UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Workshop 1

Authors:
Brayan Stiven Yate Prada - Student ID: 20192020151
Holman Andres Alvarado Diaz - Student ID: 20201020032

Universidad Distrital Francisco José de Caldas

Faculty of Engineering

Course: Databases II
Professor: Carlos Andrés Sierra Virgüez

Year: 2025

1. Business Model Canvas

Spotify operates under a freemium business model, which means it generates revenue from both subscriptions and advertising. Users can access unlimited music but with certain restrictions, such as ads or limited song skips. By contrast, if a user decides to pay a monthly subscription, they receive a range of benefits, including unlimited skips, no ads, higher audio quality, and the ability to download music for offline listening.

Most of Spotify’s income comes from its premium subscribers, as they represent a stable, recurring source of revenue. However, the platform also earns money from advertising through ad placements between songs, sponsored playlists, or brand mentions during commercial breaks, reaching specific audiences. Although this advertising-based approach brings in lower earnings, it can attract more users and potentially convert them into paying customers over time.

CFI

Key Partners

- Major labels & independent distributors.
- Publishing rights organizations & podcast networks.
- Advertising exchanges & brand sponsors.
- Device & OS manufacturers.
- Mobile operators bundling Spotify plans.

Key Activities

- Continuous platform development.
- R&D on recommendation algorithms, personalization, Hi-Fi streaming.
- Curation (editorial playlists, podcast charts) & data-driven discovery tools.
- Ad-tech sales & campaign management.
- Marketing, user acquisition & retention.
- Creator services (analytics dashboards, promotion tools).

Key Resources

- Licensed catalog of ~100 M songs & 5 M+ podcasts.
- Proprietary data & machine-learning models.
- Global brand and 600 M+ MAU community.
- Cloud streaming infrastructure & CDN edge caches.
- Ad-tech stack & payment gateways.
- Talent in engineering, data science, music industry relations.

Value Propositions

For Listeners – Instant, legal on-demand access to a global audio library; personalized discovery; offline, ad-free, Hi-Fi options via Premium.
For Creators – Worldwide distribution, real-time analytics, monetization via royalties, ads, merch, live-event tools.
For Advertisers – Highly targeted audio, video & display inventory with first-party listening data.

Customer Relationships

- Self-service, algorithm-driven experience that learns user tastes.
- Community features (Blend, Jam, collaborative playlists).
- Gamified loyalty moments (Spotify Wrapped, Daylist).
- Multichannel support (Help Center, @SpotifyCares, chat bot).
- Lifecycle e-mails, push notifications & in-app stories for retention.

Channels

- Native mobile & desktop apps.
- Web Player.
- Integrations: smart speakers, smart TVs, game consoles, wearables, in-car systems (CarPlay, Android Auto).
- Telco bundles & gift cards.
- Social media, influencer partnerships, editorial playlists.

Customer Segments

- Ad-supported listeners (free tier).
- Premium subscribers – Individual, Duo, Family, Student.
- Artists, labels & podcast creators seeking distribution & insights.
- Advertisers & agencies targeting engaged audio audiences.
- Enterprise partners (telcos, device OEMs, automotive).

Cost Structure

- Royalties & minimum-guarantee advances (~70 % of revenue).
- Cloud hosting, CDN bandwidth & data storage.
- R&D and product development (AI, UX, audio tech).
- Marketing & promotional spend.
- Payment processing fees, customer service & global operations.

Revenue Streams

- **Primary**: recurring subscription fees (Premium, Duo, Family, Student).
- **Secondary**: advertising sales (audio, video, display, podcast ads, sponsored playlists).
- **Emerging**: commissions on ticketing & merch, audiobooks marketplace, Artist marketing tools, creator subscription features.

