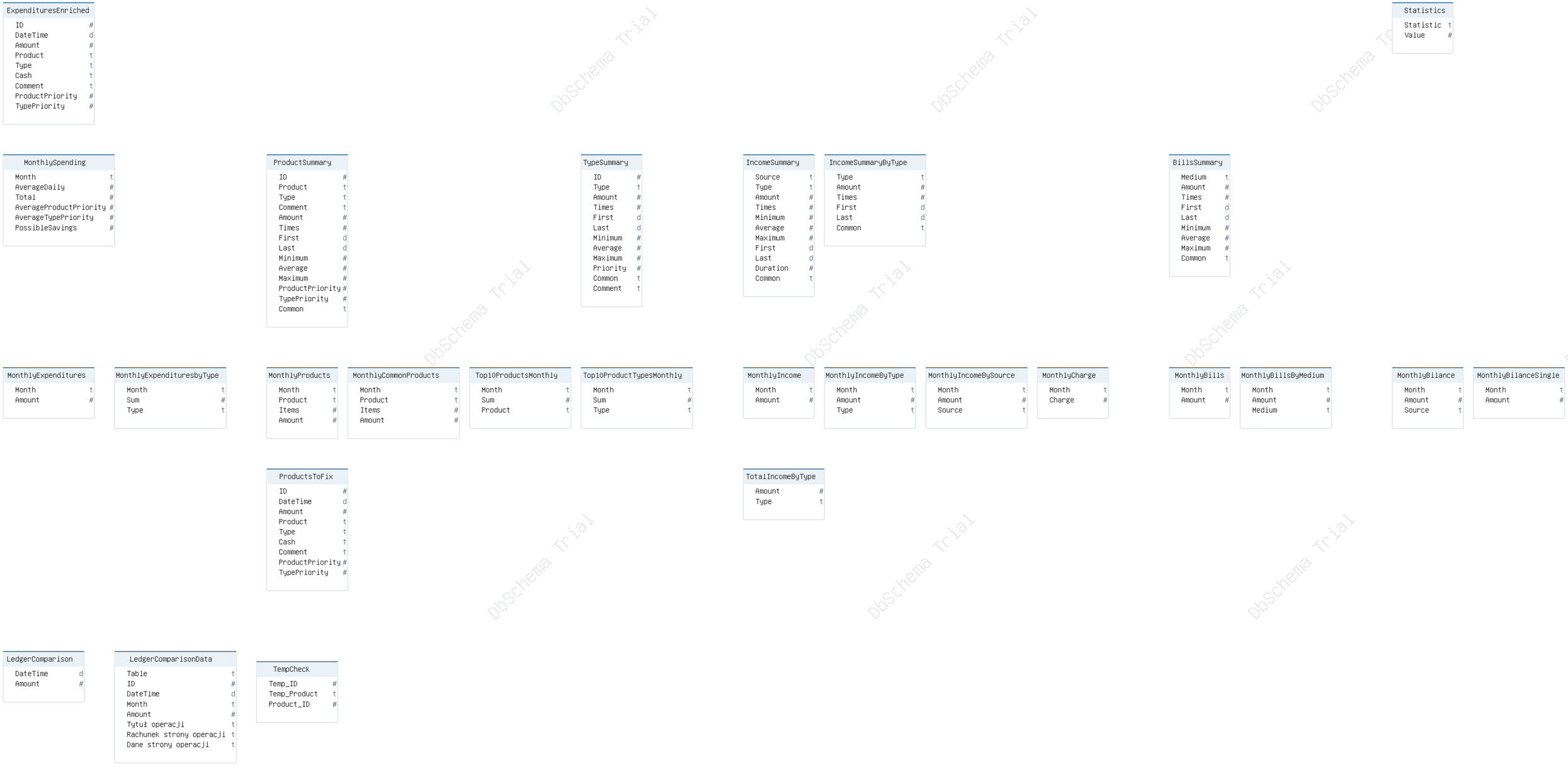
ExpendituresEnriched
ID #
DateTime d
Amount #
Product t
Type t
Cash t
Comment t
ProductPriority #
TypePriority #
Statistics
Statistic t
Value #
MonthlySpending
Month t
AverageDaily #
Total #
AverageProductPriority #
AverageTypePriority #
PossibleSavings #
ProductSummary
ID #
Product t
Type t
Comment t
Amount #
Times #
First d
Last d
Minimum #
Average #
Maximum #
ProductPriority #
TypePriority #
Common t
TypeSummary
ID #
Type t
Amount #
Times #
First d
Last d
Minimum #
Average #
Maximum #
Priority #
Common t
Comment t
IncomeSummary
Source t
Type t
Amount #
Times #
Minimum #
Average #
Maximum #
First d
Last d
Duration #
Common t
IncomeSummaryByType
Type t
Amount #
Times #
First d
Last d
Common t
BillsSummary
Medium t
Amount #
Times #
First d
Last d
Minimum #
Average #
Maximum #
Common t
MonthlyExpenditures
Month t
Amount #
MonthlyExpendituresbyType
Month t
Sum #
Type t
MonthlyProducts
Month t
Product t
Items #
Amount #
MonthlyCommonProducts
Month t
Product t
Items #
Amount #
Top10ProductsMonthly
Month t
Sum #
Product t
Top10ProductTypesMonthly
Month t
Sum #
Type t
MonthlyIncome
Month t
Amount #
MonthlyIncomeByType
Month t
Amount #
Type t
MonthlyIncomeBySource
Month t
Amount #
Source t
MonthlyCharge
Month t
Charge #
MonthlyBills
Month t
Amount #
MonthlyBillsByMedium
Month t
Amount #
Medium t
MonthlyBilance
Month t
Amount #
Source t
MonthlyBilanceSingle
Month t
Amount #
ProductsToFix
ID #
DateTime d
Amount #
Product t
Type t
Cash t
Comment t
ProductPriority #
TypePriority #
TotalIncomeByType
Amount #
Type t
LedgerComparison
DateTime d
Amount #
LedgerComparisonData
Table t
ID #
DateTime d
Month t
Amount #
Tytuł operacji t
Rachunek strony operacji t
Dane strony operacji t
TempCheck
Temp_ID #
Temp_Product t
Product_ID #
DbSchema Trial

