Documentation automatique du modèle Power BI

# Table: CONSOLIDE\_CentreDeCouts

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| CodePart1 - Département | left(CONSOLIDE\_CentreDeCouts[Code],2) |
| CodePart2 - Nature | mid(CONSOLIDE\_CentreDeCouts[Code],3,1) |
| CodePart3 - Sous-nature | mid(CONSOLIDE\_CentreDeCouts[Code], 4, 15)  //& " - " &COLLIGNON\_CentreDeCouts[Name] |
| Nom rubrique | related('Rubrique1\_Désignation'[Rubrique 1]) |
| Niveau 1 Matrice (nature) | if(CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature]= "040500", "Exceptionnel",  if(CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature] = "099999", "AUTRE-Imputation %CA",  if(CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature]="100100", "Pertes à terminaison",  CONSOLIDE\_CentreDeCouts[Niveau 1 Matrice (nature) OLD])))  // |
| Niveau 4 Matrice (sous-nature) | if(CONSOLIDE\_CentreDeCouts[Niveau 1 Matrice (nature)] = "Chantiers", "",  if(CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] = "U", "",  if(CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] = "X", "",  if(CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] = "8", "Devis",  if(CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] = "E", "",  if(CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] = "V", "",  if(CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] = "9", "",  (CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature])))))))) |
| Index | if(isblank(  switch(CONSOLIDE\_CentreDeCouts[Niveau 1 Matrice (nature)],  "Total Chantiers", "01",  "Frais Généraux", "02",  "Coûts Salariaux", "03",  "Utilitaires", "04",  "Machines", "05",  "Véhicules", "06",  "X", "07",  "9", "08",  "AUTRE-Imputation %CA", "09",  "Impôts", "10",  "Pertes à terminaison", "11",  "Exceptionnel", "12",  "Frais Financiers", "13"  )), "15",  switch(CONSOLIDE\_CentreDeCouts[Niveau 1 Matrice (nature)],  "Total Chantiers", "01",  "Frais Généraux", "02",  "Coûts Salariaux", "03",  "Utilitaires", "04",  "Machines", "05",  "Véhicules", "06",  "X", "07",  "9", "08",  "AUTRE-Imputation %CA", "09",  "Impôts", "10",  "Pertes à terminaison", "11",  "Exceptionnel", "12",  "Frais Financiers", "13"  )) |
| IndexCodePart3 | CONCATENATE(CONSOLIDE\_CentreDeCouts[Index], CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature]) |
| Niveau 2 Matrice (S) | if(CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] = "S",  switch(CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature],  "600000", "1. Ecarts salariaux Intérimaires",  "200000", "2. Ecarts salariaux Ouvriers",  "110000", "3. Ecarts salariaux Techniciens",  "120000", "4. Ecarts salariaux RA",  "100000", "5. Ecarts salariaux Employés"),  if(CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] = "U",  if( CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature] = "100960" || CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature] = "100961" || CONSOLIDE\_CentreDeCouts[CodePart3 - Sous-nature] = "040000", "Dépenses/outillages/EPI", "Camionnettes"), "")) |
| Niveau 1 Matrice (nature) OLD | if(isblank(  switch( CONSOLIDE\_CentreDeCouts[CodePart2 - Nature],  "0", "Total Chantiers",  "1", "Total Chantiers",  "2", "Total Chantiers",  "3", "Total Chantiers",  "4", "Total Chantiers",  "5", "Total Chantiers",  "A", "Frais Généraux",  "8", "Frais Généraux",  "I", "Impôts",  "F", "Frais Financiers",  "S", "Coûts Salariaux",  "U", "Utilitaires",  "V", "Véhicules",  "E", "Machines",  "M", "Total Chantiers",//rajout FEXIM  "T", "Total Chantiers", //rajout FEXIM  "R", "Total Chantiers", //rajout FEXIM  "G", "Total Chantiers",//rajout FEXIM  "P", "Total Chantiers")), //rajout FEXIM  CONSOLIDE\_CentreDeCouts[CodePart2 - Nature],  switch( CONSOLIDE\_CentreDeCouts[CodePart2 - Nature],  "0", "Total Chantiers",  "1", "Total Chantiers",  "2", "Total Chantiers",  "3", "Total Chantiers",  "4", "Total Chantiers",  "5", "Total Chantiers",  "A", "Frais Généraux",  "8", "Frais Généraux",  "F", "Frais Financiers",  "I", "Impôts",  "S", "Coûts Salariaux",  "U", "Utilitaires",  "V", "Véhicules",  "E", "Machines",  "M", "Total Chantiers",//rajout FEXIM  "T", "Total Chantiers",//rajout FEXIM  "R", "Total Chantiers",//rajout FEXIM  "G", "Total Chantiers",//rajout FEXIM  "P", "Total Chantiers"))//rajout FEXIM  //COLLIGNON\_CentreDeCouts[CodePart2 - Nature])))))))) |
| Niveau 1\_Index | CONSOLIDE\_CentreDeCouts[Index] & ". "&CONSOLIDE\_CentreDeCouts[Niveau 1 Matrice (nature)] |
| CostCenterKEYCompany | CONSOLIDE\_CentreDeCouts[Code]&CONSOLIDE\_CentreDeCouts[CC\_SCTE] |
| Index 2 | SWITCH(  TRUE,  ISBLANK('CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)]),  "(Vide)",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Total Chantiers"},  "01",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Frais Généraux"},  "02",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Coûts Salariaux"},  "03",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Utilitaires"},  "04",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Machines"},  "05",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Véhicules"},  "06",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"X"},  "07",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"9"},  "08",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"AUTRE-Imputation %CA"},  "09",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Impôts"},  "10",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Exceptionnel"},  "12",  'CONSOLIDE\_CentreDeCouts'[Niveau 1 Matrice (nature)] IN {"Frais Financiers"},  "13",  "15"  ) |
| No+Name | CONSOLIDE\_CentreDeCouts[Code]& " - "& CONSOLIDE\_CentreDeCouts[Name] |
| JobEmplName | related(CONSOLIDE\_JOB[EMPLOYEE.Name]) |
| Interne/Externe | related(CONSOLIDE\_JOB[Interne/Externe]) |
| Clients | RELATED(CONSOLIDE\_JOB[Bill-to Name]) |
| Status EEB | RELATED(CONSOLIDE\_JOB[Status EEB]) |
| Date de début | RELATED(CONSOLIDE\_JOB[Creation Date]) |
| CA Année en cours | calculate( COUNT(CONSOLIDE\_EcrituresCouts\_2022[Amount]), year(CONSOLIDE\_EcrituresCouts\_2022[Posting Date]) = year(today())) |
| chantiers\_Cloturés\_Annee | if(and(CONSOLIDE\_CentreDeCouts[Status EEB] = "CHANTIER CLOTURE", CONSOLIDE\_CentreDeCouts[CA Année en cours] <> blank()), true(), false()) |
| Statut EEB New | if(CONSOLIDE\_CentreDeCouts[chantiers\_Cloturés\_Annee] = true(),  "CHANTIER CLOTURE ANNEE EN COURS", CONSOLIDE\_CentreDeCouts[Status EEB]) |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_CentreDeCouts | import | m |  |

# Table: Rubrique1\_Désignation

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| Rubrique1\_Désignation | import | m |  |

