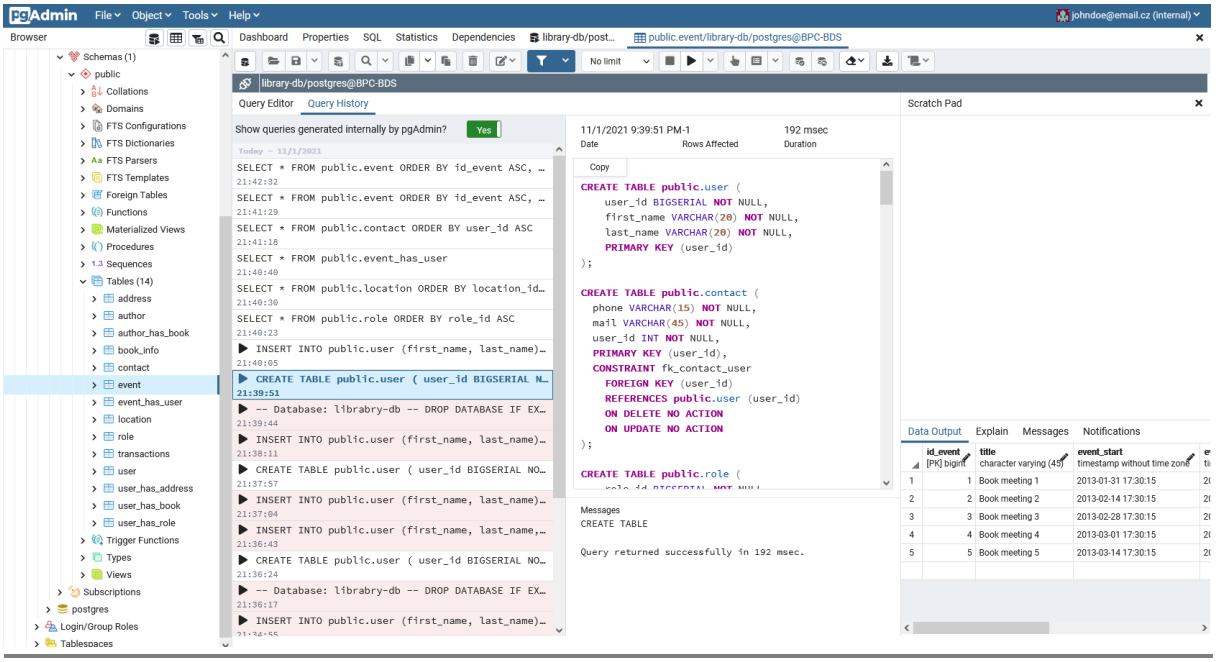
Poznámka: .sql súbor je skript ktorý funguje v mySQL workbench, ale na pgadmin je potrebné použiť skripty v .txt súbore.