Layout

View BillsSummary

```
CREATE VIEW ${nameWithSchemaName} AS SELECT DISTINCT "Bills"."Medium",
    COALESCE("Summary"."Amount", (0)::numeric) AS "Amount",
    COALESCE("Summary"."BoughtTimes", (0)::bigint) AS "Times",
    "Summary"."FirstBought" AS "First",
    "Summary"."LastBought" AS "Last",
    "Summary"."Minimum",
    "Summary"."Average",
    "Summary"."Maximum",
    COALESCE("Summary"."Common", 'Absent'::text) AS "Common"
   FROM ("Bills"
     LEFT JOIN ( SELECT unnamed_subquery."Medium",
            unnamed_subquery."Amount",
            unnamed_subquery."BoughtTimes",
            unnamed_subquery."LastBought",
            unnamed_subquery."FirstBought",
            unnamed_subquery."Average",
            unnamed_subquery."Minimum",
            unnamed_subquery."Maximum",
                CASE
                    WHEN ((unnamed_subquery."BoughtTimes")::numeric > ( SELECT avg(unnamed_subquery_1."BoughtTimes") AS
"Average"
                       FROM ( SELECT "Bills_1"."Medium",
                                round(sum("Bills_1"."Amount"), 2) AS "Amount",
                                count("Bills_1"."DateTime") AS "BoughtTimes",
                                max("Bills_1"."DateTime") AS "LastBought",
                                min("Bills_1"."DateTime") AS "FirstBought"
                               FROM "Bills" "Bills_1"
                              GROUP BY "Bills_1"."Medium"
                              ORDER BY (count("Bills_1"."DateTime")) DESC) unnamed_subquery_1)) THEN 'Common'::text
                    ELSE 'Uncommon'::text
                END AS "Common"
           FROM ( SELECT "Bills_1"."Medium",
                    round(sum("Bills_1"."Amount"), 2) AS "Amount",
                    count("Bills_1"."DateTime") AS "BoughtTimes",
                    max("Bills_1"."DateTime") AS "LastBought",
                    min("Bills_1"."DateTime") AS "FirstBought",
                    round(avg("Bills_1"."Amount"), 2) AS "Average",
                    min("Bills_1"."Amount") AS "Minimum",
                    max("Bills_1"."Amount") AS "Maximum"
                   FROM "Bills" "Bills_1"
                  GROUP BY "Bills_1"."Medium"
                  ORDER BY (count("Bills_1"."DateTime")) DESC) unnamed_subquery) "Summary" ON (("Bills"."Medium" =
"Summary"."Medium")))
```

