eurodeer_db_report

activity_data_raw schema

This schema stores the raw activity data that have still to be (partially) processed to be analyzed and merged together.

analysis schema

The schema "analysis" stores the results of analysis (home range, trajectories, and statistics).

test_home_ranges:	Table that stores the home range polygons derived from a set of possible methods. A set of fieldsis used as metadata to identify the source data: the animal (which is considered the basic "unit" for the home range computation), the time range that is used to select locations considered in the analysis,, and the total number of locations that generated the home range (this can be modified to accept a more general SQL select statements where any criteria can be used, not just starting time and ending time; this option must be discussed to identify the best implementation to satisfy NINA's requirements). The computation method is stored in a specific (coded in a look up table) field connected to other fields where are stored the parameters (the table has generic fields "parametr_1, parameter_2, parameter_3, parameter_4, the meaning of these parameters depend on the method and are explained in the look up tables of the method code). New parameters (e.g. parameter_5) can be added at any time. The area (in hectars) is computed and stored in the filed "area". These other fields are used to carachterize the analysis: user who performed the analysis, timestamp when the analysis was performed, and a general description. For home ranges that derive from a probability surface (e.g. kernel home range), the reference probability surface (raster) is stored. Note that the key is a serial number, therefore the same analysis can be performed twice and the results will be both stored in the database. The field obsolete_code will be supported by a function to detect obsolete analysis (checking the number of locations that originated the home range, and verifying that none of the original data is newer that the date of the analysis).

Attribute	Type	Description
home_ranges_id	integer	Code that identifies uniquely each home_ranges record. This information is a serial number generated by postgresql. This field is the primary key of the table.
animals_id	integer	Animal to which is related the information.
start_time	timestamp without time zone	Starting timestamp used as a selection criteria applied to the data source.
end_time	timestamp without time zone	Ending timestamp used as a selection criteria applied to the data source.

hr_method_code	integer	Meethod used to compute the home range. Many methods are available, and this filed records a code that reference the table lu_tables.lu_hr_method where each code is explained. The fields "parameter_1", "parameter_2", "parameter_3", "parameter_4" (described lu_tables.lu_hr_method) specify the values of the parameters (if any) used in the home range method. If a method with futher parameters is used, new "parametr_x" fileds can be added.
description	character varying	Text field where users can comment and describe the analysis. This field can be used to "tag" the analysis in order to retrieve theme easier.
prob_surfaces_id	integer	If the home range derive from a probability surface (e.g. kernel methods), here the raster is referenced. Note that many home ranges can be related to the same probability surface.
ref_user	character varying	Name of the user who performed the analysis.
num_locations	integer	Number of locations used to do the analysis (as a product of selection criteria applied to the data source.
area	numeric	Area of the home range computed in km^2 (with 5 decimals).
geom	USER-DEFINED	Geometry field (st_multipolygon, 2 dimension, SRID 4326 - geographic coordinates datum WGS84).
obsolete_code	integer	A function can compare the number of locations used for the computation with the number of locations available at any moment using the same selection parameter. At the same time, it is possible to see if any new location is available or has been updated since the analysis was performed. If one of these conditions is met, it means that the result of the analysis is no more updated and should be run again.
insert_timestamp	timestamp without time zone	Timestamp (in UTC zone) when the record was inserted in the table.
original_data_set	character varying	Source of data used for the selection (by default: main.view_locations_set). If a more complex source is used, e.g. a "select" statement, the whole select statement is recorded.
parameter_1	character varying	This field specify the values of the second parameter (if used) of the home range function.
parameter_4	character varying	None
parameter_3	character varying	None
parameter_2	character varying	None

view_animals_days: Number of locations per animal per day

Attribute	Type	Description
animals_id	integer	None
date	date	None
total	bigint	None

valid	bigint	None
nodata	bigint	None
invalid	bigint	None

view_convexhull: View with the convex hull of all valid locations per all the animals of Eurodeer dataset

Attribute	Type	Description
animals_id	integer	None
name	character varying	None
studies_id	integer	None
area	double precision	None
geom	USER-DEFINED	None

view_convexhull_reddeer: View with the convex hull of all valid locations per all the RED DEER of Eurodeer dataset

Attribute	Type	Description
animals_id	integer	None
name	character varying	None
studies_id	integer	None
area	double precision	None
geom	USER-DEFINED	None

view_convexhull_vhf: View with the convex hull of all valid locations per all the animals of red deer vhf dataset

Attribute	Type	Description
animals_id	integer	None
name	character varying	None
studies_id	integer	None
area	double precision	None
geom	USER-DEFINED	None

view_locations_12h: None

Attribute	Type	Description
study_areas_id	integer	None
animals_original_name	character varying	None
sex	"char"	None
age_class_code_capture	integer	None
gps_data_animals_id	integer	None
animals_id	integer	None
gps_sensors_id	integer	None
acquisition_time	timestamp with time zone	None
x_original_reference	double precision	None
y_original_reference	double precision	None
srid_original_reference	integer	None
latitude	double precision	None
longitude	double precision	None
altitude_gps	integer	None
dop	double precision	None
sats	integer	None
temperature_sensor	double precision	None
geom	USER-DEFINED	None
gps_validity_code	smallint	None
corine_land_cover_2006_code	integer	None
ndvi_modis	real	None
ndvi_vegetation	real	None
snow_modis	integer	None
sun_angle	double precision	None
utm_srid	integer	None
utm_x	integer	None
utm_y	integer	None
altitude_srtm	integer	None
slope_srtm	double precision	None
aspect_srtm	integer	None
insert_timestamp	timestamp with time zone	None
update_timestamp	timestamp with time zone	None
altitude_aster	integer	None
slope_aster	double precision	None
aspect_aster	integer	None

view_locations_12h:	View with a fix sequence at intervals of no less than 12 hrs, i.e. all locations at higher frequency are excluded (it uses the view view_locations_12h_calculation)

Attribute	Type	Description
study_areas_id	integer	None
animals_original_name	character varying	None
sex	"char"	None
age_class_code_capture	integer	None
gps_data_animals_id	integer	None
animals_id	integer	None
gps_sensors_id	integer	None
acquisition_time	timestamp with time zone	None
x_original_reference	double precision	None
y_original_reference	double precision	None
srid_original_reference	integer	None
latitude	double precision	None
longitude	double precision	None
altitude_gps	integer	None
dop	double precision	None
sats	integer	None
temperature_sensor	double precision	None
geom	USER-DEFINED	None
gps_validity_code	smallint	None
corine_land_cover_2006_code	integer	None
ndvi_modis	real	None
ndvi_vegetation	real	None
snow_modis	integer	None
sun_angle	double precision	None
utm_srid	integer	None
utm_x	integer	None
utm_y	integer	None
altitude_srtm	integer	None
slope_srtm	double precision	None
aspect_srtm	integer	None
insert_timestamp	timestamp with time zone	None
update_timestamp	timestamp with time zone	None
altitude_aster	integer	None
slope_aster	double precision	None

aspect_aster	integer	None

view_locations_12h_calculation:

Regularized selection of valid locations (1 every -at least- 12 hours)

Attribute	Type	Description
animals_id	integer	None
interval_id	integer	None
min_acquisition_time	timestamp with time zone	None

view_locations_1h:

View with a fix sequence at intervals of no less than 1 hrs, i.e. all locations at higher frequency are excluded (it uses the view view_locations_1h_calculation)

Attribute	Type	Description
study_areas_id	integer	None
animals_original_name	character varying	None
sex	"char"	None
age_class_code_capture	integer	None
gps_data_animals_id	integer	None
animals_id	integer	None
gps_sensors_id	integer	None
acquisition_time	timestamp with time zone	None
x_original_reference	double precision	None
y_original_reference	double precision	None
srid_original_reference	integer	None
latitude	double precision	None
longitude	double precision	None
altitude_gps	integer	None
dop	double precision	None
sats	integer	None
temperature_sensor	double precision	None
geom	USER-DEFINED	None
gps_validity_code	smallint	None
corine_land_cover_2006_code	integer	None
ndvi_modis	real	None
ndvi_vegetation	real	None
snow_modis	integer	None

sun_angle	double precision	None
utm_srid	integer	None
utm_x	integer	None
utm_y	integer	None
altitude_srtm	integer	None
slope_srtm	double precision	None
aspect_srtm	integer	None
insert_timestamp	timestamp with time zone	None
update_timestamp	timestamp with time zone	None
altitude_aster	integer	None
slope_aster	double precision	None
aspect_aster	integer	None

view_locations_1h_calculation:

Regularized selection of valid locations (1 every -at least- 1 hour)

Attribute	Type	Description
animals_id	integer	None
interval_id	integer	None
min_acquisition_time	timestamp with time zone	None

view_locations_24h:

View with a fix sequence at intervals of no less than 24 hrs, i.e. all locations at higher frequency are excluded (it uses the view view_locations_24h_calculation)

Attribute	Type	Description
study_areas_id	integer	None
animals_original_name	character varying	None
sex	"char"	None
age_class_code_capture	integer	None
gps_data_animals_id	integer	None
animals_id	integer	None
gps_sensors_id	integer	None
acquisition_time	timestamp with time zone	None
x_original_reference	double precision	None
y_original_reference	double precision	None
srid_original_reference	integer	None
latitude	double precision	None

longitude	double precision	None
altitude_gps	integer	None
dop	double precision	None
sats	integer	None
temperature_sensor	double precision	None
geom	USER-DEFINED	None
gps_validity_code	smallint	None
corine_land_cover_2006_code	integer	None
ndvi_modis	real	None
ndvi_vegetation	real	None
snow_modis	integer	None
sun_angle	double precision	None
utm_srid	integer	None
utm_x	integer	None
utm_y	integer	None
altitude_srtm	integer	None
slope_srtm	double precision	None
aspect_srtm	integer	None
insert_timestamp	timestamp with time zone	None
update_timestamp	timestamp with time zone	None
altitude_aster	integer	None
slope_aster	double precision	None
aspect_aster	integer	None

view_locations_24h_calculation:

Regularized selection of valid locations (1 every -at least- 24 hours)

Attribute	Type	Description
animals_id	integer	None
interval_id	integer	None
min_acquisition_time	timestamp with time zone	None

view_locations_4h:

View with a fix sequence at intervals of no less than 4 hrs, i.e. all locations at higher frequency are excluded (it uses the view view_locations_4h_calculation)

Attribute	Type	Description
study_areas_id	integer	None

animals_original_name	character varying	None
sex	"char"	None
age_class_code_capture	integer	None
gps_data_animals_id	integer	None
animals_id	integer	None
gps_sensors_id	integer	None
acquisition_time	timestamp with time zone	None
x_original_reference	double precision	None
y_original_reference	double precision	None
srid_original_reference	integer	None
latitude	double precision	None
longitude	double precision	None
altitude_gps	integer	None
dop	double precision	None
sats	integer	None
temperature_sensor	double precision	None
geom	USER-DEFINED	None
gps_validity_code	smallint	None
corine_land_cover_2006_code	integer	None
ndvi_modis	real	None
ndvi_vegetation	real	None
snow_modis	integer	None
sun_angle	double precision	None
utm_srid	integer	None
utm_x	integer	None
utm_y	integer	None
altitude_srtm	integer	None
slope_srtm	double precision	None
aspect_srtm	integer	None
insert_timestamp	timestamp with time zone	None
update_timestamp	timestamp with time zone	None
altitude_aster	integer	None
slope_aster	double precision	None
aspect_aster	integer	None

view_locations_4h_calculation:	Regularized selection of valid locations (1 every -at least- 4 hours)

Attribute	Type	Description
animals_id	integer	None
interval_id	integer	None
min_acquisition_time	timestamp with time zone	None

view_ltraj_class:	This view extracts a subset of fields from main.view_locations_set in order to have an object of class "ltraj" in the adehabitat package in R (animals_id integer, acquisition_time as second from 1am 1970 1st january, x and y coordinates in the proper UTM zone). This view can be used in R to create a ltraj class with no bursts. here an example: get the data data_traj_raw <- sqlQuery(channel, "SELECT * FROM analysis.view_ltraj_class;"); create a ltraj object create an ltraj object data_traj<-as.ltraj(xy=data_traj_raw[,c("x","y")], date=as.POSIXct(data_traj_raw [,"acquisition_epoch"], origin="1970-01-01 01:00:00"), id =data_traj_raw [,"animals_id"]) Any "where" clause can be added to "SELECT * FROM analysis.view_ltraj_class" to limits animals (e.g. "where animals_id in (1,2,3,6,7)") or starting and ending time. To create ltraj objces with bursts identificator, you have to use the tools.sam_traj_bursts function.

