POWER BI

TECHNICAL DESIGN

EXECUTIVE COMMITTEE

SALES PERFORMANCE & SCORING

DASHBOARD

Document Information	
Document Title	Power BI Executive Committee Sales Performance & Scoring Dashboard
Document Purpose	This is a technical document which outlines considerations and guidance for intercompany project and developer.
Document Owner	Trakindo Utama

List of Tables in Power BI Report:		
No	**Power BI Table Name**	**Table Source**
1	Category	*(Generated manually on Power Query, static table)*
2	dim_date	EDW_ANALYTICS.dbo.dim_date
3	forecast	EDW_ANALYTICS.CRM.EC_fact_locked_forecast
4	Currency	*(Generated manually on Power Query, static table)*
5	ECC dim_industry	EDW_ANALYTICS.ECC.dim_industry
6	invoiceDataTranform	EDW_ANALYTICS.CRM.EC_fact_invoice_progress
7	invoiceDataNew	EDW_ANALYTICS.CRM.EC_fact_invoice
8	dim_area_store	EDW_ANALYTICS.CRM.EC_dim_area_store
9	dim_exchange_rate	*SharePoint Folder (Exchange Rate EC Dashboard.xlsx)*
10	dim_business_area	edw_mds.ECC.MT_Business_Area
11	dim_area_edit	*(Generated on Power Query, reference to InvoiceDataNew)*
11	invoiceDataLogScore	EDW_ANALYTICS.CRM.EC_fact_scorelog
12	dim_productgroupmapping	*SharePoint Folder (Product Hierarchy.xlsx)*
13	dim_materialType	*(Generated manually on Power Query, static table)*
14	dim_sales_type	*(Generated manually on Power Query, static table)*
15	dim_customerType	*(Generated manually on Power Query, static table)*
16	dim_productModel	*(Generated on Power Query, reference to InvoiceDataNew)*
17	dim_market_sektor_mapping	*SharePoint Folder (dim_market_sektor_mapping.xlsx)*

1. Calculated Measures

1.1. Sales Performance Report

1.1.1. Count Invoiced Sales Document

```
CountSalesDocument =
VAR Hasil =
COUNTROWS(
  FILTER(
  invoiceDataNew,
  invoiceDataNew[BillingDocument] <> "" && invoiceDataNew[BillingDocument] <> "0"
  )
)
```

```
RETURN
IF(Hasil = BLANK(), 0, Hasil)
```

1.1.2. Count Current Month Forecast

```
Current Month forecast =
var hasil =
CALCULATE(
    COUNT(forecast[OpportunityID])
)

return
IF(hasil = BLANK(),0,hasil)
```

1.1.3. Invoiced to Forecast Percentage

```
% of Area = IFERROR(([CountSalesDocument] / [Current Month forecast]),0)
```

1.1.4. Invoiced Value with Conversion Rate

```
SumEstimation Of Invoice =
VAR SelectedCUR =
    SELECTEDVALUE(
       'Currency'[Currency],
        "IDR"
    )

VAR Result =
SUMX(
    invoiceDataNew,
    IF(
        SelectedCUR = "IDR",
        (invoiceDataNew[ActualNetInUSD] * invoiceDataNew[Rate]) / 1000000,
        invoiceDataNew[ActualNetInUSD] / 1000
    )
)
RETURN
IF(Result = BLANK(),0,Result)
```

1.1.5. Conversion Rate for Forecast Estimated Value

```
selected_exchange_rate_forecast =
CALCULATE(
SUM(dim_exchange_rate[KURS_USD]),
MONTH(dim_exchange_rate[DateTo]) = SELECTEDVALUE(dim_date[Full_Date].[MonthNo]) &&
YEAR(dim_exchange_rate[DateTo]) = SELECTEDVALUE(dim_date[Full_Date].[Year]) &&
```

```
dim_exchange_rate[DateTo] = MAX(dim_exchange_rate[DateTo]),
dim_exchange_rate[Type] = "Forecast"
)
```

1.1.6. Forecast Estimated Value with Conversion Rate

```
Sum Of estForecastArea =
VAR SelectedCUR =
    SELECTEDVALUE('Currency'[Currency])

VAR Result =
SUMX(
    forecast,
    IF(
        SelectedCUR = "IDR" && [selected_exchange_rate_forecast] <> BLANK(),
        (forecast[NetValue] * [selected_exchange_rate_forecast]) / 1000000,
        IF(
            SelectedCUR = "IDR" && [selected_exchange_rate_forecast] = BLANK(),
            (forecast[NetValue] * 14000) / 1000000,
            forecast[NetValue] / 1000
        )
    )
)
RETURN
    IF(Result = BLANK(), 0, Result)
```

