

# **TuneWork's Business Insights**

## **Technical Report**

### **Group 3**

Sheila Hayati, Simran Chandak, Roxanne Li, Katie Li, Marcus Walton,  
Ruiying Liu, Xingyan Chen, Theo Bravos

# Data Dictionary

Agents		
Field Name	Data Type	Description
AgentID	Integer	Unique identifier for each agent, employee who manages bookings and coordinates between customers and entertainer
AgtFirstName	String	Agent's first name
AgtLastName	String	Agent's last name
AgtStreetAddress	String	Agent's street address
AgtCity	String	City of the agent's address
AgtState	String	State of the agent's address
AgtZipCode	String	ZIP code of the agent's address
AgtPhoneNumber	String	Agent's phone number
DateHired	Date	Date the agent was hired in
Salary	Decimal	Agent's salary
CommissionRate	Decimal	Agent's commission rate

Customers		
Field Name	Data Type	Description
CustomerID	Integer	Unique identifier for each customer, person/organization that books entertainment for an event
CustFirstName	String	Customer's first name
CustLastName	String	Customer's last name
CustStreetAddress	String	Customer's street address
CustCity	String	City of the customer's address

CustState	String	State of the customer's address
CustZipCode	String	ZIP code of the customer's address
CustPhoneNumber	String	Customer's phone number

Engagements		
Field Name	Data Type	Description
EngagementNumber	Integer	Unique identifier for the engagement
StartDate	Date	Start date of the engagement
EndDate	Date	End date of the engagement
StartTime	Time	Start time of the engagement
StopTime	Time	Stop time of the engagement
ContractPrice	Integer	Price agreed on contract for the engagement
AgentID	Integer	Foreign key to Agents table
CustomerID	Integer	Foreign key to Customers table
EntertainerID	Integer	Foreign key to Entertainers table

Entertainers		
Field Name	Data Type	Description
EntertainerID	Integer	Unique identifier for each entertainer, solo act/group that performs at events
EntStageName	String	Entertainer's stage name
EntSSN	String	Entertainer's social security number
EntStreetAddress	String	Entertainer's street address
EntCity	String	City of the entertainer's address
EntZipCode	String	ZIP code of the entertainer's address
EntPhoneNumber	String	Entertainer's phone number

EntWebPage	String	Entertainer's webpage
EntEmailAddress	String	Entertainer's email address
DateEntered	Date	Date the entertainer entered the system

<b>Musical_Styles</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
StyleID	Integer	Unique identifier for each musical style
StyleName	String	Name of the musical style

<b>Entertainer_Styles</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
EntertainerID	Integer	Foreign key to Entertainers table
StyleID	Integer	Foreign key to Musical_Styles table
StyleStrength	Integer	Strength of entertainer in the musical style

<b>Members</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
MemberID	Integer	Unique identifier for each member, person who is part of an entertainer group
MbrFirstName	String	Member's first name
MbrLastName	String	Member's last name
MbrPhoneNumber	String	Member's phone number
Gender	String	Member's gender

<b>Entertainer_Members</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>

EntertainerID	Integer	Foreign key to Entertainers table
MemberID	Integer	Foreign key to Members table
Status	Integer	Status of member in the entertainer group 1 = Primary, 2 = Supporting

<b>Musical_Preferences</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
CustomerID	Integer	Foreign key to Customers table
StyleID	Integer	Foreign key to Musical_Styles table
PreferenceSeq	Integer	Rank order of customer's musical style preference

<b>ztblDays</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
DateField	Date	Calendar date

<b>ztblMonths</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>
MonthYear	String	Name of month and year
YearNumber	Integer	Year as a number
MonthNumber	Integer	Month as a number
MonthStart	Date	Start date of each month
MonthEnd	Date	End date of each month
January-December	Boolean	True/False columns for each month

<b>ztblWeeks</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>

WeekStart	Date	Start date of the week
WeekEnd	Date	End date of the week

ztblSkipLabels		
Field Name	Data Type	Description
LabelCount	Integer	Number of labels to skip for formatting

# Purpose

TuneWorks is a mid-sized entertainment agency. As a part of McBrainsy & Co., we will improve their currently underperforming in-house analytics operations and provide recommendations based on their current data. The goal of this exploratory data analysis on the entertainment data set is to increase profitability, increase employee utilization, and build suggestions for future marketing and promotions. In order to complete these tasks, we must observe revenue methods, operations, demand, agent performance, and engagements. By tracking the most demanded music styles by customers and which entertainment groups fit those genres, we can recommend the best groups to support to increase revenue. Along with the performers, we must analyze individual agent operations as well. Noting the agents who hold the highest amount of and most popular entertainment groups, we can determine which agents are the most valuable to the company and attract the best talent. If necessary, the firing of agent staff could decrease costs. By comparing the times of several engagements along with their contract prices, we will decide whether the contract price matches the importance of the engagements. Through this methodology, we can conclude improvements to the entertainment company to maximize profits and optimize business operations.

# Background

## 1. Data Familiarization & Documentation

### a. Unclear or Ambiguous Fields

We identified the following fields that require clarification from the client:

- **Status (in Entertainer\_Members):** Interpreted as a role designation (e.g., Lead vs. Supporting Member), but not officially documented. Clarification on all potential status values would help interpretation
- **ztblSkipLabels:** Table present with numeric sequences, but unclear purpose. Likely linked to pagination or reporting logic

**Redundant Address Fields:** Several tables (e.g., Entertainers, Customers, Agents) include detailed address information. Clarity on why these are logged is required

### b. Data Quality Evaluation

The overall data quality is good, but the following issues were noted:

- No Missing Values in key tables (Engagements, Customers, Agents)
- **Outliers:**
  1. One agent (agentid = 9) has a salary of \$50, which appears to be a dummy or erroneous entry
  2. One engagement (engagementnumber = 99) has a very high contract price of over \$14,000, which may be valid but should be verified
- No duplicate primary keys found

### c. Data Completeness

There are opportunities for enhancing data collection:

- **Booking Channel / Source:** There is no indication of how a customer discovered the agency (e.g., referral, online). Capturing this could aid marketing analytics
- **Engagement Feedback or Ratings:** No post-engagement performance or satisfaction data is stored; valuable for quality control, sentiment analysis and recommender systems

### d. Structural Improvement

- **Time Dimensions:** The ztblDays, ztblWeeks, and ztblMonths tables look ideal for trend analysis. We can ensure they are joined via calendar keys for performance and better understanding of the timestamp data in context
- **Composite Keys:** In tables like Entertainer\_Styles and Musical\_Preferences we can create explicit composite primary keys as there's data that rely on more than one field

## 2. Data Preparation & Issue Resolution

Address columns validated using regex (regular expression) in the query:

```
SELECT AgentID, AgtStreetAddress, AgtCity, AgtState, AgtZipCode,
CASE WHEN AgtState ~ '^[A-Z]{2}$' THEN 'Valid' ELSE 'Invalid' END AS StateStatus,
CASE WHEN AgtZipCode ~ '^\\d{5}$' THEN 'Valid' ELSE 'Invalid' END AS ZipStatus,
CASE WHEN AgtStreetAddress ~ '^\\d+\\s+.+$' THEN 'Valid' ELSE 'Invalid' END AS
StreetStatus
FROM Agents;
```

	123 agentid	Agtstreetaddress	Agtcity	Agtstate	Agtzipcode	Agt statestatus	Agt zipstatus	Agt streetstatus
1	1	122 Spring River Drive	Redmond	WA	98053	Valid	Valid	Valid
2	2	66 Spring Valley Drive	Seattle	WA	98125	Valid	Valid	Valid
3	3	667 Red River Road	Bellevue	WA	98006	Valid	Valid	Valid
4	4	30301 - 166th Ave. N.E.	Seattle	WA	98125	Valid	Valid	Valid
5	5	908 W. Capital Way	Tacoma	WA	98413	Valid	Valid	Valid
6	6	16679 NE 41st Court	Seattle	WA	98125	Valid	Valid	Valid
7	7	4501 Wetland Road	Redmond	WA	98052	Valid	Valid	Valid

Similar checks done for Customers and Entertainers tables.  
No abnormal data related to addresses found.