Attribute	Type	Description
animals_id	integer	None
acquisition_epoch	bigint	None
x	integer	None
y	integer	None

view_statistics_animals:	Statistics with summarized information on each animal

Attribute	Type	Description
studies_id	integer	None
animals_id	integer	None
sex	text	None
name	text	None
start	date	None
end	date	None
days_monitored	bigint	None
fix_valid	numeric	None
fix_null	numeric	None
fix_invalid	numeric	None
average_fix_per_day	numeric	None
days_1_fix	bigint	None
days_2_3_fix	bigint	None

days_4_5_fix	bigint	None
days_more6_fix	bigint	None
area_convex_hull_km2	numeric	None

view_statistics_studies: Statistics with summarized information on each study area

Attribute	Type	Description
studies_id	integer	None
study_name	text	None
num_animals	bigint	None
days_monitored	numeric	None
fix_valid	numeric	None
fix_null	numeric	None
fix_invalid	numeric	None
average_fix_per_day	numeric	None
perc_days_1_fix	numeric	None
perc_days_2_3_fix	numeric	None
perc_days_4_5_fix	numeric	None
perc_days_more6_fix	numeric	None

view_test_homeranges_points: This view taked the "probability" grid in "view_test_probability_grid_points" and extract the subset of cells that cover a defined percentage (in this test case, 80%) of the total hours. It selects the cell with most hours, then the second, and so on until the cumuated sum of hours (compared to the totl number of hours) is equal to the desired threshold.

Attribute	Type	Description
id	integer	None
animals_id	integer	None
geom	USER-DEFINED	None
cell_id	integer	None
hours_spent	integer	None
perc	numeric	None

view_test_probability_grid_points:

This view presents the sql code to calculate the time spent by an animal on every cell of a defined resolution, which correspond to a probability surface. In this case, points are considered. Each point represents half of the time between the same point and the next point and the same point and th previous point. The view "view_test_homerange" select just the cells that represent a defined, cumulare value (like the home range tool in adehabitat). This view calls the function "tools.reate_grid". At the montent, it is a view with pure SQL, but this tool can be coded into a function that using temporary tables ad some other optimized approach, can speed up the processing time. (in this example, animals 1 and 2 are considered).

Attribute	Type	Description
id	integer	None
animals_id	integer	None
cell_id	integer	None
hours_spent	integer	None
geom	USER-DEFINED	None

view_test_probability_grid_traj:

This view presents the sql code to calculate the time spent by an animal on every cell of a defined resolution, which correspond to a probability surface. In this case, trajectory (segments between locations) is considered. Each segment represents the time spent between the two locations. This view calls the function "tools.reate_grid". At the montent, it is a view with pure SQL, but htis tool can be coded into a function that using temporary tables ad some other optimized approach, can speed up the processing time. In this case, animals 1 and 2 are cosidered.

Attribute	Type	Description
id	integer	None
animals_id	integer	None
cell_id	integer	None
geom	USER-DEFINED	None
hours_spent	integer	None

view_trajectories:

Complete trajectories as linear spatial features per each of the animals of Eurodeer dataset

Attribute	Type	Description
animals_id	integer	None
name	character varying	None

studies_id	integer	None
geom	USER-DEFINED	None

view_trajectories_reddeer:	Red deer trajectories

Attribute	Type	Description
animals_id	integer	None
name	character varying	None
studies_id	integer	None
geom	USER-DEFINED	None

env_data schema

The schema "env_data" hosts all the (static) environmental and socio economic layers. Raster time series are stored in separated schemas (env_data_ts).

administrative_units:	Boundaries of administrative units (3rd, 4th, or 5th level according to the country) for the countries of Eurodeer study areas (source: www.gadm.org).

Attribute	Type	Description
administrative_units_id	integer	None
id_0	integer	None
iso	character varying	None
name_0	character varying	None
id_1	integer	None
name_1	character varying	None
type_1	character varying	None
id_2	integer	None
name_2	character varying	None
type_2	character varying	None
id_3	integer	None
name_3	character varying	None
type_3	character varying	None
id_4	integer	None
name_4	character varying	None
type_4	character varying	None

id_5	integer	None
name_5	character varying	None
type_5	character varying	None
geom	USER-DEFINED	None

aspect_aster:

Degrees calculated counterclockwise from east (source: ASTER project)

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

aspect_srtm:

Degrees calculated counterclockwise from east (source: SRTM project)

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

aster_index:

Schema of available Aster tiles.

Attribute	Type	Description
aster_index_id	integer	None
Image	character varying	None
XYSize	character varying	None
BandTypes	character varying	None
ResSpatial	character varying	None
in_db	integer	None
geom	USER-DEFINED	None

corine_land_cover_1990:

None

Attribute	Type	Description
rid	integer	None

rast	USER-DEFINED	None

corine_land_cover_2000:

None

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None

corine_land_cover_2006:

None

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None

corine_land_cover_legend:

Legend of Corine land cover. Corine is a EU project that produced a continental land cover map in 1990, 2000, and 2006. These three layers are stored in Eurodeer as raster at 100 meters resolution in etrs_1989_laea projection. The legend can be used to extract the meaning of each land cover class at 3 different semantic levels.

Attribute	Type	Description
grid_code	integer	Code stored in the raster layers.
clc_l3_code	character varying	Official Corine code (level 3).
label1	character varying	Description of the corine class at level 1.
label2	character varying	Description of the corine class at level 2.
label3	character varying	Description of the corine class at level 3.

dem_aster:

None

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

dem_srtm:

None

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

ndvi_constancy:	C (Constancy): inter-annual variability or year-to-year stochastic variation (calculates using 10-daily smoothed MODIS NDI). The higher C, the lower is the inter-annual variability in NDVI. Complete constancy (1) would indicate that NDVI is the same for all seasons in all years.

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

ndvi_contingency:	M (Contingency): seasonality or within-year variability (calculates using 10-daily smoothed MODIS NDI). Complete contingency (1) would indicate that NDVI is different for each season, but the pattern is the same for all years (seasons are highly predictable every year).

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

ndvi_predictability:	P (predictability): measures variation among successive periods in the pattern of a periodic phenomenon (calculates using 10-daily smoothed MODIS NDI). Maximum predictability can either be attained through complete M or complete C.

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

slope_aster:	None

Attribute	Type	Description

rid	integer	None
rast	USER-DEFINED	None
filename	text	None

slope_srtm: None

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

srtm_index: Schema of available SRTM tiles.

Attribute	Type	Description
srtm_index_id	integer	None
Image	character varying	None
XYSize	character varying	None
BandTypes	character varying	None
ResSpatial	character varying	None
in_db	integer	None
geom	USER-DEFINED	None

study_areas_ref: None

Attribute	Type	Description
namex	character varying	None
geom	USER-DEFINED	None

world_countries: Boundaries of all the countries of the world.

Attribute	Type	Description
world_countries_id	integer	None
name	character varying	None
iso3	character varying	None

iso2	character varying	None
continent	character varying	None
geom	USER-DEFINED	None

world_countries_simplified:	Boundaries of all the countries of the world, simplified (using ST_SimplifyPreserveTopology, 0.01 as parameter).

Attribute	Type	Description
world_countries_simplified_id	integer	None
name	character varying	None
iso3	character varying	None
iso2	character varying	None
continent	character varying	None
geom	USER-DEFINED	None

env_data_ts schema

the schema "env_data_ts" stores environmental layers in form of raster time series.

fapar_vegetation:	Table that stores the (smoothed) ten daily values of fapar (time series) from Spot Vegetation. Note that in some cases, the value 0 is associated to pixel covered by snow, so this value should be consideredas NULL. Moreover, if locations fall close to the sea, due to the coarse resolution of VEGTATION, the related pixel can be classified as water and thus no value is associated with it.

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None
acquisition_date	date	None

ndvi_modis:	None

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None

acquisition_date	date	None

ndvi_modis_boku:

None

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None
acquisition_date	date	None
filepath	character varying	None

ndvi_modis_boku_grid:

None

Attribute	Type	Description
tiles_grid_id	integer	None
geom	USER-DEFINED	None
big_tiles	character varying	None
small_tile	character varying	None
tile	character varying	None
acquired	integer	None
imported	integer	None

ndvi_modis_smoothed:

None

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None
acquisition_date	date	None

ndvi_vegetation:

Table that stores the (smoothed) ten daily values of NDVI (time series) from Spot Vegetation. Note that in some cases, the value 0 is associated to pixel covered by snow, so this value should be consideredas NULL. Moreover, if locations fall close to the sea, due to the coarse resolution of VEGTATION, the related pixel can be classified as water and thus no value is associated with it.

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None
acquisition_date	date	None

snow_modis: Table that stores information on snow from Modis.

Attribute	Type	Description
rid	integer	None
rast	USER-DEFINED	None
filename	text	None
acquisition_date	date	None

snow_modis_legend: Codes of the Modis Snow time series.

Attribute	Type	Description
modis_snow_code	integer	Code stored in the raster TS (as produced by NASA).
modis_snow_description	character varying	description of the Modis Snow code.

lu_tables schema

The schema "lu_tables" is where the look up tables (lu_tables) are stored. These tables store the list and the description of codes referenced by other tables in the database and are a kind of valid domain for specific fields.

lu_action: Look up table for action_code field (table tools.log_dbchanges): it specifies the meaning of the code used to identify the action (change) done to the daatbase).

Attribute	Type	Description
action_code	integer	Code for the action (change done to the db).
action_description	character varying	Desciption of action (change done to the db).

lu_activity_sensor_mode: None

Attribute	Type	Description
activity_sensor_mode_code	integer	None
activity_sensor_mode_description	character varying	None

lu_activity_validity: Look up table for activity data validity.

Attribute	Type	Description
activity_validity_code	integer	Code of the activity data validity
activity_validity_description	character varying	Description of the activity data validity

lu_age_class: Look up table for age_class_code field (table main.animals): it specifies the meaning of the code used to identify the age class of the roe deer).

Attribute	Type	Description
age_class_code	integer	Code for the age class.
age_class_description	character varying	Desciption of the age class.
age_class_comment	character varying	Description of the meaning of the age class.

lu_age_class_reddeer: Look up table for age_class_code field (table main_reddeer.animals): it specifies the meaning of the code used to identify the age class of the red deer).

Attribute	Type	Description
age_class_code	integer	Code for the age class.
age_class_description	character varying	Desciption of the age class.
age_class_comment	character varying	Description of the meaning of the age class.

lu_behavior_handling: Look up table for behavior_handling types.

Attribute	Type	Description
behavior_handling_code	integer	Code of the behavior during handling (capture event) type
behavior_handling_description	character varying	Description of the behavior handling (capture event) type

lu_behavior_release: Look up table for behavior_release types.

Attribute	Type	Description
behavior_release_code	integer	Code of the behavior at release (during capture event) type
behavior_release_description	character varying	Description of the behavior at release (during capture event) type

lu_capture_methods: Look up table for capture_methods types.

Attribute	Type	Description
capture_methods_code	integer	Code of the capture method
capture_methods_description	character varying	Description of the capture method
capture_methods_note	character varying	Additional notes on the capture method

lu_capture_result: Look up table for capture_result types (animals_captures table). Only animals that are monitored or that were captured to be monitored are included in the data base. This LU specifies the possible results of the capture.

Attribute	Type	Description
capture_result_code	integer	Code of the capture result
capture_result_description	character varying	Description of the capture result
capture_result_note	character varying	Additional notes on the capture result

lu_competitor_density: Look up table for predators density types.

Attribute	Type	Description
competitor_density_code	integer	Code of the density type
competitor_density_description	character varying	Description of the density type
competitor_density_note	character varying	Additional notes of the density type

lu_data_curators: Look up table for data_curators_code field (table tools.log_dbchanges): it specifies the meaning of the code used to identify the data curators.

Attribute	Type	Description
data_curators_code	integer	Code for the data_curators.

data_curators_description	character varying	Desciption of data_curators.

lu_end_monitoring:

Look up table for end_monitoring_code field: it specifies the meaning of the code used to identify the reasons of the end of deployment.