# Table: CONSOLIDE\_ObjetsDeCouts

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Dépenses/recettes RAP208 | if(  isblank(switch (CONSOLIDE\_ObjetsDeCouts[Code],  "AUX0699", "Recettes",  "AUX0299", "Recettes",  "AUX0199", "Recettes",  "099999", "Recettes",  "VEH9998", "Recettes",  "VEN-2013", "Recettes",  "VEN-2014", "Recettes",  "VEN-2015", "Recettes",  "VEN-2023", "Recettes",  "VEN-2024", "Recettes",  "DIV0401", "Recettes",  "DIV0405", "Recettes",  "DEP9996", "Recettes",  "DEP9997", "Recettes",  "DEP9998", "Recettes",  "DIV9999", "Recettes",  "VEH9999", "Recettes")),  "Dépenses",  switch (CONSOLIDE\_ObjetsDeCouts[Code],  "AUX0699", "Recettes",  "AUX0299", "Recettes",  "AUX0199", "Recettes",  "099999", "Recettes",  "VEH9998", "Recettes",  "VEN-2013", "Recettes",  "VEN-2014", "Recettes",  "VEN-2015", "Recettes",  "VEN-2023", "Recettes",  "VEN-2024", "Recettes",  "DIV0401", "Recettes",  "DIV0405", "Recettes",  "DEP9996", "Recettes",  "DEP9997", "Recettes",  "DIV9999", "Recettes",  "DEP9998", "Recettes",  "VEH9999", "Recettes")) |
| Dépenses/Recettes/CA Collignon | SWITCH(  TRUE,  ISBLANK('CONSOLIDE\_ObjetsDeCouts'[Code]),  "(Vide)",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DIV0401",  "DIV0402",  "DIV0405",  "VEN-2013",  "VEN-2014",  "VEN-2015",  "VEN-2016",  "VEN-2017",  "VEN-2018",  "VEN-2019",  "VEN-2020",  "VEN-2021",  "VEN-2022",  "VEN-2023",  "VEN-2024",  "VEN-2025",  "VEN7170"},  "Chiffre d'Affaires",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"9999999",  "AUX0100",  "AUX0200",  "AUX0600",  "AUX1101",  "AUX1103",  "AUX1104",  "AUX1105",  "AUX1106",  "AUX1107",  "DEP0101",  "DEP0102",  "DEP0103",  "DEP0104",  "DEP0105",  "DEP0106",  "DEP0107",  "DEP0108",  "DEP0198",  "DEP0199",  "DEP0201",  "DEP0202",  "DEP0203",  "DEP0299",  "DEP0301",  "DEP0302",  "DEP0303",  "DEP0304",  "DEP0305",  "DEP0306",  "DEP0307",  "DEP0308",  "DEP0309",  "DEP0310",  "DEP0311",  "DEP0312",  "DEP0313",  "DEP0314",  "DEP0399",  "DEP0401",  "DEP0402",  "DEP0403",  "DEP0499",  "DEP0501",  "DEP0502",  "DEP0503",  "DEP0504",  "DEP0505",  "DEP0506",  "DEP0599",  "DEP0601",  "DEP0602",  "DEP0603",  "DEP0604",  "DEP0699",  "DEP0701",  "DEP0702",  "DEP0703",  "DEP0704",  "DEP0705",  "DEP0708",  "DEP0710",  "DEP0799",  "DEP0801",  "DEP0802",  "DEP0803",  "DEP0804",  "DEP0805",  "DEP0806",  "DEP0807",  "DEP0808",  "DEP0809",  "DEP0810",  "DEP0811",  "DEP0812",  "DEP0813",  "DEP0814",  "DEP0899",  "DEP0901",  "DEP0902",  "DEP0903",  "DEP0998",  "DEP0999",  "DEP1001",  "DEP1002",  "DEP1003",  "DEP1004",  "DEP1005",  "DEP1006",  "DEP1007",  "DEP1008",  "DEP1009",  "DEP1010",  "DEP1013",  "DEP1014",  "DEP1015",  "DEP1016",  "DEP1017",  "DEP1018",  "DEP1099",  "DEP1101",  "DEP1102",  "DEP1103",  "DEP1104",  "DEP1105",  "DEP1106",  "DEP1107",  "DEP1201",  "DEP1202",  "DEP1203",  "DEP1301",  "DEP1302",  "DEP1303",  "DEP1304",  "DEP1305",  "DEP1306",  "DEP1307",  "DEP1308",  "DEP1309",  "DEP1310",  "DEP1311",  "DEP1312",  "DEP1313",  "DEP1314",  "DEP1315",  "DEP1316",  "DEP1317",  "DEP1318",  "DEP1319",  "DEP1320",  "DEP1321",  "DEP1322",  "DEP1323",  "DEP1324",  "DEP1325",  "DEP1326",  "DEP1327",  "DEP1328",  "DEP1329",  "DEP1330",  "DEP1331",  "DEP1332",  "DEP1333",  "DEP1334",  "DEP1335",  "DEP1336",  "DEP1337",  "DEP1399",  "DEP1401",  "DEP1402",  "DEP1403",  "DEP1404",  "DEP1405",  "DEP1406",  "DEP1407",  "DEP1408",  "DEP1409",  "DEP1410",  "DEP1411",  "DEP1412",  "DEP1413",  "DEP1414",  "DEP1415",  "DEP1416",  "DEP1417",  "DEP1499",  "DEP1501",  "DEP1502",  "DEP1503",  "DEP1504",  "DEP1505",  "DEP1599",  "DEP1601",  "DEP1602",  "DEP1603",  "DEP1604",  "DEP1605",  "DEP1606",  "DEP1607",  "DEP1608",  "DEP1609",  "DEP1610",  "DEP1611",  "DEP1612",  "DEP1613",  "DEP1614",  "DEP1615",  "DEP1616",  "DEP1617",  "DEP1618",  "DEP1619",  "DEP1620",  "DEP1621",  "DEP1622",  "DEP1623",  "DEP1624",  "DEP1625",  "DEP1626",  "DEP1627",  "DEP1628",  "DEP1629",  "DEP1630",  "DEP1631",  "DEP1632",  "DEP1633",  "DEP1634",  "DEP1635",  "DEP1636",  "DEP1699",  "DEP1701",  "DEP1702",  "DEP1703",  "DEP1704",  "DEP1705",  "DEP1706",  "DEP1707",  "DEP1708",  "DEP1709",  "DEP1710",  "DEP1711",  "DEP1801",  "DEP1802",  "DEP1803",  "DEP1804",  "DEP1805",  "DEP1806",  "DEP1807",  "DEP1901",  "DEP1902",  "DEP1903",  "DEP1904",  "DEP1905",  "DEP1906",  "DEP1907",  "DEP1908",  "DEP1909",  "DEP1910",  "DEP2001",  "DEP2002",  "DEP2003",  "DEP2004",  "DEP2005",  "DEP2006",  "DEP2007",  "DEP2008",  "DEP2009",  "DEP2010",  "DEP2011",  "DEP2012",  "DEP610000",  "DEP610200",  "DEP610250",  "DEP610900",  "DEP612000",  "DEP612010",  "DEP612020",  "DEP612030",  "DEP612170",  "DEP612310",  "DEP613000",  "DEP613100",  "DEP613400",  "DEP613410",  "DEP614000",  "DEP614002",  "DEP614100",  "DEP614104",  "DEP614200",  "DEP614201",  "DEP614300",  "DEP614301",  "DEP614310",  "DEP614312",  "DEP614313",  "DEP614600",  "DEP699994",  "DEP699995",  "DEP699996",  "DEP699997",  "DEP699998",  "DEP699999",  "DEP9998",  "DIV0101",  "DIV0201",  "DIV0301",  "DIV0404",  "DIV0501",  "DIV0601",  "DIV0701",  "DIV0801",  "DIV1013",  "DIV6300",  "DIV6310",  "DIV6370",  "DIV6700",  "EIF0024",  "EIF0001",  "EIF0025",  "EIF0026",  "EIF0002",  "EIF0027",  "EIF0003",  "EIF0004",  "EIF0005",  "EIF0032",  "EIF0006",  "EIF0033",  "EIF0007",  "EIF0034",  "EIF0008",  "EIF0035",  "EIF0009",  "EIF0036",  "EIF0010",  "EIF0037",  "EIF0011",  "EIF0038",  "EIF0012",  "EIF0039",  "EIF0013",  "EIF0040",  "EIF0014",  "EIF0041",  "EIF0015",  "EIF0042",  "EIF0016",  "EIF0043",  "EIF0017",  "EIF0044",  "EIF0018",  "EIF0045",  "EIF0019",  "EIF0046",  "EIF0020",  "EIF0047",  "EIF0021",  "EIF0048",  "EIF0022",  "EIF1201",  "EIF0023",  "EIF1202",  "EIF1203",  "EIF1204",  "EIF1205",  "EIF1206",  "EIF1207",  "EIF1301",  "EIF1302",  "EIF1303",  "EIF1304",  "EIF1305",  "EIF1306",  "EIF1307",  "EIF1401",  "EIF1402",  "EIF1403",  "EIF1404",  "EIF1405",  "EIF1406",  "EIF1407",  "EIF1501",  "EIF1502",  "EIF1503",  "EIF1504",  "EIF1505",  "EIF1506",  "EIF1507",  "EIF1601",  "EIF1602",  "EIF1603",  "EIF1604",  "EIF1605",  "EIF1606",  "EIF1607",  "EIF1701",  "EIF1702",  "EIF1703",  "EIF1704",  "EIF1705",  "EIF1706",  "EIF1707",  "EIF1801",  "EIF1802",  "EIF1803",  "EIF1804",  "EIF1805",  "EIF1806",  "EIF1807",  "EIF1901",  "EIF1902",  "EIF1903",  "EIF1904",  "EIF1905",  "EIF1906",  "EIF1907",  "EIF2001",  "EIF2002",  "EIF2003",  "EIF2004",  "EIF2005",  "EIF2006",  "EIF2007",  "EIF2101",  "EIF2102",  "EIF2103",  "EIF2104",  "EIF2105",  "EIF2106",  "EIF2107",  "EIF2201",  "EIF2202",  "EIF2203",  "EIF2204",  "EIF2205",  "EIF2206",  "EIF2207",  "EIF2301",  "EIF2302",  "EIF2303",  "EIF2304",  "EIF2305",  "EIF2306",  "EIF2307",  "EIF2401",  "EIF2402",  "EIF2403",  "EIF2404",  "EIF2405",  "EIF2406",  "EIF2407",  "EIF2501",  "EIF2502",  "EIF2503",  "EIF2504",  "EIF2505",  "EIF2506",  "EIF2507",  "EIF2601",  "EIF2602",  "EIF2603",  "EIF2604",  "EIF2605",  "EIF2606",  "EIF2607",  "EIF2701",  "EIF2702",  "EIF2703",  "EIF2704",  "EIF2705",  "EIF2706",  "EIF2707",  "EIF2801",  "EIF2802",  "EIF2803",  "EIF2804",  "EIF2805",  "EIF2806",  "EIF2807",  "EIF2901",  "EIF2902",  "EIF2903",  "EIF2904",  "EIF2905",  "EIF2906",  "EIF2907",  "EIF3001",  "EIF3002",  "EIF3003",  "EIF3004",  "EIF3005",  "EIF3006",  "EIF3007",  "EIF3101",  "EIF3102",  "EIF3103",  "EIF3104",  "EIF3105",  "EIF3106",  "EIF3107",  "EIF3201",  "EIF3202",  "EIF3203",  "EIF3204",  "EIF3205",  "EIF3206",  "EIF3207",  "EIF3301",  "EIF3302",  "EIF3303",  "EIF3304",  "EIF3305",  "EIF3306",  "EIF3307",  "EIF3401",  "EIF3402",  "EIF3403",  "EIF3404",  "EIF3405",  "EIF3406",  "EIF3407",  "EIF3503",  "EIF6300",  "FIN0010",  "MAC9999",  "MACH001",  "MACH997",  "MOE0001",  "MOE0002",  "MOE0003",  "MOE0009",  "MOE0010",  "MOE0011",  "MOE0030",  "MOE0050",  "MOE0101",  "MOE0102",  "MOE0103",  "MOE0109",  "MOE9999",  "OUI\_\_\_\_",  "OUT0010",  "REC7400",  "REPRISE",  "VEH0001",  "VEH0002",  "VEH0003",  "VEH0004",  "VEH0005",  "VEH0006",  "VEH0007",  "VEH0008",  "VEH0009",  "VEH0010",  "VEH0011",  "VEH0012",  "VEH0048",  "VEH6300",  "VEH9991",  "VEN-74",  "VEN-INT"},  "Dépenses",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"AUX0199",  "AUX0299",  "AUX0699",  "VEH9998",  "VEH9999"},  "Recettes",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DEP9996"},  "Recettes Coûts Indirects",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DEP9997",  "DIV9999"},  "Recettes Frais Généraux",  "Autre"  ) |
