

MAGAZIN GPUS

Pascaru Dan-Alexandru

Coordonatori:
Mlisan Mirela

C. N. “Mircea cel Bătrân”

Clasa a 12-a A

2023

CUPRINS

Despre MySQL

> 3

Despre Python

> 4

Tabele

> 5

ERD

> 6

Reguli structurale

> 7

Utilizare

> 8

Interogări

> 9



MySQL este un sistem de gestiune a bazelor de date relaționale, produs de compania suedeză MySQL AB și distribuit sub Licența Publică Generală GNU. Este cel mai popular SGBD open-source la ora actuală, fiind o componentă cheie a stivei LAMP (Linux, Apache, MySQL, PHP).

Deși este folosit foarte des împreună cu limbajul de programare PHP, cu MySQL se pot construi aplicații în orice limbaj major. Există multe scheme API disponibile pentru MySQL ce permit scrierea aplicațiilor în numeroase limbaje de programare pentru accesarea bazelor de date MySQL, cum ar fi: C, C++, C#, Java, Perl, PHP, Python, FreeBasic, etc., fiecare dintre acestea folosind un tip specific API. O interfață de tip ODBC denumită MyODBC permite altor limbaje de programare ce folosesc această interfață, să interacționeze cu bazele de date MySQL cum ar fi ASP sau Visual Basic. În sprijinul acestor limbaje de programare, unele companii produc componente de tip COM/COM+ sau .NET (pentru Windows) prin intermediul cărora respectivele limbaje să poată folosi acest SGBD mult mai ușor decât prin intermediul sistemului ODBC. Aceste componente pot fi gratuite (ca de exemplu MyVBQL) sau comerciale.

Licența GNU GPL nu permite încorporarea MySQL în softuri comerciale; cei care doresc să facă acest lucru pot achiziționa, contra cost, o licență comercială de la compania producătoare, MySQL AB.

MySQL este componentă integrată a platformelor LAMP sau WAMP (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Popularitatea sa ca aplicație web este strâns legată de cea a PHP-ului care este adesea combinat cu MySQL și denumit Duo-ul Dinamic. În multe cărți de specialitate este precizat faptul ca MySQL este mult mai ușor de învățat și folosit decât multe din aplicațiile de gestiune a bazelor de date, ca exemplu comanda de ieșire fiind una simplă și evidentă: „exit” sau „quit”.

Pentru a administra bazele de date MySQL se poate folosi modul linie de comandă sau, prin descărcare de pe internet, o interfață grafică: MySQL Administrator și MySQL Query Browser. Un alt instrument de management al acestor baze de date este aplicația gratuită, scrisă în PHP, phpMyAdmin.

MySQL poate fi rulat pe multe dintre platformele software existente: AIX, FreeBSD, GNU/Linux, Mac OS X, NetBSD, Solaris, SunOS, Windows.



Python este un limbaj de programare dinamic multi-paradigmă, creat în 1989 de programatorul olandez Guido van Rossum. Van Rossum este și în ziua de astăzi un lider al comunității de dezvoltatori de software care lucrează la perfecționarea limbajul Python și implementarea de bază a acestuia, CPython, scrisă în C. Python este un limbaj multifuncțional folosit de exemplu de către companii ca Google sau Yahoo! pentru programarea aplicațiilor web, însă există și o serie de aplicații științifice sau de divertisment programate parțial sau în întregime în Python. Popularitatea în creștere, dar și puterea limbajului de programare Python au dus la adoptarea sa ca limbaj principal de dezvoltare de către programatori specializați și chiar și la predarea limbajului în unele medii universitare. Din aceleași motive, multe sisteme bazate pe Unix, inclusiv Linux, BSD și Mac OS X includ din start interpretatorul CPython.

Python pune accentul pe curățenia și simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare ca C. În ceea ce privește paradigma de programare, Python poate servi ca limbaj pentru software de tipul object-oriented, dar permite și programarea imperativă, funcțională sau procedurală. Sistemul de tipizare este dinamic iar administrarea memoriei decurge automat prin intermediul unui serviciu „gunoier” (garbage collector). Alt avantaj al limbajului este existența unei ample biblioteci standard de metode.

Implementarea de referință a Python este scrisă în C și poartă deci numele de CPython. Această implementare este software liber și este administrată de fundația Python Software Foundation.