View ExpendituresEnriched

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "Expenditures"."ID",
    "Expenditures"."DateTime",
    "Expenditures"."Amount",
    "Products"."Product",
    "ProductTypes"."Type",
        CASE
            WHEN ("Expenditures"."isCash" = false) THEN 'NO'::text
            ELSE 'YES'::text
        END AS "Cash",
    "Expenditures"."Comment",
    "Products"."Priority" AS "ProductPriority",
    "ProductTypes"."Priority" AS "TypePriority"
   FROM (("Expenditures"
     LEFT JOIN "Products" ON (("Expenditures"."ProductID" = "Products"."ID")))
     LEFT JOIN "ProductTypes" ON (("Products"."TypeID" = "ProductTypes"."ID")))
  ORDER BY "Expenditures"."DateTime"
```

View IncomeSummary

```
CREATE VIEW ${nameWithSchemaName} AS SELECT DISTINCT "Income"."Source",
    "Summary"."Type",
    COALESCE("Summary"."Amount", (0)::numeric) AS "Amount",
    COALESCE("Summary"."Times", (0)::bigint) AS "Times",
    "Summary"."Minimum",
    "Summary"."Average",
    "Summary"."Maximum",
    "Summary"."First",
    "Summary"."Last",
    ("Summary"."Last" - "Summary"."First") AS "Duration",
    COALESCE("Summary"."Common", 'Absent'::text) AS "Common"
   FROM ("Income"
     LEFT JOIN ( SELECT unnamed_subquery."Source",
            unnamed_subquery."Type",
            unnamed_subquery."Amount",
            unnamed_subquery."Times",
            unnamed_subquery."Last",
            unnamed_subquery."First",
            unnamed_subquery."Average",
            unnamed_subquery."Minimum",
            unnamed_subquery."Maximum",
                CASE
                    WHEN ((unnamed_subquery."Times")::numeric > ( SELECT avg(unnamed_subquery_1."Times") AS "Average"
                       FROM ( SELECT "Income_1"."Source",
                                "Income_1"."Type",
                                round(sum("Income_1"."Amount"), 2) AS "Amount",
                                count("Income_1"."DateTime") AS "Times",
                                max("Income_1"."DateTime") AS "Last",
                                min("Income_1"."DateTime") AS "First"
                               FROM "Income" "Income_1"
                              GROUP BY "Income_1"."Source", "Income_1"."Type"
                              ORDER BY "Income_1"."Source" DESC) unnamed_subquery_1)) THEN 'Common'::text
                    ELSE 'Uncommon'::text
                END AS "Common"
           FROM ( SELECT "Income_1"."Source",
                    "Income_1"."Type",
                    round(sum("Income_1"."Amount"), 2) AS "Amount",
                    count("Income_1"."DateTime") AS "Times",
                    max("Income_1"."DateTime") AS "Last",
                    min("Income_1"."DateTime") AS "First",
                    round(avg("Income_1"."Amount"), 2) AS "Average",
                    min("Income_1"."Amount") AS "Minimum",
                    max("Income_1"."Amount") AS "Maximum"
                   FROM "Income" "Income_1"
                  GROUP BY "Income_1"."Source", "Income_1"."Type"
                  ORDER BY (count("Income_1"."DateTime")) DESC) unnamed_subquery) "Summary" ON (("Income"."Source" =
"Summary"."Source")))
```

View IncomeSummaryByType