Attribute	Type	Description
end_monitoring_code	integer	Code for the reason of the end of monitoring.
end_monitoring_description	character varying	Desciption of the reason of the end of monitoring.

lu_farming:

Look up table for farming types.

Attribute	Type	Description
farming_code	integer	Code of the farming type
farming_description	character varying	Description of the farming type

lu_forestry_work:

Look up table for forestry work types.

Attribute	Type	Description
forestry_work_code	integer	Code of the forestry work type
forestry_work_description	character varying	Description of the forestry work type

lu_gps_validity:

Look up table for GPS locations validity.

Attribute	Type	Description
gps_validity_code	integer	Code of the GPS locations validity.
gps_validity_description	character varying	Description of the GPS locations validity code.

lu_hunting_method:

Look up table for hunting method types.

Attribute	Type	Description
hunting_method_code	integer	Code of the hunting method type
hunting_method_description	character varying	Description of the hunting method type

lu_mortality:

Look up table for mortality_code field (table main.animals_contacts): it specifies the meaning of the code used to identify the reasons of the death of the animal.

Attribute	Type	Description
mortality_code	integer	Code for the reason of the death of the animal.
mortality_description	character varying	Desciption of the reason of the death of the animal.

lu_predators_density:

Look up table for predators density types.

Attribute	Type	Description
predators_density_code	integer	Code of the predators density type
predators_density_description	character varying	Description of the predators density type
predators_density_note	character varying	Additional notes of the predators density type

lu_sampling_methods:

Look up table for sampling methods types.

Attribute	Type	Description
sampling_methods_code	integer	Code of the sampling methods type
sampling_methods_description	character varying	Description of the sampling methods type

lu_seasons:

Seasons of the year

Attribute	Type	Description
seasons_code	integer	Code of the seasons type
seasons_description	character varying	Description of the seasons type

lu_seasons:

Look up table for seasons types.

Attribute	Type	Description
seasons_code	integer	Code of the seasons type
seasons_description	character varying	Description of the seasons type

lu_vhf_validity:

Look up table for vhf locations source and validity.

Attribute	Type	Description
vhf_validity_code	integer	Code of the vhf locations validity
vhf_validity_description	character varying	Description of the vhf locations validity and source code

main schema

The schema "main" is the place where all the core information of the main objects are stored: data from sensors (at the moment, GPS, VHF, activity), sensors, animals, studies, research groups.

activity_data_animals_code01:	Table with activity data associated to animals with sensor mode code = 1 (see table lu_tables.lu_activity_sensor_mode). Data come from different type of sensors, thus the information must be properly precessed to be correctly analyzed. There is 1 table per sensor mode code to reduce the overall size and to keep different things separated.

Attribute	Type	Description
activity_data_animals_code01_id	integer	Eurodeer identifier for the activity record
animals_id	integer	Eurodeer identifier for the animal
activity_sensors_id	integer	Eurodeer identifier for the activity sensor
act_1	double precision	The meaning of this field depends on the sensor_mode. Usually it is the X axis accellerometer.
act_2	double precision	The meaning of this field depends on the sensor_mode. Usually it is the y axis accellerometer.
act_3	double precision	The meaning of this field depends on the sensor_mode. Usually it is the z axis accellerometer.
acquisition_time	timestamp without time zone	Date and time of acquisition of the activity data (with time zone)
temperature_activity	double precision	Temperature measured by the sensor associated to the accelerometer
gps_positions_animals_id	bigint	This filed links the activity record to the (closer in time) valid GPS position
activity_sensor_mode_code	integer	This field explains the meaning of the act1,2,3. There 3 values can have a different meaning according to the sensor and in case also to the operational mode the sensor is used. See table activity_sensor_mode.
activity_validity_code	integer	This field marks each record as valid or invalid for different reasons: record registered when the sensor was not deployed on the animal, record with impossible values, etc.

activity_data_animals_code02:	Table with activity data associated to animals with sensor mode code = 2 (see table lu_tables.lu_activity_sensor_mode). Data come from different type of sensors, thus the information must be properly precessed to be correctly analysized. There is 1 table per sensor mode code to reduce the overall size and to keep different things separated.

Attribute	Type	Description
activity_data_animals_code02_id	integer	Eurodeer identifier for the activity record
animals_id	integer	Eurodeer identifier for the animal
activity_sensors_id	integer	Eurodeer identifier for the activity sensor
act_1	double precision	The meaning of this field depends on the sensor_mode. Usually it is the X axis accellerometer.
act_2	double precision	The meaning of this field depends on the sensor_mode. Usually it is the y axis accellerometer.
act_3	double precision	The meaning of this field depends on the sensor_mode. Usually it is the z axis accellerometer.
acquisition_time	timestamp without time zone	Date and time of acquisition of the activity data (with time zone)
temperature_activity	double precision	Temperature measured by the sensor associated to the accelerometer
gps_positions_animals_id	bigint	This filed links the activity record to the (closer in time) valid GPS position
activity_sensor_mode_code	integer	This field explains the meaning of the act1,2,3. There 3 values can have a different meaning according to the sensor and in case also to the operational mode the sensor is used. See table activity_sensor_mode.
activity_validity_code	integer	This field marks each record as valid or invalid for different reasons: record registered when the sensor was not deployed on the animal, record with impossible values, etc.

activity_data_animals_code03:	Table with activity data associated to animals with sensor mode code = 3 (see table lu_tables.lu_activity_sensor_mode). Data come from different type of sensors, thus the information must be properly precessed to be correctly analysized. There is 1 table per sensor mode code to reduce the overall size and to keep different things separated.

Attribute	Type	Description
activity_data_animals_code03_id	integer	Eurodeer identifier for the activity record
animals_id	integer	Eurodeer identifier for the animal
activity_sensors_id	integer	Eurodeer identifier for the activity sensor
act_1	double precision	The meaning of this field depends on the sensor_mode. Usually it is the X axis accellerometer.

act_2	double precision	The meaning of this field depends on the sensor_mode. Usually it is the y axis accellerometer.
act_3	double precision	The meaning of this field depends on the sensor_mode. Usually it is the z axis accellerometer.
acquisition_time	timestamp without time zone	Date and time of acquisition of the activity data (with time zone)
temperature_activity	double precision	Temperature measured by the sensor associated to the accelerometer
gps_positions_animals_id	bigint	This filed links the activity record to the (closer in time) valid GPS position
activity_sensor_mode_code	integer	This field explains the meaning of the act1,2,3. There 3 values can have a different meaning according to the sensor and in case also to the operational mode the sensor is used. See table activity_sensor_mode.
activity_validity_code	integer	This field marks each record as valid or invalid for different reasons: record registered when the sensor was not deployed on the animal, record with impossible values, etc.

activity_data_animals_code04:	Table with activity data associated to animals with sensor mode code = 4 (see table lu_tables.lu_activity_sensor_mode). Data come from different type of sensors, thus the information must be properly precessed to be correctly analysized. There is 1 table per sensor mode code to reduce the overall size and to keep different things separated.

Attribute	Type	Description
activity_data_animals_code04_id	integer	Eurodeer identifier for the activity record
animals_id	integer	Eurodeer identifier for the animal
activity_sensors_id	integer	Eurodeer identifier for the activity sensor
act_1	double precision	The meaning of this field depends on the sensor_mode. Usually it is the X axis accellerometer.
act_2	double precision	The meaning of this field depends on the sensor_mode. Usually it is the y axis accellerometer.
act_3	double precision	The meaning of this field depends on the sensor_mode. Usually it is the z axis accellerometer.
acquisition_time	timestamp without time zone	Date and time of acquisition of the activity data (with time zone)
temperature_activity	double precision	Temperature measured by the sensor associated to the accelerometer
gps_positions_animals_id	bigint	This filed links the activity record to the (closer in time) valid GPS position

activity_sensor_mode_code	integer	This field explains the meaning of the act1,2,3. There 3 values can have a different meaning according to the sensor and in case also to the operational mode the sensor is used. See table activity_sensor_mode.
activity_validity_code	integer	This field marks each record as valid or invalid for different reasons: record registered when the sensor was not deployed on the animal, record with impossible values, etc.

activity_data_animals_code05:	Table with activity data associated to animals with sensor mode code = 5 (see table lu_tables.lu_activity_sensor_mode). Data come from different type of sensors, thus the information must be properly precessed to be correctly analyzed. There is 1 table per sensor mode code to reduce the overall size and to keep different things separated.

Attribute	Type	Description
activity_data_animals_code05_id	integer	Eurodeer identifier for the activity record
animals_id	integer	Eurodeer identifier for the animal
activity_sensors_id	integer	Eurodeer identifier for the activity sensor
act_1	double precision	The meaning of this field depends on the sensor_mode. Usually it is the X axis accellerometer.
act_2	double precision	The meaning of this field depends on the sensor_mode. Usually it is the y axis accellerometer.
act_3	double precision	The meaning of this field depends on the sensor_mode. Usually it is the z axis accellerometer.
acquisition_time	timestamp without time zone	Date and time of acquisition of the activity data (with time zone)
temperature_activity	double precision	Temperature measured by the sensor associated to the accelerometer
gps_positions_animals_id	bigint	This filed links the activity record to the (closer in time) valid GPS position
activity_sensor_mode_code	integer	This field explains the meaning of the act1,2,3. There 3 values can have a different meaning according to the sensor and in case also to the operational mode the sensor is used. See table activity_sensor_mode.
activity_validity_code	integer	This field marks each record as valid or invalid for different reasons: record registered when the sensor was not deployed on the animal, record with impossible values, etc.

activity_sensors:	Catalogue of activity sensors. Each sensor belongs to a research group. The attributes include the brand and model. The id used in the original data set is also included.

Attribute	Type	Description
activity_sensors_id	integer	Eurodeer identifier for activity sensor.
research_groups_id	integer	Id of the research group that owns the activity sensor.
vendor	character varying	Company that produced the sensor.
activity_sensors_original_id	character varying	Identifier of the activity sensor in the original data set.
model	character varying	Model of the activity sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
gps_sensors_id	integer	In case the activity sensor is integrated with a GPS sensor, this field reports the Eurodeer code of the GPS sensor (in any case this relationship is implicitly contained in the two deployment tables.

activity_sensors_animals:	None

Attribute	Type	Description
activity_sensors_animals_id	integer	Eurodeer identifier of the deployment.
activity_sensors_id	integer	Eurodeer identifier of the activity sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.
end_time	timestamp with time zone	Time and date of the end of the deployment.
notes	character varying	Open field where general notes on the deployment can be added.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

activity_sensors_animals:	Table with the information on the deployments of activity sensors on animals (starting and ending date and time of the deployment).

Attribute	Type	Description

activity_sensors_animals_id	integer	Eurodeer identifier of the deployment.
activity_sensors_id	integer	Eurodeer identifier of the activity sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.
end_time	timestamp with time zone	Time and date of the end of the deployment.
notes	character varying	Open field where general notes on the deployment can be added.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

animals:	Table with the information on the animals that are part of Eurodeer project. Each animal can belong to a study area, which is related to a research group. Information included in the table are sex, age, name and id in the original data set. It also stores data on which kind of sensors were applied to animals.

Attribute	Type	Description
animals_id	integer	Database id of population. Each animal belongs to a population, which is part of a study area. The same study area can have multiple populations. Linked with the table main.populations.
study_areas_id	integer	Study area where the animal is located (reference to table main.study_area).
animals_original_id	character varying	Identifier of the animal in the original data set.
animals_original_name	character varying	Nome of the animal in the original data set.
sex	"char"	Code for sex. It can be either "f" (female) or "m" (male). When the sex is not known, the field can be left empty.
first_capture_date	date	Date when the animal was captured the first time. This information is important to interpret the age_class.
monitored_activity	integer	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_gps	integer	If the animals has GPS data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.

monitored_vhf	integer	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
age_class_code_capture	integer	Code of the age class (reference to table lu_tables.lu_age_class) at the first capture of the animal (this information must be provided to give full meaning to the age class, as it is a dynamic attribute that changes over time).
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
reintroduction	integer	If 1, the animal has been reintroduced, if 0, the animal is not reintroduced.
populations_id	integer	None
year_birth	integer	Year of birth (when known). In the year_birth_exact field it is described if this is the exact year of birth of just an extimation (minimum year of birth).
year_birth_exact	boolean	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).
gps_deployed	boolean	If a gps sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
gps_data	boolean	If the animals has gps data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.

activity_deployed	boolean	If an activity sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
activity_data	boolean	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
vhf_deployed	boolean	If a vhf sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
vhf_data	boolean	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
capture_data	boolean	If the animals has capture data associated, then this field is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.

animals: Table with the information on the animals (red deer) that are part of Eurodeer project. Each animal can belong to a study area, which is related to a research group. Information included in the table are sex, age, name and id in the original data set. It also stores data on which kind of sensors were applied to animals.

