GENERAL
New sample from TransForming table [SOURCE]
New instance [TARGET(t)] is created for every n [SOURCE] samples from TransForming where n = (1..N)
Characteric values will be stored per timeslice from latest sample
See Also Process Flows: "IRIS Starter Pack OEE Analytics Serving "

Structure	OEE_Analytics_Serving			
Identification	Field	TableField	Sequence	Process Logic
	TimeStampLocal	SOURCE.TimeStampLocal	100	TRANSFER SOURCE.TimeStampLocal
	EquipmentID	SOURCE.EquipmentID	110	TRANSFER SOURCE.EquipmentID
	ShiftOperationID	SOURCE.ShiftOperationID	120	TRANSFER SOURCE.ShiftOperationID
	WorkOrder	SOURCE.WorkOrder	130	TRANSFER SOURCE.WorkOrder
OEE_KPI				
	OEEAvailability		920	IF n=N MULTIPLY (DIVIDE OEEActualProductionTime BY OEEPotentialProductionTime) BY 100 ENDIF.
	OEEPerformanceTheoretical		950	IF n=N MULTIPLY (OEETheoreticalProductionTime) BY 100 ENDIF.
	OEEQualityGood		960	IF n=N MULTIPLY (MULTIPLY (OEEActualOutput -/- OEERejected Output) BY (OEEStandardCycleTime)) BY 100 ENDIF.
	OEEQualityActual		970	IF n=N MULTIPLY (MULTIPLY (OEEActualOutput) BY (OEEStandardCycleTime)) BY 100 ENDIF.
OEE_metrics				
	OEEActualProductionTime	TARGET(t).ProductionTime TARGET(t).LineUpTime	790	TRANSFER (TARGET(t).ProductionTime +/+ TARGET(t).LineUpTime)
	OEEPotentialProductionTime	SOURCE.DurationIncrement ShiftOperation.Active	780	IF ShiftOperation.Active = "X" TRANSFER SOURCE(n).DurationIncrement ENDIF.
	OEEActualOutput	TARGET(t).LineUpQty TARGET(t).ProductionQty	770	TRANSFER (TARGET(t).LineUpQty + TARGET(t).ProductionQty)

	OEETheoreticalProductionTime		760	IF SOURCE(n).MachineStatusPreviousSample = (<TransLineUpToProductionStatus> OR <ProductionStatus> OR <TransStandByToProduction> OR <TransForeRunnerToLineUpStatus> OR <LineUpStatus> OR <TransStandByToLineUpStatus>) THEN ADD (MULTIPLY (TARGET(t).LineUpQty + TARGET(t).ProductionQty) BY (OEEStandardCycleTime)) ENDIF.
	OEEStandardCycleTime	TARGET(t).RecipeSpeed	755	TRANSFER [INVERSE (TARGET(t).RecipeSpeed)) * 60]
	OEERejectedOutput	TARGET(t).OOEScrap TARGET(t).OEERework	750	TRANSFER (TARGET(t).OOEScrap +/+ TARGET(t).OEERework)
	OEEScrap	TARGET(t).MisPasQty	745	TRANSFER TARGET(t).MisPasQty
	OEERework		740	n.a.
	OEEUnScheduledDownTime	ShiftOperation.Active TARGET(t).OutOfOrderTime	710	IF ShiftOperation.Active = "X" TRANSFER TARGET(t).OutOfOrderTime ENDIF {when Active Shift}
	OEEFailureTime	TARGET(t).StandByProductionTime	720	TRANSFER TARGET(t).StandByProductionTime
	OEEWaitingTime	TARGET(t).ChangeOverTime	720	TRANSFER TARGET(t).ChangeOverTime
	OOELineRestraintTime	TARGET(t).StandByChangeOverTime	725	TRANSFER TARGET(t).StandByChangeOverTime
	OEEUnScheduledIdleTime		730	n.a.
	OEENotScheduledOperatingTime	ShiftOperation.Active TARGET(t).OutOfOrderTime	715	IF ShiftOperation.Active <> "X" TRANSFER TARGET(t).OutOfOrderTime ENDIF {when Inactive Shift}
Customer_Metrics_Time				
	ProductionTime	SOURCE.DurationIncrement SOURCE.MachineStatusPreviousSample	510	IF SOURCE(n).MachineStatusPreviousSample = (<TransLineUpToProductionStatus> OR <ProductionStatus> OR <TransStandByToProduction>) THEN ADD SOURCE(n).DurationIncrement ENDIF.
	StandByProductionTime	SOURCE.DurationIncrement SOURCE.MachineStatusPreviousSample SOURCE.MachineStatusPreviousStatus	540	IF ((SOURCE(n).MachineStatusPreviousSample = (<TransProdToStandBy> OR <TransStandByToLineUp> OR <TransStandByToProduction>)) OR (SOURCE(n).MachineStatusPreviousSample = <StandBy> AND SOURCE(n).MachineStatusPreviousStatus = (<TotalProduction> OR TransProductionToStandBy))) ADD SOURCE(n).DurationIncrement ENDIF.
	LineUpTime	SOURCE.DurationIncrement SOURCE.MachineStatusPreviousSample	515	IF SOURCE(n).MachineStatusPreviousSample = (<TransForeRunnerToLineUpStatus> OR <LineUpStatus> OR TransStandByToLineUp) THEN ADD SOURCE(n).DurationIncrement
	ChangeOverTime	(TARGET(t).StopTime TARGET(t).ForeRunnerTime)	560	TRANSFER (TARGET(t).StopTime + TARGET(t).ForeRunnerTime)
	ForeRunnerTime	SOURCE.DurationIncrement SOURCE.MachineStatusPreviousSample	520	IF SOURCE(n).MachineStatusPreviousSample = (TransStandByToForeRunnerStatus> OR <ForeRunnerStatus>) THEN ADD SOURCE(n).DurationIncrement

