Carmen Unified Data Model, version 3.2.0beta sk

Jeppesen Systems <info@jeppesen.com>

Version 3.2.0beta sk

15th March 2007

Abstract

This document presents the Carmen Unified Data Model. The goal is to provide a common data model used internally by all Carmen products, allowing them to operate on information stored in one and the same database. The model should support the crew management process from manpower planning to post operations calculations.

Contents

1 Status

TBD: More version history to be added.

Version 1.2.4: Added appendix with crew documents.

Version 1.2.3: Fixed bug in titlerank reference.

Version 1.2.2: Added air_manpower module. Added rave module. Changed naming conventions so that all entities that are enums by nature have the suffix _set. Updated air_crew.

The air_rail module was removed in version 1.2.1.

The detailed crew module was extended in version 1.2.1.

The detailed aircraft module is not yet implemented.

The planning module is new. Cyclic planning needs are not yet covered, it is probably sufficient to add a couple of offset fields.

The tracking module is new and has not been reviewed yet.

The IOCS module is new.

This version includes a rail "minor mode". The rail "major mode" is currently a separate module.

The passenger reservation module is not yet implemented.

There should be an air_oag module to hold the OAG in compressed form.

Added aircraft_connection as a new way to describe aircraft rotations. Aircraft rotations will probably be phased out.

Added air_airport with definitions of airport events and slots.

2 Issues

The railway module must be developed and brought into sync with the airline module. The railway module should use terms appropriate to the railway business. The modules will have names beginning with rail_.

It must be possible to selectively publish crew rosters and to control the visibility of rosters to individual crew. Preferably this should be based on the history handling capability of Dave and avoid unnecessary duplication of data.

The infrastructure model (airports) needs more work. There are too many country fields.

The use of e-tables should be explained.

The air_tracking module uses variant link entities to implement polymorphism. This does not fit well in the relational model and should be reworked to follow the style used in air_core.

The following entities previously contained udor as part of the primary key: ground_task, trip, rotation. udor has been removed from the primary keys because the id itself is unique. The reason for having udor as well was to provide ordering by date thus speeding up typical

queries. This should instead be handled by extra indexes. udor was completely removed from ground_task, since it is redundant.

We should perhaps distinguish between position and duty type in the link objects.

We should perhaps distinguish between loose and tight foreign keys. Tight foreign keys are those that should be respected all of the time, this includes references to entities that act as enum definitions. Loose foreign keys are references to other scheduling entities that may occasionally be out of order.

Should UDM and Dave support float type?

Should UDM and Dave support second level resolution of times?

3 Scope and Requirements

The goal of UDM is to provide a small core data model, that is relevant in most or all phases of operations.

The data model should be extensible, it should be possible to add phase-specific data or customer-specific data in a straight-forward manner.

It must be possible to use RAVE to access all UDM data, both core data and custom data.

It must be easy to load a trip, with all its components.

It must be easy to load all trips (with all their components), within a time period and with other constraints.

It must be easy to load a crew roster, with all its components.

It must be easy to load a published crew roster, with all its components.

It must be easy to load all crew rosters (with all their components), within a time period and with other constraints.

It should be easy to load aircraft rotations.

It should use Dave's capabilities for handling historic data.

4 Entities and Extensions

The UDM schema is defined in terms of entities and extensions. These are then grouped into modules. The modules allow tailoring systems according to different needs by selectively implementing only some of the UDM features. The basic UDM schema can be extended to cover specific customer needs.

Entities are roughly version controlled database tables and define the persistent storage items. Each entity has a primary key, possibly extra fields, and extra indexes.

Extensions are additional fields that are added to an entity. They allows supplementary modules to enhance already existing entities.

Each entity must be read as the basic entity definition and all extensions to that entity.

5 Data types

5.1 Basic types

These are the intrinsic data types.

5.1.1 int

Int is stored in the database as a number capable of holding at least the range of a signed 32 bit integer. Int corresponds to the Rave type int.

5.1.2 date

Date represents a UTC date, it is stored in the database as the number of days since the Carmen epoch, Jan 1 1986. Dates before Jan 1 1986 are negative integers.

5.1.3 time

Time represents an absolute point in time. It is stored in the database as the number of minutes since the Carmen epoch, Jan 1 1986 00:00 UTC. Times before the epoch are negative integers. Time corresponds to the Rave type abstime.

5.1.4 reltime

Reltime represents a time interval. It is stored in the database as an integral number of minutes. Reltime corresponds to the Rave type reltime.

5.1.5 bool

Bool represents a boolean value. It is stored in the database as a character, Y for True and N for false. Bool corresponds to the Rave type bool.

5.1.6 char

Char represents a single character. Unlike strings, the character set for single characters is not well defined and char values should be limited to the ASCII character set. The NUL character is not allowed because some databases cannot distinguish between the character NUL and a NULL value.

5.1.7 string

Strings are stored in the database as variable length strings encoded in the UTF-8 character set. Strings can contain the full set of Unicode characters. The Dave API is however currently restricted to ISO 8859-1. Empty strings are not allowed because some databases cannot distinguish between an empty string and a NULL value. String corresponds to the Rave type string.

5.1.8 uuid

The uuid type represents a Universally Unique IDentifier as specified by RFC 4122. A UUID is a 128 bit number that is guaranteed to be universally unique, and can therefore safely be used as an artificial key. A UUID is stored in the database as a 24 character string by base64 (RFC 1421) encoding the 16 octets of the UUID. The resulting string contains only ASCII characters and can safely be stored regardless of the character set in use.

5.2 Reference types

A reference type is the type of entity referred to by a foreign key. In the schema a foreign key is listed as one item, although it in fact may consist of several components. The type of the key component(s) may be found by looking at the definition of the referred entity.

5.3 Modifiers

5.3.1 size

The optional size modifier limits the length of a string. Without the size modifier the maximum length of a string is implementation dependent, limited by the database. Size can only be used with the string type.

5.3.2 arraylength

The optional arraylength modifier declares the field to be an array. Arraylength can be used together with all data types, both basic and reference types.

6 Schema

6.1 Data Model Outline

The data model consists roughly of three parts, see Figure 1. First there is the basic need consisting of flight_legs and ground_tasks. The scheduling problem consists of covering the

needs of these entities with crew and aircraft. The resource needs are not modeled explicitly, as it is assumed that they will be calculated by Rave rules. However, it is quite possible to extend the entities to explicitly store the resource needs if so desired.

Then there is the crew part. This contains the crew entity as well as a number of entities linking crew with various tasks. Both anonymous and individual assignments are represented.

Finally there is the aircraft part. This contains the aircraft entity as well as a number of entities linking aircraft with various tasks. Both anonymous and individual assignments are represented.

Many entity fields can be null. This is on purpose, acknowledging the fact that all information is not known at all planning stages. In principle keys must be non-null and scheduled times and places should also be non-null. Most entities contain a supplementary information field. This allows free text comments to be attached to the items, serving as notes to the schedulers.

The ambition is to model crew and aircraft assignments in a similar way. Furthermore in order to not force all systems to implement all of the data model, UDM is divided into a number of modules. The base is provided by the module air_core. It contains all of the important entities and allows representation of crew and aircraft assignments. The module contains only minimal crew and aircraft entities.

The module air_crew contains an extended crew model for systems that need to reason about crew.

The module air_aicraft contains an extended aircraft model for systems that need to reason about aircraft.

The module air_planning contains extensions to support the planning process.

The module air_tracking contains extensions to support the tracking process.

The module air_rail contains extensions to support some railway specific functionality. A complete railway model is under development but is not part of this UDM version.

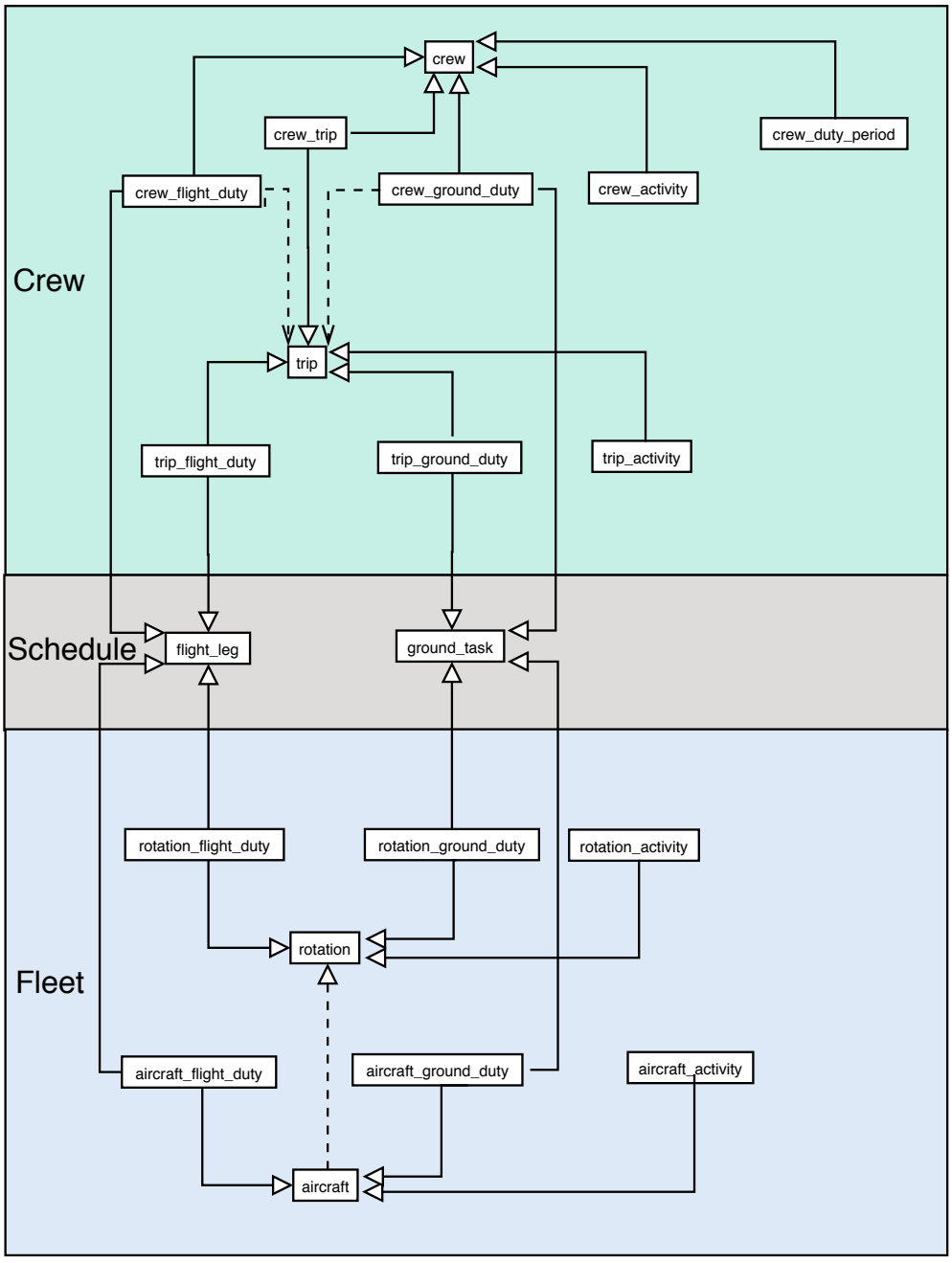
6.2 Naming conventions

Entity names have been chosen to be descriptive. Entities that are links between other entities have composite names showing the basic entities that are linked together. Naming parts are separated by underscore. Keeping in mind that some databases have low limits on maximum name lengths, entity names should be no longer than about 20 characters.

Field names have been kept short and do not contain underscores. This makes it easier to recognize composite foreign keys and arrays, and avoids hitting the maximum name length.

There are two situations where the field names defined in this document are mapped to different names that must be used in the Dave API. One case is where a field is a reference to an entity identified by a composite primary key. Assume the field *f* in entity E1 references the entity E2. Assume also that E2 has a primary key consisting of fields *k1* and *k2*. Then E1 will contain the two fields *f_k1* and *f_k2*. The other case where renaming happens is when arrays are defined. Declaring a field *g* as an array with length 2 will result in the fields *g_1* and *g_2*. Finally, it is possible to combine arrays and foreign keys. If the field *f* above was declared as an array of

UDM1.1


Crew
crew
crew_trip
crew_duty_period
crew_flight_duty
crew_ground_duty
crew_activity
trip
trip_flight_duty
trip_ground_duty
trip_activity
Schedule
flight_leg
ground_task
Fleet
rotation_flight_duty
rotation_ground_duty
rotation_activity
rotation
aircraft_flight_duty
aircraft_ground_duty
aircraft_activity
aircraft

length 2, the actual fields would be: *f_1_k1*, *f_1_k2*, *f_2_k1*, *f_2_k2*.

NB: Contrary to the explanation above, field name mapping currently occurs for all keys, not just composite keys.

6.3 Legend to the pictures

An entitity is shown as a solid box. The name above is the name of the entity. The top part of the box shows the primary key fields. The bottom part shows the non-primary-key fields. On the left side are the names of the fields, and on the right side are the types. The types can be either basic types or the names of entities. In the latter case the field is foreign key and the symbol **FK** is shown beside the the box. A boldface type name shows that the field is declared non-null. All primary key fields are non-null.

An extension is shown as a box with a dashed top line, the name above is the name of the entity it extends. The presentation is similar to an entity except that an extension can only contain non-primary-key fields.

A number within parenthesis after the field type denotes the maximum length of a string valued field. A number within brackets after the field type indicates that the field is an array of values.

6.4 Module air_aircraft

The basic aircraft entity is defined in the air_core module. The aircraft entity defined therein is only the bare minimum needed to allow references to individual aircraft. This module, air_aircraft, extends the aircraft entity with more information for systems that need to work with aircraft.

6.4.1 Entity ils_category_set

This entity lists defined ILS categories. They are: I, II, IIIa, IIIb, IIIc or N if no ILS capability.

ils_category_set		ILS categories
id	***string(10)***	ILS category
si	*string*	Supplementary information

6.4.2 Entity aircraft_opdef

This entity lists operational deficiencies for individual aircraft.

aircraft_opdef			Aircraft OpDef
ac	***aircraft***	**FK**	Aircraft identifier
opdef	***opdef_set***	**FK**	OpDef type identifier
dt	*time*		Due time
si	*string*		Supplementary information

6.4.3 Entity opdef_set

This entity describes operational deficiency types.

opdef_set		Operational Deficiencies
id	***string(20)***	OpDef identifier
si	*string*	Supplementary information

6.4.4 Extensions

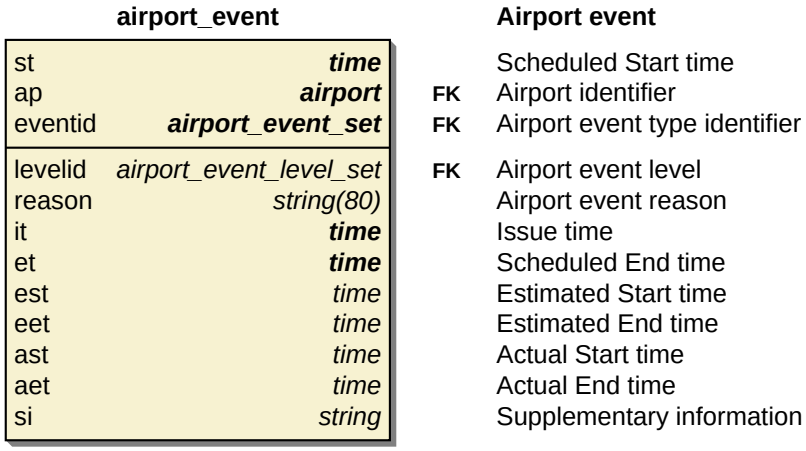
This module also extends the aircraft (6.6.33.1) and aircraft_type (6.6.39.1) entities.

6.5 Module air_airport

The basic airport entity is defined in the air_core module. This module, air_airport, extends the airport entity with more information for systems that need to work with airports.

6.5.1 Entity airport_event

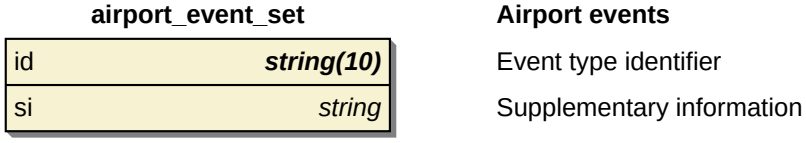
This entity lists airport events (closures, ground delay programs, etc.)



airport_event			Airport event
st	***time***		Scheduled Start time
ap	***airport***	**FK**	Airport identifier
eventid	***airport_event_set***	**FK**	Airport event type identifier
levelid	*airport_event_level_set*	**FK**	Airport event level
reason	*string(80)*		Airport event reason
it	***time***		Issue time
et	***time***		Scheduled End time
est	*time*		Estimated Start time
eet	*time*		Estimated End time
ast	*time*		Actual Start time
aet	*time*		Actual End time
si	*string*		Supplementary information

6.5.2 Entity airport_event_set

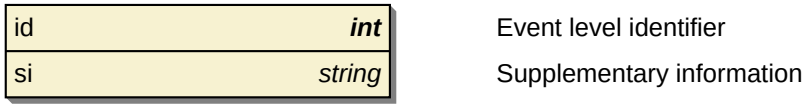
This entity lists allowed airport event types. See *airport_event*.



airport_event_set		Airport events
id	***string(10)***	Event type identifier
si	*string*	Supplementary information

6.5.3 Entity airport_event_level_set

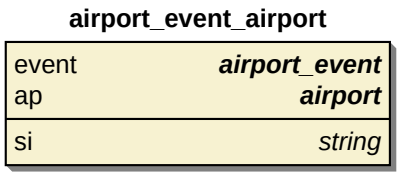
This entity lists allowed airport event levels. See *airport_event*. The event level indicates if and how much the event will spread to other airports.



airport_event_level_set		Airport event levels
id	***int***	Event level identifier
si	*string*	Supplementary information

6.5.4 Entity airport_event_airport

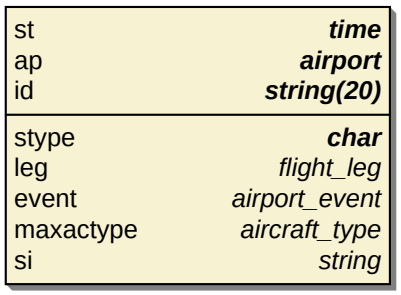
This entity lists airports affected by events at another airport.



airport_event_airport			Airport event - affected airport
event	***airport_event***	**FK**	
ap	***airport***	**FK**	Affected airport identifier
si	*string*		Supplementary information

6.5.5 Entity airport_slot

This entity lists airport slots.



airport_slot			Airport slot
st	***time***		Slot time
ap	***airport***	**FK**	Airport identifier
id	***string(20)***		Slot ID
stype	***char***		Slot type (D=departure, A=arrival)
leg	*flight_leg*	**FK**	Assigned flight leg
event	*airport_event*	**FK**	Related airport event (if any)
maxactype	*aircraft_type*	**FK**	Largest aircraft type allowed
si	*string*		Supplementary information

6.6 Module air_core

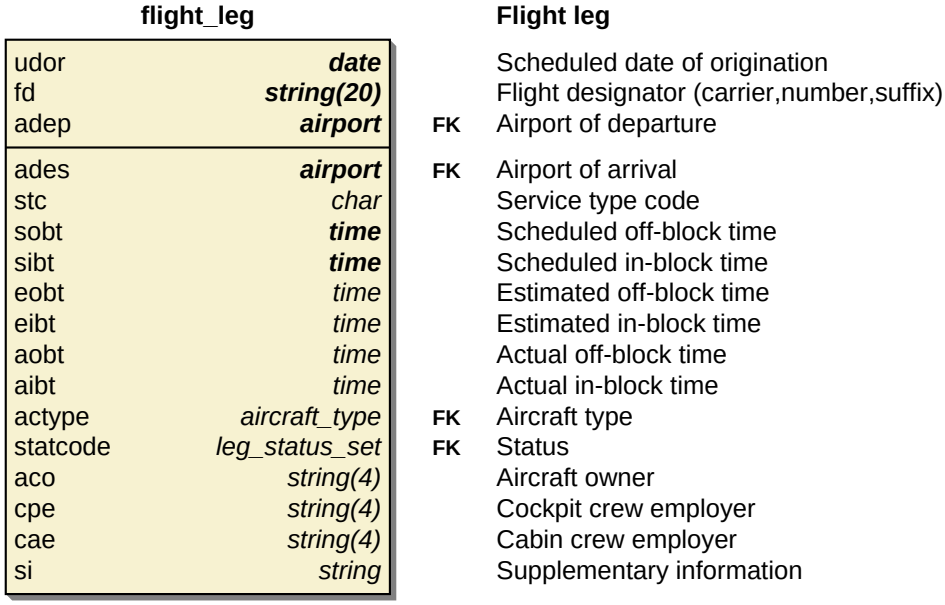
The air_core module contains the minimum necessary data model to support the Carmen CMS products for airlines. All necessary entities are present, but only a limited set of attributes are contained in the core model. Other modules may define extensions to the entities contained herein as well as introduce new entities.

6.6.1 Entity flight_leg

This entity represents a dated flight leg. UDM stores the time table as fully dated flight legs. A flight leg is defined by its scheduled date of origination, flight designator and departure airport. A multileg flight is defined by all legs having the same *udor* and *fd* fields. Notice that *udor* is the scheduled date of flight origination, and that for subsequent legs of a multileg flight the *udor* may differ from the date of operation. For the first leg of a multileg flight, the *udor* normally coincides with the date of the scheduled off block time *sobt*. But if the *sobt* should later be modified care should be taken to not change the *udor* since this would change the key of the leg.

The flight designator *fd* is a string of exactly 10 characters. The first three characters are the carrier code, expressed either as the three letter code, or as the two character code followed by a space. If possible use the two character code. The next six characters of the flight designator are the flight number, right justified and left filled with zeros. The last character of the flight designator is the operational suffix, or space if none is present.