Attribute	Type	Description
animals_id	integer	Database id of population. Each animal belongs to a population, which is part of a study area. The same study area can have multiple populations. Linked with the table main.populations.
study_areas_id	integer	Study area where the animal is located (reference to table main.study_area).
animals_original_id	character varying	Identifier of the animal in the original data set.
animals_original_name	character varying	Nome of the animal in the original data set.
sex	"char"	Code for sex. It can be either "f" (female) or "m" (male). When the sex is not known, the field can be left empty.
first_capture_date	date	Date when the animal was captured the first time. This information is important to interpret the age_class.

monitored_activity	integer	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_gps	integer	If the animals has GPS data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_vhf	integer	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
age_class_code_capture	integer	Code of the age class (reference to table lu_tables.lu_age_class) at the first capture of the animal (this information must be provided to give full meaning to the age class, as it is a dynamic attribute that changes over time).
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
reintroduction	integer	If 1, the animal has been reintroduced, if 0, the animal is not reintroduced.
populations_id	integer	None
year_birth	integer	Year of birth (when known). In the year_birth_exact field it is described if this is the exact year of birth of just an extimation (minimum year of birth).
year_birth_exact	boolean	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).
gps_deployed	boolean	If a gps sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.

gps_data	boolean	If the animals has gps data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
activity_deployed	boolean	If an activity sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
activity_data	boolean	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
vhf_deployed	boolean	If a vhf sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
vhf_data	boolean	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
capture_data	boolean	If the animals has capture data associated, then this field is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.

animals: Table with the information on the animals

Attribute	Type	Description
animals_id	integer	Database id of population. Each animal belongs to a population, which is part of a study area. The same study area can have multiple populations. Linked with the table main.populations.
study_areas_id	integer	Study area where the animal is located (reference to table main.study_area).
animals_original_id	character varying	Identifier of the animal in the original data set.
animals_original_name	character varying	Nome of the animal in the original data set.
sex	"char"	Code for sex. It can be either "f" (female) or "m" (male). When the sex is not known, the field can be left empty.
first_capture_date	date	Date when the animal was captured the first time. This information is important to interpret the age_class.

monitored_activity	integer	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_gps	integer	If the animals has GPS data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_vhf	integer	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
age_class_code_capture	integer	Code of the age class (reference to table lu_tables.lu_age_class) at the first capture of the animal (this information must be provided to give full meaning to the age class, as it is a dynamic attribute that changes over time).
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
reintroduction	integer	If 1, the animal has been reintroduced, if 0, the animal is not reintroduced.
populations_id	integer	None
year_birth	integer	Year of birth (when known). In the year_birth_exact field it is described if this is the exact year of birth of just an extimation (minimum year of birth).
year_birth_exact	boolean	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).
gps_deployed	boolean	If a gps sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.

gps_data	boolean	If the animals has gps data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
activity_deployed	boolean	If an activity sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
activity_data	boolean	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
vhf_deployed	boolean	If a vhf sensor was deployed on an animal, which does not necessarily generated data (e.g. death at capture), then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
vhf_data	boolean	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
capture_data	boolean	If the animals has capture data associated, then this field is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.

animals_captures:	Table with the information on captures. Every animals can be captured more than once. Only captures related to monitoring are included. This include successful collaring, but also recapture of a collared animal or failed collaring because for example the death of the animal during the capture (it is possible to have a capture event without any deployment registered).

Attribute	Type	Description
animals_captures_id	integer	Database ID of the capture of an animal. Each animal can have multiple captures.
animals_id	integer	Database ID of the animal (external key to the table main.animals).
capture_timestamp	timestamp with time zone	Time and date (with time zone) when the animal was captured (animal fall in the trap, box, net, etc.).
release_timestamp	timestamp with time zone	Time and date (with time zone) when the animal was released (animal taken out from the box, trap, net etc. or put back into a transportation box).

handling_start	timestamp with time zone	Time and date (with time zone) when the animal handling started (taken out from the box, trap, net etc. and is in direct contact with people).
handling_end	timestamp with time zone	Time and date (with time zone) when the animal handling ended (animals is no more in direct contact with people for marking and measurements or it is released or put back in to a transportation box).
longitude	double precision	Coordinate of the capture (can be an approximation).
latitude	double precision	Coordinate of the capture (can be an approximation).
geom	USER-DEFINED	Location (point) of the capture (can be an approximation).
first_capture	boolean	Specify if this is the first capture of the animal (yes/no).
collared	boolean	Specify if the animal was collared during the capture (yes/no).
gps_sensors_animals_id	integer	In case the animal has been collared with GPS, this is the id of the related deployment.
sedation	boolean	Specify if the animal was sedated (yes/no).
sampling_faeces	boolean	Specify a sample was taken for faeces.
sampling_biopsy	boolean	Specify a sample was taken for biopsy (tissue).
sampling_blood	boolean	Specify a sample was taken for blood.
sampling_hair	boolean	Specify a sample was taken for hair.
sampling_notes	character varying	Description of the samples taken.
injury	boolean	Specify if the animal was injured during the capture.
injury_description	character varying	Description of the injury (in any).
death	boolean	Specify if the animal died during the capture.
death_description	character varying	Description of the death (if this occurred).
behavior_handling_code	integer	Code of he behavior during handling (linked to a look up table).
behavior_release_code	integer	Code of he behavior during the release (linked to a look up table).
heart_rate_start	double precision	First measure of the heart rate (beats/min).
heart_rate_start_timestamp	timestamp with time zone	Timestamp when the first heart rate was measured.
heart_rate_end	double precision	Last measure of the heart rate (beats/min).
heart_rate_end_timestamp	timestamp with time zone	Timestamp when the last heart rate was measured.

hindfoot_length_cm	double precision	Measure of the hind foot (cm).
body_mass_kg	double precision	Measure of the body mass (Kg).
rectal_temperature_c	double precision	Measure of the rectal temperature (°C).
notes	character varying	General notes on the capture.
capture_methods_code	integer	Method used for the capture (linked to a look up table).
vhf_sensors_animals_id	integer	In case the animal has been collared with VHF, this is the id of the related deployment.
capture_result_code	integer	Code of the result of the capture. Only animals that are monitored or that were captured to be monitored are included in the data base. This field specifies what happened (animal collared, animal death, etc).

animals_captures:	Table with the information on captures. Every animals can be captured more than once.

Attribute	Type	Description
animals_captures_id	integer	Database ID of the capture of an animal. Each animal can have multiple captures.
animals_id	integer	Database ID of the animal (external key to the table main.animals).
capture_timestamp	timestamp with time zone	Time and date (with time zone) when the animal was captured (animal fall in the trap, box, net, etc.).
release_timestamp	timestamp with time zone	Time and date (with time zone) when the animal was released (animal taken out from the box, trap, net etc. or put back into a transportation box).
handling_start	timestamp with time zone	Time and date (with time zone) when the animal handling started (taken out from the box, trap, net etc. and is in direct contact with people).
handling_end	timestamp with time zone	Time and date (with time zone) when the animal handling ended (animals is no more in direct contact with people for marking and measurements or it is released or put back in to a transportation box).
longitude	double precision	Coordinate of the capture (can be an approximation).
latitude	double precision	Coordinate of the capture (can be an approximation).
geom	USER-DEFINED	Location (point) of the capture (can be an approximation).
first_capture	boolean	Specify if this is the first capture of the animal (yes/no).
collared	boolean	Specify if the animal was collared during the capture (yes/no).

gps_sensors_animals_id	integer	In case the animal has been collared with GPS, this is the id of the related deployment.
sedation	boolean	Specify if the animal was sedated (yes/no).
sampling_faeces	boolean	Specify a sample was taken for faeces.
sampling_biopsy	boolean	Specify a sample was taken for biopsy (tissue).
sampling_blood	boolean	Specify a sample was taken for blood.
sampling_hair	boolean	Specify a sample was taken for hair.
sampling_notes	character varying	Description of the samples taken.
injury	boolean	Specify if the animal was injured during the capture.
injury_description	character varying	Description of the injury (in any).
death	boolean	Specify if the animal died during the capture.
death_description	character varying	Description of the death (if this occurred).
behavior_handling_code	integer	Code of he behavior during handling (linked to a look up table).
behavior_release_code	integer	Code of he behavior during the release (linked to a look up table).
heart_rate_start	double precision	First measure of the heart rate (beats/min).
heart_rate_start_timestamp	timestamp with time zone	Timestamp when the first heart rate was measured.
heart_rate_end	double precision	Last measure of the heart rate (beats/min).
heart_rate_end_timestamp	timestamp with time zone	Timestamp when the last heart rate was measured.
hindfoot_length_cm	double precision	Measure of the hind foot (cm).
body_mass_kg	double precision	Measure of the body mass (Kg).
rectal_temperature_c	double precision	Measure of the rectal temperature (°C).
notes	character varying	General notes on the capture.
capture_methods_code	integer	Method used for the capture (linked to a look up table).
vhf_sensors_animals_id	integer	In case the animal has been collared with VHF, this is the id of the related deployment.
capture_result_code	integer	Code of the result of the capture. Only animals that are monitored or that were captured to be monitored are included in the data base. This field specifies what happened (animal collared, animal death, etc).

animals_captures_sedation:

Table with the information on sedations during captures.

Attribute	Type	Description
animals_captures_sedation_id	integer	Database ID of the sedation of an animal during the capture.
animals_captures_id	integer	Database ID of the capture of an animal (external key to the table main.animals_captures)
sedation_drug	boolean	Specify if the animal was sedated (yes/no).
sedation_drug_used	character varying	Specify what drug was used.
sedation_drug_quantity	double precision	Specify the amount of drug used (ml).
sedation_drug_timestamp	timestamp with time zone	Time and date (with time zone) when the animal was sedated.
antidote_drug	boolean	Specify if an antidote was given to the animal (yes/no).
antidote_drug_used	character varying	Specify what drug was used as antidote.
antidote_drug_quantity	double precision	Specify the amount of antidote drug used (ml).
antidote_drug_timestamp	timestamp with time zone	Time and date (with time zone) when the animal was given the antidote.
notes	character varying	General notes on th sedation.

animals_contacts:

Table with the information on contacts with animals. These can be both sightings of the animal – dead or alive – or the finding of the sensor. This information is specially usefull for survival analysis.