```
CREATE VIEW ${nameWithSchemaName} AS SELECT DISTINCT "Summary"."Type",
    COALESCE("Summary"."Amount", (0)::numeric) AS "Amount",
    COALESCE("Summary"."Times", (0)::bigint) AS "Times",
    "Summary"."First",
    "Summary"."Last",
    COALESCE("Summary"."Common", 'Absent'::text) AS "Common"
   FROM ("Income"
     LEFT JOIN ( SELECT unnamed_subquery."Type",
            unnamed_subquery."Amount",
            unnamed_subquery."Times",
            unnamed_subquery."Last",
            unnamed_subquery."First",
                CASE
                    WHEN ((unnamed_subquery."Times")::numeric > ( SELECT avg(unnamed_subquery_1."Times") AS "Average"
                       FROM ( SELECT "Income_1"."Type",
                                round(sum("Income_1"."Amount"), 2) AS "Amount",
                                count(to_char(("Income_1"."DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text)) AS
"Times",
                                max(to_char(("Income_1"."DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text)) AS
"Last",
                                min(to_char(("Income_1"."DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text)) AS
"First"
                               FROM "Income" "Income_1"
                              GROUP BY "Income_1"."Type"
                              ORDER BY "Income_1"."Type" DESC) unnamed_subquery_1)) THEN 'Common'::text
                    ELSE 'Uncommon'::text
                END AS "Common"
           FROM ( SELECT "Income_1"."Type",
                    round(sum("Income_1"."Amount"), 2) AS "Amount",
                    count(to_char(("Income_1"."DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text)) AS "Times",
                    max("Income_1"."DateTime") AS "Last",
                    min("Income_1"."DateTime") AS "First"
                   FROM "Income" "Income_1"
                  GROUP BY "Income_1"."Type"
                  ORDER BY (count(to_char(("Income_1"."DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text))) DESC)
unnamed_subquery) "Summary" ON (("Income"."Type" = "Summary"."Type")))
  ORDER BY COALESCE("Summary"."Times", (0)::bigint) DESC
```

View LedgerComparison

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "DateTime",
    "Amount"
   FROM ( SELECT unnamed_subquery_1."DateTime",
            round(sum(unnamed_subquery_1."Amount"), 2) AS "Amount"
           FROM ( SELECT "LedgerComparisonData"."Table",
                    "LedgerComparisonData"."ID",
                    "LedgerComparisonData"."DateTime",
                    "LedgerComparisonData"."Month",
                    "LedgerComparisonData"."Amount",
                    "LedgerComparisonData"."Tytuł operacji",
                    "LedgerComparisonData"."Rachunek strony operacji",
                    "LedgerComparisonData"."Dane strony operacji"
                   FROM "LedgerComparisonData") unnamed_subquery_1
          GROUP BY unnamed_subquery_1."DateTime"
          ORDER BY unnamed_subquery_1."DateTime") unnamed_subquery
  WHERE (("Amount" > (1)::numeric) OR ("Amount" < ('-1'::integer)::numeric))
```

View LedgerComparisonData

```
CREATE VIEW ${nameWithSchemaName} AS SELECT 'Account data'::text AS "Table",
    "AccountData"."Pozycje historii" AS "ID",
    "AccountData"."Data operacji" AS "DateTime",
    "substring"(to_char(("AccountData"."Data operacji")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7) AS "Month",
    "AccountData"."Kwota" AS "Amount",
    "AccountData"."Tytuł operacji",
    "AccountData"."Rachunek strony operacji",
    "AccountData"."Dane strony operacji"
   FROM "AccountData"
UNION
 SELECT 'ExpendituresEnriched'::text AS "Table",
    "ExpendituresEnriched"."ID",
    "ExpendituresEnriched"."DateTime",
    to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
    "ExpendituresEnriched"."Amount",
    "ExpendituresEnriched"."Product" AS "Tytuł operacji",
    "ExpendituresEnriched"."Comment" AS "Rachunek strony operacji",
    '-'::text AS "Dane strony operacji"
   FROM "ExpendituresEnriched"
  WHERE (("ExpendituresEnriched"."Cash" <> 'YES'::text) AND ("ExpendituresEnriched"."Product" <> 'Automat'::text))
  ORDER BY 3
```

View MonthlyBilance

```
CREATE VIEW ${nameWithSchemaName} AS SELECT to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
    round(sum("Amount"), 2) AS "Amount",
    "Source"
   FROM ( SELECT "Income"."DateTime",
            "Income"."Amount",
            'Income'::text AS "Source"
           FROM "Income"
        UNION
         SELECT "ExpendituresEnriched"."DateTime",
            (- "ExpendituresEnriched"."Amount") AS "Amount",
            'Expenditures'::text AS "Source"
           FROM "ExpendituresEnriched"
        UNION
         SELECT "Bills"."DateTime",
            (- "Bills"."Amount") AS "Amount",
            'Bills'::text AS "Source"
           FROM "Bills") unnamed_subquery
  GROUP BY (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text)), "Source"
  ORDER BY (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text))
```

