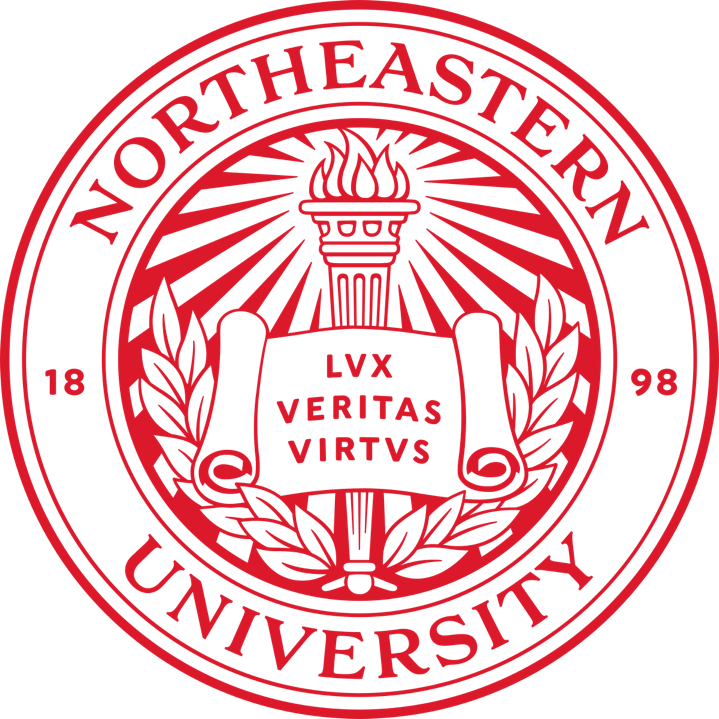
****

**Final Project Report : Hire Musicians**

Sankalp Susil Kumar Biswal

College of Professional Studies, Northeastern University Boston

ITC6000- Database Management Systems

Prof. Yu Na

October 29, 2023

Table of Contents

[Introduction 3](#_Toc149482646)

[Application cost model 3](#_Toc149482647)

[Why it interests me? 3](#_Toc149482648)

[Business challenges addressed 3](#_Toc149482649)

[Major User Types 3](#_Toc149482650)

[Primary Use case 3](#_Toc149482651)

[Business Implications 3](#_Toc149482652)

[Business Analysis 4](#_Toc149482653)

[End-users or personas- 4](#_Toc149482654)

[Business Rules- 4](#_Toc149482655)

[Out of Scope- 5](#_Toc149482656)

[Table Design and Analysis 6](#_Toc149482657)

[Data entities- 6](#_Toc149482658)

[Table Relationships 7](#_Toc149482659)

[Database Implementation 8](#_Toc149482660)

[Musician user journey- 8](#_Toc149482661)

[Employer user journey- 9](#_Toc149482662)

[Additional query- 10](#_Toc149482663)

[Analytics, Reports, and Metrics 12](#_Toc149482664)

[Musicians 12](#_Toc149482665)

[Employer/Music Companies 12](#_Toc149482666)

[Application Administrator 13](#_Toc149482667)

[Security Concerns 14](#_Toc149482668)

[Architecture 15](#_Toc149482669)

[Client/Server Architecture 15](#_Toc149482670)

[Hosting Model: 15](#_Toc149482671)

[Storage Estimation: 15](#_Toc149482672)

[Learnings & Takeaways 16](#_Toc149482673)

[Future Considerations 16](#_Toc149482674)

[References 17](#_Toc149482675)

# 

# **Introduction**

I am Sankalp and for the topic of the final project I’ve chosen to work on an idea in which musicians and music companies can hire/book other musicians either for a full-time contract or part-time gigs on our web application, depending upon their requirements in terms of years of experience, type of instrument played, hourly rate/ contract and genres played.

## **Application cost model**

The web application will use a hybrid model combining different revenue sources allowing diversification of income streams. The hybrid model would be such that –

1. Free version - Users will be limited to 10 applications or listings with ads.
2. Premium version - Users will be able to create unlimited applications or listings and the experience will be totally ad-free.

## **Why it interests me?**

I am a musician myself and have been playing guitar for the past 8 years and also have produced few beats and tracks as personal projects. I have played gigs at different venues and realized how crucial it is for a musician to have a source of income in order to support his/her art.

