Missing Money Matters - Fraud Analysis Case Study

By: Jarrett T. Bruno

Case Study Parameters

- WSDA Music is a fictional company that sells and distributes audio files ranging from music to audiobooks.
- Primary stakeholder: Adam Andrews, Manager of WSDA Music
- There is a discrepancy in the company's financials between the years 2011 and 2012

Deliverables

Utilize SQL to uncover the following:

- ➔ **Get a list of suspects.**
- ➔ **Narrow the list**
- ➔ **Pinpoint the prime suspect(s)**

About the data:

- Used SQLite for analysis
- Dataset contains 11 tables
- For our analysis we'll be using the 'customers', 'employees', and 'invoices' tables

Challenge 1

- How many transactions took place between 2011 and 2012?
- How much money did WSDA Music make between 2011 and 2012?

```sql
-- 1. How many transactions took place between the years 2011 and 2012?

SELECT
  COUNT(*)
FROM
  Invoice
WHERE
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'
```

	COUNT(*)
1	167

There were a total of 167 transactions that took place between 2011 and 2012.

WSDA Music made $1947.97 between 2011 and 2012.

```sql
-- 2. How much money did WSDA Music make during the same period?

SELECT
  SUM(total)
FROM
  Invoice
WHERE
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'
```

	SUM(total)
1	1947.97

Challenge 2

- Get a list of customers who made purchases between 2011 and 2012.
- Get a list of customers, sales reps, and total transaction amounts for each customer between 2011 and 2012.
- How many transactions are above the average transaction amount during the same time period?
- What is the average transaction amount for each year that WSDA Music has been in business?

```sql
-- 3. Get a list of customers who made a purchase between 2011 and 2012.

SELECT
  c.FirstName,
  c.LastName,
  i.total
FROM
  Invoice AS i
INNER JOIN
  Customer AS c
ON
  c.CustomerId = i.CustomerId
WHERE
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'
ORDER BY
  i.total DESC
```

	FirstName	LastName	Total
1	John	Doeein	1000.86
2	Richard	Cunningham	23.86
3	Hugh	O'Reilly	21.86
4	Victor	Stevens	18.86
5	František	Wichterlová	16.86
6	Isabelle	Mercier	16.86
7	Bjørn	Hansen	15.86
8	Fynn	Zimmermann	14.91
9	Enrique	Muñoz	13.86
10	Robert	Brown	13.86
11	Daan	Peeters	13.86
12	Wyatt	Girard	13.86

Execution finished without errors.
Result: 167 rows returned in 23ms

This query returned 167 rows of information relating to the customer's first and last name, and the totals that they've spent between 2011 and 2012.

This query returns a list of all customers, their purchases, and which sales rep fulfilled the purchase.

```sql
-- 4. Get a list of customers, sales reps, and total transaction amounts for each customer between 2011 and 2012.

SELECT
  c.FirstName AS 'Customer First Name',
  c.LastName AS 'Customer Last Name',
  e.FirstName AS 'Employee First Name',
  e.LastName AS 'Employee Last Name',
  i.total
FROM
  Invoice AS i
INNER JOIN
  Customer AS c
ON
  i.CustomerId = c.CustomerId
INNER JOIN
  Employee AS e
ON
  c.SupportRepId = e.EmployeeId
WHERE
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'
ORDER BY
  i.total DESC
```

	Customer First Name	Customer Last Name	Employee First Name	Employee Last Name	Total
1	John	Doeein	Jane	Peacock	1000.86
2	Richard	Cunningham	Margaret	Park	23.86
3	Hugh	O'Reilly	Jane	Peacock	21.86
4	Victor	Stevens	Steve	Johnson	18.86
5	František	Wichterlová	Margaret	Park	16.86
6	Isabelle	Mercier	Jane	Peacock	16.86
7	Bjørn	Hansen	Margaret	Park	15.86
8	Fynn	Zimmermann	Jane	Peacock	14.91
9	Enrique	Muñoz	Steve	Johnson	13.86

Execution finished without errors.

```sql
-- 5. How many transactions are above the average transaction amount during the same time?

-- Find the average transaction amount between 2011 and 2012
SELECT
  round(AVG(total),2) AS 'Average Transaction Amount'
FROM
  Invoice
WHERE
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'

-- Get the number of transactions above the average transaction amount
SELECT
  COUNT(total) AS '# of transactions above avg'
FROM
  Invoice
WHERE
  total >
        (SELECT
          round(AVG(total),2) AS 'Average Transaction Amount'
        FROM
          Invoice
        WHERE
          InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31')
AND
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'
```

	Average Transaction Amount
1	11.66

	# of transactions above avg
1	26

Here we execute the first part of our query to find the average transaction amount. Then we nest that query to find the number of transactions that exceeded the average.

```
95  -- 6. What was the average transaction amount for each year that WSDA Music has been in business?
96  SELECT
97    round(AVG(total),2) AS 'Avg Transaction Amount',
98    strftime('%Y', InvoiceDate) AS 'Year'
99  FROM
100   Invoice
101 GROUP BY
102   strftime('%Y', InvoiceDate )
103
```

	Avg Transaction Amount	Year
1	5.42	2009
2	5.8	2010
3	17.51	2011
4	5.75	2012
5	5.63	2013

After calculating the average transaction amount for each year WSDA Music has been in business, we notice a clear outlier in 2011.