Attribute	Type	Description
animals_contacts_id	integer	Database ID of the contact with the animal. Each animal can have multiple contacts.
animals_id	integer	Database ID of the animal (external key to the table main.animals).
status	character varying	Status of the animal recorder at contact (possible values: dead, alive, unknown).
mortality_code	integer	If the status is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
contact_timestamp	timestamp with time zone	Time and date of the contact.
notes	character varying	Open field where general notes on the contact can be added.

feeding_sites:

This is the table containing all the information on the management of the feeding sites, where these are used (i.e ten study areas)

Attribute	Type	Description
feeding_site_id	integer	primary key of the table
research_groups_id	integer	the id of the research group from which the information of the feeding site comes. The id is the same used in the table main.research_groups
study_areas_id	integer	it is the same code used in the table main.study_areas to identify the study areas
study_name	character varying	name of the study area, as indicated in the table main.study_areas
project	character varying	name of the specific project within the study area
feeding_site_original_id	character varying	name of the feeding site as it was provided by data owners
fs_id	character varying	new created id for each feeding site, for simplicity in analysis and database management. It is the primary key of the table
year_start	integer	first year of management
year_end	integer	last year of management
feeding_site_code	integer	it indicates if the site is a proper feeding station (1) or a box trap (2). See look up table feeding_site_type for details
moving_site	boolean	it indicates if the feeding site is moved (1) or not (0) during the management period
feeding_management	integer	it indicates if the feeding site is foraged ad libitum (1) or on an occasional basis (2)
feeding_frequency_code	integer	it indicates the code of frequency with whom the feeding site is provided with food. See look up table feeding_ frequency_categorized for details
food_items	character varying	it indicates the food items with whom the feeding sites are filled, as provided by data owners
food_energy_code	character varying	classification of food quality according to energetic values of the items provided. See look up table feeding_site_feeding quality for details
day_start_feeding	integer	activation day of management, does not change from year to year in case of multiple years of management
month_start_feeding	integer	activation month of management, does not change from year to year in case of multiple years of management
day_end_feeding	integer	deactivation day of management, does not change from year to year in case of multiple years of management
month_end_feeding	integer	deactivation month of management, does not change from year to year in case of multiple years of management
species_excluded	character varying	it indicates if and what species are excluded from access to the feding site
other_feeders	character varying	it indicates if and what other species feed on the feeding site
potential_competition	boolean	if there are other ungulates (red deer, muflon, wild boar, bisons) feeding on the station AND they are not prevented from access to food, then 1, otherwise 0

predator_species	character varying	it indicates if and what predators live in the proximity of feeding sites
potential_predation	boolean	if among the predators there are any which is of relevance for roe deer, then 1, otherwise 0
managed_by	character varying	managers of the feeding sites
latitude	double precision	latitude of the feeding site
longitude	double precision	longitude of the feeding site
utm_y	integer	coordinate of the feeding site in utm
utm_x	integer	coordinate of the feeding site in utm
srid_code	integer	epsg code on the utm zone of the fs
notes	character varying	None
geom	USER-DEFINED	Geometry of the location (point)
corine_land_cover_2006_code	integer	Code of the Corine lad cover class produced in 2006 (in env_data.corine_land_cover_legend there is a complete description of these codes)
altitude_srtm	integer	Meters above sea level (from SRTM project)
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm	integer	Degrees calculated counterclockwise from east (source: SRTM project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster	integer	Degrees calculated counterclockwise from east (source: ASTER project)
corine_land_cover_2000_code	integer	Code of the Corine lad cover class produced in 2000 (in env_data.corine_land_cover_legend there is a complete description of these codes)
corine_land_cover_1990_code	integer	Code of the Corine lad cover class produced in 1990 (in env_data.corine_land_cover_legend there is a complete description of these codes)

gps_data_animals:	Table with GPS locations data associated to animals and with a list of derived ancillary infomation calculated using the information on coordinates and acquisition time, and intersecting with environmental layers.

Attribute	Type	Description
gps_data_animals_id	integer	Eurodeer identifier for the location
animals_id	integer	Eurodeer identifier for the animal

gps_sensors_id	integer	Eurodeer identifier for the GPS sensor
acquisition_time	timestamp with time zone	Date and time of acquisition of the GPS coordinates (with time zone)
x_original_reference	double precision	Coordinate X as computed by the software connected to the GPS sensor (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as computed by the software connected to the GPS sensor (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates provided by the software connected to the GPS sensor
latitude	double precision	Latitude recorded by the GPS sensor
longitude	double precision	Longitude recorded by the GPS sensor
altitude_gps	integer	Altitude recorded by the GPS sensor (related to the centre of the earth)
dop	double precision	Dilution Of Precision
sats	integer	Number of satellites used by the GPS sensor to calculate the coordinates
temperature_sensor	double precision	Temperature as measured by the sensor associated to the GPS
geom	USER-DEFINED	Geometry of the location (point)
gps_validity_code	smallint	This field tags the record according to its "validity" or degree of reliability (explanation of codes in lu_tables.lu_gps_validity)
corine_land_cover_2006_code	integer	Code of the Corine lad cover class produced in 2006 (in env_data.corine_land_cover_legend there is a complete description of these codes)
ndvi_modis	real	NDVI derived from MODIS (16-daily, non smoothed, assciated to the closest [in time] image). For analysis, it is recommended to use smoothed data (smoothed or boku).
ndvi_vegetation	real	NDVI derived from SPOT VEGETATION. SPOT Vegetation sensor does not record any information sine end of 2014.
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)

aspect_srtm_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat) (source: SRTM project)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat)(source: ASTER project)
corine_land_cover_2000_code	integer	Code of the Corine lad cover class produced in 2000 (in env_data.corine_land_cover_legend there is a complete description of these codes)
corine_land_cover_1990_code	integer	Code of the Corine lad cover class produced in 1990 (in env_data.corine_land_cover_legend there is a complete description of these codes)
fapar_vegetation	real	FAPAR derived from SPOT VEGETATION.SPOT Vegetation sensor does not record any information sine end of 2014.
ndvi_modis_boku	double precision	NDVI derived from MODIS as processed by Boku (smoothed weekly images). NDVI is interpolated between the two closest images.
ndvi_modis_smoothed	double precision	None
update_core_user	character varying	User who modified the core elements of record (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
update_user	character varying	User who modified the record (last time).
insert_user	character varying	User who created the record.
update_core_timestamp	timestamp with time zone	Date and time when the core elements of record was updated (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
aspect_srtm_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat) (source: SRTM project)
aspect_aster_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat)(source: ASTER project)

gps_data_animals: None

Attribute	Type	Description
gps_data_animals_id	integer	Eurodeer identifier for the location

animals_id	integer	Eurodeer identifier for the animal
gps_sensors_id	integer	Eurodeer identifier for the GPS sensor
acquisition_time	timestamp with time zone	Date and time of acquisition of the GPS coordinates (with time zone)
x_original_reference	double precision	Coordinate X as computed by the software connected to the GPS sensor (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as computed by the software connected to the GPS sensor (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates provided by the software connected to the GPS sensor
latitude	double precision	Latitude recorded by the GPS sensor
longitude	double precision	Longitude recorded by the GPS sensor
altitude_gps	integer	Altitude recorded by the GPS sensor (related to the centre of the earth)
dop	double precision	Dilution Of Precision
sats	integer	Number of satellites used by the GPS sensor to calculate the coordinates
temperature_sensor	double precision	Temperature as measured by the sensor associated to the GPS
geom	USER-DEFINED	Geometry of the location (point)
gps_validity_code	smallint	This field tags the record according to its "validity" or degree of reliability (explanation of codes in lu_tables.lu_gps_validity)
corine_land_cover_2006_code	integer	Code of the Corine lad cover class produced in 2006 (in env_data.corine_land_cover_legend there is a complete description of these codes)
ndvi_modis	real	NDVI derived from MODIS (16-daily, non smoothed, assciated to the closest [in time] image). For analysis, it is recommended to use smoothed data (smoothed or boku).
ndvi_vegetation	real	NDVI derived from SPOT VEGETATION. SPOT Vegetation sensor does not record any information sine end of 2014.
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
altitude_srtm	integer	Meters above sea level (from SRTM project)

slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat) (source: SRTM project)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat)(source: ASTER project)
corine_land_cover_2000_code	integer	Code of the Corine lad cover class produced in 2000 (in env_data.corine_land_cover_legend there is a complete description of these codes)
corine_land_cover_1990_code	integer	Code of the Corine lad cover class produced in 1990 (in env_data.corine_land_cover_legend there is a complete description of these codes)
fapar_vegetation	real	FAPAR derived from SPOT VEGETATION.SPOT Vegetation sensor does not record any information sine end of 2014.
ndvi_modis_boku	double precision	NDVI derived from MODIS as processed by Boku (smoothed weekly images). NDVI is interpolated between the two closest images.
ndvi_modis_smoothed	double precision	None
update_core_user	character varying	User who modified the core elements of record (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
update_user	character varying	User who modified the record (last time).
insert_user	character varying	User who created the record.
update_core_timestamp	timestamp with time zone	Date and time when the core elements of record was updated (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
aspect_srtm_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat) (source: SRTM project)
aspect_aster_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat)(source: ASTER project)

gps_data_animals:	Table with GPS locations data associated to animals (red deer) with a list of derived ancillary infomation calculated using the information on coordinates and acquisition time, and intersecting with environmental layers.

Attribute	Type	Description
gps_data_animals_id	integer	Eurodeer identifier for the location
animals_id	integer	Eurodeer identifier for the animal
gps_sensors_id	integer	Eurodeer identifier for the GPS sensor
acquisition_time	timestamp with time zone	Date and time of acquisition of the GPS coordinates (with time zone)
x_original_reference	double precision	Coordinate X as computed by the software connected to the GPS sensor (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as computed by the software connected to the GPS sensor (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates provided by the software connected to the GPS sensor
latitude	double precision	Latitude recorded by the GPS sensor
longitude	double precision	Longitude recorded by the GPS sensor
altitude_gps	integer	Altitude recorded by the GPS sensor (related to the centre of the earth)
dop	double precision	Dilution Of Precision
sats	integer	Number of satellites used by the GPS sensor to calculate the coordinates
temperature_sensor	double precision	Temperature as measured by the sensor associated to the GPS
geom	USER-DEFINED	Geometry of the location (point)
gps_validity_code	smallint	This field tags the record according to its "validity" or degree of reliability (explanation of codes in lu_tables.lu_gps_validity)
corine_land_cover_2006_code	integer	Code of the Corine lad cover class produced in 2006 (in env_data.corine_land_cover_legend there is a complete description of these codes)
ndvi_modis	real	NDVI derived from MODIS (16-daily, non smoothed, assciated to the closest [in time] image). For analysis, it is recommended to use smoothed data (smoothed or boku).
ndvi_vegetation	real	NDVI derived from SPOT VEGETATION. SPOT Vegetation sensor does not record any information sine end of 2014.
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_y	integer	Y coordinate projected in the utm_srid UTM zone

altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat) (source: SRTM project)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat)(source: ASTER project)
corine_land_cover_2000_code	integer	Code of the Corine lad cover class produced in 2000 (in env_data.corine_land_cover_legend there is a complete description of these codes)
corine_land_cover_1990_code	integer	Code of the Corine lad cover class produced in 1990 (in env_data.corine_land_cover_legend there is a complete description of these codes)
fapar_vegetation	real	FAPAR derived from SPOT VEGETATION.SPOT Vegetation sensor does not record any information sine end of 2014.
ndvi_modis_boku	double precision	NDVI derived from MODIS as processed by Boku (smoothed weekly images). NDVI is interpolated between the two closest images.
ndvi_modis_smoothed	double precision	None
update_core_user	character varying	User who modified the core elements of record (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
update_user	character varying	User who modified the record (last time).
insert_user	character varying	User who created the record.
update_core_timestamp	timestamp with time zone	Date and time when the core elements of record was updated (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
aspect_srtm_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat) (source: SRTM project)
aspect_aster_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat)(source: ASTER project)

gps_sensors:	None

Attribute	Type	Description
gps_sensors_id	integer	Eurodeer identifier for GPS sensors.
research_groups_id	integer	Id of the research group that owns the GPS sensor.
gps_sensors_original_id	character varying	Identifier of the GPS sensor in the original data set.
vendor	character varying	Company that produced the sensor.
model	character varying	Model of the GPS sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

gps_sensors:	Catalogue of GPS sensors (red deer). Each sensor belongs to a research group. The attributes include the brand and the model. The id used in the original data set is also included.

Attribute	Type	Description
gps_sensors_id	integer	Eurodeer identifier for GPS sensors.
research_groups_id	integer	Id of the research group that owns the GPS sensor.
gps_sensors_original_id	character varying	Identifier of the GPS sensor in the original data set.
vendor	character varying	Company that produced the sensor.
model	character varying	Model of the GPS sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

gps_sensors:	Catalogue of GPS sensors. Each sensor belongs to a research group. The attributes include the brand and the model. The id used in the original data set is also included.

Attribute	Type	Description
gps_sensors_id	integer	Eurodeer identifier for GPS sensors.
research_groups_id	integer	Id of the research group that owns the GPS sensor.
gps_sensors_original_id	character varying	Identifier of the GPS sensor in the original data set.
vendor	character varying	Company that produced the sensor.
model	character varying	Model of the GPS sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

gps_sensors_animals: None

Attribute	Type	Description
gps_sensors_animals_id	integer	Eurodeer identifier of the deployment.
gps_sensors_id	integer	Eurodeer identifier of the GPS sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.
end_time	timestamp with time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
end_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

gps_sensors_animals: None

Attribute	Type	Description
gps_sensors_animals_id	integer	Eurodeer identifier of the deployment.
gps_sensors_id	integer	Eurodeer identifier of the GPS sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.
end_time	timestamp with time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.

notes	character varying	Open field where general notes on the deployment can be added.
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
end_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

gps_sensors_animals:	Table with the information on the deployments of GPS sensors on animals (red deer) (starting and ending date and time of the deployment).