	StandByChangeOverTime	SOURCE.DurationIncrement SOURCE.MachineStatusPreviousSample SOURCE.MachineStatusPreviousStatus	550	IF ((SOURCE(n).MachineStatusPreviousSample = <TransStopToStandBy>) OR (SOURCE(n).MachineStatusPreviousSample = <StandBy> AND SOURCE(n).MachineStatusPreviousStatus = NOT (<TotalProduction> OR TransProductionToStandBy))) ADD SOURCE(n).DurationIncrement
	StopTime	SOURCE.DurationIncrement SOURCE.MachineStatusPreviousSample	525	IF SOURCE(n).MachineStatusPreviousSample = (<TransOoOToStopStatus> OR <StopStatus> OR <TransProductionToStopStatus>) THEN ADD SOURCE(n).DurationIncrement
	OutOfOrderTime	SOURCE.DurationIncrement SOURCE.MachineStatusPreviousSample	530	IF (SOURCE(n).MachineStatusPreviousSample = <OutOfOrderStatus>) THEN ADD SOURCE(n).DurationIncrement
	ChangeOverTotalTime	(TARGET(t).ChangeOverTime Target(t).StandByChangeOverTime	570	TRANSFER (TARGET(t).ChangeOverTime + Target(t).StandByChangeOverTime)
Customer_Metrics_Qty				
	MisPasQty	SOURCE.MisPasQtyIncrement	320	ADD SOURCE(n).MisPasQtyIncrement
	LineUpQty	SOURCE.ProducedQtyIncrement SOURCE.MachineStatusPreviousSample	330	IF SOURCE(n).MachineStatusPreviousSample = (<TransForeRunnerToLineUpStatus> OR <LineUpStatus> OR <TransStandByToLineUpStatus>) THEN ADD SOURCE(n).ProducedQtyIncrement ENDIF
	ProductionQty	SOURCE.ProducedQtyIncrement SOURCE.MachineStatusPreviousSample	340	IF SOURCE(n).MachineStatusPreviousSample = (<TransLineUpToProductionStatus> OR <ProductionStatus> OR <TransStandByToProductionStatus>) THEN ADD SOURCE(n).ProducedQtyIncrement ENDIF.
	ProducedQtyOrder	SOURCE.ProducedQtyOrder	350	TRANSFER SOURCE(n).ProducedQTYOrder
	UoM	SOURCE.UoM	310	TRANSFER SOURCE(n).UoM
Customer_Order_Characteristics				
	Recipe	SOURCE.Recipe	200	TRANSFER SOURCE(n).Recipe
	RecipeSpeed	SOURCE.RecipeSpeed	220	TRANSFER SOURCE(n).RecipeSpeed

Calculation of OEE KPI's per Time Period		
OEE_Availability		AVERAGE of OEEAvailability
OEE_Performance		DIVIDE (SUM (OEEPerformanceTheoretical)) BY (SUM (OEEActualProductionTime))
OEE_Quality		DIVIDE (SUM (OEEQualityGood) * 100) BY (SUM (OEEQualityActual))
OEE_Overall		DIVIDE (OEE_Availability * OEE_Performance * OEE_Quality) BY 10000
OEE_Productivity		DIVIDE (OEE_Availability * OEE_Performance) BY 100