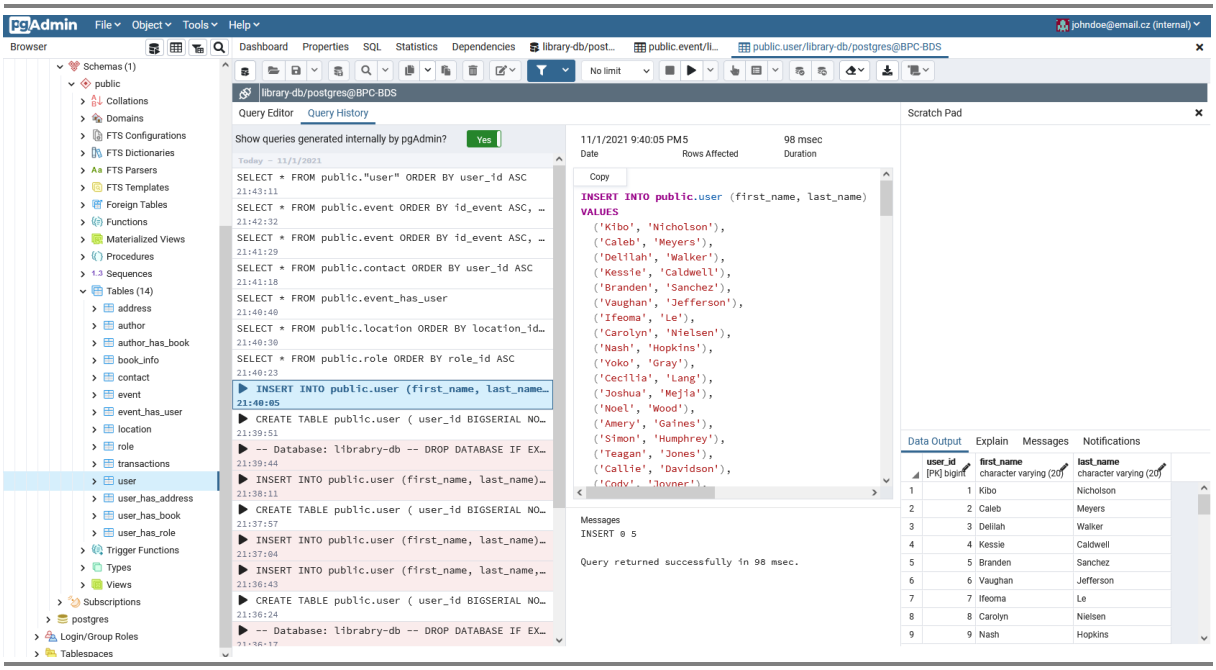
DDL script – mySQL

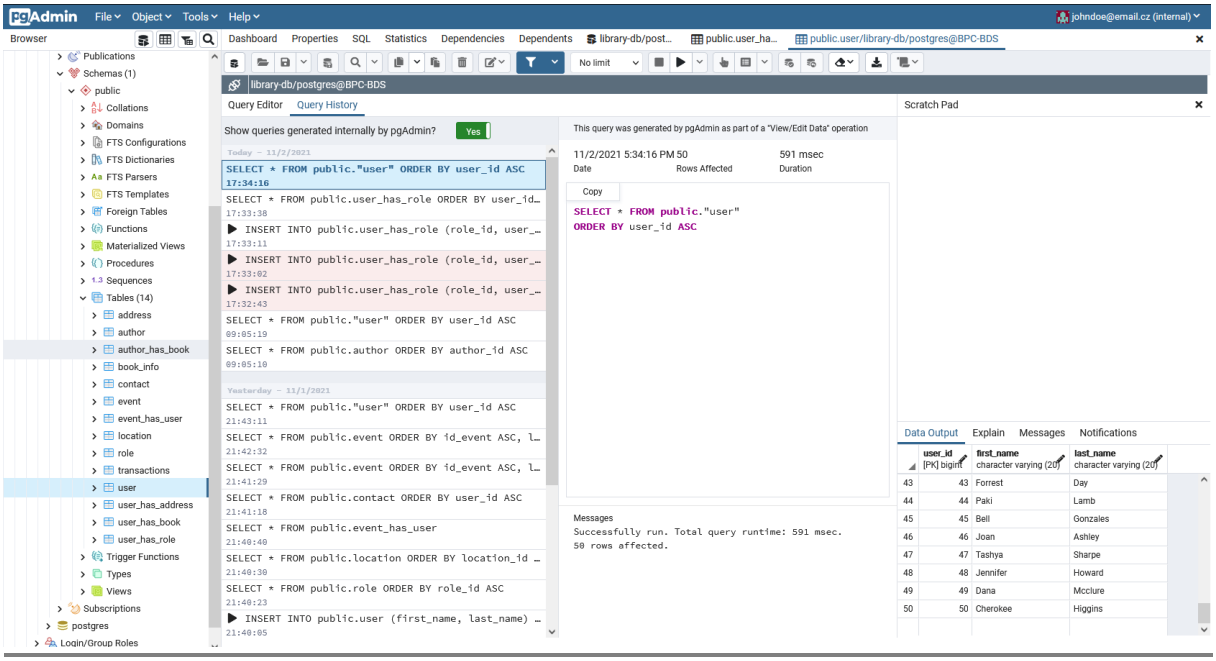

db_scripts_mySQL.sql