Attribute	Type	Description
gps_sensors_animals_id	integer	Eurodeer identifier of the deployment.
gps_sensors_id	integer	Eurodeer identifier of the GPS sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.
end_time	timestamp with time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
end_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

gps_sensors_animals:	Table with the information on the deployments of GPS sensors on animals (starting and ending date and time of the deployment).

Attribute	Type	Description

gps_sensors_animals_id	integer	Eurodeer identifier of the deployment.
gps_sensors_id	integer	Eurodeer identifier of the GPS sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.
end_time	timestamp with time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
end_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

populations:	Table with the list of populations. Each study area can have multiple populations. Each population is characterized by a set of information (human disturbance, performance, predators, interspecific competition, hunting pressure, rad density, which are collected every one or more years.

Attribute	Type	Description
populations_id	integer	Database id that uniquely identifies each population.
study_areas_id	integer	Database id that uniquely identifies each population. Linked with the table main.study_areas.
geom	USER-DEFINED	Area of the population, defined as the union of the MCP of all the animals belonging to the population, plus a buffer of 200 meters.
population_name	character varying	Extended name of the population (if any).
note	character varying	Note related to the population.
hunting	boolean	Presence (yes/no) of roe deer hunting. More information should then be provided in the tables populations.hunting_pressure and populations.hunting_stats.
hunting_others	boolean	Presence (yes/no) of hunting of species differet from roe deer. More information should then be provided in the table populations.hunting_pressure_others.

populations_density:	Table with estimation of roe deer density. The information is linked to the method of estimation, and can have an estimation per year per methodof estimation. If know, also the start and end date of the sampling is reported, otherwise only the year and optionally the season.

Attribute	Type	Description
populations_density_id	integer	Database id of each record of the population density table.
populations_id	integer	Database id of each population (linked to main.populations table).
reference_year	integer	Reference year when data were collected/estimated.
seasons_code	integer	Season when the sampling was done (linked to a look up table).
start_date	date	Start date of the sampling (if known).
end_date	date	End date of the sampling (if known).
density	double precision	Roe deer densinty (individuals per squared kilometre).
density_se	double precision	Standard error of roe deer density.
sampling_methods_code	integer	Sampling method used for the estimation (linked to a look up table).
note	character varying	Notes related to the population density for a defined year.

populations_environment:	Table with the description of the environment of each population based on the intersection between the boundaries of the population's area and the environmental layers stored in the urodeer database. The table stil has to be developed.

Attribute	Type	Description
populations_environment_id	integer	Database id of each record of the environmental characterization table.
populations_id	integer	Database id of each population (linked to main.populations table).

populations_human_disturbance:	Table with data related to human disturbance (tourism agriculture, forestry works). This is connected to each population (you can have multiple population in the same study area). This information is expected to change very slowly, but a reference year is given in case a long term monitoring is possible.

Attribute	Type	Description
populations_human_disturbance_id	integer	Database id of each record of the human disturcance table.
populations_id	integer	Database id of each population (linked to main.populations table).

tourism_summer	boolean	Presence of tourists during summer (yes/no).
tourism_summer_pressure	integer	Estimation of the number of tourists during summer.
tourism_winter	boolean	Presence of tourists during winter (yes/no).
tourism_winter_pressure	integer	Estimation of the number of tourists during winter.
farming_code	integer	Type of farming, related to a look up table.
forestry_work_code	integer	Type of forestry work, related to a look up table.
note	character varying	Notes related to human disturbance.
farming	boolean	None
forestry_work	boolean	None
reference_year_start	integer	Reference start year when data were collected/ estimated and are considered valid.
reference_year_end	integer	Reference end year when data were collected/ estimated and are considered valid.

populations_hunting_pressure:	Table with the description of the hunting periods per sex (male and female) per population. The periods have a start (and end) month and day. More than a period per year is possible. If the period is across years, it should be devided in two sub periods (until 31 december and from 1st of january) As this information can change (e.g. modification to the hunting regulation) the time interval of validity can also be specified.

Attribute	Type	Description
populations_hunting_pressure_id	integer	Database id of each record of the population hunting pressure table.
populations_id	integer	Database id of each population (linked to main.populations table).
validity_start_date	date	Start date of the validity of the hunting intervals (i.e. start of validity of the regulation).
validity_end_date	date	End date of the validity of the hunting intervals (i.e. end of validity of the regulation). This is null if the regulation is still valid.
hunting_start_month	integer	Start month of the hunting period.
hunting_start_day	integer	Start day of the hunting period.
hunting_end_month	integer	End date of the hunting period.
hunting_end_day	integer	End date of the hunting period.
dogs	boolean	Presence (yes/no) of dogs.
sex	character varying	Code for sex. It can be either "f" (female) or "m" (male). When the sex is not known, the field can be left empty.
note	character varying	Notes related to the population hunting pressure.
hunting_presence	boolean	None

hunting_marked_animals	boolean	None
hunting_method_code	integer	None

populations_hunting_pressure_method:

Table with the list of hunting methods used in a hunting period (table main.population_hunting_pressure). Each hunting period can have more than one hunting method.

Attribute	Type	Description
populations_hunting_pressure_method_id	integer	Database id of each record of the population hunting pressure table (external key).
hunting_method_code	integer	Hunting method (linked to a look up table).
note	character varying	Notes related to the population hunting method.

populations_hunting_pressure_others:

Table with the description of the hunting periods for species different from roe deer. The species is not reported. The periods have a start (and end) month and day. More than a period per year is possible. If the period is across years, it should be devided in two sub periods (until 31 december and from 1st of january) As this information can change (e.g. modification to the hunting regulation) the time interval of validity can also be specified.

Attribute	Type	Description
populations_hunting_pressure_others_id	integer	Database id of each record of the population hunting (for species different from roe deer) table.
populations_id	integer	Database id of each population (linked to main.populations table).
validity_start_date	date	Start date of the validity of the hunting intervals (i.e. start of validity of the regulation).
validity_end_date	date	End date of the validity of the hunting intervals (i.e. end of validity of the regulation). This is null if the regulation is still valid.
hunting_start_month	integer	Start month of the hunting period.
hunting_start_day	integer	Start day of the hunting period.
hunting_end_month	integer	End date of the hunting period.
hunting_end_day	integer	End date of the hunting period.
dogs	boolean	Presence (yes/no) of dogs.
note	character varying	Notes related to the population hunting pressure.
hunting_presence	boolean	None

populations_hunting_stats:	Table with information of statistics taken from animals hunted and measured for biometry related to a specific year. This is connected to populations.

Attribute	Type	Description
populations_hunting_stats_id	integer	Database id of each record of the hunting statistcs table.
populations_id	integer	Database id of each population (linked to main.populations table).
reference_year	integer	reference year when data were collected/estimated.
males_hunted	integer	Number of adult males hunted and measured for biometry
females_hunted	integer	Number of adult females hunted and measured for biometry
fawns_hunted	integer	Number of fawns (<1 year) hunted and measured for biometry
males_bodymass_avg	double precision	Average body mass (in kg) of adult males, from hunting statistics
males_bodymass_se	double precision	Standard error of the body mass of adult males, from hunting statistics
females_bodymass_avg	double precision	Average body mass (in kg) of adult females, from hunting statistics
females_bodymass_se	double precision	Standard error of the body mass of adult females, from hunting statistics
fawns_bodymass_avg	double precision	Average body mass (in kg) of fawns, from hunting statistics
fawns_bodymass_se	double precision	Standard error of the body mass of fawns, from hunting statistics
males_hindfootlenght_avg	double precision	Average hindfoot lenght (in cm) of adult males, from hunting statistics
males_hindfootlenght_se	double precision	Standard error of the hindfoot lenght of adult males, from hunting statistics
females_hindfootlenght_avg	double precision	Average hindfoot lenght (in cm) of adult females, from hunting statistics
fawns_hindfootlenght_avg	double precision	Average hindfoot lenght (in cm) of fawns, from hunting statistics
fawns_hindfootlenght_se	double precision	Standard error of the hindfoot lenght of fawns, from hunting statistics
females_fawns_hunting	boolean	Yes if adult females and fawns are hunted in the same period
note	character varying	Notes related to hunting statistcs for a defined year.

populations_interspecific_competitors:

Table with estimation of roe deer interspecific competitors. The information is linked to the method of estimation, and can have an estimation per year per methodof estimation. If know, also the start and end date of the sampling is reported, otherwise only the year and optionally the season.

Attribute	Type	Description
populations_interspecific_competitors_id	integer	Database id of each record of the population interspecific_competitors table.
populations_id	integer	Database id of each population (linked to main.populations table).
species_id	integer	Id of the species of the competitor (linked to main.species).
reference_year	integer	Reference year when data were collected/ estimated.
seasons_code	integer	Season when the sampling was done (linked to a look up table).
start_date	date	Start date of the sampling (if known).
end_date	date	End date of the sampling (if known).
competitors_presence	boolean	Presence (yes/no) of the specific competitor.
competitors_density_code	integer	Class of density of the competitor (linked to a look up table).
competitors_density	double precision	Competitor density (individuals per squared kilometre) if available.
competitors_density_se	double precision	Standard error of roe deer interspecific_competitors.
sampling_methods_code	integer	Sampling method used for the estimation of the competitor density (linked to a look up table).
note	character varying	Notes related to the interspecific competitors for a defined year/sampling method.

populations_predators:

Table with estimation of predators density. Each estimation for each predator for each year and for each sampling method corresonds to a row. The id of the species is linked with main.species where all the species are listed. If available, an estimation of the density is given as class and (if possible) number of individual per squared kilometre, otherwise only presence/absence is reported.

Attribute	Type	Description
populations_predators_id	integer	Database id of each record of the predators table.
populations_id	integer	Database id of each population (linked to main.populations table).
reference_year	integer	Reference year when data were collected/estimated.
species_id	integer	Id of the species of the predator (linked to main.species).

predators_presence	boolean	Presence (yes/no) of the specific predator.
predators_density_code	integer	Class of density of the predator (linked to a look up table).
predators_density	double precision	Density of the predators (individual per squared kilometre) if available.
predators_density_se	double precision	Standard error of predator density (if available).
sampling_methods_code	integer	Sampling method used for the estimation (linked to a look up table).
note	character varying	Notes related to the predator density for a defined year/ sampling method.

populations_roads_density:	Table with estimation of road density density in the study area of a specific population. The value is linked to a specific year. It might change both because the road network change or because the areal of the population changes.

Attribute	Type	Description
populations_roads_density_id	integer	Database id of each record of the population roads density table.
populations_id	integer	Database id of each population (linked to main.populations table).
density	double precision	Road density in the study area (kilometres of roads per squared kilometre).
density_se	double precision	Standard error of road density.
data_source	character varying	Data used to estimate the road density (description).
note	character varying	Notes related to the road density.
reference_year_start	integer	Reference start year when data were collected/estimated and are considered valid.
reference_year_end	integer	Reference end year when data were collected/estimated and are considered valid.

research_groups:	Research groups are the highest level in the hierarchy of the database. Each research group can have many study areas and can own many collars.

Attribute	Type	Description
research_groups_id	integer	Eurodeer identifier for research groups.
research_group_name	character varying	Name of the research group.
contact	character varying	Contact person of the research group for the Eurodeer project.

institution	character varying	Institute of the research group.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
short_name	character varying	Short name of the research group.
country	character varying	Country of the research group.
geom	USER-DEFINED	Approximate location of the research group.
year_joined	integer	Year when the group joined the Eurodeer project.
data_roedeer	boolean	Red deer data in the database.
data_reddeer	boolean	None

species:	List of species with common name, scientific name, genus, family, order, class. This informmation is used by different tables in the database to reference species with a database id.