View MonthlyBilanceSingle

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "Month",
    round(sum("Amount"), 2) AS "Amount"
   FROM "MonthlyBilance"
  GROUP BY "Month"
  ORDER BY "Month"
```

View MonthlyBills

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text),
1, 7) AS "Month",
    round(sum("Amount"), 2) AS "Amount"
   FROM "Bills"
  GROUP BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7))
  ORDER BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7))
```

View MonthlyBillsByMedium

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text),
1, 7) AS "Month",
    round(sum("Amount"), 2) AS "Amount",
    "Medium"
   FROM "Bills"
  GROUP BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7)), "Medium"
  ORDER BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7))
```

View MonthlyCharge

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "Month",
    "Charge"
   FROM ( SELECT to_char(("Income"."DateTime")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
            round((sum("Income"."Amount") / sum("Income"."Hours")), 2) AS "Charge"
           FROM "Income"
          WHERE ("Income"."Type" = ANY (ARRAY['Praca'::text, 'Premia'::text]))
          GROUP BY (to_char(("Income"."DateTime")::timestamp with time zone, 'YYYY-MM'::text))
          ORDER BY (to_char(("Income"."DateTime")::timestamp with time zone, 'YYYY-MM'::text))) unnamed_subquery
  WHERE ("Charge" IS NOT NULL)
```

View MonthlyCommonProducts

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "Month",
    "Product",
    "Items",
    "Amount"
   FROM "MonthlyProducts"
  WHERE (("Items")::numeric > ( SELECT avg("MonthlyProducts_1"."Items") AS avg
           FROM "MonthlyProducts" "MonthlyProducts_1"))
```

View MonthlyExpenditures

```
CREATE VIEW ${nameWithSchemaName} AS SELECT to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
    round(sum("Amount"), 2) AS "Amount"
   FROM "ExpendituresEnriched"
  GROUP BY (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text))
  ORDER BY (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text))
```

View MonthlyExpendituresbyType

```
CREATE VIEW ${nameWithSchemaName} AS SELECT to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
    round(sum("Amount"), 2) AS "Sum",
    "Type"
   FROM ( SELECT "ExpendituresEnriched"."DateTime",
            "ExpendituresEnriched"."Type",
            "ExpendituresEnriched"."Amount"
           FROM "ExpendituresEnriched") unnamed_subquery
  GROUP BY "Type", (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text))
  ORDER BY (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text)) DESC, (round(sum("Amount"), 2)) DESC
```

View MonthlyIncome

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text),
1, 7) AS "Month",
    round(sum("Amount"), 2) AS "Amount"
   FROM "Income"
  GROUP BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7))
  ORDER BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7))
```

View MonthlyIncomeBySource

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text),
1, 7) AS "Month",
    round(sum("Amount"), 2) AS "Amount",
    "Source"
   FROM "Income"
  GROUP BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7)), "Source"
  ORDER BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7))
```

View MonthlyIncomeByType

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text),
1, 7) AS "Month",
    round(sum("Amount"), 2) AS "Amount",
    "Type"
   FROM "Income"
  GROUP BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7)), "Type"
  ORDER BY ("substring"(to_char(("DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text), 1, 7))
```

View MonthlyProducts

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "Month",
    "Product",
    "Items",
    "Amount"
   FROM ( SELECT to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
            "ExpendituresEnriched"."Product",
            count("ExpendituresEnriched"."Product") AS "Items",
            round(sum("ExpendituresEnriched"."Amount"), 2) AS "Amount"
           FROM "ExpendituresEnriched"
          GROUP BY "ExpendituresEnriched"."Product", (to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone,
'YYYY-MM'::text))
          ORDER BY (to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-MM'::text)) DESC,
(round(sum("ExpendituresEnriched"."Amount"), 2)) DESC) unnamed_subquery
```

View MonthlySpending