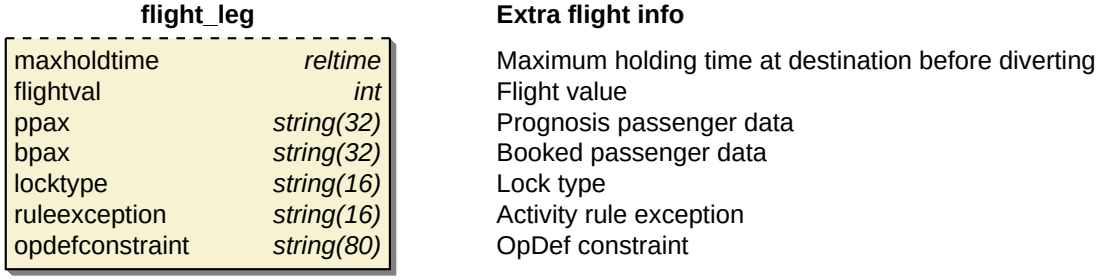
The departure and arrival airport fields should contain the IATA codes.



flight_leg			Flight leg
udor	***date***		Scheduled date of origination
fd	***string(20)***		Flight designator (carrier,number,suffix)
adep	***airport***	**FK**	Airport of departure
ades	***airport***	**FK**	Airport of arrival
stc	*char*		Service type code
sobt	***time***		Scheduled off-block time
sibt	***time***		Scheduled in-block time
eobt	*time*		Estimated off-block time
eibt	*time*		Estimated in-block time
aobt	*time*		Actual off-block time
aibt	*time*		Actual in-block time
actype	*aircraft_type*	**FK**	Aircraft type
statcode	*leg_status_set*	**FK**	Status
aco	*string(4)*		Aircraft owner
cpe	*string(4)*		Cockpit crew employer
cae	*string(4)*		Cabin crew employer
si	*string*		Supplementary information

6.6.1.1 Extension from module air_iocs

Extra flight leg info needed by IOCS.



flight_leg		Extra flight info
maxholdtime	*reltime*	Maximum holding time at destination before diverting
flightval	*int*	Flight value
ppax	*string(32)*	Prognosis passenger data
bpax	*string(32)*	Booked passenger data
locktype	*string(16)*	Lock type
ruleexception	*string(16)*	Activity rule exception
opdefconstraint	*string(80)*	OpDef constraint

6.6.1.2 Extension from module air_planning

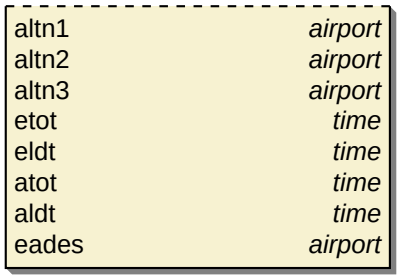
Some programs make explicit reference to the sequence number within multileg flights. As a convenience it is added here as a flight_leg field.

Planning attributes

Leg sequence number

6.6.1.3 Extension from module air_tracking

Tracking extensions for flight_leg.



Flight times

FK	First alternate airport
FK	Second alternate airport
FK	Third alternate airport
	Estimated take-off time
	Estimated landing time
	Actual take-off time
	Actual landing time
FK	Estimated/actual arrival airport (for diversions)

6.6.1.4 Extension from module sas_air_core

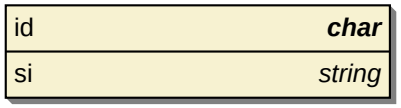
Additional SAS specific information about a flight leg.

Extension to flight_leg

Aircraft version

6.6.2 Entity leg_status_set

This entity lists allowed leg status codes. Basic codes are: S = Scheduled, D = Departed, A = Arrived, C = Cancelled, R = Returned to ramp, I = Diverted.



Leg statuses

Status code

Description

6.6.3 Entity adhoc_flight

This entity represents ad-hoc flights. This is used when processing SSM and ASM schedule update messages. The udor and fd fields follow the same conventions as in *flight_leg*.

adhoc_flight		Adhoc flight
udor	***date***	Scheduled date of origination
fd	***string(20)***	Flight designator (carrier,number,suffix)
si	*string*	Supplementary information

6.6.4 Entity ground_task

This entity represents non-flying tasks that need crew and/or aircraft. In contrast, entities containing *activity* in the name are activities that are inserted to make a roster, trip or rotation complete and legal.

ground_task			Ground task
udor	***date***		Date of ground task
id	***uuid***		Unique ground task identifier
st	***time***		Start time
et	***time***		End time
adep	***airport***	**FK**	Airport of departure
ades	***airport***	**FK**	Airport of destination
activity	***activity_set***	**FK**	Activity identifier
si	*string*		Supplementary information

6.6.5 Entity airport

This entity represents an airport. The identity of the airport is the IATA code.

tz is the number of minutes to add or subtract from UTC to obtain local winter time. If *dst* is non-null it identifies a DST rule describing the local use of daylight savings time.

airport			Airport
id	***string(10)***		Airport identifier
name	*string(30)*		Name
city	*city*	**FK**	City identifier
state	*state*	**FK**	State identifier
country	***country***	**FK**	Country identifier
tz	***int***		Minutes offset from UTC
dst	*dst*	**FK**	DST rule identifier
latitude	***int***		Latitude in arc minutes (-S +N)
longitude	***int***		Longitude in arc minutes (-W +E)
si	*string*		Supplementary information

6.6.6 Entity city

This entity represents a city.

city			City
id	***string(10)***		City identifier
name	***string(30)***		Name
state	*state*	**FK**	State identifier
country	***country***	**FK**	Country identifier
si	*string*		Supplementary information

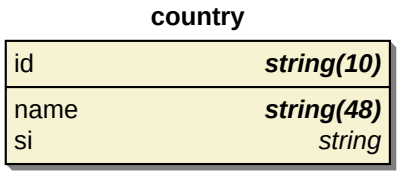
6.6.7 Entity state

This entity represents a state or province within a country.

state			State
id	***string(10)***		State identifier
country	***country***	**FK**	Country identifier
name	***string(30)***		Name
si	*string*		Supplementary information

6.6.8 Entity country

This entity represents a country.



Country

Country identifier

Name
Supplementary information

6.6.8.1 Extension from module sas_paxlst

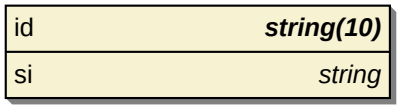
This entity is an extension of the UDM country table and is used to convert country codes into the ISO 3166 standard.

Country conversion code

3-letter Country Code
Digit Country Code

6.6.9 Entity dst

This entity represents a daylight savings time rule identifier.



DST rule identifier

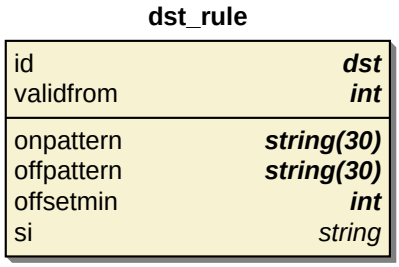
DST rule identifier

Supplementary information

6.6.10 Entity dst_rule

This entity represents a rule describing the beginning and end of daylight savings time. *offset-min* is the number of minutes to add to local winter time to obtain local time when DST is in effect.

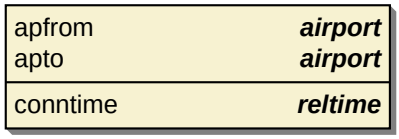
validfrom specifies the first year the rule is applicable. Only one rule with a given name can be valid at a time, the last applicable rule supersedes all previous rules.



	DST rule
FK	DST rule identifier
	Valid from year
	On pattern
	Off pattern
	Minutes offset when DST in effect
	Supplementary information

6.6.11 Entity airport_transfer_time

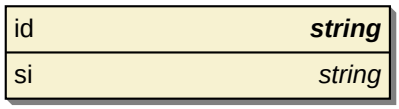
This entity lists known transfer times between close airports.



	Transfer times between airports
FK	From airport
FK	To airport
	Transfer time in minutes

6.6.12 Entity crew_base_set

This entity lists supported crew bases.



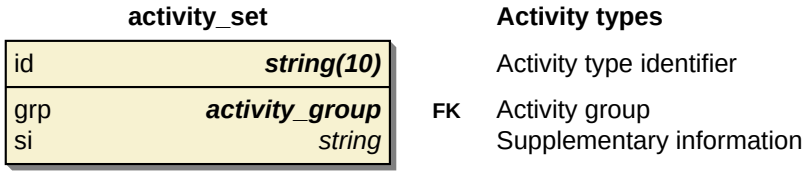
	Crew bases
	Base, use IATA city code if applicable
	Supplementary information

6.6.12.1 Extension from module sas_air_core

	County information added
FK	Matching country

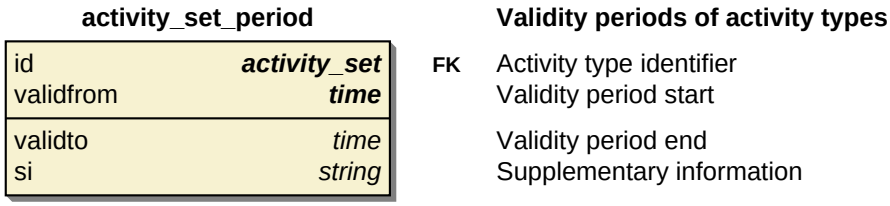
6.6.13 Entity activity_set

This entity lists allowed activity types. See *ground_task*, *trip_activity* and *crew_activity*.



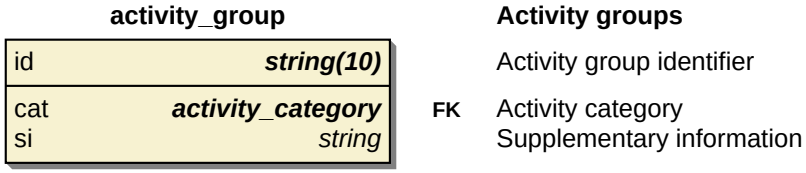
6.6.14 Entity activity_set_period

This entity lists validity periods of activity types. Validity periods of activity types define when the corresponding activities are available for operation.



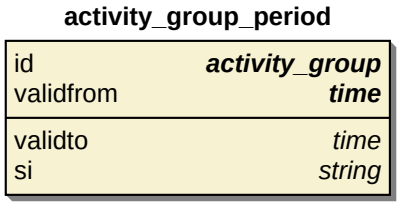
6.6.15 Entity activity_group

This entity lists allowed activity groups. Activity type attributes should primarily be defined on group level (as extensions to *activity_group_period*).



6.6.16 Entity activity_group_period

This entity lists validity periods of activity groups. Validity periods of activity groups are used when group attributes change over time.

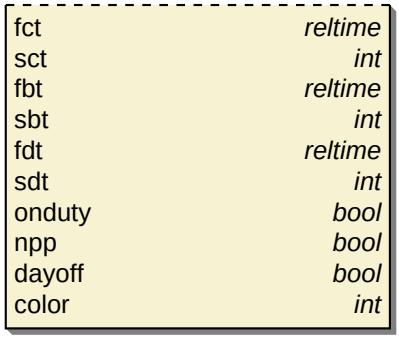


Validity periods of activity groups

id	FK	Activity group identifier
validfrom		Validity period start
validto		Validity period end
si		Supplementary information

6.6.16.1 Extension from module air_crew

Activity group attributes.

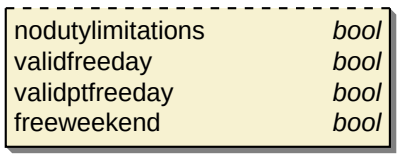


Activity group attributes

fct	Fixed credit time
sct	Supplementary credit time (%)
fbt	Fixed block time
sbt	Supplementary block time (%)
fdt	Fixed duty time
sdt	Supplementary duty time (%)
onduty	On duty flag
npp	Non-plannable period
dayoff	Day off flag
color	Activity color

6.6.16.2 Extension from module sas_air_core

Additional SAS specific information about activity codes

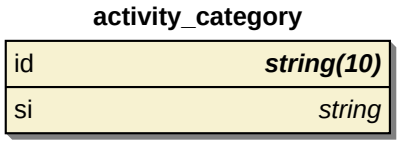


Attributes on activity groups

nodutylimitations	No duty time contribution
validfreeday	Valid freedays
validptfreeday	Valid part-time freedays
freeweekend	Activity allowed on free weekend

6.6.17 Entity activity_category

This entity lists allowed activity categories.

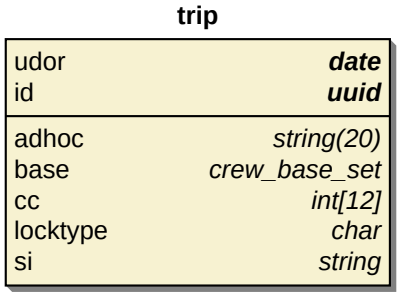


activity_category		Activity categories
id	***string(10)***	Activity category identifier
si	*string*	Supplementary information

6.6.18 Entity trip

This entity represents a crew trip assignable to one or several crew members.

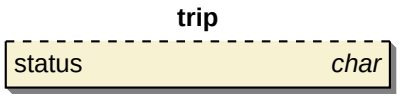
The adhoc field can be used for storing an alternate trip ID from an external system.



trip			Anonymous crew trip
udor	***date***		Scheduled trip origination date
id	***uuid***		Unique trip id
adhoc	*string(20)*		Ad-hoc trip identifier
base	*crew_base_set*	FK	Crew base
cc	*int[12]*		Crew complement
locktype	*char*		Lock indicator
si	*string*		Supplementary information

6.6.18.1 Extension from module air_iocs

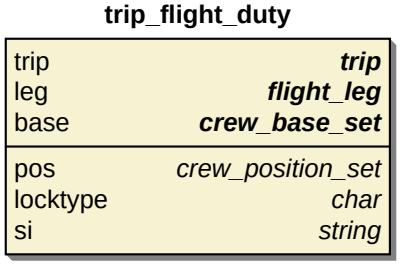
Extra trip info needed by IOCS.



trip		Trip status code
status	*char*	

6.6.19 Entity trip_flight_duty

This entity links a trip with a flight leg.



trip_flight_duty			Trip flight duty assignment
trip	***trip***	FK	Trip identifier
leg	***flight_leg***	FK	Flight leg identifier
base	***crew_base_set***	FK	Base variant
pos	*crew_position_set*	FK	Position identifier
locktype	*char*		Lock indicator
si	*string*		Supplementary information

6.6.19.1 Extension from module air_tracking

Tracking extensions for trip_flight_duty.

trip_flight_duty		Trip flight duty extension
bookref	*string(50)*	Booking reference

6.6.19.2 Extension from module air_urm

URM extension for trip_flight_duty.

trip_flight_duty			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

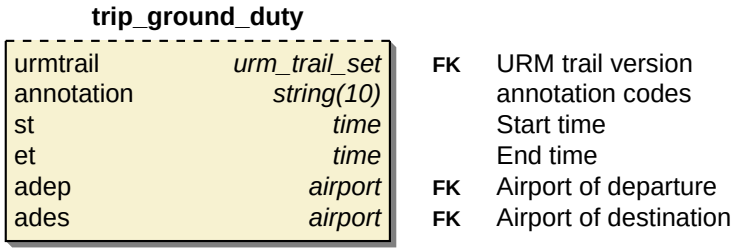
6.6.20 Entity trip_ground_duty

This entity links a trip with a ground task.

trip_ground_duty			Trip ground duty assignment
trip	***trip***	**FK**	Trip identifier
task	***ground_task***	**FK**	Ground task identifier
base	***crew_base_set***	**FK**	Base variant
pos	*crew_position_set*	**FK**	Position identifier
locktype	*char*		Lock indicator
si	*string*		Supplementary information

6.6.20.1 Extension from module air_urm

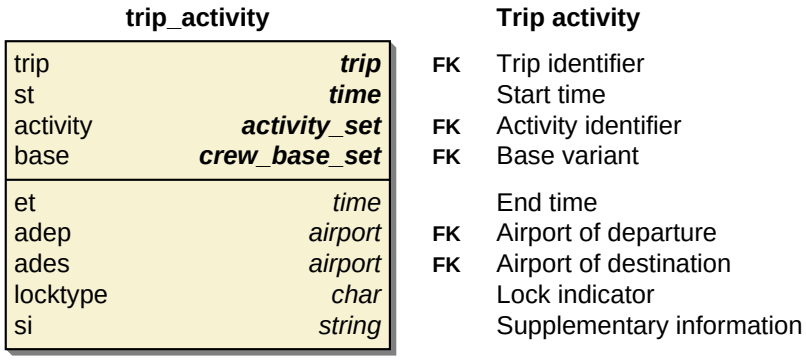
URM extension for trip_ground_duty.



trip_ground_duty			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.21 Entity trip_activity

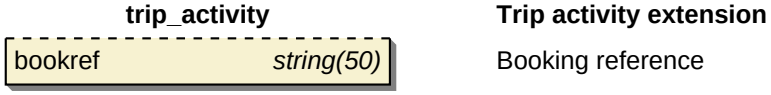
This entity links a trip with a non-flying activity. See also *ground_task*.



trip_activity			Trip activity
trip	***trip***	**FK**	Trip identifier
st	***time***		Start time
activity	***activity_set***	**FK**	Activity identifier
base	***crew_base_set***	**FK**	Base variant
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination
locktype	*char*		Lock indicator
si	*string*		Supplementary information

6.6.21.1 Extension from module air_tracking

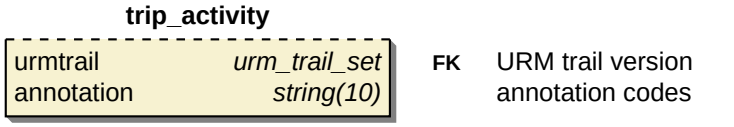
Tracking extensions for trip_activity.



trip_activity		Trip activity extension
bookref	*string(50)*	Booking reference

6.6.21.2 Extension from module air_urm

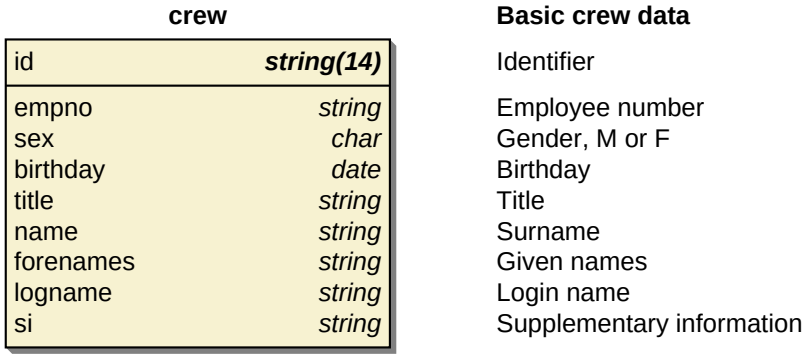
URM extension for trip_activity.



trip_activity			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes

6.6.22 Entity crew

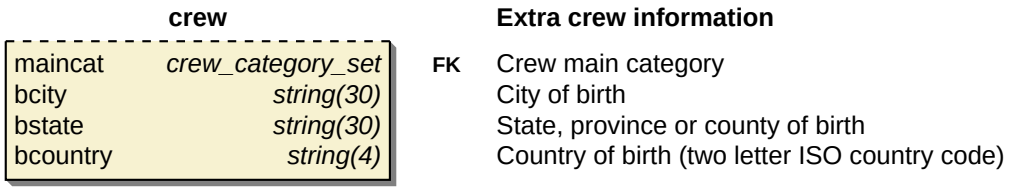
This entity represents a crew member. The entity can be extended to cover all crew information that changes infrequently and where future updates will not affect the planning process.



6.6.22.1 Extension from module air_crew

This extension contains additional information for a crew member, including main category and place-of-birth information.

NB: the *maincat* field is provided for convenience (it should match the information in crew_employment -> crewrank -> maincat).



6.6.22.2 Extension from module air_planning

The *alias* field is a legacy need. It should probably be dropped in favor of *logname* in air_core.

6.6.22.3 Extension from module sas_air_crew

crew		Ext. to crew
employmentdate	*date*	Date of employment
retirementdate	*date*	Date of retirement

6.6.23 Entity crew_flight_duty

This entity links a crew member with a flight leg. If the flight duty was assigned from a trip, the *trip* field allows identification of the trip from whence it came.

crew_flight_duty			Crew flight duty assignment
leg	***flight_leg***	**FK**	Flight leg identifier
crew	***crew***	**FK**	Crew identifier
pos	*crew_position_set*	**FK**	Position identifier
trip	*trip*	**FK**	Trip identifier
locktype	*char*		Lock indicator
si	*string*		Supplementary information

6.6.23.1 Extension from module air_planning

The *personaltrip* field supports tying together personal scheduled items into chains.

crew_flight_duty		Planning attributes
personaltrip	*uuid*	ID of a personal trip

6.6.23.2 Extension from module air_tracking

Tracking extensions for crew_flight_duty.

crew_flight_duty		Crew flight duty extension
bookref	*string(50)*	Booking reference

6.6.23.3 Extension from module air_urm

URM extension for crew_flight_duty.

crew_flight_duty

urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.24 Entity crew_ground_duty

This entity links a crew member with a ground task. If the ground task was assigned from a trip, the *trip* field allows identification of the trip from whence it came. See *crew_flight_duty* for a description of urmtrail.

crew_ground_duty — **Crew ground duty assignment**

task	***ground_task***	**FK**	Ground task identifier
crew	***crew***	**FK**	Crew identifier
pos	*crew_position_set*	**FK**	Position identifier
trip	*trip*	**FK**	Trip identifier
locktype	*char*		Lock indicator
si	*string*		Supplementary information

6.6.24.1 Extension from module air_planning

The *personaltrip* field supports tying together personal scheduled items into chains.

crew_ground_duty — **Planning attributes**

personaltrip	*uuid*	ID of a personal trip

6.6.24.2 Extension from module air_urm

URM extension for crew_ground_duty.

crew_ground_duty			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.25 Entity crew_activity

This entity links individual crew members with non-flying activity. A crew activity may be personal or stem from *trip_activity* that has been assigned to a crew member. If the crew activity was assigned from a trip, the *trip* field allows identification of the trip from whence it came. See also *ground_task*.

crew_activity			Crew activity
st	***time***		Start time
crew	***crew***	**FK**	Crew identifier
activity	***activity_set***	**FK**	Activity identifier
et	***time***		End time
adep	***airport***	**FK**	Airport of departure
ades	***airport***	**FK**	Airport of destination
trip	*trip*	**FK**	Trip identifier
locktype	*char*		Lock indicator
si	*string*		Supplementary information

6.6.25.1 Extension from module air_planning

The *personaltrip* field supports tying together personal scheduled items into chains.

crew_activity		Planning attributes
personaltrip	*uuid*	ID of a personal trip

6.6.25.2 Extension from module air_tracking

Tracking extensions for crew_activity.

crew_activity			Crew activity extension
bookref	*string(50)*		Booking reference

6.6.25.3 Extension from module air_urm

URM extension for crew_activity.

crew_activity			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes

6.6.26 Entity crew_trip

This entity links a crew member with an entire trip. All of the activities, flight duties and ground duties of the trip are assigned to the crew member. If the trip is modified it directly affects all crew assigned to the trip.

crew_trip			Crew trip assignment
trip	***trip***	**FK**	Trip identifier
crew	***crew***	**FK**	Crew identifier
base	***crew_base_set***	**FK**	Base variant
pos	*crew_position_set*	**FK**	Position identifier
locktype	*char*		Lock indicator
si	*string*		Supplementary information

6.6.26.1 Extension from module air_urm

URM extension for crew_trip.

crew_trip			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.27 Entity crew_position_set

This entity lists allowed crew positions. See *crew_flight_duty*, *crew_ground_duty*, *crew_trip*, *trip_flight_duty*, *trip_ground_duty*.

crew_position_set			Crew positions
id	***string(10)***		Position type identifier
si	*string*		Supplementary information

6.6.27.1 Extension from module air_manpower

Adds the reference to the crew position type set

crew_position_set			Manpower addition
typ	*crew_position_type_set*	**FK**	The type code for this position

6.6.28 Entity rotation

This entity represents an aircraft rotation assignable to an aircraft.

rotation			Anonymous aircraft rotation
udor	***date***		Scheduled date of origination
id	***uuid***		Unique identifier
adhoc	*string(20)*		Ad-hoc identifier
actype	*aircraft_type*	**FK**	Aircraft type
si	*string*		Supplementary information

6.6.29 Entity rotation_flight_duty

This entity links a rotation with a flight leg.

rotation_flight_duty			Rotation flight duty assignment
leg	***flight_leg***	**FK**	Flight leg identifier
pos	***aircraft_position_set***	**FK**	Position identifier
rot	*rotation*	**FK**	Rotation identifier
si	*string*		Supplementary information

6.6.29.1 Extension from module air_urm

URM extension for rotation_flight_duty.

rotation_flight_duty

urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.30 Entity rotation_ground_duty

This entity links a rotation with a ground task.

rotation_ground_duty — **Rotation ground duty assignment**

task	***ground_task***	**FK**	Ground task identifier
pos	***aircraft_position_set***	**FK**	Position identifier
rot	*rotation*	**FK**	Rotation identifier
si	*string*		Supplementary information

6.6.30.1 Extension from module air_urm

URM extension for rotation_ground_duty.

rotation_ground_duty

urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.31 Entity rotation_activity

This entity links a rotation with a non-flying activity. See also *ground_task*.

rotation_activity			Rotation activity
st	***time***		Start time
rot	***rotation***	**FK**	Rotation identifier
activity	***aircraft_activity_set***	**FK**	Activity type identifier
et	***time***		End time
adep	***airport***	**FK**	Airport of departure
ades	***airport***	**FK**	Airport of destination
si	*string*		Supplementary information

6.6.31.1 Extension from module air_urm

URM extension for rotation_activity.

rotation_activity			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes

6.6.32 Entity aircraft_connection

This entity is an alternative way of describing aircraft rotations in the form of pairs of duties served by the same aircraft. The rotation_xxx entities will likely be phased out once all products switch to using aircraft_connection.