Attribute	Type	Description
species_id	integer	Database id that uniquely identifies each species.
common_name	character varying	The column shows the species common name in English.
species_scientific_name	character varying	The column shows the scientific name of the species.
species_genus	character varying	The column shows the genus of the species.
species_family	character varying	This column shows the family of the species.
species_order	character varying	This column shows the order name of the species.
species_class	character varying	This column shows the class of the species.
note	character varying	Notes related to the species.

study_areas:	Study areas are the areas monitored by research groups. Each study area can have many animals (red deer). Study areas can have defined, approximated, or no specific spatial boundaries.

Attribute	Type	Description
study_areas_id	integer	Eurodeer identifier for study areas.
study_name	character varying	Name of the study area.
geom	USER-DEFINED	Multi polygons layer of study areas. This spatial layer can be used as a reference to locate the study areas. Study areas can have defined boundaries (e.g. fenced). In this case, the field "defined_bundaries" is set to 1. Otherwise a reference boundary is created as the convex hull polygon (plus a buffer of 1 km) of the existing locations. These boundaries should be updated whenever a new set of locations is uploaded in the database.

research_groups_id	integer	Indentifier of the research group that is monitoring animals in this study area.
defined_boundaries	integer	If the study area boundaries are defined by research groups (e.g. areas fenced, or know area of animals' movement), this filed is set to 1. There polygons are not modified if new locations are uploaded. For areas with non defined boundaries (tag = 0), the boundaries are calculated on the convex hull polygon of the existing locations, and thus are updated whenever new locations are uploaded.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
short_name	character varying	Short version of the study area name for maps and short reports.
geom_mcp_individuals	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the MCP of each individual is calulated, then al mcp are merged and 500 meters buffer is added. These boundaries should be updated whenever a new set of locations is uploaded in the database. the code to update: update main.study_areas set geom_mcp_individuals = st_multi(foo.qq) from (select studies_id as ww,st_buffer(((st_multi(st_union (geom))))::geometry(multipolygon, 4326)::geography, 500)::geometry qq from analysis.view_convexhull group by studies_id) as foo where defined_boundaries = 0 and study_areas.study_areas_id = foo.ww;
geom_traj_buffer	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the buffer of 1 km (on both sides) is calculated from the trajectories of all the animals (1 location every 24 hours is considered). Th polygons of the same study area are then merged to generate a multipolygon for each study area). These boundaries should be updated whenever a new set of locations is uploaded in the database. DROP TABLE IF EXISTS temp.locations_12h_traj; CREATE TABLE temp.locations_24h_traj AS SELECT animals_id, study_areas_id , foo2.geom::geometry(LineString,4326) AS geom FROM (SELECT foo.animals_id,study_areas_id, st_makeline(foo.geom) AS geom FROM (SELECT study_areas_id, geom, animals_id, acquisition_time FROM temp.locations_24h ORDER BY study_areas_id, animals_id, acquisition_time) foo GROUP BY foo.animals_id, study_areas_id) foo2 WHERE st_geometrytype (foo2.geom) = 'ST_LineString'::text; ALTER TABLE temp.locations_24h_traj ADD COLUMN geom_buffer geometry (polygon,4326); UPDATE temp.locations_24h_traj set geom_buffer = (st_buffer(st_simplify(geom, 0.001)::geography, 1000))::geometry DROP TABLE IF EXISTS temp.locations_24h_studyareas; CREATE TABLE temp.locations_24h_studyareas AS SELECT study_areas_id, st_multi(st_union(geom_buffer))::geometry(multipolygon, 4326) geom from temp.locations_24h_traj group by study_areas_id; update main.study_areas set geom_traj_buffer = locations_24h_studyareas.geom from temp.locations_24h_studyareas where defined_boundaries = 0 and study_areas.study_areas_id = locations_24h_studyareas.study_areas_id;

geom_grid300	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, trajectories (1 location every 12 hours) are intersected with a grid of 250 meters (modis grid). Only cells with a minimum of time spent on it are kept. A final buffer of 1 km is added. These boundaries should be updated whenever a new set of locations is uploaded in the database.
geom_kernel95_5km_buffer	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the kernel home range is calculated using all the data of a study area (1 location every 12 hours) + a buffer of 5 km. These boundaries should be updated whenever a new set of locations is uploaded in the database.
geom_vhf	USER-DEFINED	Multi polygons layer of study areas for vhf locations. This spatial layer can be used as a reference to locate the study areas. Study areas can have defined boundaries (e.g. fenced). In this case, the field "defined_bundaries" is set to 1. Otherwise a reference boundary is created as the convex hull polygon (plus a buffer of 1 km) of the existing locations. These boundaries should be updated whenever a new set of locations is uploaded in the database.

study_areas:	Study areas are the areas monitored by research groups. Each study area can have many animals. Study areas can have defined, approximated, or no specific spatial boundaries.

Attribute	Type	Description
study_areas_id	integer	Eurodeer identifier for study areas.
study_name	character varying	Name of the study area.
geom	USER-DEFINED	Multi polygons layer of study areas. This spatial layer can be used as a reference to locate the study areas. Study areas can have defined boundaries (e.g. fenced). In this case, the field "defined_bundaries" is set to 1. Otherwise a reference boundary is created as the convex hull polygon (plus a buffer of 1 km) of the existing locations. These boundaries should be updated whenever a new set of locations is uploaded in the database.
research_groups_id	integer	Indentifier of the research group that is monitoring animals in this study area.
defined_boundaries	integer	If the study area boundaries are defined by research groups (e.g. areas fenced, or know area of animals' movement), this filed is set to 1. There polygons are not modified if new locations are uploaded. For areas with non defined boundaries (tag = 0), the boundaries are calculated on the convex hull polygon of the existing locations, and thus are updated whenever new locations are uploaded.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

short_name	character varying	Short version of the study area name for maps and short reports.
geom_mcp_individuals	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the MCP of each individual is calulated, then al mcp are merged and 500 meters buffer is added. These boundaries should be updated whenever a new set of locations is uploaded in the database. the code to update: update main.study_areas set geom_mcp_individuals = st_multi(foo.qq) from (select studies_id as ww,st_buffer(((st_multi(st_union (geom))))::geometry(multipolygon, 4326)::geography, 500)::geometry qq from analysis.view_convexhull group by studies_id) as foo where defined_boundaries = 0 and study_areas.study_areas_id = foo.ww;
geom_traj_buffer	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the buffer of 1 km (on both sides) is calculated from the trajectories of all the animals (1 location every 24 hours is considered). Th polygons of the same study area are then merged to generate a multipolygon for each study area). These boundaries should be updated whenever a new set of locations is uploaded in the database. DROP TABLE IF EXISTS temp.locations_12h_traj; CREATE TABLE temp.locations_24h_traj AS SELECT animals_id, study_areas_id , foo2.geom::geometry(LineString,4326) AS geom FROM (SELECT foo.animals_id,study_areas_id, st_makeline(foo.geom) AS geom FROM (SELECT study_areas_id, geom, animals_id, acquisition_time FROM temp.locations_24h ORDER BY study_areas_id, animals_id, acquisition_time) foo GROUP BY foo.animals_id, study_areas_id) foo2 WHERE st_geometrytype (foo2.geom) = 'ST_LineString'::text; ALTER TABLE temp.locations_24h_traj ADD COLUMN geom_buffer geometry (polygon,4326); UPDATE temp.locations_24h_traj set geom_buffer = (st_buffer(st_simplify(geom, 0.001)::geography, 1000))::geometry DROP TABLE IF EXISTS temp.locations_24h_studyareas; CREATE TABLE temp.locations_24h_studyareas AS SELECT study_areas_id, st_multi(st_union(geom_buffer))::geometry(multipolygon, 4326) geom from temp.locations_24h_traj group by study_areas_id; update main.study_areas set geom_traj_buffer = locations_24h_studyareas.geom from temp.locations_24h_studyareas where defined_boundaries = 0 and study_areas.study_areas_id = locations_24h_studyareas.study_areas_id;
geom_grid300	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, trajectories (1 location every 12 hours) are intersected with a grid of 250 meters (modis grid). Only cells with a minimum of time spent on it are kept. A final buffer of 1 km is added. These boundaries should be updated whenever a new set of locations is uploaded in the database.
geom_kernel95_5km_buffer	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the kernel home range is calculated using all the data of a study area (1 location every 12 hours) + a buffer of 5 km. These boundaries should be updated whenever a new set of locations is uploaded in the database.

geom_vhf	USER-DEFINED	Multi polygons layer of study areas for vhf locations. This spatial layer can be used as a reference to locate the study areas. Study areas can have defined boundaries (e.g. fenced). In this case, the field "defined_bundaries" is set to 1. Otherwise a reference boundary is created as the convex hull polygon (plus a buffer of 1 km) of the existing locations. These boundaries should be updated whenever a new set of locations is uploaded in the database.

vhf_data_animals:	Table with VHF locations data (red deer), inluding a list of derived ancillary infomation calculated using the information on coordinates and acquisition time, and intersecting with environmental layers.

Attribute	Type	Description
vhf_data_animals_id	integer	Eurodeer identifier for the VHF location
animals_id	integer	Eurodeer identifier for the animal
vhf_sensors_id	integer	Eurodeer identifier for the VHF sensor
geom	USER-DEFINED	Geometry of the location (point)
acquisition_time	timestamp with time zone	Date and time of acquisition of the VHF coordinates (with time zone)
x_original_reference	double precision	Coordinate X as originally recorded (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as originally recorded (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates used in the orignal coordinate recording
latitude	double precision	Latitude of the VHF location
longitude	double precision	Longitude of the VHF location
vhf_validity_code	integer	This field tags the record according to its source ir validity, which is a measure of the degree of reliability (explanation of codes in lu_tables.lu_vhf_validity)
notes	character varying	General comments about the specific location
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster	integer	Degrees calculated counterclockwise from east (source: ASTER project)
altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)

aspect_srtm	integer	Degrees calculated counterclockwise from east (source: SRTM project)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
corine_land_cover_2006_code	integer	Code of the Corine lad cover class (in env_data.corine_land_cover_legend there is a complete description of these codes)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time)

vhf_data_animals:	Table with VHF locations data, inluding a list of derived ancillary infomation calculated using the information on coordinates and acquisition time, and intersecting with environmental layers.

Attribute	Type	Description
vhf_data_animals_id	integer	Eurodeer identifier for the VHF location
animals_id	integer	Eurodeer identifier for the animal
vhf_sensors_id	integer	Eurodeer identifier for the VHF sensor
geom	USER-DEFINED	Geometry of the location (point)
acquisition_time	timestamp with time zone	Date and time of acquisition of the VHF coordinates (with time zone)
x_original_reference	double precision	Coordinate X as originally recorded (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as originally recorded (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates used in the orignal coordinate recording
latitude	double precision	Latitude of the VHF location
longitude	double precision	Longitude of the VHF location

vhf_validity_code	integer	This field tags the record according to its source ir validity, which is a measure of the degree of reliability (explanation of codes in lu_tables.lu_vhf_validity)
notes	character varying	General comments about the specific location
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster	integer	Degrees calculated counterclockwise from east (source: ASTER project)
altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm	integer	Degrees calculated counterclockwise from east (source: SRTM project)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
corine_land_cover_2006_code	integer	Code of the Corine lad cover class (in env_data.corine_land_cover_legend there is a complete description of these codes)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time)

vhf_sensors: Catalogue of VHF sensors. Each sensor belongs to a research group. The attributes include the brand and the model. The id used in the original data set is also included.

Attribute	Type	Description
vhf_sensors_id	integer	Eurodeer identifier for VHF sensors.
research_groups_id	integer	Id of the research group that owns the VHF sensor.
vhf_sensors_original_id	character varying	Identifier of the VHF sensor in the original data set.
vendor	character varying	Company that produced the sensor.
model	character varying	Model of the VHF sensor.

insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

vhf_sensors:	Catalogue of VHF sensors (red deer). Each sensor belongs to a research group. The attributes include the brand and the model. The id used in the original data set is also included.

Attribute	Type	Description
vhf_sensors_id	integer	Eurodeer identifier for VHF sensors.
research_groups_id	integer	Id of the research group that owns the VHF sensor.
vhf_sensors_original_id	character varying	Identifier of the VHF sensor in the original data set.
vendor	character varying	Company that produced the sensor.
model	character varying	Model of the VHF sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

vhf_sensors_animals:	None

Attribute	Type	Description
vhf_sensors_animals_id	integer	Eurodeer identifier of the deployment.
vhf_sensors_id	integer	Eurodeer identifier of the vhf sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.
end_time	timestamp with time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
end_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality

insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

vhf_sensors_animals:	Table with the information on the deployments of VHF sensors on animals (red deer) (starting and ending date and time of the deployment).