| Dépenses/Recettes/CA Fexim | SWITCH(  TRUE,  ISBLANK('CONSOLIDE\_ObjetsDeCouts'[Code]),  "(Vide)",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DIV0401",  "DIV0402",  "DIV0405",  "VEN-2013",  "VEN-2014",  "VEN-2015",  "VEN-2016",  "VEN-2017",  "VEN-2018",  "VEN-2019",  "VEN-2020",  "VEN-2021",  "VEN-2022",  "VEN-2023",  "VEN-2024",  "VEN-2025",  "VEN7170"},  "Chiffre d'Affaires",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"9999999",  "AUX0100",  "AUX0200",  "AUX0600",  "AUX1101",  "AUX1103",  "AUX1104",  "AUX1105",  "AUX1106",  "AUX1107",  "DEP0101",  "DEP0102",  "DEP0103",  "DEP0104",  "DEP0105",  "DEP0106",  "DEP0107",  "DEP0108",  "DEP0198",  "DEP0199",  "DEP0201",  "DEP0202",  "DEP0203",  "DEP0299",  "DEP0301",  "DEP0302",  "DEP0303",  "DEP0304",  "DEP0305",  "DEP0306",  "DEP0307",  "DEP0308",  "DEP0309",  "DEP0310",  "DEP0311",  "DEP0312",  "DEP0313",  "DEP0314",  "DEP0399",  "DEP0401",  "DEP0402",  "DEP0403",  "DEP0499",  "DEP0501",  "DEP0502",  "DEP0503",  "DEP0504",  "DEP0505",  "DEP0506",  "DEP0599",  "DEP0601",  "DEP0602",  "DEP0603",  "DEP0604",  "DEP0699",  "DEP0701",  "DEP0702",  "DEP0703",  "DEP0704",  "DEP0705",  "DEP0708",  "DEP0710",  "DEP0799",  "DEP0801",  "DEP0802",  "DEP0803",  "DEP0804",  "DEP0805",  "DEP0806",  "DEP0807",  "DEP0808",  "DEP0809",  "DEP0810",  "DEP0811",  "DEP0812",  "DEP0813",  "DEP0814",  "DEP0899",  "DEP0901",  "DEP0902",  "DEP0903",  "DEP0998",  "DEP0999",  "DEP1001",  "DEP1002",  "DEP1003",  "DEP1004",  "DEP1005",  "DEP1006",  "DEP1007",  "DEP1008",  "DEP1009",  "DEP1010",  "DEP1013",  "DEP1014",  "DEP1015",  "DEP1016",  "DEP1099",  "DEP1201",  "DEP1202",  "DEP1203",  "DEP1301",  "DEP1302",  "DEP1303",  "DEP1304",  "DEP1305",  "DEP1306",  "DEP1307",  "DEP1308",  "DEP1309",  "DEP1310",  "DEP1311",  "DEP1312",  "DEP1313",  "DEP1314",  "DEP1315",  "DEP1316",  "DEP1317",  "DEP1318",  "DEP1319",  "DEP1320",  "DEP1321",  "DEP1322",  "DEP1323",  "DEP1324",  "DEP1325",  "DEP1326",  "DEP1327",  "DEP1328",  "DEP1329",  "DEP1330",  "DEP1331",  "DEP1332",  "DEP1333",  "DEP1334",  "DEP1335",  "DEP1336",  "DEP1337",  "DEP1399",  "DEP1401",  "DEP1402",  "DEP1403",  "DEP1404",  "DEP1405",  "DEP1406",  "DEP1407",  "DEP1408",  "DEP1409",  "DEP1410",  "DEP1411",  "DEP1412",  "DEP1413",  "DEP1414",  "DEP1415",  "DEP1416",  "DEP1417",  "DEP1499",  "DEP1501",  "DEP1502",  "DEP1503",  "DEP1504",  "DEP1505",  "DEP1599",  "DEP1601",  "DEP1602",  "DEP1603",  "DEP1604",  "DEP1605",  "DEP1606",  "DEP1607",  "DEP1608",  "DEP1609",  "DEP1610",  "DEP1611",  "DEP1612",  "DEP1613",  "DEP1614",  "DEP1615",  "DEP1616",  "DEP1617",  "DEP1618",  "DEP1619",  "DEP1620",  "DEP1621",  "DEP1622",  "DEP1623",  "DEP1624",  "DEP1625",  "DEP1626",  "DEP1627",  "DEP1628",  "DEP1629",  "DEP1630",  "DEP1631",  "DEP1632",  "DEP1633",  "DEP1634",  "DEP1635",  "DEP1636",  "DEP1699",  "DEP1701",  "DEP1702",  "DEP1703",  "DEP1704",  "DEP1705",  "DEP1706",  "DEP1707",  "DEP1708",  "DEP1709",  "DEP1710",  "DEP1711",  "DEP1801",  "DEP1802",  "DEP1803",  "DEP1804",  "DEP1805",  "DEP1806",  "DEP1807",  "DEP1901",  "DEP1902",  "DEP1903",  "DEP1904",  "DEP1905",  "DEP1906",  "DEP1907",  "DEP1908",  "DEP1909",  "DEP1910",  "DEP2001",  "DEP2002",  "DEP2003",  "DEP2004",  "DEP2005",  "DEP2006",  "DEP2007",  "DEP2008",  "DEP2009",  "DEP2010",  "DEP2011",  "DEP2012",  "DEP610000",  "DEP610200",  "DEP610250",  "DEP610900",  "DEP612000",  "DEP612010",  "DEP612020",  "DEP612030",  "DEP612170",  "DEP612310",  "DEP613000",  "DEP613100",  "DEP613400",  "DEP613410",  "DEP614000",  "DEP614002",  "DEP614100",  "DEP614104",  "DEP614200",  "DEP614201",  "DEP614300",  "DEP614301",  "DEP614310",  "DEP614312",  "DEP614313",  "DEP614600",  "DEP699994",  "DEP699995",  "DEP699996",  "DEP699997",  "DEP699998",  "DEP699999",  "DEP9000",  "DEP9998",  "DIV0010",  "DIV0101",  "DIV0201",  "DIV0301",  "DIV0403",  "DIV0404",  "DIV0501",  "DIV0601",  "DIV0701",  "DIV0801",  "DIV1013",  "DIV6300",  "DIV6310",  "DIV6370",  "DIV6700",  "EIF0026",  "EIF0001",  "EIF0027",  "EIF0002",  "EIF0003",  "EIF0031",  "EIF0004",  "EIF0032",  "EIF0005",  "EIF0033",  "EIF0006",  "EIF0034",  "EIF0007",  "EIF0035",  "EIF0008",  "EIF0036",  "EIF0009",  "EIF0037",  "EIF0010",  "EIF0038",  "EIF0011",  "EIF0039",  "EIF0012",  "EIF0040",  "EIF0013",  "EIF0041",  "EIF0014",  "EIF0042",  "EIF0015",  "EIF0043",  "EIF0016",  "EIF0044",  "EIF0017",  "EIF0045",  "EIF0018",  "EIF0046",  "EIF0019",  "EIF0047",  "EIF0020",  "EIF0048",  "EIF0021",  "EIF1201",  "EIF0022",  "EIF1202",  "EIF0023",  "EIF1203",  "EIF0024",  "EIF1204",  "EIF0025",  "EIF1205",  "EIF1206",  "EIF1207",  "EIF1301",  "EIF1302",  "EIF1303",  "EIF1304",  "EIF1305",  "EIF1306",  "EIF1307",  "EIF1401",  "EIF1402",  "EIF1403",  "EIF1404",  "EIF1405",  "EIF1406",  "EIF1407",  "EIF1501",  "EIF1502",  "EIF1503",  "EIF1504",  "EIF1505",  "EIF1506",  "EIF1507",  "EIF1601",  "EIF1602",  "EIF1603",  "EIF1604",  "EIF1605",  "EIF1606",  "EIF1607",  "EIF1701",  "EIF1702",  "EIF1703",  "EIF1704",  "EIF1705",  "EIF1706",  "EIF1707",  "EIF1801",  "EIF1802",  "EIF1803",  "EIF1804",  "EIF1805",  "EIF1806",  "EIF1807",  "EIF1901",  "EIF1902",  "EIF1903",  "EIF1904",  "EIF1905",  "EIF1906",  "EIF1907",  "EIF2001",  "EIF2002",  "EIF2003",  "EIF2004",  "EIF2005",  "EIF2006",  "EIF2007",  "EIF2101",  "EIF2102",  "EIF2103",  "EIF2104",  "EIF2105",  "EIF2106",  "EIF2107",  "EIF2201",  "EIF2202",  "EIF2203",  "EIF2204",  "EIF2205",  "EIF2206",  "EIF2207",  "EIF2301",  "EIF2302",  "EIF2303",  "EIF2304",  "EIF2305",  "EIF2306",  "EIF2307",  "EIF2401",  "EIF2402",  "EIF2403",  "EIF2404",  "EIF2405",  "EIF2406",  "EIF2407",  "EIF2501",  "EIF2502",  "EIF2503",  "EIF2504",  "EIF2505",  "EIF2506",  "EIF2507",  "EIF2601",  "EIF2602",  "EIF2603",  "EIF2604",  "EIF2605",  "EIF2606",  "EIF2607",  "EIF2701",  "EIF2702",  "EIF2703",  "EIF2704",  "EIF2705",  "EIF2706",  "EIF2707",  "EIF2801",  "EIF2802",  "EIF2803",  "EIF2804",  "EIF2805",  "EIF2806",  "EIF2807",  "EIF2901",  "EIF2902",  "EIF2903",  "EIF2904",  "EIF2905",  "EIF2906",  "EIF2907",  "EIF3001",  "EIF3002",  "EIF3003",  "EIF3004",  "EIF3005",  "EIF3006",  "EIF3007",  "EIF3101",  "EIF3102",  "EIF3103",  "EIF3104",  "EIF3105",  "EIF3106",  "EIF3107",  "EIF3201",  "EIF3202",  "EIF3203",  "EIF3204",  "EIF3205",  "EIF3206",  "EIF3207",  "EIF3301",  "EIF3302",  "EIF3303",  "EIF3304",  "EIF3305",  "EIF3306",  "EIF3307",  "EIF3401",  "EIF3402",  "EIF3403",  "EIF3404",  "EIF3405",  "EIF3406",  "EIF3407",  "EIF3503",  "EIF6300",  "FIN0010",  "MAC9999",  "MOE0009",  "MOE0010",  "MOE0011",  "MOE0030",  "MOE0050",  "MOE0101",  "MOE0102",  "MOE0103",  "MOE0109",  "MOE9999",  "OUI\_\_\_\_",  "OUT0010",  "REC7400",  "REPRISE",  "VEH0001",  "VEH0002",  "VEH0003",  "VEH0004",  "VEH0005",  "VEH0006",  "VEH0007",  "VEH0008",  "VEH0009",  "VEH0010",  "VEH0011",  "VEH0012",  "VEH0048",  "VEH6300",  "VEH9991",  "VEN-74",  "VEN-INT"},  "Dépenses",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"AUX0199",  "AUX0299",  "AUX0699",  "MOE0001",  "MOE0002",  "MOE0003",  "VEH9998",  "VEH9999"},  "Recettes",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DEP9996"},  "Recettes Couts Indirects",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DEP9997",  "DIV9999"},  "Recettes Frais Généraux",  "Autre"  ) |
| 201 MULTI SCT | SWITCH(  TRUE,  ISBLANK('CONSOLIDE\_ObjetsDeCouts'[Code]),  "(Vide)",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DEP1013",  "DEP1014",  "DEP1015",  "DEP1016",  "DEP1017",  "DEP1018",  "DEP3199",  "DEP9996",  "DEP9997",  "DEP9998",  "DIV0403",  "DIV1013"},  "DEP-FG",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DIV0401"},  "EN COURS",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DIV0402"},  "FAE",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"VEN-INT"},  "INT",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"VEN-2013",  "VEN-2014",  "VEN-2015",  "VEN-2016",  "VEN-2017",  "VEN-2018",  "VEN-2019",  "VEN-2020",  "VEN-2021",  "VEN-2022",  "VEN-2023",  "VEN-2024",  "VEN-2025",  "VEN7170"},  "REC",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"VEN-74"},  "VEN-74",  "DEP"  ) |