NB: This entity can only describe flight leg connections. It is unclear if it will be necessary to model connections relating to ground_tasks and rotation activities. Should we have different entities for these or reuse this entity? How define the primary key?

aircraft_connection			Aircraft connection
legfrom	***flight_leg***	**FK**	First flight leg identifier
legto	***flight_leg***	**FK**	Second flight leg identifier
si	*string*		Supplementary information

6.6.33 Entity aircraft

This entity represents an aircraft.

aircraft			Aircraft
id	***string(10)***		Aircraft identifier (registration)
actype	***aircraft_type***	**FK**	Aircraft type identifier
si	*string*		Supplementary information

6.6.33.1 Extension from module air_aircraft

Extra aircraft information needed for IOCS.

aircraft			Extra AC info
altid	*string(10)*		Alternative aircraft identifier (used internally by customer)
nationality	*string(4)*		Country of registration, two letter ISO code
owner	*string(4)*		Owner
ilscat	*ils_category_set*	**FK**	ILS capability
seatconfig	*string(32)*		Seat configuration
st	*time*		Start time of service life
et	*time*		End time of service life

6.6.34 Entity aircraft_flight_duty

This entity links an aircraft with a flight leg. If the flight duty was assigned from a rotation, the *rot* field allows identification of the rotation from whence it came.

aircraft_flight_duty			Aircraft flight duty assignment
leg	***flight_leg***	**FK**	Flight leg identifier
ac	***aircraft***	**FK**	Aircraft identifier
rot	*rotation*	**FK**	Rotation identifier
pos	*aircraft_position_set*	**FK**	Position identifier
si	*string*		Supplementary information

6.6.34.1 Extension from module air_urm

URM extension for aircraft_flight_duty.

aircraft_flight_duty

urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.35 Entity aircraft_ground_duty

This entity links an aircraft with a ground task. If the ground task was assigned from a rotation, the *rot* field allows identification of the rotation from whence it came.

aircraft_ground_duty — **Aircraft ground duty assignment**

task	***ground_task***	**FK**	Ground task identifier
ac	***aircraft***	**FK**	Aircraft identifier
rot	*rotation*	**FK**	Rotation identifier
si	*string*		Supplementary information

6.6.35.1 Extension from module air_urm

URM extension for aircraft_ground_duty.

aircraft_ground_duty

urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.36 Entity aircraft_activity

This entity links individual aircraft with non-flying activity. An aircraft activity may be preassigned or stem from the *rotation_activity* that has been assigned to an aircraft. If the aircraft activity was assigned from a rotation, the *rot* field allows identification of the rotation from whence it came. See also *ground_task*.

aircraft_activity			Aircraft activity
st	***time***		Start time
ac	***aircraft***	**FK**	Aircraft identifier
activity	***aircraft_activity_set***	**FK**	Activity type identifier
et	***time***		End time
adep	***airport***	**FK**	Airport of departure
ades	***airport***	**FK**	Airport of destination
rot	*rotation*	**FK**	Rotation identifier
si	*string*		Supplementary information

6.6.36.1 Extension from module air_iocs

Extra flight leg info needed by IOCS.

aircraft_activity		Extra maintenance info
actid	*string(32)*	Activity ID
est	*time*	Estimated start time
eet	*time*	Estimated end time
ast	*time*	Actual start time
aet	*time*	Actual end time
locktype	*string(16)*	Lock type
ruleexception	*string(16)*	Activity rule exception
opdefconstraint	*string(80)*	OpDef constraint

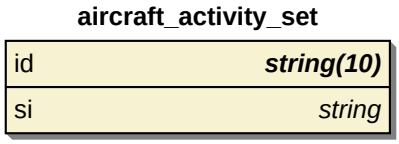
6.6.36.2 Extension from module air_urm

URM extension for aircraft_activity.

aircraft_activity			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes

6.6.37 Entity aircraft_activity_set

This entity lists allowed aircraft activity types. See *ac_activity*.

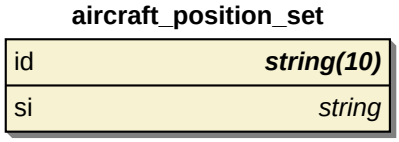


Aircraft activities

Activity type identifier

Supplementary information

6.6.38 Entity aircraft_position_set



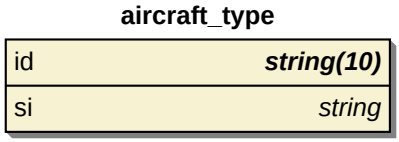
Aircraft positions

Position type identifier

Supplementary information

6.6.39 Entity aircraft_type

This entity lists allowed aircraft types.



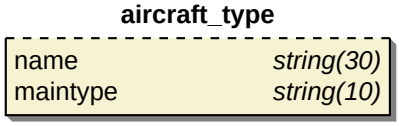
Aircraft types

Type identifier

Supplementary information

6.6.39.1 Extension from module air_aircraft

Extra aircraft information needed for IOCS.



Extra AC type info

Full aircraft type name

Main type

6.6.39.2 Extension from module sas_air_aircraft

Additional SAS information for an aircraft type.

aircraft_type			aircraft_type, SAS extension
carrier	*crew_company_set*	**FK**	Carrier
crewbunkfc	*int*		Crew bunks for flight crew
crewbunkcc	*int*		Crew bunks for cabin crew
maxfc	*int*		Max available seats for flight crew
maxcc	*int*		Max available seats for cabin crew

6.6.40 Entity equipment

This entity represents an equipment item that may be required on a flight or ground task.

equipment			Equipment
id	***string(10)***		Equipment identifier
etype	***equipment_set***	**FK**	Equipment type identifier
si	*string*		Supplementary information

6.6.41 Entity equipment_flight_duty

This entity links an equipment item with a flight leg.

equipment_flight_duty			Equipment flight duty assignment
leg	***flight_leg***	**FK**	Flight leg identifier
eqt	***equipment***	**FK**	Equipment identifier
si	*string*		Supplementary information

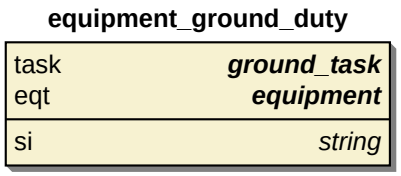
6.6.41.1 Extension from module air_urm

URM extension for equipment_flight_duty.

equipment_flight_duty			
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination

6.6.42 Entity equipment_ground_duty

This entity links an equipment item with a ground task.

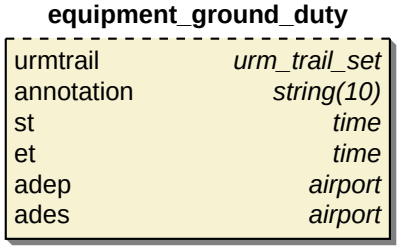


Equipment ground duty assignment

task	FK	Ground task identifier
eqt	FK	Equipment identifier
si		Supplementary information

6.6.42.1 Extension from module air_urm

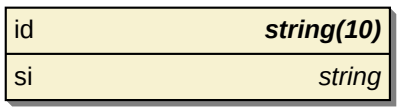
URM extension for equipment_ground_duty.



urmtrail	FK	URM trail version
annotation		annotation codes
st		Start time
et		End time
adep	FK	Airport of departure
ades	FK	Airport of destination

6.6.43 Entity equipment_set

This entity lists allowed equipment types.



Equipment types

id	Type identifier
si	Supplementary information

6.6.44 Entity rule_exception

rule_exception			Rule exception
crew	***crew***	**FK**	Crew id
ruleid	***string***		Rule id
starttime	***time***		Start time
activitykey	*string*		Fail object key
ruleremark	*string*		Rule text
limitval	*string*		Limit
actualval	*string*		Actual value
overrel	*reltime*		Overshoot reltime
overint	*int*		Overshoot integer
username	*string*		System user
ctime	*time*		Date and time of creation
reason	*exception_reason_set*	**FK**	Reason
si	*string*		Supplementary information

6.6.45 Entity exception_reason_set

exception_reason_set		Rule exception reasons
id	***string***	Reason code
si	*string*	Supplementary information

6.7 Module air_crew

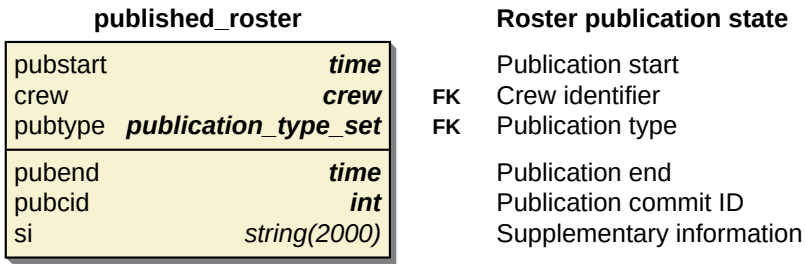
The basic crew entity is defined in the air_core module. The crew entity defined therein is only the bare minimum needed to allow references to crew members. This module, air_crew, extends the crew entity with more information for systems that need to work with crew.

More things to consider:

- holiday table
- allowance -> add to country (but allowance may be different from different bases)
- duty periods?
- customs and schengen?
- min conn times?
- country areas?

6.7.1 Entity published_roster

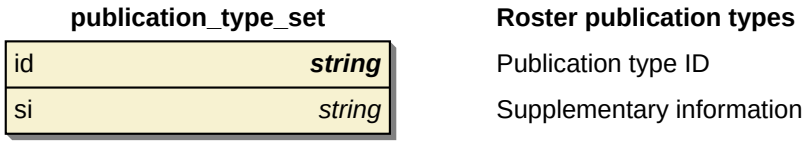
This entity describes the latest published state of a roster.



published_roster			Roster publication state
pubstart	***time***		Publication start
crew	***crew***	**FK**	Crew identifier
pubtype	***publication_type_set***	**FK**	Publication type
pubend	***time***		Publication end
pubcid	***int***		Publication commit ID
si	*string(2000)*		Supplementary information

6.7.2 Entity publication_type_set

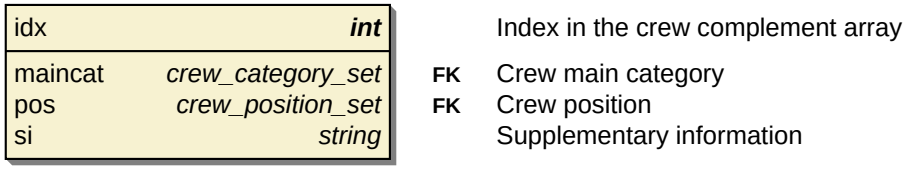
This entity lists allowed roster publication types.



publication_type_set		Roster publication types
id	***string***	Publication type ID
si	*string*	Supplementary information

6.7.3 Entity crew_complement

This entity defines how the crew complement string or integer array should be interpreted.



crew_complement			Crew complement definition
idx	***int***		Index in the crew complement array
maincat	*crew_category_set*	**FK**	Crew main category
pos	*crew_position_set*	**FK**	Crew position
si	*string*		Supplementary information

6.7.4 Entity crew_address

This entity represents a crew member address. A crew member may have only one active address at a time, but it is possible to list future known address changes. The state field should contain the state, province or county if applicable, otherwise null. Use the official abbreviations if available.

crew_address			Crew address
crew	***crew***	**FK**	Crew identifier
validfrom	***time***		Validity period start
validto	*time*		Validity period end
street	***string***		Street, may be several lines
city	***string(30)***		City
state	*string(30)*		State, province or county
postalcode	*string(10)*		Postal code
country	***string(4)***		Two letter ISO country code
si	*string*		Supplementary information

6.7.4.1 Extension from module sas_air_crew

Additional SAS information for crew address.

crew_address		Ext. to crew_address
street1	*string*	Street, may be several lines
city1	*string(30)*	City
state1	*string(30)*	State, province or county
postalcode1	*string(10)*	Postal code
country1	*string(4)*	Two letter ISO country code

6.7.5 Entity crew_contact

This entity lists contact information such as telephone numbers and email addresses for crew members. A crew member can have several contacts. Always include the country code in telephone numbers.

crew_contact			Crew contact
crew	***crew***	**FK**	Crew member
typ	***string***		Type of contact: tel,fax,mobile,pager,email
val	***string***		Contact
which	*string*		Where: work,home,main,other
si	*string*		Supplementary information

6.7.6 Entity crew_employment

This entity represents the details of a crew members employment. A crew member may have only one active employment status at a time, but it is possible to list future known employment

status changes.

crew_employment			Crew employment status
crew	***crew***	**FK**	Crew identifier
validfrom	***time***		Validity period start
validto	*time*		Validity period end
carrier	*crew_carrier_set*	**FK**	Carrier
company	*crew_company_set*	**FK**	Company
base	*crew_base_set*	**FK**	Base
crewrank	*crew_rank_set*	**FK**	Rank
titlerank	*crew_rank_set*	**FK**	Title rank
si	*string*		Supplementary information

6.7.6.1 Extension from module sas_air_crew

Additional SAS specific information about a crew member's employment.

crew_employment			Ext. to crew_employment
region	*crew_region_set*	**FK**	Region
civicstation	*string(5)*		Civic station
station	*string(5)*		Station
country	*country*	**FK**	Country of employment
extperkey	*string(8)*		External Perkey

6.7.7 Entity crew_carrier_set

This entity lists supported crew carriers.

crew_carrier_set		Crew carriers
id	***string***	Carrier, use the carrier code
si	*string*	Supplementary information

6.7.8 Entity crew_company_set

This entity lists supported crew companies.

crew_company_set			Crew companies
id	***string***		Company
si	*string*		Supplementary information

6.7.9 Entity crew_rank_set

This entity lists supported crew ranks.

crew_rank_set			Crew categories
id	***string***		Rank
maincat	*crew_category_set*	**FK**	Main category
si	*string*		Supplementary information

6.7.10 Entity crew_category_set

This entity lists available crew main categories.

crew_category_set			Crew main categories
id	***string(16)***		Main category
si	*string*		Supplementary information

6.7.11 Entity crew_seniority

This entity represents the type of contract for each crew member at any given point in time.

crew_seniority			Crew seniority
crew	***crew***	**FK**	Crew identifier
grp	***crew_sen_grp_set***	**FK**	Seniority group
validfrom	***time***		Validity period start
validto	*time*		Validity period end
seniority	*int*		Seniority
si	*string*		Supplementary information

6.7.12 Entity crew_sen_grp_set

This entity lists all crew seniority groups.

crew_sen_grp_set		Crew seniority groups
id	***string***	Contract
si	*string*	Supplementary information

6.7.13 Entity crew_contract

This entity represents the type of contract for each crew member at any given point in time.

crew_contract			Crew contract status
crew	***crew***	**FK**	Crew identifier
validfrom	***time***		Validity period start
validto	*time*		Validity period end
contract	*crew_contract_set*	**FK**	Contract
si	*string*		Supplementary information

6.7.13.1 Extension from module air_manpower

This is an extension to crew_contract.

crew_contract		Manpower additions
endreason	*string*	Reason for end of contract

6.7.13.2 Extension from module sas_air_crew

Additional SAS specific information for a crew member's different contracts.

crew_contract		Ext. to crew_contract
bxmodule	*string*	Module
cyclestart	*int*	Cycle start
laborunion	*string(10)*	Labor union
noofvadays	*int*	Number of vacation days

6.7.14 Entity crew_contract_set

This entity lists supported crew contract types.

crew_contract_set		Crew contracts
id	***string***	Contract
si	*string*	Supplementary information

6.7.14.1 Extension from module sas_air_crew

Additional SAS information for a crew contract.

crew_contract_set			Ext. to crew_contract_set
dutypercent	*int*		Duty in percent
grouptype	*string(4)*		Group type
pattern	*pattern_set*	**FK**	Pattern to use
nooffreedays	*int*		Number of free days
noofparttime	*int*		Number of part time days
parttimecode	*string(4)*		Activity code for part time days
descshort	*string(20)*		Short descriptive text
desclong	*string(20)*		Long descriptive text

6.7.15 Entity crew_contract_valid

This entity lists availability of crew contract types.

crew_contract_valid			Valid crew contract types
contract	***crew_contract_set***	**FK**	Contract
validfrom	***time***		Validity period start
validto	*time*		Validity period end
si	*string*		Supplementary information

6.7.15.1 Extension from module sas_air_crew

Additional SAS information for crew contract validity.

crew_contract_valid			Ext. to crew_contract_valid
maincat	*string(2)*		Main category
base	*crew_base_set*	**FK**	Crew base
company	*crew_company_set*	**FK**	Company

6.7.16 Entity crew_group

This entity lists planned crew groups. It drives the planning process by listing the assumed crew groups at various points in time. A crew member can have only one active group at a time.

crew_group			Crew group
crew	***crew***	**FK**	Crew identifier
grp	***crew_group_set***	**FK**	Group
validfrom	***time***		Validity period start
validto	*time*		Validity period end
si	*string*		Supplementary information

6.7.17 Entity crew_group_set

This entity lists known groups.

crew_group_set		Crew groups
id	***string***	Group
si	*string*	Supplementary information

6.7.17.1 Extension from module cmp_transition

This entity ...

crew_group_set		
demand	*int*	
baseassign	*bool*	
settings	*crew_group_setting*	**FK**

6.7.18 Entity crew_group_valid

This entity lists availability of crew group types.

crew_group_valid			Valid crew group types
grp	***crew_group_set***	**FK**	Group
validfrom	***time***		Validity period start
validto	*time*		Validity period end
si	*string*		Supplementary information

6.7.19 Entity crew_qualification

This entity lists planned crew qualifications. It drives the planning process by listing the assumed crew qualifications at various points in time. This must not be confused with the currently held set of qualifications which is described in crew_document. A crew member can have several active qualifications at a time.

crew_qualification			Crew qualification
crew	***crew***	**FK**	Crew identifier
qual	***crew_qualification_set***	**FK**	Qualification
validfrom	***time***		Validity period start
validto	*time*		Validity period end
lvl	*string*		Level
si	*string*		Supplementary information

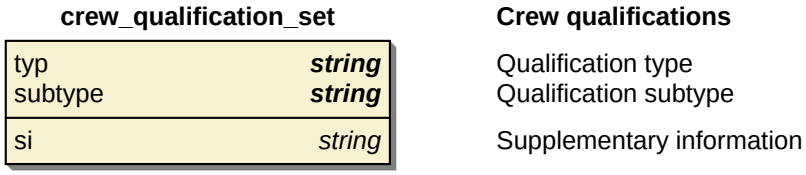
6.7.19.1 Extension from module sas_air_crew

Additional SAS specific information for a crew member's qualifications.

crew_qualification		Ext. to crew_qualification
acstring	*string(50)*	Comma separated list of qual.subtype in other crew_qualification rows

6.7.20 Entity crew_qualification_set

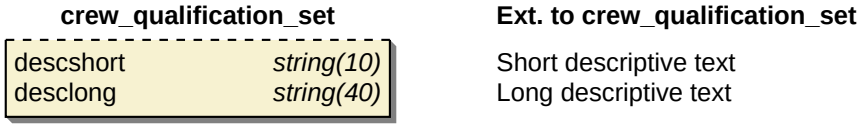
This entity lists known qualification types/subtypes. Possible types: ac, arp, route, visa, med, vacc, either_seat, instructor, lang.



crew_qualification_set		Crew qualifications
typ	***string***	Qualification type
subtype	***string***	Qualification subtype
si	*string*	Supplementary information

6.7.20.1 Extension from module sas_air_crew

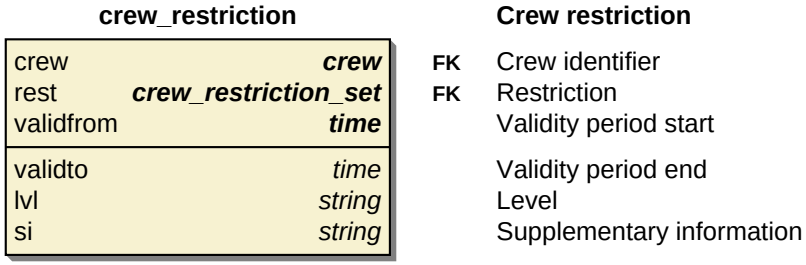
Additional SAS information for crew qualifications.



crew_qualification_set		Ext. to crew_qualification_set
descshort	*string(10)*	Short descriptive text
desclong	*string(40)*	Long descriptive text

6.7.21 Entity crew_restriction

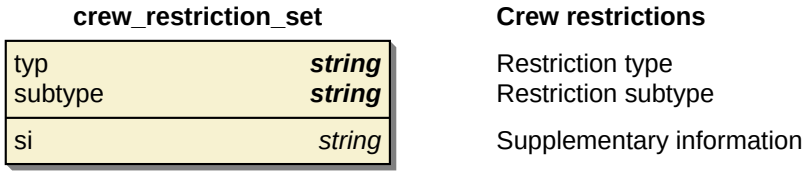
This entity lists planned crew restrictions. It drives the planning process by listing the assumed crew restrictions at various points in time. A restriction does not necessarily mean that the crew member has lost a qualification, but instead restricts the rosters that are built for the crew member. A crew member can have several active restrictions at a time.



crew_restriction			Crew restriction
crew	***crew***	**FK**	Crew identifier
rest	***crew_restriction_set***	**FK**	Restriction
validfrom	***time***		Validity period start
validto	*time*		Validity period end
lvl	*string*		Level
si	*string*		Supplementary information

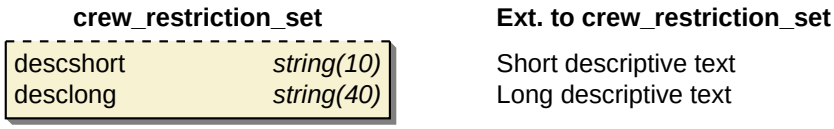
6.7.22 Entity crew_restriction_set

This entity lists known restriction types/subtypes. Possible types: sickness, part time, no early flights, grounded.



