**TS Database Structure Overview**

Contents

[1. Database Overview 2](#_Toc457495210)

[2. Getting table related information in SQL 2](#_Toc457495211)

[3. Most Important Guidelines 3](#_Toc457495212)

[4. Supplier and Price Contract Related Tables 3](#_Toc457495213)

[5. Service 5](#_Toc457495214)

[6. Service related Tables 7](#_Toc457495215)

[7. Client related tables 14](#_Toc457495216)

[8. Passenger related Tables 15](#_Toc457495217)

[9. Package related Tables 16](#_Toc457495218)

[10. Linked Packages Specific Table Structures 18](#_Toc457495219)

[11. Rail Specific Table structures 19](#_Toc457495220)

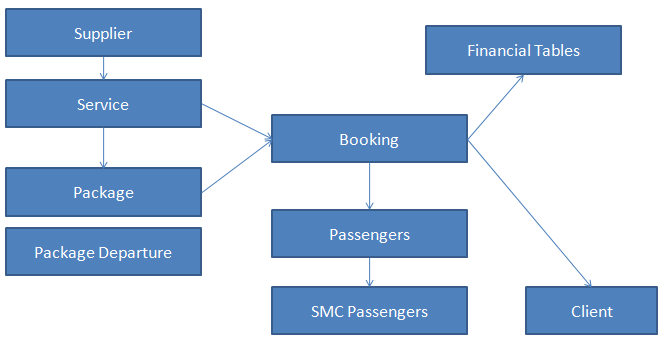
[12. Booking related tables in TS 20](#_Toc457495221)

[13. Booking Finance related tables in TS 23](#_Toc457495222)

[1. Region Table and its Linked Tables 27](#_Toc457495223)

# Database Overview

Below is the basic structure of the TS database, most of the table structures belong to one of these major data areas of TS. The below structure also displays the major relationships and dependency flow between the major areas of TS.



# Getting table related information in SQL

Using the SQL manager you get the details of each table like the list of Column definitions, Primary/foreign keys, Index etc.

**Find tables in database**

To find certain tables you can use sysobjects with xtype = ‘u’

select\*from sysobjects wherextype='u' and name like'%supplier%'orderbyname

--OR

SELECT \* FROM sys.tables

WHERE name LIKE '%SUPPLIER%'

**Find SPs in database**

To find certain tables you can use sysobjects with xtype = ‘p’

Select \* from sysobjects where xtype='p'and name like '%ins%organisa%' orderby name

--OR

SELECT \* FROM sys.procedures

WHERE name LIKE '%SUPPLIER%'

**Get Details of Table**

Sp\_help <Table Name>

**Get Stored Procedure’s Details**

SP\_HELPTEXT <SP Name>

**To find out any column name in all tables of db**

SELECT t.name AS table\_name, SCHEMA\_NAME(schema\_id) AS schema\_name, c.name AS column\_name

FROMsys.tables AS t

INNERJOINsys.columns c ONt.OBJECT\_ID = c.OBJECT\_ID

WHERE c.name LIKE'%SupplierID%'

ORDERBYschema\_name, table\_name;

SELECT T.NAME AS TableName,

C.name AS ColumnName

FROM sys.tables T

INNER JOIN sys.columns C ON T.object\_id = C.object\_id

WHERE c.name LIKE '%SUPPLIERID%'

# Most Important Guidelines

* **Table and colum**n names should be **Meaningful** and expresses its usage
* **Primary Key** should always be defined for new tables
* **Foreign Key reference**s should be created for all for ID fields which are referenced from other tables
* Validate if **INDEX is** needed and appropriate index should be created.
* **Amount** Fields should be defined using **Money**datatype
* **Numeric** fields with decimal places like Percentage etc should be defined as **Decimal or float**
* Add appropriate/correct field sizes for varchar and char fields
* Table Name with Multiple words are separated by **Underscore**

Refer to coding guidelines document for more guidelines on SQL

# Supplier and Price Contract Related Tables

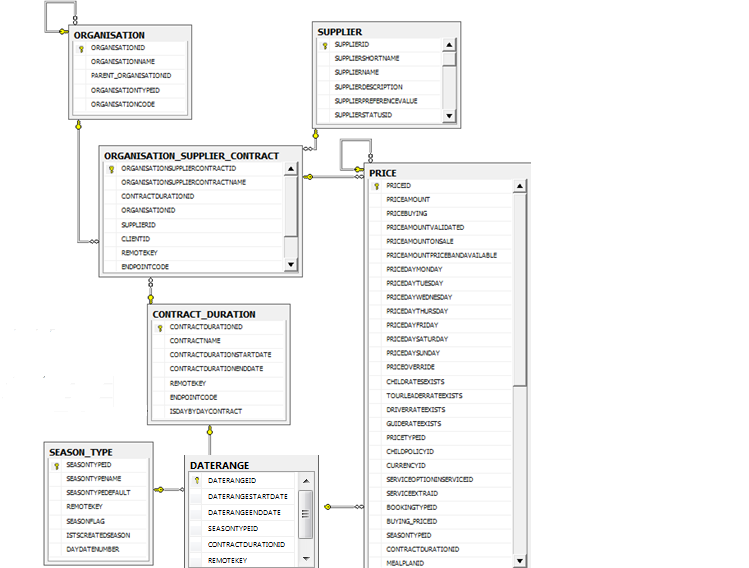
A Supplier is a company which provides the actual hotel, flight and other products.

***CONTRACT***

*Contract is an Agreement between the Tour Operator and the Supplier.*

*A supplier contract has Contract Duration and is further split into Seasons.*

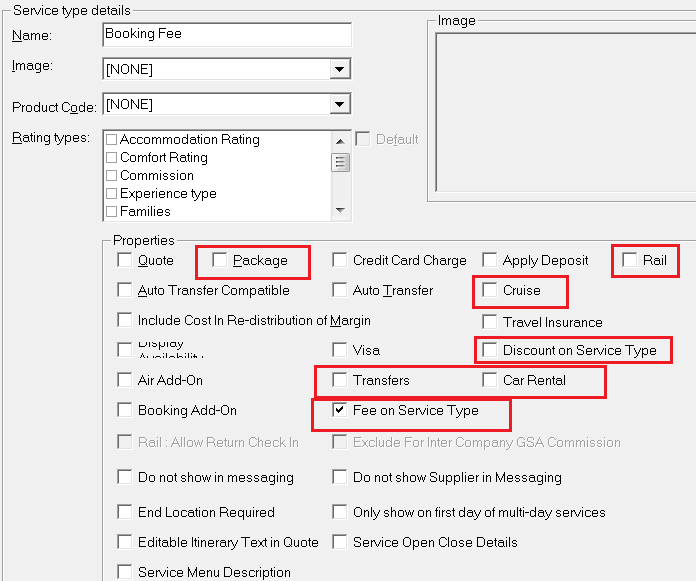
|  |  |
| --- | --- |
| *Service with Contract for 2 years with different seasons* | ***Shared contract***  *1 contract can be shared with multiple services*  *Better to have separate contract for each service. When a season split occurs the price needs to be updated in both service with can be bit difficult.* |



# Service

Service is a term referring to any entity which the Tour Operator can sell. It can be of a specific type like hotel, transfer, car-hire or it could also be types like a one day city trip, or some extra charge or ferry tickets etc.