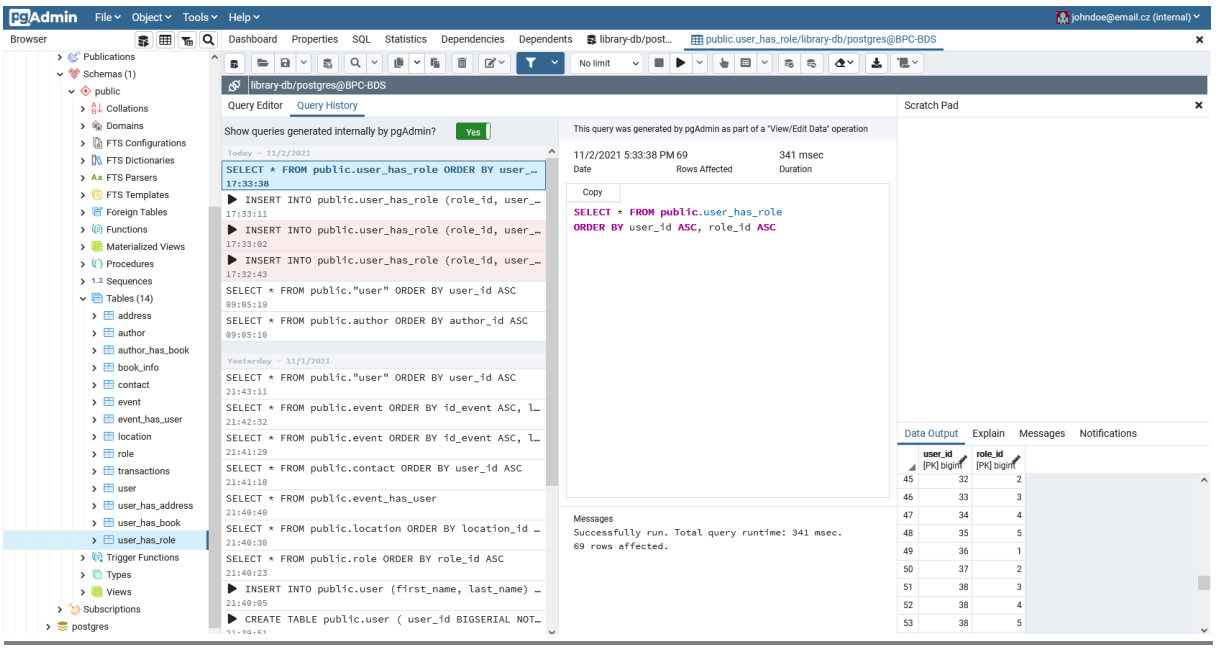

db_scripts_mySQL.txt

Created Database (screenshots – pgAdmin)






