

DSO 552: SQL Databases for Business Analysts

Homework 5 (Due 11:59pm PST November 16th, 2020)

Trojan Entertainment Agency Database

This database manages entertainers, agents, customers, and bookings. Check Figure 1 (on the last page) for more details on the database structure.

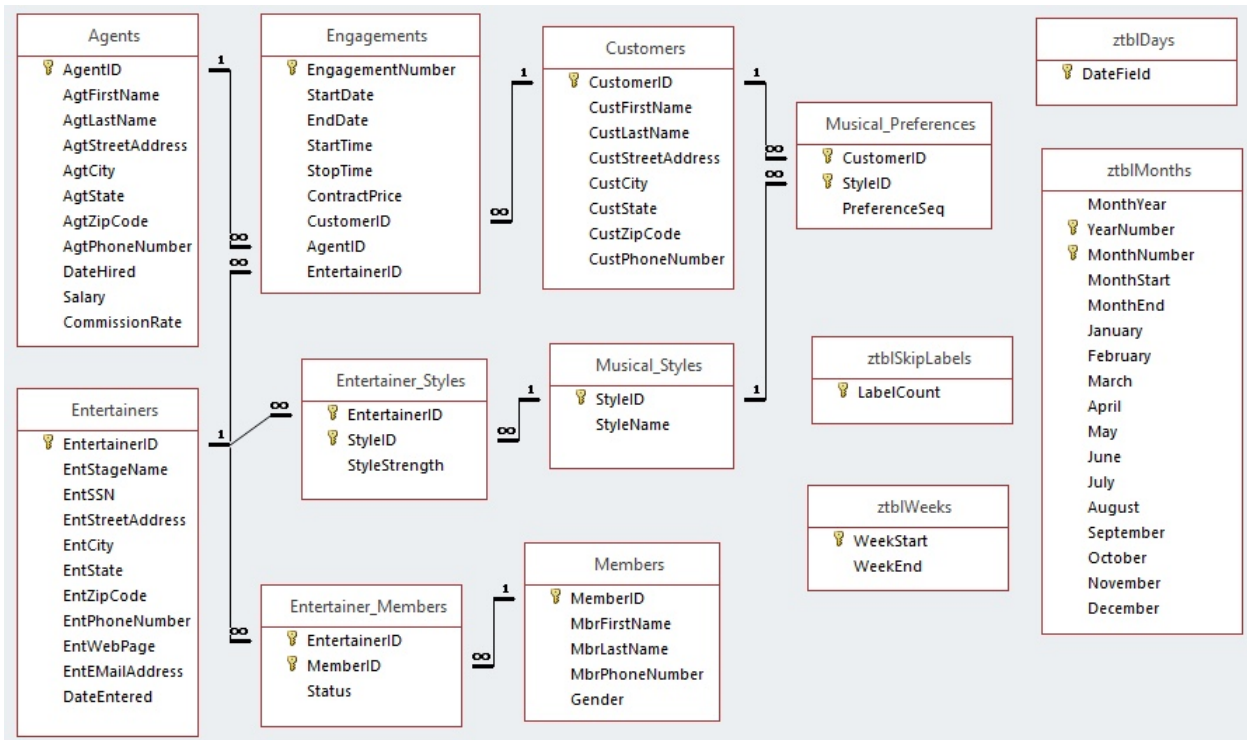


Figure 1: Trojan Entertainment Schema

1. (2 points) Provide a count of customers and the number of engagements they have placed where the contract price is in the top quartile of all contract prices.

Expected Result:

customerid	top_quartile_engagements
10001	3
10002	4
10003	1
10004	3
10005	3
10006	5
10009	1
10012	2
10013	2
10014	4
10015	3

2. (2 points) Whenever entertainers in our agency complete their 10th engagement, a blog post is written about the entertainers and the members. We also give their contractprice for the engagement a 10% bonus. Please list the details of the 10th engagement for each entertainer and the adjusted contracted price.

```
SELECT e.*, ee.entstagename, contractprice * 1.1 AS adjusted_contractprice
FROM engagements e
      JOIN entertainers ee ON ee.entertainerid = e.entertainerid
WHERE engagementnumber IN (
    SELECT engagementnumber
    FROM (
        SELECT *,
              ROW_NUMBER() OVER (
                PARTITION BY entertainerid ORDER BY startdate)
        FROM engagements) t1
    WHERE t1.row_number = 10)
```

Expected:

engagementnumber	startdate	enddate	starttime	stoptime	contractprice	customerid	agentid
87	2018-01-05	2018-01-07	16:00:00	19:00:00	275	10007	6
110	2018-02-12	2018-02-20	15:00:00	19:00:00	1670	10006	8
115	2018-02-20	2018-02-23	12:00:00	18:00:00	1490	10007	5
126	2018-02-25	2018-03-04	18:00:00	20:00:00	1010	10009	6
131	2018-03-04	2018-03-13	15:00:00	17:00:00	1850	10014	1