1.1.7. Invoiced Value to Forecast Value Percentage

```
% of EstArea = IFERROR(([SumEstimation Of Invoice] / [Sum Of estForecastArea]),0)
```

1.2. Scoring Dashboard

1.2.1. Count GCI Product Model

```
CountGCI =
CALCULATE(
COUNT(invoiceDataLogScore[ProductModel]), invoiceDataLogScore[GCI_Status] = "GCI"
)
```

1.2.2. Count Non-GCI Product Model

```
CountGCI =
CALCULATE(
COUNT(invoiceDataLogScore[ProductModel]), invoiceDataLogScore[GCI_Status] = "Non-
GCI"
)
```

1.3. Filter and Title Measures

1.3.1. Filter Dim Product Model

```
Filter Dim Model = COUNTX(RELATEDTABLE(invoiceDataNew), [SalesDocument_])
```

1.3.2. Report Date Title

```
Lasted Date =
VALUES(dim_date[Full_Date].[Month]) &" "&
YEAR(SELECTEDVALUE(dim_date[Full_Date],today())) & " - Final"
```

1.3.3. Selected Material Type Title

```
SelectedMaterialType =
SWITCH(
    TRUE(),
    SELECTEDVALUE(dim_materialType[MaterialType]) = BLANK(), "ALL",
    SELECTEDVALUE(dim_materialType[MaterialType]) = "ENGINE", "ENGINE",
    SELECTEDVALUE(dim_materialType[MaterialType]) = "FORK_LIFT", "FORKLIFT",
    SELECTEDVALUE(dim_materialType[MaterialType]) = "MACHINE", "MACHINE"
)
```

1.3.4. Waterfall Chart Title

```
Chart Title =
VAR selectedmonth = SELECTEDVALUE(dim_date[Full_Date].[Month]) & " " &
SELECTEDVALUE(dim_date[Full_Date].[Year])
VAR selectedarea = SELECTEDVALUE(dim_area_edit[area_name_])
RETURN

IF(
    selectedarea = BLANK(),
    "Invoice Progress Data - All Area (" & selectedmonth & ")",
    "Invoice Progress Data - " & selectedarea & " (" & selectedmonth & ")"
)
```

2. Calculated Column

2.1. Forecast Table

2.1.1. Area Name & Customer Type

```
area_name = RELATED(dim_area_store[area_name])
```

```
area_name_ =
SWITCH(
  TRUE(),
  forecast[MaterialType] IN {"MACHINE", "FORK_LIFT"} &&
    CONTAINSSTRING(forecast[area_name], "Java"), "Java",
  forecast[MaterialType] IN {"MACHINE", "FORK_LIFT"} &&
    CONTAINSSTRING(forecast[area_name], "Sumatera"), "Sumatera",
  forecast[area_name]
)
```

```
Customer_Type =
SWITCH(
    TRUE(),
    CONTAINSSTRING(forecast[area_name_], " MA"), "Major Account", "Retail Account"
)
```

2.1.2. MaterialType Column (Power Query)

```
= Table.AddColumn(#"Removed Columns", "MaterialType", each if [Category_ID] = "Opp
Engine" then "ENGINE" else if [Category_ID] = "Opp  ForkLift" then "FORK_LIFT"
else if [Category_ID] = "Opp  Machine" then "MACHINE" else null)
```