Issue Resolution:

Issue	Resolution	Comments
Extremely low agent salary (\$50)	Retained and flagged for review	Could be a test/error record
Very high contract price engagement (\$14,105)	Retained as legitimate outlier	Skewed but cannot remove
Missing emails/websites in some Entertainers	Retained and accepted as is	Does not hamper analysis
Ambiguous fields like Status = 2	Interpreted as “supporting member”	Confirmed from pattern in data
Formatting inconsistencies	Address fields and names cleaned for uniformity	e.g., standard casing

# Queries & Analysis

## Optimize Operational Efficiencies

To evaluate TuneWorks' operational efficiency, we analyzed three metrics: (1) agent productivity; (2) entertainer utilization; and (3) average engagement duration. These metrics offer insights into the performance and workload distribution of both agents and entertainers, helping identify operational strengths as well as opportunities for optimization in resource usage and talent deployment.

For agent productivity, we analyzed the number of bookings and revenue generated by each agent. By joining Agents with Engagements, we calculated total bookings, total revenue, and average deal size per agent. This helped us identify high-performing agents and assess whether any agents might be underutilized or less effective at closing high-value deals.

### **SELECT**

```
a.AgentID,  
a.AgtFirstName, a.AgtLastName,  
COUNT(e.EngagementNumber) AS TotalBookings,  
SUM(e.ContractPrice) AS TotalRevenue,  
ROUND(AVG(e.ContractPrice), 2) AS AvgDealSize  
FROM Agents a  
LEFT JOIN Engagements e ON a.AgentID = e.AgentID  
GROUP BY a.AgentID, a.AgtFirstName, a.AgtLastName  
ORDER BY TotalRevenue DESC
```

	123 agentid	A-Z agtfirstname	A-Z agtlastname	123 totalbookings	123 totalrevenue	123 avgdealsize
1	9	Daffy	Dumbwit	0	[NULL]	[NULL]
2	3	Carol	Viescas	19	24,800	1,305.26
3	6	John	Kennedy	12	24,435	2,036.25
4	5	Marianne	Wier	18	22,635	1,257.5
5	1	William	Thompson	16	19,895	1,243.44
6	4	Karen	Smith	17	18,595	1,093.82
7	8	Maria	Patterson	15	12,825	855
8	7	Caleb	Viescas	8	10,645	1,330.63
9	2	Scott	Bishop	6	6,720	1,120

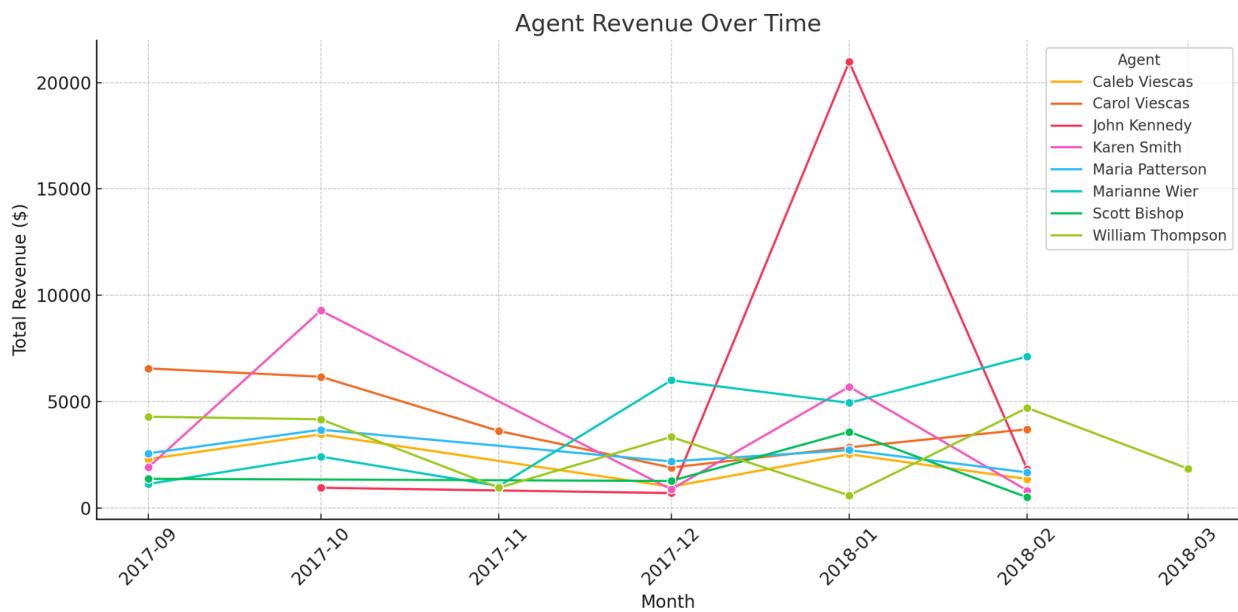
Among all agents, Carol Viescas and John Kennedy were the top performers, each generating over \$24,000 in revenue. John also secured the highest average deal size at \$2,036.25. In contrast, Daffy Dumbwit recorded no bookings, signaling potential onboarding or performance issues. These disparities highlight varying productivity levels across the team and suggest opportunities for improvement through targeted coaching, performance tracking, or reassignment of underperforming agents. Additionally, replicating the strategies of top performers could help raise the overall efficiency of the salesforce.

```

SELECT
  a.agentid,
  CONCAT(a.agtfirstname, ' ', a.agtlastname) AS agent_name,
  m.monthyear,
  COUNT(e.engagementnumber) AS total_bookings,
  ROUND(SUM(e.contractprice), 2) AS total_revenue,
  ROUND(AVG(e.contractprice), 2) AS avg_deal_size
FROM engagements e
JOIN agents a ON e.agentid = a.agentid
JOIN ztblMonths m ON e.startdate BETWEEN m.monthstart AND m.monthend
WHERE e.startdate IS NOT NULL
GROUP BY a.agentid, agent_name, m.monthyear
ORDER BY agent_name, m.monthyear;

```

	123 agentid	A-Z agent_name	A-Z monthlyyear	123 total_bookings	123 total_revenue	123 avg_deal_size
1	7	Caleb Viescas	December 2017	2	1,000	500
2	7	Caleb Viescas	February 2018	2	1,360	680
3	7	Caleb Viescas	January 2018	1	2,525	2,525
4	7	Caleb Viescas	October 2017	2	3,460	1,730
5	7	Caleb Viescas	September 2017	1	2,300	2,300
6	3	Carol Viescas	December 2017	2	1,900	950
7	3	Carol Viescas	February 2018	2	3,700	1,850



To expand on static metrics such as total revenue and average deal size, we analyzed each agent's revenue performance over time. A month-by-month breakdown reveals distinct patterns in agent

productivity. For instance, John Kennedy reached a peak of over \$21,000 in January 2018—the highest single-month revenue across all agents—demonstrating his ability to secure high-value deals during peak periods. In contrast, Carol Viescas maintained a stable performance, securing two bookings in both December 2017 and February 2018 with consistently high average deal sizes, suggesting reliable client engagement. Meanwhile, Caleb Viescas, despite consistently booking engagements, exhibited fluctuations in monthly revenue—from \$1,000 in December to \$3,460 in October—highlighting inconsistency in deal value. These temporal dynamics are vital in assessing not only who brings in revenue, but who does so consistently, allowing for more informed decisions on agent coaching, incentive alignment, and lead distribution.