2. Requirements Documentation

Functional Requirements

Aspect	ID	Requirement
Account & Identity	F01	The system shall provide user registration and authentication via email + password, social OAuth, and enterprise SSO, storing only hashed credentials and MFA tokens.
	F02	The system shall allow users to create, read, update, and delete profile data (display name, avatar, language, payment country, parental controls).
	F03	The system shall manage subscription lifecycle events (upgrade, downgrade, cancel, pause, re-activate) with prorated billing, grace periods, and an immutable event log.
Content Catalog & Ingestion	F04	The system shall ingest audio assets (tracks, albums, podcasts, artwork, lyrics, chapters, merch links) and validate metadata before persisting to the catalog database and object storage.
	F05	The system shall enrich audio with ML-generated features (tempo, key, loudness, genre, mood) and write them to a feature store.
Discovery & Personalization	F06	The system shall provide full-text search with autocomplete and typo-tolerance over music, podcasts, playlists, and profiles, meeting the performance targets in NFR-P1.
	F07	The recommendation engine shall generate ranked lists (Daily Mix, Discover Weekly, radio). (desirable: via gRPC)
	F08	Editors shall be able to create, version, and A/B-test editorial playlists; playlist metadata is stored relationally.
Playback & Delivery	F09	The system shall negotiate streaming sessions (codec, bitrate, CDN edge) and return signed chunk URLs.
	F10	The system shall perform adaptive bit-rate streaming, adjusting quality based on telemetry and caching session state in memory.
Social & Community	F11	The system shall enable collaborative playlists with optimistic locking and publish edit events to an activity feed.
Advertising & Monetization	F12	The ad platform shall allow CRUD for campaigns, creatives, and targeting rules; impressions are logged to an append-only column store.
	F13	The ad decision engine shall select an ad in ≤ 50 ms using listener attributes from an in-memory feature cache.
	F14	The finance service shall aggregate stream counts, apply contractual overrides, and calculate royalties for payouts.
Creator & Analytics Tools	F15	The creator dashboard shall surface near–real-time plays, geographic heat maps, and playlist additions via interactive queries.

	F16	The promo-tool workflow shall let creators schedule and pay for promotions, persisting billing events to a ledger.
Governance & Compliance	F17	The system shall write immutable audit logs of admin actions, data exports, and rights changes.
	F18	The system shall fulfill data-subject requests—export or delete all personal information — within 30 days.

Non-Functional Requirements

Id	Category	Requirement	Priority
NFR-P1	Performance	Playback start-up ≤ 300 ms 95% of the time.	High
NFR-P2	Performance	Search response ≤ 150 ms 95% of the time.	High
NFR-P3	Performance	Ad decision latency ≤ 50 ms.	High
NFR-S1	Scalability	Support ≥ 1 Billion of monthly active users and 20 million of concurrent streams with horizontal sharding & auto-scaling.	High
NFR-S2	Scalability	Auto-scale ingestion to 10 TB new audio/day without interruption.	High
NFR-A1	Availability	Playback service level agreement ≥ 99.95 % monthly.	High
NFR-A2	Availability	Graceful degradation: fallback CDN or lower bitrate on overload.	Medium
NFR-C1	Consistency & Integrity	Strong consistency for subscription and billing tables.	High
NFR-C2	Consistency & Integrity	Eventual consistency ≤ 5 s for social interactions (likes, follows, playlist edits).	Medium
NFR-SEC1	Security & Privacy	TLS 1.3 for all traffic in transit; AES-256 for all data at rest.	High
NFR-SEC2	Security & Privacy	Role-based access with row-level security on catalog data.	High
NFR-SEC3	Security & Privacy	Tokenization for payment info.	High
NFR-OBS1	Observability	Distributed tracing on every playback session	Medium

NFR-OBS2	Observability	Central log aggregation with 30-day retention.	Medium
NFR-MNT1	Maintainability	Service interfaces documented in OpenAPI.	Medium
NFR-MNT2	Maintainability	Each microservice ≤ 3000 LOC.	Medium
NFR-MNT3	Maintainability	Zero-downtime deploys via blue/green and automated schema migrations.	Low
NFR-EXT1	Extensibility	New media types added without core-schema change via polymorphic content table.	Medium
NFR-DQ1	Data Quality	≥ 99.9 % accuracy on track metadata.	High

3. User Stories

For these user stories, we consolidate the following actors and roles for the application:

Application Roles

Listener: anyone who consumes audio content (free or Premium)

Creator: artists, podcasters, and editorial staff who upload or curate content

Advertiser: brands or agencies running audio/video campaigns

Engineer: developers and platform engineers who build & operate the system

Compliance Officer: staff who enforce privacy, security, and audit obligations

User Stories

Listener Stories

Title: Register Account	**Priority: High**	**Estimate:**
User Story: As a **Listener**, I want to create an account with email or social login so that I can start using Spotify immediately.		
Acceptance Criteria: Given I'm on the sign-up page, When I submit valid credentials or authorise with a social provider, Then a new account is created and I receive confirmation.		