## **Business challenges addressed**

This sector is fairly unorganized and musicians encounter the following problems-

1. Not paid timely for the gigs.
2. Not paid at all due to lack of a formal agreement before the gig/event.
3. Uncertainty in terms of when will they get their next gig/show.

So, In order to help fellow musicians and address the above stated problems, I have chosen to work on this idea.

## **Major User Types**

1. Musicians/Producers/Composers
2. Music companies/ Artist Managers/Band Managers

## **Primary Use case**

1. To be able to book/hire musicians part-time or full-time – Musicians / Music companies will log on to the application and provide details to be able to either view details about other musicians and hire/book them according to the user’s needs.

## **Business Implications**

1. Increased opportunities for musicians to earn money and increase their online and offline presence

# **Business Analysis**

## **End-users or personas-**

1. **Musician Seeking Gigs:**

* Who are they: Musicians of various skill levels, from aspiring artists to seasoned professionals.
* Why they use: To find paid gigs or opportunities that match their skills, preferences, and availability.
* How they use: They use the platform to create musician profiles, showcase their work, search for gigs, apply for opportunities, communicate with potential employers, and manage their schedules.

1. **Music Company or Event Organizer:**

* Who are they: Event organizers, music producers, recording studios, and entertainment agencies.
* Why they use: To discover talented musicians for live performances, studio recording, or other musical projects.
* How they use: They create profiles for their companies, post gig or project listings, browse musician profiles, negotiate terms, and schedule musicians for events or recording sessions.

1. **Talent Scouts and A&R Professionals:**

* Who are they: Talent scouts, music producers, and artist and repertoire (A&R) representatives.
* Why they use: To identify and recruit promising musicians for record labels or talent development.
* How they use: They create employer profiles to search for emerging talent, view portfolios, and initiate discussions with musicians for potential collaborations or record deals.

1. **Venue Managers:**

* Who are they: Managers of venues like bars, clubs, and concert halls.
* Why they use: To book musicians and bands for performances at their venues.
* How they use: They create employer profiles for their venues, post gig opportunities, and interact with musicians to schedule performances.

## **Business Rules-**

1. A user can make a musician or employer
2. A musician profile or employer profile will be associated with one user.
3. A user with a musician profile can apply to multiple job listings via a job application.
4. A user with an employer profile can create multiple job listings.
5. Many users can send message to many users.

## **Out of Scope-**

1. Users can post and see reviews for musicians and employers.
2. Users can upload their profile picture and also upload videos of sample work.

# **Table Design and Analysis**

## **Data entities-**

1. **User**:

* **Description**: The central entity representing all users of the platform. Users can have different roles, including musicians, employers, or music company representatives.
* **Attributes**: UserID (primary key), username, email, and password.

1. **Musician\_Profile**:

* **Description**: This entity stores detailed information about users who are musicians. It is linked to the User entity via the User ID and includes information specific to musicians.
* **Attributes**: MusicianID (primary key), FK\_UserID\_Musician (foreign key), name, Genres, Experience, Skills, and Links.

1. **Employer\_Profile**:

* **Description**: Similar to the Musician\_Profile, this entity stores detailed information about users who are employers or representatives of music companies. It is linked to the User entity via the User ID.
* **Attributes**: EmployerID (primary key), FK\_UserID\_Employer (foreign key), CompanyName, and ContactInfo.

1. **Job\_Application**:

* **Description**: This entity manages job applications created by musicians in response to job listings posted by employers or music companies. It tracks the application details and status.
* **Attributes**: ApplicationID (primary key), FK\_MusicianID (foreign key), and FK\_JobID (foreign key).

1. **Job\_Listing**:

* **Description**: This entity stores data related to job offers or gig listings created by employers or music companies. It contains information about available opportunities for musicians to apply.
* **Attributes**: JobID (primary key), FK\_EmployerID (foreign key), JobTitle, Description, location, and Payment.

1. **Messages**:

* **Description**: The Messages entity is responsible for managing conversations and messages exchanged between users on the platform. It facilitates communication and collaboration.
* **Attributes**: Message ID (primary key), FK\_SenderID\_UserID (foreign key), FK\_ReceiverID\_UserID (foreign key), and content.