For entertainer utilization, we measured the total number of engagements each entertainer completed, their total earnings from these engagements, and their average earnings per engagement. This metric helps identify which performers are in high demand, as well as those who may be underutilized. We used a LEFT JOIN between the Entertainers and Engagements tables to ensure all entertainers were included, even those with no bookings. The query grouped by entertainer ID and stage name, and calculated counts, sums, and averages to reveal utilization patterns across the talent pool.

## SELECT

```

ent.EntertainerID,
ent.EntStageName,
COUNT(e.EngagementNumber) AS TotalEngs,
SUM(e.ContractPrice) AS TotalEarnings,
ROUND(AVG(e.ContractPrice), 2) AS AvgEarningsPerEng
FROM Entertainers ent
LEFT JOIN Engagements e ON ent.EntertainerID = e.EntertainerID
GROUP BY ent.EntertainerID, ent.EntStageName
ORDER BY TotalEngs DESC
```

	123 entertainerid	A-Z entstagename	123 totalengs	123 totalearnings	123 avgearningspereng
1	1,008	Country Feeling	15	34,080	2,272
2	1,013	Caroline Coie Cuartet	11	15,070	1,370
3	1,001	Carol Peacock Trio	11	11,080	1,007.27
4	1,006	Modern Dance	10	14,600	1,460
5	1,003	JV & the Deep Six	10	17,150	1,715
6	1,004	Jim Glynn	9	3,030	336.67
7	1,010	Saturday Revue	9	11,550	1,283.33
8	1,007	Coldwater Cattle Company	8	14,875	1,859.38
9	1,011	Julia Schnebly	8	4,345	543.13
10	1,002	Topazz	7	6,620	945.71
11	1,005	Jazz Persuasion	7	5,480	782.86
12	1,012	Susan McLain	6	2,670	445
13	1,009	Katherine Ehrlich	0	[NULL]	[NULL]

Country Feeling led all entertainers with 15 engagements and the highest total earnings (\$34,080), along with the highest average earnings per engagement (\$2,272). In contrast, several

entertainers had relatively few engagements and lower total revenue, with one entertainer receiving no bookings at all. This imbalance suggests that TuneWorks could improve operational efficiency by promoting underutilized entertainers, offering targeted incentives to agents, or realigning booking decisions based on customer demand and performer strengths.

To evaluate how effectively TuneWorks utilizes its talent, we analyzed the average performance time per engagement in hours. This metric captures how long entertainers are actively scheduled for each booking, directly impacting availability, scheduling flexibility, and revenue potential. By factoring in both the number of engagement days and daily performance hours, we gain a more accurate view of operational workload. Shorter, higher-turnover engagements may increase reach and revenue efficiency, while longer bookings can tie up resources but offer more stable income per event.

```
SELECT
ROUND(AVG((EndDate - StartDate) + 1) *
CASE
    WHEN e.StopTime >= e.StartTime THEN
        EXTRACT(EPOCH FROM (e.StopTime - e.StartTime)) / 3600.0
    ELSE
        EXTRACT(EPOCH FROM (TIME '24:00:00' - e.StartTime + e.StopTime)) / 3600.0
    END
),
2
) AS AvgEngagementHours
FROM Engagements e
WHERE
    StartDate IS NOT NULL
    AND EndDate IS NOT NULL
    AND StartTime IS NOT NULL
    AND StopTime IS NOT NULL
    AND EndDate >= StartDate;
```

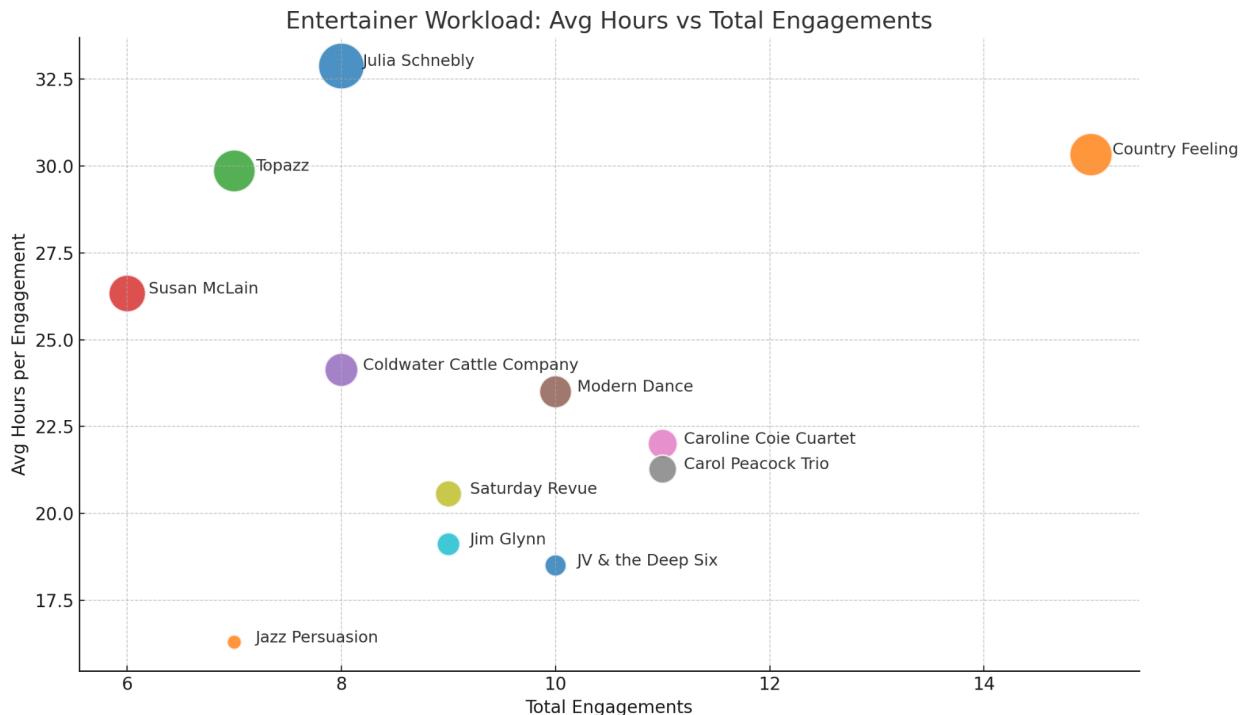


The average engagement at TuneWorks spans multiple days, but performers are typically scheduled for only a few hours each day, resulting in an average of 23.83 total performance hours per engagement. This structure allows for scheduling flexibility but may leave significant gaps of idle time between performances. To boost operational efficiency, TuneWorks could explore back-to-back engagements, bundled event packages, or filling open time slots with additional short bookings to maximize entertainer utilization and overall revenue.