| 210 MULTI SCT | SWITCH(  TRUE,  ISBLANK('CONSOLIDE\_ObjetsDeCouts'[Code]),  "(Vide)",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"VEN-74"},  "74",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"9999999",  "AUX0100",  "AUX0199",  "AUX0200",  "AUX0299",  "AUX0600",  "AUX0699",  "AUX1101",  "AUX1103",  "AUX1104",  "AUX1105",  "AUX1106",  "AUX1107",  "DEP0101",  "DEP0102",  "DEP0103",  "DEP0104",  "DEP0105",  "DEP0106",  "DEP0107",  "DEP0108",  "DEP0198",  "DEP0199",  "DEP0201",  "DEP0202",  "DEP0203",  "DEP0299",  "DEP0300",  "DEP0301",  "DEP0302",  "DEP0303",  "DEP0304",  "DEP0305",  "DEP0306",  "DEP0307",  "DEP0308",  "DEP0309",  "DEP0310",  "DEP0311",  "DEP0312",  "DEP0313",  "DEP0314",  "DEP0399",  "DEP0400",  "DEP0401",  "DEP0402",  "DEP0403",  "DEP0450",  "DEP0499",  "DEP0500",  "DEP0501",  "DEP0502",  "DEP0503",  "DEP0504",  "DEP0505",  "DEP0506",  "DEP0599",  "DEP0600",  "DEP0601",  "DEP0602",  "DEP0603",  "DEP0604",  "DEP0699",  "DEP0700",  "DEP0701",  "DEP0702",  "DEP0703",  "DEP0704",  "DEP0705",  "DEP0708",  "DEP0710",  "DEP0799",  "DEP0800",  "DEP0801",  "DEP0802",  "DEP0803",  "DEP0804",  "DEP0805",  "DEP0806",  "DEP0807",  "DEP0808",  "DEP0809",  "DEP0810",  "DEP0811",  "DEP0812",  "DEP0813",  "DEP0814",  "DEP0899",  "DEP0900",  "DEP0901",  "DEP0902",  "DEP0903",  "DEP0998",  "DEP0999",  "DEP1000",  "DEP1001",  "DEP1002",  "DEP1003",  "DEP1004",  "DEP1005",  "DEP1006",  "DEP1007",  "DEP1008",  "DEP1009",  "DEP1010",  "DEP1099",  "DEP1100",  "DEP1101",  "DEP1102",  "DEP1103",  "DEP1104",  "DEP1105",  "DEP1106",  "DEP1107",  "DEP1200",  "DEP1201",  "DEP1202",  "DEP1203",  "DEP1300",  "DEP1301",  "DEP1302",  "DEP1303",  "DEP1304",  "DEP1305",  "DEP1306",  "DEP1307",  "DEP1308",  "DEP1309",  "DEP1310",  "DEP1311",  "DEP1312",  "DEP1313",  "DEP1314",  "DEP1315",  "DEP1316",  "DEP1317",  "DEP1318",  "DEP1319",  "DEP1320",  "DEP1321",  "DEP1322",  "DEP1323",  "DEP1324",  "DEP1325",  "DEP1326",  "DEP1327",  "DEP1328",  "DEP1329",  "DEP1330",  "DEP1331",  "DEP1332",  "DEP1333",  "DEP1334",  "DEP1335",  "DEP1336",  "DEP1337",  "DEP1399",  "DEP1400",  "DEP1401",  "DEP1402",  "DEP1403",  "DEP1404",  "DEP1405",  "DEP1406",  "DEP1407",  "DEP1408",  "DEP1409",  "DEP1410",  "DEP1411",  "DEP1412",  "DEP1413",  "DEP1414",  "DEP1415",  "DEP1416",  "DEP1417",  "DEP1499",  "DEP1500",  "DEP1501",  "DEP1502",  "DEP1503",  "DEP1504",  "DEP1505",  "DEP1599",  "DEP1600",  "DEP1601",  "DEP1602",  "DEP1603",  "DEP1604",  "DEP1605",  "DEP1606",  "DEP1607",  "DEP1608",  "DEP1609",  "DEP1610",  "DEP1611",  "DEP1612",  "DEP1613",  "DEP1614",  "DEP1615",  "DEP1616",  "DEP1617",  "DEP1618",  "DEP1619",  "DEP1620",  "DEP1621",  "DEP1622",  "DEP1623",  "DEP1624",  "DEP1625",  "DEP1626",  "DEP1627",  "DEP1628",  "DEP1629",  "DEP1630",  "DEP1631",  "DEP1632",  "DEP1633",  "DEP1634",  "DEP1635",  "DEP1636",  "DEP1699",  "DEP1700",  "DEP1701",  "DEP1702",  "DEP1703",  "DEP1704",  "DEP1705",  "DEP1706",  "DEP1707",  "DEP1708",  "DEP1709",  "DEP1710",  "DEP1711",  "DEP1800",  "DEP1801",  "DEP1802",  "DEP1803",  "DEP1804",  "DEP1805",  "DEP1806",  "DEP1807",  "DEP1900",  "DEP1901",  "DEP1902",  "DEP1903",  "DEP1904",  "DEP1905",  "DEP1906",  "DEP1907",  "DEP1908",  "DEP1909",  "DEP1910",  "DEP2000",  "DEP2001",  "DEP2002",  "DEP2003",  "DEP2004",  "DEP2005",  "DEP2006",  "DEP2007",  "DEP2008",  "DEP2009",  "DEP2010",  "DEP2011",  "DEP2012",  "DEP2100",  "DEP2200",  "DEP2300",  "DEP2400",  "DEP2500",  "DEP2600",  "DEP2700",  "DEP2800",  "DEP2850",  "DEP2900",  "DEP3000",  "DEP3100",  "DEP3101",  "DEP3102",  "DEP3103",  "DEP3104",  "DEP3105",  "DEP3106",  "DEP3107",  "DEP3199",  "DEP3200",  "DEP3201",  "DEP3202",  "DEP3203",  "DEP3299",  "DEP3301",  "DEP3302",  "DEP3303",  "DEP3304",  "DEP3305",  "DEP3306",  "DEP3399",  "DEP3401",  "DEP3402",  "DEP3403",  "DEP3404",  "DEP3405",  "DEP3406",  "DEP3407",  "DEP3408",  "DEP3499",  "DEP3500",  "DEP3501",  "DEP3502",  "DEP3503",  "DEP3504",  "DEP3505",  "DEP3599",  "DEP3601",  "DEP3602",  "DEP3603",  "DEP3604",  "DEP3605",  "DEP3606",  "DEP3607",  "DEP3608",  "DEP3609",  "DEP3610",  "DEP3611",  "DEP3699",  "DEP3701",  "DEP3702",  "DEP3703",  "DEP3704",  "DEP3705",  "DEP3706",  "DEP3707",  "DEP3708",  "DEP3709",  "DEP3710",  "DEP3799",  "DEP3801",  "DEP3802",  "DEP3803",  "DEP3804",  "DEP3805",  "DEP3806",  "DEP3807",  "DEP3808",  "DEP3899",  "DEP3901",  "DEP3902",  "DEP3903",  "DEP3904",  "DEP3905",  "DEP3906",  "DEP3907",  "DEP3999",  "DEP4000",  "DEP4001",  "DEP4002",  "DEP4099",  "DEP4100",  "DEP4101",  "DEP4102",  "DEP4103",  "DEP4199",  "DEP4201",  "DEP4202",  "DEP4203",  "DEP4299",  "DEP4301",  "DEP4302",  "DEP4303",  "DEP4399",  "DEP4401",  "DEP4402",  "DEP4403",  "DEP4499",  "DEP4501",  "DEP4502",  "DEP4503",  "DEP4504",  "DEP4505",  "DEP4506",  "DEP4507",  "DEP4508",  "DEP4509",  "DEP4599",  "DEP5000",  "DEP5001",  "DEP5500",  "DEP5501",  "DEP6000",  "DEP6090",  "DEP6100",  "DEP610000",  "DEP610200",  "DEP610250",  "DEP610900",  "DEP612000",  "DEP612010",  "DEP612020",  "DEP612030",  "DEP612170",  "DEP612310",  "DEP613000",  "DEP613100",  "DEP613400",  "DEP613410",  "DEP614000",  "DEP614002",  "DEP614100",  "DEP614104",  "DEP614200",  "DEP614201",  "DEP614300",  "DEP614301",  "DEP614310",  "DEP614312",  "DEP614313",  "DEP614600",  "DEP699994",  "DEP699995",  "DEP699996",  "DEP699997",  "DEP699998",  "DEP699999",  "DEP7000",  "DEP8000",  "DEP9000",  "DEP9100",  "DEP9999",  "DIV0010",  "DIV0020",  "DIV0030",  "DIV0035",  "DIV0040",  "DIV0050",  "DIV0055",  "DIV0060",  "DIV0070",  "DIV0080",  "DIV0101",  "DIV0102",  "DIV0201",  "DIV0301",  "DIV0402",  "DIV0403",  "DIV0404",  "DIV0501",  "DIV0601",  "DIV0701",  "DIV0801",  "DIV6300",  "DIV6310",  "DIV6370",  "DIV6410",  "DIV6430",  "DIV6600",  "DIV6700",  "DIV6900",  "DIV9999",  "EIF0001",  "EIF0002",  "EIF0003",  "EIF0004",  "EIF0005",  "EIF0006",  "EIF0007",  "EIF0008",  "EIF0009",  "EIF0010",  "EIF0011",  "EIF0012",  "EIF0013",  "EIF0014",  "EIF0015",  "EIF0016",  "EIF0017",  "EIF0018",  "EIF0019",  "EIF0020",  "EIF0021",  "EIF0022",  "EIF0023",  "EIF0024",  "EIF0025",  "EIF0026",  "EIF0027",  "EIF0030",  "EIF0031",  "EIF0032",  "EIF0033",  "EIF0034",  "EIF0035",  "EIF0036",  "EIF0037",  "EIF0038",  "EIF0039",  "EIF0040",  "EIF0041",  "EIF0042",  "EIF0043",  "EIF0044",  "EIF0045",  "EIF0046",  "EIF0047",  "EIF0048",  "EIF1201",  "EIF1202",  "EIF1203",  "EIF1204",  "EIF1205",  "EIF1206",  "EIF1207",  "EIF1301",  "EIF1302",  "EIF1303",  "EIF1304",  "EIF1305",  "EIF1306",  "EIF1307",  "EIF1401",  "EIF1402",  "EIF1403",  "EIF1404",  "EIF1405",  "EIF1406",  "EIF1407",  "EIF1501",  "EIF1502",  "EIF1503",  "EIF1504",  "EIF1505",  "EIF1506",  "EIF1507",  "EIF1601",  "EIF1602",  "EIF1603",  "EIF1604",  "EIF1605",  "EIF1606",  "EIF1607",  "EIF1701",  "EIF1702",  "EIF1703",  "EIF1704",  "EIF1705",  "EIF1706",  "EIF1707",  "EIF1801",  "EIF1802",  "EIF1803",  "EIF1804",  "EIF1805",  "EIF1806",  "EIF1807",  "EIF1901",  "EIF1902",  "EIF1903",  "EIF1904",  "EIF1905",  "EIF1906",  "EIF1907",  "EIF2001",  "EIF2002",  "EIF2003",  "EIF2004",  "EIF2005",  "EIF2006",  "EIF2007",  "EIF2101",  "EIF2102",  "EIF2103",  "EIF2104",  "EIF2105",  "EIF2106",  "EIF2107",  "EIF2201",  "EIF2202",  "EIF2203",  "EIF2204",  "EIF2205",  "EIF2206",  "EIF2207",  "EIF2301",  "EIF2302",  "EIF2303",  "EIF2304",  "EIF2305",  "EIF2306",  "EIF2307",  "EIF2401",  "EIF2402",  "EIF2403",  "EIF2404",  "EIF2405",  "EIF2406",  "EIF2407",  "EIF2501",  "EIF2502",  "EIF2503",  "EIF2504",  "EIF2505",  "EIF2506",  "EIF2507",  "EIF2601",  "EIF2602",  "EIF2603",  "EIF2604",  "EIF2605",  "EIF2606",  "EIF2607",  "EIF2701",  "EIF2702",  "EIF2703",  "EIF2704",  "EIF2705",  "EIF2706",  "EIF2707",  "EIF2801",  "EIF2802",  "EIF2803",  "EIF2804",  "EIF2805",  "EIF2806",  "EIF2807",  "EIF2901",  "EIF2902",  "EIF2903",  "EIF2904",  "EIF2905",  "EIF2906",  "EIF2907",  "EIF3001",  "EIF3002",  "EIF3003",  "EIF3004",  "EIF3005",  "EIF3006",  "EIF3007",  "EIF3101",  "EIF3102",  "EIF3103",  "EIF3104",  "EIF3105",  "EIF3106",  "EIF3107",  "EIF3201",  "EIF3202",  "EIF3203",  "EIF3204",  "EIF3205",  "EIF3206",  "EIF3207",  "EIF3301",  "EIF3302",  "EIF3303",  "EIF3304",  "EIF3305",  "EIF3306",  "EIF3307",  "EIF3401",  "EIF3402",  "EIF3403",  "EIF3404",  "EIF3405",  "EIF3406",  "EIF3407",  "EIF3503",  "EIF6300",  "EIF6310",  "FIN0010",  "MAC0100",  "MAC0110",  "MAC0120",  "MAC0200",  "MAC0300",  "MAC0310",  "MAC0320",  "MAC0330",  "MAC0340",  "MAC0350",  "MAC0360",  "MAC0370",  "MAC0400",  "MAC0410",  "MAC0420",  "MAC0430",  "MAC0440",  "MAC0499",  "MAC0500",  "MAC0600",  "MAC0700",  "MAC0710",  "MAC9999",  "MACH001",  "MACH995",  "MACH996",  "MACH997",  "MACH998",  "MOB0000",  "MOB0010",  "MOB9999",  "MOE0001",  "MOE0002",  "MOE0003",  "MOE0009",  "MOE0010",  "MOE0011",  "MOE0030",  "MOE0050",  "MOE0060",  "MOE0101",  "MOE0102",  "MOE0103",  "MOE0109",  "MOE0110",  "MOE0130",  "MOE0199",  "MOE0299",  "MOE0699",  "MOE9999",  "OUI\_\_\_\_",  "OUT0010",  "OUT0040",  "REC-010",  "REC7400",  "REC7410",  "REC7500",  "REC7600",  "REC7700",  "REC7900",  "REPRISE",  "VEH0001",  "VEH0002",  "VEH0003",  "VEH0004",  "VEH0005",  "VEH0006",  "VEH0007",  "VEH0008",  "VEH0009",  "VEH0010",  "VEH0011",  "VEH0012",  "VEH0013",  "VEH0014",  "VEH0048",  "VEH6300",  "VEH9991",  "VEH9998",  "VEH9999"},  "DEP",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DIV0401",  "DIV0405"},  "ENC",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"DEP1013",  "DEP1014",  "DEP1015",  "DEP1016",  "DEP9996",  "DEP9997",  "DEP9998",  "DIV1013"},  "FG",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"VEN-INT"},  "INT",  'CONSOLIDE\_ObjetsDeCouts'[Code] IN {"VEN-2013",  "VEN-2014",  "VEN-2015",  "VEN-2016",  "VEN-2017",  "VEN-2018",  "VEN-2019",  "VEN-2020",  "VEN-2021",  "VEN-2022",  "VEN-2023",  "VEN-2024",  "VEN-2025",  "VEN7170"},  "REC",  "Autre"  ) |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_ObjetsDeCouts | import | m |  |