Title: Listen for Free (Ad-Supported)	**Priority: High**	**Estimate:**
User Story: As a **Listener** on the free tier, I want to listen to music at no cost so that I accept occasional ads and limited features.		
Acceptance Criteria: Given I'm logged in as a free user, When I hit Play, Then playback is shuffle-only, ads appear between songs, skips are limited, and offline download is disabled.		

Title: Ad-Free Premium Playback	**Priority: High**	**Estimate:**
User Story: As a **Listener** with Premium, I want uninterrupted, high-quality audio so that I enjoy music without ads.		

Acceptance Criteria: I’m a Premium user, When I play any track, Then no ads are inserted, Hi-Fi quality is available, and I can download for offline use.

Title: Create Personal Playlist	**Priority: Medium**	**Estimate:**
User Story: As a **Listener**, I want to create and customize playlists so that I can organise songs by mood or activity.		
Acceptance Criteria: I tap “New Playlist”. When I name and add songs, Then I can reorder, describe, and share the playlist across devices.		

Title: Search Content	**Priority: High**	**Estimate:**
User Story: As a **Listener**, I want typo-tolerant search so that I can quickly find my favourite songs, albums, or artists.		
Acceptance Criteria: Given I type a query, When I pause typing, Then suggestions appear and full results return in ≤ 150 ms p95.		

Title: Personalised Recommendations	**Priority: High**	**Estimate:**
User Story: As a **Listener**, I want music suggestions based on my tastes so that I discover new songs and artists.		
Acceptance Criteria: Given a recommendation cycle runs, When I open Home, Then Discover Weekly and other auto-generated mixes appear tailored to my history.		

Title: Smart-Device Integration	**Priority: Medium**	**Estimate:**
User Story: As a **Listener**, I want to play music on my smart TV or speaker so that I enjoy Spotify across devices.		
Acceptance Criteria: Given my Google Home is linked, When I say “Play Spotify”, Then playback starts and I can hand off control to my phone.		

Title: View Listening Statistics	**Priority: Medium**	**Estimate:**
User Story: As a **Listener**, I want to see my top tracks and monthly stats so that I understand my music habits.		
Acceptance Criteria: Given a button to open my profile, When stats load, Then I see top songs/artists, monthly/yearly charts, and share buttons.		

Title: Social Features & Sharing	**Priority: Medium**	**Estimate:**
User Story: As a **Listener**, I want to follow friends and share music so that I engage with others and find new songs.		
Acceptance Criteria: Given I follow a friend, When they play music (if setting enabled), Then I can view their activity feed and share tracks via messages.		

Title: Manage Subscription & Payment	**Priority: High**	**Estimate:**
User Story: As a **Listener**, I want to update payment methods or cancel my plan so that my subscription stays accurate.		
Acceptance Criteria: Given I'm a Premium user, When I edit payment details or hit "Cancel", Then billing info updates or the plan ends with a confirmation e-mail.		

Title: Parental Controls	**Priority: Medium**	**Estimate:**
User Story: As a **Listener** who is a parent, I want to set content restrictions for my child so that they hear age-appropriate music.		
Acceptance Criteria: Given I open Family settings, When I toggle "Explicit Content Off", Then the child's account is blocked from explicit tracks and time limits apply.		

Title: Adaptive Streaming	**Priority: Medium**	**Estimate:**
User Story:		

As a **Listener** on a poor connection, I want audio quality to adjust automatically so that playback doesn’t stall.
Acceptance Criteria: Given bandwidth drops, When telemetry signals low throughput, Then bit-rate downgrades without more than one rebuffer per hour.