3. (2 points) Show the total revenue generated by our agency at the end of each month. Note that the revenue generated is not the contract price. It is 10% for all engagements typically. However, for entertainers who have at least 10 bookings with us, it is 8% of the contract price.

```
WITH top_entertainers AS (
    SELECT ee.entertainerid
    FROM entertainers ee
         NATURAL JOIN engagements e
    GROUP BY 1
    HAVING COUNT(e.engagementnumber) >= 10)

SELECT e.*,
       CASE
         WHEN entertainerid IN
```

```

        (SELECT * FROM top_entertainers) THEN contractprice * .08
        ELSE contractprice * .10 END
SUM(CASE
    WHEN entertainerid IN (SELECT * FROM top_entertainers)
    THEN contractprice * .08
    ELSE contractprice * .10 END) OVER
    (ORDER BY startdate, engagementnumber) AS running_total
FROM engagements e

```

4. (2 points) Produce a report that lists the top five agents and the top five musical styles in terms of number of engagements. Hint: You'll likely use a UNION in this problem. First find the query to get the top five agents and their counts, then a query to get the top five musical styles and their counts.

```

SELECT 'category' AS type, *
FROM (SELECT ms.stylename AS name,
    COUNT(DISTINCT e.engagementnumber) AS num_engagements
    FROM engagements e
        JOIN entertainer_styles es ON es.entertainerid = e.entertainerid
        JOIN musical_styles ms ON ms.styleid = es.styleid
    GROUP BY 1
    ORDER BY 2 DESC
    LIMIT 5) t1
UNION
SELECT 'agent' AS type, *
FROM (SELECT a.agtfirstname || ' ' || a.agtlastname AS agent_name,
    COUNT(DISTINCT e.engagementnumber) AS num_engagements
    FROM engagements e
        JOIN agents a ON a.agentid = e.agentid
    GROUP BY 1
    ORDER BY 2 DESC
    LIMIT 5) t2
ORDER BY 3 DESC

```

Expected Result:

type	name	num_engagements
category	60's Music	25
category	Country	23
category	Contemporary	22
category	Standards	20
agent	Carol Viescas	19
category	Show Tunes	19
agent	Marianne Wier	18
agent	Karen Smith	17
agent	William Thompson	16
agent	Maria Patterson	15

5. (2 points) We use the first two digits after the area code of a phone number to determine if the number is a landline or a mobile phone number. For example, if a phone number is 234-2191, then the type block to consider is 21. If the type block is greater than 25, it will be a landline phone number. If it is 25 or less, it is a mobile phone number. Classify all agents and customers phone numbers and count the number of landline and mobile numbers.

Hint - First extract the type block from the phone number. With a phone number 827-8102, then mobile block is 81. You'll likely need a UNION here to combine the results from the `agents` and `customers`. Finally, the type block you extract may be a text data type. You can cast it to an integer using `type_block::INTEGER`.

```
WITH agents_with_area_code AS (
  SELECT LEFT(RIGHT(a.agtphonenumber,
    POSITION('-', a.agtphonenumber)), 2) AS type_block
  FROM agents a),
  customers_with_area_code AS (
    SELECT LEFT(RIGHT(c.custphonenumber,
      POSITION('-', c.custphonenumber)), 2) AS type_block
    FROM customers c
  )

SELECT CASE WHEN type_block::INT > 25 THEN 'landline' ELSE 'mobile'
END AS phone_types, COUNT(*) AS num_phone_numbers
FROM (
  SELECT *
  FROM agents_with_area_code
  UNION ALL
  SELECT *
  FROM customers_with_area_code) t1
GROUP BY 1
```

Expected Result:

phone_types	num_phone_numbers
landline	10
mobile	14

6. (2 points) We use the first two digits after the area code of a phone number to determine if the number is a landline or a mobile phone number. For example, if a phone number is 234-2191, then the type block to consider is 21. If the type block is greater than 25, it will be a landline phone number. If it is 25 or less, it is a mobile phone number. Classify all agents and customers phone numbers and count the number of landline and mobile numbers.

Hint - First extract the type block from the phone number. With a phone number 827-8102, then mobile block is 81. You'll likely need a UNION here to combine the results from the `agents` and `customers`. Finally, the type block you extract may be a text data type. You can cast it to an integer using `type_block::INTEGER`.

```
WITH agent_compensation AS (
  SELECT a.agentid, a.salary, SUM(e.contractprice * a.commissionrate)
  AS commission
  FROM engagements e
  JOIN agents a USING (agentid)
  GROUP BY 1, 2),

high_performer_agents AS (
  SELECT a.agentid
  FROM engagements e
  JOIN agents a USING (agentid)
  GROUP BY 1
  HAVING COUNT(e.engagementnumber) > 15),

agent_compensation_raw AS (
```

```

SELECT *,
       CASE WHEN agentid IN (SELECT * FROM high_performer_agents)
            THEN .10 ELSE 0 END AS high_performer_bonus
FROM agent_compensation)

SELECT *, (salary + commission) * (1 + high_performer_bonus)
       AS final_compensation
FROM agent_compensation_raw;

```

Expected Result:

agentid	salary	commission	high_performer_bonus	final_compensation
4	22000	1022.725	0.1	25325.00
6	33000	1466.100	0.0	34466.10
2	27000	268.800	0.0	27268.80
7	22100	372.575	0.0	22472.58
3	30000	1240.000	0.1	34364.00
1	35000	795.800	0.1	39375.38
5	24500	1018.575	0.1	28070.43
8	30000	513.000	0.0	30513.00