TuneWorks Operations & Customer Insight Analysis Technical Report

University of Southern California DSO 555 Advanced SQL for Business Analysts

Team G4
Qianying Fu, Yijia Dong, Hsu Sint, Charudisha Ashok Ashjay, Kaveri Verma, Sahiti Reddy, Shiv Gupta, Tzu Han Lin

1. Data Dictionary

🟨 Indicates ambiguous columns and tables which further clarification is needed from the client. Refer to data quality notes (2D) for details.

- **Agents:** Stores information about agents who manage customer bookings.

Column Name	Description
agentid (Primary key)	Unique ID for the agent
agtfirstname	Agent's first name
agtlastname	Agent's last name
agtstreetaddress	Agent's street address
agtcity	City of the agent
agtstate	State of the agent (2-letter code)
agtzipcode	Zip code of the agent
agtphonenumber	Agent's phone number
datehired	Date the agent was hired (yyyy-mm-dd)
salary	Base salary of the agent
commissionrate	Commission rate (decimal format)

- **Customers:** Stores customer information for booking engagements.

Column Name	Description
customerid (Primary key)	Unique ID for the customer
custfirstname	Customer's first name
custlastname	Customer's last name
custstreetaddress	Customer's street address
custcity	City of the customer
custstate	State of the customer (2-letter code)
custzipcode	Zip code of the customer
custphonenumber	Customer's phone number

- **Entertainers:** Holds data on performers, including stage names and contact information.

Column Name	Description
entertainerid (Primary key)	Unique ID for the entertainer
entstagename	Entertainer's stage name
entssn	SSN/registration number
entstreetaddress	Street address
entcity	City
entstate	State
entzipcode	Zip code
entphonenumber	Phone number
entwebpage	Webpage URL
entemailaddress	Email address
dateentered	Date entertainer entered into system (yyyy-mm-dd)

- **Members:** Lists individual members with basic contact info.

Column Name	Description
memberid (Primary key)	Unique ID for member
mbrfirstname	Member's first name
mbrlastname	Member's last name
mbrphonenumber	Member's phone number
gender	Gender (M/F)

- **Engagements:** Tracks each booking’s dates, times, contract price, and links to the agent, customer, and entertainer.

Column Name	Description
Engagementnumber (Primary key)	Unique ID for the engagement
startdate	Start date of the engagement (yyyy-mm-dd)
enddate	End date of the engagement (yyyy-mm-dd)
starttime	Start time of performance (hh:mm:ss)
stoptime	End time of performance (hh:mm:ss)
contractprice	Price for the engagement
customerid	Linked customer ID
agentid	Linked agent ID
entertainerid	Linked entertainer ID

- **entertainer_members:** Defines which members belong to each entertainer's troupe. Composite Primary key: (entertainerid, memberid)

Column Name	Description
entertainerid	Linked entertainer ID
memberid	Linked member ID
status	Member's status in group (1/2)

- **entertainer_styles:** Records the musical styles each entertainer performs. Composite Primary key: (entertainerid, styleid)

Column Name	Description
entertainerid	Linked entertainer ID
styleid	Linked musical style ID
stylestrength	Strength ranking of the style (1-3)

- **Musical_style:** Reference list of all possible musical styles by name.

Column Name	Description
styleid (Primary key)	Unique ID for the musical style
stylename	Name of the musical style

- **Musical_preferences:** Captures customers' style preferences for tailoring entertainment suggestions. Composite Primary key: (customerid, styleid)

Column Name	Description
customerid	Customer ID
styleid	Style ID preferred
preferenceseq	Preference sequence

- **Ztbldays**

Column Name	Description
DateField (Primary key)	Every calendar date

- **Ztblmonths**

Composite Primary key: (yearnumber, monthnumber)

Column Name	Description
monthyear	Full month name and four-digit yea
yearnumber	Four-digit year, displayed with a comma as thousands separator (e.g., 2,017)
monthnumber	Numeric month (1 = January … 12 = December)
monthstart	Date of the first day of that month (yyyy-mm-dd)
monthend	Date of the last day of that month (yyyy-mm-dd)
january	Flag: 1 if monthNumber = 1; otherwise 0
february	Flag: 1 if monthNumber = 2; otherwise 0
march	Flag: 1 if monthNumber = 3; otherwise 0
april	Flag: 1 if monthNumber =4; otherwise 0
may	Flag: 1 if monthNumber = 5; otherwise 0
june	Flag: 1 if monthNumber = 6; otherwise 0
july	Flag: 1 if monthNumber = 7; otherwise 0
august	Flag: 1 if monthNumber = 8; otherwise 0
september	Flag: 1 if monthNumber = 9; otherwise 0
october	Flag: 1 if monthNumber = 10; otherwise 0
november	Flag: 1 if monthNumber = 11; otherwise 0
december	Flag: 1 if monthNumber = 12; otherwise 0

- **ztblskiplabels**

Column Name	Description
labelcount (Primary key)	sequential values from 1 to 60

- **ztblweeks**

Column Name	Description
weekstart (Primary key)	First day of the week (yyyy-mm-dd)
weekend	Last day of the week (yyyy-mm-dd)

2. Data Quality Notes

A. Data Timeframe:

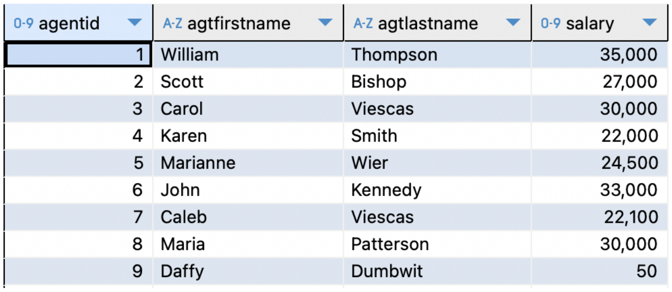
- 2017/9 ~ 2018/3

B. Missing Values

- Entertainer_members – missing value for one gender
- Entertainers – missing values in multiple entwebpage and entmailaddress
- Engagements – missing some engagement records, there are 131 engagements but only shows 111 rows

C. Outliers

- One outlier in an agent's salary.



0-9 agentid	A-Z agtfirstname	A-Z agtlastname	0-9 salary
1	William	Thompson	35,000
2	Scott	Bishop	27,000
3	Carol	Viescas	30,000
4	Karen	Smith	22,000
5	Marianne	Wier	24,500
6	John	Kennedy	33,000
7	Caleb	Viescas	22,100
8	Maria	Patterson	30,000
9	Daffy	Dumbwit	50