```
CREATE VIEW ${nameWithSchemaName} AS SELECT to_char(("Day")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
    round(avg("DailyAount")) AS "AverageDaily",
    round(sum("DailyAount")) AS "Total",
    round(avg("AverageDailyProductPriority")) AS "AverageProductPriority",
    round(avg("AverageDailyTypePriority")) AS "AverageTypePriority",
    round(sum("PossibleSavings"), 2) AS "PossibleSavings"
   FROM ( SELECT "ExpendituresEnriched"."DateTime" AS "Day",
            sum("ExpendituresEnriched"."Amount") AS "DailyAount",
            (sum(("ExpendituresEnriched"."Amount" * ("ExpendituresEnriched"."ProductPriority")::numeric)) /
sum("ExpendituresEnriched"."Amount")) AS "AverageDailyProductPriority",
            (sum(("ExpendituresEnriched"."Amount" * ("ExpendituresEnriched"."TypePriority")::numeric)) /
sum("ExpendituresEnriched"."Amount")) AS "AverageDailyTypePriority",
            sum(
                CASE
                    WHEN (("ExpendituresEnriched"."ProductPriority")::numeric < (( SELECT "UserSettings"."Value"
                       FROM "UserSettings"
                      WHERE ("UserSettings"."Setting" = 'ProductPriorityTarget'::text)))::numeric) THEN
"ExpendituresEnriched"."Amount"
                    ELSE (0)::numeric
                END) AS "PossibleSavings"
           FROM "ExpendituresEnriched"
          GROUP BY "ExpendituresEnriched"."DateTime") unnamed_subquery
  GROUP BY (to_char(("Day")::timestamp with time zone, 'YYYY-MM'::text))
  ORDER BY (to_char(("Day")::timestamp with time zone, 'YYYY-MM'::text))
```

View ProductSummary

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "Products"."ID",
    "Products"."Product",
    "ProductTypes"."Type",
    "Products"."Comment",
    COALESCE("Summary"."Amount", (0)::numeric) AS "Amount",
    COALESCE("Summary"."BoughtTimes", (0)::bigint) AS "Times",
    "Summary"."FirstBought" AS "First",
    "Summary"."LastBought" AS "Last",
    "Summary"."Minimum",
    "Summary"."Average",
    "Summary"."Maximum",
    "Products"."Priority" AS "ProductPriority",
    "ProductTypes"."Priority" AS "TypePriority",
    COALESCE("Summary"."Common", 'Absent'::text) AS "Common"
   FROM (("Products"
     LEFT JOIN ( SELECT unnamed_subquery."Product",
            unnamed_subquery."Amount",
            unnamed_subquery."BoughtTimes",
            unnamed_subquery."LastBought",
            unnamed_subquery."FirstBought",
            unnamed_subquery."Average",
            unnamed_subquery."Minimum",
            unnamed_subquery."Maximum",
                CASE
                    WHEN ((unnamed_subquery."BoughtTimes")::numeric > ( SELECT avg(unnamed_subquery_1."BoughtTimes") AS
"Average"
                       FROM ( SELECT "ExpendituresEnriched"."Product",
                                round(sum("ExpendituresEnriched"."Amount"), 2) AS "Amount",
                                count(to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-MM-
DD'::text)) AS "BoughtTimes",
                                max(to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-MM-
DD'::text)) AS "LastBought",
                                min(to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-MM-
DD'::text)) AS "FirstBought"
                               FROM "ExpendituresEnriched"
                              GROUP BY "ExpendituresEnriched"."Product"
                              ORDER BY (count(to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-
MM-DD'::text))) DESC) unnamed_subquery_1)) THEN 'Common'::text
                    ELSE 'Uncommon'::text
                END AS "Common"
           FROM ( SELECT "ExpendituresEnriched"."Product",
                    round(sum("ExpendituresEnriched"."Amount"), 2) AS "Amount",
                    count(to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-MM-DD'::text)) AS
"BoughtTimes",
                    max("ExpendituresEnriched"."DateTime") AS "LastBought",
                    min("ExpendituresEnriched"."DateTime") AS "FirstBought",
                    round(avg("ExpendituresEnriched"."Amount"), 2) AS "Average",
                    min("ExpendituresEnriched"."Amount") AS "Minimum",
                    max("ExpendituresEnriched"."Amount") AS "Maximum"
                   FROM "ExpendituresEnriched"
                  GROUP BY "ExpendituresEnriched"."Product"
                  ORDER BY (count(to_char(("ExpendituresEnriched"."DateTime")::timestamp with time zone, 'YYYY-MM-
DD'::text))) DESC) unnamed_subquery) "Summary" ON (("Products"."Product" = "Summary"."Product")))
     LEFT JOIN "ProductTypes" ON (("Products"."TypeID" = "ProductTypes"."ID")))
```