To better understand the operational workload imposed on entertainers, we conducted a time series analysis of engagement durations across months. The results reveal that most bookings last between 4 to 8 days and average 3–4 hours per performance per day. Notably, December and January exhibit a concentration of longer, multi-day bookings, likely due to holidays and events. These patterns suggest TuneWorks is best optimized for sustained, multi-day events rather than quick-turnaround gigs. Incorporating shorter, high-frequency engagements could unlock additional availability in non-peak months and expand customer reach.

```
WITH engagement_durations AS (
    SELECT
        e.engagementnumber,
        e.entertainerid,
        ((e.enddate - e.startdate) + 1) AS num_days,
        CASE
            WHEN e.stoptime >= estarttime THEN
                EXTRACT(EPOCH FROM (e.stoptime - e starttime)) / 3600.0
            ELSE
                EXTRACT(EPOCH FROM (TIME '24:00:00' - e starttime + e stoptime)) / 3600.0
        END AS hours_per_day
    FROM engagements e
    WHERE e.startdate IS NOT NULL AND e.enddate IS NOT NULL
        AND e.starttime IS NOT NULL AND e.stoptime IS NOT NULL
        AND e.enddate >= e.startdate
)
SELECT
    ent.entertainerid,
    ent.entstagename,
    COUNT(ed.engagementnumber) AS total_engagements,
    ROUND(AVG(ed.num_days * ed.hours_per_day), 2) AS avg_hours_per_engagement
FROM entertainers ent
LEFT JOIN engagement_durations ed ON ent.entertainerid = ed.entertainerid
GROUP BY ent.entertainerid, ent.entstagename
HAVING ROUND(AVG(ed.num_days * ed.hours_per_day), 2) is not NULL
ORDER BY avg_hours_per_engagement DESC;
```

	123 entertainerid	A-Z entstagename	123 total_engagements	123 avg_hours_per_engagement
1	1,011	Julia Schnebly	8	32.88
2	1,008	Country Feeling	15	30.33
3	1,002	Topazz	7	29.86
4	1,012	Susan McLain	6	26.33
5	1,007	Coldwater Cattle Company	8	24.13
6	1,006	Modern Dance	10	23.5
7	1,013	Caroline Coie Cuartet	11	22
8	1,001	Carol Peacock Trio	11	21.27



### Understanding Supply & Demand - Entertainers vs. Customers

In order to maximize profit, we want to make sure there are no discrepancies between the type of music customers are demanding and the musical styles of TuneWork's current talent roster. The below queries break down the musical preferences per genre and the number of entertainers that specialize in each genre:

```
select ms.stylename, count(mp.styleid) pref_count from musical_styles ms
join musical_preferences mp
on ms.styleid = mp.styleid
group by ms.stylename
order by count(mp.styleid) desc;
```

A-Z stylename	123 pref_count
Standards	16
Rhythm and Blues	12
Contemporary	12
Jazz	12
Classic Rock & Roll	8
Classical	8
Salsa	8
40's Ballroom Music	8
Modern Rock	8

```
select ms.stylename, count(es.styleid) entertainer_count from entertainer_styles es
join musical_styles ms
on ms.styleid = es.styleid
group by ms.stylename
order by count(es.styleid) desc;
```

A-Z stylename	123 entertainer_count
Classical	12
Standards	12
Jazz	8
Contemporary	8
Salsa	8
Show Tunes	8
Variety	8
Country	8
Rhythm and Blues	8

With musical preference as a common column, we can combine the two tables and subtract the preference count from the entertainer count to understand if our entertainers align with the musical preferences of customers. The below query outlines the musical genres that have an oversupply of entertainers:

```
with style_popularity as (select ms.stylename,
count(mp.styleid) pref_count from musical_styles ms
join musical_preferences mp
on ms.styleid = mp.styleid
group by ms.stylename
order by count(mp.styleid) desc),
```

```

entertainer_styles as (select ms.stylename stylename,
count(es.styleid) entertainer_count from entertainer_styles es
join musical_styles ms
on ms.styleid = es.styleid
group by ms.stylename
order by count(es.styleid) desc)
select sp.stylename, sp.pref_count, ents.entertainer_count,
(sp.pref_count-ents.entertainer_count) difference from style_popularity sp
join entertainer_styles ents
on sp.stylename = ents.stylename
where (sp.pref_count-ents.entertainer_count) < 0
order by (sp.pref_count-ents.entertainer_count);

```

A-Z stylename	123 pref_count	123 entertainer_count	123 difference
Classical	8	12	-4
Country	4	8	-4
Folk	4	8	-4
Variety	4	8	-4
60's Music	4	8	-4
Chamber Music	4	8	-4

According to the above table, the musical genres where we have more entertainers than customers of that musical preference are Classical, Country, Folk, Variety, 60's Music and Chamber Music. With this in mind, it makes sense to take a closer look at entertainers who specialize in these 6 musical styles - analyzing the number of engagements they book and their total earnings from these engagements.

To further enrich our analysis of customer demand versus entertainer supply, we conducted a **time series evaluation of customer preferences across musical styles**, tracking how preference scores evolved month by month. This trend-based view revealed that styles such as *Jazz*, *Rhythm & Blues*, and *Standards* sustained strong or rising preference scores over time, suggesting consistent audience appeal. In contrast, genres like *Chamber Music*, *60's Music*, and *Modern Rock* showed sharp fluctuations, indicating inconsistent consumer interest. By isolating both top-trending and most divergent styles, we gain valuable insight into which genres deserve continued investment, roster expansion, or potential repositioning. These dynamics offer a powerful foundation for proactive, seasonally aligned marketing and booking strategies.

```

WITH preferences_scored AS (
  SELECT
    mp.customerid, mp.styleid,
    CASE
      WHEN preferenceseq = 1 THEN 3
      WHEN preferenceseq = 2 THEN 2
)

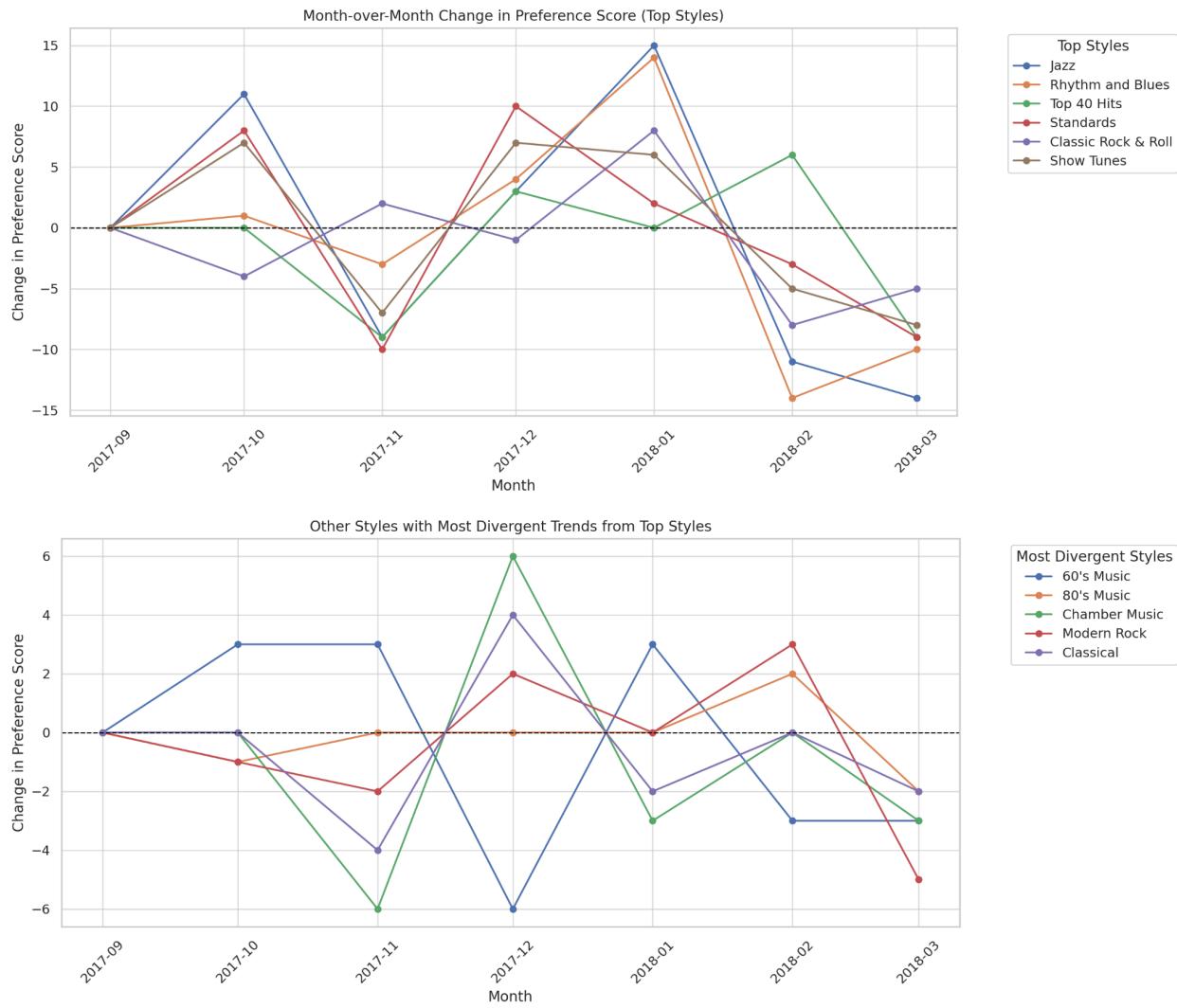
```

```

WHEN preferenceseq = 3 THEN 1
END AS score
FROM musical_preferences mp
),
monthly_prefs AS (
SELECT
m.monthyear,
ms.stylename,
SUM(ps.score) AS total_score
FROM preferences_scored_ps
JOIN customers c ON ps.customerid = c.customerid
JOIN musical_styles ms ON ps.styleid = ms.styleid
JOIN engagements e ON c.customerid = e.customerid
JOIN ztblMonths m ON e.startdate BETWEEN m.monthstart AND m.monthend
GROUP BY m.monthyear, ms.stylename
)
SELECT *
FROM monthly_prefs
ORDER BY stylename, monthyear;