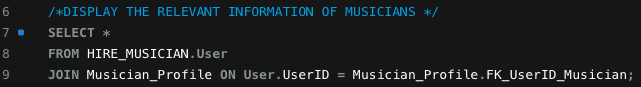
## **Table Relationships**

1. User and Musician\_Profile have one to one relationship.
2. User and Employer\_Profile have one to one relationship.
3. User and Messages have many to many relationship.
4. Employer\_Profile and Job\_Listing have one to optional many relationship.
5. Musician\_Profile and Job\_Application have one to optional many relationship.

# **Database Implementation**

## **Musician user journey-**

1. **Retrieving musicians and their profiles**

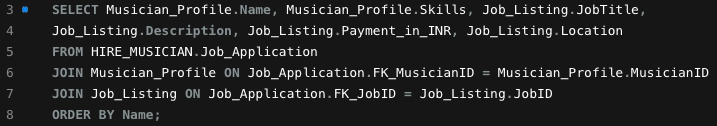
*Raw SQL -*

*Output -*

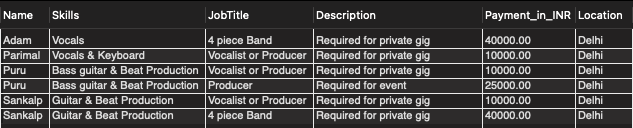


1. **Retrieving musician profiles and their job application**

*Raw SQL*

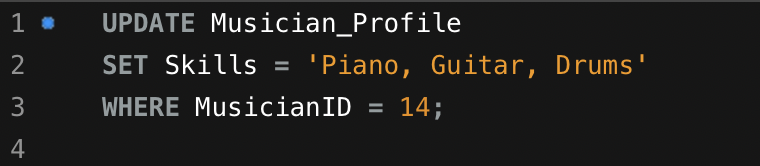


*Output*



1. **Updating skills**

*Raw SQL*

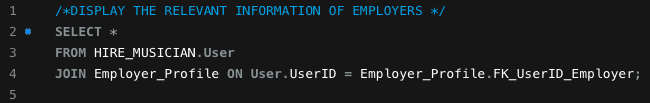


*Output*



## **Employer user journey-**

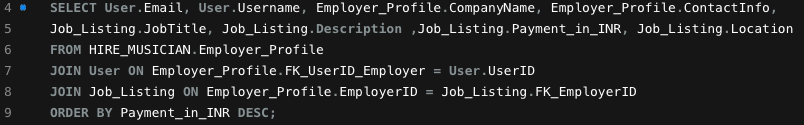
* 1. **Retrieving employers and their profiles**