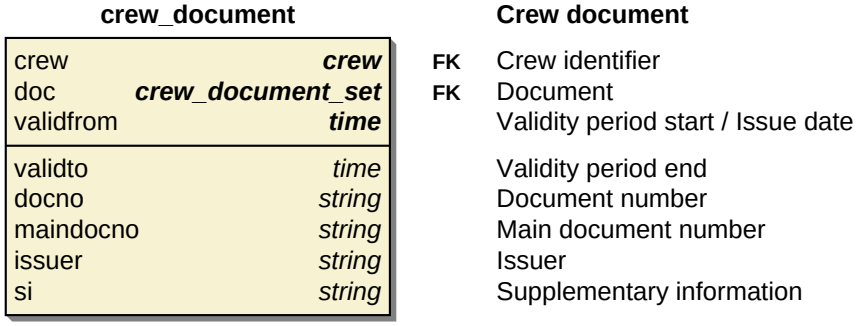
6.7.22.1 Extension from module sas_air_crew

Additional SAS information for crew restrictions.



6.7.23 Entity crew_document

This entity can be used to record currently held documents, qualifications and completed recurrent training for individual crew members. This information should be an as accurate as possible reflection of the currently held qualifications. It should not be confused with other entities used for grouping and planning purposes.



6.7.24 Entity crew_document_set

This entity lists known crew document types/subtypes. The type field should indicate the main type of document concerned, e.g. "visa". The subtype further qualifies the document, e.g. "CN". Possible types: passport, visa, vaccination, language, licence, rating, medical, recurrent training (emergency, line, route, airport).

crew_document_set		Crew documents
typ	***string***	Document type
subtype	***string***	Document subtype
si	*string*	Supplementary information

6.7.25 Entity exchange_rate

Exchange rate table. The rate is given in reference currency per unit foreign currency. Example: ref is SEK. cur is EUR. unit is 1000. rate is 9334. This means one gets 9334 SEK for 1000 EUR. The unit is present because of the lack of a floating point representation.

exchange_rate			Exchange rate
reference	***currency_set***	**FK**	Reference currency
cur	***currency_set***	**FK**	Exchange currency
validfrom	***time***		Validity period start
validto	*time*		Validity period end
rate	*int*		Exchange rate
unit	*int*		Unit value

6.7.26 Entity currency_set

Currency table. Keys should be the ISO 4217 codes.

currency_set		Currency codes
id	***string(8)***	Currency code
si	*string*	Supplementary information

6.7.27 Entity hotel

This entity describes a hotel with name, address and telephone numbers.

hotel		Hotel
id	***string***	Hotel code, e.g. HEL3
name	***string***	Hotel name
street	***string***	Street, may be several lines
city	***string(30)***	City
state	*string(30)*	State, province or county
postalcode	*string(10)*	Postal code
country	***string(4)***	Two letter ISO country code
tel	*string*	Telephone number
fax	*string*	Fax number
si	*string*	Supplementary information

6.7.27.1 Extension from module sas_air_crew

hotel		Ext. to hotel
email	*string(40)*	Email address

6.7.28 Entity airport_hotel

This entity describes a link between an airport and a hotel.

airport_hotel			Airport hotel
airport	***airport***	**FK**	Airport
hotel	***hotel***	**FK**	Hotel
si	*string*		Supplementary information

6.7.29 Entity hotel_contract

This entity describes an agreement with a hotel.

hotel_contract			Hotel contract
hotel	***hotel***	**FK**	Hotel
validfrom	***time***		Validity period start
validto	*time*		Validity period end
xeci	*reltime*		Extra early check-in time
eci	*reltime*		Early check-in time
lco	*reltime*		Late check-out time
xlco	*reltime*		Extra late check-out time
costco	*int*		Cost for cockpit crew
costca	*int*		Cost for cabin crew
cur	*currency_set*	**FK**	Hotel cost currency
contact	*string*		Contact person
si	*string*		Supplementary information

6.7.30 Entity preferred_hotel

This entity describes a preferred hotel.

preferred_hotel			Preferred hotel
airport	***airport***	**FK**	Airport
validfrom	***time***		Validity period start
validto	*time*		Validity period end
hotel	***hotel***	**FK**	Hotel
si	*string*		Supplementary information

6.7.31 Entity account_entry

Account entries and transactions, created manually or created based on assignments and rules.

account_entry			Account entry
id	***uuid***		Entry identifier
crew	*crew*	**FK**	Crew identifier
tim	*time*		Time of transactions
account	*account_set*	**FK**	Account type
source	*string*		Source of transaction
amount	*int*		Number of days (*100) in entry, positive for deposit
man	*bool*		True if the entry was created manually, and not calculated based on e.g
si	*string*		Supplementary information

6.7.31.1 Extension from module sas_accounts

account_entry		Ext. to account_entry
published	*bool*	True if the scheduled day is published to crew and visible
rate	*int*	The rate which the day has been booked for
reasoncode	*string*	The strictly typed trasactioncode for the transaction
entrytime	*time*	Post updated at time
username	*string*	Post updated by process owned by

6.7.32 Entity account_set

The available accounts referenced by account_entry.

account_set		Account set
id	***string***	Account identifier
si	*string*	Supplementary information

6.7.33 Extensions

This module also extends the activity_group_period (6.6.16.1) and crew (6.6.22.1) entities.

6.8 Module air_iocs

This module adds definitions required by the Integrated Operations Control products.

6.8.1 Entity activity_connection

This entity lists required links between activities or a minimum connection time.

activity_connection		Activity Connection
udor2	***date***	UDOR of second activity
act1	***string(32)***	ID of first activity (before connection)
act2	***string(32)***	ID of second activity (after connection)
fixedlink	*string(20)*	Fixed link type
minconntime	*reltime*	Minimum connection time
ruleexception	*string(16)*	Connection rule exception
si	*string*	Supplementary information

6.8.2 Entity sched_ac_flight_duty

This entity shows the original scheduled aircraft flight duty assignment.

sched_ac_flight_duty — **Scheduled aircraft flight duty assignment**

leg	***flight_leg***	**FK**	Flight leg identifier
ac	***aircraft***	**FK**	Aircraft identifier
si	*string*		Supplementary information

6.8.3 Entity oag_ssim

This entity represents SSIM flight records. A SSIM record is defined by the first day of the record (bdor), its flight designator(fd), its departure airport(adep), and its weekday pattern (wdays). The bdor and edor may be in local time (mode='L') or in UTC (mode='U'). The off-block start minutes (sobm) and in-block end minutes (sibm) are added to the day of origin (1440*dor) to create the actual start (sobt) and end (sibt) times. sobm/sibm are always in UTC. The weekday pattern is a bit mask of days, where the least significant bit (bit0) represents sunday, then monday all the way up to saturday (bit6). The end day of origin (edor) marks the first day after the record.

oag_ssim — **OAG SSIM records**

bdor	***date***		First date of origination
fd	***string(20)***		Flight designator (carrier,number,suffix)
adep	***airport***	**FK**	Airport of departure
wdays	***int***		Weekday pattern
edor	***date***		End date of origination
ades	***airport***	**FK**	Airport of arrival
stc	*char*		Service type code
sobm	***reltime***		Scheduled off-block offset in minutes
sibm	***reltime***		Scheduled in-block offset in minutes
timemode	*char*		Time mode, L or U
actype	*aircraft_type*	**FK**	Aircraft type
oper	*string(4)*		Operating carrier
si	*string*		Supplementary information

6.8.4 Entity oag_flight_leg

This entity lists individual OAG flight legs.

oag_flight_leg			OAG flight leg
udor	***date***		Scheduled date of origination
fd	***string(20)***		Flight designator (carrier,number,suffix)
adep	***airport***	**FK**	Airport of departure
ades	***airport***	**FK**	Airport of arrival
stc	*char*		Service type code
sobt	*time*		Scheduled off-block time
sibt	*time*		Scheduled in-block time
eobt	*time*		Estimated off-block time
eibt	*time*		Estimated in-block time
aobt	*time*		Actual off-block time
aibt	*time*		Actual in-block time
actype	*aircraft_type*	**FK**	Aircraft type
statcode	*leg_status_set*	**FK**	Status
aco	*string(4)*		Aircraft owner
cpe	*string(4)*		Cockpit crew employer
cae	*string(4)*		Cabin crew employer
si	*string*		Supplementary information

6.8.5 Entity crew_oag_duty

This entity links a crew member with an OAG leg.

crew_oag_duty			Crew oag assignments
leg	***oag_flight_leg***	**FK**	OAG leg identifier
crew	***crew***	**FK**	Crew identifier
pos	*crew_position_set*	**FK**	Position identifier
trip	*trip*	**FK**	Trip identifier
si	*string*		Supplementary information
urmtrail	*urm_trail_set*	**FK**	URM trail version
annotation	*string(10)*		annotation codes
st	*time*		Start time
et	*time*		End time
adep	*airport*	**FK**	Airport of departure
ades	*airport*	**FK**	Airport of destination
personaltrip	*uuid*		ID of a personal trip

6.8.6 Entity cs_flight_leg

This entity links code-shared flight legs to their real counterparts.

cs_flight_leg			Code-share flight leg
udor	***date***		Code-share udor
fd	***string(20)***		Code-share flight designator
adep	***airport***	**FK**	Code-share departure airport
opleg	***flight_leg***	**FK**	Operating flight leg identifier
si	*string*		Supplementary information

6.8.7 Entity pnr

This entity represents a PNR record. It will be moved to the air_pax module.

pnr			PNR (Passenger Name Record)
dt	***date***		PNR date
num	***string(10)***		PNR number
onumber	*string(10)*		Original PNR number
groupname	*string(30)*		Group name
groupsize	*int*		Group size
si	*string*		Supplementary information

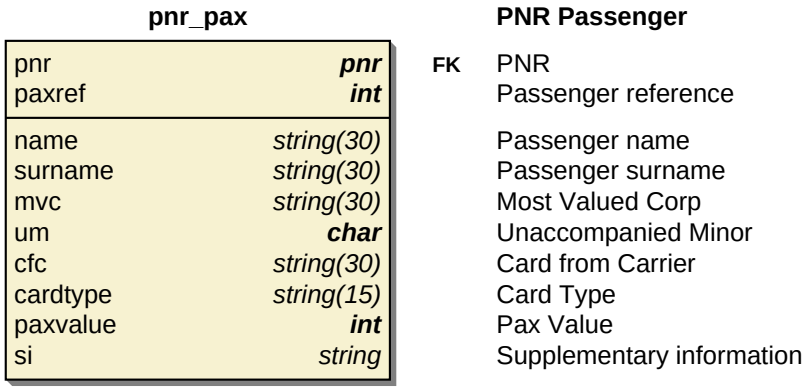
6.8.8 Entity pnr_seg

This entity represents a PNR segment record. It will be moved to the air_pax module.

pnr_seg			PNR Segment
pnr	***pnr***	**FK**	PNR
leg	***flight_leg***	**FK**	Flight leg
cabinclass	***char***		Cabin class
fareclass	***string(2)***		Fare class
si	*string*		Supplementary information

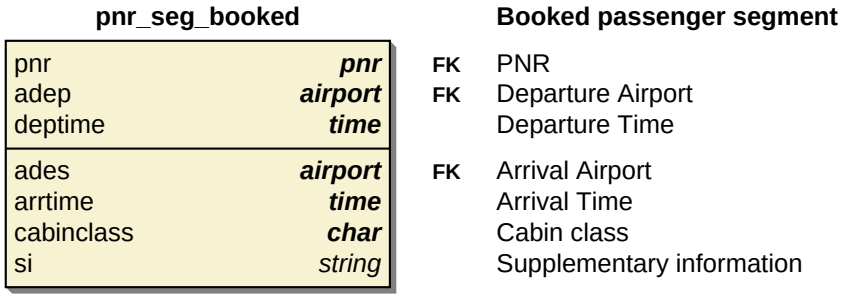
6.8.9 Entity pnr_pax

This entity represents a PNR passenger record. It will be moved to the air_pax module.



6.8.10 Entity pnr_seg_booked

This entity represents a PNR segment booking record. It will be moved to the air_pax module.



6.8.11 Entity pnr_pax_ssr

This entity represents a Special Service Request (SSR) for a passenger and a segment of a PNR record. It will be moved to the air_pax module.

pnr_pax_ssr		Special Service Request
pnr_dt	***date***	PNR Creation Date
pnr_num	***string***	PNR Record Locator
paxref	***int***	Passenger Reference Number
leg_udor	***date***	Flight Departure Date
leg_fd	***string***	Flight Designator
leg_adep	***string***	Departure Airport
faxitemnum	***int***	Fax Item Number
ssrcode	***string***	SSR IATA code
ssrstatus	***string***	SSR Confirmation Status
si	*string*	Supplementary information

6.8.12 Extensions

This module also extends the flight_leg (6.6.1.1), aircraft_activity (6.6.36.1) and trip (6.6.18.1) entities.

6.9 Module air_manpower

The air_manpower module contains the necessary entities to support the Carmen CMS products Manpower Planning and Training Workspace for training and leave planning.

Standard parameter forms are included here. Additional user specific parameter forms can be added in customer projects.

6.9.1 Entity accumulator_int_run

Stores data about latest accumulator runs for accumulator_int

accumulator_int_run		Information about accumulator runs
accname	***string(50)***	Name of accumulator
acckey	***string(255)***	The accumulator key, do not have to be the same key as in accumulato
accstart	*date*	Accumulator period start date
accend	*date*	Accumulator period end date
lastrun	*time*	Time of last accumulation

6.9.2 Entity crew_filter

This entity represents a crew filter.

crew_filter			Crew Selection Filter
name	***string***		The name of the Selection filter
cat	***string(10)***		Defines the category
typ	*crew_filter_type_set*	**FK**	Type of crew filter
selvalue	*string*		The value of the crew selection
si	*string*		Supplementary information

6.9.3 Entity crew_filter_type_set

This entity represents a crew filter type. The following id values will be available:
id = "PLANNING" defines a planning crew selection, can be used both for views and in parameters
id = "VIEW" defines a crew selection, can be used for views
id = "TRAINING" defines a crew selection suitable for a specific training, can be used in training

crew_filter_type_set		Crew filter types
id	***string(10)***	Crew filter type identifier
si	*string*	Supplementary information

6.9.4 Entity crew_filter_user

Defines which crew filters a specific user has activated.

crew_filter_user			User activated crew filter
username	***string***		The identification of the user
crewfilter	***crew_filter***	**FK**	The crew filter

6.9.5 Entity crew_position_type_set

Set of crew position types
id = "FLYING" defines a flying position
id = "INSTR" defines an instructor position
id = "EXTRA" defines an extra position

crew_position_type_set	
id	***string(10)***
si	*string*

Set of crew position types

The identification of the type code

Supplementary information

6.9.6 Entity workset

This entity represents a workset.

workset	
name	***string***
cat	*string(10)*
startdate	*date*
enddate	*date*
od	*string*
revision	*string*
si	*string*

Manpower Workset definition

The name of the Selection filter

Defines the category
Start Date
End date
Operation day
Revision identifier
Supplementary information

6.9.7 Entity color

This entity represents a color.

color	
name	***string(15)***

Nordic Light Color

6.9.8 Entity bid_consume_leave

This entity represents a wish from a crew member to consume leave for a certain period of time.

bid_consume_leave			Leave consumption bids
id	***uuid***		A unique identifier for the bid
crew	*crew*	**FK**	
prio	*int*		Priority within bids from same crew member
altprio	*int*		Alternate priority (if allowed) or null
block	*int*		Block number for the bid, several rows with the same id and block are c
leavetype	*string*		The type of leave
leavestart	*time*		The start date for the leave that this bid represents
leaveend	*time*		The end date for the leave that this bid represents
si	*string*		Supplementary information

6.9.9 Entity course_template

This entity represents a course template.

course_template			Course Template
name	***string***		The name of the course template
cat	***string(10)***		Defines the category the course is valid for, eg cabin, cockpit
ctype	*course_type*	**FK**	Course type
qualobt	*crew_qualification_set*	**FK**	Qualification obtained
maxparticipants	*string*		Crew complement indicating the maximum number of participants
sr	*string*		Name of Rave or python selection rule script
si	*string*		Supplementary information

6.9.10 Entity course_block_template

This entity represents a block template.

course_block_template			Course Block Template
ct	***course_template***	**FK**	The course to which the block belongs
nr	***int***		Block id number in course
ordernr	*int*		Block order number in course
name	*string*		The name of the block template
cbtype	*course_block_type*	**FK**	Block Template Type
actcode	*activity_set*	**FK**	Activity code for generated assignments
freedayactcode	*activity_set*	**FK**	Activity code for generated training free days
noact	*int*		Activitys per participant.
duration	*reltime*		Activity duration per day
brieftime	*reltime*		Briefing time
debrieftime	*reltime*		Debriefing time
fixedlength	*bool*		Fixed length block to maxdays
maxdays	*int*		Maximum number of days for the block
daystouse	*string(7)*		Days to use. E.g. 1234567 = all days of the week
instrquallist	*string*		Instructor Qualifications needed, comma separated list
resourceneed	*resource_group*	**FK**	Resource group needed
nrslotsperday	*int*		Max resource slots for this course block.
freedaysforprod	*bool*		Free days in block can be used for procuction
minpause	*int*		Minimum number of pause days after the block
si	*string*		Supplementary information

6.9.11 Entity course_type

This entity represents a course template type.
id = "REC" for recurrent courses
id = "TRANS" for transisiton courses
id = "OTHER" for other courses

course_type		Course Type
id	***string(10)***	Course Template type identifier
si	*string*	Supplementary information

6.9.12 Entity course_block_type

This entity represents a block type. The following id values will be available:
id = "GROUND_CLASS"
id = "GROUND_SIM"
id = "GROUND_FLIGHT"
id = "GROUND_CHECK"
id = "PROD_FLIGHT"
id = "PROD_OBS"

id = "PROD_CHECK"

course_block_type — **Course Block Type**

id	***string(30)***		Block type identifier/name
color	*string*		Color to be assocciated with the block type
text	*string*		Text to be displayed for the block type
actcode	*activity_set*	**FK**	Default activity code for generated assignments
freedayactcode	*activity_set*	**FK**	Default activity code for generated training free days
resourceneed	*resource_group*	**FK**	Default resource needneed
validfor	*string*		A list of course types that this block type is valid e.g. TRANS,REC
instrprod	*bool*		Instructor is in production.
participantprod	*bool*		Student is in procuction.
instrdefinable	*bool*		Instructor need
resourcedefinable	*bool*		Resource need
durationdefinable	*bool*		Activity duration definable.
actcreated	*bool*		Activities created
si	*string*		Supplementary information

6.9.13 Entity course

This entity represents a course. A course is created from a course_template. A course has one or more course_block connected to it. A course_block has course_block_trip connected to it. The number of course_block_trip a course_block has depends on the number of course participants and the resource need. Each course_block_trip has a number of course_activites.

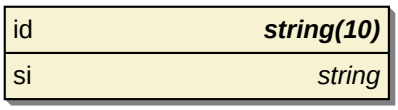
The course has course_participants connected to it. A course_participant can be coupled to a crew.

course — **Course**

name	***string***		The name of the course
cat	***string(10)***		Crew category identifier
status	*course_status_set*	**FK**	Course status
carrier	*string*		Carrier name
ctype	*course_type*	**FK**	Course type
qualobt	*crew_qualification_set*	**FK**	Qualification obtained
createdate	*time*		Course create date (time)
templname	*string*		Course template name
startdate	*date*		Start Date
enddate	*date*		End date
cc	*string*		Crew complement
ref	*string*		Planning reference code. ex: School season, Promotion campain
airport	*airport*	**FK**	Airport course is performed
sr	*string*		Name of Rave or python selection rule script
si	*string*		Supplementary information.