pgAdmin
File
Object
Tools
Help
johndoe@email.cz (internal)
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
library-db/post...
public.user_ha...
public.user/library-db/postgres@BPC-BDS
No limit
library-db/postgres@BPC-BDS
Query Editor
Query History
Show queries generated internally by pgAdmin?
Yes
Today – 11/2/2021
SELECT * FROM public."user" ORDER BY user_id ASC
17:34:16
SELECT * FROM public.user_has_role ORDER BY user_id…
17:33:38
INSERT INTO public.user_has_role (role_id, user_…
17:33:11
INSERT INTO public.user_has_role (role_id, user_…
17:33:02
INSERT INTO public.user_has_role (role_id, user_…
17:32:43
SELECT * FROM public."user" ORDER BY user_id ASC
09:05:19
SELECT * FROM public.author ORDER BY author_id ASC
09:05:10
Yesterday – 11/1/2021
SELECT * FROM public."user" ORDER BY user_id ASC
21:43:11
SELECT * FROM public.event ORDER BY id_event ASC, l…
21:42:32
SELECT * FROM public.event ORDER BY id_event ASC, l…
21:41:29
SELECT * FROM public.contact ORDER BY user_id ASC
21:41:18
SELECT * FROM public.event_has_user
21:40:40
SELECT * FROM public.location ORDER BY location_id …
21:40:30
SELECT * FROM public.role ORDER BY role_id ASC
21:40:23
INSERT INTO public.user (first_name, last_name) …
21:40:05
This query was generated by pgAdmin as part of a "View/Edit Data" operation
11/2/2021 5:34:16 PM
Date
50
Rows Affected
591 msec
Duration
Copy
SELECT * FROM public."user"
ORDER BY user_id ASC
Messages
Successfully run. Total query runtime: 591 msec.
50 rows affected.
Scratch Pad
Data Output
Explain
Messages
Notifications
user_id
[PK] bigint
first_name
character varying (20)
last_name
character varying (20)
43 43 Forrest Day
44 44 Paki Lamb
45 45 Bell Gonzales
46 46 Joan Ashley
47 47 Tashya Sharpe
48 48 Jennifer Howard
49 49 Dana Mcclure
50 50 Cherokee Higgins
Publications
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (14)
address
author
author_has_book
book_info
contact
event
event_has_user
location
role
transactions
user
user_has_address
user_has_book
user_has_role
Trigger Functions
Types
Views
Subscriptions
postgres
Login/Group Roles


pgAdmin
File
Object
Tools
Help
johndoe@email.cz (internal)
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
library-db/post...
public.user_has_role/library-db/postgres@BPC-BDS
No limit
library-db/postgres@BPC-BDS
Query Editor
Query History
Show queries generated internally by pgAdmin?
Yes
Today – 11/2/2021
SELECT * FROM public.user_has_role ORDER BY user_…
17:33:38
INSERT INTO public.user_has_role (role_id, user_…
17:33:11
INSERT INTO public.user_has_role (role_id, user_…
17:33:02
INSERT INTO public.user_has_role (role_id, user_…
17:32:43
SELECT * FROM public."user" ORDER BY user_id ASC
09:05:19
SELECT * FROM public.author ORDER BY author_id ASC
09:05:10
Yesterday – 11/1/2021
SELECT * FROM public."user" ORDER BY user_id ASC
21:43:11
SELECT * FROM public.event ORDER BY id_event ASC, l…
21:42:32
SELECT * FROM public.event ORDER BY id_event ASC, l…
21:41:29
SELECT * FROM public.contact ORDER BY user_id ASC
21:41:18
SELECT * FROM public.event_has_user
21:40:40
SELECT * FROM public.location ORDER BY location_id …
21:40:30
SELECT * FROM public.role ORDER BY role_id ASC
21:40:23
INSERT INTO public.user (first_name, last_name) …
21:40:05
CREATE TABLE public.user (user_id BIGSERIAL NOT…
This query was generated by pgAdmin as part of a "View/Edit Data" operation
11/2/2021 5:33:38 PM
Date
69
Rows Affected
341 msec
Duration
Copy
SELECT * FROM public.user_has_role
ORDER BY user_id ASC, role_id ASC
Messages
Successfully run. Total query runtime: 341 msec.
69 rows affected.
Scratch Pad
Data Output
Explain
Messages
Notifications
user_id
[PK] bigint
role_id
[PK] bigint
45 32 2
46 33 3
47 34 4
48 35 5
49 36 1
50 37 2
51 38 3
52 38 4
53 38 5
Publications
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (14)
address
author
author_has_book
book_info
contact
event
event_has_user
location
role
transactions
user
user_has_address
user_has_book
user_has_role
Trigger Functions
Types
Views
Subscriptions
postgres