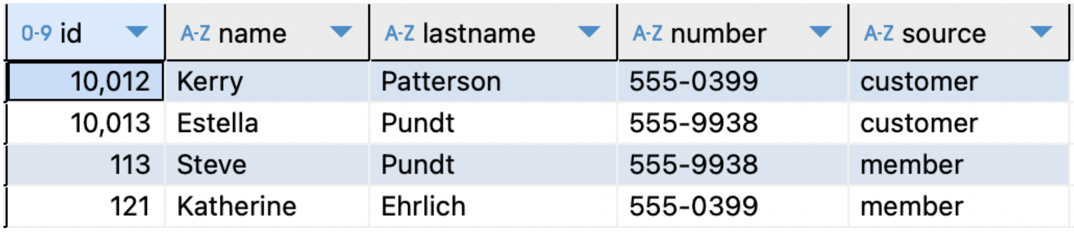
Solution: Update the salary and commission rate for Daffy

```
WITH stats AS (
 SELECT
   -- Salary quartiles & median
   PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY Salary)
AS q1_salary,
   PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY Salary)
AS med_salary,
   PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY Salary)
AS q3_salary,
   -- CommissionRate quartiles & median
   PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY
CommissionRate) AS q1_comm,
   PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY
CommissionRate) AS med_comm,
   PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY
CommissionRate) AS q3_comm
 FROM Agents
)
UPDATE Agents
SET
 Salary = stats.med_salary,
 CommissionRate = stats.med_comm
FROM stats
WHERE
   Salary  < stats.q1_salary - 1.5 * (stats.q3_salary - stats.q1_salary)
 OR Salary  > stats.q3_salary + 1.5 * (stats.q3_salary - stats.q1_salary)
 OR CommissionRate < stats.q1_comm   - 1.5 * (stats.q3_comm   -
stats.q1_comm)
```

```
 OR CommissionRate > stats.q3_comm  + 1.5 * (stats.q3_comm  -
stats.q1_comm)
RETURNING Agents.AgentID;
```

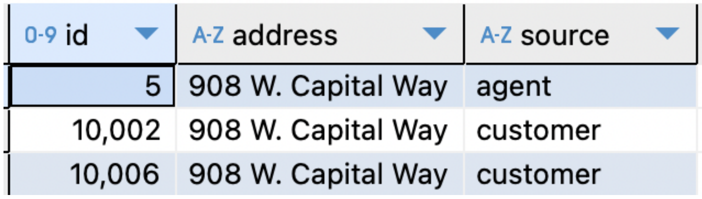
D. Ambiguous Data (Need further explanation from client)

- Entertainer_members – What does the numeric codes (e.g. 1, 2) represent in status column
- entertainer_styles – What's the scale/range (e.g. 1–3, 1–5)? What does "strength" measure?
- Musical_preferences – What is preferencessq (1,2,3)?
- Ztblmonths, Ztbldays, Ztblweeks, ztblskiplabels – What are the meanings or purposes of these tables?
- Ztblmonths – Since the MonthNumber column already specifies the month, what's the rationale for including twelve separate binary columns (January through December)—most of which are zero—instead of relying solely on MonthNumber?
- Entertainer_members & Customers –Two phone numbers are shared between a customer and a member:555-0399 and 555-9938



0-9 id	A-Z name	A-Z lastname	A-Z number	A-Z source
10,012	Kerry	Patterson	555-0399	customer
10,013	Estella	Pundt	555-9938	customer
113	Steve	Pundt	555-9938	member
121	Katherine	Ehrlich	555-0399	member

- Agents & Customers – There are two customer and one agent who have the same address



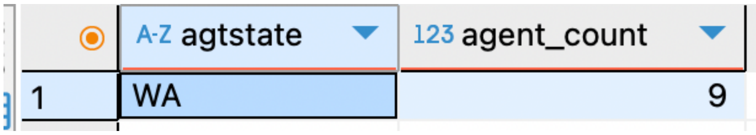
0-9 id	A-Z address	A-Z source
5	908 W. Capital Way	agent
10,002	908 W. Capital Way	customer
10,006	908 W. Capital Way	customer

3. Descriptive Analysis & Observations

A. Customer and Agents Analysis – Entertainment Dataset

- Agent Coverage by Geography

```
SELECT AgtState, COUNT(*) AS agent_count
FROM Agents
GROUP BY AgtState
ORDER BY agent_count DESC;
```



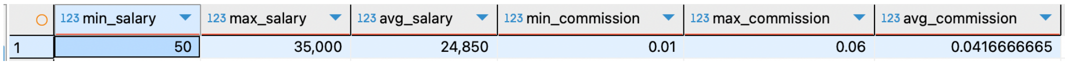
	A-Z agtstate	123 agent_count
1	WA	9

Key Observations:

- All agents are based in Washington (WA), indicating a geographically limited operational presence.
- Operational Risk: A disruption in WA could paralyze business activities.
- Growth Bottleneck: No agent coverage in other states may result in missed revenue, lost customers, and poor service reach.
- Scalability Insight: The agency has potential for growth via geographic expansion.

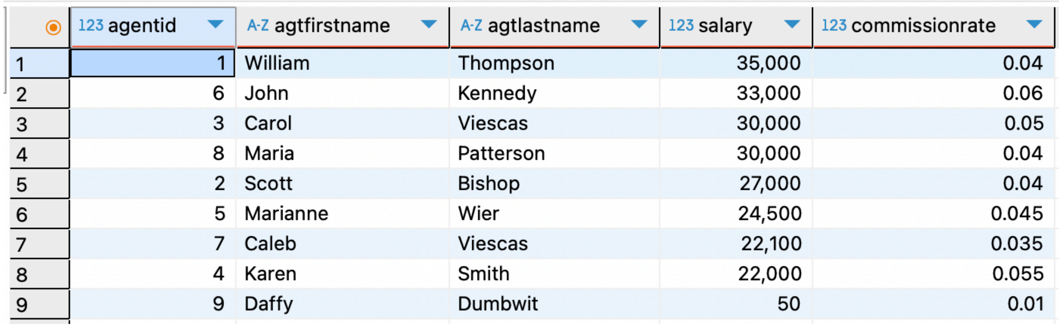
- **Agent Salary and Commission Structure**

```
SELECT
MIN(Salary) AS min_salary,
MAX(Salary) AS max_salary,
AVG(Salary) AS avg_salary,
MIN(CommissionRate) AS min_commission,
MAX(CommissionRate) AS max_commission,
AVG(CommissionRate) AS avg_commission
FROM Agents;
```



○	123 min_salary	123 max_salary	123 avg_salary	123 min_commission	123 max_commission	123 avg_commission
1	50	35,000	24,850	0.01	0.06	0.0416666665

```
SELECT AgentID, AgtFirstName, AgtLastName, Salary, CommissionRate
FROM Agents
ORDER BY Salary DESC;
```



○	123 agentid	A-Z agtfirstname	A-Z agtlastname	123 salary	123 commissionrate
1	1	William	Thompson	35,000	0.04
2	6	John	Kennedy	33,000	0.06
3	3	Carol	Viescas	30,000	0.05
4	8	Maria	Patterson	30,000	0.04
5	2	Scott	Bishop	27,000	0.04
6	5	Marianne	Wier	24,500	0.045
7	7	Caleb	Viescas	22,100	0.035
8	4	Karen	Smith	22,000	0.055
9	9	Daffy	Dumbwit	50	0.01

Key Observations:

- Salaries range from $22,000 to $35,000 with one outlier ($50), indicating a potential data error.
- Commission rates are low and flat, between 3.5% to 6%.