View ProductsToFix

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "ID",
    "DateTime",
    "Amount",
    "Product",
    "Type",
    "Cash",
    "Comment",
    "ProductPriority",
    "TypePriority"
   FROM "ExpendituresEnriched"
  WHERE (("Product" = ANY (ARRAY[NULL::text, 'UNKNOWN'::text])) OR ("Comment" ~~ '%"TODO"%'::text) OR ("Comment" ~~ '%"do
wyjaśnienia"%'::text) OR ("Comment" ~~ '%"?"%'::text))
```

View Statistics

```
CREATE VIEW ${nameWithSchemaName} AS WITH "StatisticsBase" AS (
         SELECT 'Average income year to date'::text AS "Statistic",
            round(avg(unnamed_subquery."Amount"), 2) AS "Value"
           FROM ( SELECT "MonthlyIncome"."Month",
                    "MonthlyIncome"."Amount"
                   FROM "MonthlyIncome"
                  WHERE ("MonthlyIncome"."Month" < to_char(now(), 'YYYY-MM'::text))
                  ORDER BY "MonthlyIncome"."Month" DESC
                 LIMIT 12) unnamed_subquery
        UNION
         SELECT 'Average bills year to date'::text AS "Statistic",
            round(avg(unnamed_subquery."Amount"), 2) AS "Value"
           FROM ( SELECT "MonthlyBills"."Month",
                    "MonthlyBills"."Amount"
                   FROM "MonthlyBills"
                  WHERE ("MonthlyBills"."Month" < to_char(now(), 'YYYY-MM'::text))
                  ORDER BY "MonthlyBills"."Month" DESC
                 LIMIT 12) unnamed_subquery
        UNION
         SELECT 'Average expenditures year to date'::text AS "Statistic",
            round(avg(unnamed_subquery."Amount"), 2) AS "Value"
           FROM ( SELECT "MonthlyExpenditures"."Month",
                    "MonthlyExpenditures"."Amount"
                   FROM "MonthlyExpenditures"
                  WHERE ("MonthlyExpenditures"."Month" < to_char(now(), 'YYYY-MM'::text))
                  ORDER BY "MonthlyExpenditures"."Month" DESC
                 LIMIT 12) unnamed_subquery
        )
 SELECT "StatisticsBase"."Statistic",
    "StatisticsBase"."Value"
   FROM "StatisticsBase"
UNION
 SELECT 'Average spending year to date'::text AS "Statistic",
    sum("StatisticsBase"."Value") AS "Value"
   FROM "StatisticsBase"
  WHERE ("StatisticsBase"."Statistic" = ANY (ARRAY['Average bills year to date'::text, 'Average expenditures year to
date'::text]))
UNION
 SELECT 'FIRE savings requirement'::text AS "Statistic",
    ((sum("StatisticsBase"."Value") * (12)::numeric) * (25)::numeric) AS "Value"
   FROM "StatisticsBase"
  WHERE ("StatisticsBase"."Statistic" = ANY (ARRAY['Average bills year to date'::text, 'Average expenditures year to
date'::text]))
UNION
 SELECT 'Net Worth'::text AS "Statistic",
    sum("MonthlyBilance"."Amount") AS "Value"
   FROM "MonthlyBilance"
```

View TempCheck

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "Temp_ID",
    "Temp_Product",
    "Product_ID"
   FROM ( SELECT "ExpendituresTransitory"."ID" AS "Temp_ID",
            "ExpendituresTransitory"."Product" AS "Temp_Product",
            "Products"."ID" AS "Product_ID"
           FROM ("ExpendituresTransitory"
             LEFT JOIN "Products" ON (("ExpendituresTransitory"."Product" = "Products"."Product")))) unnamed_subquery
  WHERE ("Product_ID" IS NULL)
```

