

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO – MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Projektni zadatak – Poslovna inteligencija

Amazon Redshift – skladište podataka u oblaku

Mentor: dr. sc. Ognjen Orel, v. pred.

Dominik Horvat

4. studenog 2024.

Sadržaj

Uvod	1
Općenito	1
Arhitektura	1
Baze podataka	3
Columnar storage	3
Povezivanje s AWS uslugama	5
Amazon S3	5
Klasteri.....	7
Amazon Redshift Serverless	8
Query editor v2	11
Sintaksa SQL naredbi	12
Testno skladište podataka	13
Izrada skladišta podataka	14
Upiti na skladište i vizualizacija	16
Načini plaćanja i usluge	19
Zaključak	20
Literatura	22

Uvod

U ovom projektu bit će predstavljena usluga Amazon Web Servicesa (skraćeno, AWS) pod nazivom Amazon Redshift. Kroz rad bit će obrađena arhitektura, pohrana podataka i povezivanje s ostalim uslugama kako bi se upotpunili zahtjevi koji prethode izradi skladišta i načini pristupa stvaranju skladišta podataka. Na kraju rada prikazat će se dva upita na izgrađeno skladište putem Redshift uređivača uz popratnu vizualizaciju istih. Kroz ovaj projektni rad trebao bi se steći dojam koliko Amazon Redshift nudi mogućnosti za skladišta podataka u oblaku.

Općenito

Amazon Redshift je usluga koja omogućuje upravljanje skladištem podataka veličine petabajta (PB, 1PB = 1024TB) u oblaku. Izgrađen je oko industrijskog standarda SQL-a, s dodanom funkcionalnošću za upravljanje vrlo velikim skupovima podataka. Podržava analizu visokih performansi i izvješćivanje podataka. Dakle, bez obzira na veličinu skupa podataka, Amazon Redshift omogućuje upite brzih izvedbi služeći se alatima temeljenim na SQL-u te integriranje s aplikacijama poslovne inteligencije koje se danas koriste. Amazon Redshift postiže učinkovitu pohranu i optimalnu izvedbu upita kroz kombinaciju masivne paralelne obrade, pohrane podataka u stupcima (eng. *Columnar storage*) i vrlo učinkovite sheme kodiranja ciljane kompresije podataka. Bitno je istaknuti kako se radi o OLAP (*On-line Analytical Processing*) stilu. Kroz iduću točku bit će opisana arhitektura Amazon Redshifta s ciljem boljeg razumijevanja komponenti koje čine ovo skladište podataka u oblaku.

Arhitektura

Razumijevanje osnovnih komponenti arhitekture skladišta podataka u Amazon Redshiftu pomaže u oblikovanju upita i tablica za optimalnu izvedbu. Skladište podataka u Amazon Redshiftu sastoji se od sljedećih osnovnih komponenti: klastera, vodećeg čvora, računalnih čvorova, odsječka čvorova, masivne paralelne obrade, klijentskih aplikacije.

Klasteri se sastoje od jednog ili više računalnih čvorova te jednog vodećeg čvora. Ovisno o vrsti čvora, broj čvorova može varirati između 1 i 128. Čvorovi čine osnovnu infrastrukturu komponenata Amazon Redshift skladišta podataka. Klijentska aplikacija izravno komunicira samo s vodećim čvorom, a računalni čvorovi su transparentni za vanjske aplikacije.

Vodeći čvor (eng. *Leader node*) koordinira komunikaciju za klijentske programe i sve računalne čvorove te je odgovoran za generiranje planova izvršenja upita. On priprema planove za izvršenje upita kad god se upit pošalje klasteru. Vodeći čvor u trenutku konačne spremnosti upita kompilira kod, distribuira kompilirani kod računalnim čvorovima i dodjeljuje dijelove podatka svakom računalnom čvoru za obradu rezultata upita.

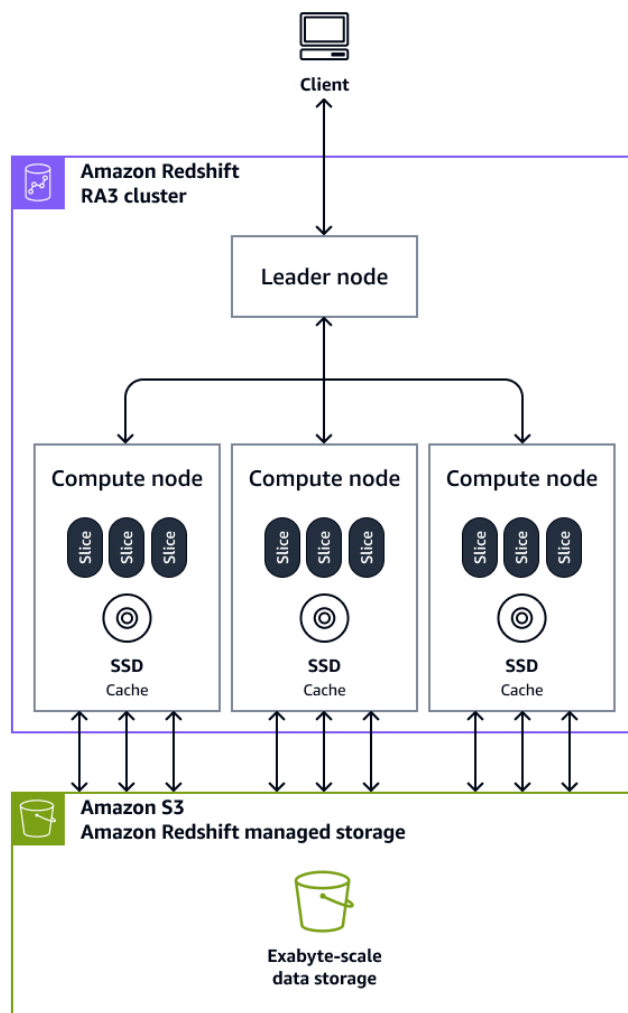
Računalni čvor (eng. *Computer node*) služi za pokretanje kompiliranog koda. Pri završetku obrade računalni čvor šalje međurezultat natrag u vodeći čvor zbog konačnog spajanja svih rezultata u jedan. Svaki od njih ima svoju središnju jedinicu za obradu (eng. *Central processing unit, CPU*), memoriju i priključenu diskovnu pohranu.

Računalni čvor podijeljen je u jedinice koje se nazivaju odsječci. Na svaki odsječak raspoređuje se određen dio memorije čvora i prostora na disku koji služi za obrađivanje djela radnog

opterećenja koji je dodijeljen čvoru. Također, omogućen je paralelan rad odsječaka. Ovdje se podaci uspoređuju na temelju njihove distribucije i ključa distribucije koji je određen tablicom. Upravo distribucija, štoviše ravnomjerna distribucija podataka omogućuje Redshiftu ravnomjerno dodjeljivanje radnih opterećenja odsječcima. Tako se povećava i paralelnost u radu.

Masivna paralelna obrada (eng. *Massively parallel processing, MPP*) u Amazon Redshiftu omogućuje brzu obradu podataka i složenih upita. Više računalnih čvorova pokreće isti kod upita na dijelovima podataka kako bi se povećala paralelna obrada.

Ono što je bitno za napomenuti jest da se Amazon Redshift integrira s alatima za izdvajanje, transformaciju i učitavanje podataka (eng. *extract, transform, load – ETL*). Ostale integracije su s alatima za poslovnu inteligenciju, rudarenje podataka i analitički alati. Sve klijentske aplikacije (eng. *Client application*) provode komunikaciju s klasterom samo kroz već navedeni vodeći čvor.



Slika 1. Komponente arhitekture

Na slici 1 prikazan je dijagram komponenata navedene arhitekture Amazon Redshifta koje rade zajedno s ciljem ubrzanja upita. Na toj slici je prikazan i Amazon S3, ali o tome nešto više u kasnijem dijelu projektnog rada.