Creator Stories

Title: Upload New Audio	**Priority: High**	**Estimate:**
User Story: As a **Creator**, I want to upload tracks or podcast episodes with metadata so that my audience can stream them.		
Acceptance Criteria: Given I submit a file, When validation passes, Then the asset is stored and indexed in the catalog.		

Title: View Real-Time Analytics	**Priority: Medium**	**Estimate:**
User Story: As a **Creator**, I want live play counts and geography heat maps so that I understand my reach.		
Acceptance Criteria: Given I open my dashboard, When plays occur, Then charts refresh within 30 minutes.		

Title: Schedule Promotion	**Priority: Medium**	**Estimate:**
User Story: As a **Creator**, I want to schedule and pay for promotions so that I can boost a release.		
Acceptance Criteria: Given I configure a promo, When I confirm payment, Then it is stored and activates on the set date.		

Title: Catalog Metadata	**Priority: Medium**	**Estimate:**
User Story: As a **Creator**, I want to update track metadata so that errors are corrected.		
Acceptance Criteria: Given I submit new artwork, When validation succeeds, Then the catalog shows updated metadata and a version history entry.		

Title: Review Promotion Metrics	**Priority: Medium**	**Estimate:**
User Story: As a **Creator**, I want post-campaign reports so that I can gauge ROI (Return of Investment).		
Acceptance Criteria: Given a promo ends, When I open its report, Then impressions and incremental streams are displayed.		

Advertiser Stories

Title: Create Ad Campaign	**Priority: High**	**Estimate:**
User Story: As an **Advertiser**, I want to create and target audio campaigns so that my brand reaches the right listeners.		
Acceptance Criteria: Given I complete campaign details, When I click "Launch", Then the campaign starts delivering impressions.		

Title: Real-Time Ad Decision	**Priority: High**	**Estimate:**
User Story: As an **Advertiser**, I need my ad chosen within 50 ms so that listeners hear it at the right moment.		
Acceptance Criteria: Given a stream hits an ad break, When decision logic runs, Then an ad returns in ≤ 50 ms p99.		

Title: Update Campaign Creative	**Priority: Medium**	**Estimate:**
User Story: As an **Advertiser**, I want to swap creatives mid-flight so that messaging stays fresh.		
Acceptance Criteria: Given a campaign is active, When I upload a new creative, Then it serves within one hour.		

Title: View Campaign Analytics	**Priority: Medium**	**Estimate:**
User Story: As an **Advertiser**, I want real-time dashboards so that I can optimize spend.		

Acceptance Criteria: Given my campaign runs, When I open analytics, Then metrics update within five minutes.

Title: Pause or Resume Campaign	**Priority: Medium**	**Estimate:**
User Story: As an **Advertiser**, I want to pause or resume a campaign instantly so that I control delivery.		
Acceptance Criteria: Given a campaign is live, When I hit “Pause”, Then no new impressions serve until I “Resume”.		

Engineer Stories

Title: Meet Playback SLA	**Priority: High**	**Estimate:**
User Story: As an **Engineer**, I want 99.95 % monthly playback uptime so that users trust the service.		
Acceptance Criteria: Given monitoring is active, When a month ends, Then total downtime < 22 minutes.		

Title: Auto-Scale Ingestion	**Priority: High**	**Estimate:**
User Story: As an **Engineer**, I need pipelines to auto-scale to 10 TB/day so that catalog drops don’t backlog.		
Acceptance Criteria: Given a 10 TB/day load, When scaling triggers, Then queue delay stays < 15 minutes.		

Title: Distributed Tracing	**Priority: Medium**	**Estimate:**
User Story: As an **Engineer**, I want every playback session traced end-to-end so that I can debug latency.		
Acceptance Criteria: Given 100 random sessions, When I query traces, Then 100 % have complete trace IDs.		

Title: Blue/Green Deployments	**Priority: Medium**	**Estimate:**
User Story: As an **Engineer**, I want zero-downtime releases so that deploys don’t disrupt listeners.		

Acceptance Criteria: Given a rollout starts, When traffic shifts, Then error rates remain baseline and rollback < 5 minutes.