6.9.14 Entity course_status_set

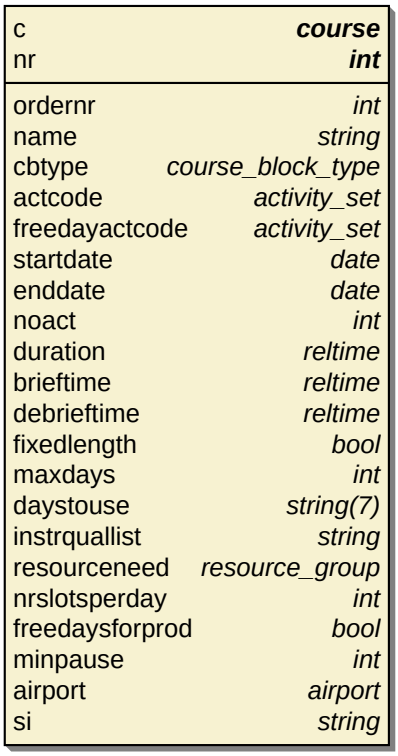
This entity represents a course status set.
id = "OPEN" open courses - no restrictions
id = "CONFIRMED" confirmed courses - warnings for changes, force revison updates
id = "PUBLISHED" published course - no changes can be made to the course



course_status_set		Course Status set
id	***string(10)***	Course Status identifier
si	*string*	Supplementary information

6.9.15 Entity course_block

This entity represents the course block values. A course block is created from a course_block_template.



course_block			Block
c	***course***	**FK**	The course the block belongs to
nr	***int***		Block id number in course
ordernr	*int*		Block order number in course
name	*string*		The name of the block
cbtype	*course_block_type*	**FK**	Block Template Type
actcode	*activity_set*	**FK**	Activity code for generated assignments
freedayactcode	*activity_set*	**FK**	Activity code for generated training free days
startdate	*date*		Start Date
enddate	*date*		End Date
noact	*int*		Activities per participant.
duration	*reltime*		Activity duration per day
brieftime	*reltime*		Briefing time
debrieftime	*reltime*		Debrief time
fixedlength	*bool*		Fixed length block to maxdays
maxdays	*int*		Maximum number of days for the block
daystouse	*string(7)*		Days to use. E.g. 1234567 = all days
instrquallist	*string*		Instructor Qualifications needed, comma separated list
resourceneed	*resource_group*	**FK**	Resource group needed
nrslotsperday	*int*		Max resource slots for this course block.
freedaysforprod	*bool*		Free days in block can be used for procuction
minpause	*int*		Minimum number of pause days after the block
airport	*airport*	**FK**	Airport course is performed
si	*string*		Supplementary information

6.9.16 Entity course_block_trip

This entity represents a block trip.

course_block_trip			Course block trip
cb	***course_block***	**FK**	Block identifier
nr	***int***		Course block trip nr
startdate	*date*		Start Date
enddate	*date*		End Date
cc	*string*		Crew complement need vector
trip	*trip*	**FK**	Trip identifier
si	*string*		Supplementary information

6.9.17 Entity course_activity

This entity represents a course activity. A course activity is the the action that one or more (depending on the resource need) course_participant will do on one day.

course_activity			Course Activity
cbt	***course_block_trip***	**FK**	Course block trip identifier
nr	***int***		Activity number in course block trip
name	*string*		The name of the course activity
startdate	*date*		Start Date (day)
duration	*reltime*		Duration
rb	*resource_booking*	**FK**	Resource booking identifier
brieftime	*reltime*		Briefing time
debrieftime	*reltime*		Debrief time
actcode	*activity_set*	**FK**	Activity code for generated assignments
freedayactcode	*activity_set*	**FK**	Activity code for generated training free days
instrquallist	*string*		Instructor Qualifications needed, comma separated list
resourceneed	*resource_group*	**FK**	Resource group needed
task	*ground_task*	**FK**	Ground task identifier (after publication)

6.9.18 Entity course_participant

This entity represents a participant in the course.

course_participant			Course Participant
nr	***int***		Course participant identifier
c	***course***	**FK**	Cource identifier.
startofcourse	*date*		Start Date
endofcourse	*date*		End date
pos	*string*		The position of the participant
fromcrewgroup	*crew_filter*	**FK**	Current crew group of the pariticpant
tocrewgroup	*crew_filter*	**FK**	New crew group of the pariticpant
crew	*crew*	**FK**	Crew identifier.
si	*string*		Supplementary information

6.9.19 Entity course_participant_trip

This entity represents a course participant trip. Couples a course_block_trip to ca course_participant.

course_participant_trip			Course Participant Block trip
p	***course_participant***	**FK**	Course participant identifier
cbt	***course_block_trip***	**FK**	Course block trip identifier

6.9.20 Entity course_revision

This entity represents a course revision.

course_revision			Course Revisons
c	***course***	**FK**	Course identifier
nr	***int***		Revision number
changetime	*time*		Time of this revision
si	*string*		Supplementary information

6.9.21 Entity tr_effect_template

Defines a training effect. Defines an add or change to an entry in the specified table. If the keyvalue is empty there will only be an effect if the current value are END or PAUSE. If the keyvalue is non empty a new entry will be added to the specified table.

tr_effect_template — **Course effect template**

cbtemplate	***course_block_template***	**FK**	The course block template
name	***string(50)***		Crew filter type identifier
tablename	*string(50)*		The name of the table
fieldname	*string*		The name of the field to change
fieldvalue	*string*		The new value
calcvalue	*bool*		Boolean that defines if the value should be calculated
vfday	*tr_effect_day_set*	**FK**	What date to use for validfrom
vfadddays	*int*		Number of days to add to the validfrom date
vtday	*tr_effect_day_set*	**FK**	What date to use for validto
vtadddays	*int*		Number of days to add to the validto date
effectonold	*tr_effect_on_old_set*	**FK**	Action on current value
si	*string*		Supplementary information

6.9.22 Entity tr_effect

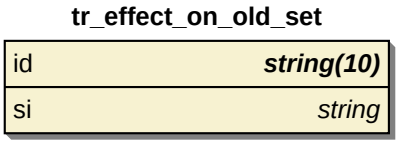
Defines a training effect. Defines an add or change to an entry in the specified table. If the keyvalue is empty there will only be an effect if the current value are END or PAUSE. If the keyvalue is non empty a new entry will be added to the specified table.

tr_effect — **Training effects**

cb	***course_block***	**FK**	The course block
name	***string(50)***		Crew filter type identifier
tablename	*string(50)*		The name of the table
fieldname	*string*		The name of the field to change
fieldvalue	*string*		The new value
calcvalue	*bool*		Boolean that defines if the value should be calculated
vfday	*tr_effect_day_set*	**FK**	What date to use for validfrom
vfadddays	*int*		Number of days to add to the validfrom date
vtday	*tr_effect_day_set*	**FK**	What date to use for validto
vtadddays	*int*		Number of days to add to the validto date
effectonold	*tr_effect_on_old_set*	**FK**	Action on current value
si	*string*		Supplementary information

6.9.23 Entity tr_effect_on_old_set

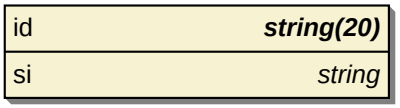
This entity represents the valid effect on the current settings. The following id values will be available: id = "LEAVE" Leave the current settings unchange id = "END" End the current settings by setting validto to before new validfrom date id = "PAUSE" End the current settings (see END) and start them again at the new validto date



tr_effect_on_old_set		Training effects on current settings
id	***string(10)***	Current effect identifier
si	*string*	Supplementary information

6.9.24 Entity tr_effect_day_set

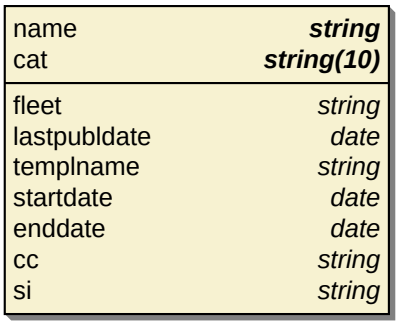
This entity represents the dyas that can be used to get effect period date from. The following id values will be available:: id = "COURSE_START" Use course start date id = "COURSE_END" Use course end date id = "BLOCK_TRIP_START" Use course block trip start date id = "BLOCK_TRIP_END" Use course block trip end date id = "INFINITE" Use default start and stop dates



tr_effect_day_set		Training Effect day set
id	***string(20)***	Current effect identifier
si	*string*	Supplementary information

6.9.25 Entity recurrent_training

This entity represents a recurrent training. Add also driver info.



recurrent_training		Recurrent course
name	***string***	The name of the recurrent training
cat	***string(10)***	Crew category identifier
fleet	*string*	Fleet name
lastpubldate	*date*	Last publication date
templname	*string*	Course template name
startdate	*date*	Start Date
enddate	*date*	End date
cc	*string*	Crew complement
si	*string*	Supplementary information.

6.9.26 Entity resource_booking

This entity represents a resource booking. A resourse booking can be part of a resource booking order. A resource booking can be connected to a training activity.

resource_booking			Resource Booking
id	***uuid***		Resource booking identifier
cc	*string*		Booked crew complement
code	*string*		Activity code
typ	*string*		Booking type (course,rec,maint,other)
refname	*string*		Reference name
carrier	*string*		Carrier name
rg	*resource_group*	**FK**	Resource group
bookdate	*date*		Date of booking
duration	*reltime*		Duration of booking
status	*resource_booking_status*	**FK**	Status of the booking.
r	*resource_def*	**FK**	Resource identifier
orderid	*resource_booking_order*	**FK**	Resource booking identifier.
slotstart	*time*		Start date
slotend	*time*		End date
acgd	*aircraft_ground_duty*	**FK**	Ac ground duty identifier (only if published)
si	*string*		Supplementary information

6.9.27 Entity resource_booking_order

This entity represents a resource booking order. It connects resource bookings to an order.

resource_booking_order			Resource Booking Order
id	***uuid***		Resource booking identifier
orderspec	*string*		Resource booking order identification string
bookstatus	*int*		Status of the order. 0=UNHANDLED, 1=SENT, 2=ANSWERED
task	*ground_task*	**FK**	Ground task identifier

6.9.28 Entity resource_booking_status

This entity represents a resource booking status. The following values can be used.
id = "OPEN"
id = "ASSIGNED"
id = "RESERVED"
id = "CONFIRMED"
id = "DENIED"

resource_booking_status		Resource Booking Status
id	***string***	Resource booking status
color	*string*	Booking status color
si	*string*	Supplementary information

6.9.29 Entity resource_group

This entity represents a resource group. Defines the type of resources available.

resource_group		Resource group
name	***string(50)***	The name of the resource type
loc	*string*	The location of the resource type
fleet	*string*	Defines the fleet the resource is valid for
cc	*string*	Crew complement need vector
unlimited	*bool*	True if unlimited resource group
si	*string*	Supplementary information

6.9.30 Entity resource_def

This entity represents a resource definition. Defines the physical resources that are available in a resource_group.

resource_def			Resource definition
name	***string(50)***		Resource name
rg	***resource_group***	**FK**	Resource group
validfrom	***time***		Validity period start
validto	*time*		Validity period end
firstslotstart	*reltime*		Start time (since midnight) for first slot of day
lastslotend	*reltime*		End time of the last slot of day
si	*string*		Supplementary information

6.9.31 Entity est_activity_group

This entity represents an establishment activity group.

est_activity_group			Activity Group
name	***string(15)***		Activity group characher id
cat	***string(10)***		Defines the category the activity group is valid for
color	*color*	**FK**	Activity group color
issupply	*bool*		Is supply activity group on/off
displayorder	*int*		Display order number
si	*string*		Supplementary information

6.9.32 Entity est_activity

This entity represents an establishment activity.

est_activity			Activity
name	***string(15)***		Activity characher id
cat	***string(10)***		Defines the category the activity is valid for
ag	*est_activity_group*	**FK**	The activity group to which the activity belongs
displayorder	*int*		Display order number
display	*bool*		Display on/off
calc	*bool*		Calculate on/off
strategy	*est_strategy_set*	**FK**	The calculation strategy
startdate	*string(20)*		The start date of the calculation strategy, syntax OD +- int or normal da
si	*string*		Supplementary information

6.9.33 Entity est_driver

This entity represents an establishment driver.

est_driver			Driver
name	***string(15)***		Driver id
cat	***string(10)***		Defines the category the driver is valid for
activity	*est_activity*	**FK**	The activity to which the driver belongs
calcorder	*int*		Calculate order number
calc	*bool*		Calculate on/off
dclass	*est_driver_class_set*	**FK**	The driver class where the behaviour is defined
dependact	*est_activity*	**FK**	The activity of which the driver depends on
dependstrat	*est_strategy_set*	**FK**	The strategy to apply on the activity defined in the activity_dependence
si	*string*		Supplementary information

6.9.34 Entity est_task_group

This entity represents an establishment task group.

est_task_group			CMP task group
code	***string(20)***		Est task code
cat	***string(10)***		Defines the category the task is valid for
activity	*est_activity*	**FK**	The activity to which the driver belongs
si	*string*		Supplementary information

6.9.35 Entity est_task

This entity represents an establishment task.

est_task			CMP task
code	***activity_set***	**FK**	Leg task code
cat	***string(10)***		Defines the category the task is valid for
taskgroup	*est_task_group*	**FK**	The task group to which the task belongs
si	*string*		Supplementary information

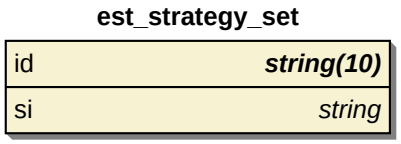
6.9.36 Entity est_driver_class_set

This entity represents an establishment driver class. Defines the python driver classes.

est_driver_class_set		Driver Class set
classname	***string***	The python class
cat	***string(10)***	Defines the category the class is valid for

6.9.37 Entity est_strategy_set

This entity represents a set of establishment calculation strategies. The following id values will be available:
id = "PLANNED"
id = "ASSIGNED"
id = "MAX"
id = "ADJUSTED"

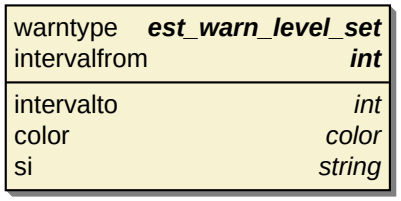


Establishment stragegy set

id	The strategy id
si	Supplementary information

6.9.38 Entity est_warn_level

This entity represents the establishment warning levels.

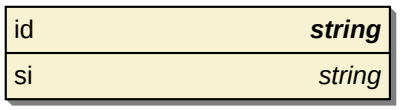


Warning levels establishment

warntype	FK	The name of leveltype
intervalfrom		Defines the from value
intervalto		Defines the to value
color	FK	Defines the color
si		Supplementary information

6.9.39 Entity est_warn_level_set

This entity represents the different establishment warningleveltypes. The following id values will be available:
id = "Count"
id = "Percentage"



Warning leveltype

id	The name of leveltype
si	Supplementary information

6.9.40 Entity est_filter_driver

The filter driver table. Defines settings for a driver that can be applied to a filter.

est_filter_driver			The filter driver table
driver	***est_driver***	**FK**	Driver this parameter is defined for
crewfilter	***crew_filter***	**FK**	Crew selection value is valid for
calcorder	*int*		Name of parameter
valid	*bool*		True if the driver is valid for the filter
validfrom	*time*		Start use of parameters
validto	*time*		End use of parameters
si	*string*		Supplementary information

6.9.41 Entity est_std_paramtable

The establishment standard parameter table. Defines a parameter for a driver and a period. Values for each day/week/month (depending of the resolution value) in the period will be created in the est_param_value table. If the issnake attribute is true will the values define a snake that should be roled out over the period. The length of the snake will be snakelength. The start date for the snake will be the day or the first day of the week or month, depending on the resulotion setting, that the validfrom date is in.

est_std_paramtable			The standard param table
driver	***est_driver***	**FK**	Driver this parameter is defined for
name	***string***		Name
crewfilter	***crew_filter***	**FK**	Crew selection value is valid for
calcorder	*int*		Name of parameter
validfrom	*time*		Start use of parameters
validto	*time*		End use of parameters
validstart	*string(6)*		OD +- int
validend	*string(6)*		OD +- int
issnake	*bool*		True if the values are snake values
valuetype	*est_param_type_set*	**FK**	
si	*string*		Supplementary information

6.9.42 Entity est_param_value

A value connected to a specified establishment standard driver est_std2_paramtable

est_param_value			The param value
param	***est_std_paramtable***	**FK**	The parameter identifier
validfromdate	***date***		The date, if resolution is higer than day, first day in period
paramvalue	*int*		The value

6.9.43 Entity est_param_type_set

This entity represents the establishment standard parameter type set. The following id values will be available:
id = "PERC"
id = "ABS"

est_param_type_set		The standard param table
param_type	***string(4)***	Percentage or absolute

6.9.44 Entity est_resolution_set

This entity represents the resoulotion of the establishment standard parameter value. The following id values will be available: id = "DAY" A value for each day in the period id = "WEEK" A value for each week in the period id = "MONTH" A value for each month in the period

est_resolution_set		Establishment parameter resolution set
id	***string(10)***	Establishment resulotion identifier
si	*string*	Supplementary information

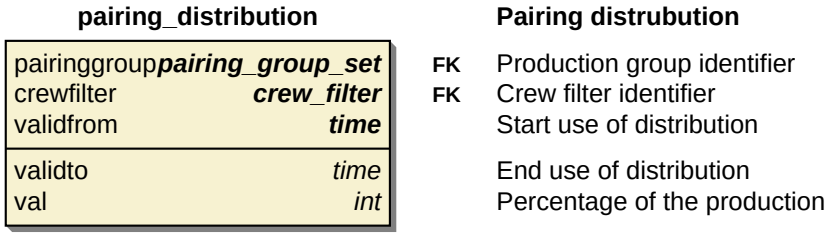
6.9.45 Entity pairing_group_set

This entity represents a production group.

pairing_group_set		Pairing group set
id	***string(20)***	Production group identifier
cat	***string(10)***	Defines the category the row is valid for
si	*string*	Supplementary information

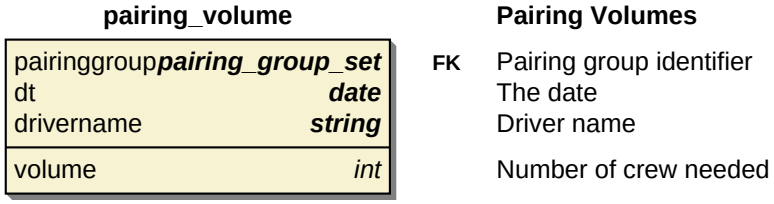
6.9.46 Entity pairing_distribution

This entity represents the distribution of production on crew filters. The specified percentage of the production from the production group should be done by the crew in the crew filter.



6.9.47 Entity pairing_volume

This entity represents the volumes (number of crew needed) obtained from pairingdata for a specified day and driver.



6.9.48 Extensions

This module also extends the crew_contract (6.7.13.1) and crew_position_set (6.6.27.1) entities.

6.10 Module air_planning

The air_planning module extends air_core with information needed for working with the data in a crew planning environment.

6.10.1 Extensions

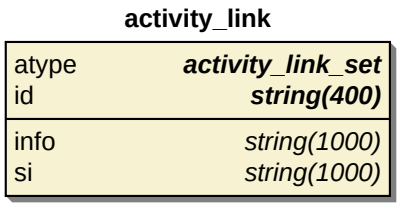
This module extends the flight_leg (6.6.1.2), crew (6.6.22.2), crew_flight_duty (6.6.23.1), crew_ground_duty (6.6.24.1) and crew_activity (6.6.25.1) entities.

6.11 Module air_tracking

The air_tracking module extends air_core with information needed for working with the data in a crew tracking environment.

6.11.1 Entity activity_link

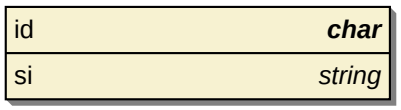
This entity represents a polymorphic link to some kind of activity.



	Link to activity	
FK	Target entity type	atype
	Encoded target entity primary key	id
	Encoded target entity properties for initial view	info
	Supplementary information for initial view	si

6.11.2 Entity activity_link_set

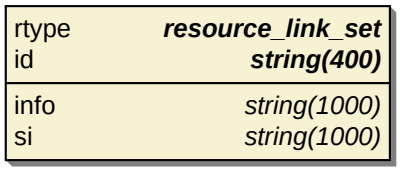
This entity lists allowed activity link types. See *activity_link*. Basic types are: "F" = Leg, "G" = Ground task, "P" = PACT, "D" = Duty, "T" = Trip.



Activity links	
Link type	id
Description	si

6.11.3 Entity resource_link

This entity represents a polymorphic link to some kind of resource.



	Link to resource	
FK	Target entity type	rtype
	Encoded target entity primary key	id
	Encoded target entity properties for initial view	info
	Supplementary information for initial view	si

6.11.4 Entity resource_link_set

This entity lists allowed resource link types. See *resource_link*. Basic types are: "O" = Open time, "C" = Crew, "A" = Aircraft, "N" = None.

resource_link_set		Resource links
id	***char***	Link type
si	*string*	Description

6.11.5 Entity track_alert

This entity represents a tracking alert.

track_alert			Tracking alert
link	***resource_link***	**FK**	Resource information
activity	***activity_link***	**FK**	Activity information
rule	***string(80)***		Violated Rule name
isactive	***bool***		Alert is currently active
status	***alert_status_set***	**FK**	Alert resolution and assigned status
alertgroup	***string***		Alert belongs to filter group
severity	***int***		Alert severity code
alerttime	***time***		When will it happen
deadline	*time*		Resolution deadline
snoozefrom	*time*		Snooze start time
snoozeto	*time*		Snooze expiry time
snoozeuser	*string*		Alert last snoozed by
limitval	*string*		Limit, including rule exception
actualval	*string*		Actual value
exceptionstarttime	*time*		Start time for rule exception
exceptionallowed	*bool*		True if the alert can have a rule exception
signature	***string(20)***		Signature of alert record, should be record creator
generatedtime	***time***		Alert first generated time
updatedtime	***time***		Record last changed
updatedby	***string***		Record last changed by
description	*string(200)*		Alert description, usually rule failure text
si	*string(1000)*		Supplementary information, entered by users

6.11.5.1 Extension from module sas_air_tracking

track_alert		Alert filter properties for SAS
region	*string(3)*	crew region at alerttime
maincat	*string(2)*	crew main cat at alerttime

6.11.6 Entity alert_status_set

This entity lists allowed resolution states for alerts. See *track_alert*. Basic types are: "O" = Open, "A" = Assigned, "R" = Resolved.

alert_status_set		Resolution states for alerts
id	***char***	Resolution
si	*string*	Description

6.11.7 Entity todo

This entity represents a todo item.

todo		ToDo item
id	***uuid***	ToDo key
duedate	*date*	Date when the item should be handled
si	*string(1000)*	Supplementary information, entered by users

6.11.8 Entity todo_activity

This entity represents a todo activity.

todo_activity			ToDoActivityConnection
todo	***todo***	**FK**	ToDo identifier
activity	***activity_link***	**FK**	Activity information

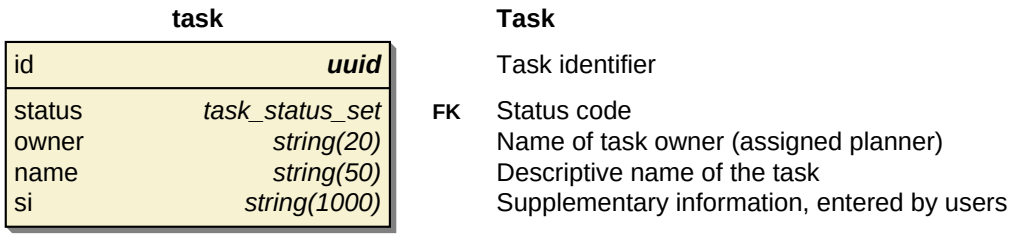
6.11.9 Entity todo_resource

This entity represents a todo resource.

todo_resource			ToDoResourceConnection
todo	***todo***	**FK**	ToDo identifier
link	***resource_link***	**FK**	Resource information

6.11.10 Entity task

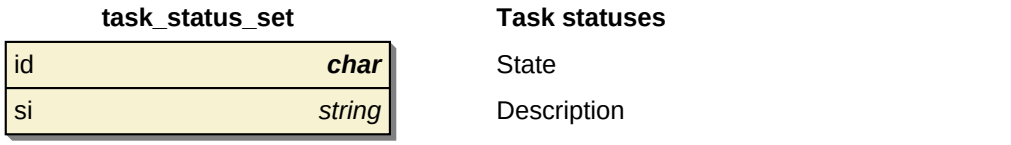
This entity represents a task.



task			Task
id	***uuid***		Task identifier
status	*task_status_set*	**FK**	Status code
owner	*string(20)*		Name of task owner (assigned planner)
name	*string(50)*		Descriptive name of the task
si	*string(1000)*		Supplementary information, entered by users

6.11.11 Entity task_status_set

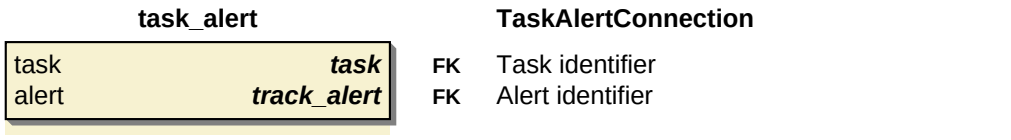
This entity lists allowed states for tasks. See *task*. Basic types are: "O" = Open, "R" = Rejected, "C" = Closed, "E" = Edited, "S" = Saved.



task_status_set		Task statuses
id	***char***	State
si	*string*	Description

6.11.12 Entity task_alert

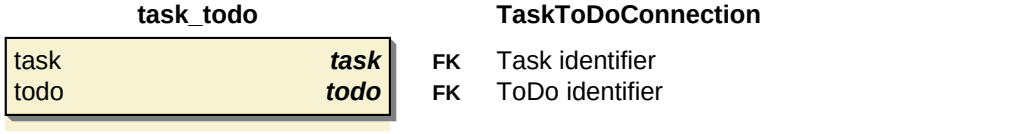
This entity represents a task alert.



task_alert			TaskAlertConnection
task	***task***	**FK**	Task identifier
alert	***track_alert***	**FK**	Alert identifier

6.11.13 Entity task_todo

This entity represents a task todo.



task_todo			TaskToDoConnection
task	***task***	**FK**	Task identifier
todo	***todo***	**FK**	ToDo identifier

6.11.14 Extensions

This module also extends the flight_leg (6.6.1.3), trip_flight_duty (6.6.19.1), trip_activity (6.6.21.1), crew_flight_duty (6.6.23.2) and crew_activity (6.6.25.2) entities.