2.2. Invoice Table

2.2.1. Actual Net In IDR

```
ActualNetInIDR_ = invoiceDataNew[ActualNetInUSD] * invoiceDataNew[Rate]
```

2.2.2. Invoice Progress Category

```
CatGroup =
SWITCH(
    TRUE(),
    invoiceDataNew[Workable] = "Yes"
    && OR(invoiceDataNew[ConfidenceLevel] >= 75,
      invoiceDataNew[ConfidenceLevel] = BLANK())
    && invoiceDataNew[StatusPGI] = "C"
    && invoiceDataNew[DeliveryDate] <> BLANK()
    && invoiceDataNew[BASTSigndate] <> BLANK()
    && invoiceDataNew[BillingDocument] <> ""
    && invoiceDataNew[BillingDocument] <> "0", "Invoiced",

    invoiceDataNew[Workable] = "Yes"
    && OR(invoiceDataNew[ConfidenceLevel] >= 75,
      invoiceDataNew[ConfidenceLevel] = BLANK())
    && invoiceDataNew[ReleaseApproval] = "Yes"
    && invoiceDataNew[StatusPGI] = "C"
    && invoiceDataNew[DeliveryDate] <> BLANK()
```

```
    && invoiceDataNew[BASTSigndate] <> BLANK(), "BAST",

    invoiceDataNew[sales_type] IN {"ST3", "ST5"}
    && invoiceDataNew[BillingDocument] <> ""
    && invoiceDataNew[BillingDocument] <> BLANK(), "Invoiced",

    invoiceDataNew[Workable] = "Yes"
    && OR(invoiceDataNew[ConfidenceLevel] >= 75,
      invoiceDataNew[ConfidenceLevel] = BLANK())
    && invoiceDataNew[ReleaseApproval] = "Yes"
    && invoiceDataNew[StatusPGI] = "C"
    && invoiceDataNew[DeliveryDate] <> BLANK(), "Delivery to Cust Site",

    invoiceDataNew[Workable] = "Yes"
    && OR(invoiceDataNew[ConfidenceLevel] >= 75,
      invoiceDataNew[ConfidenceLevel] = BLANK())
    && invoiceDataNew[ReleaseApproval] = "Yes"
    && invoiceDataNew[StatusPGI] = "C", "GI",

    invoiceDataNew[Workable] = "Yes"
    && OR(invoiceDataNew[ConfidenceLevel] >= 75,
      invoiceDataNew[ConfidenceLevel] = BLANK())
    && invoiceDataNew[ReleaseApproval] = "Yes", "RA",

    invoiceDataNew[POLeasing] = "Yes", "PO Leasing",

    invoiceDataNew[Workable] = "Yes"
    && OR(invoiceDataNew[ConfidenceLevel] >= 75,
      invoiceDataNew[ConfidenceLevel] = BLANK()),
    "Workable SO (DP Paid)",

    "High Confidence"
)
```

2.2.3. Financing Company

```
FinancingCompany_ =
IF(
    invoiceDataNew[FinancingCompany] = BLANK(),
    invoiceDataNew[Payer],
    invoiceDataNew[FinancingCompany]
)
```

2.2.4. Make Column

```
Make =
SWITCH(
    TRUE(),
    invoiceDataNew[SalesDocumentType] = "ZEPP", "AA",
    invoiceDataNew[SalesDocumentType] <> "ZEPP"
```

```
        && invoiceDataNew[MaterialType] = "MACHINE", "AA-MACH",
    invoiceDataNew[SalesDocumentType] <> "ZEPP"
        && invoiceDataNew[MaterialType] = "ENGINE", "AA-ENGN",
    invoiceDataNew[SalesDocumentType] <> "ZEPP"
        && invoiceDataNew[MaterialType] = "FORK_LIFT", "AA-FLT"
)
```

2.2.5. MappingID Column

```
MappingID = invoiceDataNew[MarketSector] & "-" & invoiceDataNew[MaterialType]
```

2.2.6. Payername Column

```
PayerName_ =
SWITCH(
    TRUE(),
    invoiceDataNew[Payername] = BLANK(), invoiceDataNew[AccountName],
    invoiceDataNew[Payername]
)
```

2.2.7. ProductModel Column

```
ProductModel_ =
SWITCH(
    TRUE(),
    invoiceDataNew[ProductModel] = BLANK(), invoiceDataNew[Model],
    invoiceDataNew[ProductModel]
)
```

2.2.8. Sales Office Code Column

```
sales_off_code_ =
IF(
    ISBLANK(invoiceDataNew[sales_off_code]),
    invoiceDataNew[SalesOffice],
    invoiceDataNew[sales_off_code]
)
```

2.2.9. Sales Document and Sales Document Item Column

```
SalesDocumentNo =
invoiceDataNew[SalesDocument_] & "-" & invoiceDataNew[SalesDocumentItem]
```

2.2.10. Invoice Status Column (Power Query)

```
= Table.AddColumn(dbo_invoiceDataNew, "invStatus", each if [isForecast] = "Yes"
then "Forecasted" else "Unforecasted")
```

2.2.11. Created Column (Power Query)

```
= Table.AddColumn(#"Added Conditional Column", "Created_", each if [SOID] = null
then "Yes" else if [Created] = null then "Yes" else [Created])
```