Service is provided by a supplier. A supplier can have many different kinds of services



**SERVICE TYPE**

A service can be of different types like, hotel, car-hire, transfer, flights. In TS the service Type setup defines the type of service and for some of the types we have specific flags to identify the type and provide slightly different functionality.

* **Accommodation or Hotel** – No specific flag
* **Car-Hire** – Has a flag in service type **“Car Rental”**
* **Transfers** – Has a flag in service type **“Transfer”**
* **Rail** – Has a flag in service type **“Rail”**
* **Cruise** – Has a flag in service type **“Cruise”**
* **Discount** – Has a flag **“Discount on Service Type”**
* **Booking Fee** - Has a flag **“Fee on Service Type”**

Service consists of following details

**OPTIONS**

* + These are the Main selling items of a service and is usually chargeable.
  + Hotel Service will have Room options like “Single room”, ”Sea facing Room”, “Double Room” etc
  + Transfer will have Options like “4 seater”, “7 seater” etc

**Option Status**

This is an important field.

Each option can have a status which can be set in “Details” tab which indicated if it can be returned in search results or not. IF not allowed it should not returned in search and should not be allowed to be booked

**EXTRAS**

* + Additional optional selling items of a service which can either be chargeable or free.
  + Not mandatory to book an extra

**Extra Status**

This is an important field.

Each extra can have a status which can be set in “Details” tab which indicated if it can be returned in search results or not. IF not allowed it should not returned in search and should not be allowed to be booked

**FACILITIES**

* + Facilities provided at the service like gym, library.
  + This is non-Chargeable and TS does not maintain any prices.
  + It is used for information purpose
  + It is a very important field used on websites to filter services in search as usually customers search hotels by facilities they provide.

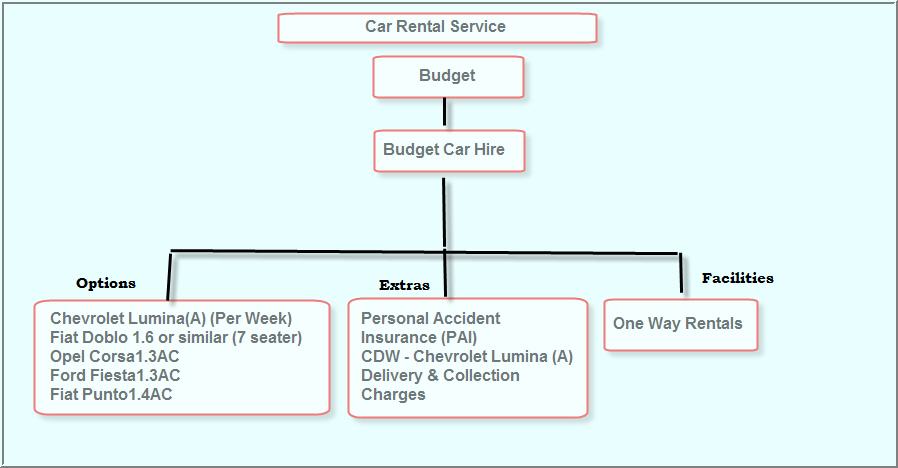
**RATINGS**

* + The ratings for a service e.g. for hotel it can be 3 star, 5 star etc.
  + User can define custom ratings also like Trip-advisor ratings etcIt can also be used to set some categories to the service.
  + Multiple rating types can be set against the service. E.g. Hotel can be marked as “3 star” and “budget” hotel
  + **Ratings are also very important for generating Sell Price.**

**Linked Options/Extras/Facilities**

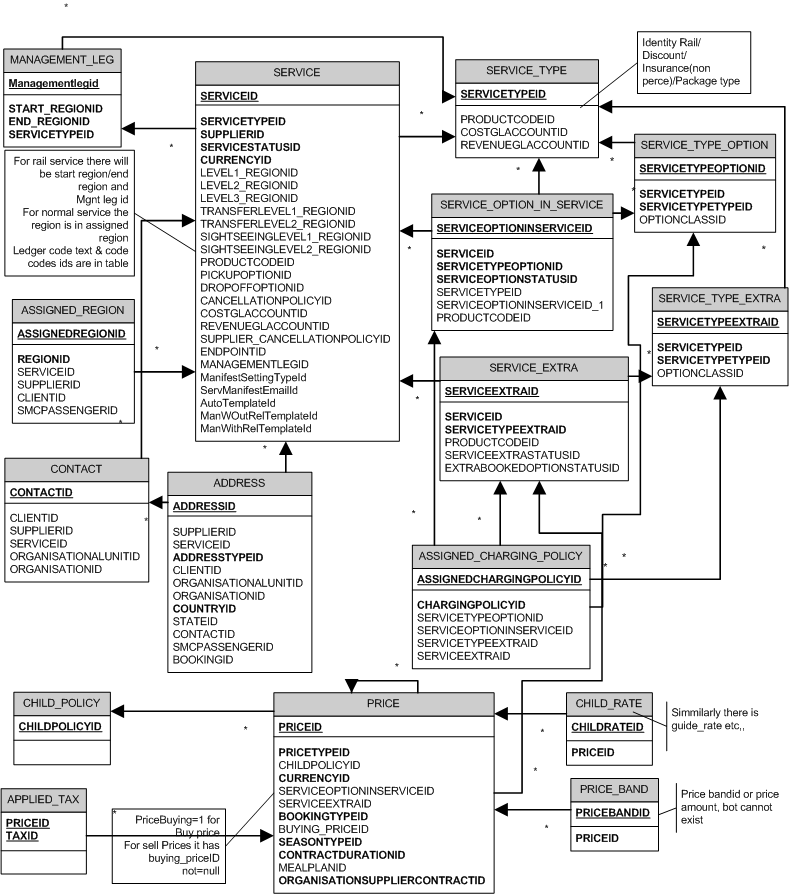
* Options can also be linked to other options. This means booking of both options is mandatory.
* The Extras can be linked to options. This means that, that extra is available only for that option.
* Facilities can be linked to options which means certain facilities are available only for certain options.
* **Mandatory Extras**: Extras can be made mandatory. This is useful when extras need to be booked compulsorily. E.g. Christmas Dinner.

|  |  |
| --- | --- |
|  |  |



# Service related Tables

Below is a database structure defining the important tables involved with a TS service, for each table the important fields are mentioned.