OPINIE & PREFERINȚE

Am ales MySQL și Python deoarece sunt două programe populare și mi-am dorit să le cunosc, dar și deoarece FoxPro este destul de vechi și am văzut ca nu mai este LTS(long time support).

Am ales MySQL deoarece sintaxa este relativ simplă. Inițial m-am uitat la SQLite deoarece are integrare cu Python fără software suplimentar, dar am găsit mai multe surse de informații despre MySQL și conectarea acestuia la Python.

Am ales Python deoarece este o limba modernă și relativ ușoară pentru cineva care știe deja C++.

PRIMARY KEY	FOREIGN KEY	NOT USED IN RELATIONSHIPS
-------------	-------------	---------------------------

STOCK

gpu_id INT	price INT	deadstock BOOL	size DECIMAL	sup_id INT	unique_id INT AUTO
1001	450	true	12	3	1
1002	525	true	8	1	2
1003	800	true	10	2	3
1003	850	true	10	6	etc.
1004	300	true	12	3	
1005	600	true	12	5	
1006	1,200	true	10	5	
1007	1,800	true	24	4	
1008	340	true	8	1	
1009	150	true	8	3	
2001	250	true	8	3	
3001	300	false	12	2	
3002	800	true	16	5	
3002	810	true	16	4	
3002	830	true	16	4	
3003	340	true	12	1	
3004	1,050	true	20	5	
3007	250	true	16	1	
3005	1,075	true	16	6	
3005	370	true	16	6	
3006	120	true	8	2	
4001	1,300	true	16	4	
4002	470	true	12	3	
4003	280	false	8	5	
5002	625	true	12	5	
5003	760	true	8	3	
5001	490	true	12	4	
6001	675	false	12	6	
6002	770	true	16	2	
6003	1,500	true	16	2	

BRANDS

brand VARCHAR	brand_id INT	site VARCHAR
Nvidia	1	nvidia.com
Intel	2	intel.com
AMD	3	amd.com
Corsair	4	corsair.com
MSI	5	msi.com
ASUS	6	asus.com

GPU

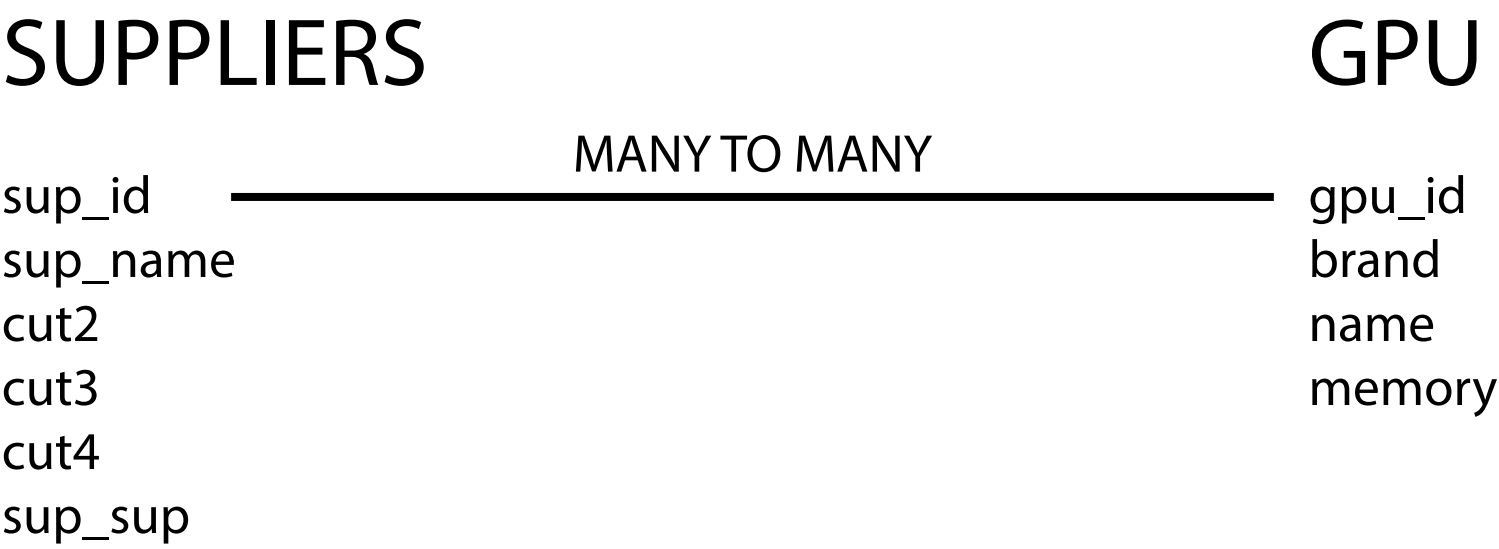
gpu_id INT	brand VARCHAR	name VARCHAR	memory VARCHAR
1001	Nvidia	RTX 3060	12 GB
1002	Nvidia	RTX 3070	8 GB
1003	Nvidia	RTX 3080	10 GB
1004	Nvidia	RTX 2060	12 GB
1005	Nvidia	RTX 3060 Ti	12 GB
1006	Nvidia	RTX 3080 Ti	10 GB
1007	Nvidia	RTX 4090	24 GB
1008	Nvidia	RTX 2070 SUPER	8 GB
1009	Nvidia	RTX 3050	8 GB
2001	Intel	Arc A750	8 GB
3001	AMD	Radeon RX 6750	12 GB
3002	AMD	Radeon RX 6900	16 GB
3003	AMD	Radeon RX 6700	12 GB
3004	AMD	Radeon RX 7900	20 GB
3005	AMD	Radeon RX 6800	16 GB
3006	AMD	Radeon RX 6600	8 GB
3007	AMD	Radeon RX 6950	16 GB
4001	Corsair	RTX 4080	16 GB
4002	Corsair	RTX 3060	12 GB
4003	Corsair	ARC A770	8 GB
5001	Asus	ROG Strix Radeon RX 6700	12 GB
5002	Asus	RTX 3060 Ti Rog Strix	12 GB
5003	Asus	RTX 3070 Noctua	8 GB
6001	MSI	RX 6750 XT	12 GB
6002	MSI	Radeon RX 6800	16 GB
6003	MSI	RTX 4080	16 GB