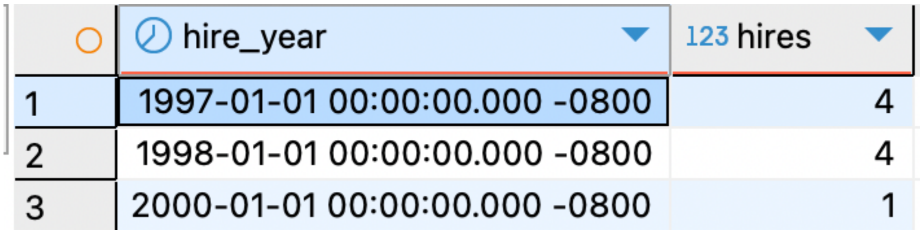
Business Interpretation:

- Flat Incentive Structure: May demotivate performance-focused agents.

    - Data Outlier: One agent with a $50 salary should be reviewed for accuracy.
    - No performance-tiered compensation: Limits motivation and growth potential.

- **Agent Hiring Trend Over Time**

```
SELECT DATE_TRUNC('year', DateHired) AS hire_year, COUNT(*) AS hires
FROM Agents
WHERE DateHired IS NOT NULL
GROUP BY hire_year
ORDER BY hire_year;
```



	hire_year	hires
1	1997-01-01 00:00:00.000 -0800	4
2	1998-01-01 00:00:00.000 -0800	4
3	2000-01-01 00:00:00.000 -0800	1

Key Observations:

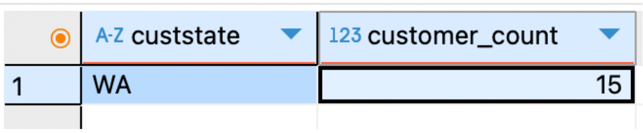
- Most hires occurred between 1997–2000.

Business Interpretation:

- No recent hires suggest either stagnation or missing data.
- Long-term lack of hiring may hurt adaptability and market expansion.

- **Customer Distribution by State**

```
SELECT CustState, COUNT(*) AS customer_count
FROM Customers
GROUP BY CustState
ORDER BY customer_count DESC;
```



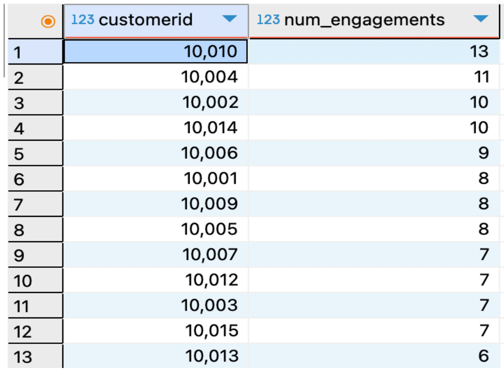
	custstate	customer_count
1	WA	15

Business Interpretation:

- All customers are also located in WA.
- Strategic Implication: Perfect alignment between agent and customer geography, but the agency has no reach outside WA.
- Recommend expansion into neighboring states (e.g., OR, CA) to capture new markets.

- **New vs. Repeat Customers**

```
    SELECT e.CustomerID, COUNT(*) AS num_engagements
    FROM Engagements e
    GROUP BY e.CustomerID
ORDER BY num_engagements DESC;
```



	customerid	num_engagements
1	10,010	13
2	10,004	11
3	10,002	10
4	10,014	10
5	10,006	9
6	10,001	8
7	10,009	8
8	10,005	8
9	10,007	7
10	10,012	7
11	10,003	7
12	10,015	7
13	10,013	6

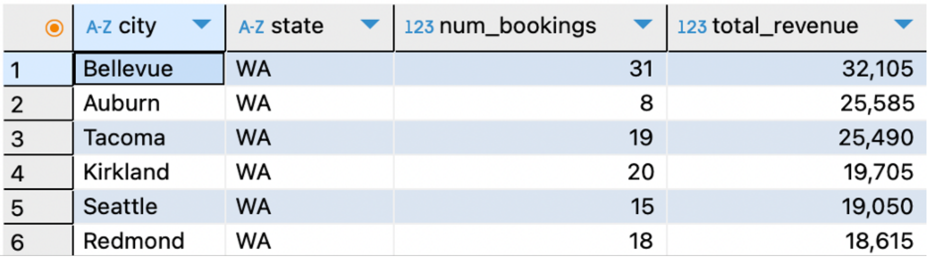
Interpretation:

- All customers have 6 or more bookings.
- **100% Repeat Clients**: Indicates high customer loyalty and satisfaction.
- Opportunity for launching loyalty programs, premium services, or referral incentives

B. **Geographic Analysis (Washington State)**

- **Top Revenue-Generating Cities**

```
SELECT c.CustCity  AS city, c.CustState AS state, COUNT(*) AS num_bookings,
SUM(e.ContractPrice) AS total_revenue FROM Engagements e
JOIN Customers c ON e.CustomerID = c.CustomerID
GROUP by  c.CustCity, c.CustState
ORDER by total_revenue DESC;
```



	city	state	num_bookings	total_revenue
1	Bellevue	WA	31	32,105
2	Auburn	WA	8	25,585
3	Tacoma	WA	19	25,490
4	Kirkland	WA	20	19,705
5	Seattle	WA	15	19,050
6	Redmond	WA	18	18,615

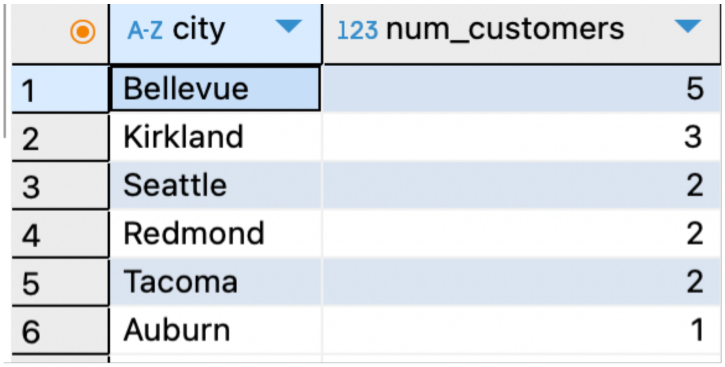
Key Observations:

- Bellevue is the clear leader in both engagement volume and revenue.
- Auburn stands out with high revenue despite low booking count, suggesting higher-value events.

    - This dual view helps identify high-potential cities for focused campaigns or premium service offerings.

- **Customer Concentration by City**

```
SELECT CustCity AS city, COUNT(*) AS num_customers
FROM Customers
GROUP BY CustCity
ORDER BY num_customers DESC;
```



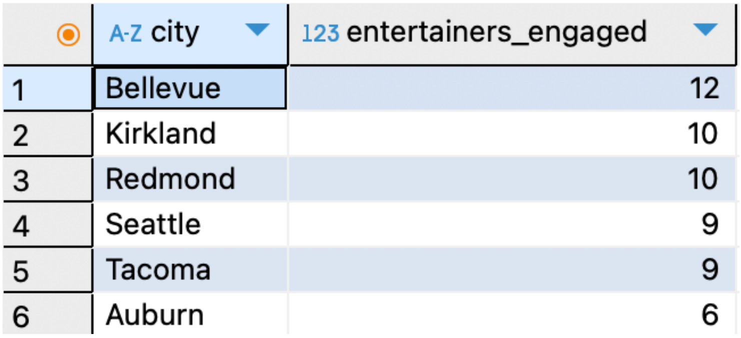
	city	num_customers
1	Bellevue	5
2	Kirkland	3
3	Seattle	2
4	Redmond	2
5	Tacoma	2
6	Auburn	1

Key Observations:

- Bellevue has the highest number of customers (5), followed by Kirkland (3).
- This doesn’t always align with revenue, suggesting some cities (like Auburn) may have high-value customers despite low count.
- These patterns can guide customer retention, local outreach, or expansion planning.