|  |  |
| --- | --- |
| **TABLE NAME** | **DESCRIPTION** |
| **Supplier** | Stores header information of Supplier like Supplier Name, Status, Supplier Reference, etc. |
| **Service\_Type** | Stores Service Type details like Name, Description, etc.  Flags to determine if the type is Accommodation, Package, Car-Hire, Transport, etc. |
| **Service\_Type\_Option** | Stores information about Optionslike Name , Type etc |
| **Service\_Type\_Extra** | Stores Option about Extras like ExtraName, Type etc |
| **Service** | All Service header details are stored here like LongName, Short name, Description, Service type, some details from terms/others tab like Inclusions/exclusions etc |
| **Service\_Option\_In\_Service** | Stores information about Service Options added in a Service. Details like ServiceOptionInServiceQuantity, ServiceID, etc |
| **Service\_Extra** | Stores information aboutService Extras added for Service. Details like ServiceExtraMandatory, ServiceID |
| **Organisation\_Supplier\_Contract** | Stores details of Contracts of the Organization Suppliers. Details like OrganizationSupplierContractName, supplierid, organisationid, contractdurationidetc |
| **Contract\_Duration** | Stores details of Contract's duration. Details like Contact Name, Contract Start Date, Contract End Date are stored here. |
| **Season\_Type** | Stores details of the Season Type like SeasonTypeName, SeasonTypeDefault |
| **Daterange** | Stores the link from the season\_type and contract\_duration tables to the price table. This table would give us the season start and end dates. |
| **Price** | Stores Buy and Sell Prices as separate records details for each option or extra and each BTPT combination. Important columns PriceBuying,OptionID, ExtraID, BookingTypeID, PriceTypeIDContractDurationID, SeasonID,PriceAmountetc |
| **NewPrices** | Denormalized cache table to store sell prices. It is to be used only in search for faster retrival and sell prices in this table should always be in sync with prices in “Price” table |
| **Price\_Band** | Stores information on Price Bands. Details like PriceBandMin, PriceBandMax, PriceBandAmount, etc are stored. |
| **Applied\_Tax** | Stores information on Tax Applied. Details like the AppliedTaxSequence, TaxInclusiveCodeId, PriceID, etc are stored |
| **Child\_Rate** | Stores information on rates for children. Details like the ChildRateFromAge, ChildRateToAge, ChildRateAmount, etc are stored. |
| **Child\_Policy** | Stores information on Child Policies. Details like ChildPolicyName, ChildPolicyDefault, etc are stored. |

**Supplier (Hotel Chain or group providing service, can have one or more services)**

[Supplier] -> 1:n -> [Services]

**Service (each unit/property/Hotel provided by the supplier)**

[Service] -> 1:n -> [Option]

**Option (individual Room i.e. double room, triple room)**

**Price**

[Option] -> 1:n -> [Prices]

(one or more prices for different Rate Types (@BookingTypeID, @PriceTypeId combinations))  
**Allocation**

[Option] -> 1:n -> [Allocation]

(for different dates and either Shared with other options OR defined separately for each option (room types))

**OR**

[Option] -> 1:n -> [Allocation]

(for combination of dates, options (room types) &**RateType**).

**PRICE TABLE**

The Pricing table stores information of

* Both **buy and sell** prices as separate records
  + If column “PriceBuying” value stores 1, then it indicates it is a Buying Price record else it is a Selling Price.
* For each **option and extra**.
* And **each BTPT** combination

***PRICING***

Prices are of 2 types – **buy and sell**

**Buy:** The price at which Tour operator buys the service from supplier.

**Sell:** The price at which the Tour operator sell to the passenger or agent. The tour operator applies various types of calculations on the buy price to arrive at the final sell price.

Markup

Margin

Sell

Buy

Final

Sell

Exchange Rate

Tax

Supplements

Yield Rule

***CHARGING POLICY AND CHARGING DURATIONS***

These indicate how the price will be applied, whether per person or per room and which duration.

There are two important settings:

1. **Charging Duration**
   1. Indicates the duration for calculating the charge
   2. This can be specified in Minutes, Hours, Days Weeks and Months.
   3. Consider a Booking for 6 days with 2 passengers with price 100

Duration 1 day: Specifies to apply the charge per-day i.e. 100 \* 2 \* 6 = 1200.

Duration 3 days: Indicates apply the charge every 3 days. i.e. 100 \* 2 \* 2 = 400

1. **Unit Based flag** – will indicate if rate is per person or for the unit/room

Consider Booking with 4 adults in a Quad room of price 100 for 5 days.

* 1. **Unit Based Ticked**: Price \* number of days (100 \* 5 = 500)
  2. **Unit Based Un-Ticked:** Price \* Number of persons \* Number of days = (100\*4\*5 = 2000)

1. **Day Overlap** –indicates if last day of booking is to be included in final price
   1. If ticked then the last day will not be considered in the Booking total price.
   2. If not-ticked then the last day will be considered in the Booking total price.

***BOOKING/PRICE TYPE COMBINATIONS***

The rate code combination that is available for pricing and allocation.

The Rate Types are a **combination of 2 fields**, Booking Type and Price Type.

For example,

* Booking Type = FIT / Price Type = Room Only
* Booking Type = FIT / Price Type = Room & Breakfast
* Booking Type = FIT / Price Type = Non-Cancellable

A service can have multiple rate types. Within each rate type, options can store one or more prices.

For example;

* **JW Marriot – New York (Service)**
* Contract – 01Apr14–31Mar15

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | **Low** | **High** | **Mid** |
| **Option** | **Booking Type** | **Price Type** | **01Apr14-30Sep14** | **01Oct14-31Jan15** | **01Feb15-31Mar15** |
| Single Room | FIT | Room Only | 150 | 180 | 160 |
| Double Room | FIT | Room Only | 200 | 220 | 210 |
| FIT | Non-Cancellable | 180 | 200 | 190 |
| FIT | BAR | 170 | 190 | 180 |

* Contract – 01Apr15–31Mar16

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | **Low** | **High** | **Mid** |
| **Option** | **Booking Type** | **Price Type** | **01Apr15-30Sep15** | **01Oct15-31Jan16** | **01Feb16-31Mar16** |
| Single Room | FIT | Room Only | 170 | 200 | 180 |
| Double Room | FIT | Room Only | 220 | 240 | 230 |
| FIT | Non-Cancellable | 200 | 220 | 210 |
| FIT | BAR | 190 | 200 | 200 |

***CHILD PRICING***

***Rates for Children can be defined against the service option.***

***Child can be of two types:***

***Child Sharing:*** *They share the same bed with adults.*

***Child Not Sharing:*** *They don’t share the bed with adults.*

*Child sharing is only for accomodation type of option. Non accomodation option don’t have child sharing.*

***Two ways of defining child rates.***

1. ***Fixed rates:*** *Rates can be defined as a fixed amount. This is done in the policies column. Fixed Rates for child Sharing and Child Not Sharing can be defined.*
2. ***Percentage rates:*** *Child Rates can be defined as a percentage of the adult rate. It’s a two step method.****a)*** *Define child rates using Product🡪Child policies. Rates can be defined for Child Sharing and Child Not Sharing.****b)*** *apply the child policy against the service option price under Child policy column.*

*The advantage of using the percentage based Child policy is that if the adult rates changes, child prices are automatically recalculated at service pricing.*

***Additional Child***

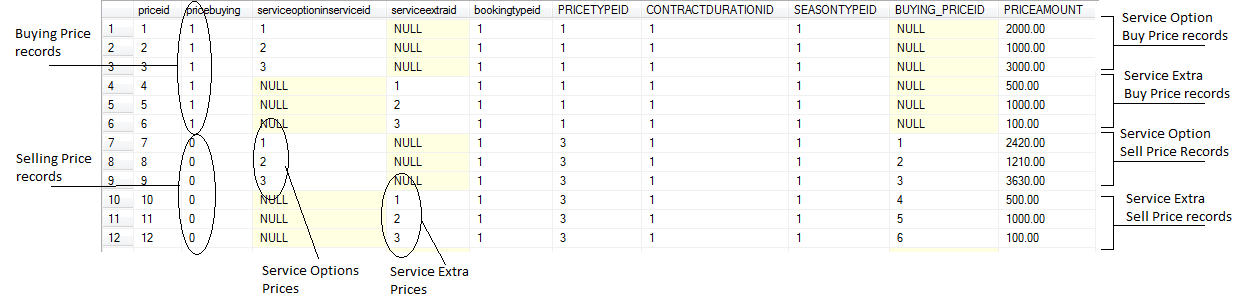
***PRICE BANDS:*** *Rates can be defined for service options based on price bands.*