Važno je napomenuti kako postoji sedam faza životnog ciklusa upita:

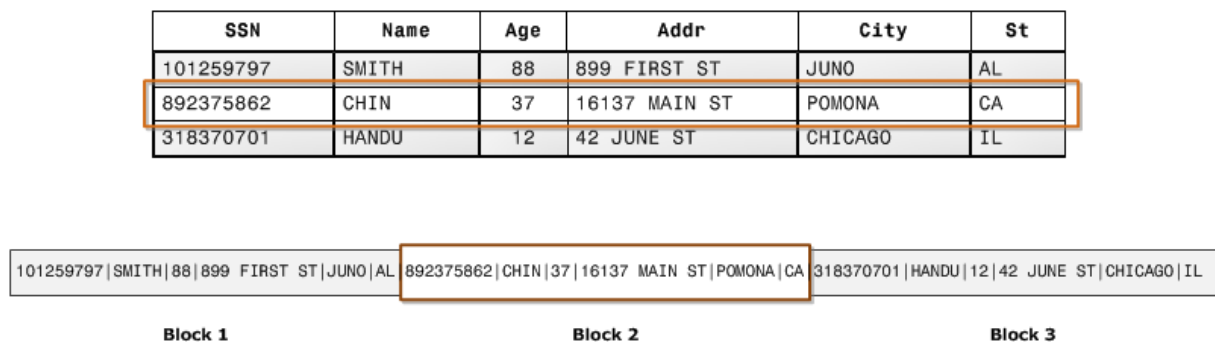
1. Prihvat i raščlanjivanje upita
2. Optimizacija upita
3. Generiranje plana upita
4. Prevođenje upita
5. Paralelno izvođenje
6. Obrada streama
7. Konačno razvrstavanje i povezivanje

Baze podataka

Ključna komponenta svakog skladišta podataka u većini su same baze podataka, najčešće relacijske baze podataka koje služe za kasnije oblikovanje dimenzijskih i činjeničnih tablica. Tako je i ovdje bitno spomenuti njihovu ulogu i način na koji Amazon Redshift upravlja istim. Općenito, klaster sadrži jednu ili više baza podataka. S druge strane SQL klijent komunicira s vodećim čvorom koji zauzvrat koordinira izvođenjem upita s računalnim čvorovima. Amazon Redshift je ujedno i sustav za upravljanje relacijskim bazama podataka (eng. *relational database management system*, skraćeno *RDBMS*). Ovo omogućuje kompatibilnost Redshifta s drugim RDBMS aplikacijama te pruža istu funkcionalnost kao tipični RDBMS, uključujući funkcije mrežne obrade transakcija (eng. *On-line Transactional Processing*, skraćeno *OLTP*) kao što su umetanje i brisanje podataka. Iako se Amazon Redshift temelji na PostgreSQL-u, Amazon Redshift i PostgreSQL imaju niz vrlo važnih razlika koje se moraju uzeti u obzir pri dizajnu i razvijanju skladišta podataka. U nastavku bit će nešto više riječi o stupčastom skladištenju podataka po kojem se i Redshift razlikuje od ostalih usluga i servisa u oblaku.

Columnar storage

Najveća razlika Amazon Redshifta u odnosu na druge usluge skladištenja podataka u oblaku je stupčasta struktura pohrane podataka (eng. *columnar storage*). Ovo je učinkovita metoda za pohranjivanje tabličnih podataka. Ovakav način pohrane izrazito je važan čimbenik u optimizaciji performansi analitičkih upita jer drastično smanjuje I/O zahtjeve diska. Ukratko, Input/Output zahtjevi diska odnose se na operacije čitanja i pisanja podataka na disk, koje računalo ili aplikacije traže kako bi obradili podatke pohranjene na fizičkom uređaju za pohranu poput hard diska ili SSD-a. Samom pohranom podataka u obliku stupčaste pohrane smanjuje se količina podataka koja se treba učitati s diska.



Slika 2. Pohranjivanje u blokove diska po redu

Priložena slika 2 iznad prikazuje kako se zapisi iz tablica baze podataka obično pohranjuju u blokove diska po redu.

U klasičnoj tablici relacijske baze podataka svaki red sadrži vrijednost polja za jedan zapis. Kroz takav način pohrane prema redovima, podatkovni blokovi sadrže uzastopne stupce za svaki red. Ako je veličina bloka veća od veličine zapisa, pohrana cijelog zapisa može trajati manje od jednog bloka. To rezultira neučinkovitosti u korištenju prostora na disku. Ovakav način pohranjivanja prema redovima optimalan je za OLTP baze podataka.

Slika 3 prikazuje kako se pri pohranjivanju u stupcima vrijednosti za svaki stupac sekvencijalno pohranjuju u blokove diskova.

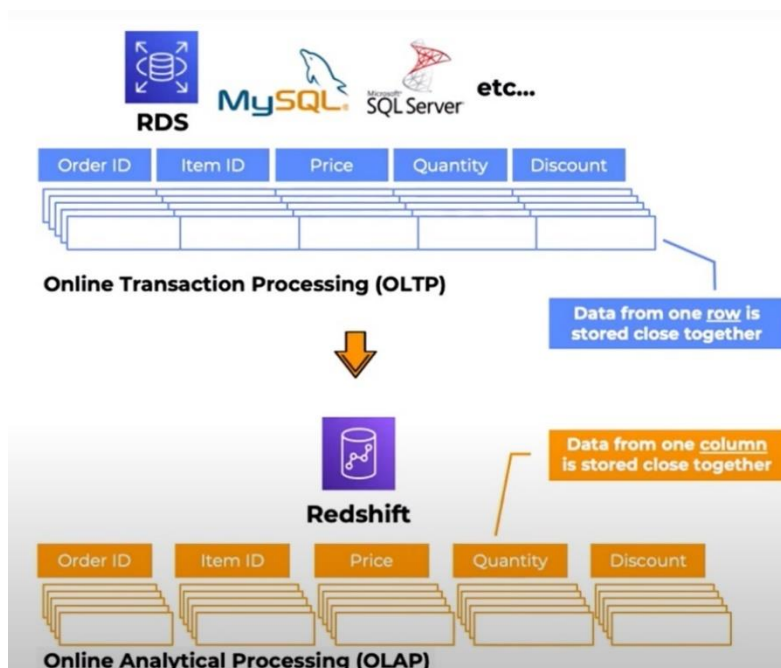
SSN	Name	Age	Addr	City	St
101259797	SMITH	88	899 FIRST ST	JUNO	AL
892375862	CHIN	37	16137 MAIN ST	POMONA	CA
318370701	HANDU	12	42 JUNE ST	CHICAGO	IL

101259797	892375862	318370701	468248180	378568310	231346875	317346551	770336528	277332171	455124598	735885647	387586301
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Block 1

Slika 3. Pohrana u stupcima

Dakle, uporabom pohrane u stupcima svaki blok podataka pohranjuje vrijednosti jednog stupca za više redaka. Moramo imati u vidu da kako zapisi ulaze u sustav, Amazon Redshift transparentno pretvara podatke u stupčastu pohranu za svaki stupac. Primjerice, za svaki „CREATE TABLE“ upit u Amazon Redshiftu automatski se koristi pohrana u stupcima, bez obzira na jednostavnost definicije tablice. Čim tablicu popunimo podacima iz S3 bucketa (o kojem će biti nekoliko riječi kasnije), Redshift će ih pohraniti u stupčastom formatu.



Slika 4. OLTP -> OLAP

Zahvaljujući pohrani u stupcima, svaki podatkovni blok može sadržavati vrijednosti polja stupca za čak tri puta više zapisa u odnosu na pohranu temeljenu na redcima. To znači da čitanje istog broja vrijednosti polja stupaca za isti broj zapisa zahtijeva trećinu I/O operacija u odnosu

na pohranom u redcima. U stvarnoj primjeni, ako baratamo s tablicama koje posjeduju veliki broj stupaca i veliki broj redaka, učinkovitost ovakvog načina pohrane je još veća. Pojavljuje se i tehnika pod nazivom kodiranje kompresije (eng. *Compression encodings*) koja se primjenjuje na podatke kako bi se smanjila njihova veličina pri pohrani. Ova tehnika doprinosi smanjenju troškove pohrane i ubrzanju pristup podacima.

Ušteda prostora za pohranu podataka također se prenosi na pohranjivanje podataka u memoriji. Budući da mnoge operacije baze podataka zahtijevaju pristup samo jednom ili manjem broju stupaca, moguće je uštedjeti memorijski prostor dohvaćanjem samo onih blokova koji su stvarno potrebni za izvršavanje upita. Nasuprot tome, OLTP transakcije često koriste većinu ili sve stupce za manji broj zapisa, dok upiti u skladištima podataka uglavnom pristupaju samo nekim stupcima, ali za velik broj redaka. Drugim riječima, za isti broj redaka, čitanje vrijednosti polja stupaca zahtijeva znatno manje I/O operacija. U praksi, primjena tablica s izrazito velikim brojem stupaca i redaka rezultira proporcionalno većim povećanjem učinkovitosti. Amazon Redshift koristi veličinu bloka od 1MB što je učinkovitije i dodatno smanjuje broj I/O zahtjeva potrebnih za izvođenje učitavanja baze podataka ili drugih operacija u sklopu upita. Važno je naglasiti da općenita veličina blokova baze podataka je između 2KB i 32KB. Ukratko rečeno, pohrana po stupcima i OLAP omogućuju tri put bolje performanse za razliku od ostalih skladišta podataka u oblaku.