- **Entertainer Reach by City**

```
SELECT c.CustCity AS city, COUNT(DISTINCT e.EntertainerID) AS
entertainers_engaged
FROM Engagements e
JOIN Customers c ON e.CustomerID = c.CustomerID
GROUP BY c.CustCity
ORDER BY entertainers_engaged DESC;
```



	city	entertainers_engaged
1	Bellevue	12
2	Kirkland	10
3	Redmond	10
4	Seattle	9
5	Tacoma	9
6	Auburn	6

Key Observations:

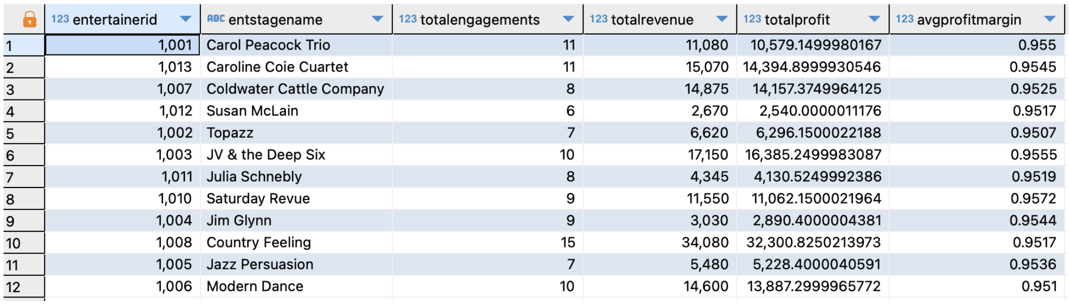
- Bellevue leads again, engaging 12 unique entertainers, suggesting a broad demand variety.
- High entertainer counts in Kirkland and Redmond imply diverse entertainment preferences in those markets.
- Could inform regional event strategy or style-specific targeting.

4. Advanced Analytical Findings

A. Revenue

- What is the profit margin for entertainers?

```
SELECT
  en.EntertainerID,
  en.EntStageName,
  COUNT(e.EngagementNumber) AS TotalEngagements,
  SUM(e.ContractPrice) AS TotalRevenue,
  SUM(e.ContractPrice * (1 - a.CommissionRate)) AS TotalProfit,
  ROUND(AVG(1 - a.CommissionRate)::NUMERIC, 4) AS AvgProfitMargin
FROM
  Engagements e
JOIN
  Entertainers en ON e.EntertainerID = en.EntertainerID
JOIN
  Agents a ON e.AgentID = a.AgentID
GROUP BY
  en.EntertainerID, en.EntStageName;
```

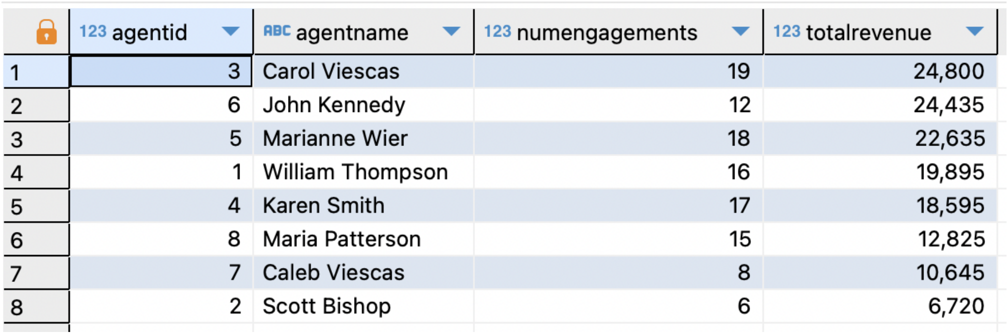


	entertainerid	entstagename	totalengagements	totalrevenue	totalprofit	avgprofitmargin
1	1,001	Carol Peacock Trio	11	11,080	10,579.1499980167	0.955
2	1,013	Caroline Coie Cuartet	11	15,070	14,394.8999930546	0.9545
3	1,007	Coldwater Cattle Company	8	14,875	14,157.3749964125	0.9525
4	1,012	Susan McLain	6	2,670	2,540.0000011176	0.9517
5	1,002	Topazz	7	6,620	6,296.1500022188	0.9507
6	1,003	JV & the Deep Six	10	17,150	16,385.2499983087	0.9555
7	1,011	Julia Schnebly	8	4,345	4,130.5249992386	0.9519
8	1,010	Saturday Revue	9	11,550	11,062.1500021964	0.9572
9	1,004	Jim Glynn	9	3,030	2,890.4000004381	0.9544
10	1,008	Country Feeling	15	34,080	32,300.8250213973	0.9517
11	1,005	Jazz Persuasion	7	5,480	5,228.4000040591	0.9536
12	1,006	Modern Dance	10	14,600	13,887.2999965772	0.951

Findings: Profit margins for entertainers range from 0.9507 to 0.9572, indicating a relatively narrow band of high profitability (over 95% for all).

- Which agents are responsible for the highest total contract revenue?

```
SELECT
  a.AgentID,
  a.AgtFirstName || ' ' || a.AgtLastName AS AgentName,
  COUNT(e.EngagementNumber) AS NumEngagements,
  SUM(e.ContractPrice) AS TotalRevenue
FROM
  Agents a
JOIN
  Engagements e ON a.AgentID = e.AgentID
GROUP BY
  a.AgentID, a.AgtFirstName, a.AgtLastName
ORDER BY
  TotalRevenue DESC;
```



	agentid	agentname	numengagements	totalrevenue
1	3	Carol Viescas	19	24,800
2	6	John Kennedy	12	24,435
3	5	Marianne Wier	18	22,635
4	1	William Thompson	16	19,895
5	4	Karen Smith	17	18,595
6	8	Maria Patterson	15	12,825
7	7	Caleb Viescas	8	10,645
8	2	Scott Bishop	6	6,720

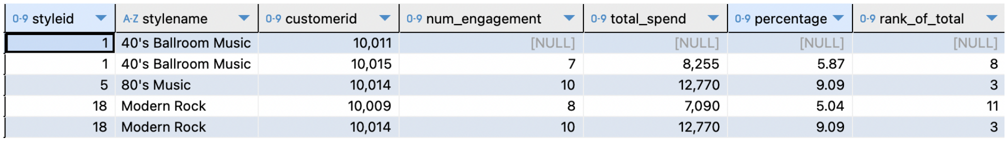
Findings: Carol Viescas generated the highest total contract revenue at $24,800 across 19 engagements; John Kennedy generates ~$2,036 per engagement, higher than Carol Viescas' — indicating better deal quality per client.