```

	A-Z monthyear	A-Z stylename	123 total_score
1	December 2017	40's Ballroom Music	2
2	February 2018	40's Ballroom Music	1
3	January 2018	40's Ballroom Music	2
4	October 2017	40's Ballroom Music	2
5	December 2017	60's Music	3
6	February 2018	60's Music	3
7	January 2018	60's Music	6
8	November 2017	60's Music	9



To further explore how well TuneWorks aligns entertainer supply with customer demand, we conducted a monthly breakdown comparing preferences, engagements, and revenue across musical styles. For example, in December 2017, while styles like Classic Rock & Roll, Contemporary, and Classical each received multiple preferences, the most engaged style was 60's Music—despite receiving only one explicit preference. Similarly, Jazz, a top-performing style in engagement count and revenue (\$3,420), did not appear at the top of the preference list for that month. This mismatch suggests that some genres generate high revenue without proportional expressed preference, while others with solid preference counts, such as 40's Ballroom Music or Chamber Music, lag in actual bookings. Visualizing preferences and engagements side-by-side over time reveals persistent gaps, reinforcing the need for dynamic realignment between talent offerings and evolving customer interests.

**SELECT**  
**m.monthyear**

```

ms.stylename,
COUNT(mp.customerid) AS preference_count
FROM musical_preferences mp
JOIN musical_styles ms ON mp.styleid = ms.styleid
JOIN customers c ON mp.customerid = c.customerid
JOIN engagements e ON c.customerid = e.customerid
JOIN ztblMonths m ON e.startdate BETWEEN m.monthstart AND m.monthend
GROUP BY m.monthyear, ms.stylename
ORDER BY m.monthyear, ms.stylename;

```

	A-Z monthyear	A-Z stylename	123 preference_count
1	December 2017	40's Ballroom Music	2
2	December 2017	60's Music	1
3	December 2017	70's Music	1
4	December 2017	80's Music	1
5	December 2017	Chamber Music	2
6	December 2017	Classic Rock & Roll	2
7	December 2017	Classical	2
8	December 2017	Contemporary	2

```

SELECT
m.monthyear,
ms.stylename,
COUNT(DISTINCT e.engagementnumber) AS num_engagements
FROM engagements e
JOIN entertainer_styles es ON e.entertainerid = es.entertainerid
JOIN musical_styles ms ON es.styleid = ms.styleid
JOIN ztblMonths m ON e.startdate BETWEEN m.monthstart AND m.monthend
GROUP BY m.monthyear, ms.stylename
ORDER BY m.monthyear, ms.stylename;

```

	A-Z monthyear	A-Z stylename	123 num_engagements
1	December 2017	60's Music	4
2	December 2017	Chamber Music	2
3	December 2017	Classic Rock & Roll	2
4	December 2017	Classical	4
5	December 2017	Contemporary	3
6	December 2017	Country	2
7	December 2017	Folk	3
8	December 2017	Jazz	3

```

SELECT
m.monthyear,
ms.stylename,
SUM(e.contractprice) AS total_revenue

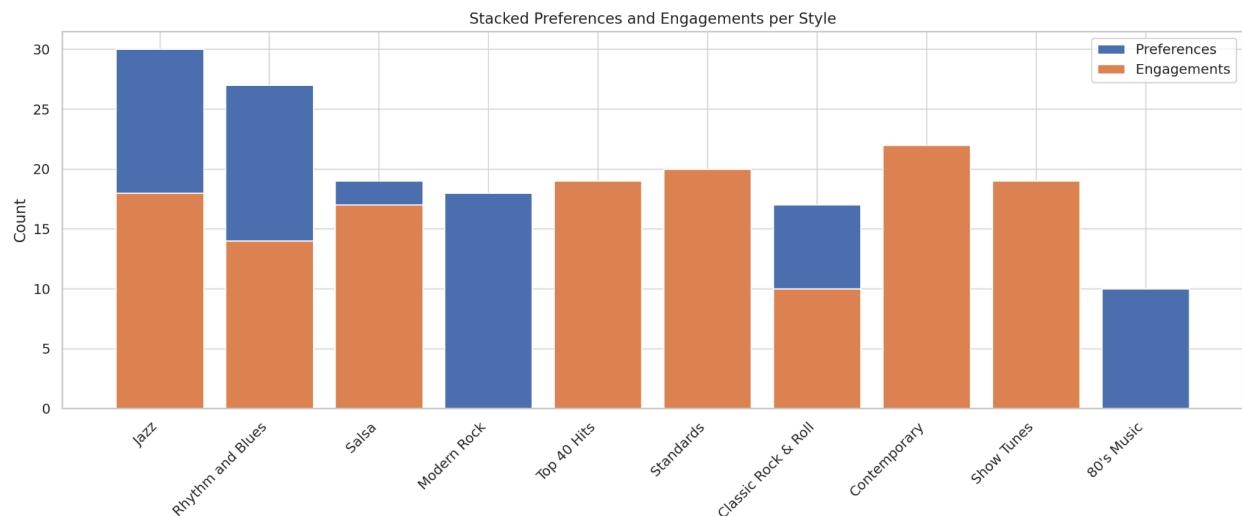
```

```

FROM engagements e
JOIN entertainer_styles es ON e.entertainerid = es.entertainerid
JOIN musical_styles ms ON es.styleid = ms.styleid
JOIN ztblMonths m ON e.startdate BETWEEN m.monthstart AND m.monthend
GROUP BY m.monthyear, ms.stylename
ORDER BY m.monthyear, ms.stylename;

```

	A-Z monthyear	A-Z stylename	123 total_revenue
1	December 2017	60's Music	4,490
2	December 2017	Chamber Music	1,285
3	December 2017	Classic Rock & Roll	3,340
4	December 2017	Classical	2,075
5	December 2017	Contemporary	3,060
6	December 2017	Country	1,150
7	December 2017	Folk	1,020
8	December 2017	Jazz	3,420



```

SELECT e.entstagename, count(distinct en.engagementnumber) num_engagements FROM
entertainers e
join engagements en
on e.entertainerid = en.entertainerid
where e.entstagename in (select distinct e.entstagename from entertainer_styles es
join entertainers e
on e.entertainerid = es.entertainerid
join musical_styles ms

```

```

on es.styleid = ms.styleid
where ms.stylename in ('Classical','Country','Folk','Variety','60"s Music','Chamber Music'))
group by e.entertainerid, e.entstagename
order by count(distinct en.engagementnumber);