2.3. Scoring Table

2.3.1. GCI Status

```
GCI_Status = RELATED(dim_productgroupmapping[GCI_Status])
```

2.3.2. MappingID Column

```
MappingID =
invoiceDataLogScore[MarketSector] & "-" & invoiceDataLogScore[MaterialType]
```

2.3.3. Scoring Status Column

```
Status_ =
SWITCH(
    TRUE(),
    invoiceDataLogScore[Status] = "BACK OUT", "UNSCORED",
    invoiceDataLogScore[Status] = "EX-BACK OUT", "UNSCORED",
    invoiceDataLogScore[Status] = "FAILED", "UNSCORED",
    invoiceDataLogScore[Status] = "OTHER DEALER", "UNSCORED",
    invoiceDataLogScore[Status] = "INVOICED PREVIOUS MONTH", "SCORED",
    invoiceDataLogScore[Status] = "NEW UNIT", "SCORED"
)
```

3. Stored Procedures

This section contains information regarding stored procedures and which tables they produce.

3.1. EDW_ANALYTICS.CRM.sp_EC_Insert_MappingAreaSales:

- Generate table EDW_ANALYTICS.CRM.EC_dim_area_store
- Generate table EDW_ANALYTICS.CRM.EC_dim_area_sales

3.2. EDW_ANALYTICS.CRM.sp_EC_Insert_LockedForecast:

- Insert values to table EDW_ANALYTICS.CRM.EC_fact_locked_forecast

3.3. EDW_ANALYTICS.CRM.sp_EC_Insert_InvoiceActual:

- Insert values to table EDW_ANALYTICS.CRM.EC_fact_invoice

3.4. EDW_ANALYTICS.CRM.sp_EC_Update_RateInvoiceActual:

- Update conversion rate column in EDW_ANALYTICS. CRM.EC_fact_invoice

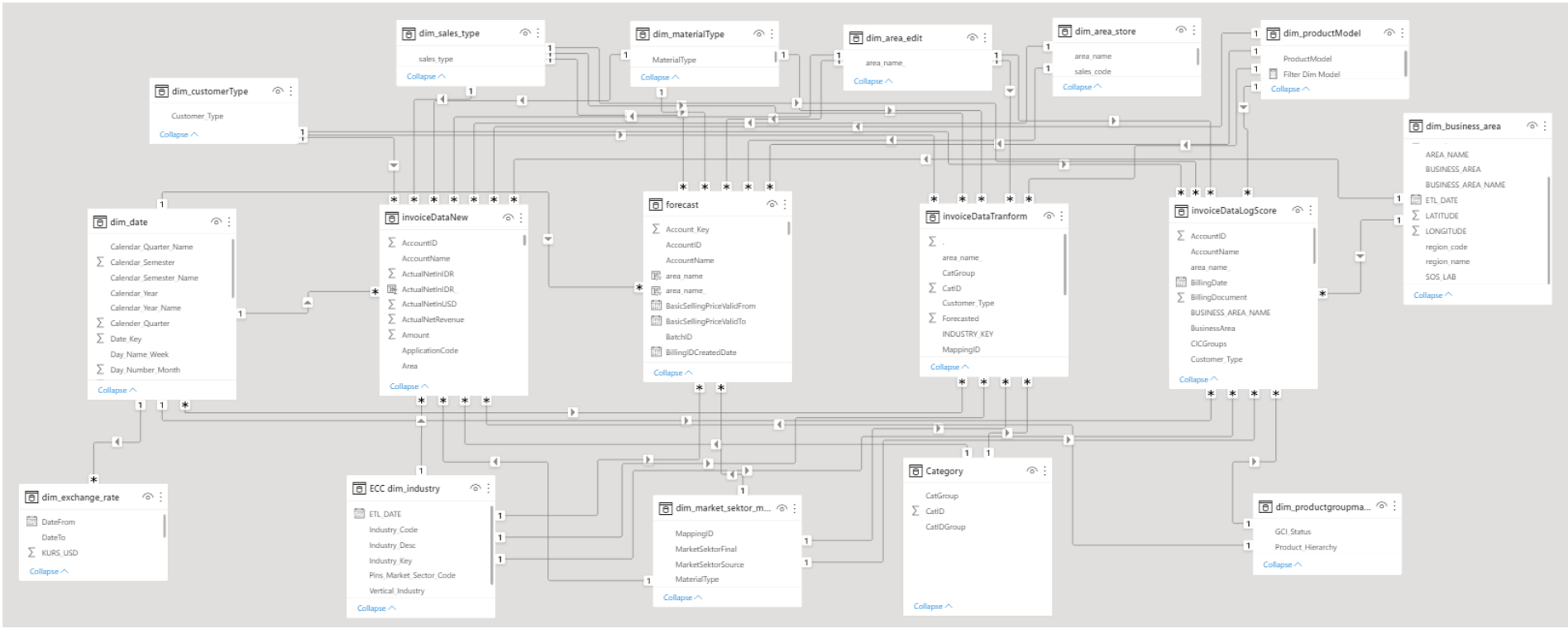
3.5. EDW_ANALYTICS.CRM.sp_Insert_InvoiceProgress:

- Insert values to table EDW_ANALYTICS.CRM.EC_fact_invoice_progress

3.6. EDW_ANALYTICS.CRM.sp_Insert_Scorelog:

- Insert values to table EDW_ANALYTICS.CRM.EC_fact_scorelog

4. Relationship Model



Active	From: Table (Column)	To: Table (Column)
Yes	dim_exchange_rate (DateTo)	dim_date (Full_Date)
Yes	forecast (area_name_)	dim_area_edit (area_name_)
Yes	forecast (Customer_Type)	dim_customerType (Customer_Type)
Yes	forecast (DeliveryDate)	dim_date (Full_Date)
Yes	forecast (INDUSTRY_KEY)	ECC dim_industry (Industry_Key)
Yes	forecast (MappingID)	dim_market_sector (MappingID)
Yes	forecast (MaterialType)	dim_materialType (MaterialType)
Yes	forecast (ProductModel)	dim_productModel (ProductModel)
Yes	forecast (sales_off_code)	dim_area_store (sales_code)
Yes	forecast (sales_type)	dim_sales_type (sales_type)
Yes	invoiceDataLogScore (area_name_)	dim_area_edit (area_name_)
Yes	invoiceDataLogScore (BusinessArea)	dim_business_area (BUSINESS_AREA)
Yes	invoiceDataLogScore (Customer_Type)	dim_customerType (Customer_Type)
Yes	invoiceDataLogScore (INDUSTRY_KEY)	ECC dim_industry (Industry_Key)
Yes	invoiceDataLogScore (MappingID)	dim_market_sector (MappingID)
Yes	invoiceDataLogScore (MTD)	dim_date (Full_Date)
Yes	invoiceDataLogScore (Product_Hierarchy)	dim_productgroupmapping (Product_Hierarchy)
Yes	invoiceDataLogScore (ProductModel)	dim_productModel (ProductModel)
Yes	invoiceDataLogScore (sales_type)	dim_sales_type (sales_type)
Yes	invoiceDataNew (area_name_)	dim_area_edit (area_name_)
Yes	invoiceDataNew (BusinessArea)	dim_business_area (BUSINESS_AREA)
Yes	invoiceDataNew (CatGroup)	Category (CatGroup)
Yes	invoiceDataNew (Customer_Type)	dim_customerType (Customer_Type)
Yes	invoiceDataNew (INDUSTRY_KEY)	ECC dim_industry (Industry_Key)
Yes	invoiceDataNew (MappingID)	dim_market_sector (MappingID)
Yes	invoiceDataNew (MaterialType)	dim_materialType (MaterialType)
Yes	invoiceDataNew (MTD)	dim_date (Full_Date)
Yes	invoiceDataNew (Product_Hierarchy)	dim_productgroupmapping (Product_Hierarchy)
Yes	invoiceDataNew (ProductModel)	dim_productModel (ProductModel)
Yes	invoiceDataNew (sales_off_code)	dim_area_store (sales_code)
Yes	invoiceDataNew (sales_type)	dim_sales_type (sales_type)
Yes	invoiceDataTranform (area_name_)	dim_area_edit (area_name_)
Yes	invoiceDataTranform (CatGroup)	Category (CatGroup)
Yes	invoiceDataTranform (Customer_Type)	dim_customerType (Customer_Type)
Yes	invoiceDataTranform (INDUSTRY_KEY)	ECC dim_industry (Industry_Key)
Yes	invoiceDataTranform (MappingID)	dim_market_sector (MappingID)
Yes	invoiceDataTranform (MaterialType)	dim_materialType (MaterialType)
Yes	invoiceDataTranform (MonthDeliveryDate)	dim_date (Month_Number_Year)
Yes	invoiceDataTranform (ProductModel)	dim_productModel (ProductModel)
Yes	invoiceDataTranform (sales_type)	dim_sales_type (sales_type)