B. Content & Customer Fit

- Are there styles customers prefer but aren't well-represented in the entertainer pool?

```
with missing as
(select distinct mp.styleid ,ms.stylename, mp.customerid, es.styleid as entertainment_style
from musical_preferences mp
left join entertainer_styles es
on mp.styleid = es.styleid
join musical_styles ms
on ms.styleid = mp.styleid
where es.styleid is null
order by styleid),
ranking as
(select customerid , count(*) as num_engagement, sum(contractprice) as total_spend,
        round(100*sum(contractprice)/(select sum(contractprice)from engagements e),2)
as percentage_of_total_spend,
        rank()over(order by sum(contractprice)desc) as rank_of_total from engagements
e
        group by customerid)
select m.styleid ,m.stylename, m.customerid, r.num_engagement, r.total_spend,
r.percentage_of_total_spend, r.rank_of_total from ranking r
```

```
right join missing m
on r.customerid = m.customerid
```



styleid	stylename	customerid	num_engagement	total_spend	percentage	rank_of_total
1	40's Ballroom Music	10,011	[NULL]	[NULL]	[NULL]	[NULL]
1	40's Ballroom Music	10,015	7	8,255	5.87	8
5	80's Music	10,014	10	12,770	9.09	3
18	Modern Rock	10,009	8	7,090	5.04	11
18	Modern Rock	10,014	10	12,770	9.09	3

Findings:

There is unmet demand in the market. Several customers — including some of TuneWorks' top spenders — have musical preferences that the current entertainer lineup does not fulfill.

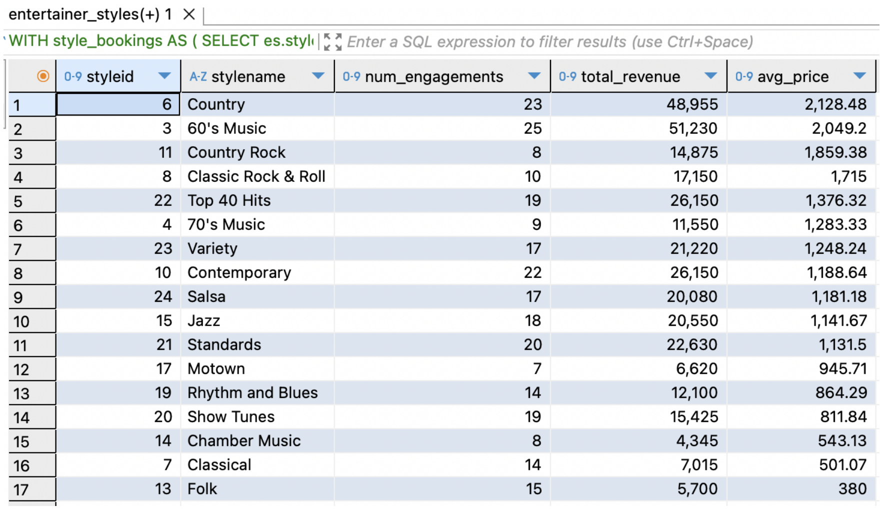
- **Are some musical styles more popular or more lucrative than others**

```
WITH style_bookings AS (

SELECT

  es.styleid,

  ms.stylename,
  COUNT(e.engagementnumber) AS num_engagements,
  SUM(e.contractprice) AS total_revenue,
  ROUND(AVG(e.contractprice)::numeric, 2) AS avg_price
 FROM entertainer_styles es
 JOIN musical_styles ms ON es.styleid = ms.styleid
 JOIN engagements e ON es.entertainerid = e.entertainerid
 GROUP BY es.styleid, ms.stylename
)
SELECT *
FROM style_bookings
ORDER BY avg_price DESC;
```



entertainer_styles(+) 1 ×

WITH style_bookings AS (SELECT es.styl

Enter a SQL expression to filter results (use Ctrl+Space)

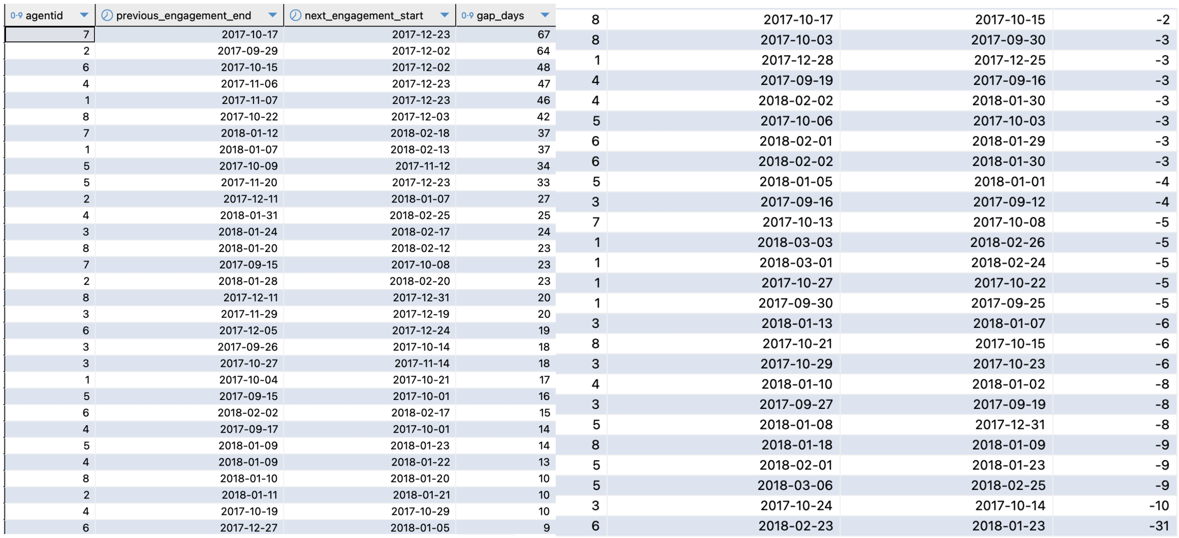
	styleid	stylename	num_engagements	total_revenue	avg_price
1	6	Country	23	48,955	2,128.48
2	3	60's Music	25	51,230	2,049.2
3	11	Country Rock	8	14,875	1,859.38
4	8	Classic Rock & Roll	10	17,150	1,715
5	22	Top 40 Hits	19	26,150	1,376.32
6	4	70's Music	9	11,550	1,283.33
7	23	Variety	17	21,220	1,248.24
8	10	Contemporary	22	26,150	1,188.64
9	24	Salsa	17	20,080	1,181.18
10	15	Jazz	18	20,550	1,141.67
11	21	Standards	20	22,630	1,131.5
12	17	Motown	7	6,620	945.71
13	19	Rhythm and Blues	14	12,100	864.29
14	20	Show Tunes	19	15,425	811.84
15	14	Chamber Music	8	4,345	543.13
16	7	Classical	14	7,015	501.07
17	13	Folk	15	5,700	380

Findings: Styles like Country and 60’s Music not only attract a high number of engagements but also command the highest average contract prices (>$2,000), making them both popular and lucrative. In contrast, styles like Folk and Classical appear frequently but yield the lowest average revenue, indicating lower pricing power or niche appeal.

