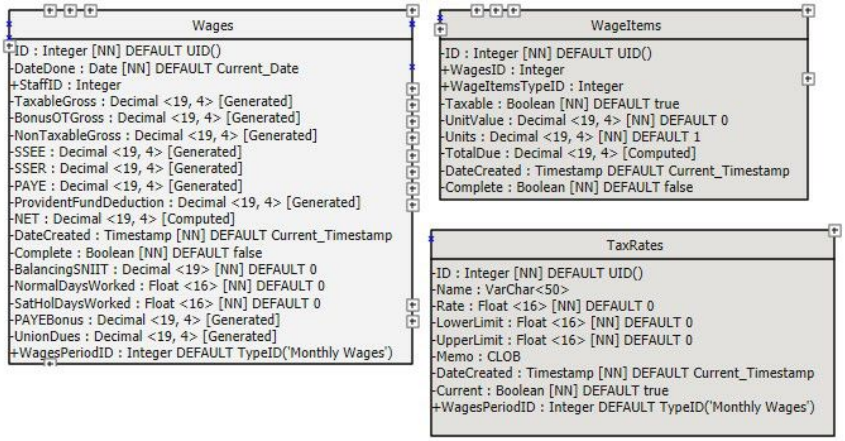
Tax, PAYE and SSNIT Computation in SerendiSys

> **SerendiSys includes automation to compute tax rates for staff wages. How the database procedures and data-tables link together to enable this is explained in the following article.**
>
> **To understand this article it may be useful to first review how GENERATION and COMPUTATION of columns works in Orixa and SQL databases. This is covered in the linked article, below.**

How data-field computation works in Orixa

Wages Data Table



Wages, WageItems and TaxRates

```
CREATE TABLE "Wages"
(
"ID" INTEGER DEFAULT UID() NOT NULL,
"DateDone" DATE DEFAULT Current_Date NOT NULL,
"StaffID" INTEGER,
"TaxableGross" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN "TaxableGross" ELSE
TaxableGross(ID)),
"BonusOTGross" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN "BonusOTGross" ELSE
BonusOTGross(ID)),
"NonTaxableGross" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN "NonTaxableGross" ELSE
NonTaxableGross(ID)),
"SSEE" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN SSEE ELSE SSNIT('SSEE', StaffID,
WagesPeriodID)),
"SSER" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN SSER ELSE SSNIT('SSER', StaffID,
WagesPeriodID)),
"PAYE" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN "PAYE" ELSE PAYE("TaxableGross" -
SSEE, StaffID, WagesPeriodID)),
"ProvidentFundDeduction" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN
"ProvidentFundDeduction" ELSE ProvidentFundDeduction(StaffID)) ,
"NET" DECIMAL(19,4) COMPUTED ALWAYS AS "TaxableGross"
+ "NonTaxableGross"
+ "BonusOTGross"
- "PAYE"
- "PAYEBonus"
- "SSEE"
- "ProvidentFundDeduction"
- "UnionDues" ,
"DateCreated" TIMESTAMP DEFAULT Current_Timestamp NOT NULL,
"Complete" BOOLEAN DEFAULT false NOT NULL,
"BalancingSNIIT" DECIMAL DEFAULT 0 NOT NULL,
"NormalDaysWorked" FLOAT DEFAULT 0 NOT NULL,
"SatHolDaysWorked" FLOAT DEFAULT 0 NOT NULL,
"PAYEBonus" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN "PAYEBonus" ELSE
BonusPAYE("BonusOTGross", StaffID, WagesPeriodID)),
```

```
"UnionDues" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN UnionDues ELSE
UnionDues(StaffID)),
"WagesPeriodID" INTEGER DEFAULT TypeID('Monthly Wages'),
CONSTRAINT "PK_Wages" PRIMARY KEY ("ID"),
CONSTRAINT "StaffID" FOREIGN KEY ("StaffID") REFERENCES "Staff" ("ID")
ON UPDATE NO ACTION ON DELETE NO ACTION
DESCRIPTION 'Default',
CONSTRAINT "WagesPeriodID" FOREIGN KEY ("WagesPeriodID") REFERENCES "Types" ("ID")
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

1. Child data-records are added to the "WageItems" data-table for individual items of staff pay, such as "Salary", "Allowances" etc.
2. The individual amounts added here are summed and computed into the Parent record of the "Wages" data-table using database functions. In the case of each field in the Wages table, a function is called, which pulls data from the "WageItems" data-table, and then undertakes computations on them to compute the correct values.
3. The whole computation process depends on correct values for all types of tax being included in the "TaxRates" reference data-table.

Explanations for each column of the Wages data-table

"TaxableGross" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN "TaxableGross" ELSE TaxableGross(ID))

The TaxableGross value is set by calling the TaxableGross Function. The SQL of this function can be viewed in the DB Utility in SerendiSys. It returns the **total** from the following SQL:

```
SELECT
WagesID,
SUM(TotalDue) as Total
FROM WageItems
WHERE WagesID = ?
AND Taxable = true
AND NOT WageItemsTypeID IN (17372, 22257198)
GROUP BY WagesID
```

Note that this returns **all** WageItems where "Taxable" = true, provided that the WageItemsType is**not** a "Bonus" or "Overtime" item. These wage items are excluded because they are taxed differently.

"BonusOTGross" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN "BonusOTGross" ELSE BonusOTGross(ID))

The BonusOTGross is computed in the same way as TaxableGross, but calls the "BonusOTGross" function. This SQL of this function can be viewed in the DB Utility. It returns the **total** from the following SQL:

```
SELECT
SUM(TotalDue) as Total
FROM WageItems
WHERE WagesID = ?
AND WageItemsTypeID IN (17372, 22257198)
```

Note that in this case **all** wage-items that are "Bonus" or "Overtime" are included. The separate heading is included as Bonuses are taxed differently from the "Main" wage.

"NonTaxableGross" DECIMAL(19,4) GENERATED ALWAYS AS NonTaxableGross(ID)

This function works in the same way as the previous 2, with a SELECT statement returning a SUM of values from the WageItems data-table. This time the value is the sum of all WageItems where "Taxable = false."

"SSEE" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN SSEE ELSE SSNIT('SSEE', StaffID, WagesPeriodID))

This column is computed using the "SSNIT" function, which can be viewed in the DB Utility, and which is explained below. It brief, it fetches the "BasicSalary" for a staff member, and then references the "TaxRates" table to make the computation of how much SSNIT should be paid.

"SSER" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN SSER ELSE SSNIT('SSER', StaffID, WagesPeriodID))

This column is computed the same way as the "SSEE" Column, but uses records in the TaxRates table for SSER instead of SSEE.

"PAYE" DECIMAL(19,4) GENERATED ALWAYS AS PAYE(TaxableGross - SSEE, StaffID, WagesPeriodID)

This column is computed in a similar way to SSEE and SSER, except that the "PAYE" Function is referenced instead of the "SSNIT" Function, and the **taxable amount** of wages (TaxableGross - SSEE) are passed into the Function.

"ProvidentFundDeduction" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN "ProvidentFundDeduction" ELSE ProvidentFundDeduction(StaffID))

This column is computed by passing the "StaffID" for the current row into the "ProvidentFundDeduction" Function, and returning the value it generates into the data-table.

"NET" DECIMAL(19,4) COMPUTED ALWAYS AS

"TaxableGross"
+ "NonTaxableGross"
+ "BonusOTGross"
- "PAYE"
- "PAYEBonus"
- "SSEE"
- "ProvidentFundDeduction"
- "UnionDues"

This column is a simple mathmatical sum of other columns. Note that because it is a simple mathematical process, it is defined using the "COMPUTED" rather than "GENERATED" SQL Key-word.

"BalancingSNIIT", "NormalDaysWorked", "SatHolDaysWorked"

These are all simple number columns which users can edit and add data to. The values in them are also set by other processes in SerendiSys, such as rapid data entry grids.

"PAYEBonus" DECIMAL(19,4) GENERATED ALWAYS ASIF(Complete = true THEN "PAYEBonus" ELSE BonusPAYE("BonusOTGross", StaffID, WagesPeriodID))

This calls the BonusPAYE Function.

"UnionDues" DECIMAL(19,4) GENERATED ALWAYS AS IF(Complete = true THEN UnionDues ELSE UnionDues(StaffID))

This calls the UnionDues Function

Functions called by the Wages data-table

The following functions can all be viewed and reviewed in the DB Management Utility within SerendiSys

SSNIT

IN Parameters: TaxType ("SSEE" or "SSER"), StaffID (the person being taxed) WagePeriodID (Weekly or Monthly).

The staff member's "BasicSalary" is returned from the record in the "WageItems" data-table. This has previously been generated from the Staff Members "Basic Salary" (in the SalaryDetails table) multiplied by the number of work units (days, hours or Months). Tax Rates for either SSEE or SSER are returned from the TaxRates data-table. A**WHILE** loop is used to work through each row in the cursor containing TaxRates data, and use it to compute the taxable amount.

```
FETCH FIRST FROM Crsr('Rate', 'LowerLimit', 'UpperLimit') INTO Rate, LowerLimit, UpperLimit;
WHILE NOT EOF(Crsr) DO
IF (BasicSalary) > LowerLimit THEN
IF (BasicSalary) > UpperLimit THEN
SET Result = Result + (Rate * (UpperLimit - LowerLimit));
ELSE
SET Result = Result + (Rate * (BasicSalary - LowerLimit));
END IF;
END IF;
FETCH NEXT FROM Crsr('Rate', 'LowerLimit', 'UpperLimit') INTO Rate, LowerLimit, UpperLimit;
```

```
END WHILE;
```

Note in the above:
FETCH FIRST FROM Crsr('FieldName') INTO VariableName;
This "sucks in" the first value in the data into a variable.
WHILE NOT EOF(Crsr) DO
The "EOF" SQL Keyword above stands for "End Of File", so the above line reads as "while you have not reached the last record in the data-set held by the cursor, run the lines of code which follow"
A basic understanding of simple programatic concepts such as WHILE ... END WHILE, IF ... END IF, CASE ... END CASE. Is useful here. If you need more information on this, please Google "SQL WHILE KEY WORD".

PAYE and PAYEBonus

These functions work in almost exactly the same way as the SSNIT Functions. Please review their SQL in the DB Management Utility to understand them better.

ProvidentFundDeduction

This function looks up the BasicSalary in the same way as the SSNIT Function, and returns the ProvidentFundContribution from the SalaryDetails data-table, this is a percentage value such as "0.03" or "0.05". The function multiplies ProvidentFundContribution and BasicSalary alues together to create a result.

UnionDues

This function retrieves the "Union Dues" Rate from the Tax Rates data-table, the staff-members BasicSalary is computed as for other Functions, the code also checks whether a staff member is a Union Member by referencing the "UnionMember" field in the Staff data-table, and then returns the value computed by the UnionDues Rate multipled by the Basic Salary.