Challenge 3

- Get a list of employees who exceeded the average transaction amount from sales they generated during 2011 and 2012.
- Create a Commission Payout column that displays each employee's commission based on 15% of the sales transaction amount.
- Which employee made the highest commission?
- List the customers that the employee identified in the last question.
- Which customer made the highest purchase?
- Is there anything suspicious?
- Who is our primary person of interest?

```sql
-- 7. Get a list of employees who exceeded the average transaction amount from sales they generated during 2011 and 2012.
SELECT
  e.FirstName,
  e.LastName,
  sum(i.total) AS 'Total Sales'
  FROM
  Invoice AS i
INNER JOIN
  Customer AS c
ON
  i.CustomerId = c.CustomerId
INNER JOIN
  Employee AS e
ON
  e.EmployeeId = c.SupportRepId
WHERE
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'
AND
  i.total > 11.66
GROUP BY
  e.FirstName,
  e.LastName
ORDER BY
  e.LastName
```

	FirstName	LastName	Total Sales
1	Steve	Johnson	100.1
2	Margaret	Park	139.74
3	Jane	Peacock	1137.65

Here we calculated which employees exceeded the average transaction amount and found Jane Peacock has the most total sales over the average by far.

```sql
-- 8. Create a Commission Payout column that displays each employee's commission based on15% of the sales transaction amount.
SELECT
  e.FirstName,
  e.LastName,
  SUM(i.total) AS 'Total Sales',
  round(sum(i.total)*.15,2) AS 'Commission Payout'
FROM
  Invoice AS i
INNER JOIN
  Customer AS c
ON
  i.CustomerId = c.CustomerId
INNER JOIN
  Employee AS e
ON
  e.EmployeeId = c.SupportRepId
WHERE
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'
GROUP BY
  e.FirstName,
  e.LastName
ORDER BY
  e.LastName
```

	FirstName	LastName	Total Sales	Commission Payout
1	Steve	Johnson	293.2	43.98
2	Margaret	Park	322.97	48.45
3	Jane	Peacock	1331.8	199.77

We also see that Jane has the most to gain with the 15% commission split to sales agents. Again we see another outlier here.

```sql
-- 10. List the customers that the employee identified in the last question
SELECT
  c.FirstName AS 'Customer First Name',
  c.LastName AS 'Customer Last Name',
  e.FirstName AS 'Employee First Name',
  e.LastName AS 'Employee Last Name',
  sum(i.total) AS [Total Sales],
  round(sum(i.total)*.15,2) AS 'Commission Payout'
FROM
  Invoice AS i
INNER JOIN
  Customer AS c
ON
  c.CustomerId = i.CustomerId
INNER JOIN
  employee AS e
ON
  e.EmployeeId = c.SupportRepId
WHERE
  InvoiceDate >= '2011-01-01' AND InvoiceDate <= '2012-12-31'
AND
  c.SupportRepId = '3'
GROUP BY
  c.FirstName,
  c.LastName,
  e.FirstName,
  e.LastName
ORDER BY
  [Total Sales] DESC
```

Now, let's take a look at all of Jane Peacock's customers. In the next slide we'll be able to see the results and if any stand out.

The total sales from customer John Doeein far exceed the others.

	Customer First Name	Customer Last Name	Employee First Name	Employee Last Name	Total Sales	Commission Payout
1	John	Doeein	Jane	Peacock	1000.86	150.13
2	Hugh	O'Reilly	Jane	Peacock	32.75	4.91
3	Wyatt	Girard	Jane	Peacock	26.75	4.01
4	Niklas	Schröder	Jane	Peacock	24.75	3.71
5	Puja	Srivastava	Jane	Peacock	24.75	3.71
6	Robert	Brown	Jane	Peacock	24.75	3.71
7	Isabelle	Mercier	Jane	Peacock	19.83	2.97
8	Fynn	Zimmermann	Jane	Peacock	16.89	2.53
9	Terhi	Hämäläinen	Jane	Peacock	16.87	2.53
10	Luís	Gonçalves	Jane	Peacock	16.83	2.52
11	Frank	Ralston	Jane	Peacock	15.88	2.38
12	Edward	Francis	Jane	Peacock	15.84	2.38
13	Tim	Goyer	Jane	Peacock	13.87	2.08
14	Jennifer	Peterson	Jane	Peacock	12.87	1.93
15	Ladislav	Kovács	Jane	Peacock	11.88	1.78
16	Phil	Hughes	Jane	Peacock	11.88	1.78
17	Manoj	Pareek	Jane	Peacock	10.89	1.63
18	Ellie	Sullivan	Jane	Peacock	8.91	1.34
19	Roberto	Almeida	Jane	Peacock	8.91	1.34
20	Emma	Jones	Jane	Peacock	6.93	1.04
21	François	Tremblay	Jane	Peacock	5.94	0.89
22	Michelle	Brooks	Jane	Peacock	2.97	0.45

```sql
--12. Look at this customer record - do you see anything suspicious?
SELECT
  *
FROM
  Customer AS c
WHERE
  c.LastName = 'Doeein'

--Yes, there are no other records pertaining to John Doeein. Our Primary person of interest is Jane Peacock.
```

	CustomerId	FirstName	LastName	Company	Address	City	State	Country	PostalCode	Phone	Fax	Email	SupportRepId
1	60	John	Doeein		NULL	NULL	NULL	NULL	NULL	NULL	NULL		3

Upon trying to pull more information on John Doeein, we notice that there is no other available information about him. We already know that the responsible sales rep is Jane Peacock, who is now the primary suspect in our financial discrepancy.

View All Relevant Files Below

https://github.com/jarrettbruno/MissingMoneyMatters.git