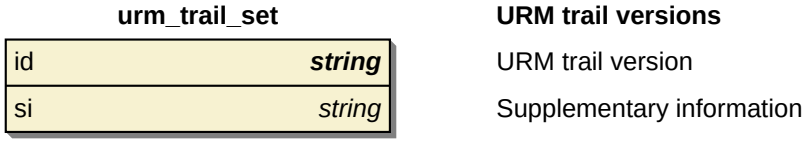
6.12 Module air_urm

This module, air_urm, defines entities and extensions required for working with URM (Unified Rave Model).

URM trail versions provide support for reasoning about roster changes in URM. This also covers delayed publication.

6.12.1 Entity urm_trail_set

This entity lists allowed URM trail version codes.



6.12.2 Extensions

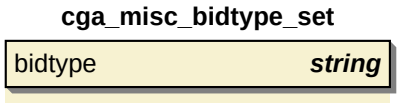
This module also extends the trip_flight_duty (6.6.19.2), trip_ground_duty (6.6.20.1), trip_activity (6.6.21.2), crew_flight_duty (6.6.23.3), crew_ground_duty (6.6.24.2), crew_activity (6.6.25.3), crew_trip (6.6.26.1), rotation_flight_duty (6.6.29.1), rotation_ground_duty (6.6.30.1), rotation_activity (6.6.31.1), aircraft_flight_duty (6.6.34.1), aircraft_ground_duty (6.6.35.1), aircraft_activity (6.6.36.2), equipment_flight_duty (6.6.41.1) and equipment_ground_duty (6.6.42.1) entities.

6.13 Module cmp_transition

The cmp_transition module contains the nessecary data model to support the Carmen CMS products (Manpower Planing) for transition.

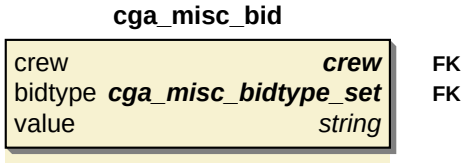
6.13.1 Entity cga_misc_bidtype_set

This entity ...



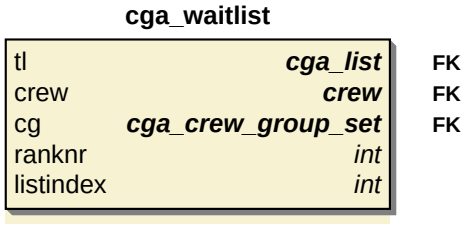
6.13.2 Entity cga_misc_bid

This entity ...



6.13.3 Entity cga_waitlist

This entity ...



6.13.4 Entity cga_assignment

This entity ...

cga_assignment

crew	***crew***	**FK**
assigndate	***date***	
cg	*cga_crew_group_set*	**FK**
approvedasfc	*bool*	
si	*string*	
fcguarantee	*date*	
forced	*bool*	
locked	*bool*	
newpromo	*bool*	
newlypromoted	*bool*	
notlh	*bool*	
reason	*string*	
ranknr	*int*	
remainfq	*bool*	
excluded	*bool*	
reasoncode	*int*	

6.13.5 Entity cga_bid

This entity ...

cga_bid

crew	***crew***	**FK**
prio	***int***	
bidtype	***cga_bid_set***	**FK**
cg	*string*	
origcg	*string*	

6.13.6 Entity cga_bid_set

This entity ...

cga_bid_set

id	***string***

6.13.7 Entity cga_list

This entity ...

cga_list

id	***int***
calculationdate	*date*
assignm	*int*
targetm	*int*

6.13.8 Entity cga_bidtype_set

This entity ...

cga_bidtype_set

bidtype	***cga_bid_set***	**FK**
value	***string***	

6.13.9 Entity cga_parm

This entity ...

cga_parm

id	***int***	ID
mxagefmv	*int*	maxAgeForMove
mxagefmvdtygrpex	*int*	maxAgeForMoveWhenDutyGroupExpires
mxagefmvngrntlhpassign	*int*	maxAgeForMoveWhenNotGrantedLHInPreviousAssignment
retirementage	*int*	retirementAge
mintimeindutygroup	*int*	minTimeInDutyGroup
mintimeindtygrpnewlypr	*int*	minTimeInDutyGroupWhenNewlyPromoted
transconvcost	*int*	transitionConversionCost
transupgrdcost	*int*	transitionUpgradeCost

6.13.10 Entity cga_rules

This entity ...

cga_rules

id	***int***	ID
agerule	*bool*	AgeRule
retirementrule	*bool*	RetirementRule
timeindutygrouprule	*bool*	TimeInDutyGroupRule
moverestrictionrule	*bool*	MovementRestrictionsRule
promotionqualifrule	*bool*	PromotionQualificationRule
dlicencerule	*bool*	DLicenceRule
lockbidrule	*bool*	LockBidRule
seniorityrule	*bool*	SeniorityRule
workraterule	*bool*	WorkRateRule
autompromotionsrule	*bool*	AutomaticPromotionsRule
manuallyexcludedrule	*bool*	ManuallyExcludedRule
forcedassignmentrule	*bool*	ForcedAssignmentRule
promotionguarantyrule	*bool*	PromotionGuarantyRule
moverestricbtwdgrule	*bool*	MovementRestrictionsBetweenDutyGroupsRule
basecaptsurplusrule	*bool*	BaseCaptainSurplusRule
basepilotseniordrule	*bool*	BasePilotSeniorityOrderRule
allowbasetranrule	*bool*	AllowBaseTransitionRule
cgrequirementsrule	*bool*	CrewGroupRequirementsRule

6.13.11 Entity crew_group_setting

This entity ...

crew_group_setting

cg	***string***		CrewGroup
color	*color*	**FK**	
colorallallotted	*int*		
colornewallotted	*int*		
colorlastyear	*int*		
colorwaitinglist	*int*		

6.13.12 Entity cga_p_cg_age

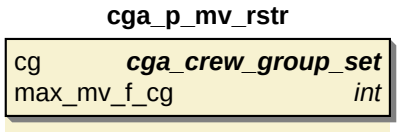
This entity ...

cga_p_cg_age

cg	***cga_crew_group_set***	**FK**	
max_age_mv	*int*		maxAgeForMove
max_age_mv_cg_exp	*int*		maxAgeForMoveWhenDutyGroupExpires
max_age_mv_n_lh	*int*		maxAgeForMoveWhenNotGrantedLHInPreviousAssignment

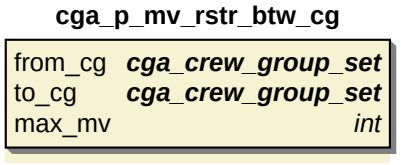
6.13.13 Entity cga_p_mv_rstr

This entity ...



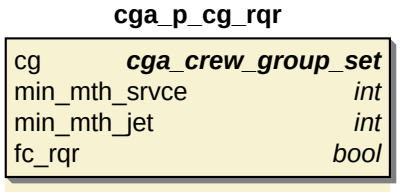
6.13.14 Entity cga_p_mv_rstr_btw_cg

This entity ...



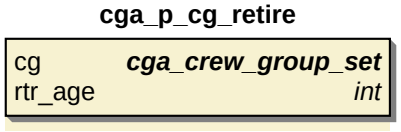
6.13.15 Entity cga_p_cg_rqr

This entity ...



6.13.16 Entity cga_p_cg_retire

This entity ...



6.13.17 Entity cga_p_t_d_mv_rtr

This entity ...

cga_p_t_d_mv_rtr

cg	***cga_crew_group_set***	**FK**
min_in_cg	*int*	
min_new_pro	*int*	

6.13.18 Entity cga_p_t_d_mv_bw_rtr

This entity ...

cga_p_t_d_mv_bw_rtr

from_cg	***cga_crew_group_set***	**FK**
to_cg	***cga_crew_group_set***	**FK**
min_in_cg	*int*	
min_new_pro	*int*	

6.13.19 Entity cga_p_cm_age

This entity ...

cga_p_cm_age

crew	***crew***	**FK**	
max_age_mv	*int*		maxAgeForMove
max_age_mv_cg_exp	*int*		maxAgeForMoveWhenDutyGroupExpires
max_age_mv_n_lh	*int*		maxAgeForMoveWhenNotGrantedLHInPreviousAssignment

6.13.20 Entity cga_p_cm_retire

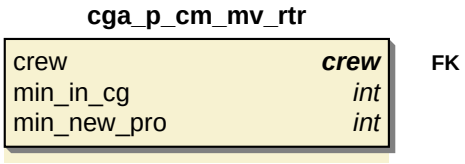
This entity ...

cga_p_cm_retire

crew	***crew***	**FK**	
rtr_age	*int*		maxAgeForMove

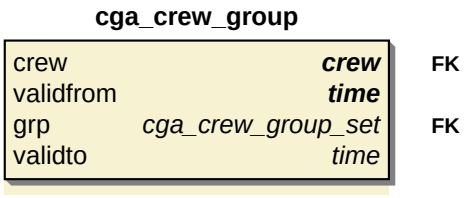
6.13.21 Entity cga_p_cm_mv_rtr

This entity ...



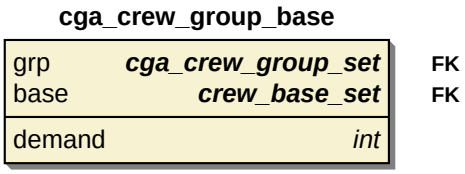
6.13.22 Entity cga_crew_group

This entity ...



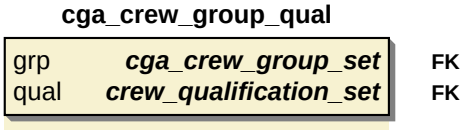
6.13.23 Entity cga_crew_group_base

This entity ...



6.13.24 Entity cga_crew_group_qual

This entity ...



6.13.25 Entity cga_crew_group_rank

This entity ...

cga_crew_group_rank

grp	***cga_crew_group_set***	**FK**
crewrank	***crew_rank_set***	**FK**

6.13.26 Entity cga_crew_group_set

This entity ...

cga_crew_group_set

id	***string***		
si	*string*		
ranknumber	*int*		
fqname	*string*		
demand	*int*		
baseassign	*bool*		
longhaul	*bool*		
settings	*cga_crew_group_setting*	**FK**	
name	*string*		
visible	*bool*		
cat	*string*		Defines the category

6.13.27 Entity cga_crew_group_setting

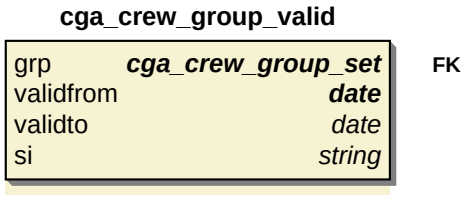
This entity ...

cga_crew_group_setting

cg	***string***		CrewGroup
color	*color*	**FK**	
colorallallotted	*int*		
colornewallotted	*int*		
colorlastyear	*int*		
colorwaitinglist	*int*		

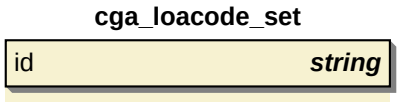
6.13.28 Entity cga_crew_group_valid

This entity ...



6.13.29 Entity cga_loacode_set

This entity ...



6.13.30 Extensions

This module also extends the crew_group_set (6.7.17.1) and crew_group_base (**??**) entities.

6.14 Module sas_accounts

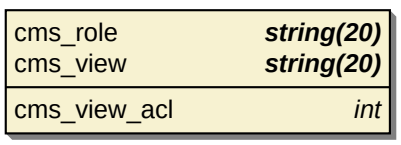
6.14.1 Extensions

This module extends the account_entry (6.7.31.1) entity.

6.15 Module sas_admin

6.15.1 Entity cms_views

This entity contains mapping between user role and security views.



Mappings between roles and views

CMS role (CARMROLE)
CMS security view

ACL, octal number where 0 = no rights, 2 = permission to read, 4 = per

6.15.2 Entity cms_view_objects

Listing of objects that are contained within a security view.

cms_view_objects		Listing of objects within a security view
cms_view	***string(20)***	CMS security view
cms_object_type	***string(20)***	CMS object type, e.g. TABLE
cms_object_name	***string(40)***	Name of the object

6.16 Module sas_air_aircraft

The basic aircraft related entities are defined in the air_crew module. This module extends some of these entities with SAS specific fields and also adds new SAS specific entities.

6.16.1 Extensions

This module extends the aircraft_type (6.6.39.2) entity.

6.17 Module sas_air_core

The fundamental planning entities are defined in the air_core module. This module extends some of these entities with SAS specific fields and also adds new SAS specific entities.

6.17.1 Entity flight_leg_delay

This entity contains detailed information about flight delays.

flight_leg_delay			Flight leg delays
leg	***flight_leg***	**FK**	Flight leg
seq	***uuid***		Sequence id
code	*string(2)*		Delay code
subcode	*string(2)*		Delay sub-code
duration	*string(8)*		Duration of delay (as a coded string)
reasontext	*string*		Delay reason text

6.17.2 Entity flight_leg_pax

flight_leg_pax

PAX numbers for flight leg

leg	***flight_leg***	**FK**	Flight leg
svc	***char***		Service class (C, M, Y)
ppax	*int*		PAX prognosis
bpax	*int*		PAX prognosis, based on booked PAX
apax	*int*		Actual PAX figures

6.17.3 Entity flight_message

This entity contains messages relating to a specific flight.

flight_message

fd	***string(20)***		Flight designator
udor	***date***		UTC day of origin
msgtype	***flight_message_set***	**FK**	Message type
logtime	***time***		Message creation time, UTC
msgtext	*string(720)*		Message text

6.17.4 Entity flight_leg_message

This entity contains messages relating to a specific flight leg.

flight_leg_message

leg	***flight_leg***	**FK**	Flight
msgtype	***flight_message_set***	**FK**	Message type
logtime	***time***		Message creation time, UTC
msgtext	*string(720)*		Message text

6.17.5 Entity flight_message_set

This entity contains the set of valid flight message types. Message types can currently be one of GEN (General Info), SICK (Sick Transport), VIP (VIP Passenger), SLOT (Slot Time).

flight_message_set

id	***string(20)***	Message type id
si	*string(40)*	Supplementary information

6.17.6 Entity bases

This entity contains information about the SAS bases and the stations included in those bases.

bases

station	***string(3)***	Station
base	*string(3)*	Base
si	*string(40)*	Supplementary information

6.17.7 Entity flight_owner

This entity contains information about the owner of all flights between two stations and for which dates this information is valid. This information comes from the FIA system. The selection of primary keys is made to fit the quality of the input data available.

flight_owner

fd	***string(8)***		Flight descriptor
adep	***airport***	**FK**	Airport of departure
ades	***airport***	**FK**	Airport of destination
validfrom	***date***		Valid from (including)
region	*string(3)*		Owner (region)
validto	*date*		Valid to (excluding)

6.17.8 Extensions

This module also extends the flight_leg (6.6.1.4), activity_group_period (6.6.16.2) and crew_base_set (6.6.12.1) entities.

6.18 Module sas_air_crew

The basic crew related entities are defined in the air_crew module. This module extends some of these entities with SAS specific fields and also adds new SAS specific entities.

6.18.1 Entity crew_region_set

Lists valid SAS regions of operation.

crew_region_set		crew_region_set
id	***string(6)***	Region id
name	*string(20)*	Name
si	*string(40)*	Supplementary information

6.18.2 Entity crew_attr

Additional SAS specific personal information about crew members. Overrules name/logname in crew table and adds validfrom/validto time frame as imported from HR-system.

crew_attr			Crew additional personal info
id	***crew***	**FK**	Crew identifier
validfrom	***date***		First valid date
validto	*date*		Last valid date
name	*string(40)*		Last name
forenames	*string(40)*		First name
logname	*string(40)*		SAS name
personalno	*string(12)*		Personal number
nationality	*string(3)*		Nationality
nationality2	*string(3)*		Second nationality
maritalstatus	*string(10)*		Marital status
si	*string(40)*		Supplementary information

6.18.3 Entity pattern_set

This entity contains patterns used in crew contracts.

pattern_set		Contract patterns
id	***int***	Pattern id
fixcyclestart	*date*	Fixed pattern cycle start date
allowedstartdays	*string*	Allowed start days, comma-separated list
noofdays	*int*	Number of days in pattern (length of pattern
noofproddays	*int*	Number of production days in pattern (also used in "VG"-patterns for co
descshort	*string(20)*	Short description

6.18.4 Entity pattern_acts

This entity contains the day types/activities used in contract patterns.

pattern_acts — **Pattern activities**

pattern	***pattern_set***	**FK**	Pattern id
startpos	***int***		Start position (first day=1)
daytype	*string(3)*		Day type (FPXDV)
activity	*string(4)*		Activity code
endpos	*int*		End position

6.18.5 Entity ac_qual_map

This entity maps AcType and AOC for a flight leg into the needed qualification for a crew-member

ac_qual_map — **ACQUAL mappings from AcType and AOC**

ac_type	***string***	AcType
aoc	***string***	AOC
ac_qual_fc	*string*	ACQUAL for FC
ac_qual_cc	*string*	ACQUAL for CC

6.18.6 Entity crew_landing

This entity contains data about performed landings.

crew_landing — **Crew landings**

leg	***flight_leg***	**FK**	Landing flight leg
crew	***crew***	**FK**	Crew identifier
airport	***airport***	**FK**	Landing airport
nr_landings	*int*		Number of landings for crew on airport
activ	*bool*		Active during landing

6.18.7 Entity crew_sim_landing

This entity contains data about performed simulator landings.

crew_sim_landing			PREL: Crew simulator landings
activity	***ground_task***	**FK**	Landing simulator session
crew	***crew***	**FK**	Crew identifier
airport	***airport***	**FK**	Landing airport
nr_landings	*int*		Number of landings for crew on airport
activ	*bool*		Active during landing

6.18.8 Entity crew_rest

This entity contains data about crew rest periods.

crew_rest			Crew rest
crew	***crew***	**FK**	Crew identifier
flight	***flight_leg***	**FK**	Flight leg
reststart	*time*		Rest start
restend	*time*		Rest end
si	*string(40)*		Supplementary Information

6.18.9 Entity country_req_docs

This entity contains data about documents required for entering a country.

country_req_docs			Required documents for staying entering a country
country	***country***	**FK**	Country
doc	***crew_document_set***	**FK**	Document type
validfrom	***date***		Valid from
validto	*date*		Valid to
si	*string(40)*		Supplementary information

6.18.10 Entity preferred_hotel_exc

This entity describes exceptions to the preferred hotel.

preferred_hotel_exc			Preferred hotel exception
airport	***airport***	**FK**	Airport
region	***crew_region_set***	**FK**	Region
category	***char***		Crew category (cabin 'C' or flight 'F')
airport_hotel	***bool***		Indicates if hotel is located at airport
validfrom	***date***		First valid date
validto	*date*		Last valid date
hotel	***hotel***	**FK**	Hotel
si	*string(40)*		Supplementary information

6.18.11 Entity hotel_transport

This entity describes local transport from an airport to an hotel.

hotel_transport			Hotel transport
airport	***airport***	**FK**	Airport
hotel	***hotel***	**FK**	Hotel
validfrom	***date***		First valid date
validto	*date*		Last valid date
duration	*reltime*		Local transport time
cost	*int*		Cost for local transport
si	*string(40)*		Supplementary information

6.18.12 Entity crew_dental_info

List containing the contact information to each crew member's dentist. The table is maintained from the Dave Explorer where people with the role administrator has the right to update the table.

crew_dental_info			Crew Dental Info
crew	***crew***	**FK**	Crew identifier
validfrom	***date***		First valid date
validto	*date*		Last valid date
dentistname	*string(40)*		Dentist name
street	*string(40)*		Street address
city	*string(40)*		City
state	*string(40)*		State or region
postalcode	*string(20)*		Postal code
country	*string(40)*		Country of residence
phone	*string(20)*		Phone number
si	*string(40)*		Supplementary information

6.18.13 Entity crew_not_fly_with

List of crew that are prohibited from flying together. List is checked both ways so double entries are not needed.

crew_not_fly_with			Crew prohibited to fly together
crew1	***crew***	**FK**	Crew identifier 1
crew2	***crew***	**FK**	Crew identifier 2
validfrom	***time***		First valid date
validto	*time*		Last valid date
si	*string(40)*		Gossip about reason

6.18.14 Entity crew_passport

Additional SAS information for crew passport names.

crew_passport			Crew passport name
crew	***crew***	**FK**	Crew identifier
validfrom	***time***		First valid date
validto	*time*		Last valid date
firstname	*string*		Crews passport first name
lastname	*string*		Crews passport last name
si	*string(40)*		Gossip about reason

6.18.15 Entity crew_relatives

Additional SAS information for crew relatives.

crew_relatives			Crew relatives
crew	***crew***	**FK**	Crew identifier
subtype	***string(3)***		3=contactperson1, 4=contactperson2
validfrom	***time***		First valid date
validto	*time*		Last valid date
co_name	*string*		Name of contact person
street	*string*		Address of contact person
appartment	*string*		Evt. appartment number of contact person, only used in DK
postalcode	*string*		ZIP code
city_quarter	*string*		Evt. city quarter, only used in DK
city	*string*		City
phone	*string*		Contact persons primary phone number
com_type1	*string*		Type of alternate communication number 1
com_type2	*string*		Type of alternate communication number 2
com_type3	*string*		Type of alternate communication number 3
com_type4	*string*		Type of alternate communication number 4
com_number1	*string*		Alternate communication number 1
com_number2	*string*		Alternate communication number 2
com_number3	*string*		Alternate communication number 3
com_number4	*string*		Alternate communication number 4
si	*string(40)*		Gossip about reason

6.18.16 Entity crew_qual_restr

Restrictions on qualifications.

crew_qual_restr			Crew qualification restriction
crewqual	***crew_qualification***	**FK**	Crew qualification
restr	***crew_restriction_set***	**FK**	Restriction
validfrom	***time***		First valid date
validto	*time*		Last valid date
si	*string(40)*		Supplementary information

6.18.17 Extensions

This module also extends the crew (6.6.22.3), crew_employment (6.7.6.1), crew_contract (6.7.13.2), crew_qualification (6.7.19.1), crew_contract_set (6.7.14.1), crew_contract_valid (6.7.15.1), crew_qualification_ (6.7.20.1), crew_restriction_set (6.7.22.1), hotel (6.7.27.1) and crew_address (6.7.4.1) entities.