*Price band is of 3 types:*

1. ***Pax based:*** *Rates can be defined based on number of passengers booking the service option. Usually pax based price bands are used for non accomodation services.*
2. ***Day based:*** *Rates can be defined based on number of days the service option is booked****.***
3. ***Time based :****This is used in Car Rental and Transfer service type. This is very useful when defining rates based on time of the day. Eg peak hours, night hours etc.*

***GROSS PRICE***

*A supplier may give commission to a tour operator. To reflect this, we can make use of the Gross price functionality.*



**NEWPRICES TABLE**

* The table “NewPrices” is a **cache** table for **storing “Sell”** prices. Buy prices are not stored in this table.
* The sell prices are stored in a de-normalised table structure to help in faster retrieval in search.
* This table is to be **used “only in Search”**. It should not be used in booking/amendor any other flows.
* The sell price in NewPrices table **should always be in sync** with the sell prices in the “price” table. Therefore when any price change occurs either from service maintenance, SE, Channel manager, batch sell or any other area the NewPrices table needs to be updated for the changed prices.

There are some SP’s created which will refresh the NewPrice table based on different criteria.

* usp\_Ins\_Opt\_Price\_For\_TSHotel\_By\_CostPriceID
* usp\_Ins\_Opt\_Price\_For\_TSHotel\_By\_CurrencyID
* usp\_Ins\_Opt\_Price\_For\_TSHotel\_By\_DateRange
* usp\_Ins\_Opt\_Price\_For\_TSHotel\_By\_DateRange\_AND\_BTPT
* usp\_Ins\_Opt\_Price\_For\_TSHotel\_By\_ServiceID
* Usp\_ins\_opt\_price\_for\_tshotel\_by\_servicetypeoption\_or\_extraid

**ALLOCATION TABLE**

***These are the units/rooms available.***

Allocations are defined for

1. A**date-range** and;
2. Linked to either

* **Each Option** (Room type) could have different units available, e,g. single 10. Double 20
* Or it can be **shared with multiple Options** (room types).

E.g. 30 units overall for single, double, triple.Any room booked will reduce the allocation from the shared value.

1. Optionally, in addition to being linked to dates and Options (room type), Allocations can also be linked to a set of Rate Types

e.g. 10 Units for Bar Rates with Booking Type “FIT” and price type “BAR”

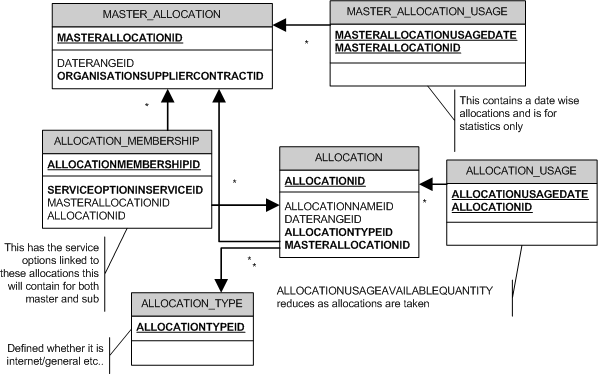
e.g. 20 Units for Regular Rates with Booking Type “FIT” and price type “Room Only”

**Note:** Link to rate type is Optional. If Allocation is not linked to a rate type then it is applicable to all rate-types and any booking done with a specific rate-type will reduce the allocation.

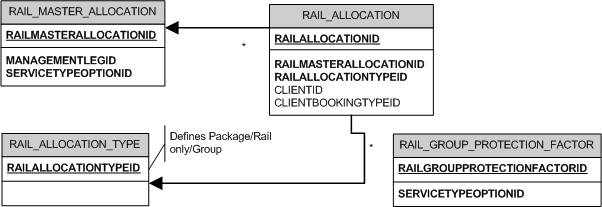
**ALLOCATION TYPES**

* **General** – Used in TS/TSv2,
* **Internet** – Used in B2C/B2B API. If flag “use General Allocation” is ticked in organization settings it will skip and used general allocations.
* **Client** – Used for Agent Bookings.
* **Package** – used for service added to a package. The allocation name has to match correctly.

Below is a structure that shows the tables involved in defining the Allocation structure for a TS service.



Below is a structure that shows the tables involved in defining allocations for Rail services in TS



There is not central quantity that is reduced or totals the bookings. It is calculated on the fly.

For each date seen on the screen there will be multiple rows in rail\_allocation, 1 for rail only, one for package, and one each for defined client/Booking Type combination

**usp\_get\_rail\_allocation\_availability** is the SP used to calculate and get allocations for rail. The execution structure for this SP is as below, the variables will have to be replaced with the equivalent Database IDs/Values.