# Table: Dates

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Date |  |
| DateAsInteger |  |
| Year |  |
| Monthnumber |  |
| YearMonthnumber |  |
| YearMonthShort |  |
| MonthNameShort |  |
| MonthNameLong |  |
| DayOfWeekNumber |  |
| DayOfWeek |  |
| DayOfWeekShort |  |
| Quarter |  |
| YearQuarter |  |
| Mois-A |  |
| DateKey |  |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| Dates | import | calculated | ['ADDCOLUMNS (', 'CALENDAR (DATE(2014,1,1), today()+365),', '"DateAsInteger", FORMAT ( [Date], "YYYYMMDD" ),', '"Year", YEAR ( [Date] ),', '"Monthnumber", FORMAT ( [Date], "MM" ),', '"YearMonthnumber", FORMAT ( [Date], "YYYY/MM" ),', '"YearMonthShort", FORMAT ( [Date], "YYYY/mmm" ),', '"MonthNameShort", FORMAT ( [Date], "mmm" ),', '"MonthNameLong",FORMAT ( [Date], "mmmm" ),', '"DayOfWeekNumber", WEEKDAY ( [Date] ),', '"DayOfWeek", FORMAT ( [Date], "dddd" ),', '"DayOfWeekShort", FORMAT ( [Date], "ddd" ),', '"Quarter", "Q" & FORMAT ( [Date], "Q" ),', '"YearQuarter", FORMAT ( [Date], "YYYY" ) & "/Q" & FORMAT ( [Date], "Q" ),', '"Mois-A", FORMAT([Date], "MMM YY"),', '"DateKey", FORMAT([Date], "YYYY-MM-DD"))'] |