```

A-Z entstagename	123 num_engagements
Susan McLain	6
Topazz	7
Coldwater Cattle Company	8
Julia Schnebly	8
Jim Glynn	9
Modern Dance	10
JV & the Deep Six	10
Country Feeling	15

Using the above query, we see that the entertainers of “low-popularity” music styles with the 3 least number of engagements are Susan McLain, Topazz and Coldwater Cattle Company. If we are to assess on booking frequency alone, TuneWork’s can consider dropping these 3 entertainers from their roster. However, analyzing this through a profit generating point of view, it may be worthwhile to look deeper into how much money each entertainer is bringing in - both in terms of contract price and commissions to agents.

```

with commissions as (select distinct a.agentid, e.entstagename, a.commissionrate,
en.contractprice,
round(a.commissionrate * en.contractprice) commission_total from agents a
join engagements en
on a.agentid = en.agentid
join entertainers e
on e.entertainerid = en.entertainerid
where e.entstagename in (select distinct e.entstagename from entertainer_styles es
join entertainers e
on e.entertainerid = es.entertainerid
join musical_styles ms
on es.styleid = ms.styleid
where ms.stylename in ('Classical','Country','Folk','Variety','60"s Music','Chamber Music'))
group by a.agentid, e.entertainerid, e.entstagename, a.commissionrate, en.contractprice
order by e.entstagename)
select entstagename, sum(commission_total) total_commission_earned,
sum(contractprice) total_engagement_earnings from commissions
group by entstagename
order by sum(commission_total);

```

When we break down the total contract price (engagement) earnings and total commission fees to agents earned per entertainer, we see that because of varying contract prices/commission rates, the entertainers with the least bookings do not necessarily correspond to those with the least earnings. The 3 entertainers with the least engagement earnings are Jim Glynn, Susan McLain and Julia Schnebly. To close the gap in supply and demand, it makes sense to consider dropping some artists from the roster in order to reallocate costs to artists of high demand musical styles (Standards, R&B, Contemporary, Jazz, Classic Rock).

```
with style_popularity as (select ms.stylesname,
count(mp.styleid) pref_count from musical_styles ms
join musical_preferences mp
on ms.styleid = mp.styleid
group by ms.stylesname
order by count(mp.styleid) desc),
entertainer_styles as (select ms.stylesname stylename,
count(es.styleid) entertainer_count from entertainer_styles es
join musical_styles ms
on ms.styleid = es.styleid
group by ms.stylesname
order by count(es.styleid) desc)
select sp.stylesname, sp.pref_count, ents.entertainer_count,
(sp.pref_count-ents.entertainer_count) difference from style_popularity sp
join entertainer_styles ents
on sp.stylesname = ents.stylename
where (sp.pref_count-ents.entertainer_count)>0
order by (sp.pref_count-ents.entertainer_count);
```

A-Z stylename ▾	123 pref_count ▾	123 entertainer_count ▾	123 difference ▾
Standards	16	12	4
Rhythm and Blues	12	8	4
Contemporary	12	8	4
Jazz	12	8	4
Classic Rock & Roll	8	4	4

---

## Agents to be fired/optimize their pay - incentivize

```

select
a.agentid,
a.agtfirstname,
a.agtlastname,
count(e.engagementnumber) as total_engagements,
sum(e.contractprice) as total_revenue,
avg(e.contractprice) as avg_contract_price,
min(e.contractprice) as min_contract_price,
max(e.contractprice) as max_contract_price
from agents a
left join engagements e using(agentid)
group by
a.agentid,
a.agtfirstname,
a.agtlastname
order by
total_revenue desc;

```

	<a href="#">123 agentid</a>	<a href="#">A-Z agtfirstname</a>	<a href="#">A-Z agtlastname</a>	<a href="#">123 total_engagements</a>	<a href="#">123 total_revenue</a>	<a href="#">123 avg_contract_price</a>	<a href="#">123 min_contract_price</a>	<a href="#">123 max_contract_price</a>
1	9	Daffy	Dumbwit	0	[NULL]	[NULL]	[NULL]	[NULL]
2	3	Carol	Viescas	19	24,800	1,305.2631578947	230	3,650
3	6	John	Kennedy	12	24,435	2,036.25	275	14,105
4	5	Marianne	Wier	18	22,635	1,257.5	275	2,930
5	1	William	Thompson	16	19,895	1,243.4375	185	2,750
6	4	Karen	Smith	17	18,595	1,093.8235294118	200	3,800
7	8	Maria	Patterson	15	12,825	855	110	2,450
8	7	Caleb	Viescas	8	10,645	1,330.625	410	2,525
9	2	Scott	Bishop	6	6,720	1,120	500	2,930

```

select a.agentid, a.agtfirstname, a.agtlastname,
count(distinct e.customerid) as unique_customers
from agents a
left join engagements e using(agentid)
group by a.agentid, a.agtfirstname, a.agtlastname
order by unique_customers DESC;

```

	<a href="#">123 agentid</a>	<a href="#">A-Z agtfirstname</a>	<a href="#">A-Z agtlastname</a>	<a href="#">123 unique_customers</a>
1	1	William	Thompson	12
2	4	Karen	Smith	12
3	8	Maria	Patterson	11
4	3	Carol	Viescas	11
5	5	Marianne	Wier	10
6	6	John	Kennedy	8
7	7	Caleb	Viescas	7
8	2	Scott	Bishop	4
9	9	Daffy	Dumbwit	0

These results show each agent's sales performance, highlighting Carol Viescas, John Kennedy, and Marianne Wier as the top earners by total revenue, while Daffy Dumbwit recorded no engagement activity. Notably, John Kennedy achieved a high average contract price and the highest single contract value, suggesting a proficiency with larger deals. In contrast, other active agents demonstrate varied engagement levels and contract values. The bottom query shows the number of unique customers each agent has.

---

### **Prorating Salaries**

Prorating the salaries ensures a fairer comparison between agents and their true profitability, especially when they have worked for different lengths of time. We did this by adjusting the salary cost to consider only the portion relevant to their active engagement period, leading to the net result more accurately showing their financial impact to the time they generated revenue. This prevents agents who only worked part of a period from being unfairly brought down in performance calculations.

```
with agent_periods as (
  select
    a.agentid,
    a.agtfirstname,
    a.agtlastname,
    min(e.startdate) as period_start,
    max(e.enddate) as period_end
  from agents a
  left join engagements e
    on a.agentid = e.agentid
  group by a.agentid, a.agtfirstname, a.agtlastname
),
metrics as (
  select
    ap.agentid,
    ap.agtfirstname,
    ap.agtlastname,
    coalesce(sum(e.contractprice), 0) as total_revenue,
    coalesce(sum(e.contractprice * a.commissionrate), 0) as total_commission,
    case
      when ap.period_start is not null
      then a.salary
        * ((ap.period_end - ap.period_start + 1)::numeric / 365)
      else 0
    end as prorated_salary
  from agent_periods ap
  join agents a
    on ap.agentid = a.agentid
  left join engagements e
```

```

on ap.agentid = e.agentid
group by
    ap.agentid,
    ap.agtfirstname,
    ap.agtlastname,
    ap.period_start,
    ap.period_end,
    a.salary
)
select
    agentid,
    agtfirstname,
    agtlastname,
    total_revenue,
    total_commission,
    round(prorated_salary, 2) as prorated_salary,
    total_revenue
        - ( total_commission + prorated_salary )
        as net_result,
    case
        when total_revenue
            - ( total_commission + prorated_salary ) >= 0
        then 'net positive'
        else 'net negative'
    end as status
from metrics
order by net_result desc;