EXECUTE USP\_GET\_RAIL\_ALLOCATION\_AVAILABILITY

@ri\_MANAGEMENTLEGID

,@ri\_SERVICETYPEOPTIONID

,@rd\_ALLOCATIONDATE

,@ri\_SERVICEID

,@ri\_CLIENTID

,@ri\_PACKAGEDEPARTUREID

,@ri\_AVAILABLEQTY OUTPUT

,@ri\_RAILALLOCATIONID OUTPUT

,@rb\_SEARCHALLOCATIONSONLY

,@ri\_BOOKINGTYPEID

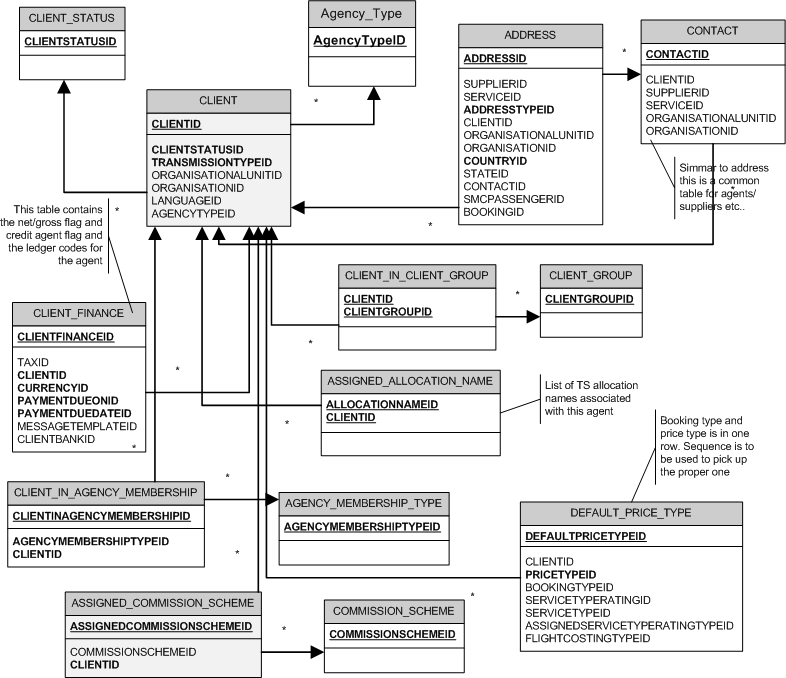
,@ri\_RAILALLOCATIONTYPEID

,@ri\_ORG\_SERVICEID

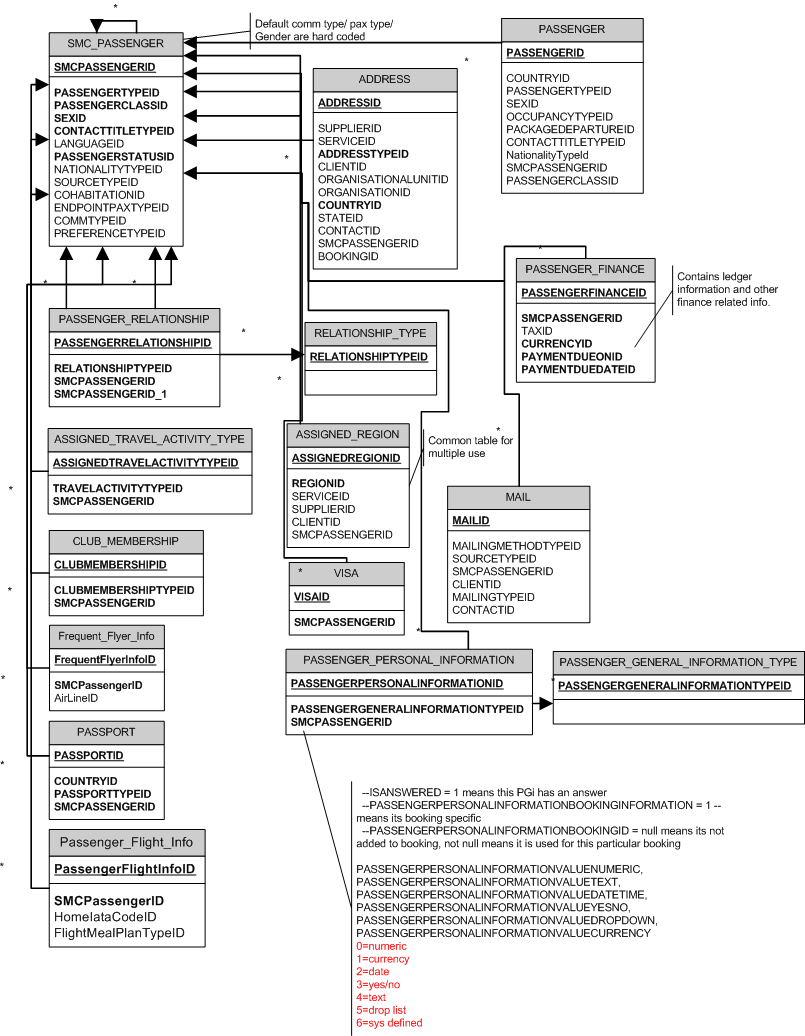
,@ri\_ORIGRAILALLOCATIONID

# Client related tables

Below structure defines the various tables involved in defining a Client in TS.



# Passenger related Tables



# Package related Tables

* Package
  + Is a collection of services of different kinds like hotel, car, transfer, flights etc and sold at a specific price.
  + A package is a template based on which departures are created. The entire details are copied to the departure when it is created. In booking a departure of the package is booked and not the actual package.
  + Types – Package are of 3 types

1. **Seat on Bus (SOB)**

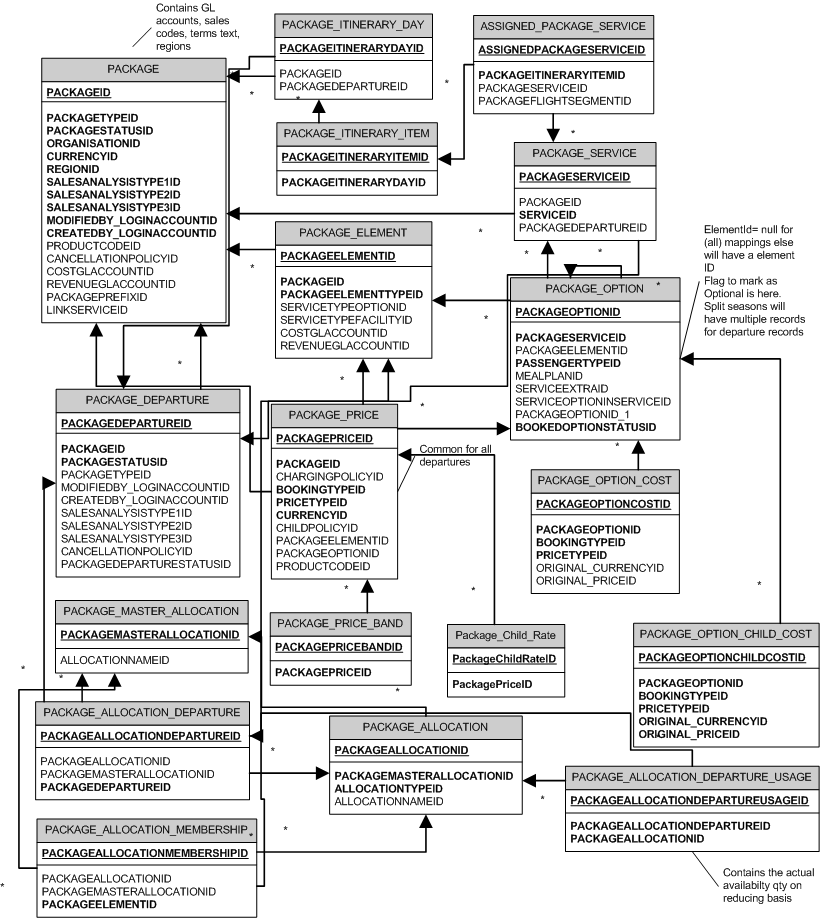
Planned tour, mostly with fixed departure dates and fixed seats. Allows complete departure maintenance.

1. **Fixed departure (FD)**

Fixed departure dates. Does not allow departure maintenance.

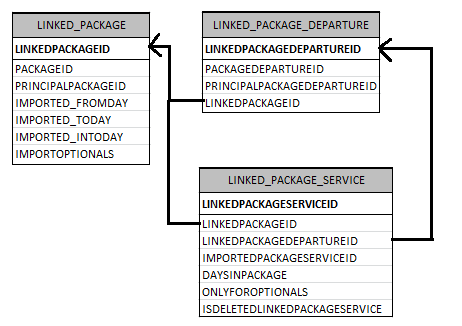
1. **Non-fixed Departure (NFD)**

These are usually customised tours which can be planned on any dates and therefore departures are not created. Only when the package is booked a departure is created dynamically at the time of booking. Package allocations are not used for NFD.