6.19 Module sas_air_tracking

The alerts are defined in the air$_{t}rackingmodule.Thisfileaddscolumnswithpropertiesforfilteringalerts.$

6.19.1 Extensions

This module extends the track_alert (6.11.5.1) entity.

6.20 Module sas_annotations

6.20.1 Entity annotation_set

This entity lists supported annotation types.

annotation_set		Annotation set
code	***string(2)***	Annotation Code
descript	*string(40)*	Description
forcrew	*bool*	for Crew or Flight
incct	*bool*	In Tracking
inccr	*bool*	In Rostering
hasprop	*bool*	Has Property
isvisible	*bool*	Is Visible in Studio
validfrom	*date*	Start Date
validto	*date*	End Date

6.20.2 Entity crew_annotations

This entity list all available annotations for the crew.

crew_annotations			Crew Annotation
crew	***crew***	**FK**	Crew ID
seqnr	***int***		Sequence Number
entrytime	*time*		Entry Time
code	*annotation_set*	**FK**	Code
property	*int*		Property
validfrom	*date*		First Valid Date
validto	*date*		Last Valid Date
isvisible	*bool*		Is Visible in Studio
text	*string*		Annotation Text
username	*string(10)*		User Name

6.21 Module sas_base_breaks

6.21.1 Entity crew_flight_base_break

...

crew_flight_base_break			Base breaks
crew_duty	***crew_flight_duty***	**FK**	Crew Flight
last_in_trip	*bool*		
first_in_trip	*bool*		

6.22 Module sas_bought_days

6.22.1 Entity bought_days

This entity is used to store data for days bought by the tracker.

bought_days			Bought freedays, compensation and vacation days
crew	***crew***	**FK**	Crew Identifier
day_date	***date***		Date for bought day
day_type	*string(10)*		Type for bought day
uname	*string(10)*		Record commiter
si	*string(40)*		Suplementary information

6.23 Module sas_calloutlist

6.23.1 Entity callout_list

This entity is the callout list.

callout_list			Callout List
crew	***crew***	**FK**	Crew ID
standbystart	***time***		Standby Start Time
callouttime	*time*		Callout Time
transport	*reltime*		Transport to Standby

6.24 Module sas_checkin

6.24.1 Entity cio_event

This entity contains crew check-in/out events, e.g. card swipe or check-in via crew portal. The event is the same for both check-in and check-out. The proper action is calculated and updated status is written to cio_status.

Some extra information are also written to the cio_event table to facilitate debugging and possibility to list last check-in events with associated actions.

cio_event — **Raw data for cio-events (crew login)**

crew	***crew***	**FK**	Crew which cio event belongs to
ciotime	***time***		UTC time of check-in/out event
assisted	*bool*		Assisted or automatic event
dutystartutc	*time*		Duty associated with event
seqno	*int*		Seq.nr of duty-pass on UTC day
statuscode	*int*		Short note on what happened

6.24.2 Entity cio_status

This entity contains crew check-in status on a certain duty-day.

Non existant entry is not checked in.
Status 0 is an undoed check-in.

Status 1 is crew checked in
Status 2 is crew checked out
Status 3 is crew re-checked in

cio_status — **Current status for crew on a certain duty day**

crew	***crew***	**FK**	Crew which cio status belongs to
utcday	***time***		UTC day of scheduled departure of first leg in duty
seqno	***int***		Seq.nr of duty-pass on UTC day
status	*int*		Check-in status
ciotime	*time*		Time of last changed status, i.e. cio-event

6.24.3 Entity cio_override

This entity contains crew check-in/out times on a certain duty-day.

changetype determines what cio time shall be overwritten.

changetype 1 is check-in changetype 2 is check-out

cio_override

crew	***crew***
utcday	***time***
seqno	***int***
changetype	***int***
ciotime	*time*
si	*string(40)*

Changed cio-times on a certain duty day

FK	Crew which cio status belongs to
	UTC day of scheduled departure of first leg in duty
	Seq.nr of duty-pass on UTC day
	Changed cio-event time
	New time for cio-event
	Tracker comment on changed checkin time

6.25 Module sas_crew_meals

6.25.1 Entity meal_code

Meal code entities.

meal_code

code	***char***
description	*string(30)*
si	*string(40)*

Meal codes.

	Meal code identifier
	Meal type description
	Supplementary information.

6.25.2 Entity meal_supplier

Meal supplier information.

meal_supplier		Meal suppliers (caterers).
supplier_id	***string(8)***	Supplier identifier.
valid_from	***date***	Valid from date.
company	*string(30)*	Company name.
department	*string(20)*	Department name.
station	*string(3)*	Supplier station.
opening_time	*reltime*	Opening time
closing_time	*reltime*	Closing time
phone1	*string(20)*	Phone number 1.
phone2	*string(20)*	Phone number 2.
fax	*string(20)*	Fax number.
email	*string(60)*	E-mail adress.
valid_to	*date*	Valid to date.
pdf	*bool*	PDF order format
xml	*bool*	XML order format
preferred	*bool*	Preferred supplier flag.
si	*string(40)*	Supplementary information.

6.25.3 Entity meal_customer

Meal customer information.

meal_customer			Meal customers.
company	***crew_company_set***	**FK**	Company name.
region	***crew_region_set***	**FK**	Customer region.
department	*string(20)*		Department name.
phone1	*string(20)*		Phone number 1.
phone2	*string(20)*		Phone number 2.
fax	*string(20)*		Fax number.
email	*string(40)*		E-mail adress.
si	*string(40)*		Supplementary information.

6.25.4 Entity meal_order

Meal order/forecast entities.

meal_order			Meal order/forecast information.
order_number	***int***		Order/Forecast number.
forecast	***bool***		Indicates if order or forecast.
order_date	*date*		Order date.
supplier	*meal_supplier*	**FK**	Supplier reference.
customer	*meal_customer*	**FK**	Customer reference.
load_station	*string(5)*		Meal load station.
sent	*bool*		Indicates if order is sent to supplier and customer.
cancelled	*bool*		Indicates if order has been cancelled.
from_date	*date*		Start of order/forecast.
to_date	*date*		End of order/forecast.
username	*string(30)*		User name of order creator.

6.25.5 Entity meal_order_line

Meal order/forecast line entities.

meal_order_line			Meal order/forecast line information.
order	***meal_order***	**FK**	Order reference.
load_flight	***flight_leg***	**FK**	Meal load flight.
cons_flight	***flight_leg***	**FK**	Meal consumption leg.
main_cat	***char***		Flight deck (F) or cabin crew (C).
meal_code	***meal_code***	**FK**	Meal code.
amount	*int*		Amount of meals to be served.

6.25.6 Entity meal_consumption_code

Meal consumption code mappings.

meal_consumption_code			Meal consumption code mappings.
region	***crew_region_set***	**FK**	Region.
main_cat	***char***		Flight deck (F) or cabin crew (C).
stc	***char***		Service type code.
meal_code	***string(2)***		Meal code defined in CCR.
start_time	***reltime***		Interval start time.
valid_from	***date***		Valid from date.
end_time	*reltime*		Interval end time.
valid_to	*date*		Valid to date.
cons_code	*meal_code*	**FK**	Meal consumption code.
si	*string(40)*		Supplementary information.

6.25.7 Entity meal_cons_correction

Meal consumption correction.

meal_cons_correction			Meal consumption correction.
flt_nr	***string(4)***		Flight number or '*' for all flights.
stn_from	***string(3)***		Departure station
stn_to	***string(3)***		Arrival station
valid_from	***date***		Valid from date
corr_type	***char***		Correction type - A, N, O, S
valid_to	*date*		Valid to date
corr_code	*meal_code*	**FK**	Meal code
valid_day1	*bool*		Valid Monday
valid_day2	*bool*		Valid Tuesday
valid_day3	*bool*		Valid Wednesday
valid_day4	*bool*		Valid Thursday
valid_day5	*bool*		Valid Friday
valid_day6	*bool*		Valid Saturday
valid_day7	*bool*		Valid Sunday
si	*string(40)*		Supplementary information

6.25.8 Entity meal_load_correction

Meal consumption correction.

meal_load_correction		Meal load correction.
cons_flt	***int***	Flight number
cons_stn	***string(3)***	Departure station
valid_from	***date***	Valid from date
valid_to	*date*	Valid to date
load_flt	*int*	Flight number
load_stn	*string(3)*	Departure station
valid_day1	*bool*	Valid Monday
valid_day2	*bool*	Valid Tuesday
valid_day3	*bool*	Valid Wednesday
valid_day4	*bool*	Valid Thursday
valid_day5	*bool*	Valid Friday
valid_day6	*bool*	Valid Saturday
valid_day7	*bool*	Valid Sunday
si	*string(40)*	Supplementary information

6.25.9 Entity meal_airport

Airport information related to crew meal. Opening times and min connection time.

meal_airport — **Airport information related to crew meal**

station	***string(3)***		Station code
region	***crew_region_set***	**FK**	Region
valid_from	***date***		Valid from date
valid_to	*date*		Valid to date
mealstop_mincnx	*reltime*		Minimum connection time for meal stop
rest_open	*reltime*		Restaurant opening time
rest_close	*reltime*		Restaurant closing time
si	*string(40)*		Supplementary information

6.26 Module sas_crew_needs

6.26.1 Entity crew_need_jarops

This entity contains the crew need as required per JAROPS. An empty type field will be the wildcard entry for lookups based only on acfamily.

crew_need_jarops — **Crew requirements on flights (JAROPS)**

acfamily	***string(10)***	Aircraft family, as defined in aircraft_type maintype
actype	***string(10)***	Aircraft type, as defined in aircraft_type (SAS custom)
pos1	*int*	Crew need in position
pos2	*int*	Crew need in position
pos3	*int*	Crew need in position
pos4	*int*	Crew need in position
pos5	*int*	Crew need in position
pos6	*int*	Crew need in position
pos7	*int*	Crew need in position
pos8	*int*	Crew need in position
pos9	*int*	Crew need in position
pos10	*int*	Crew need in position

6.26.2 Entity crew_need_service

This entity contains the crew service level need. The documentation is in the crc module crew$_{pos}$.

crew_need_service			Crew service need
region	***string(10)***		Region, i.e. SK, SKD, SKN or SKS
area	***string(10)***		Departure area
service	***string(2)***		Service type, e.g. C-charter, J-normal servic
prio	***int***		Priority order depending on region, area and service type
keytype	***string(10)***		AcType, Flight or Route
keyvalue	***string(10)***		Value for type, e.g. M87, 1417, OSL-LHR
weekdays	***string(7)***		Valid weekdays, e.g. 12345-7
sdate	***time***		Start date, flight should depart inside interval
edate	*time*		End date, flight should depart inside interval
pos6	*int*		Additional crew need for position 6
pos7	*int*		Additional crew need for position 7

6.26.3 Entity crew_need_exception

This entity contains exceptions to the crew service level need. The documentation is in the crc module crew$_{pos}$.

crew_need_exception			Crew service need exception
flight	***flight_leg***	**FK**	The flight to which the exception is applied
pos6	*int*		Additional crew need for position 6, or -1 to use service need
pos7	*int*		Additional crew need for position 7, or -1 to use service need

6.27 Module sas_financial

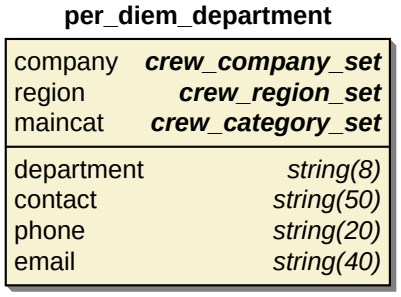
6.27.1 Entity per_diem_compensation

This entity contains per diem compensation amounts for different stops. The compensation is mapped for stop countries and home countries. Meal reduction is also included here.

per_diem_compensation			Per diem compensation for stops.
stop_country	***string(4)***		Stop country code or meal reduction (MEAL)
home_country	***string(2)***		Home country code
maincat	***crew_category_set***	**FK**	Crew category Flight (F) or Cabin (C)
valid_from	***date***		Valid from date
valid_to	*date*		Valid to date
compensation	*int*		Per diem compensation amount
currency	*string(3)*		Compensation currency

6.27.2 Entity per_diem_department

This entity holds contact information about the departments that handle per diem.

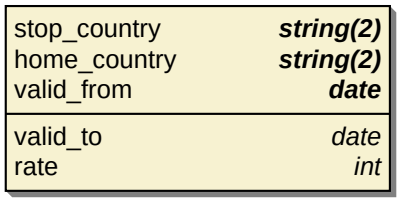


SAS Departments to handle per diem.

company	**FK**	Company code.
region	**FK**	Region code.
maincat	**FK**	Crew category Flight (F) or Cabin (C)
department		Department name.
contact		Contact person
phone		Contact phone
email		Contact e-mail

6.27.3 Entity per_diem_tax

This entity contains tax deduct amounts for per diem for different stops. Empty string for stop country is wild card.

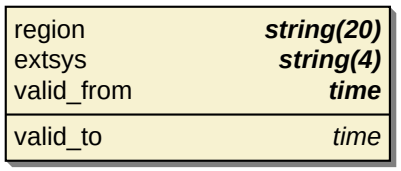


Per diem tax deduction amounts.

stop_country	Stop country code or default country (empty string)
home_country	Home country code
valid_from	Valid from date
valid_to	Valid to date
rate	Per diem tax deduct amount

6.27.4 Entity salary_region

Mapping between Home base/Region and Salary system.



Home base or Region - Salary system

region	Region or Home base
extsys	The Salary system that supports the Region or Home base
valid_from	Date from when the definition is valid
valid_to	Date until when the definition is valid

6.27.5 Entity salary_admin_code

List of available administrative codes. Such a code could for instance be: 'Test run', 'Retro', 'Export'.

salary_admin_code			Available administrative codes. Such a code code for instance be:
admcode	***string(2)***		Administrative code, e.g. (Test-Run)
description	*string(40)*		Description of the code

6.27.6 Entity salary_article

Mapping between Article ID in our system and in the upstream systems.

salary_article			Mapping between Article ID in the downstream system and in the
extsys	***string(4)***		The Salary system where the article is registered
extartid	***string(20)***		Article ID definition used within the external Salary system
valid_from	***time***		Date from when the definition is valid
valid_to	*time*		Date until when the definition is valid
intartid	*string(20)*		Article identification used within our system
note	*string(60)*		Note or comment

6.27.7 Entity salary_run_id

Meta data about the salary run. This entity is updated after each run.

salary_run_id			Contains information about the Run
runid	***int***		Integer, next run will have RunId = max(RunId) + 1
starttime	*time*		Date and time of the Run
runtype	*string(10)*		Type of run, can be 'Per Diem' or 'Overtime'
admcode	*salary_admin_code*	**FK**	Administrative code, e.g. (Test-Run)
selector	*string(40)*		An expression that would give the same selection (to re-create the Run
extsys	*string(4)*		Code identifying the salary system (upstream system)
firstdate	*date*		Start date of the salary period
lastdate	*date*		End date of the salary period
note	*string(60)*		Note or comment

6.27.8 Entity salary_basic_data

Contains results of the calculations. Basic data that is updated by the salary run.

salary_basic_data			Basic data to be uploaded to the upstream system
runid	***salary_run_id***	**FK**	Metadata for the Run
extperkey	***string(5)***		Extperkey = SAS Employee number, can change with time
extartid	***string(20)***		Article ID definition used within the Salary system
crewid	*crew*	**FK**	Crew ID
amount	*int*		An amount times 100 (100*amount)

6.27.9 Entity salary_mail_rcpt

Mail addresses to where the summary information after a salary run is sent.

salary_mail_rcpt		Mail recipients of summary information.
extsys	***string(4)***	Code identifying the salary system (upstream system)
mailaddr	***string(80)***	Email address to recipient.

6.28 Module sas_hotel_bookings

6.28.1 Entity hotel_booking

Hotel bookings for each crew member. Open trips does not refer to a crew member.

hotel_booking			Hotel booking information.
id	***uuid***		Crew booking id
crew	*crew*	**FK**	Crew used for booking.
hotel	*hotel*	**FK**	Hotel reference.
checkin	*date*		Hotel check-in date local time.
checkout	*date*		Hotel check-out date local time.
nights	*int*		Number of nights
rooms	*int*		Number of rooms
customer	*hotel_customer*	**FK**	Customer reference.
arrival_flight	*flight_leg*	**FK**	Crew arrives with this flight.
arrival_passive	*bool*		Indicates if flight is passive
departure_flight	*flight_leg*	**FK**	Crew departs with this flight.
departure_passive	*bool*		Indicates if flight is passive
sent	*bool*		Indicates if reservation is sent.
cancelled	*bool*		Indicates if reservation is cancelled.

6.28.2 Entity hotel_customer

Contact information for the different sas regions regarding hotel bookings.

hotel_customer

region	***crew_region_set***	
name	*string(60)*	
careof	*string(60)*	
department	*string(60)*	
postalcode	*string(20)*	
city	*string(60)*	
country	*string(60)*	
contact	*string(60)*	
phone	*string(40)*	
fax	*string(40)*	
email	*string(40)*	
si	*string(40)*	

Hotel customer information.	
FK	Customer region.
	Name of hotel customer.
	Customer care of.
	Department name.
	Post code.
	City.
	Country.
	Hotel booking contact.
	Phone number.
	Fax number.
	E-mail adress.
	Supplementary information.

6.29 Module sas_integration

6.29.1 Entity sas_40_1_cbr

The Crew Baggage Reconciliation interface needs to remember what was sent in the latest message for a crew member. This entity will store the full record for each crew.

sas_40_1_cbr

extperkey	***string(5)***
entrydate	*date*
records	*int*
msg	*string(261)*

Previous Crew Baggage Reconciliation message.
Numeric part of SAS external perkey.
Date of last submitted message to the external system.
Number of records in the message (0-9).
Last submitted message (encoded according to 40.1).

6.30 Module sas_leave_parameters

6.30.1 Entity leave_popular_periods

leave_popular_periods

cat	***string(10)***	
base	***crew_base_set***	**FK**
company	***crew_company_set***	**FK**
startdate	***date***	
enddate	*date*	
mindays	*int*	

Popular periods definitions.

	Main category
FK	Crew base
FK	Crew company
	Startdate
	Enddate
	Min days in period to be considered

6.30.2 Entity crew_type_set

crew_type_set

id	***string***
si	*string*

Different categories for crew, definitions done in rave.

	Name of category

6.30.3 Entity leave_period

leave_period

cat	***string(10)***	
base	***crew_base_set***	**FK**
company	***crew_company_set***	**FK**
crewtype	***crew_type_set***	**FK**
lseason	***leave_season***	**FK**
startdate	*date*	
enddate	*date*	

Leave planning season including dates.

	Main category
FK	Crew base
FK	Crew company
FK	Crew type
FK	Leave Season of planning
	Starttime
	Enddate

6.30.4 Entity leave_rotation_set

leave_rotation_set — **Set of leave rotations, summer and winter**

name	***string***		Name of rotation
season	*leave_season_set*	**FK**	Valid in yearly season
si	*string*		

6.30.5 Entity leave_rotation_order

leave_rotation_order — **Defines how rotations follows each other**

rotation	***leave_rotation_set***	**FK**	Rotation
nextrotation	*leave_rotation_set*	**FK**	Rotation that follows

6.30.6 Entity leave_crew_rotation

leave_crew_rotation — **Holds the season a crew starts a rotation**

crew	***crew***	**FK**	Crew
lseason	***leave_season***	**FK**	Season rotation starts from
rotation	*leave_rotation_set*	**FK**	Rotation that season

6.30.7 Entity leave_actual_rotation

leave_actual_rotation — **Holds the resulting rotation after temp changes**

crew	***crew***	**FK**	Crew
lseason	***leave_season***	**FK**	Season rotation starts from
rotation	*leave_rotation_set*	**FK**	Rotation that season

6.30.8 Entity leave_points

leave_points

Holds the vacation points given in specific periods

cat	***string(10)***		Main category
base	***crew_base_set***	**FK**	Crew base
company	***crew_company_set***	**FK**	Crew company
startdate	***time***		Start of period
enddate	*time*		End of period
points	*int*		The points for day

6.30.9 Entity leave_comparer_set

leave_comparer_set

Set of things that rank bids

name	***string(20)***	Name
si	*string*	Supplementary info

6.30.10 Entity leave_comparer

leave_comparer

Comparer for vacation bids

cat	***string(10)***		Main category
crewfilter	***crew_filter***	**FK**	Crew filter reference
season	***leave_season_set***	**FK**	Yearly leave season
comparer	***leave_comparer_set***	**FK**	Comparer
sortorder	*int*		Rank of comparer