# Table: 00Mesures

Mesures:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| CONSOLIDE\_201\_TRESO | calculate(sum('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Amount (LCY)])) |
| CONSOLIDE\_201\_TRESO ECHU | calculate(sum('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Amount (LCY)]), 'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Echue] = 1) |
| CONSOLIDE208\_Reel | -sum(CONSOLIDE\_EcrituresCouts\_2022[Amount]) |
| CONSOLIDEBudget | CALCULATE(sum(CONSOLIDE\_EcrituresCoutsBudget2022[Amount]), CONSOLIDE\_EcrituresCoutsBudget2022[Budget Name] = "BUDGET") |
| CONSOLIDE\_Previsions | CALCULATE(sum(CONSOLIDE\_EcrituresCoutsBudget2022[Amount]), CONSOLIDE\_EcrituresCoutsBudget2022[Budget Name] = "PREVI") |
| CONSOLIDE\_201\_BudgetFAC | CALCULATE(-[CONSOLIDEBudget], CONSOLIDE\_EcrituresCoutsBudget2022[Cost Object Code (groupes)] = "FAC") |
| CONSOLIDE\_201\_%\_Margenettebudget | DIVIDE([CONSOLIDE\_201\_ResultatBudget],[CONSOLIDE\_201\_BudgetFAC]) |
| CONSOLIDE\_201\_%\_Margenetteresultat | DIVIDE([CONSOLIDE\_201\_ResultatNet (prv)],[CONSOLIDE\_201\_PrevisionsFAC]) |
| CONSOLIDE\_201\_BudgetPrixR | CALCULATE([CONSOLIDEBudget], CONSOLIDE\_EcrituresCoutsBudget2022[Cost Object Code (groupes)] = "PR") |
| CONSOLIDE\_201\_Controle | [CONSOLIDE\_201\_EcartFacturation] - [CONSOLIDE\_201\_RecettesENCOURS] - [CONSOLIDE208\_Reel] |
| CONSOLIDE\_201\_CoutsDEP | CALCULATE(-[CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "DEP")) |
| CONSOLIDE\_201\_CoutsDEP-FG | CALCULATE(-[CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "DEP-FG")) |
| CONSOLIDE\_201\_CoutsVEN-74 | CALCULATE(-[CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "VEN-74")) |
| CONSOLIDE\_201\_Depenses | [CONSOLIDE\_201\_CoutsDEP-FG]+[CONSOLIDE\_201\_CoutsDEP]+[CONSOLIDE\_201\_CoutsVEN-74] |
| CONSOLIDE\_201\_EcartFacturation | [CONSOLIDE\_201\_Realiseintroduit] - [CONSOLIDE\_201\_Depenses] |
| CONSOLIDE\_201\_EcartFacturation M-1 | VAR MinDate = min(Dates[Date])  VAR MaxDate = edate(max(Dates[Date]), -1)  RETURN  calculate([CONSOLIDE\_201\_EcartFacturation], CONSOLIDE\_EcrituresCouts\_2022[Posting Date] <= MaxDate  ) |
| CONSOLIDE\_201\_Facturation | [CONSOLIDE\_201\_RecettesINT] + [CONSOLIDE\_201\_RecettesREC] |
| CONSOLIDE\_201\_PrevisionsFAC | CALCULATE(-[CONSOLIDE\_Previsions], CONSOLIDE\_EcrituresCoutsBudget2022[Cost Object Code (groupes)] = "FAC") |
| CONSOLIDE\_201\_PrevisionsPrixR | CALCULATE([CONSOLIDE\_Previsions], CONSOLIDE\_EcrituresCoutsBudget2022[Cost Object Code (groupes)] = "PR") |
| CONSOLIDE\_201\_RecettesENCOURS | CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "EN COURS"))  // a filtrer sur + et - |
| CONSOLIDE\_201\_RecettesINT | CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "INT")) |
| CONSOLIDE\_201\_RecettesREC | CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "REC")) |
| CONSOLIDE\_201\_ResultatBudget | [CONSOLIDE\_201\_BudgetFAC] - [CONSOLIDE\_201\_BudgetPrixR] |
| CONSOLIDE\_201\_ResultatNet (prv) | [CONSOLIDE\_201\_PrevisionsFAC] - [CONSOLIDE\_201\_PrevisionsPrixR] |
| CONSOLIDE\_201\_SAG\_Fact | [CONSOLIDE\_201\_PrevisionsFAC]-[CONSOLIDE\_201\_Realiseintroduit]- [CONSOLIDE\_201\_RecettesENCOURS] |
| CONSOLIDE\_201\_SAGCOUT | [CONSOLIDE\_201\_PrevisionsPrixR] - [CONSOLIDE\_201\_Depenses] |
| CONSOLIDE\_210\_74 | CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[210 MULTI SCT] = "74")) |
| CONSOLIDE\_210\_Autres | CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[210 MULTI SCT] = "Autres")) |
| CONSOLIDE\_210\_CONTROLE | [CONSOLIDE\_210\_MARGENETTE]  // SOMME AMOUNT |
| CONSOLIDE\_210\_CA | CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[210 MULTI SCT] = "REC")) |
| CONSOLIDE\_210\_DEP | CALCULATE(-[CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[210 MULTI SCT] = "DEP")) |
| CONSOLIDE\_210\_ENCOURS | CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[210 MULTI SCT] = "ENC")) |
| CONSOLIDE\_210\_INT | CALCULATE(-[CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[210 MULTI SCT] = "INT")) |
| CONSOLIDE\_210\_FG | CALCULATE(-[CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[210 MULTI SCT] = "FG")) |
| CONSOLIDE\_210\_MARGENETTE | [CONSOLIDE\_210\_TOTALVENTES]-[CONSOLIDE\_210\_TOTALCOUT] |
| CONSOLIDE\_210\_TOTALCOUT | [CONSOLIDE\_210\_TOTALCOUTBRUT]+[CONSOLIDE\_210\_FG] |
| CONSOLIDE\_210\_TOTALCOUTBRUT | [CONSOLIDE\_210\_DEP]+[CONSOLIDE\_210\_74]+[CONSOLIDE\_210\_INT] |
| CONSOLIDE\_210\_TOTALVENTES | '00Mesures'[CONSOLIDE\_210\_CA]+'00Mesures'[CONSOLIDE\_210\_ENCOURS] |
| CONSOLIDE\_Previ\_ResultatNET\_Date Param 1 | VAR MinDate= min(Dates[Date])  VAR MaxDate = date(selectedvalue('Date calages Année'[AnneeSaisie]), selectedvalue('Date calages Mois'[Mois]), day(max(Dates[Date])))    Return  calculate( [CONSOLIDE\_201\_ResultatNet (prv)], all(Dates[Date]),CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie] >= MinDate, CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie] <= MaxDate) |
| CONSOLIDE\_Previ\_ResultatNET\_Date Param 2 | VAR MinDate= min(Dates[Date])  VAR MaxDate = date(selectedvalue('Date calages Année'[AnneeSaisie]), selectedvalue('Date calages Mois'[Mois]), day(max(Dates[Date])))  VAR MaxDate2 = edate(MaxDate, -3)    Return  calculate( [CONSOLIDE\_201\_ResultatNet (prv)], all(Dates[Date]),CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie] >= MinDate, CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie] <= MaxDate2) |
| CONSOLIDE\_201\_RecettesFAE | CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "FAE"))  //chiffres à valider en mars  //bleu |
| CONSOLIDE\_201\_Realiseintroduit | [CONSOLIDE\_201\_Facturation]+ [CONSOLIDE\_201\_RecettesFAE]  //bleu  //les trois colonnes restent ensemble  // somme en gras (valeurs)  //titre = réalisé cumul |
| CONSOLIDE\_201\_VariationPFAFact | [CONSOLIDE\_201\_PrevisionsFAC]-[CONSOLIDE\_Previ\_Fact\_Date Param 1]  //bleu |
| CONSOLIDE\_201\_VariationPFAmarge nette | [CONSOLIDE\_201\_ResultatNet (prv)] -[CONSOLIDE\_Previ\_ResultatNET\_Date Param 1]  //jaune |
| CONSOLIDE\_201\_RecettesPCA | if(sumx(CONSOLIDE\_CentreDeCouts, CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "EN COURS")))>0 , 0,sumx(CONSOLIDE\_CentreDeCouts, CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "EN COURS") )))  // a filtrer sur + et - )) |
| CONSOLIDE\_201\_RecettesENCOURS2 | if(sumx(CONSOLIDE\_CentreDeCouts, CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "EN COURS")))<0 , 0,sumx(CONSOLIDE\_CentreDeCouts, CALCULATE([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "EN COURS") )))  // a filtrer sur + et - )))) |
| CONSOLIDE\_201\_SumXPCA | if(HASONEVALUE(CONSOLIDE\_CentreDeCouts[Code]), [CONSOLIDE\_201\_RecettesPCA],  sumx(CONSOLIDE\_CentreDeCouts, [CONSOLIDE\_201\_RecettesPCA]))  //blanc |
| CONSOLIDE\_201\_SumX EnCours | if(HASONEVALUE(CONSOLIDE\_CentreDeCouts[Code]), [CONSOLIDE\_201\_RecettesENCOURS2],  sumx(CONSOLIDE\_CentreDeCouts, [CONSOLIDE\_201\_RecettesENCOURS2]))  //blanc |
| CONSOLIDE\_201\_MargeNetteAvecEncours% | divide([CONSOLIDE\_201\_RealiseMargeNette], [CONSOLIDE\_201\_ChiffresAffaires], 0)  //jaune |
| CONSOLIDE\_201\_CA YTD | VAR MinDate = min('Date2 YTD'[Date])  VAR MaxDate = max('Date2 YTD'[Date])  RETURN  calculate([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[Dépenses/Recettes/CA Collignon] = "Chiffre d'Affaires"),  //CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "INT" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "REC" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "FAE" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "EN COURS"),  filter(CONSOLIDE\_EcrituresCouts\_2022, CONSOLIDE\_EcrituresCouts\_2022[Posting Date] <= MaxDate && CONSOLIDE\_EcrituresCouts\_2022[Posting Date] >= MinDate ))  //bleu |
| CONSOLIDE\_201\_MargeNetteYTD | VAR MinDate = min('Date2 YTD'[Date])  VAR MaxDate = max('Date2 YTD'[Date])  RETURN  CALCULATE([CONSOLIDE208\_Reel], CONSOLIDE\_EcrituresCouts\_2022[Posting Date] >= MinDate && CONSOLIDE\_EcrituresCouts\_2022[Posting Date] <=MaxDate)  //jaune  //la date doit être indépendante de celle sélectionnée par l'utilisateur dans le rapport, c'est un filtre au-dessus du tableau |
| CONSOLIDE\_201\_MargeNette YTD % | divide([CONSOLIDE\_201\_MargeNetteYTD], [CONSOLIDE\_201\_CA YTD],0)  //jaune |
| CONSOLIDE\_201\_RealiseMargeNette | [CONSOLIDE\_201\_RecettesENCOURS] + [CONSOLIDE\_201\_EcartFacturation]  //jaune |
| CONSOLIDE\_201\_ChiffresAffaires | [CONSOLIDE\_201\_Realiseintroduit] + [CONSOLIDE\_201\_RecettesENCOURS] |
| CONSOLIDE\_201\_Carnet2026 | VAR SelectedDate = Max(Dates[Date])  return  Calculate(sum('Carnet de commande'[2026]), filter('Carnet de commande', 'Carnet de commande'[Date] = SelectedDate))  //bleu |
| CONSOLIDE\_201\_Carnet2027 | VAR SelectedDate = Max(Dates[Date])  return  Calculate(sum('Carnet de commande'[2027 et +]), filter('Carnet de commande', 'Carnet de commande'[Date] = SelectedDate))  //bleu |
| CONSOLIDE\_201\_Carnet2025 | [CONSOLIDE\_201\_PrevisionsFAC] - [CA Test] - [CONSOLIDE\_201\_Carnet2026] - [CONSOLIDE\_201\_Carnet2027]  // |
| CONSOLIDE\_201\_CALAGE 1% | [CONSOLIDE\_Previ\_ResultatNET\_Date Param 1]/[CONSOLIDE\_Previ\_Fact\_Date Param 1] |
| CONSOLIDE\_Previ\_Fact\_Date Param 1 | VAR MinDate= min(Dates[Date])  VAR MaxDate = date(selectedvalue('Date calages Année'[AnneeSaisie]), selectedvalue('Date calages Mois'[Mois]), day(max(Dates[Date])))    Return  calculate( [CONSOLIDE\_201\_PrevisionsFAC], all(Dates[Date]),CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie] >= MinDate, CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie] <= MaxDate) |
| CONSOLIDE\_Previ\_Fact \_Date Param 2 | VAR MinDate= min(Dates[Date])  VAR MaxDate = date(selectedvalue('Date calages Année'[AnneeSaisie]), selectedvalue('Date calages Mois'[Mois]), day(max(Dates[Date])))  VAR MaxDate2 = edate(MaxDate, -3)    Return  calculate( [CONSOLIDE\_201\_PrevisionsFAC], all(Dates[Date]),CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie] >= MinDate, CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie] <= MaxDate2) |
| CONSOLIDE\_201\_Calage2% bis | [CONSOLIDE\_Previ\_ResultatNET\_Date Param 2]/[CONSOLIDE\_Previ\_Fact \_Date Param 2] |
| CONSOLIDE\_TRESO\_NonEchu | CALCULATE(  SUM('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Amount (LCY)]),  FILTER(  'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry','CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Echue]= 0  )  ) |