SUPPLIERS

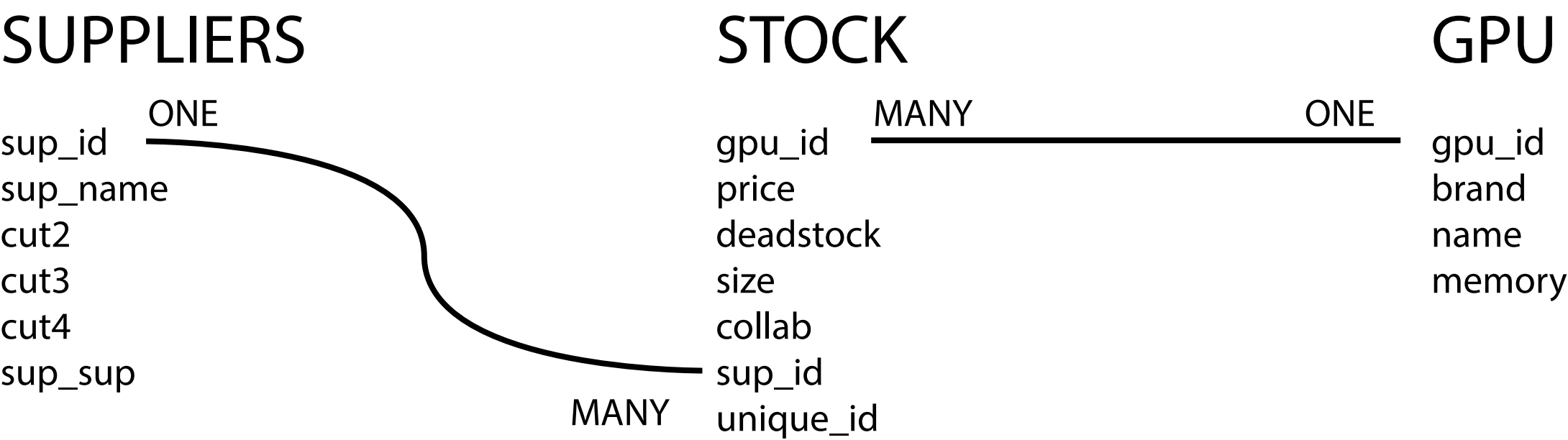
sup_id INT	sup_name VARCHAR	cut2 INT	cut3 INT	cut4 INT	sup_sup INT
1	Alex	0	0	10	Null
2	Matei	0	0	25	Null
3	Tudor	0	10	30	1
4	Vlad	10	15	20	Null
5	Bianca	10	10	10	2
6	Razvan	0	10	15	2