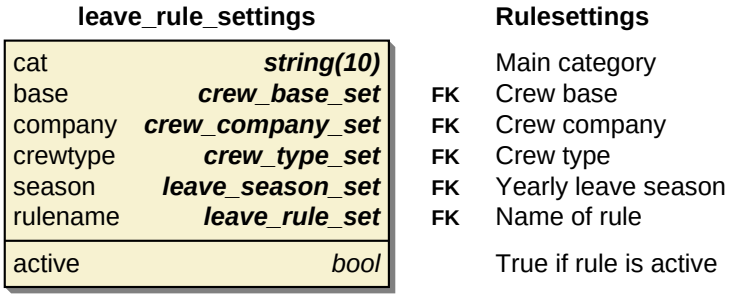
6.30.11 Entity leave_rule_set

leave_rule_set

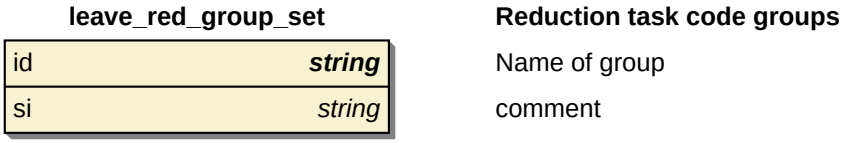
Vacation Ruleset

name	***string***	Rule name string
si	*string*	Description

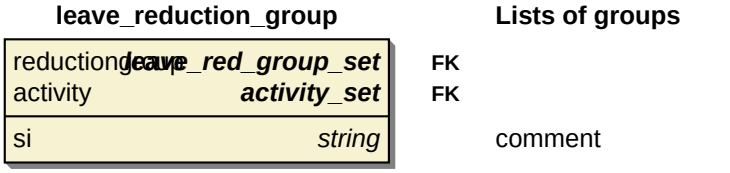
6.30.12 Entity leave_rule_settings



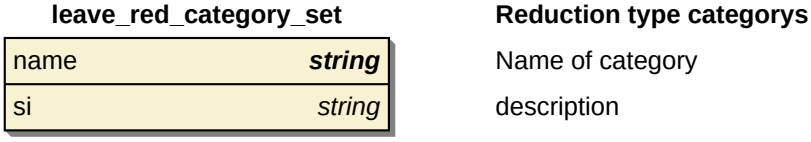
6.30.13 Entity leave_red_group_set



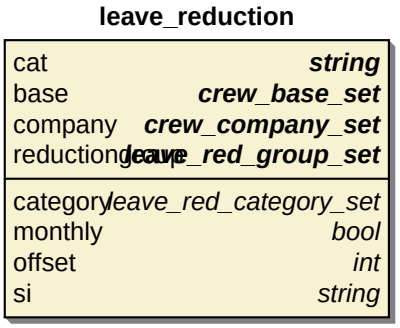
6.30.14 Entity leave_reduction_group



6.30.15 Entity leave_red_category_set

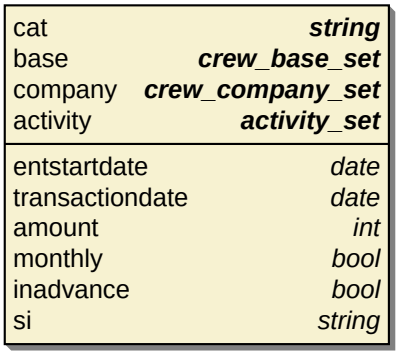


6.30.16 Entity leave_reduction



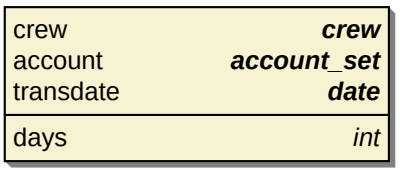
Parameter for Rave		
Main cat		
	FK	
	FK	
	FK	
	FK	
comment		

6.30.17 Entity leave_entitlement



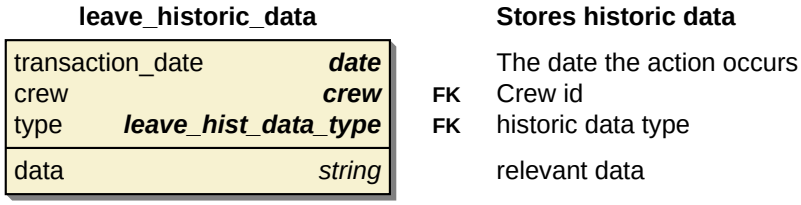
Parameter for entitlement		
Main cat		
	FK	
	FK	
	FK	
Year not important		
Year not important		
The amount of entitlement		
Entitlement each month		
Entitlement year same as consumption year		
comment		

6.30.18 Entity leave_manual_entry

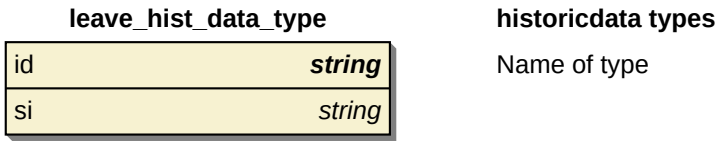


Manual entrys of leave accounts		
Crew id	**FK**	
Transaction account	**FK**	
Date of transaction		
Amount (x100)		

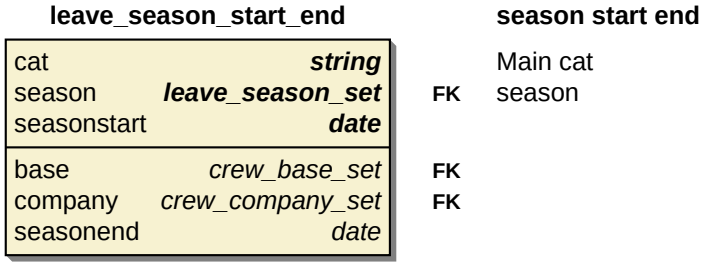
6.30.19 Entity leave_historic_data



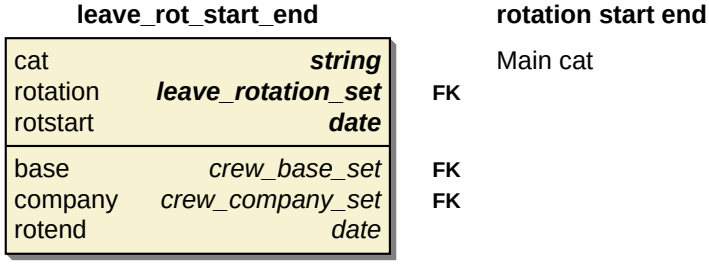
6.30.20 Entity leave_hist_data_type



6.30.21 Entity leave_season_start_end



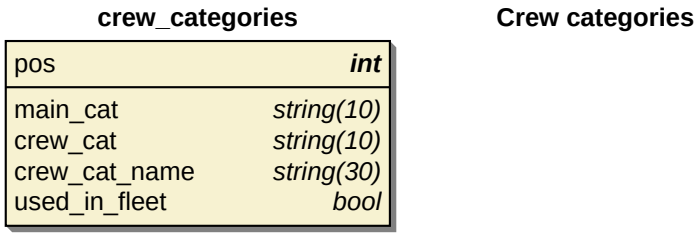
6.30.22 Entity leave_rot_start_end



6.31 Module sas_legality

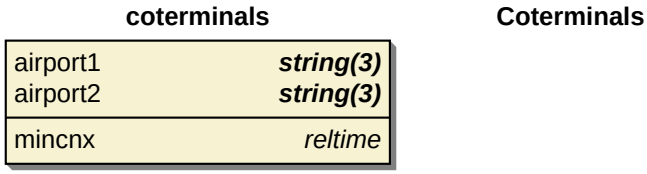
6.31.1 Entity crew_categories

crew$_{c}ategoriesasusedbythestandarduser$

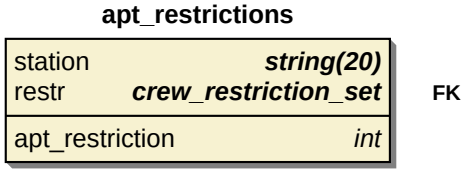


6.31.2 Entity coterminals

Temporary Coterminals



6.31.3 Entity apt_restrictions



6.31.4 Entity apt_requirements

apt_requirements

airport	***string(20)***
area	*string(20)*
fcreq	*bool*
fpreq	*bool*
fcqlnreq	*bool*
simreq	*bool*

6.31.5 Entity lh_apt_exceptions

lh_apt_exceptions

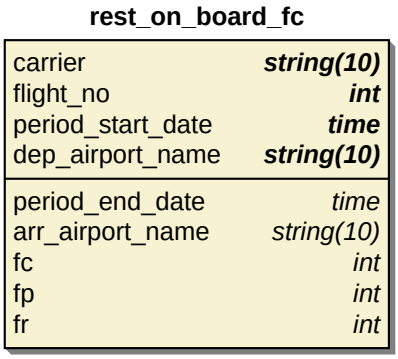
maincat	***string(10)***
region	***string(10)***
station	***string(10)***
validfrom	***date***
validto	*date*

6.31.6 Entity minimum_connection

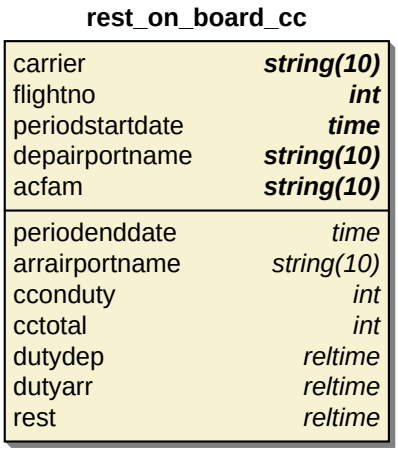
minimum_connection

region	***string(10)***
place	***string(10)***
islonghaul	***string(10)***
area	***string(10)***
nextarea	***string(10)***
cnxtype	***string(10)***
sdate	***time***
edate	*time*
mincnxfc	*reltime*
mincnxcc	*reltime*

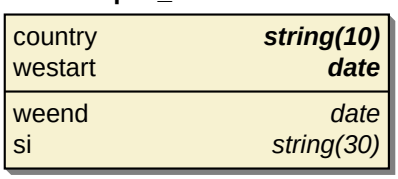
6.31.7 Entity rest_on_board_fc



6.31.8 Entity rest_on_board_cc



6.31.9 Entity spec_weekends



6.32 Module sas_notification

6.32.1 Entity notification_set

This entity lists supported notification types.

notification_set		Notification set
typ	***string(30)***	Notification Type
subtype	***string(30)***	Notification Subtype
descript	*string(20)*	Description
validfrom	*time*	Valid From
validto	*time*	Valid To

6.32.2 Entity notification_systems

This entity lists all supported receiver systems.

notification_systems		Notification receiver systems
system	***string(10)***	Receiver system name
subsystem	***string(10)***	Receiver subsystem
descript	*string(20)*	Description
validfrom	*time*	Valid From
validto	*time*	Valid To

6.32.3 Entity crew_notification

This entity list all available notifications for the crew.

crew_notification			Crew Notification
idnr	***uuid***		Unique Number
crew	*crew*	**FK**	Crew ID
created	*time*		Created Time
created_by	*string(12)*		Created By
system	*notification_systems*	**FK**	Receiver System
typ	*notification_set*	**FK**	Type
deadline	*time*		Deadline
reqconfirmation	*bool*		Requires Confirmation
isdelivered	*bool*		Is Delivered
confirmed	*time*		Confirmation Time
confirmed_by	*string(12)*		Confirmed By
message	*string(300)*		Not. message
si	*string(100)*		Supplementary Info
username	*string(12)*		User Name

6.33 Module sas_passive_bookings

6.33.1 Entity passive_booking

Table with passive bookings for crew. The sent and cancelled flags indicates the current state for a booking.

passive_booking			Passive booking information.
crew	***crew***	**FK**	
flight	***flight_leg***	**FK**	
typ	***char***		P for passive D for deadhead.
book_class	***char***		Booking class for booking (C or M).
cancelled	*bool*		Indicates if reservation is cancelled.
sent	*bool*		Indicates if reservation is sent.

6.34 Module sas_paxlst

6.34.1 Entity paxlst_log

This entity is used for logging every generated Master Crew List or Crew Manifest report. The table is used both for generating sequence numbers and for keeping track of what reports have been generated.

paxlst_log		Log file for General Declaration reports
country	***string(2)***	Report Country
rep_type	***string(10)***	Report Type
gendate	***date***	Report Generation Date
counter	***int***	Report Counter
activity_id	*string(40)*	Activity Id
refnr	*string(20)*	Used Reference Number

6.34.2 Extensions

This module also extends the country (6.6.8.1) entity.

6.35 Module sas_published

6.35.1 Entity crew_publish_info

This entity stores data regarding published roster. Its main purpose is to support the evaluation of rescheduling rules.

crew_publish_info			Published roster data
crewid	***crew***	**FK**	Crew ID.
start_date	***date***		Start date of period with same characteristics
end_date	*date*		End date of period with same characteristics
pcat	*int*		Major category of roster activity in period
checkin	*time*		Check-in performed in period
checkout	*time*		Check-out performed in period
flags	*string(20)*		Significant events/characteristics used for rescheduling rules

6.36 Module sas_special_schedules

6.36.1 Entity special_schedules_set

Special schedule types.

special_schedules_set		Special schedule types.
typ	***string(20)***	Special schedule type identifier.
si	*string(100)*	Special schedule type description.

6.36.2 Entity special_schedules

Special schedules (personal crew information).

special_schedules

crewid	***crew***	
typ	***special_schedules_set***	
valid_from	***date***	
str_val	***string(5)***	
valid_to	*date*	
int_from	*int(1)*	
int_to	*int(1)*	
time_val	*reltime*	

Special schedules.

FK	Crew id.
FK	Special schedule type.
	Valid from date.
	Forbidden activity, destination or Lower.
	Valid to date.
	Valid from integer - nr legs, days or weekday.
	Valid to integer - nr legs, days or weekday.
	Valid time limit.

6.37 Module sas_special_local_transport

6.37.1 Entity spec_local_trans

This entity lists all special local transports. The column leg contains a key string identifying an individual leg. The key is built up as according to the following example: F+SK+4116++24MAY2007+OSL.

Field 1: F (flight leg)
Field 2: carrier code
Field 3: flight number
Field 4: always empty for flight legs
Field 5: departure day
Field 6: departure airport

spec_local_trans

leg	***string(30)***
rest_stat	***string(3)***
crew	***crew***
to_rest	*reltime*
from_rest	*reltime*

Special Local Transport

	Last leg in duty
	Rest station
FK	Crew affected by the SLT
	SLT to hotel
	SLT from hotel

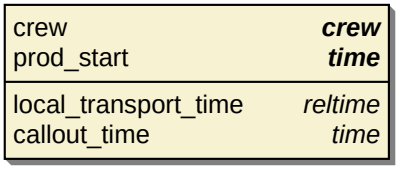
6.38 Module sas_standby

6.38.1 Entity standby_callouts

This entity is used to store data for standbys used by the tracker.

standby_callouts

Standby callout times as well as local transport to from standby

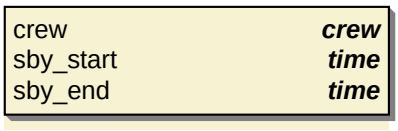


crew	***crew***	**FK**	Crew Identifier
prod_start	***time***		Start time for sby
local_transport_time	*reltime*		Transport time to and from standby
callout_time	*time*		Callout time

6.38.2 Entity published_standbys

published_standbys

Time periods which, at the latest publish, was standbys



crew	***crew***	**FK**	Crew Identifier
sby_start	***time***		Standby Start
sby_end	***time***		Standby End

6.39 Module sas_table_accumulators

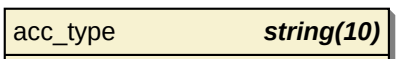
This module contains entities that can be used to accumulate values (e.g. Block Hours) for a certain crew, A/C model and year+month.

6.39.1 Entity crew_log_acc_set

This entity lists supported accumulator types. Example: ’landings’ for crew landings.

crew_log_acc_set

Accumulator type set



acc_type	***string(10)***	Accumulator Type.
is_reltime	*bool*	Indicates that the value field of crew_log_acc and crew_log_acc_mod i
si	*string(40)*	Supplementary Information.

6.39.2 Entity crew_log_acc

This entity contains accumulated values for each crew, type and A/C model partitioned by year and month.

crew_log_acc — **Table for accumulated values.**

crew	***crew***	**FK**	Crew ID.
typ	***crew_log_acc_set***	**FK**	Accumulator type.
acfamily	***string(10)***		A/C family.
year	***int***		Year (four digits).
month	***int***		Month (1-12).
value	*int*		Value accumulated for this year and month.

6.39.3 Entity crew_log_acc_mod

This entity contains manual modifications to the accumulator table, e.g. adjustments to no of Block Hours.

crew_log_acc_mod — **Manual modifications to the accumulator table.**

crew	***crew***	**FK**	Crew ID.
typ	***crew_log_acc_set***	**FK**	Accumulator type.
acfamily	***string(10)***		A/C family.
year	***int***		Year (four digits).
month	***int***		Month (1-12).
value	*int*		Modified value.

6.40 Module sas_training_codes

6.40.1 Entity crew_flight_attr

This entity contains additional attributes on a crew$_{f}light_{d}uty.Someattributespreviouslycoveredbythedutycodec$

crew_flight_attr — **Flight attributes**

crew	***crew***	**FK**	Crew ID
flight	***flight_leg***	**FK**	Flight Id
attr	*string(20)*		Training Code

6.40.2 Entity pgt_need

This entity contains minimum and maximum values for assigned crew on PGT activities.

pgt_need		PGT min and max values
pgt_code	***string(10)***	PGT code compact
valid_from	***date(10)***	Season (period) start
valid_to	***date(10)***	Season (period) end
min_val	*int(10)*	Minimum value
max_val	*int(10)*	Maximum value

6.40.3 Entity crew_training_log

This entity contains a cached view of all crew training activities. It is created when there has been updates to a crew roster at save or when Crew Training Log is started.

crew_training_log			Logged training activities
crew	***crew***	**FK**	Crew ID
typ	***string(20)***		Log type
code	***string(10)***		Activity code
tim	***time***		Start time UTC

6.40.4 Entity crew_training_c_set

List of courses

crew_training_c_set		Valid courses in the crew_training_need table
typ	***string(40)***	Course type
subtype	***string(40)***	Course sub-type
si	*string(100)*	Supplementary information

6.40.5 Entity crew_training_t_set

List of training types

crew_training_t_set		Valid types in the crew_training_need table
id	***string(40)***	Course type
si	*string(100)*	Supplementary information

6.40.6 Entity crew_training_need

This entity contains the required training need for crew. Training legality is triggered by entries in this table.

crew_training_need

crew	***crew***	
ctype	***course_type***	
ix	***int***	
typ	***crew_training_t_set***	
trngstart	***date***	
trngend	*date*	
nrflts	*int*	
maxdays	*int*	
ac_qual	*string(10)*	
si	*string(40)*	

Planned flight training activities

FK	Crew ID
FK	Course type
	Order in course block
FK	Training type
	Training period start
	Training period end
	Number of needed flights
	Training must be completed within this number of days
	Optional ACQUAL for required sectors

6.40.7 Entity training_tasks

Specification of cabin crew courses

training_tasks

task_code	***string(10)***
ac_qual	***string(10)***
base	***string(10)***
typ	***string(10)***
valid_from	***date***
valid_to	*date*

Specification of cabin crew courses

Task code
Valid AC quals
Base
Training type group
Valid from
Valid to

6.40.8 Entity crew_recurrent_set

Lists recurrent training categories

crew_recurrent_set		Lists recurrent training categories
typ	***string(10)***	Type
valid_from	***date***	Valid from
valid_to	*date*	Valid to
main_cat	*string(10)*	Main category
ac_qual	*string(10)*	AC qualification
aoc_sk	*bool*	Valid for SK AOC
aoc_bu	*bool*	Valid for BU AOC
validity	*int*	Valid from
season1_start	*int*	Season1 start
season2_start	*int*	Season2 start
assignment_ival	*int*	Assignment value
si	*string*	

6.40.9 Entity crew_rehearsal_rec

Crew valid for recurrent rehearsal

crew_rehearsal_rec		Crew valid for recurrent rehearsal
crew	***string(10)***	Crew

7 Appendix - Enumeration of types

7.1 Crew Documents

This section lists a suggested set of standard documents.

*** UNFINISHED:

- Language?
- Are type ratings tied to CP/FO?
- Recurrent training (emergency, line, route, airport)?

7.1.1 Document Number and Issuer

Type	Docno	Maindocno	Issuer
passport	passport number	-	IATA city code where issued
visa	visa number	passport number	IATA city code where issued
vaccination	-	-	-
licence	licence number	-	-
rating	-	licence number	-
medical	-	licence number	-

7.1.2 General Documents

Type	Subtype	Comment
passport	DK	Passport
passport	NO	Passport
passport	SE	Passport
visa	US,B1	Visa
visa	US,B1/B2	Visa
visa	US,C1	Visa
visa	US,C1/D	Visa
visa	US,D	Visa
visa	CN,crew	Visa
vaccination	diphtheria	Vaccination
vaccination	havrix	Vaccination
vaccination	tetanus	Vaccination
licence	SE,B	BCL B Certificate
licence	SE,D	BCL D Certificate

7.1.3 JAA Documents

Type	Subtype	Comment
medical	JAA,class1	JAA Medical
medical	JAA,class2	JAA Medical
licence	JAA,ATPL	JAA ATPL(A) Licence
licence	JAA,CPL	JAA CPL(A) Licence
rating	JAA,FI	JAA Flight Instructor Rating
rating	JAA,IRI	JAA Instrument Rating Instructor Rating
rating	JAA,TRI	JAA Type Rating Instructor Rating
rating	JAA,CRI	JAA Class Rating Instructor Rating
rating	JAA,IR	JAA Instrument Rating
rating	JAA,ME	JAA Multiengine Rating
rating	JAA,A300	JAA Type Rating
rating	JAA,A300FFCC	JAA Type Rating
rating	JAA,A310 300-600	JAA Type Rating
rating	JAA,A320	JAA Type Rating
rating	JAA,A330	JAA Type Rating
rating	JAA,A340	JAA Type Rating
rating	JAA,B737 100-200	JAA Type Rating
rating	JAA,B737 300-500	JAA Type Rating
rating	JAA,B737 600-900	JAA Type Rating
rating	JAA,B747 100-300	JAA Type Rating
rating	JAA,B747 400	JAA Type Rating
rating	JAA,B757/767	JAA Type Rating
rating	JAA,B777	JAA Type Rating
rating	JAA,DC9 80/MD88/MD90	JAA Type Rating *Is this really one rating?*
rating	JAA,DC10	JAA Type Rating
rating	JAA,MD11	JAA Type Rating

7.1.4 FAA Documents

Type	Subtype	Comment
medical	FAA,class1	FAA Medical
medical	FAA,class2	FAA Medical
medical	FAA,class3	FAA Medical
licence	FAA,E	FAA Flight Engineer Certificate
licence	FAA,F	FAA Flight Instructor Certificate
licence	FAA,G	FAA Ground Instructor Certificate
licence	FAA,P,A	FAA Airline Transport Pilot Certificate
licence	FAA,P,C	FAA Commercial Pilot Certificate
licence	FAA,Z	FAA Flight Attendant Certificate
rating	FAA,A/AMEL	FAA Multiengine Rating, A Level
rating	FAA,C/AMEL	FAA Multiengine Rating, C Level
rating	FAA,C/INST	FAA Instrument Rating, C Level
rating	FAA,E/JET	FAA Turbojet Rating
rating	FAA,E/TPROP	FAA Turbopropeller Rating
rating	FAA,F/AME	FAA Multiengine Instructor Rating
rating	FAA,F/INSTA	FAA Instrument Instructor Rating
rating	FAA,G/FAR	FAA FAR Instructor Rating
rating	FAA,Z/GROUPI	FAA Cabin Attendant Rating
rating	FAA,Z/GROUPII	FAA Cabin Attendant Rating
rating	FAA,A-300	FAA Type Rating
rating	FAA,A-310	FAA Type Rating
rating	FAA,A-320	FAA Type Rating
rating	FAA,A-340	FAA Type Rating
rating	FAA,B-737	FAA Type Rating
rating	FAA,B-747	FAA Type Rating
rating	FAA,B-747-4	FAA Type Rating
rating	FAA,B-757	FAA Type Rating
rating	FAA,B-767	FAA Type Rating
rating	FAA,B-777	FAA Type Rating
rating	FAA,DC-9	FAA Type Rating
rating	FAA,MD-11	FAA Type Rating