Povezivanje s AWS uslugama

Amazon Redshift omogućuje integraciju s ostalim AWS (skraćeno od *Amazon Web Services*) uslugama kako bi omogućio brzo i pouzdano premještanje, transformaciju i učitavanje podataka te koristi značajke sigurnosti podataka. Neke od AWS usluga su Amazon S3, DynamoDB, AWS Data Pipeline, AWS DMS i mnoge druge. U ovom projektu dotaknut ćemo se najviše Amazon S3 usluge koja nam omogućuje stvaranje jezera podataka (eng. *data lake*) za naše skladište podataka. Prije nego što krenemo u obradu usluge Amazon S3 bitno je naglasiti kako neke značajke Amazon Redshifta zahtijevaju Amazon Redshift za pristup drugim AWS uslugama u naše ime. Kao dobar primjer imamo sljedeće naredbe:

- COPY i UNLOAD mogu učitati ili ukloniti podatke u Amazon Redshift klasteru pomoću Amazon S3 spremnika.
- CREATE EXTERNAL FUNCTION može pozvati AWS Lambda funkciju pomoću skalarnih Lambda korisnički definirane funkcije.

Kako bi upravljali Amazon Redshift klasterima u naše ime trebamo definirati sigurnosne identifikacijske podatke. Općenito to ostvarujemo kroz „IAM role“ (*Identity and Access Management roles*) dozvole. One definiraju što određeni korisnici ili usluge mogu ili ne mogu raditi unutar Redshift klastera, ali i Amazon Redshift Serverlessa. Time omogućujemo administraciju pristupa resursima AWS-a i integraciju s drugim AWS uslugama na siguran i kontroliran način. Njihovo kreiranje je vrlo jednostavno s mogućnošću postavljanja raznih modifikacija pristupa i ovlasti.

Amazon S3

Općenito u procesu izrade skladišta podataka potrebni su prethodno izvori podataka i jezero podataka gdje se prikupe svi podaci iz izvora. Usluga Amazon S3 nam omogućuje pohranu objekata uz mnoge prednosti kao što su skalabilnost, trajnost i dostupnost, sigurnost i zaštita podataka, najniža cijena i najbolje performanse. Amazon S3 služi za pohranjivanje, upravljanje

i analiziranje bilo koje količine podataka za gotovo sve slučajeve uporabe, bilo to podatkovna jezera, aplikacije u oblaku ili mobilne aplikacije. Uz razne ponuđene značajke moguće je i konfigurirati kontrole pristupa podacima unutar S3 usluge. Ono što nas zanima u ovom kontekstu je već prethodno spomenuto stvaranje podatkovnog jezera – podatkovne platforme koja može sadržavati podatke namijenjene analizama iz više izvora. Kako bismo pohranili podatke u Amazon S3, radimo sa spremnicima i objektima. Kanta (eng. *bucket*) je spremnik za objekte, to jest datoteke i sve metapodatke koji opisuju tu datoteku. Postupak pohranjivanja objekata u Amazon S3 i stvaranje kante je slijedeći:

1. Otvorimo Amazonovu konzolu (uz pretpostavku da smo prethodno stvorili korisnički račun ili posjedujemo isti).
2. U tražilicu koja se nalazi gore lijevo napišemo „S3“ i odaberemo u ponuđenom izborniku odgovarajuću uslugu.
3. Pod opcijama koje su ponuđene lijevo u listi odaberemo „Buckets“ i odaberemo „Create bucket“.
4. U „General configuration“ ostavimo sve kako je trenutno namješteno, samo trebamo upisati ime naše kante (*bucket*) u odgovarajuće polje (slika 5), ali tako da ono bude jedinstveno na globalnoj razini. Ostale postavke možemo ostaviti onako kako je zadano.

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type [Info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

neki-naziv-koji-ne-postoji

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

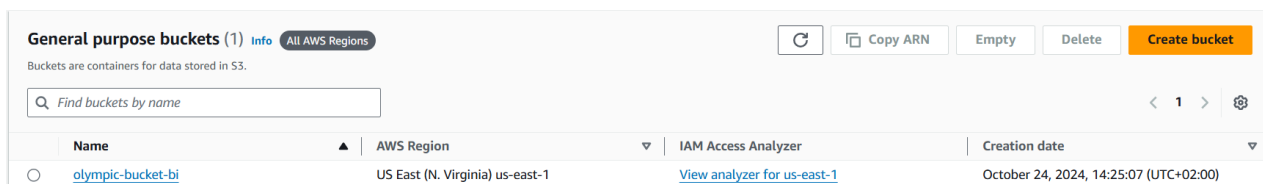
Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

Slika 5. Stvaranje kante

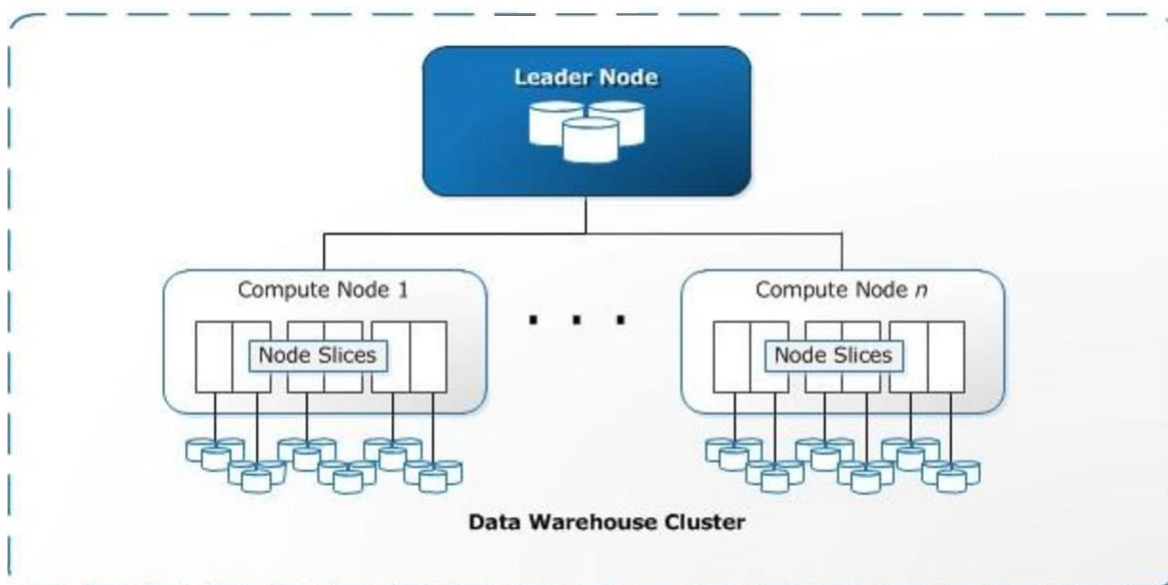
5. Nakon ovoga naš kanta (*bucket*) bit će kreiran (slika 6) i možemo kliknuti na njega. Odabirom opcije prenesi (eng. *upload*) prenosimo s našeg računala odgovarajuće datoteke koje želimo kasnije koristiti u punjenju naših tablica.



Slika 6. Prikaz postojeće kante

Klasteri

Amazon Redshift skladište podataka je skup računalnih resursa (čvorova) koji su organizirani u grupu zvanu klaster. Svaki klaster pokreće Amazon Redshift mehanizam i sadrži jednu ili više baza podataka. Broj čvorova u klasteru ovisi o veličini naših podataka, broja upita koje pokrećemo i performansama vremena izvođenja potrebnih da se izvrši željeni upit. U točki o *Arhitektura* navedeno je kako broj čvorova može varirati između 1 i 128 ovisno o vrsti čvora. Ako se odlučimo pri stvaranju klastera staviti broj čvorova na 1, time će vodeći čvor preuzeti uloge računalnih čvorova. Sasvim je u redu pitati se kako se to onda odražava na performanse upita i optimizaciju. Jasno je kako odabirom 1 čvora gubimo i paralelnost u obradi jer vodeći čvor nema dodatne računalne čvorove. U suštini klaster s manjim brojem čvorova je prikladan za male skupove podataka i svrhe raznih testiranja, a klasteri s velikim brojem čvorova za veće skupove podataka i proizvodna okruženja.



Slika 7. Prikaz klastera

Postupak stvaranja klastera u Amazon Redshiftu je sljedeći (podrazumijeva se da klaster niti opcija Amazon Redshift Serverless nisu još stvoreni):

1. Kroz tražilicu u Amazon konzoli pristupimo usluzi „Amazon Redshift“. Ako još nismo stvorili klaster ili pristupili Amazon Redshift Serverlessu s desne strane dobivamo opciju „Create cluster“
2. U formi „Cluster configuration“ pridodajemo ime našem klasteru te odabiremo vrstu čvora i broj čvorova. Također, postoji i opcija „Help me choose“ koja otvara dodatne mogućnosti odabira čvora ovisno o našim potrebama i podacima. U padajućem izborniku „Node type“ postoji nekoliko vrsta čvorova uz dodatne opise (vidi sliku 8).