Title: Central Log Aggregation	**Priority: Medium**	**Estimate:**
User Story: As an **Engineer**, I want 30-day log retention so that incidents can be diagnosed.		
Acceptance Criteria: Given logs are ingested, When I query the past 30 days, Then data is present and searchable.		

Compliance Officer Stories

Title: Tokenized Payment Data	**Priority: High**	**Estimate:**
User Story: As a **Compliance Officer**, I want all payment info tokenized so that raw card data is never stored.		
Acceptance Criteria: Given a payment occurs, When data is saved, Then only tokens are persisted.		

Title: Fulfil GDPR Requests	**Priority: High**	**Estimate:**
User Story: As a **Compliance Officer**, I want to complete data export or deletion within 30 days so that we meet legal duties.		
Acceptance Criteria: Given a valid request, When 30 days pass, Then data is exported or erased and logged.		

Title: Immutable Audit Logs	**Priority: Medium**	**Estimate:**
User Story: As a **Compliance Officer**, I want every admin action recorded immutably so that audits prove no tampering.		
Acceptance Criteria: Given an admin edits data, When the action completes, Then an entry appears in the append-only log.		

Title: Conduct Access Reviews	**Priority: Medium**	**Estimate:**
User Story: As a **Compliance Officer**, I want quarterly role-based access reviews so that least-privilege is enforced.		
Acceptance Criteria: Given the quarter ends, When I run the access-report tool, Then all user-role mappings export for sign-off.		

Title: Enforce Log-Retention Policy	**Priority: Medium**	**Estimate:**
User Story: As a **Compliance Officer**, I want logs older than 30 days purged automatically so that we respect retention rules.		
Acceptance Criteria: Given logs exceed 30 days, When the retention job runs, Then outdated records delete and a summary report is filed.		

4. Initial Database Architecture

The project implements a distributed database architecture designed to meet the demands of consistency, performance, analytics, and global availability—leveraging entirely free and open-source technologies. Each layer serves a distinct purpose within the platform ecosystem and interconnects to ensure the efficient processing of operational data, user sessions, multimedia content, event streams, and analytics.

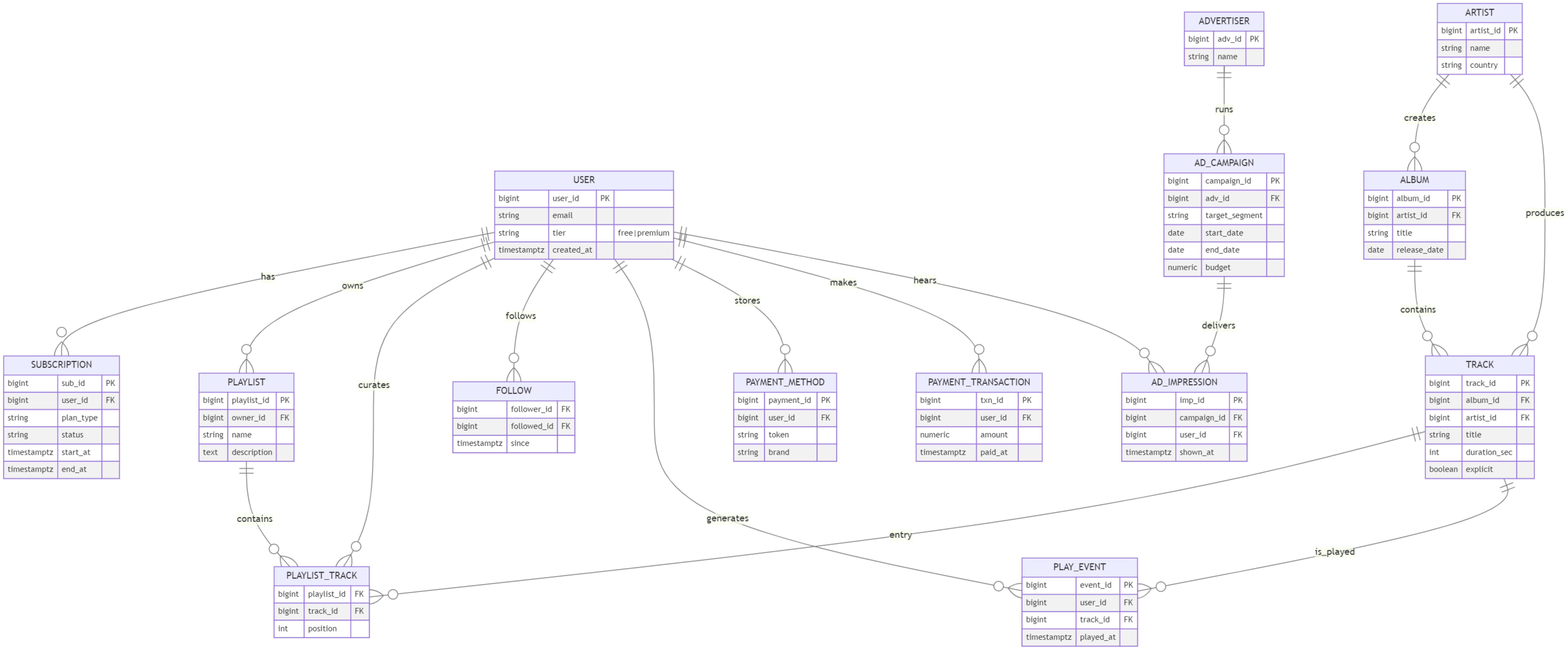
For example, critical operations such as user management, subscriptions, and billing are handled through the transactional data layer (Hot OLTP Data). This layer ensures strong consistency using technologies like CockroachDB or PostgreSQL with Citus, leveraging geo-partitioning by user_id or asset_id.