| CONSOLIDE\_TRESO\_Echu | CALCULATE(  SUM('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Amount (LCY)]),  FILTER(  'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry','CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Echue]= 1  )  ) |
| CONSOLIDE\_201\_ResteAFairePV | [CONSOLIDE\_201\_RecettesENCOURS] - [CONSOLIDE\_201\_Realiseintroduit] - [CONSOLIDE\_201\_VariationPFAFact]  //bleu |
| CONSOLIDE\_201\_MargeNetteCarnet\_A+1 | if(HASONEVALUE(CONSOLIDE\_CentreDeCouts[CostCenterKEYCompany]), divide([CONSOLIDE\_201\_ResultatNet (prv)], [CONSOLIDE\_201\_PrevisionsFAC],0)\*[CONSOLIDE\_201\_Carnet2026], sumx(CONSOLIDE\_CentreDeCouts, divide([CONSOLIDE\_201\_ResultatNet (prv)], [CONSOLIDE\_201\_PrevisionsFAC],0)\* [CONSOLIDE\_201\_Carnet2026])) |
| CONSOLIDE\_201\_MargeNetteCarnet\_A+2 | if(HASONEVALUE(CONSOLIDE\_CentreDeCouts[CostCenterKEYCompany]), divide([CONSOLIDE\_201\_ResultatNet (prv)], [CONSOLIDE\_201\_PrevisionsFAC],0)\*[CONSOLIDE\_201\_Carnet2027], sumx(CONSOLIDE\_CentreDeCouts, divide([CONSOLIDE\_201\_ResultatNet (prv)], [CONSOLIDE\_201\_PrevisionsFAC],0)\* [CONSOLIDE\_201\_Carnet2027])) |
| CONSOLIDE\_201\_MargeNetteAnneeEnCours | [CONSOLIDE\_201\_ResultatNet (prv)] - [CONSOLIDE\_201\_RealiseMargeNette] - [CONSOLIDE\_201\_MargeNetteCarnet\_A+1] - [CONSOLIDE\_201\_MargeNetteCarnet\_A+2] |
| CONSOLIDE\_201\_% des coûts | if(and(hasonefilter(CONSOLIDE\_CentreDeCouts[CostCenterKEYCompany]), [CONSOLIDE\_201\_PrevisionsPrixR] = 0), 1,  if(divide([CONSOLIDE\_201\_Depenses],[CONSOLIDE\_201\_PrevisionsPrixR], 0) >=1, 1,  divide([CONSOLIDE\_201\_Depenses], [CONSOLIDE\_201\_PrevisionsPrixR], 0))) |
| Prévisions Prix R | calculate([CONSOLIDE\_201\_PrevisionsPrixR], filter(CONSOLIDE\_CentreDeCouts, CONSOLIDE\_CentreDeCouts[Nom rubrique] = "Chantiers")) |
| CONSOLIDE\_201\_MontantAvantFG | [CONSOLIDE\_201\_% des coûts] \* [CONSOLIDE\_201\_ResultatNet (prv)] |
| CONSOLIDE\_201\_MontantAprèsFG | if(hasonevalue(CONSOLIDE\_CentreDeCouts[CostCenterKEYCompany]),  ([CONSOLIDE\_201\_MontantAvantFG] - [CONSOLIDE\_201\_EcartFacturation])\* (1+SELECTEDVALUE(CONSOLIDE\_JOB[Taux\_à\_appliquer])),  sumx(CONSOLIDE\_CentreDeCouts, ([CONSOLIDE\_201\_MontantAvantFG] - [CONSOLIDE\_201\_EcartFacturation]) \* (1+SELECTEDVALUE(CONSOLIDE\_JOB[Taux\_à\_appliquer])))) |
| test | edate(max(Dates[Date]), -1) |
| CONSOLIDE\_404 | calculate(-sum(CONSOLIDE\_EcrituresCoutsBudget2022[Amount]), filter(CONSOLIDE\_EcrituresCoutsBudget2022, CONSOLIDE\_EcrituresCoutsBudget2022[Cost Object Code] = "VEN-2015" || [Cost Object Code] = "VEN-INT" || [Cost Object Code] = "VEN-2014"), filter( CONSOLIDE\_EcrituresCoutsBudget2022, CONSOLIDE\_EcrituresCoutsBudget2022[Budget Name] = "BUDGET")) |
| CONSOLIDE\_404\_Dont Interne | calculate(-sum(CONSOLIDE\_EcrituresCoutsBudget2022[Amount]), filter(CONSOLIDE\_EcrituresCoutsBudget2022, CONSOLIDE\_EcrituresCoutsBudget2022[Cost Object Code] = "VEN-2015" || [Cost Object Code] = "VEN-INT" || [Cost Object Code] = "VEN-2014"), filter( CONSOLIDE\_EcrituresCoutsBudget2022, CONSOLIDE\_EcrituresCoutsBudget2022[Budget Name] = "BUDGET"), filter( CONSOLIDE\_JOB, CONSOLIDE\_JOB[Interne/Externe]="Interne")) |
| Reporting\_FinEstimée | BLANK() |
| Reporting\_%\_Avancement | DIVIDE([CONSOLIDE\_201\_ChiffresAffaires], [CONSOLIDE\_201\_PrevisionsFAC]) |
| Reporting\_Montant\_encaissé | [CONSOLIDE\_201\_Realiseintroduit]-[CONSOLIDE\_201\_TRESO] |
| Reporting\_Montant\_Facturé (%) | DIVIDE([CONSOLIDE\_201\_Realiseintroduit],[CONSOLIDE\_201\_ChiffresAffaires]) |
| Reposting\_Montant\_encaissé (%) | DIVIDE([Reporting\_Montant\_encaissé],[CONSOLIDE\_201\_Realiseintroduit]) |
| Reporting\_CA\_cumule Ex ANT | VAR SelectedDate = max('Date2 YTD'[Date])  VAR MinDate = min(Dates[Date])  VAR MaxDate = date(year(SelectedDate) -2, 12, 31)  RETURN  calculate([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "INT" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "REC" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "FAE" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "EN COURS"), filter(CONSOLIDE\_EcrituresCouts\_2022, CONSOLIDE\_EcrituresCouts\_2022[Posting Date] <= MaxDate && CONSOLIDE\_EcrituresCouts\_2022[Posting Date] >= MinDate ))  //bleu  // CA depuis de début jusqu'à 1 an avant la date sélectionnée par l'utilisateur |
| Reporting\_Marge\_cumule Ex ANT | VAR SelectedDate = max('Date2 YTD'[Date])  VAR MinDate = min(Dates[Date])  VAR MaxDate = date(year(SelectedDate) -2, 12, 31)  RETURN  CALCULATE([CONSOLIDE208\_Reel], CONSOLIDE\_EcrituresCouts\_2022[Posting Date] >= MinDate && CONSOLIDE\_EcrituresCouts\_2022[Posting Date] <=MaxDate)  //jaune  // Marge depuis le début jusqu'à 1 an avant |
| Reporting\_CA\_cumule Ex N-1 | VAR SelectedMaxDate = max('Date2 YTD'[Date])  VAR MinDate = date(year(SelectedMaxDate) -1, 01, 01)  VAR MaxDate = date(year(SelectedMaxDate) -1, 12,31)  RETURN  calculate([CONSOLIDE208\_Reel], filter(CONSOLIDE\_ObjetsDeCouts, CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "INT" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "REC" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "FAE" || CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT] = "EN COURS"), filter(CONSOLIDE\_EcrituresCouts\_2022, CONSOLIDE\_EcrituresCouts\_2022[Posting Date] <= MaxDate && CONSOLIDE\_EcrituresCouts\_2022[Posting Date] >= MinDate ))  //bleu  // CA depuis de toute l'année précédente |
| Reporting\_Marge\_cumule Ex N-1 | VAR SelectedMaxDate = max('Date2 YTD'[Date])  VAR MinDate = date(year(SelectedMaxDate) -1, 01, 01)  VAR MaxDate = date(year(SelectedMaxDate) -1, 12,31)  RETURN  CALCULATE([CONSOLIDE208\_Reel], CONSOLIDE\_EcrituresCouts\_2022[Posting Date] >= MinDate && CONSOLIDE\_EcrituresCouts\_2022[Posting Date] <=MaxDate)  //jaune  // Réalisé N-1 |
| REPORTING\_Reste\_A\_Faire\_CA | [CONSOLIDE\_201\_PrevisionsFAC]-[CONSOLIDE\_201\_ChiffresAffaires] |
| REPORTING\_Reste\_A\_Faire\_Marge | [CONSOLIDE\_201\_ResultatNet (prv)]-[CONSOLIDE\_201\_RealiseMargeNette] |
| DateSaisieTest | if(hasonevalue(CONSOLIDE\_CentreDeCouts[No+Name]), calculate(min(CONSOLIDE\_EcrituresCoutsBudget2022[Date de saisie]), CONSOLIDE\_EcrituresCoutsBudget2022[Budget Name] = "BUDGET"), blank()) |
| Look&field date | "la société sélectionnée est : " & SELECTEDVALUE(CONSOLIDE\_JOB[SCTE]) |
| GRAPHIQUE\_TRESO\_Echu | CALCULATE(  SUM('TRESO Excel'[Amount (LCY)]),  FILTER(  'TRESO Excel','TRESO Excel'[Echue]= 1  )  ) |
| GRAPHIQUE\_TRESO\_NonEchu | CALCULATE(  SUM('TRESO Excel'[Amount (LCY)]),  FILTER(  'TRESO Excel','TRESO Excel'[Echue]= 0  )  ) |
| VIDE2 | blank()  //if(hasonevalue(CONSOLIDE\_CentreDeCouts[No+Name]), blank(), blank()) |
| CA Test | CALCULATE(-sum(CONSOLIDE\_EcrituresCouts\_2022[Amount]), CONSOLIDE\_ObjetsDeCouts[Dépenses/Recettes/CA Collignon] = "Chiffre d'Affaires")  // pas en cumulé mais par mois |
| Trésorerie Échue Mensuelle | CALCULATE(  -SUM('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Amount (LCY)]),  FILTER(  ALL('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'),  MONTH('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date]) = MONTH(MAX('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date])) &&  YEAR('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date]) = YEAR(MAX('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date])) &&  'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Initial Entry Due Date] <= EOMONTH(MAX('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date]), 0)  //&&  //'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Document Type] = "0" || 'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Document Type] = "2" || 'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Document Type] = "3")  )) |
| Trésorerie non-échue mensuelle | CALCULATE(  SUM('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Amount (LCY)]),  FILTER(  ALL('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'),  MONTH('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date]) = MONTH(MAX('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date])) &&  YEAR('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date]) = YEAR(MAX('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date])) &&  'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Initial Entry Due Date] > EOMONTH(MAX('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date]), 0)  //&&  //'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Document Type] = "0" || 'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Document Type] = "2" || 'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Document Type] = "3")  )) |
| CONSOLIDE\_Treso\_JGS | VAR MinDate = min('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date])  VAR MaxDate = max('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date])  RETURN  CALCULATE(SUM('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Amount (LCY)]),filter('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry','CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date]<= MaxDate && 'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Posting Date]>=MinDate))  //filter(CONSOLIDE\_EcrituresCouts\_2022, CONSOLIDE\_EcrituresCouts\_2022[Posting Date] <= MaxDate && CONSOLIDE\_EcrituresCouts\_2022[Posting Date] >= MinDate )) |
| CONSOLIDE208\_ReelCumulé | TOTALYTD([CONSOLIDE208\_Reel], Dates[Date]) |
| CA cumulé | TOTALYTD([CA Test], Dates[Date]) |
| CONSOLIDE\_201\_SumXPCA\_Graphique | calculate([CONSOLIDE\_201\_SumXPCA], filter(ALL(Dates), Dates[Date] <= max(Dates[Date]))) |
| CONSOLIDE\_201\_SumXEncours\_Graphique | calculate([CONSOLIDE\_201\_SumX EnCours], filter(ALL(Dates), Dates[Date] <= max(Dates[Date]))) |
| COLLIGNONBudget Excel | SUM('BUDGET Excel'[AMOUNT]) |
| COLLIGNON\_BudgetCumulMois | Var Mois = month(max(Dates[Date]))  Var Multiplicateur = divide(Mois, 12)  var BudgetCumulMois = calculate([COLLIGNONBudget Excel] \* Multiplicateur, 'BUDGET Excel'[Année] = year(max(Dates[Date])))  return  BudgetCumulMois |
| Montant ouvert | sum('CONSOLIDE\_Cust Ledger Entry'[Amount]) |
| CONSOLIDE208\_Reel avant impôt | CALCULATE(SUM(CONSOLIDE\_EcrituresCouts\_2022[Amount]),FILTER(CONSOLIDE\_CentreDeCouts, CONSOLIDE\_CentreDeCouts[CodePart2 - Nature] ="I")) |
| CONSOLIDE208\_ReelAvantImpot | [CONSOLIDE208\_Reel]+[CONSOLIDE208\_Reel avant impôt] |

