Roman Bureacov

# Query 1: Yearly Expenses

An SQL script that fetches the main PaymentDetails database and references all payment\_id entries combined into one table. It fetches the individually grouped columns of summed payment details, then groups all data by year and sorts it by year, ascending.

|  |
| --- |
| SELECT  YEAR(paid\_date) AS paid\_year,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN SalaryData USING (payment\_id) GROUP BY YEAR(paid\_date)) AS salary\_expenses,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN MaintenanceDetails USING (payment\_id) GROUP BY YEAR(paid\_date)) AS maintenance\_expenses,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN InventoryDetails USING (payment\_id) GROUP BY YEAR(paid\_date)) AS inventory\_expenses,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN AquaticSpeciesList USING (payment\_id) GROUP BY YEAR(paid\_date)) AS aquatic\_species\_expenses  FROM PaymentDetails  GROUP BY paid\_year  ORDER BY paid\_year DESC |

# Query 2: Monthly Expenses Greater Than Threshold

An SQL script that fetches the main PaymentDetails database and references all payment\_id entries combined into one table. It fetches the individually grouped columns of summed payment details, then groups all data by month-year and sorts by total expenses, ascending, if the entries are above the threshold 200,000 in total expenses.

|  |
| --- |
| SELECT  DATE\_FORMAT(paid\_date, '%b-%y') AS formatted\_paid\_date,  SUM(paid\_amount) AS total\_expenses,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN SalaryData USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) AS salary\_expenses,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN MaintenanceDetails USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) AS maintenance\_expenses,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN InventoryDetails USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) AS inventory\_expenses,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN AquaticSpeciesList USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) AS aquatic\_species\_expenses  FROM PaymentDetails  WHERE paid\_amount > 200000  GROUP BY formatted\_paid\_date  ORDER BY total\_expenses DESC |

# Query 3: Employee-Maintenance and Aquatic Species-Inventory

An SQL script that fetches the main PaymentDetails database and references all payment\_id entries combined into one table. It sums the employee salaries and maintenance costs together along with expenses on aquatic species and inventory into another column. Then, data is grouped and sorted monthly, ascending.

|  |
| --- |
| SELECT  DATE\_FORMAT(paid\_date, '%b-%y') AS formatted\_paid\_date,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN SalaryData USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y'))  + (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN MaintenanceDetails USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) AS employee\_maintenance\_costs,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN InventoryDetails USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y'))  + (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN AquaticSpeciesList USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) AS aquatic\_species\_inventory\_expenses  FROM PaymentDetails  GROUP BY formatted\_paid\_date  ORDER BY formatted\_paid\_date ASC |

# Query 4: Monthly Employee Salary Payments

An SQL script that fetches the main PaymentDetails database and references all payment\_id entries to fetch all payments in accordance to SalaryData. Conglomerates spending on all employee salaries and splits it up monthly by year. Orders by paid\_month, ascending.

|  |
| --- |
| SELECT  DATE\_FORMAT(paid\_date, '%b-%y') AS formatted\_paid\_month,  SUM(paid\_amount) AS paid\_amount  FROM PaymentDetails  INNER JOIN SalaryData USING (payment\_id)  GROUP BY formatted\_paid\_month ORDER BY formatted\_paid\_month |

# Query 5: Yearly Aquatic Species Bought

An SQL script that fetches the main PaymentDetails database and references all payment\_id entries to fetch all payments in accordance to AquaticSpeciesList. Conglomerates spending on aquatics species and amount bought; splits it up by year. Sorts in a descending fashion by paid amount per year.

|  |
| --- |
| SELECT  YEAR(paid\_date) AS paid\_year,  SUM(paid\_amount) AS paid\_amount,  count(paid\_amount) AS amount\_bought  FROM PaymentDetails  INNER JOIN AquaticSpeciesList USING (payment\_id)  GROUP BY paid\_year  ORDER BY paid\_amount DESC |

# Query 6: Monthly Maintenance Expenses

An SQL script that fetches the main PaymentDetails database and references all payment\_id entries to fetch all payments in accordance to MaintenanceDetails. Conglomerates spending on maintenance by month.

|  |
| --- |
| SELECT  DATE\_FORMAT(paid\_date, '%b-%y') AS formatted\_paid\_month,  SUM(paid\_amount) AS paid\_amount  FROM PaymentDetails  INNER JOIN MaintenanceDetails USING (payment\_id)  GROUP BY formatted\_paid\_month  ORDER BY formatted\_paid\_month ASC |

# Query 7: Monthly Aquatic Species Bought

An SQL script that fetches the main PaymentDetails table and references all payment\_id entries to fetch all payments in accordance to AquaticSpeciesList. Then, the fish names that were bought are put into a single string of a column, and entries are grouped by month and order by month-year, descending.

|  |
| --- |
| SELECT  DATE\_FORMAT(paid\_date, '%b-%y') AS formatted\_paid\_date,  (SELECT GROUP\_CONCAT(species\_name SEPARATOR ', ') AS fish\_bought FROM PaymentDetails INNER JOIN AquaticSpeciesList USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) As fish\_bought  FROM PaymentDetails  GROUP BY formatted\_paid\_date  ORDER BY formatted\_paid\_date DESC |

# Query 8: Employee Salary vs Other Expenses

An SQL script that fetches the main PaymentDetails table and references all payment\_id entries to fetch all payments in accordance to all other tables. It sets one column as the employee salary, and the other as all other expenses (buying aquatic species, maintenance, and inventory). Groups by month and orders by month-year, ascending.

|  |
| --- |
| SELECT  DATE\_FORMAT(paid\_date, '%b-%y') AS formatted\_paid\_date,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN SalaryData USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) AS employee\_salaries,  (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN AquaticSpeciesList USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y'))  + (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN MaintenanceDetails USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y'))  + (SELECT SUM(paid\_amount) AS paid\_amount FROM PaymentDetails INNER JOIN InventoryDetails USING (payment\_id) GROUP BY DATE\_FORMAT(paid\_date, '%b-%y')) AS other\_expenses  FROM PaymentDetails  GROUP BY formatted\_paid\_date  ORDER BY formatted\_paid\_date ASC |

# Query 9: Annual Inventory Expenses

An SQL script that fetches the main PaymentDetails table and references all payment\_id entries to fetch all payments in accordance to InventoryDetails. Sets one column as the year and the other as the amount spent on inventory after the year 2020. Groups by year, orders by year, descending.

|  |
| --- |
| SELECT  YEAR(paid\_date) AS formatted\_paid\_date,  SUM(paid\_amount) AS amount\_paid  FROM PaymentDetails  INNER JOIN InventoryDetails USING (payment\_id) WHERE paid\_date > 2020  GROUP BY formatted\_paid\_date  ORDER BY formatted\_paid\_date DESC |

# Query 10: Suppliers and Offered Aquatic Species

An SQL script that fetches the main PaymentDetails table and references all payment\_id entries to fetch all payments in accordance to SupplierDetails and AquaticSpeciesList. Takes the supplier name, and concatenates all fish bought from the said supplier into one string to inform where different aquatic species were bought before.

|  |
| --- |
| SELECT  supplier\_name,  GROUP\_CONCAT(species\_name SEPARATOR ', ') AS aquatic\_species\_bought  FROM PaymentDetails  INNER JOIN SupplierDetails USING (supplier\_id)  INNER JOIN AquaticSpeciesList USING (payment\_id)  GROUP BY supplier\_name |