Meanwhile, session data and ephemeral attributes like playback state or adaptive telemetry are stored in the session layer, where Cassandra or Scylla provide fast access and fault-tolerant replication. Audio files, artwork, and audit logs are stored in object storage solutions such as MinIO or Ceph RGW, optimized through multi-site replication and lifecycle management rules.

Every user interaction—from a song play to an ad click—is captured as an immutable event stream via Kafka, while recommendation logic and social graphs are modeled using JanusGraph or the Neo4j Community Edition. These data streams feed real-time analytics systems and long-term warehouses such as Trino and Apache Iceberg, where dashboards, royalty calculations, and predictive model training are performed. On the following table there is a summary of the previous information.

Layer	Purpose	Technology	Distribution & Scalability
Hot OLTP Data	Critical data with strong consistency (users, subscriptions, payments).	*CockroachDB*, *PostgreSQL + Citus*	Geo-partitioning by `user_id` / `asset_id`; ACID transactions.
Feature / Session Store	Low-latency access to session states, recommendations, and telemetry.	*Apache Cassandra*, *Scylla Open Source*	Quorum writes, TTL policies, rack-aware replication.
Object Storage	Multimedia files, artwork, audit logs.	*MinIO* (S3-compatible), *Ceph RGW*	Erasure coding, lifecycle rules, cross-region replication.
Search Index	Full-text search and autocomplete for songs, artists, and playlists.	*OpenSearch*	Denormalized JSON, real-time CDC, nightly batch reprocessing.
Graph Store	Social relationships and personalized recommendation modeling.	*JanusGraph* on Cassandra, *Neo4j Community Edition*	Vertex-hash partitioning, Gremlin query language support.
Stream Backbone	Immutable log of user events (plays, likes, impressions).	*Apache Kafka + Kafka Connect*	Exactly-once producers, event-domain-based topic design.
Real-Time Analytics	Near-instant dashboards for creators and advertisers.	*Apache Flink*, *Apache Spark SS → ClickHouse*, *Apache Druid*	Sliding window aggregations, REST/gRPC API access.
Deep Analytics Warehouse	Historical analysis, machine learning, royalty calculation.	*Trino + Apache Iceberg*, *ClickHouse*, *Apache Druid*	Columnar storage (Parquet/ORC), decoupled compute and storage layers.