```

	agentid	agtfirstname	agtlastname	total_revenue	total_commission	prorated_salary	net_result	status
1	5	Marianne	Wier	22,635	1,018.5750404745	11,813.7	9,802.7263293885	net positive
2	6	John	Kennedy	24,435	1,466.0999672301	13,200	9,768.9000327699	net positive
3	3	Carol	Viescas	24,800	1,240.0000184774	14,136.99	9,423.0136801527	net positive
4	4	Karen	Smith	18,595	1,022.7249944583	10,909.59	6,662.6859644459	net positive
5	1	William	Thompson	19,895	795.7999822125	16,972.6	2,126.5972780615	net positive
6	7	Caleb	Viescas	10,645	372.5750015862	10,172.05	100.3702038932	net positive
7	9	Daffy	Dumbwit	0	0	0	0	net positive
8	8	Maria	Patterson	12,825	512.9999885336	12,739.73	-427.7260159308	net negative
9	2	Scott	Bishop	6,720	268.7999939919	12,131.51	-5,680.3068433069	net negative

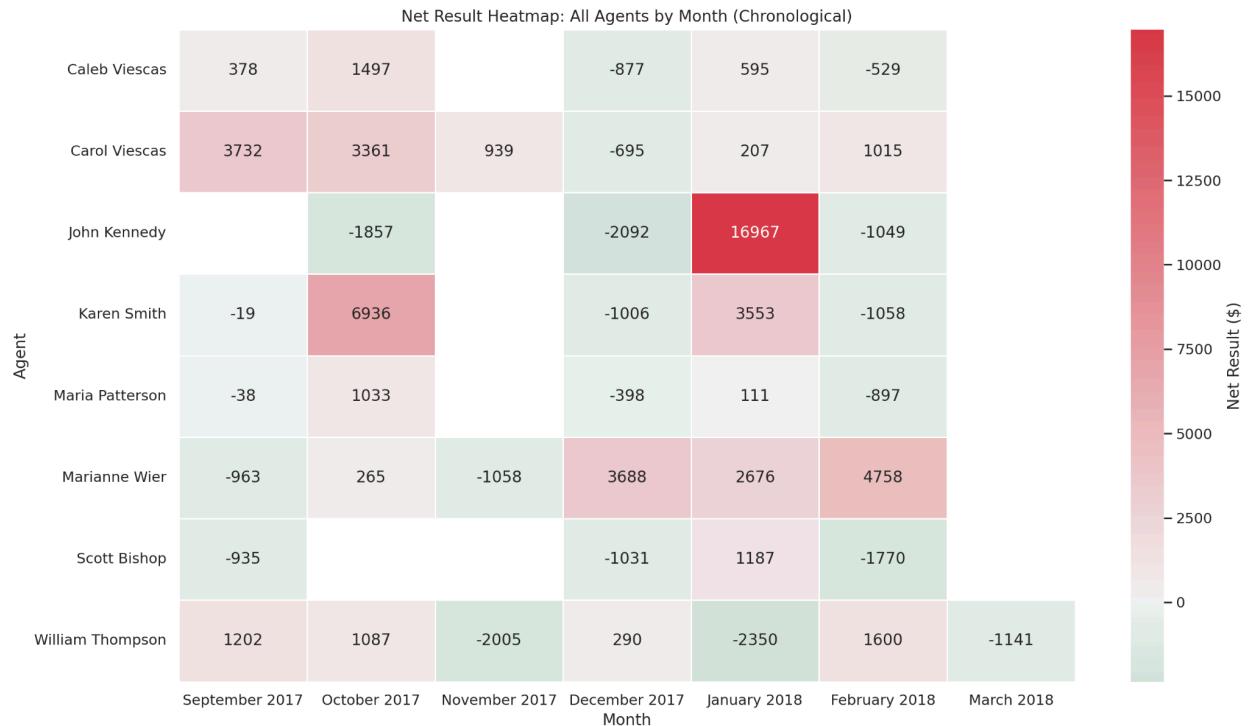
These results indicate that most agents, such as Marianne Wier and John Kennedy, who lead with over \$9,700 each in net positive contributions, are profitable for the company after accounting for their revenue, commissions, and prorated salaries. However, Maria Patterson and Scott Bishop are classified as 'net negative', suggesting that their costs exceeded their revenue contributions during the analyzed period. This implies the need for a closer examination of the engagements and cost structures for these two agents to understand the deficit, while also affirming the strong performance of the top earners.

To deepen our understanding of profitability beyond cumulative values, we performed a time series analysis of each agent's monthly net result—defined as revenue minus commission and

prorated salary. The heatmap highlights stark contrasts in agent contributions over time. While John Kennedy recorded a standout \$16,967 net gain in January 2018, he had negative margins in several other months, underscoring volatility. Caleb Viescas shifted from negative margins in December and February to positive performance in January and October. Meanwhile, Carol Viescas maintained consistent net positive margins across most months. This temporal view of agent profitability provides clearer guidance for performance management, enabling the company to reward sustained contributors and reevaluate those who incur repeated monthly losses.

```
WITH agent_monthly AS (
    SELECT
        a.agentid,
        CONCAT_WS(' ', a.agtfirstname, a.agtlastname) AS agent_name,
        m.monthyear,
        COUNT(e.engagementnumber) AS total_bookings,
        SUM(e.contractprice) AS total_revenue,
        SUM(e.contractprice * a.commissionrate) AS total_commission,
        a.salary / 12.0 AS monthly_salary
    FROM agents a
    LEFT JOIN engagements e ON e.agentid = a.agentid
    LEFT JOIN ztblMonths m ON e.startdate BETWEEN m.monthstart AND m.monthend
    WHERE e.startdate IS NOT NULL
    GROUP BY a.agentid, agent_name, m.monthyear, a.salary
)
SELECT
    agentid,
    agent_name,
    monthyear,
    total_bookings,
    ROUND(total_revenue) AS total_revenue,
    ROUND(total_commission) AS total_commission,
    ROUND(monthly_salary) AS monthly_salary,
    ROUND(total_revenue - total_commission - monthly_salary) AS net_result,
    CASE
        WHEN total_revenue - total_commission - monthly_salary >= 0 THEN 'net positive'
        ELSE 'net negative'
    END AS status
FROM agent_monthly
ORDER BY agent_name, monthyear;
```

	agentid	agent_name	monthyear	total_bookings	total_revenue	total_commission	monthly_salary	net_result	status
1	7	Caleb Viescas	December 2017	2	1,000	35	1,842	-877	net negative
2	7	Caleb Viescas	February 2018	2	1,360	48	1,842	-529	net negative
3	7	Caleb Viescas	January 2018	1	2,525	88	1,842	595	net positive
4	7	Caleb Viescas	October 2017	2	3,460	121	1,842	1,497	net positive
5	7	Caleb Viescas	September 2017	1	2,300	81	1,842	378	net positive
6	3	Carol Viescas	December 2017	2	1,900	95	2,500	-695	net negative
7	3	Carol Viescas	February 2018	2	3,700	185	2,500	1,015	net positive
8	3	Carol Viescas	January 2018	3	2,850	143	2,500	207	net positive



## Strategic Marketing Campaigns

For effective marketing campaigns, we can find star entertainers – entertainers who are popular — and recommend them to the consumers with the styles they want.

```

select e.entertainerid, e.entstagename, ms.stylename, count(e2.engagementnumber) as
  num_of_contract from entertainers e
left join engagements e2
on e.entertainerid = e2.entertainerid
left join entertainer_styles es
on e.entertainerid = es.entertainerid
left join musical_styles ms
on es.styleid = ms.styleid
group by e.entertainerid, e.entstagename, ms.stylename
order by num_of_contract desc