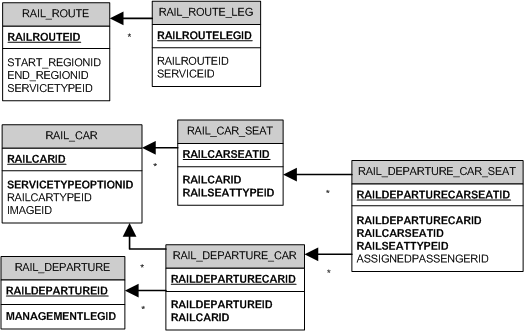
|  |  |
| --- | --- |
| **TABLE NAME** | **DESCRIPTION** |
| **Package** | Stores Header details of Packages like Package Long, short names, Description etc |
| **Package\_Departure** | Stores details of package departures like package departure Date, Name, Reference, package status idetc |
| **Package\_Element** | Stores the details of Elements in a Package. |
| **Package\_Option** | Stores details on Options added to a Package. Details like Package Option ID, Package Option Number of Nights, Package Option Quantity, etc. |
| **Package\_Option\_Cost** | Stores the cost for each Package Option. Details like Package Option Cost ID, Package Option Cost Amount |
| **Package\_Allocation** | Stores the link between Package Allocation ID and Package Master Allocation ID. |
| **Package\_Allocation\_Departure** | Stores details on Package Allocation Departure like the Package Allocation Departure ID, Package Allocation Departure Release Period, |
| **Package\_Allocation\_Membership** | Store the link details between the Package Allocation ID and Package Master Allocation ID |
| **Package\_Child\_rate** | Stores details Package Child Rate ID, the Child rate From Age, the child rate To Age, Child Rate Amount, etc. |
| **Package\_Itinerary\_Day** | Stores details about Days present in the Package Itinerary. Details like Package Itinerary Day ID, Package Itinerary Day sequence |
| **Package\_Itinerary\_Item** | Stores details about Items added to Days in a Package Itinerary. Details like Package Itinerary, |

# Linked Packages Specific Table Structures

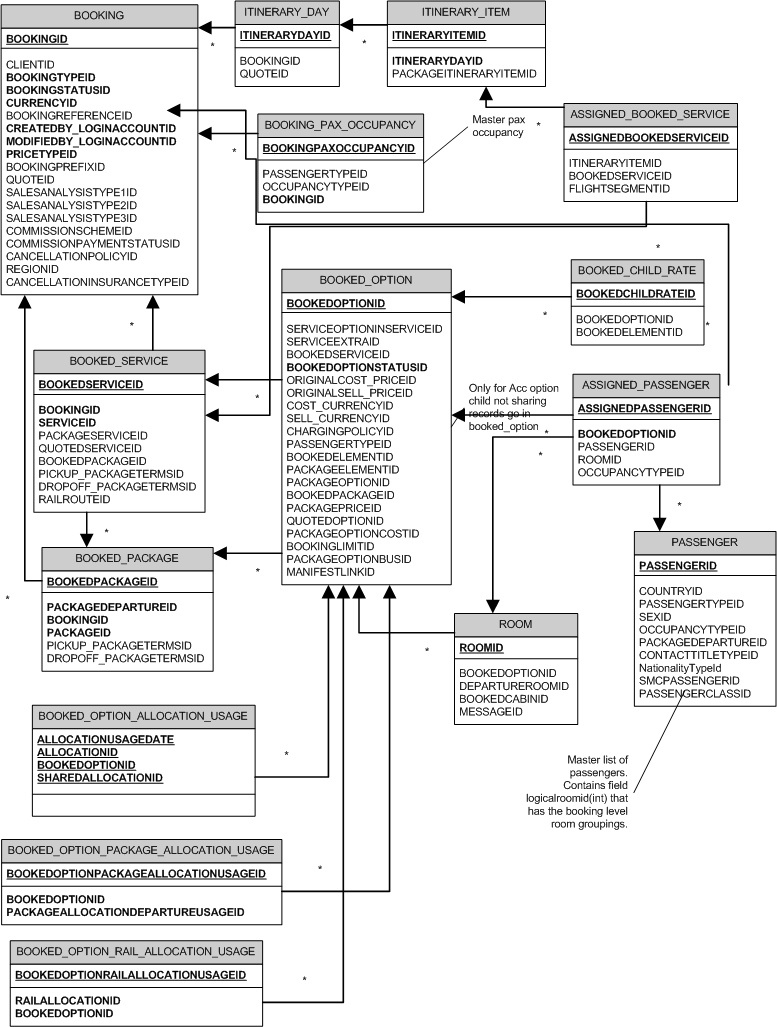


|  |  |
| --- | --- |
| **TABLE NAME** | **DESCRIPTION** |
| **Linked\_Package** | This table is w.r.t. Sub Package and stores the details of imported Principal Package.  **PackageID**: Sub-PackageID  **PrincipalPackageID**: Master/Principal PackageID  **Imported\_FromDay**: From which day the Principal Package is imported in Sub-Package  **Imported\_ToDay**: Till which day the Principal Package is imported in Sub-Package  **Imported\_InToDay**: On which day of the Sub-Package the Principal Package is added [eg: 2; means the Principal Package starts on Day-2 of Sub Package]  **ImportOptionals**: Determined if optional options/extras are imported  Example: PackageID = 122; PrincipalPackageid = 51; Imported\_FromDay = 5; Imported\_ToDay = 7; Imported\_InToDay = 2  This implies that; The Principal Package with PackageID= 51 is imported into Sub-Package with PackageID = 122 from Day-5 to Day-7of the Principal Package and starts on Day-2 of Sub-Package |
| **Linked\_Package\_Departure** | This table is w.r.t. Sub Package departure and stores the details of linked Package departure  **PackageDepartureID**: DepartureID of Sub-Package  **PrincipalPackageDepartureID**: DepartureID of the Principal Package for which its linked |
| **Linked\_Package\_Service** | Stores the details Imported Package service.  **ImportedPackageServiceID**: PackageServiceID of the Principal package  **DaysInPackage**: No. of days for which the Service is added in Sub-Package |

# Rail Specific Table structures



# Booking related tables in TS



|  |  |
| --- | --- |
| **TABLE NAME** | **DESCRIPTION** |
| **Booking** | Stores details on Bookings. Details like the Booking Name, Booking Reference Number etc are stored |
| **Booked\_Service** | Stores the details of Booked Service. Details like BookedServiceID, BookedServiceTime, etc are stored |
| **Itinerary\_Day** | Stores details for the Itinerary Day, Itinerary Day Date and Itinearary Day Description, |
| **Itinerary\_Item** | Stores details for the Itineary Item, like ID, From Time, TO Time, etc |
| **Booking\_Pax\_Occupancy** | Stores details for the Booking Pax occupancy like ID, Passenger No, Passenger Total Only, etc |
| **Assigned\_Book\_Service** | Stores details like Itinerary Item Id, Book Service ID, Assigned Booked Service Description, etc |
| **Booked\_Option** | Stores details for the Booked Option ID, Booked Option Cost amount, etc |
| **Booked\_Package** | Stores details for the Booked Package, like Package ID, Package Time of Service, Duration of Service, etc. |
| **Booked\_Option\_Allocation\_Usage** | Stores the links between Booked Option and Allocation |
| **Booked\_Option\_Package\_Allocation** | Stores the links between Booked Option Package Allocation and Booked Option |
| **Booked\_Option\_Rail\_Allocation** | Stores details of the Booked\_Option\_Rail\_Allocation. |
| **Booked\_Child\_Rate** | Stores details on the Booked Child Rates. Details like the Child rate cost amount, child rate sell amount, etc |
| **Assigned\_Passenger** | Stores details on the assigned Passenger ID and BookedOptionID, RoomID and PassengerID |
| **Passenger** | Stores details of Passenger Info. Details like First Name, Last name, Age, Date of Birth, etc are stored. |
| **Room** | Stores details of Rooms. Details of RoomId, RoomOperatorID, RoomRoomNumber, BookedOptionID, etc are stored. |

**Booked Option Records**

Booked Option stores Cost and Sell records for each option or extra booked for a service or a package.