ER Diagram



USER			
bigint	user_id	PK	
string	email		
string	tier		free|premium
timestamptz	created_at		

SUBSCRIPTION		
bigint	sub_id	PK
bigint	user_id	FK
string	plan_type	
string	status	
timestamptz	start_at	
timestamptz	end_at	

PLAYLIST		
bigint	playlist_id	PK
bigint	owner_id	FK
string	name	
text	description	

FOLLOW		
bigint	follower_id	FK
bigint	followed_id	FK
timestamptz	since	

PAYMENT_METHOD		
bigint	payment_id	PK
bigint	user_id	FK
string	token	
string	brand	

PAYMENT_TRANSACTION		
bigint	txn_id	PK
bigint	user_id	FK
numeric	amount	
timestamptz	paid_at	

AD_IMPRESSION		
bigint	imp_id	PK
bigint	campaign_id	FK
bigint	user_id	FK
timestamptz	shown_at	

ADVERTISER		
bigint	adv_id	PK
string	name	

AD_CAMPAIGN		
bigint	campaign_id	PK
bigint	adv_id	FK
string	target_segment	
date	start_date	
date	end_date	
numeric	budget	

ARTIST		
bigint	artist_id	PK
string	name	
string	country	

ALBUM		
bigint	album_id	PK
bigint	artist_id	FK
string	title	
date	release_date	

TRACK		
bigint	track_id	PK
bigint	album_id	FK
bigint	artist_id	FK
string	title	
int	duration_sec	
boolean	explicit	

PLAYLIST_TRACK		
bigint	playlist_id	FK
bigint	track_id	FK
int	position	

PLAY_EVENT		
bigint	event_id	PK
bigint	user_id	FK
bigint	track_id	FK
timestamptz	played_at	

Entity–relationship diagram description

Entity	Purpose	Key Relationships
User	Listener account & profile.	1-to-many **Subscription**, **Playlist**, events, payments, impressions.
Subscription	Tracks plan state, billing window, status.	Belongs to **User**.
Artist / Album / Track	Core catalog.	Artist ⇢ Album ⇢ Track hierarchy; Track appears in many playlists & play events.
Playlist / Playlist Track	User-curated or editorial lists.	Many-to-many between **Playlist** and **Track** with order.
Follow	Social graph (user ↔ user or user ↔ artist).	Self-referencing relation on **User**.
Play Event	Immutable record of every stream (for royalties, recommendations).	Links **User** and **Track**.
Payment Method / Transaction	Tokenized billing info & charges.	Both reference **User**.
Advertiser / Ad Campaign / Ad Impression	Ad inventory & delivery for monetisation.	**Advertiser** runs many **Campaigns**; each campaign yields many **Impressions** tied to **User**.

5. References

- Corporate Finance Institute. (n.d.). Business model canvas examples. Corporate Finance Institute. Retrieved from https://corporatefinanceinstitute.com/resources/management/business-model-canvas-examples/
- Business Model Analyst. (n.d.). Spotify business model. Business Model Analyst. Retrieved from https://businessmodelanalyst.com/spotify-business-model/
- Music Business Research. (2024, August 19). The music streaming economy part 10: Spotify's business model. Music Business Research. Retrieved from https://musicbusinessresearch.wordpress.com/2024/08/19/the-music-streaming-economy-part-10-spotifys-business-model/
- Investopedia. (n.d.). How Spotify makes money. Investopedia. Retrieved from https://www.investopedia.com/articles/investing/120314/spotify-makes-internet-music-make-money.asp
- IIDE. (n.d.). Business model of Spotify. IIDE. Retrieved from https://iide.co/case-studies/business-model-of-spotify/

- GrowthX Club. (n.d.). Spotify business model. GrowthX Club. Retrieved from https://growthx.club/blog/spotify-business-model#spotify-revenue-model
- Software Developer Diaries [@SoftwareDeveloperDiaries]. (n.d.). How Spotify's playback works under the hood. Youtube. Retrieved from https://www.youtube.com/watch?v=K26bGXVR-mE
- Lopez, M. A. (n.d.). Deep dive into the tech stack used by Spotify. LinkedIn. Retrieved from https://www.linkedin.com/pulse/deep-dive-tech-stack-used-spotify-marny-a-lopez
- Intuji. (n.d.). How does Spotify work? Tech stack explored. Intuji. Retrieved from https://intuji.com/how-does-spotify-work-tech-stack-explored/