Odabirom vrste čvora podešavamo i željeni broj čvorova koje želimo da klaster posjeduje. Na kraju ove forme dobivamo cijenu koju plaćamo za korištenje ovih čvorova (cijena po mjesecu, cijena pohrane – koja može biti navedena, ali i ne mora) i maksimalan kapacitet komprimirane pohrane.

Node type [Info](#)
Choose a node type that meets your CPU, RAM, storage capacity, and drive type requirements.

ra3.4xlarge

Q

ra3.large

Managed storage: up to 8 TB/node
\$0.543/node/hour \$0.024/GB/month 2 vCPU (gen 3)

ra3.xlplus

Managed storage: up to 4 TB for single-node clusters. For clusters with 2 or more nodes, each node has storage up to 32 TB.
\$1.086/node/hour \$0.024/GB/month 4 vCPU (gen 3)

ra3.4xlarge

Managed storage: up to 128 TB/node
\$3.26/node/hour \$0.024/GB/month 12 vCPU (gen 3)

ra3.16xlarge

Managed storage: up to 128 TB/node
\$13.04/node/hour \$0.024/GB/month 48 vCPU (gen 3)

DC2

High performance with fixed local SSD storage

dc2.large

Storage: 160 GB/node
\$0.25/node/hour 2 vCPU (gen 2)

dc2.8xlarge

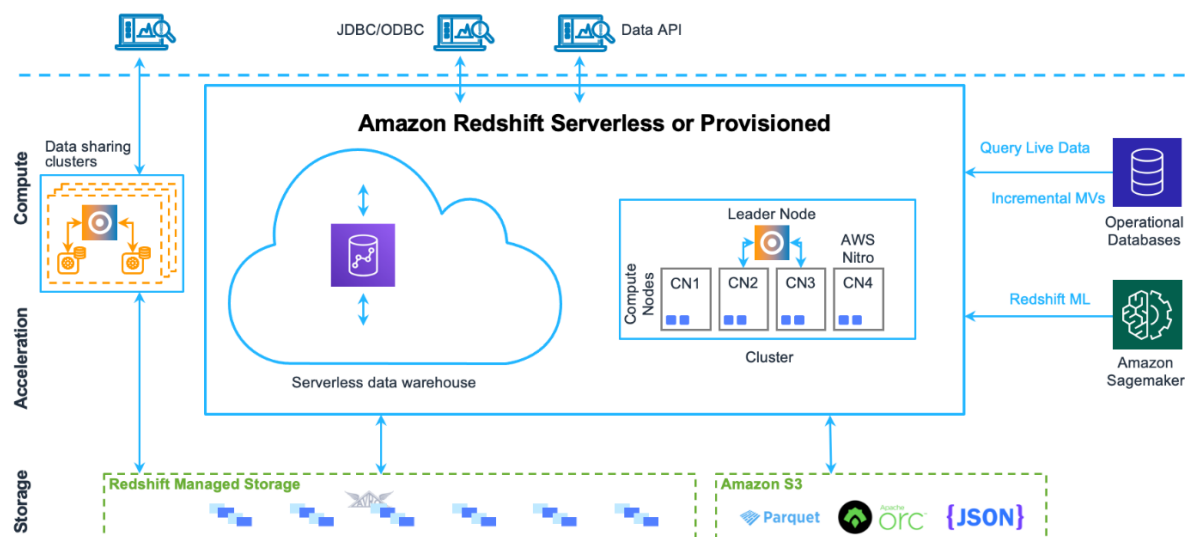
Storage: 2.6 TB/node
\$4.80/node/hour 32 vCPU (gen 2)

Slika 8. Vrste čvorova u klasteru

3. Iduća opcija je učitavanje uzorka podataka koji bi omogućio osnovne upite na već pripremljenim podacima, ovaj potvrdni okvir nije nužno označiti
4. U formi „Database configurations“ pridodajemo podatke za prijavu u našu bazu podataka te možemo odabrati hoće li naša baza imati enkripciju
5. U „Cluster permissions“ postavljamo IAM role o kojoj je bila riječ u podnaslovu „Povezivanje s AWS uslugama“ te ovdje uočavamo kako „IAM role“ povezujemo s dopuštenjima pristupa postojećim S3 bucketima.
6. Ostale konfiguracije možemo koristiti kao što su i zadane
7. Za kraj kliknemo na gumb „Create cluster“ i time je naš klaster stvoren

Amazon Redshift Serverless

Koncept Amazon Redshift Serverless poprilično je novi pojam u pružanju usluga Amazon Redshifta. Odnosi se na arhitekturu bez poslužitelja kao što i sam naziv govori. Ovakva arhitektura je način za izgradnju i pokretanje aplikacija i usluga bez potrebe za upravljanje infrastrukturom.

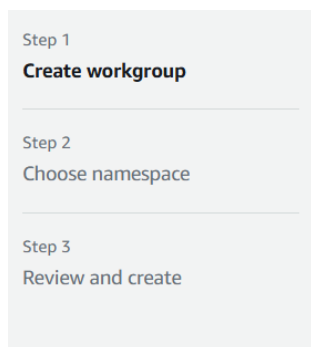


Slika 9. Amazon Redshift

Dakle, sada se uz klastere javlja i „Serverless data warehouse“ kao što je prikazano na slici 9. Rezultat toga su dva načina na koje možemo manipulirati skladištem podataka u oblaku.

Amazon Redshift Serverless omogućuje brze rezultate upita za sve podatke preko Amazon Web Services (AWS) u poznatom SQL-u bez potrebe za razmišljanjem o infrastrukturi skladišta podataka. Također, automatski se osigurava odgovarajući kapacitet za sve nepredvidive potrebe. Omogućeno je jednostavno učitavanje naših podataka na način koji je već prethodno naveden kroz projektni rad. Nije izostavljeno niti povezivanje alata koji se koriste za poslovnu inteligenciju. Za razliku od klastera, u Amazon Redshift Serverlessu plaćamo korištenje po sekundi. Dakle, ako je skladište podataka u stanju mirovanja ne plaća se ništa, a uz to postoji i opcija postavljanja ograničenja potrošnje tako da se nikada ne prekorači proračun.

Pristup Amazon Redshift Serverlessu ostvarujem kroz tražilicu Amazon konzole tako da prvo pristupimo usluzi „Amazon Redshift“. S desne strane ponuđen je pristup Redshift Serverlessu i stvaranje Redshift klastera. Ovog puta odabiremo pristup Redshift Serverless resursima. U slučaju da nikada prethodno nismo koristili Redshift Serverless i imamo novi račun unutar posljednjih 12 mjeseci, ta opcija bit će ponuđena pod nazivom „Try Redshift Serverless Free Trial“. Zatim se pojavljuju tri koraka koja je potrebno izvršiti kako bi uspješno koristili Redshift Serverless (slika 10).



Slika 10. Prikaz koraka u pristupu Redshift Serverlessu

U prvom koraku stvaramo radnu grupu (eng. *workspace*). Ovo označava skup računalnih resursa koje Redshift Serverless koristi za izvršavanje računalnih zadataka. Izrazito je bitno pridodati naziv našoj radnoj grupi i odabrati broj RPU-ova (*Real-time Processing Unit*).

Workgroup

Workgroup is a collection of compute resources from which an endpoint is created. Compute properties include network and security settings.

Workgroup name

This is a unique name that defines the workgroup.

The name must be from 3-64 characters. Valid characters are a-z (lowercase only), 0-9 (numbers), and - (hyphen).

Performance and cost controls [Info](#)

Set a base capacity to indicate the base amount of Redshift processing units (RPUs) that Amazon Redshift can use to run queries. Alternatively, set price-performance target to optimize resources. Amazon Redshift uses AI-driven scaling and optimization to automatically adjust your resources when running queries.

Slika 11. Naziv radne grupe i izvedba

Prema slici 11 jasno je naslutiti kako veći broj RPU-ova pridodaje i većoj cijeni korištenja. Ovisno o našim potrebama i željama za boljom izvedbom zadataka skaliramo broj RPU-ova od 8 do 1024. Ostale stvari možemo ostaviti kao što su i zadane i preći na drugi korak (slika 10). Kroz ovaj korak dajemo ime „skladištu podataka“, to jest imeničkom prostoru (eng. *namespace*). Detaljnije, imenički prostor je kolekcija objekata baze podataka i korisnika. Imenički prostori grupiraju sve resurse koje koristimo u Redshift Serverlessu.

Choose namespace

Namespace

Namespace is a collection of database objects and users. Data properties include database name and password, permissions, and encryption and security.