It can have multiple rows for each option/extra based on split prices.

Consider an Option has prices defined as follows in service maintenance

|  |  |
| --- | --- |
| 1-5 Apr | 100 |
| 6-8 Apr | 90 |
| 9-30 Apr | 110 |

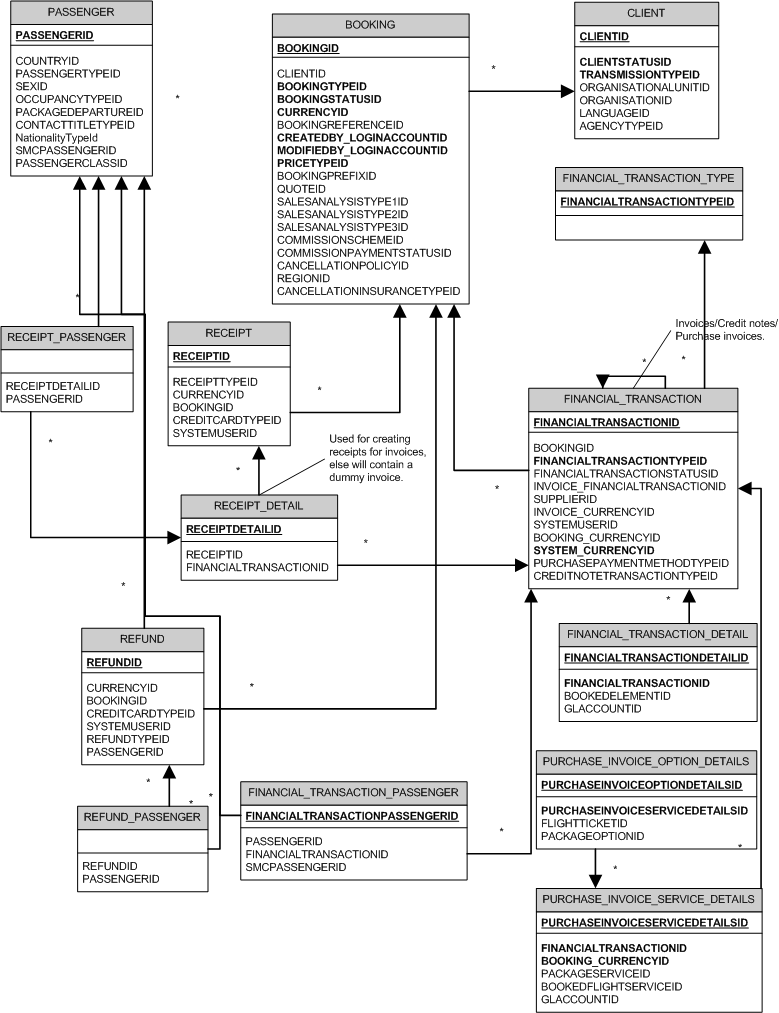
Booking are created as mentioned below. SplitPrice column stores the link between each split. It is incremented for each booked option and all related booked option records will get the same SplitPrice.

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Booking Date** | **Records in Booked\_Option** |
| 1 | 2-5 Apr | Only 1 record created as price is same for all dates booked. SplitPrice will be 0. |
| 2 | 2-7Apr | 2 Records created in Booked\_Option. Both records will have same SplitPrice  2-5 with price 100 per day and  6-7 Apr with price as 90 per day |
| 3 | 3 – 12 Apr | 3 records created, All 3 records will have same SplitPrice  3-5 Apr with price 100 per day,  6-8 with price 90 per day,  9-12 with price 110 per day |

Splits can also occur on fields other than Price like meal plan, child prices when there is change in these fields etc. E.g. if price is same for all days but meal plan is different for 2 sets of booked dates then it will create 2 split records for the change in meal plan.

****

# Booking Finance related tables in TS



**NOTE**: Financial data that is exported to external Financial accounting systems are present in special tables that are specific to the accounting system.

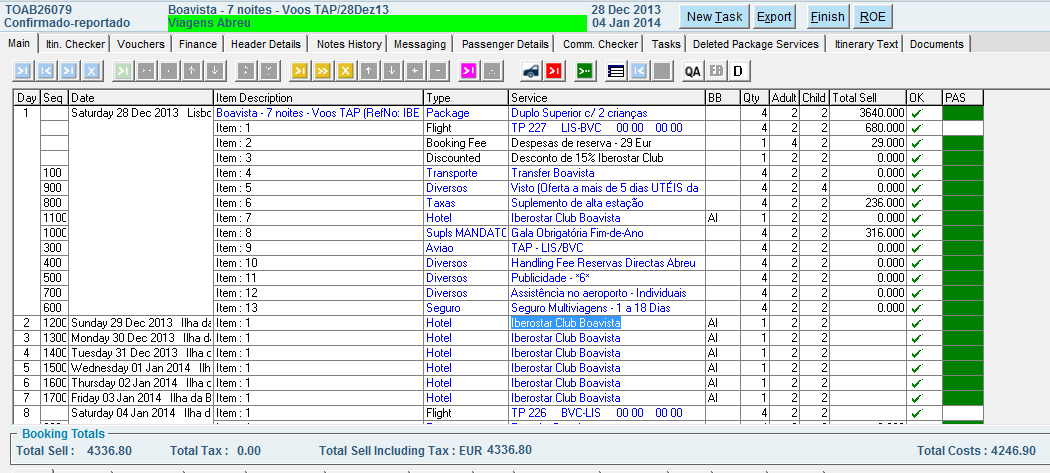
**EXPLANATIONS on Booked Option Table**

In this explanation I have explained about Cost only. Same thing is applicable for Sell Columns.

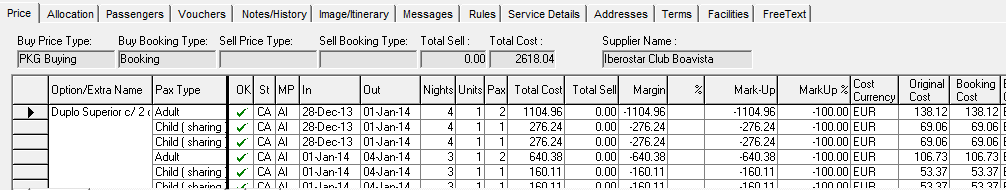
From this Above Query I think you need data for Supplier Purchase Invoice.

Where you need Total Cost of all the Booked Services in a Package Departure Booking. These values you want in Supplier Currency.

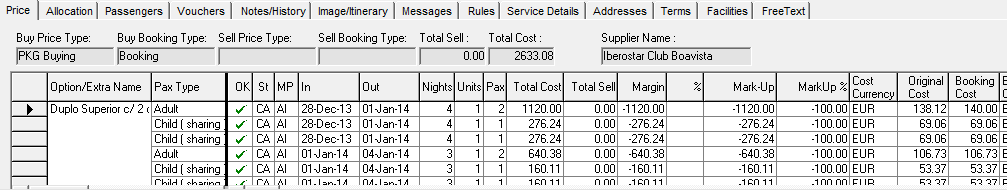
Consider booking “TOAB26079” Before amendments (I am taking example with amendment of Cost to Explain difference between Booking Cost and Original Booking Cost)



Below are the Price Details for One of the service “Iberostar Club Boavista”



After Amendment of Booking Cost of Adult from 138.12 to 140.