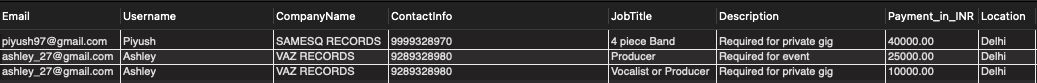
*Raw SQL*

*****Output*

* 1. **Retrieving employer details and the job listings they’ve posted**

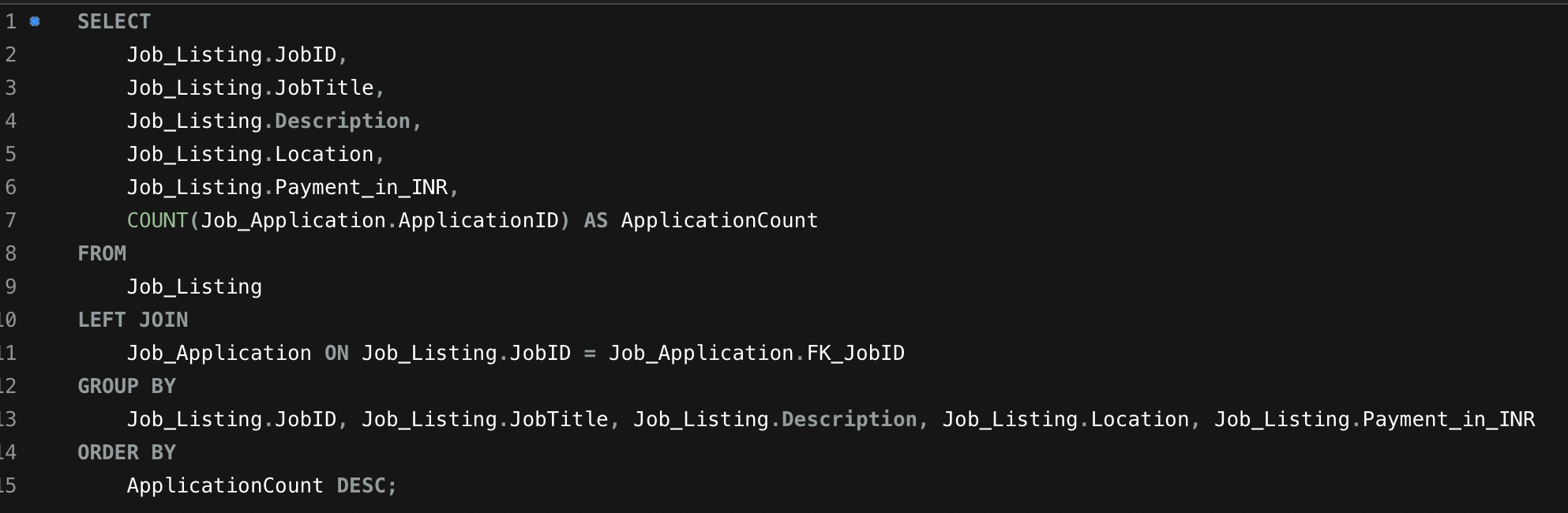
*Raw SQL*



*Output*

* 1. **Retrieving number of applications applied to each job listing**

*Raw SQL*

****

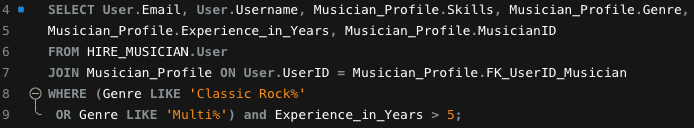
*Output*

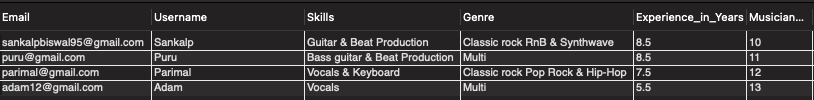
****

## **Additional query-**

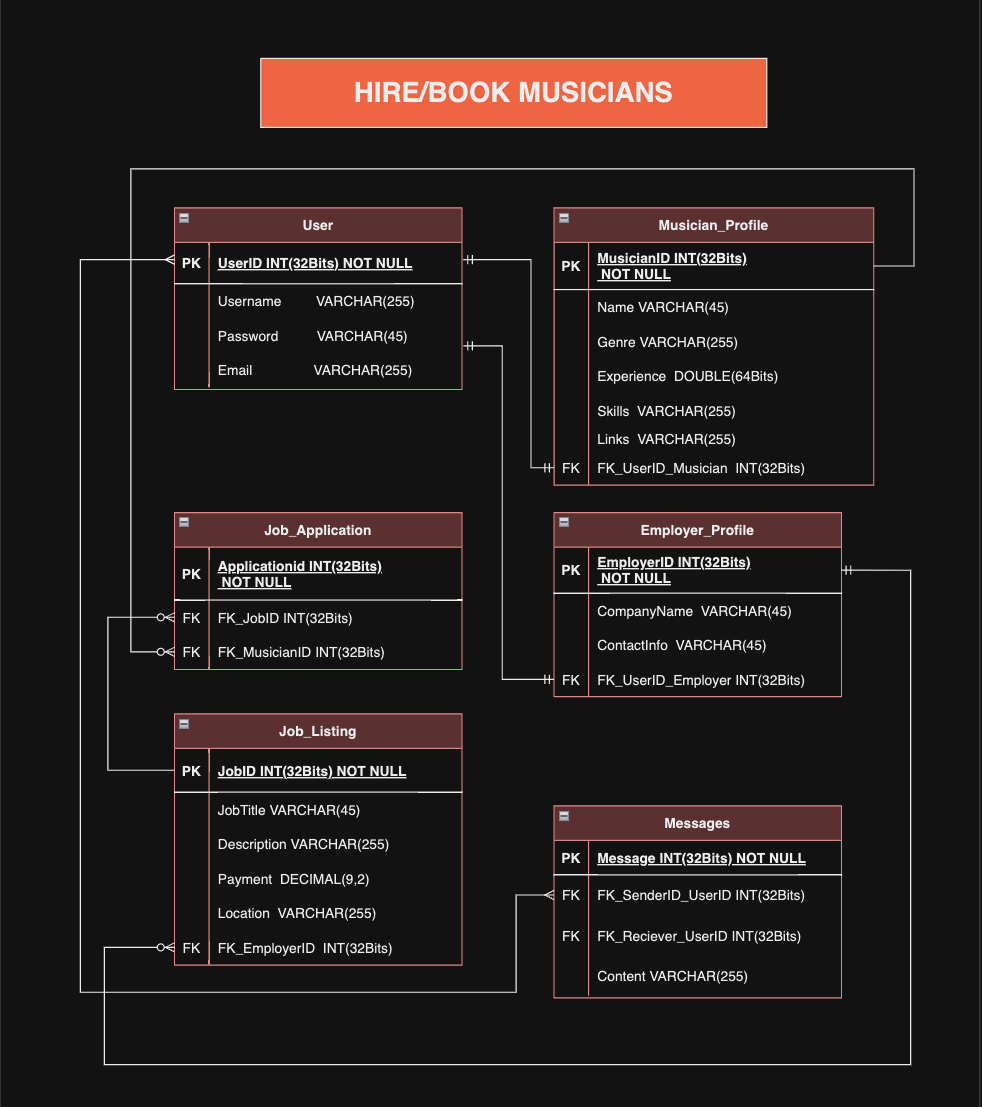
1. **Find a musician with experience more than five years who plays either ‘Classic rock’ or ‘Multi’**