View Top10ProductTypesMonthly

```
CREATE VIEW ${nameWithSchemaName} AS SELECT to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
    round(sum("Amount"), 2) AS "Sum",
    "Type"
   FROM ( SELECT "ExpendituresEnriched"."DateTime",
            "ExpendituresEnriched"."Type",
            "ExpendituresEnriched"."Amount"
           FROM "ExpendituresEnriched") unnamed_subquery
  WHERE ("Type" IN ( SELECT "TypeSummary"."Type"
           FROM "TypeSummary"
          ORDER BY "TypeSummary"."Times" DESC
         LIMIT 10))
  GROUP BY "Type", (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text))
  ORDER BY (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text)) DESC, (round(sum("Amount"), 2)) DESC
```

View Top10ProductsMonthly

```
CREATE VIEW ${nameWithSchemaName} AS SELECT to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text) AS "Month",
    round(sum("Amount"), 2) AS "Sum",
    "Product"
   FROM ( SELECT "ExpendituresEnriched"."DateTime",
            "ExpendituresEnriched"."Product",
            "ExpendituresEnriched"."Amount"
           FROM "ExpendituresEnriched") unnamed_subquery
  WHERE ("Product" IN ( SELECT "ProductSummary"."Product"
           FROM "ProductSummary"
          ORDER BY "ProductSummary"."Times" DESC
         LIMIT 10))
  GROUP BY "Product", (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text))
  ORDER BY (to_char(("DateTime")::timestamp with time zone, 'YYYY-MM'::text)) DESC, (round(sum("Amount"), 2)) DESC
```

View TotalIncomeByType

```
CREATE VIEW ${nameWithSchemaName} AS SELECT sum("Amount") AS "Amount",
    "Type"
   FROM "IncomeSummaryByType"
  GROUP BY "Type"
  ORDER BY (sum("Amount")) DESC
```

View TypeSummary

```
CREATE VIEW ${nameWithSchemaName} AS SELECT "ProductTypes"."ID",
    "ProductTypes"."Type",
    COALESCE("Summary"."Amount", (0)::numeric) AS "Amount",
    COALESCE("Summary"."Times", (0)::bigint) AS "Times",
    "Summary"."First",
    "Summary"."Last",
    "Summary"."Minimum",
    "Summary"."Average",
    "Summary"."Maximum",
    "ProductTypes"."Priority",
    COALESCE("Summary"."Common", 'Absent'::text) AS "Common",
    "ProductTypes"."Comment"
   FROM ("ProductTypes"
     LEFT JOIN ( SELECT unnamed_subquery."Type",
            unnamed_subquery."Amount",
            unnamed_subquery."Times",
            unnamed_subquery."Last",
            unnamed_subquery."First",
            unnamed_subquery."Minimum",
            unnamed_subquery."Average",
            unnamed_subquery."Maximum",
                CASE
                    WHEN ((unnamed_subquery."Times")::numeric > ( SELECT avg(unnamed_subquery_1."Times") AS "Average"
                       FROM ( SELECT "ExpendituresEnriched"."Type",
                                round(sum("ExpendituresEnriched"."Amount"), 2) AS "Amount",
                                count("ExpendituresEnriched"."DateTime") AS "Times",
                                max("ExpendituresEnriched"."DateTime") AS "Last",
                                min("ExpendituresEnriched"."DateTime") AS "First"
                               FROM "ExpendituresEnriched"
                              GROUP BY "ExpendituresEnriched"."Type"
                              ORDER BY (count("ExpendituresEnriched"."DateTime")) DESC) unnamed_subquery_1)) THEN
'Common'::text
                    ELSE 'Uncommon'::text
                END AS "Common"
           FROM ( SELECT "ExpendituresEnriched"."Type",
                    round(sum("ExpendituresEnriched"."Amount"), 2) AS "Amount",
                    count("ExpendituresEnriched"."DateTime") AS "Times",
                    max("ExpendituresEnriched"."DateTime") AS "Last",
                    min("ExpendituresEnriched"."DateTime") AS "First",
                    min("ExpendituresEnriched"."Amount") AS "Minimum",
                    round(avg("ExpendituresEnriched"."Amount"), 2) AS "Average",
                    max("ExpendituresEnriched"."Amount") AS "Maximum"
                   FROM "ExpendituresEnriched"
                  GROUP BY "ExpendituresEnriched"."Type"
                  ORDER BY (count("ExpendituresEnriched"."DateTime")) DESC) unnamed_subquery) "Summary" ON
(("ProductTypes"."Type" = "Summary"."Type")))
```