☒ Create a new namespace

☐ Add to an existing namespace

Namespace

This is a unique name that defines the namespace.

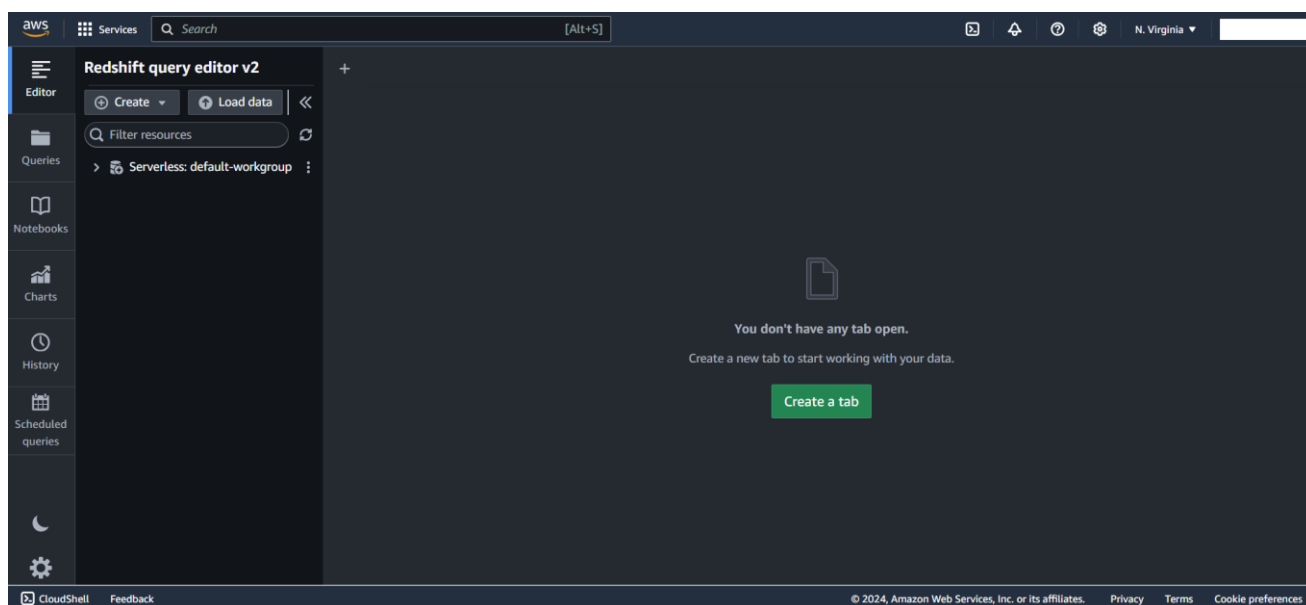
The name must be from 3-64 characters. Valid characters are a-z (lowercase only), 0-9 (numbers), and - (hyphen).

Slika 12. Stvaranje imeničkog prostora (eng. *namespace*)

Ostale postavke poput postavljanja administracije, pristupa i IAM role identične su kao i kod stvaranja klastera. U zadnjem, to jest trećem koraku dobijemo prikaz prva dva koraka, ali ovaj put onako kako smo ispunili u prva dva koraka i mogućnostima za povratak na prethodne korake ili potvrdu stvaranja radne grupe (eng. *workspace*).

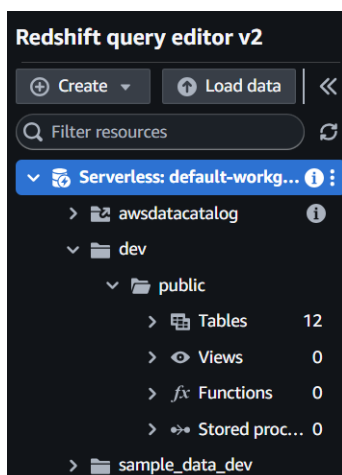
Query editor v2

Nakon odluke o tome što ćemo koristiti za manipulaciju skladištem podataka, bilo to Redshift klaster ili Redshift Serverless možemo pristupiti uređivaču za upite, upravljanje skladištem i mnoge druge namjene. Trenutno je to Amazon Query editor v2 u Redshiftu. Query editor v2 je zasebna web bazirana SQL klijentska aplikacija koja se koristi za izradu i pokretanje upita na Amazon Redshift skladištu podataka. U hrpi opcija Query editor v2 ima glavne namjene poput uređenja i pokretanja upita, vizualizacije rezultata i dijeljenje rada s timom ukoliko se radi o grupnim projektima. Neovisno nalazimo li se u radnoj grupi ili klasteru prikazana je opcija pod nazivom „Query data“. Nakon klika na tu opciju otvara se uređivač kao što je prikazano na slici 13.



Slika 13. Query editor v2

Kroz ovaj projektni zadatak koristio se Redshift Serverless tako da s lijeve strane slike 13 u *Editoru* možemo vidjeti trenutne radne grupe kojima možemo pristupiti kroz uređivač. Kako bi došli do naših tablica proširujemo prikaz (slika 14).



Slika 14 prikaz sadržaja Redshift Serverlessa

Sve stvorene tablice nalaze se unutar baze nazvane „dev“ (naziv se može dodijeliti prilikom stvaranja radne grupe) u shemi „public“ koja sadrži „Tables“. Pored samog naziva gdje su pohranjene sve tablice prikazan je i broj postojećih tablica.

Sintaksa SQL naredbi

Prije nego što krenemo opisivati izradu skladišta podatka i izvršavati upite uz popratnu vizualizaciju istih, bitno je prikazati oblik sintakse SQL naredbi u Amazon Redshiftu. Ukratko, opisujemo za naredbu *CREATE TABLE* na primjeru stvaranja jedne tablice za relacijsku bazu Olimpijskih igara. Možemo se pridržavati pisanja sintaksi naredbi SQL kao za PostgreSQL, ali to nije ono što Redshift odjeljuje od drugih servisa. Ako želimo pospješiti brzinu naših upita i izvedbu koristimo sintaksu za Redshift. Upravo to je ono što je navedeno u točki *Baze podataka* ovog projektnog rada.

Klasično stvaranje tablice *CREATE TABLE* naredbom:

```
1 CREATE TABLE IF NOT EXISTS games (  
2     games_id      INT PRIMARY KEY,  
3     games_year    INT,  
4     games_season  CHAR(255),  
5     city_id       INT,  
6     games_name    VARCHAR(255),  
7     CONSTRAINT fk_city_games FOREIGN KEY (city_id) REFERENCES city(city_id)  
8 );
```

Slika 15.

U skladu sa sintaksom Amazon Redshifta:

```
50 --7.GAMES  
51 CREATE TABLE IF NOT EXISTS games (  
52     games_id INT PRIMARY KEY,  
53     games_year INT ENCODE zstd,  
54     games_season CHAR(255) ENCODE zstd,  
55     city_id INT REFERENCES city(city_id),  
56     games_name VARCHAR(255) ENCODE zstd  
57 )  
58 DISTSTYLE EVEN  
59 SORTKEY(games_id);
```

Slika 16. Sintaksa namijenjena Redshiftu (pr.1.)

```

30  --5.ATHLETE
31  CREATE TABLE IF NOT EXISTS athlete (
32      athlete_id INT PRIMARY KEY,
33      athlete_name VARCHAR(255) ENCODE zstd,
34      athlete_gender CHAR(1) ENCODE bytedict,
35      athlete_height INT,
36      athlete_weight INT,
37      athlete_yob INT
38  )
39  DISTSTYLE KEY
40  DISTKEY(athlete_id)
41  SORTKEY(athlete_id);

```

Slika 17. Sintaksa namijenjena Redshiftu (pr.2.)

Objašnjenje:

- ENCODE zstd/bytedict – koristi se za kompresiju, *zstd* je preporučen za stupce s tekstualnim ili numeričkim podacima, dok *bytedict* bolje radi s niskim brojem različitih vrijednosti (npr. M i F za athlete_gender)
- DISTSTYLE KEY/EVEN – označava način distribucije podataka, *EVEN* označava ravnomjernost distribucije redova po čvorovima, a *KEY* da će distribuciju podataka biti temeljena na određenom ključu
- DISTKEY(...) – određuje prema kojem ključu se distribuiraju podaci
- SORTKEY(...) – postavlja ključ za sortiranje podataka

Za sve ostale naredbe može se pogledati u *Literatura* pod [2].