*Raw SQL*

****

**** *Output*

**Full Page ERD on next page 🡪**

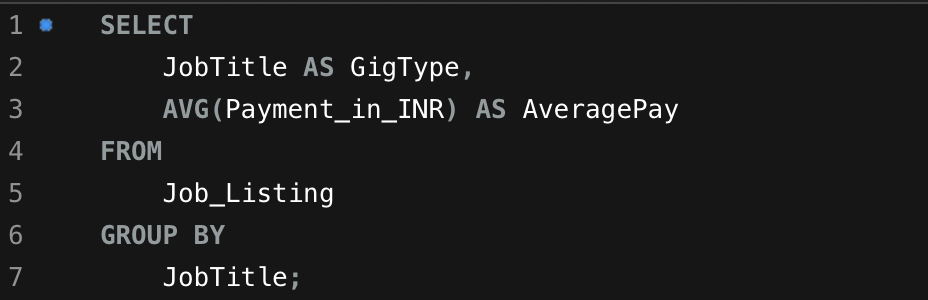


# **Analytics, Reports, and Metrics**

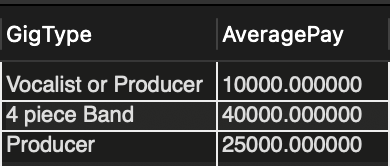
## **Musicians**

* Average pay for certain types of gigs -

*Raw SQL*

****

*Output*

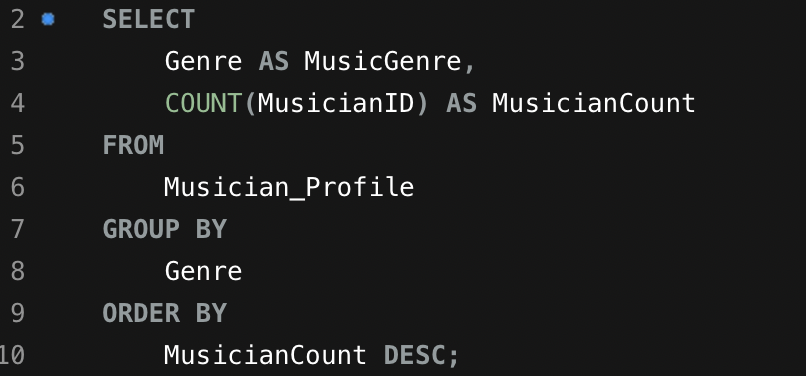
****

* Number of gigs performed within a month.