C. Operation Efficiency

- Are there any agents has overlap on engagement schedules?

```
WITH engagement_gaps AS (
SELECT AgentID, StartDate, EndDate,
LAG(EndDate) OVER (PARTITION BY AgentID ORDER BY StartDate) AS prev_end_date
FROM Engagements
)
SELECT AgentID,
prev_end_date AS previous_engagement_end,
StartDate AS next_engagement_start,
(StartDate - prev_end_date) AS gap_days
FROM engagement_gaps
WHERE prev_end_date IS NOT NULL
ORDER BY gap_days desc
```



agentid	previous_engagement_end	next_engagement_start	gap_days
7	2017-10-17	2017-12-23	67
2	2017-09-29	2017-12-02	64
6	2017-10-15	2017-12-02	48
4	2017-11-06	2017-12-23	47
1	2017-11-07	2017-12-23	46
8	2017-10-22	2017-12-03	42
7	2018-01-12	2018-02-18	37
1	2018-01-07	2018-02-13	37
5	2017-10-09	2017-11-12	34
5	2017-11-20	2017-12-23	33
2	2017-12-11	2018-01-07	27
4	2018-01-31	2018-02-25	25
3	2018-01-24	2018-02-17	24
8	2018-01-20	2018-02-12	23
7	2017-09-15	2017-10-08	23
2	2018-01-28	2018-02-20	23
8	2017-12-11	2017-12-31	20
3	2017-11-29	2017-12-19	20
6	2017-12-05	2017-12-24	19
3	2017-09-26	2017-10-14	18
3	2017-10-27	2017-11-14	18
1	2017-10-04	2017-10-21	17
5	2017-09-15	2017-10-01	16
6	2018-02-02	2018-02-17	15
4	2017-09-17	2017-10-01	14
5	2018-01-09	2018-01-23	14
4	2018-01-09	2018-01-22	13
8	2018-01-10	2018-01-20	10
2	2018-01-11	2018-01-21	10
4	2017-10-19	2017-10-29	10
6	2017-12-27	2018-01-05	9
8	2017-10-17	2017-10-15	-2
8	2017-10-03	2017-09-30	-3
1	2017-12-28	2017-12-25	-3
4	2017-09-19	2017-09-16	-3
4	2018-02-02	2018-01-30	-3
5	2017-10-06	2017-10-03	-3
6	2018-02-01	2018-01-29	-3
6	2018-02-02	2018-01-30	-3
5	2018-01-05	2018-01-01	-4
3	2017-09-16	2017-09-12	-4
7	2017-10-13	2017-10-08	-5
1	2018-03-03	2018-02-26	-5
1	2018-03-01	2018-02-24	-5
1	2017-10-27	2017-10-22	-5
1	2017-09-30	2017-09-25	-5
3	2018-01-13	2018-01-07	-6
8	2017-10-21	2017-10-15	-6
3	2017-10-29	2017-10-23	-6
4	2018-01-10	2018-01-02	-8
3	2017-09-27	2017-09-19	-8
5	2018-01-08	2017-12-31	-8
8	2018-01-18	2018-01-09	-9
5	2018-02-01	2018-01-23	-9
5	2018-03-06	2018-02-25	-9
3	2017-10-24	2017-10-14	-10
6	2018-02-23	2018-01-23	-31

Findings:

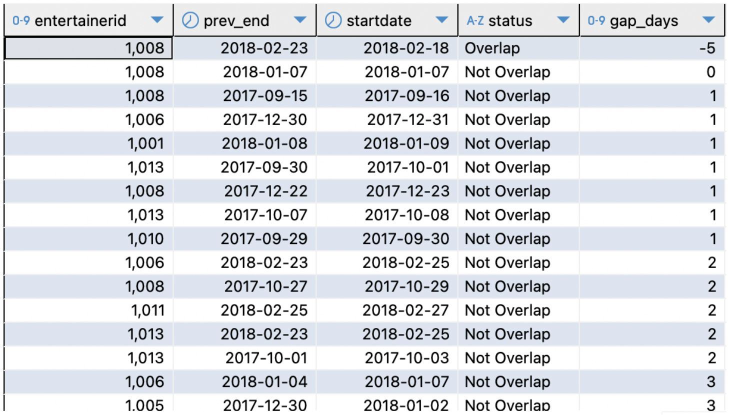
The analysis of agent scheduling uncovered two key issues:

- Gap Days measure: the interval between the start of each engagement and the end of the previous one.

- Workload imbalance: Agent 3 is over-booked with 7 overlapping engagements, creating service risk; Agent 2 has no overlaps but sits idle an average of 26 days, indicating wasted capacity.

- Are there any entertainers has overlap on engagement schedule ?

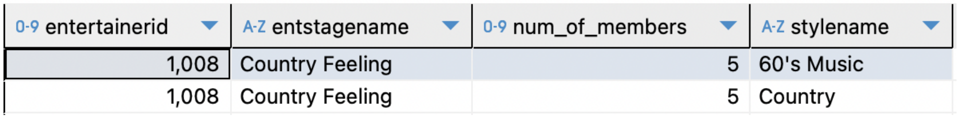
```
WITH booking_gaps AS (
SELECT EntertainerID, StartDate, EndDate,
LAG(EndDate) OVER (PARTITION BY EntertainerID ORDER BY StartDate
) AS prev_end
 FROM Engagements
)
SELECT
EntertainerID, prev_end, StartDate,
CASE
  WHEN StartDate < prev_end THEN 'Overlap'
ELSE 'Not Overlap'
         END AS status,
(StartDate - prev_end) AS gap_days
FROM booking_gaps
WHERE prev_end IS NOT NULL
ORDER BY gap_days
```



0-9 entertainerid	prev_end	startdate	A-Z status	0-9 gap_days
1,008	2018-02-23	2018-02-18	Overlap	-5
1,008	2018-01-07	2018-01-07	Not Overlap	0
1,008	2017-09-15	2017-09-16	Not Overlap	1
1,006	2017-12-30	2017-12-31	Not Overlap	1
1,001	2018-01-08	2018-01-09	Not Overlap	1
1,013	2017-09-30	2017-10-01	Not Overlap	1
1,008	2017-12-22	2017-12-23	Not Overlap	1
1,013	2017-10-07	2017-10-08	Not Overlap	1
1,010	2017-09-29	2017-09-30	Not Overlap	1
1,006	2018-02-23	2018-02-25	Not Overlap	2
1,008	2017-10-27	2017-10-29	Not Overlap	2
1,011	2018-02-25	2018-02-27	Not Overlap	2
1,013	2018-02-23	2018-02-25	Not Overlap	2
1,013	2017-10-01	2017-10-03	Not Overlap	2
1,006	2018-01-04	2018-01-07	Not Overlap	3
1.005	2017-12-30	2018-01-02	Not Overlap	3

```
select e.entertainerid, e.entstagename, count(em.memberid) as num_of_members,
ms.stylename from entertainers e
join entertainer_members em
on e.entertainerid = em.entertainerid
join entertainer_styles es
on es.entertainerid = em.entertainerid
join musical_styles ms
```

```
on ms.styleid = es.styleid
where e.entertainerid = 1008
group by e.entertainerid, e.entstagename,ms.stylename
```



entertainerid	entstagename	num_of_members	stylename
1,008	Country Feeling	5	60's Music
1,008	Country Feeling	5	Country