```

123 entertainerid	A-Z entstagename	A-Z stylename	123 num_of_contract
1,008	Country Feeling	Country	15
1,008	Country Feeling	60's Music	15
1,013	Caroline Coie Cuartet	Contemporary	11
1,001	Carol Peacock Trio	Show Tunes	11
1,001	Carol Peacock Trio	Contemporary	11
1,013	Caroline Coie Cuartet	Jazz	11
1,001	Carol Peacock Trio	Standards	11
1,006	Modern Dance	Variety	10
1,003	JV & the Deep Six	60's Music	10
1,003	JV & the Deep Six	Classic Rock & Roll	10
1,006	Modern Dance	Salsa	10
1,006	Modern Dance	Top 40 Hits	10

We further create a rank based on the num\_of\_contract for each stylename, and find the most popular entertainer for each style.

with *ep* as (

```
select e.entertainerid, e.entstagename, ms.stylename,
count(e2.engagementnumber) as num_of_contract,
rank()over (partition by ms.stylename order by count(e2.engagementnumber) desc )
from entertainers e
left join engagements e2
on e.entertainerid = e2.entertainerid
left join entertainer_styles es
on e.entertainerid = es.entertainerid
left join musical_styles ms
on es.styleid = ms.styleid
group by e.entertainerid, e.entstagename, ms.stylename
order by num_of_contract desc)
select * from ep
where rank = 1
```

123 entertainerid	A-Z entstagename	A-Z stylename	123 num_of_contract	123 rank
1,008	Country Feeling	Country	15	1
1,008	Country Feeling	60's Music	15	1
1,013	Caroline Coie Cuartet	Jazz	11	1
1,001	Carol Peacock Trio	Contemporary	11	1
1,013	Caroline Coie Cuartet	Contemporary	11	1
1,001	Carol Peacock Trio	Show Tunes	11	1
1,001	Carol Peacock Trio	Standards	11	1
1,006	Modern Dance	Variety	10	1
1,006	Modern Dance	Salsa	10	1
1,003	JV & the Deep Six	Classic Rock & Roll	10	1
1,006	Modern Dance	Top 40 Hits	10	1
1,010	Saturday Revue	70's Music	9	1
1,004	Jim Glynn	Folk	9	1
1,007	Coldwater Cattle Compa	Country Rock	8	1
1,011	Julia Schnebly	Classical	8	1
1,011	Julia Schnebly	Chamber Music	8	1
1,002	Topazz	Rhythm and Blues	7	1
1,002	Topazz	Motown	7	1
1,005	Jazz Persuasion	Rhythm and Blues	7	1

When we create poster or mail campaigns to consumers we are able to mention these entertainers and let the consumers get interested. We will Increase promotion of star entertainers through posters and visual materials, and record and share sample performance videos to give consumers a preview of their style and appeal.

Another important need is to find consumers' needs. The following code provides the preference\_score for each style, therefore we are able to let the consumer know that we offer entertainers that align with consumer preferences.

```
with cs as (
select mp.styleid,
case
    when preferenceseq =1 then 3
    when preferenceseq =2 then 2
    when preferenceseq =3 then 1
end as score
from musical_preferences mp )
select cs.styleid, ms.stylesname ,sum(cs.score) as preference_score from cs
left join musical_styles ms
on cs.styleid = ms.styleid
group by cs.styleid, ms.stylesname
order by preference_score desc
```

	styleid	stylename	preference_score
1	21	Standards	10
2	15	Jazz	8
3	10	Contemporary	7
4	19	Rhythm and Blues	7
5	22	Top 40 Hits	6
6	8	Classic Rock & Roll	5
7	20	Show Tunes	5
8	14	Chamber Music	3
9	11	Country Rock	3
10	24	Salsa	3
11	3	60's Music	3
12	7	Classical	3

We want to do Email Marketing focusing on promoting genres like standards, jazz, contemporary, and rhythm for new consumers and Establish a long-term relationship with former consumers by recommending relevant music genres and seeking for next opportunities.

---

### **Whether length of engagements are worth their contract price**

To better evaluate contract profitability, we can assess how engagement duration relates to its hourly value. By calculating the price per hour, we can identify whether longer engagements provide better value or if short-term bookings are more profitable. This helps in setting smarter pricing tiers and optimizing package offers.

```
select
engagementnumber, contractprice,
((EndDate - StartDate) * 24 +
```

```

extract(hour from stoptime) - extract(hour from starttime)) as DurationHours,
contractprice / ((enddate - startdate) * 24 +
extract(hour from stoptime) - extract(hour from starttime)) as PricePerHour
from engagements e
where ((enddate - startdate) * 24 +
extract(hour from StopTime) - extract(hour from starttime)) > 0
order by PricePerHour desc
limit 10;

```

	engagementnumber	contractprice	durationhours	priceperhour
1	82	950	5	190
2	11	950	6	158.33333333333
3	51	650	5	130
4	62	500	5	100
5	43	140	2	70
6	48	950	30	31.666666666667
7	95	1,550	53	29.2452830189
8	97	110	4	27.5
9	92	1,925	77	25

Introduce a tiered pricing model that maintains strong hourly value across engagement lengths. Shorter bookings can continue to command premium rates, but longer engagements should have a minimum hourly floor to protect against underpricing. This balances profitability with customer value and supports more predictable revenue from extended bookings.

# Recommendations

To drive sustainable growth and operational efficiency, TuneWorks should align talent supply, customer preferences, and agent performance across both current metrics and evolving trends. Our findings support the following strategic recommendations:

## 1. Realign Talent Roster with Evolving Customer Preferences

Static preference counts highlight genres like Jazz, Standards, R&B, and Classic Rock as customer favorites. However, time series analysis reveals *shifts in preference momentum*—with some styles consistently trending upward (e.g., Contemporary, R&B) and others (e.g., Chamber Music, 60's Music) showing volatile or declining interest. TuneWorks should:

- Expand roster and promotion for styles with sustained or growing preference trends.
- Phase out or reposition entertainers in styles that underperform in both preference and engagement over time.
- Use monthly preference score tracking to adjust seasonal marketing efforts proactively.

## 2. Enhance Performer Utilization Based on Engagement Duration and Demand

Most bookings span several days with 3–4 hours of performance per day. While high-revenue, long-duration engagements dominate peak periods (e.g., December–January), this ties up performers for extended times. To improve efficiency and attract a broader range of clients, we recommend offering tiered engagement packages that provide more flexibility in booking length and pricing. These packages could include:

- Standard Engagements: Multi-day events, ideal for festivals or conferences.
- Short-Term Bookings: 1–2 day events for clients who need performers for a single weekend or evening.
- Hourly or Partial-Day Options: For smaller events, such as weddings, corporate parties, or weekday gigs.

Introducing flexible booking options will help fill downtime during off-peak periods, increase overall performer utilization, and appeal to clients with more diverse needs and budgets.

## 3. Reward Agent Performance Based on Net Contribution Over Time

While agents like John Kennedy and Marianne Wier are top earners based on total revenue, our monthly analysis of net results (revenue – commission – prorated salary) reveals that not all high-revenue agents are consistently profitable. TuneWorks should:

- Use monthly net profitability as a core KPI to evaluate agent performance.
- Incentivize agents with bonuses tied to sustained profitability, not just revenue spikes.
- Reallocate leads from agents with repeated net losses to those demonstrating consistent gains.

#### 4. Address Supply-Demand Gaps in Booking Behavior

Our combined analysis of preferences, engagements, and revenue across styles and months shows misalignments: some low-preference styles are overbooked, while high-preference styles (e.g., Contemporary, R&B) may be underrepresented in engagements. TuneWorks should:

- Match customer preferences with booking trends monthly to close demand gaps.
- Avoid overbooking entertainers in low-demand genres, even if they are available.
- Highlight top entertainers in popular styles in marketing materials and recommendation engines.

#### 5. Introduce Time-Aware Reporting Dashboards

Given the valuable insights gained through time series analysis, we recommend the creation of internal dashboards that track:

- Monthly revenue, booking count, and net result per agent
- Preference score trends and engagement counts per musical style
- Performer utilization rates over time