STOCK	GPU	BRANDS	SUPPLIERS
gpu_id	gpu_id	brand	sup_id
price	brand	brand_id	sup_name
deadstock	name	site	cut2
size	memory		cut3
collab			cut4
sup_id			sup_sup
unique_id			

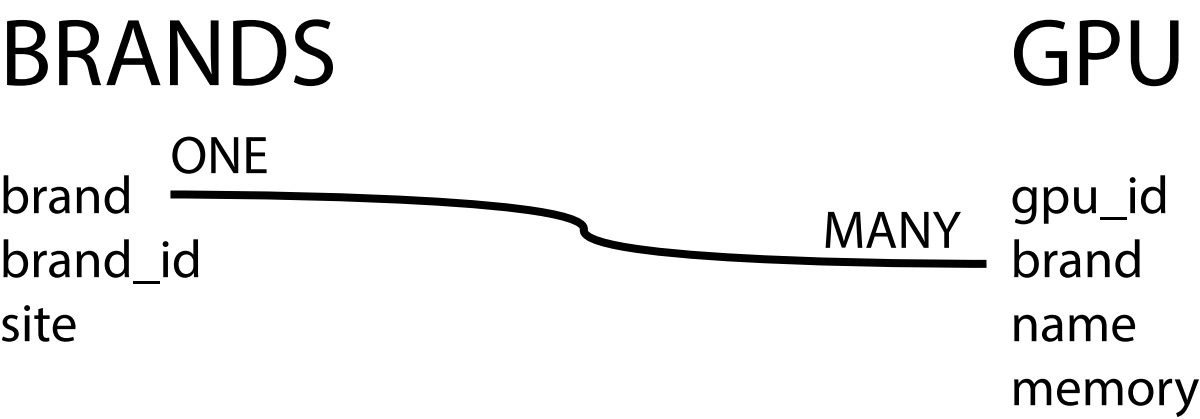
ERD INIȚIAL



ERD FINAL



EXTRA



Reguli Structurale & Restricții

1. $\text{snkr_id} = \text{brand_id} * 1000 + \text{Numarul de GPU-uri introduse anterior de la același brand} + 1$
2. Datorită regulei 1 nu se pot șterge intrări din tabelul gpu.
3. Fiecare supplier are reduceri specifice.

Utilizare

Utilizarea in terminal:

- > python scriptName.py - rulează scriptul "scriptName"
- > reset.py resetează toate tabelele la forma din document și are mai mulți pași
 - > dropdb.py șterge baza de date "bonk"
 - > createdb.py crează baza de date "bonk"
 - > createtables.py crează cele 4 tabele după structură
 - > populate_tableName.py inserează intrările default

Astfel pot să editez baza de date, și în caz că devine neutilizabilă o pot aduce la forma inițială.

- > st.py afișează tabele primite ca argumente

Eg. python st.py stock suppliers va afișa tabelele stock și suppliers

- > nc.py execută toate interogările una după cealaltă

Utilizare in UI:

- > lansat prin comanda "python launchqt.py" in terminal (asum că Python este adăugat în Path la instalare)
- > utilizatorul va fi anunțat in UI sau în terminal de fiecare dată când trebuie să introducă date și va primi un răspuns în terminal

Interogări

1. Afișați stocul de GPU vândut de supplierul 1.
2. Afișați toate GPU-urile de la MSI și Nvidia.
3. Găsiți toate perechile de GPU de memorie citită de la tastatură cu brandul citit de la tastatură.
4. Sortați supplieri crescător după reducerea la 4 plăci, apoi descrescător după nume.
5. Inserați un brand cu toate datele citite de la tastatură.
Inserați un GPU cu toate datele citite de la tastatură, de la brandul inserat anterior.
Inserați GPU-ul în stoc cu toate datele citite de la tastatură.
6. Calculați media prețurilor GPU-urilor fără atributul deadstock.
7. Găsiți GPU-ul cel mai ieftin cu memoria și brandul citit de la tastatură, cu atributul deadstock.
- 8 Găsiți cel mai ieftin și cel mai scump GPU vândut de un supplier cu numele citit de la tastatură, afișați detalii despre GPU si calculați-le prețul total, după aplicarea reducerii.
9. Găsiți toți supplieri (nume) care vând GPU-urile de la brandul citit de la tastatură.
10. Găsiți de la care alt supplier (nume) își ia GPU-uri supplierul cu numele citit de la tastatură.