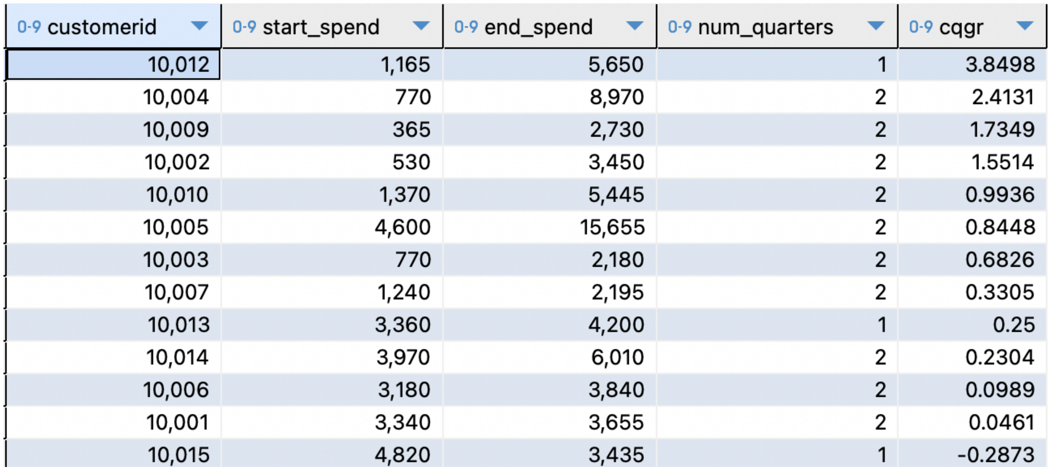
Findings:

- Critical Booking Conflict: Only one entertainer (ID 1008) experienced a 5-day overlap, an isolated but high-impact scheduling error that risks missed performances and brand reputation.
- Minimal Rest Windows: The vast majority of engagements have 0–2 days' turnaround, maximizing utilization but leaving little time for travel, setup, or recovery.

D. Customer Loyalty and Retention

- Which customers are increasing their spending the fastest over time?

```
WITH base AS (SELECTcustomerid,
  EXTRACT(YEAR FROM startdate) AS year,
  EXTRACT(QUARTER FROM startdate) AS quarter,
  COUNT(*) AS num_engagements,
  SUM(contractprice) AS total_spend
 FROM engagements
 GROUP BY customerid, EXTRACT(YEAR FROM startdate), EXTRACT(QUARTER
FROM startdate)
),
ranked AS (
 SELECT
  customerid,
  year,
  quarter,
  (year * 4 + quarter) AS quarter_index,
  total_spend,
  ROW_NUMBER() OVER (PARTITION BY customerid ORDER BY year, quarter
ASC) AS rn_start,
  ROW_NUMBER() OVER (PARTITION BY customerid ORDER BY year DESC,
quarter DESC) AS rn_end
 FROM base
),
start_values AS (
 SELECT customerid, total_spend AS start_spend, quarter_index AS start_q
 FROM ranked
 WHERE rn_start = 1
),
end_values AS (
 SELECT customerid, total_spend AS end_spend, quarter_index AS end_q
 FROM ranked
 WHERE rn_end = 1
)
SELECT
 s.customerid,
 s.start_spend,
 e.end_spend,
 (e.end_q - s.start_q) AS num_quarters,
 ROUND(
   (POWER(e.end_spend / NULLIF(s.start_spend, 0), 1.0 / NULLIF(e.end_q - s.start_q,
0)) - 1)::numeric,
   4
 ) AS cqgr
FROM start_values s
JOIN end_values e ON s.customerid = e.customerid
WHERE s.start_spend > 0 AND e.end_spend > 0
ORDER BY cqgr DESC
```



0-9 customerid	0-9 start_spend	0-9 end_spend	0-9 num_quarters	0-9 cqgr
10,012	1,165	5,650	1	3.8498
10,004	770	8,970	2	2.4131
10,009	365	2,730	2	1.7349
10,002	530	3,450	2	1.5514
10,010	1,370	5,445	2	0.9936
10,005	4,600	15,655	2	0.8448
10,003	770	2,180	2	0.6826
10,007	1,240	2,195	2	0.3305
10,013	3,360	4,200	1	0.25
10,014	3,970	6,010	2	0.2304
10,006	3,180	3,840	2	0.0989
10,001	3,340	3,655	2	0.0461
10,015	4,820	3,435	1	-0.2873

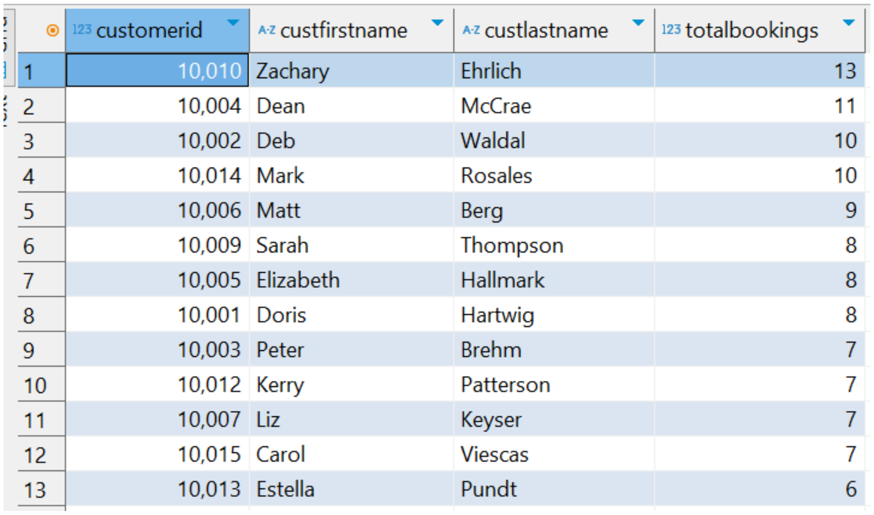
Findings

- Customer 10,012: Highest CQGR at 3.8498, spending jumped from $1,165 → $5,650 in just 1 quarter
- Customer 10,004 and 10,009: Strong 2-quarter growth at 2.41x and 1.73x respectively.
- Customer 10,015: Negative CQGR (−0.2873), spending dropped from $4,820 → $3,435 in 1 quarter

E. Customer Lifetime Value

- Here we find the important customers (High CLV) and what their preferences are.

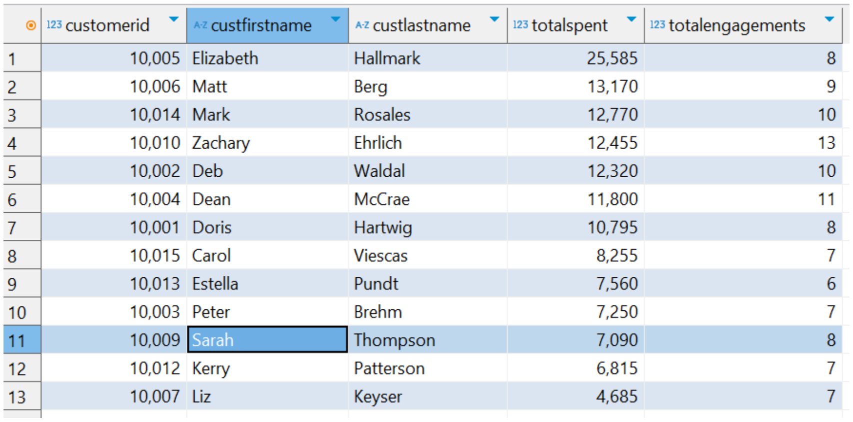
```
SELECT
  e.CustomerID,
  c.CustFirstName,
  c.CustLastName,
  COUNT(e.EngagementNumber) AS TotalBookings
FROM
  Engagements e
JOIN
  Customers c ON e.CustomerID = c.CustomerID
GROUP BY
  e.CustomerID, c.CustFirstName, c.CustLastName
HAVING
  COUNT(e.EngagementNumber) > 1
order by COUNT(e.EngagementNumber) desc
```