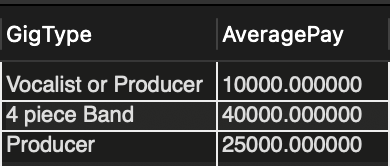
## **Employer/Music Companies**

* Number of musicians playing a certain type of genre -

*Raw SQL*

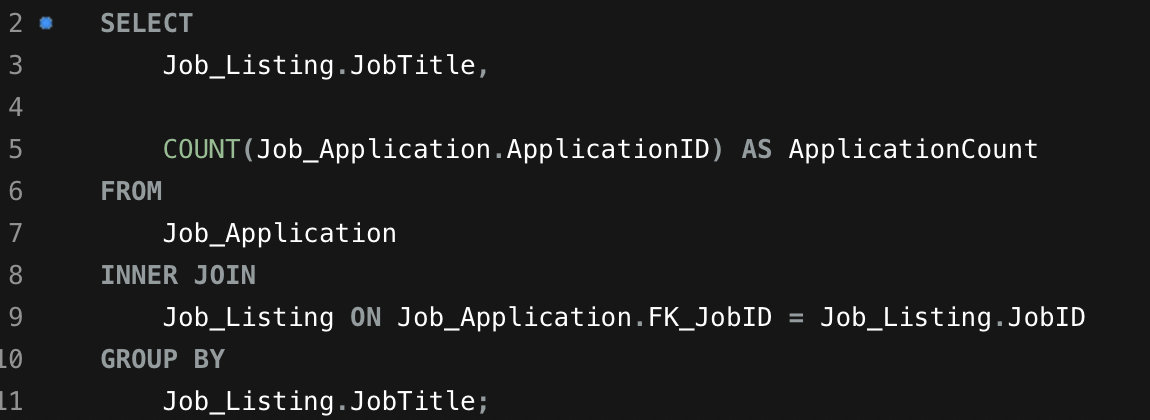
****

*Output*

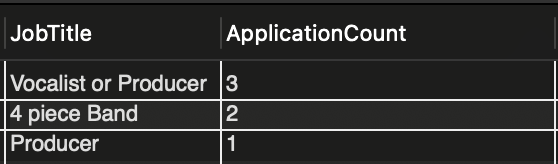
****

* Number of applicants for each job listing -

*Raw SQL*

****

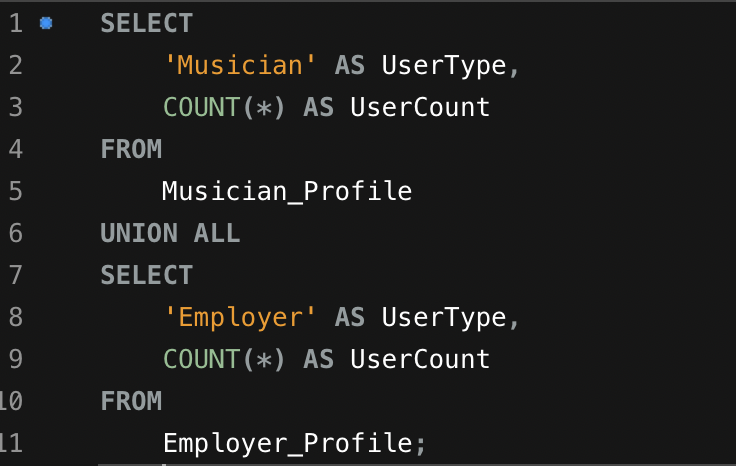
*Output*

****

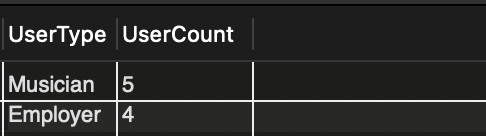
## **Application Administrator**

* Total number of users and ratio of musicians to employers

*Raw SQL*

****

*Output*

****

# **Security Concerns**

As the data expert for the project involving hiring musicians, it's important to be aware of potential security and privacy concerns related to the data. While I am not a security expert, I can identify some data-related concerns:

1. **Personal Information:** The database contains personal information of musicians and employers, including names, email addresses, and contact information. Unauthorized access to this data could lead to privacy breaches.
2. **Messaging Data:** Messages exchanged between users should be secure to prevent interception or unauthorized access. Ensuring the confidentiality of communications is crucial.
3. **Authentication and Authorization:** Implementing strong user authentication and authorization mechanisms is essential to prevent unauthorized access to user profiles and data.
4. **Data Encryption:** Data at rest and in transit should be encrypted to protect against data breaches. Secure socket layer (SSL) encryption for web communication is important.
5. **Data Backups:** Regular data backups are necessary to ensure data recovery in case of data loss or system failures.