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Value |  |
| VIDE | BLANK() |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| 00Mesures | import | calculated | {1} |

# Table: Société

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| Société | import | m |  |

# Table: CONSOLIDE\_JOB

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| JOBKEYCompany | CONSOLIDE\_JOB[No\_]&CONSOLIDE\_JOB[SCTE] |
| Interne/Externe | if(CONSOLIDE\_JOB[Interne] = 0, "Externe", "Interne") |
| CompanyDepKey | CONSOLIDE\_JOB[SCTE]&left(CONSOLIDE\_JOB[No\_],2) |
| Taux\_à\_appliquer | if(CONSOLIDE\_JOB[Interne/Externe] = "Interne", LOOKUPVALUE('Taux FG - En Cours'[TauxInterne], 'Taux FG - En Cours'[CompanyDepKey], CONSOLIDE\_JOB[CompanyDepKey]),  lookupvalue('Taux FG - En Cours'[TotalTaux], 'Taux FG - En Cours'[CompanyDepKey], CONSOLIDE\_JOB[CompanyDepKey]))  //LOOKUPVALUE(RAP201\_PrevisionsEnCours[TotalTaux], RAP201\_PrevisionsEnCours[CompanyDepKey], CONSOLIDE\_JOB[CompanyDepKey]) |
| RAP404\_Interne | if(CONSOLIDE\_JOB[Interne/Externe] = "Externe", blank(), "Dont Interne") |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_JOB | import | m |  |

# Table: Date calages Mois

Mesures:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| MoisSaisie | SELECTEDVALUE('Date calages Mois'[Mois]) |

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Mois |  |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| Date calages Mois | import | calculated | GENERATESERIES(1, 12, 1) |

# Table: Date calages Année

Mesures:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Valeur AnneeSaisie | SELECTEDVALUE('Date calages Année'[AnneeSaisie]) |

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| AnneeSaisie |  |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| Date calages Année | import | calculated | GENERATESERIES(2022, 2030, 1) |

# Table: CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| KeyCompanyCostCenter | 'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Initial Entry Global Dim\_ 1]&'CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[SCTE] |
| Echue | if('CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry'[Initial Entry Due Date] - today() <0, 1,0) |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_Detailed Cust\_ Ledg\_ Entry | import | m |  |

# Table: CONSOLIDE\_EcrituresCouts\_2022

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Encours? | related(CONSOLIDE\_ObjetsDeCouts[201 MULTI SCT]) |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_EcrituresCouts\_2022 | import | m |  |

# Table: CONSOLIDE\_EcrituresCoutsBudget2022

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Cost Object Code (groupes) | SWITCH(  TRUE,  ISBLANK('CONSOLIDE\_EcrituresCoutsBudget2022'[Cost Object Code]),  "(Vide)",  'CONSOLIDE\_EcrituresCoutsBudget2022'[Cost Object Code] IN {"VEN-2013",  "VEN-2014",  "VEN-2015",  "VEN-INT"},  "FAC",  'CONSOLIDE\_EcrituresCoutsBudget2022'[Cost Object Code] IN {"DEP0101",  "DEP0199",  "DEP1013",  "DEP1099",  "DEP1699",  "MACH001",  "MOE0001",  "MOE0002",  "MOE0003"},  "PR",  'CONSOLIDE\_EcrituresCoutsBudget2022'[Cost Object Code]  ) |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_EcrituresCoutsBudget2022 | import | m |  |

# Table: Date2 YTD

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Date |  |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| Date2 YTD | import | calculated | calendar(date(2014,01,01), today() + 365) |

# Table: TEST

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| CentreCoutCompanyKEY | TEST[n°]&TEST[SCTE] |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| TEST | import | m |  |

# Table: Carnet de commande

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| Carnet de commande | import | m |  |

# Table: RAP201\_PrevisionsEnCours

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| TotalTaux | RAP201\_PrevisionsEnCours[TauxDep] + RAP201\_PrevisionsEnCours[TauxFixe] |
| CompanyDepKey | RAP201\_PrevisionsEnCours[Entite]&RAP201\_PrevisionsEnCours[Department] |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| RAP201\_PrevisionsEnCours | import | m |  |

# Table: TRESO Excel

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| Echue | if('TRESO Excel'[Initial Entry Due Date] - 'TRESO Excel'[Période ] <0, 1,0) |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| TRESO Excel | import | m |  |

# Table: EnCours\_CentreDeCouts

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| CostCenterKEYCompany |  |
| EnCours | calculate(-sum(CONSOLIDE\_EcrituresCouts\_2022[Amount]), filter(CONSOLIDE\_EcrituresCouts\_2022, CONSOLIDE\_EcrituresCouts\_2022[CostCenterKEYCompany] = EnCours\_CentreDeCouts[CostCenterKEYCompany] && CONSOLIDE\_EcrituresCouts\_2022[Encours?] = "EN COURS")) |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| EnCours\_CentreDeCouts | import | calculated | distinct(CONSOLIDE\_CentreDeCouts[CostCenterKEYCompany]) |

# Table: CONSOLIDE\_TRESOBC\_TestJGS

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_TRESOBC\_TestJGS | import | m |  |

# Table: CONSOLIDE\_Client

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_Client | import | m |  |

# Table: Taux FG - En Cours

Colonnes calculées:

|  |  |
| --- | --- |
| **Nom** | **Expression** |
| TauxInterne | 'Taux FG - En Cours'[TauxFg Dep] + 'Taux FG - En Cours'[TauxCout Indirect Dep] |

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| Taux FG - En Cours | import | m |  |

# Table: BUDGET Excel

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| BUDGET Excel | import | m |  |

# Table: CONSOLIDE\_Cust Ledger Entry

Partitions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Mode** | **Type source** | **Expression source** |
| CONSOLIDE\_Cust Ledger Entry | import | m |  |