	customerid	custfirstname	custlastname	totalbookings
1	10,010	Zachary	Ehrlich	13
2	10,004	Dean	McCrae	11
3	10,002	Deb	Waldal	10
4	10,014	Mark	Rosales	10
5	10,006	Matt	Berg	9
6	10,009	Sarah	Thompson	8
7	10,005	Elizabeth	Hallmark	8
8	10,001	Doris	Hartwig	8
9	10,003	Peter	Brehm	7
10	10,012	Kerry	Patterson	7
11	10,007	Liz	Keyser	7
12	10,015	Carol	Viescas	7
13	10,013	Estella	Pundt	6

```
-- Top money spent
SELECT
  c.CustomerID,
  c.CustFirstName,
  c.CustLastName,
  SUM(e.ContractPrice) AS TotalSpent,
  COUNT(e.EngagementNumber) AS TotalEngagements
FROM
  Customers c
JOIN
  Engagements e ON c.CustomerID = e.CustomerID
GROUP BY
  c.CustomerID, c.CustFirstName, c.CustLastName
```

```
ORDER BY
  TotalSpent DESC;
```

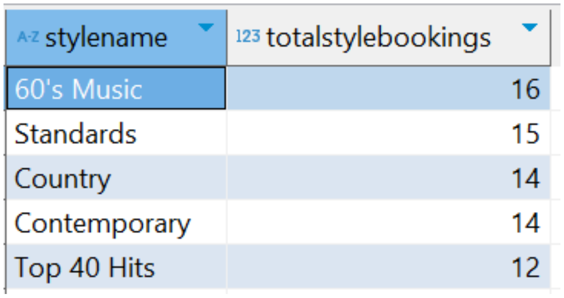


	customerid	custfirstname	custlastname	totalspent	totalengagements
1	10,005	Elizabeth	Hallmark	25,585	8
2	10,006	Matt	Berg	13,170	9
3	10,014	Mark	Rosales	12,770	10
4	10,010	Zachary	Ehrlich	12,455	13
5	10,002	Deb	Waldal	12,320	10
6	10,004	Dean	McCrae	11,800	11
7	10,001	Doris	Hartwig	10,795	8
8	10,015	Carol	Viescas	8,255	7
9	10,013	Estella	Pundt	7,560	6
10	10,003	Peter	Brehm	7,250	7
11	10,009	Sarah	Thompson	7,090	8
12	10,012	Kerry	Patterson	6,815	7
13	10,007	Liz	Keyser	4,685	7

Findings:

The top 6 customers who are both high-engagement and high-spenders identified have high customer lifetime value, and thus we should be prioritizing them by offering them premium discounts, loyalty programs, early booking, etc. We can also identify the traits of these customers to define target demographics for future marketing.

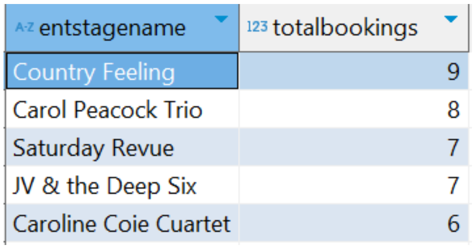
```
-- Identify top 5 Music styles of Top 6 Customers
SELECT
  ms.StyleName,
  COUNT(*) AS TotalStyleBookings
FROM
  Engagements e
JOIN
  Entertainer_Styles es ON e.EntertainerID = es.EntertainerID
JOIN
  Musical_Styles ms ON es.StyleID = ms.StyleID
WHERE
  e.CustomerID IN (10005, 10006, 10014, 10010, 10002, 10004)
GROUP BY
  ms.StyleName
ORDER BY
  TotalStyleBookings DESC
limit 5
```



stylename	totalstylebookings
60's Music	16
Standards	15
Country	14
Contemporary	14
Top 40 Hits	12

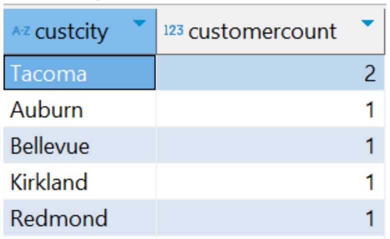
```
-- Identify top 5 Entertainers of Top 6 Customers
```

```
SELECT
  ent.EntStageName,
  COUNT(*) AS TotalBookings
FROM
  Engagements e
JOIN
  Entertainers ent ON e.EntertainerID = ent.EntertainerID
WHERE
  e.CustomerID IN (10005, 10006, 10014, 10010, 10002, 10004)
GROUP BY
  ent.EntStageName
ORDER BY
  TotalBookings desc
limit 5
```



entstagename	totalbookings
Country Feeling	9
Carol Peacock Trio	8
Saturday Revue	7
JV & the Deep Six	7
Caroline Coie Cuartet	6

```
--  Summary City for top 6 CTV
 SELECT
   CustCity,
  COUNT(*) AS CustomerCount
FROM
 Customers
WHERE
  CustomerID IN (10005, 10006, 10014, 10010, 10002, 10004)
GROUP BY
  CustCity
ORDER BY
  CustomerCount DESC;
```



custcity	customercount
Tacoma	2
Auburn	1
Bellevue	1
Kirkland	1
Redmond	1

```
-- Track Customer Style Preferences Over Time
SELECT
```

```
  e.CustomerID,
  e.EngagementNumber,
  e.StartDate,
  ms.StyleName,
  es.StyleStrength
FROM
  Engagements e
JOIN
  Entertainers ent ON e.EntertainerID = ent.EntertainerID
JOIN
  Entertainer_Styles es ON ent.EntertainerID = es.EntertainerID
JOIN
  Musical_Styles ms ON es.StyleID = ms.StyleID
ORDER BY
  e.CustomerID, e.StartDate;
```

Findings:

Preferences for most customers appear to change over time. However, there's too much data to summarize and no recognizable pattern, so will not be included in the final.

5. Conclusion and Future Strategy Recommendations

The analysis of TuneWorks' operations reveals a highly loyal customer base but an operationally inefficient business model. To sustain growth and mitigate risks of a rapidly shifting entertainment industry, the company should consider employing some of the following strategies.

Revamp Agent Compensation: Agent salaries and commission rates are flat, with no clear performance incentives. This is despite most hires occurring before 2000 and a high average tenure with the company. Introducing tiered commission structures, incentive-based bonuses, and annual hiring plans would boost agent motivation, increase the quality of service, and create a succession pipeline for long-term operations.

Optimize High-Value City Performance: TuneWorks is geographically limited to Washington State, and it would be smart to expand into nearby markets that share cultural and entertainment preferences. However, before geographic expansion, the company should launch premium packages and loyalty programs to exploit the substantial revenue in high-earning cities such as Bellevue, Auburn, and Kirkland. The goal would be to increase margins with improved customer retention and recurring revenue.

Focus on Customer Preferences: There are discrepancies between the music style preferences of high-spending customers and entertainer offerings. TuneWorks should work to meet customer demands. The company should expand its entertainment offerings and use customer preference data to inform talent acquisition.