Testno skladište podataka

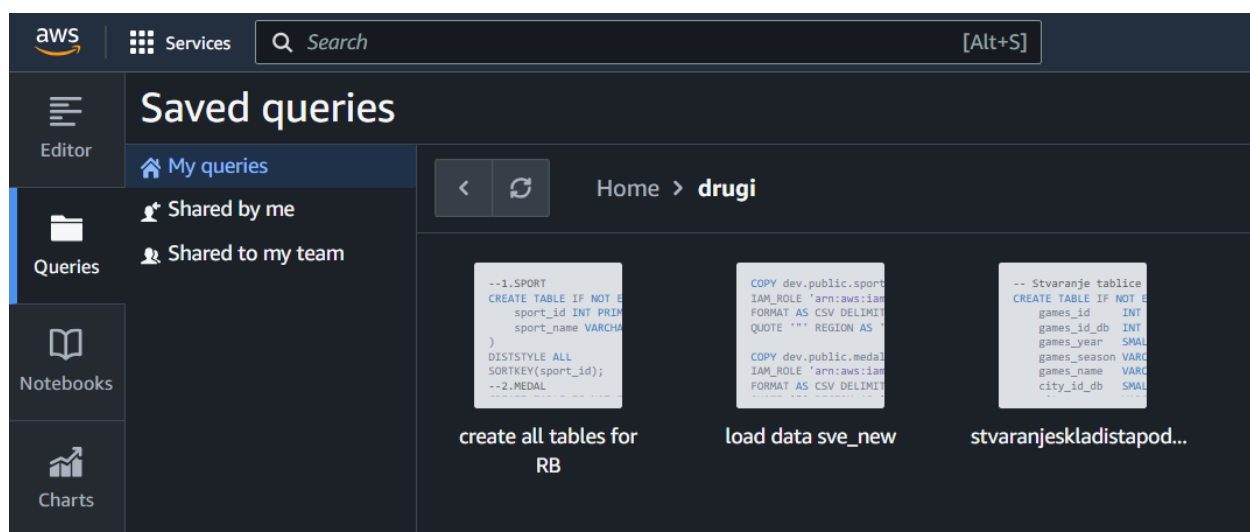
Skladište podataka temeljeno je na podacima o sudjelovanju sportaša na olimpijskim igrama, do 2016. godine. Baza koja se koristila je dostupna u sklopu predmeta Poslovna inteligencija na platformi Merlin (akademska godina 2024/2025). Za tablice u ovoj bazi podataka može se vidjeti u narednoj točki. Kroz konzultacije s nastavnikom modeliralo se skladište na način da su stvorene sljedeće tablice:

- Dimenzijska:
 - **d_games** s atributima: games_id, games_id_db, games_year, games_season, games_name, city_id_db, city_name, noc_id_db, region, total_medals
 - **d_athlete** s atributima: athlete_id, athlete_id_db, athlete_name, athlete_gender, athlete_height, athlete_weight, athlete_yob, athlete_noc_id_db, athlete_noc_region
 - **d_event** s atributima: event_id, event_id_db, event_name, sport_id_db, sport_name
- Činjenična:
 - **f_participation** s atributima: athlete_id, event_id, games_id, athlete_age, home_field, medal, gold, silver, bronze

Kao što se može zaključiti skladište podataka ima zvjezdastu shemu.

Izrada skladišta podataka

Za početak stvaramo tablice kako bi prvobitno napravili relacijsku bazu podataka iz koje izgrađujemo skladište podataka. Relacijska baza koju izgrađujemo odnosi se na već prethodno navedene olimpijske igre, do 2016. godine. U tu svrhu stvaramo tablice s nazivima: athlete, athlete_event, city, event, games, medal, noc i sport. Prva mogućnost je stvaranje tablice tako da otvorimo novi prozor u uređivaču i napravimo odgovarajuću CREATE TABLE, i tako za svaku zasebno. Ili otvorimo jedan prozor i napišemo sve CREATE TABLE naredbe smislenim redoslijedom. Ono što je omogućeno u Redshift query editor v2 je spremanje naših naredbi u selekciju pod nazivom „Queries“, koju možemo vidjeti lijevo na vertikalnoj traci. Tamo će biti pohranjene sve naše naredbe i upiti koje spremimo (način spremanja uz kombinaciju tipki *Ctrl* + *S*).



Slika 18. Spremljeni upiti

Nakon što uspješno stvorimo sve željene tablice trebamo ih napuniti podacima. Na slici 13 možemo vidjeti opciju „Load data“ koja nam omogućuje učitavanje podataka na dva načina. Jedan preko S3 bucketa, a drugi s lokalne datoteke. Ukoliko odaberemo prvu opciju moramo izabrati i S3 bucket, a potom i datoteku iz bucketa koju želimo učitati. Preostale opcije su dosta intuitivne. Nakon što je to odrađeno prelazimo na idući korak u kojemu odabiremo popunjavanje već postojeće tablice ili stvaranje nove na temelju podataka. Ipak nama zanimljivije je korištenje *COPY* naredbe koja je prethodno navedena kod Amazon S3 usluge.


```

+ create all tables for RB x load data sve_new* x
Run Limit 100 Explain Isolated session Serverless: de... dev
11 COPY dev.public.athlete FROM 's3://olympic-bucket-bi/athlete.csv'
12 IAM_ROLE 'arn:aws:iam::account-id:role/RedshiftRole'
13 FORMAT AS CSV DELIMITER ','
14 QUOTE '"' REGION AS 'us-east-1';
15
16 COPY dev.public.athlete_event FROM 's3://olympic-bucket-bi/athlete_event.csv'
17 IAM_ROLE 'arn:aws:iam::account-id:role/RedshiftRole'
18 FORMAT AS CSV DELIMITER ','
19 QUOTE '"' REGION AS 'us-east-1';
20

```

Slika 19. Ubacivanje podataka u tablice naredbom COPY

Dakle, *COPY* naredba koristi se za učitavanje podataka u tablicu *athlete* (pomoću *dev.public.athlete*) u Amazon Redshiftu. Prvo navodimo ime tablice i kante (*bucket*) u S3 iz kojeg učitavamo našu datoteku (u ovom slučaju ona je u .csv formatu). Zatim navodimo IAM role kako bi dobili pristup podacima u S3. *FORMAT AS CSV DELIMITER ','* označava da su podaci u formatu CSV i da je zarez separator između vrijednosti u toj datoteci. *QUOTE '"'* opcija omogućuje da vrijednosti koje sadrže navodnike budu pravilno učitane, na primjer "Ivo". *REGION* definira regiju S3 bucketa kako bi dali do znanja Redshiftu gdje treba tražiti datoteku. Uz ovo je moguće i nadodati *IGNOREHEADER 1* naredbu koja omogućuje preskakanje prvog retka u slučaju da su u njemu navedeni nazivi atributa tablice. Pritiskom opcije „Run“ koja je vidljiva na slika 16 izvršavamo naredbe. Za svaku naredbu dobit ćemo zaseban rezultat je li provedena uspješno ili ne. Kao kraj ove točke prikazuje se stvaranje i popunjavanje jedne dimenzijske tablice kako bi se dobila sama intuicija u stvaranje skladišta podataka.

```

51 -- Stvaranje tablice d_athlete
52 CREATE TABLE IF NOT EXISTS d_athlete (
53     athlete_id          INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
54     athlete_id_db       INT UNIQUE ENCODE zstd,
55     athlete_name        VARCHAR(255) ENCODE zstd,
56     athlete_gender      CHAR(1) ENCODE bytedict,
57     athlete_height      SMALLINT ENCODE zstd,
58     athlete_weight      SMALLINT ENCODE zstd,
59     athlete_yob         SMALLINT ENCODE zstd,
60     athlete_noc_id_db   CHAR(3) ENCODE zstd,
61     athlete_noc_region  VARCHAR(255) ENCODE zstd
62 )
63 DISTSTYLE EVEN
64 SORTKEY(athlete_id);

```

Slika 20. Stvaranje tablice d_athlete za skladište podataka

```

66 -- Popunjavanje tablice d_athlete
67 INSERT INTO d_athlete (athlete_id_db, athlete_name, athlete_gender, athlete_height,
68 athlete_weight, athlete_yob, athlete_noc_id_db, athlete_noc_region)
69 SELECT DISTINCT a.athlete_id,
70 a.athlete_name,
71 a.athlete_gender,
72 a.athlete_height,
73 a.athlete_weight,
74 a.athlete_yob,
75 n.noc_id,
76 n.region
77 FROM athlete_event ae
78 LEFT JOIN athlete a ON ae.athlete_id = a.athlete_id
79 LEFT JOIN noc n ON ae.noc_id = n.noc_id;

```

Slika 21. Popunjavanje tablice d_athlete

Upiti na skladište i vizualizacija

Nakon što je napravljeno skladište podataka možemo izvršiti upite nad skladištem i vizualizirati rezultate kroz Redshift query editor v2.