# **Architecture**

**Client/Server Architecture:** This project would require a 3-tier architecture since a user will interact with a system that retrieves and presents data from database.

*Front-end - React*

1. **Client Side:** This would be side with which our end-user would interact. It would be initially a web-application. The Front-end will be designed using React

*Back****-****end – Ruby on rails or Django (Python)*

1. **Server Side:** The server side is responsible for processing requests, managing databases, and handling business logic.
2. **Database:** The ERD tables (User, Musician Profile, Job Application, Employer Profile, Job Listing, Messages) would be stored and managed on the server-side database. We would likely use a database management system (DBMS) like MySQL, PostgreSQL, or a NoSQL database like MongoDB, depending on your data requirements.

## **Hosting Model:**

1. **Cloud Hosting:** Cloud hosting is a popular choice for web applications because it offers scalability, reliability, and flexibility. Services like Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), or platform-as-a-service (PaaS) providers like Heroku can be used to host the server-side components, including the web server, application server, and database server. Cloud hosting will allow scaling resources as needed
2. **Content Delivery:** If we include the feature wherein users can upload their sample work in the form of a video or audio file, then we might use a content delivery network (CDN) to serve static assets like audio files and videos to users from servers located closer to their geographical locations for faster retrieval times.

## **Storage Estimation:**

**For Project (Total users 10, 8 Musicians and 2 Employers) –**

1. User Profiles - 1MB per user, 10MB Total
2. Messages - 1MB per message, 10MB Total
3. Media Files – Average 10MB per user, 100MB Total
4. Job Applications/Job Listings – 1MB per application and Listing, 20MB
5. Total Requirement – Approximately 150MB

**For Real world Implementation(1 Million users)**

1. User Profiles –  1MB per user, 1000GB Total
2. Messages - 1MB per message, 1000GB Total
3. Media Files – Average 10MB per user, 1000GB Total
4. Job Applications/Job Listings – 1MB per application and Listing, 2000GB
5. Total Requirement – Approximately 10TB

# **Learnings & Takeaways**

* **Data Modelling:** I've learned how to design a complex database system that caters to the needs of multiple user roles, such as musicians and employers. Understanding the entities, their relationships, and attributes is essential for creating an effective data model.
* **Data Privacy:** I've become aware of the importance of data privacy and the need to handle sensitive user information, such as personal details and passwords, with care. Implementing encryption (For e.g SHA1() implemented in this project to encrypt passwords), access controls, and data retention policies is crucial.
* **Data Backups and Recovery:** Regular data backups are a safety net. Implementing robust data backup and recovery strategies ensures data resilience in the face of unexpected events.

# **Future Considerations**

* A review column can be added to both the Employer Profile and Musician Profile for credibility.
* User will be able to upload their videos as sample work.
* Make a fully-functioning website, an android app and IOS app for maximum reach.
* Option for video-conferencing within application for features like Live video which can help users monetize their content. (Similar to Super chat)

# **References**

1. Piotr Kononow. (2018, August 6). How to reverse engineer a database with MySQL Workbench. Dataedo. <https://dataedo.com/kb/tools/mysql-workbench/how-to-reverse-engineer-database>
2. W3Schools. (2023). MySQL Tutorials. W3Schools. <https://www.w3schools.com/MySQL/default.asp>
3. Kymberly Fergusson.(2018, March 8). Entity Relationship Diagrams with draw.io. draw.io . <https://drawio-app.com/blog/entity-relationship-diagrams-with-draw-io/>
4. Leigh van Der Veen. (2021, March 11). What Is Cardinality in Data Modeling? The Theory and Practice of Database Cardinality. Vertabelo. <https://vertabelo.com/blog/cardinality-in-data-modeling/>
5. IBM. (2023). What is a data architecture. IBM . <https://www.ibm.com/topics/data-architecture>
6. Jackson (N.D). What Is Web Hosting? 4 Types Of Web Hosting You Must Know. VPS MALAYSIA. <https://www.vpsmalaysia.com.my/blog/what-is-web-hosting-4-types-of-web-hosting-you-must-know/>
7. Arijit Bose(2021, June 5). The Great Middling: Hybrid Pricing Model and its Growing SaaS Relevance. Chargebee blog. <https://www.chargebee.com/blog/hybrid-pricing-model-in-saas/>