Original Cost and Booking Cost are in Cost Currency i.e Service Currency. Original Cost will never change and it’s the contracted rate, But Booking cost can be changed (you can get discount or any additional fees) or even supplement rule from supplier can change booking cost value.

And based on Booking Cost we calculate Total Cost. i.e. in Booking Currency.

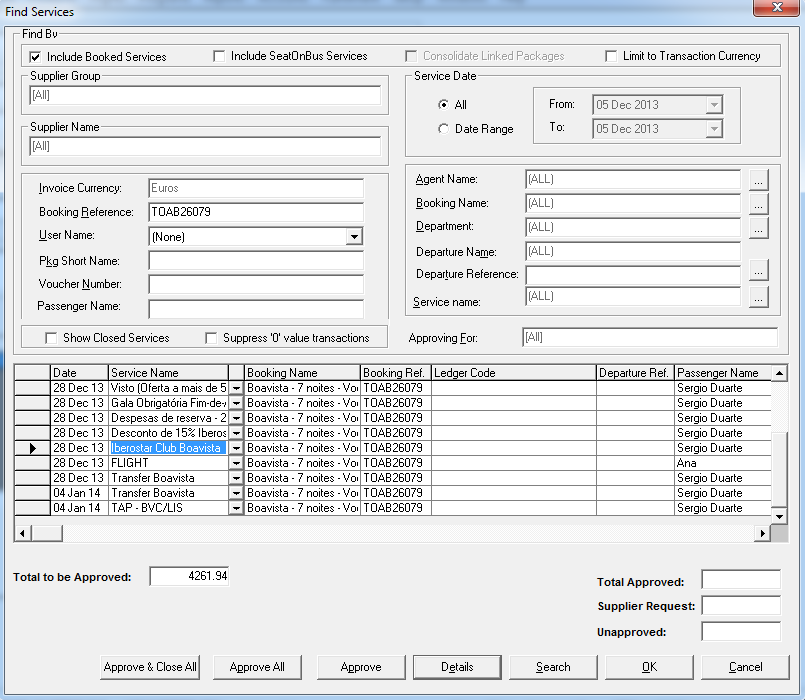
**Total Cost** - It is in Booking Currency.

**Cost Currency** - It is Service Cost Currency.

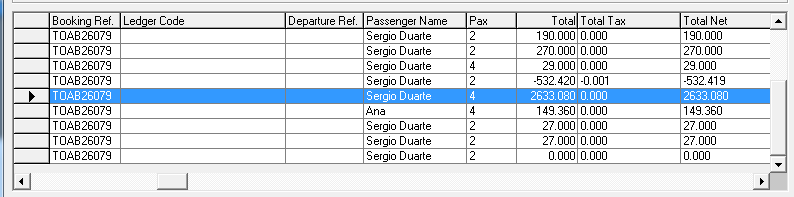
**Original Cost** - This value is in Service Currency (Cost Currency). This value **cannot** be changed.

**Booking Cost** - This value is also in Service Currency (Cost Currency) and can be changed

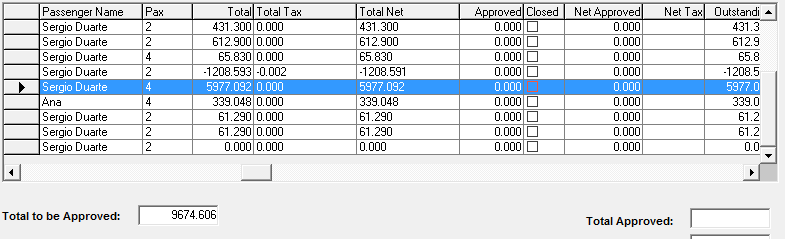
So while doing calculations (for Purchase invoice) of cost we need to refer actual Cost, which is Booking Cost and Total Cost.



This is in EUR



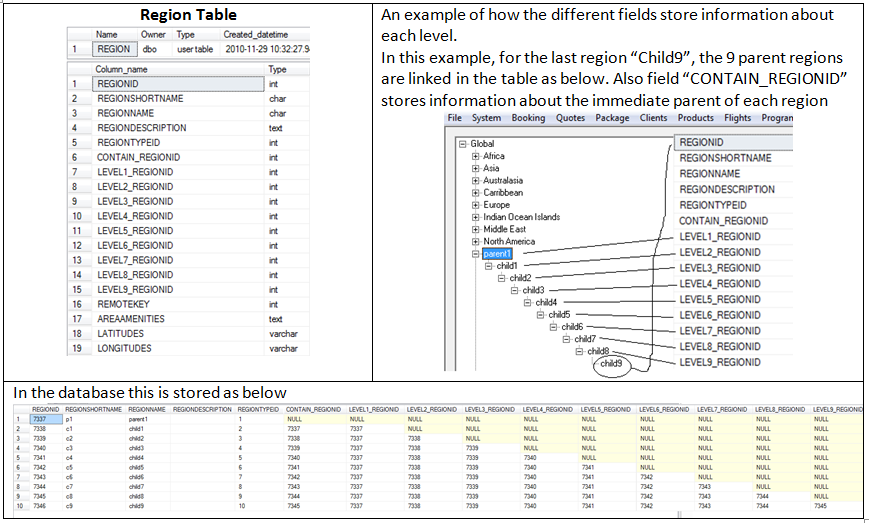
This is in TND



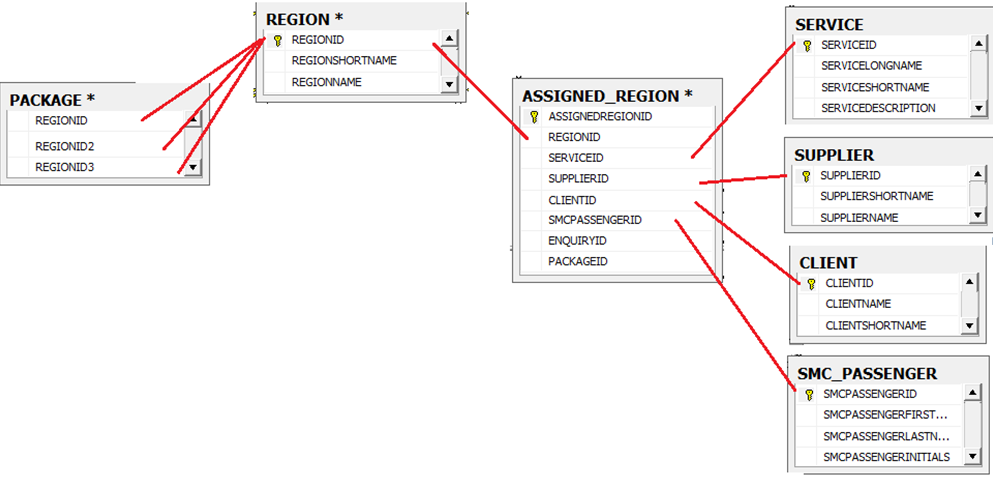
Note:- If there is no valid Exchange rate from Booking Currency to Service Currency then Zero (0.00) Prices will be shown.

# Region Table and its Linked Tables

The use of region table and how parent and child regions are stored in DB is explained below



**Basic Region Table Links**



The full database diagram to know all the links where the “Region” table is linked is given below.