Prvi upit na skladište:

```

1 --Top 10 najuspješnijih sportaša na ljetnim (zimskim) OI nakon 1950:
2 SELECT athlete_name, SUM(gold) as gold, SUM(silver) as silver, SUM(bronze) as bronze
3 FROM f_participation,
4 d_athlete,
5 d_games
6 WHERE f_participation.athlete_id = d_athlete.athlete_id
7 AND f_participation.games_id = d_games.games_id
8 AND games_season = 'Summer'
9 AND games_year > 1950
10 GROUP BY 1
11 ORDER BY 2 DESC, 3 DESC, 4 DESC
12 LIMIT 10;

```

Slika 22. Primjer upita na skladište (1.)

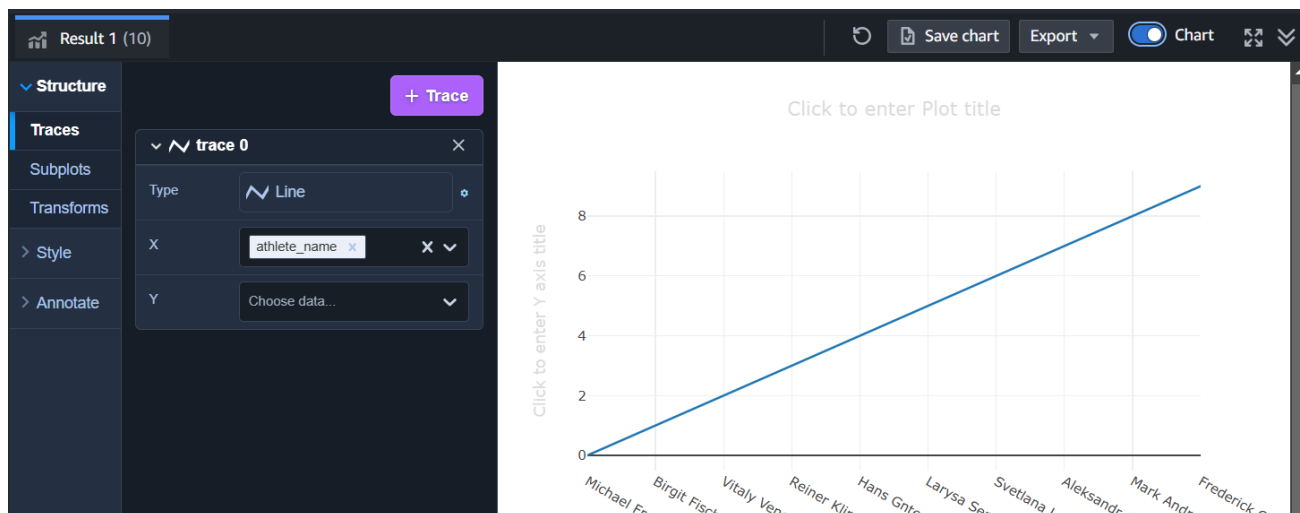
Rezultat:

athlete_name	gold	silver	bronze
Michael Fred Phelps, II	23	3	2
Birgit Fischer-Schmidt	16	8	0
Vitaly Venediktovich Shch...	12	0	8
Reiner Klimke	12	0	4
Hans Gnter Winkler	10	2	2
Larysa Semenivna Latyni...	9	5	4
Svetlana Leonidovna Bog...	9	3	3
Aleksandr Aleksandrovich...	9	3	0
Mark Andrew Spitz	9	1	1
Frederick Carlton "Carl" L...	9	1	0

Slika 23. Rezultat upita

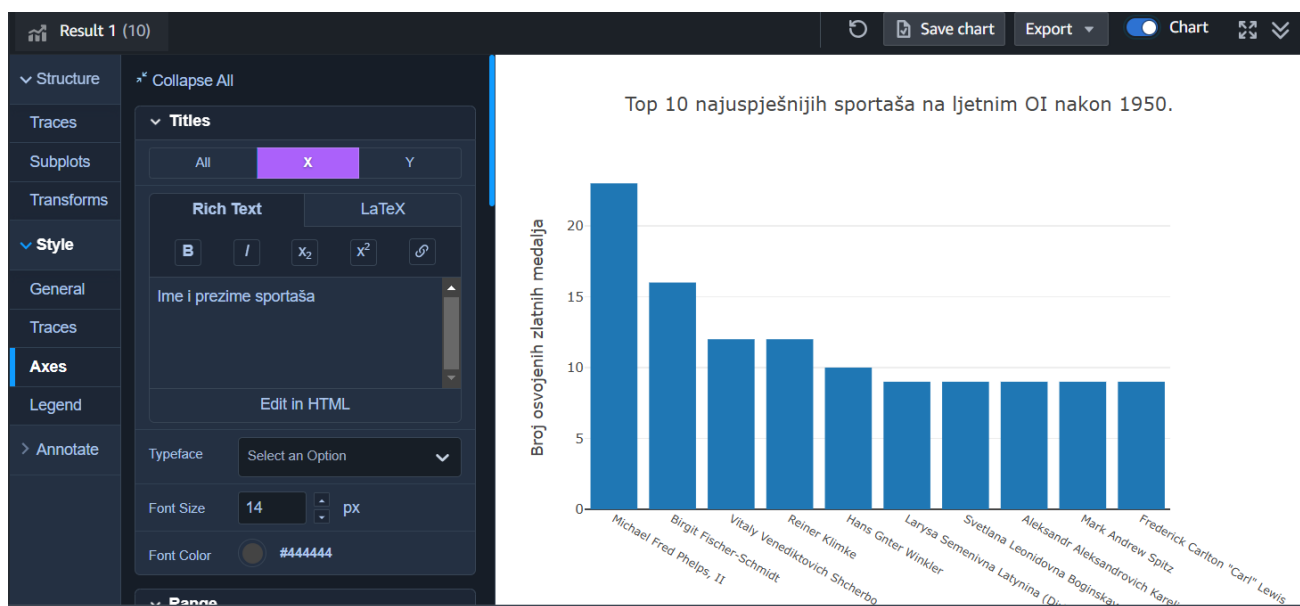
Na slici 23 možemo uočiti u gornjem desnom kutu dvije opcije, „Export“ i „Chart“. Prva opcija nam omogućuje izvoz podataka rezultata u formatima JSON ili CSV. Druga opcija nam služi za

vizualizaciju dobivenog rezultata upita. Početno ćemo dobiti vrlo neintuitivan grafički prikaz kao što je prikazano na sljedećoj slici:



Slika 24. Prvobitan prikaz rezultata

Ali uz male modifikacije dobivamo sljedeće:



Slika 25. Prikaz rezultata nakon sređivanja grafa

Također, postoji i opcija izvoza slike u formatima *JPEG* i *PNG*.

Idući upit odnosi se na istraživanje prednosti domaćeg terena – postotak medalja na pojedinim igrama koje su osvojili domaći sportaši u odnosu na ostale (ali top 10, rangiranih silazno).

```

1  --istraživanje prednosti domaćeg terena - postotak medalja na pojedinim
2  --igrama koje su osvojili domaći sportaši u odnosu na ostale top 10
3  SELECT games_name, region,
4         ((COUNT(*)::FLOAT / d_games.total_medals) * 100)::NUMERIC(3, 1)
5  FROM f_participation,
6       d_games
7  WHERE f_participation.games_id = d_games.games_id
8         AND home_field = 1
9         AND medal = 1
10 GROUP BY games_name, region, total_medals
11 ORDER BY numeric DESC
12 LIMIT 10;

```

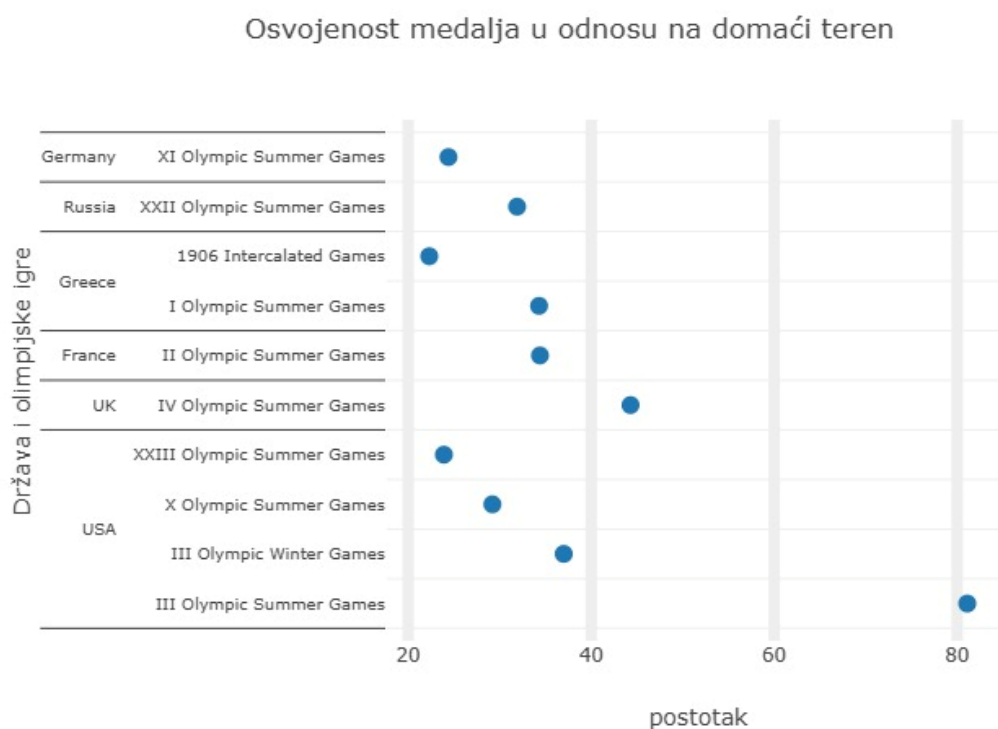
Slika 26. Primjer upita na skladište (2.)

Rezultat ovog upita je:

Result 1 (10)			
<input type="checkbox"/>	games_name	region	numeric
<input type="checkbox"/>	III Olympic Summer Games	USA	81.1
<input type="checkbox"/>	IV Olympic Summer Games	UK	44.3
<input type="checkbox"/>	III Olympic Winter Games	USA	37
<input type="checkbox"/>	II Olympic Summer Games	France	34.4
<input type="checkbox"/>	I Olympic Summer Games	Greece	34.3
<input type="checkbox"/>	XXII Olympic Summer Ga...	Russia	31.9
<input type="checkbox"/>	X Olympic Summer Games	USA	29.2
<input type="checkbox"/>	XI Olympic Summer Games	Germany	24.4
<input type="checkbox"/>	XXIII Olympic Summer G...	USA	23.9
<input type="checkbox"/>	1906 Intercalated Games	Greece	22.3

Slika 27. Rezultat upita

Te vizualno, ali ovaj put u priloženoj slici koja je izvezena u .PNG formatu:




Slika 28. Vizualizacija rezultata odgovarajućeg upita

Načini plaćanja i usluge

Amazon Redshift je usluga koja posluje na način „Plati ono što koristiš“. U ovom projektnom radu pristupilo se besplatnom probnom roku (*free trail/free tier*) korištenja usluga AWS-a. Prilikom izrade računa potrebno je unijeti broj naše kartice te izabrati jednu od opcija kao što je prikazano na slici ispod.


☒ **Basic support - Free**

- Recommended for new users just getting started with AWS
- 24x7 self-service access to AWS resources
- For account and billing issues only
- Access to Personal Health Dashboard & Trusted Advisor




☐ **Developer support - From \$29/month**

- Recommended for developers experimenting with AWS
- Email access to AWS Support during business hours
- 12 (business)-hour response times



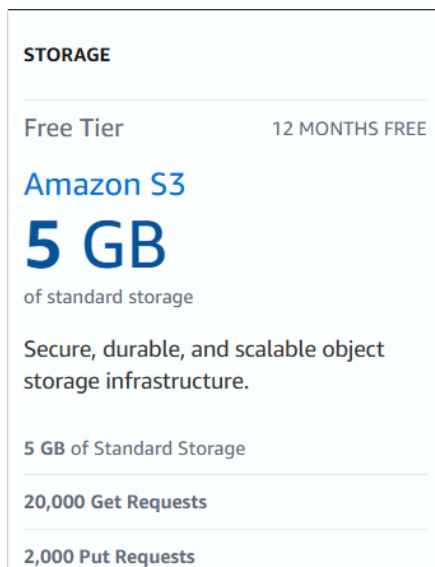
☐ **Business support - From \$100/month**

- Recommended for running production workloads on AWS
- 24x7 tech support via email, phone, and chat
- 1-hour response times
- Full set of Trusted Advisor best-practice recommendations



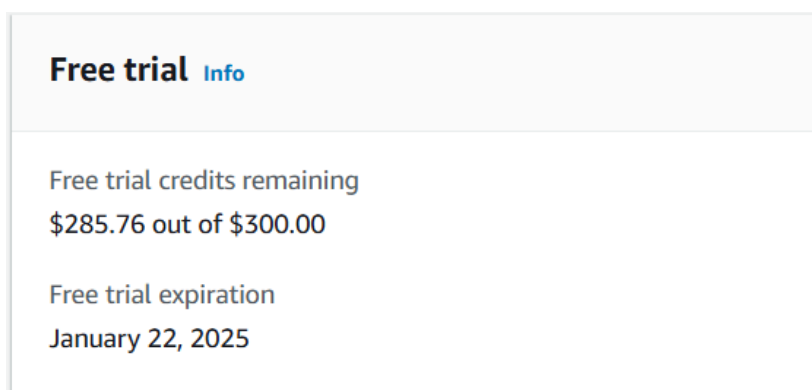
Slika 29. Odabir opcije

Za potrebe ovog rada koristio se Amazon S3 koji nudi probno razdoblje od 12 mjeseci uz određena ograničenja.



Slika 30. Mogućnosti korištenja S3 u free tier

Amazon Redshift ima dvije mogućnosti korištenja u sklopu besplatnog probnog razdoblja. Prva se odnosi na korištenje jednog čvora tipa *dc2.large* na dva mjeseca, a druga na Redshift Serverless u sklopu koje se dobiva 300\$ kredita. Za ovaj projektni rad odabran je Amazon Redshift Serverless koji se „nameće“. Kada se pristupi prvi puta Redshiftu kroz Amazon konzolu navedeno je kako se može isprobati ova mogućnost te da regije u kojima još uvijek nije navedeno mogu preći na korištenje *dc2.large* čvora (ograničenog na samo jedan čvor) kao besplatne opcije s ograničenim periodom. Tokom svih upita i cjelokupnog korištenja Redshift serverlessa potrošeno je \$15 dolara.



Slika 31. Potrošenost kredita u Serverlessu

Bitno je za istaknuti kako u klasterima ovisno koju vrstu čvorova odaberemo i koliki broj, tolika će biti i cijena koja može dosegnuti vrtoglave iznose. Naravno, to sve zavisi od potreba korisnika.

Zaključak

Amazon Redshift pokazuje se kao odlično rješenje kada je u pitanju skladište podataka u oblaku. Uz mnogobrojne opcije podešavanja performansi obrade podatka u pogledu odabira

vrste čvora i pripadnog broja istih ili pak odabira RPU-ova za Serverless. Sve to omogućuje brzo i skalabilno upravljanje velikim količinama podataka. Kroz rad je navedeno kako se Redshift temelji na OLAP-u i stupčastoj pohrani što ga uvelike ističe naspram ostalih usluga koje barataju skladištima podataka. Također, kroz samo demonstraciju Redshift query editor v2 vidjeli smo kako je lako pisati naredbe, upite i dobiti vizualne rezultat. Sve je to ipak samo mali dio mogućnosti vezanih za Amazon Redshift koji je prikazan u ovom radu. Ne zaboravljajući i samu cijenu koja je na principu „Plati ono što koristiš“. Ako stavimo bok uz bok performanse i cijenu dobivamo jednu od najboljih usluga na tržištu u ovom aspektu računarstva.

Literatura

- [1] Web stranica Amazon Web Services, <https://aws.amazon.com/redshift/> (datum pristupa: 24.10.2024.)
- [2] Web stranica Amazon Web Services, https://docs.aws.amazon.com/redshift/latest/dg/c_SQL_commands.html (datum pristupa: 24.10.2024.)
- [3] Web stranica Amazon Redshift Serverless, <https://aws.amazon.com/redshift/redshift-serverless/> (datum pristupa: 24.10.2024.)
- [4] Krivov, A. The Complete Guide to Amazon Redshift Architecture and its Components <https://medium.com/@arskrivov/the-complete-guide-to-amazon-redshift-architecture-and-its-components-e56f7d33e533> (datum pristupa: 24.10.2024.)
- [5] Amazon Redshift Serverless Explained in 90 Seconds | Amazon Web Services <https://www.youtube.com/watch?v=7vVmZhc4DS8> (datum pristupa: 24.10.2024.)
- Ostalo:
- [6] Slika 1 - <https://docs.aws.amazon.com/prescriptive-guidance/latest/query-best-practices-redshift/data-warehouse-arch-components.html>
- [7] Slika 2 - <https://docs.aws.amazon.com/images/redshift/latest/dg/images/03a-Rows-vs-Columns.png>
- [8] Slika 3 - <https://docs.aws.amazon.com/images/redshift/latest/dg/images/03b-Rows-vs-Columns.png>
- [9] Slika 9 - <https://docs.aws.amazon.com/images/redshift/latest/dg/images/architecture.png>