Attribute	Type	Description
vhf_sensors_animals_id	integer	Eurodeer identifier of the deployment.
vhf_sensors_id	integer	Eurodeer identifier of the vhf sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.
end_time	timestamp with time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
end_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

vhf_sensors_animals:	Table with the information on the deployments of VHF sensors on animals (starting and ending date and time of the deployment).

Attribute	Type	Description
vhf_sensors_animals_id	integer	Eurodeer identifier of the deployment.
vhf_sensors_id	integer	Eurodeer identifier of the vhf sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp with time zone	Time and date of the start of the deployment.

end_time	timestamp with time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
end_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
insert_user	character varying	User who created the record.
update_user	character varying	User who modified the record (last time).

view_eurodeer_gps_positions:	Animal locations with valid coordinates and information on animals (study area, research group, age and sex)

Attribute	Type	Description
animals_id	integer	None
study_areas_id	integer	None
research_groups_id	integer	None
animals_original_name	character varying	None
sex	"char"	None
age	integer	None
gps_data_animals_id	integer	None
gps_sensors_id	integer	None
acquisition_time	timestamp with time zone	None
x_original_reference	double precision	None
y_original_reference	double precision	None
srid_original_reference	integer	None
latitude	double precision	None
longitude	double precision	None
altitude_gps	integer	None
dop	double precision	None
sats	integer	None
temperature_sensor	double precision	None

geom	USER-DEFINED	None
gps_validity_code	smallint	None
corine_land_cover_2006_code	integer	None
ndvi_modis	real	None
ndvi_vegetation	real	None
snow_modis	integer	None
sun_angle	double precision	None
utm_srid	integer	None
utm_x	integer	None
utm_y	integer	None
altitude_srtm	integer	None
slope_srtm	double precision	None
aspect_srtm	integer	None
altitude_aster	integer	None
slope_aster	double precision	None
aspect_aster	integer	None

view_locations_set:

View that stores the core information of locations data (id of the animal, the acquisition time and the geometry). Non valid records are represented without the geometry. Duplicated timestamps are excluded.

Attribute	Type	Description
animals_id	integer	None
acquisition_time	timestamp with time zone	None
geom	USER-DEFINED	None

view_mortality:

Animals and study areas for which mortality data are currently avaialable

Attribute	Type	Description
animals_id	integer	None
mortality_code	integer	None
contact_timestamp	timestamp with time zone	None
animals_contacts_id	integer	None
notes	character varying	None
status	character varying	None
year_birth	integer	None
monitored_gps	integer	None

monitored_vhf	integer	None
populations_id	integer	None
reintroduction	integer	None
study_areas_id	integer	None
insert_timestamp	timestamp with time zone	None
update_timestamp	timestamp with time zone	None
year_birth_exact	boolean	None
first_capture_date	date	None
monitored_activity	integer	None
animals_original_id	character varying	None
animals_original_name	character varying	None
age_class_code_capture	integer	None
sex	"char"	None

main_reddeer schema

The schema "main_reddeer" is the place where all the core information of the main objects about red deer are stored: data from sensors (at the moment, GPS, VHF, activity), sensors, animals, studies. It has the same structure as "main", where data for roe deer are stored. Information on research groups is shared between roe deer and red deer and is located in "main".

animals: Table with the information on the animals

Attribute	Type	Description
animals_id	integer	Database id of population. Each animal belongs to a population, which is part of a study area. The same study area can have multiple populations. Linked with the table main.populations.
study_areas_id	integer	Study area where the animal is located (reference to table main.study_area).
animals_original_id	character varying	Identifier of the animal in the original data set.
animals_original_name	character varying	Nome of the animal in the original data set.
sex	"char"	Code for sex. It can be either "f" (female) or "m" (male). When the sex is not known, the field can be left empty.
first_capture_date	date	Date when the animal was captured the first time. This information is important to interpret the age_class.

monitored_activity	integer	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_gps	integer	If the animals has GPS data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_vhf	integer	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
age_class_code_capture	integer	Code of the age class (reference to table lu_tables.lu_age_class) at the first capture of the animal (this information must be provided to give full meaning to the age class, as it is a dynamic attribute that changes over time).
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
estimated_birth_year	integer	If 1, the animal has been reintroduced, if 0, the animal is not reintroduced.
estimated_minimum_birth_year	integer	None
reintroduction	integer	Year of birth (when known). In the year_birth_exact field it is described if this is the exact year of birth of just an extimation (minimum year of birth).

animals:	Table with the information on the animals that are part of Eurodeer project. Each animal can belong to a study area, which is related to a research group. Information included in the table are sex, age, name and id in the original data set. It also stores data on which kind of sensors were applied to animals.

Attribute	Type	Description

animals_id	integer	Database id of population. Each animal belongs to a population, which is part of a study area. The same study area can have multiple populations. Linked with the table main.populations.
study_areas_id	integer	Study area where the animal is located (reference to table main.study_area).
animals_original_id	character varying	Identifier of the animal in the original data set.
animals_original_name	character varying	Nome of the animal in the original data set.
sex	"char"	Code for sex. It can be either "f" (female) or "m" (male). When the sex is not known, the field can be left empty.
first_capture_date	date	Date when the animal was captured the first time. This information is important to interpret the age_class.
monitored_activity	integer	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_gps	integer	If the animals has GPS data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_vhf	integer	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
age_class_code_capture	integer	Code of the age class (reference to table lu_tables.lu_age_class) at the first capture of the animal (this information must be provided to give full meaning to the age class, as it is a dynamic attribute that changes over time).
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
estimated_birth_year	integer	If 1, the animal has been reintroduced, if 0, the animal is not reintroduced.

estimated_minimum_birth_year	integer	None
reintroduction	integer	Year of birth (when known). In the year_birth_exact field it is described if this is the exact year of birth of just an extimation (minimum year of birth).

animals: Table with the information on the animals (red deer) that are part of Eurodeer project. Each animal can belong to a study area, which is related to a research group. Information included in the table are sex, age, name and id in the original data set. It also stores data on which kind of sensors were applied to animals.

Attribute	Type	Description
animals_id	integer	Database id of population. Each animal belongs to a population, which is part of a study area. The same study area can have multiple populations. Linked with the table main.populations.
study_areas_id	integer	Study area where the animal is located (reference to table main.study_area).
animals_original_id	character varying	Identifier of the animal in the original data set.
animals_original_name	character varying	Nome of the animal in the original data set.
sex	"char"	Code for sex. It can be either "f" (female) or "m" (male). When the sex is not known, the field can be left empty.
first_capture_date	date	Date when the animal was captured the first time. This information is important to interpret the age_class.
monitored_activity	integer	If the animals has activity data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_gps	integer	If the animals has GPS data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
monitored_vhf	integer	If the animals has vhf data associated, then this filed is tagged as "1". If no data are associated, it is tagged with "0". This value is updated by the function tools.eurodeer_monitored_animals that (at the moment) is not associated by any automatic procedure, so it has to be run on users' request.
age_class_code_capture	integer	Code of the age class (reference to table lu_tables.lu_age_class) at the first capture of the animal (this information must be provided to give full meaning to the age class, as it is a dynamic attribute that changes over time).
notes	character varying	Open field where general notes on the animals can be added.

notes	character varying	Open field where general notes on the animals can be added.
notes	character varying	Flag (yes/no) that specifies if the year of birth is exact (yes) or just an estimation (no) (e.g. when I know that the animal is at least 4 years old but I do not know the exact age.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
estimated_birth_year	integer	If 1, the animal has been reintroduced, if 0, the animal is not reintroduced.
estimated_minimum_birth_year	integer	None
reintroduction	integer	Year of birth (when known). In the year_birth_exact field it is described if this is the exact year of birth of just an extimation (minimum year of birth).

gps_data_animals:	Table with GPS locations data associated to animals and with a list of derived ancillary infomation calculated using the information on coordinates and acquisition time, and intersecting with environmental layers.

Attribute	Type	Description
gps_data_animals_id	integer	Eurodeer identifier for the location
animals_id	integer	Eurodeer identifier for the animal
gps_sensors_id	integer	Eurodeer identifier for the GPS sensor
acquisition_time	timestamp with time zone	Date and time of acquisition of the GPS coordinates (with time zone)
x_original_reference	double precision	Coordinate X as computed by the software connected to the GPS sensor (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as computed by the software connected to the GPS sensor (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates provided by the software connected to the GPS sensor
latitude	double precision	Latitude recorded by the GPS sensor
longitude	double precision	Longitude recorded by the GPS sensor
altitude_gps	integer	Altitude recorded by the GPS sensor (related to the centre of the earth)
dop	double precision	Dilution Of Precision
sats	integer	Number of satellites used by the GPS sensor to calculate the coordinates

temperature_sensor	double precision	Temperature as measured by the sensor associated to the GPS
geom	USER-DEFINED	Geometry of the location (point)
gps_validity_code	smallint	This field tags the record according to its "validity" or degree of reliability (explanation of codes in lu_tables.lu_gps_validity)
corine_land_cover_2006_code	integer	Code of the Corine lad cover class produced in 2006 (in env_data.corine_land_cover_legend there is a complete description of these codes)
ndvi_modis	real	NDVI derived from MODIS (16-daily, non smoothed, assciated to the closest [in time] image). For analysis, it is recommended to use smoothed data (smoothed or boku).
ndvi_vegetation	real	NDVI derived from SPOT VEGETATION. SPOT Vegetation sensor does not record any information sine end of 2014.
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat) (source: SRTM project)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat)(source: ASTER project)
corine_land_cover_2000_code	integer	Code of the Corine lad cover class produced in 2000 (in env_data.corine_land_cover_legend there is a complete description of these codes)
corine_land_cover_1990_code	integer	Code of the Corine lad cover class produced in 1990 (in env_data.corine_land_cover_legend there is a complete description of these codes)
fapar_vegetation	real	FAPAR derived from SPOT VEGETATION.SPOT Vegetation sensor does not record any information sine end of 2014.

ndvi_modis_boku	double precision	None
ndvi_modis_smoothed	double precision	None
update_core_timestamp	timestamp with time zone	User who modified the core elements of record (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
update_core_user	character varying	User who modified the record (last time).
update_user	character varying	User who created the record.
insert_user	character varying	Date and time when the core elements of record was updated (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
aspect_srtm_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat) (source: SRTM project)
aspect_aster_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat)(source: ASTER project)

gps_data_animals: None

Attribute	Type	Description
gps_data_animals_id	integer	Eurodeer identifier for the location
animals_id	integer	Eurodeer identifier for the animal
gps_sensors_id	integer	Eurodeer identifier for the GPS sensor
acquisition_time	timestamp with time zone	Date and time of acquisition of the GPS coordinates (with time zone)
x_original_reference	double precision	Coordinate X as computed by the software connected to the GPS sensor (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as computed by the software connected to the GPS sensor (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates provided by the software connected to the GPS sensor
latitude	double precision	Latitude recorded by the GPS sensor
longitude	double precision	Longitude recorded by the GPS sensor
altitude_gps	integer	Altitude recorded by the GPS sensor (related to the centre of the earth)
dop	double precision	Dilution Of Precision
sats	integer	Number of satellites used by the GPS sensor to calculate the coordinates

temperature_sensor	double precision	Temperature as measured by the sensor associated to the GPS
geom	USER-DEFINED	Geometry of the location (point)
gps_validity_code	smallint	This field tags the record according to its "validity" or degree of reliability (explanation of codes in lu_tables.lu_gps_validity)
corine_land_cover_2006_code	integer	Code of the Corine lad cover class produced in 2006 (in env_data.corine_land_cover_legend there is a complete description of these codes)
ndvi_modis	real	NDVI derived from MODIS (16-daily, non smoothed, assciated to the closest [in time] image). For analysis, it is recommended to use smoothed data (smoothed or boku).
ndvi_vegetation	real	NDVI derived from SPOT VEGETATION. SPOT Vegetation sensor does not record any information sine end of 2014.
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat) (source: SRTM project)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat)(source: ASTER project)
corine_land_cover_2000_code	integer	Code of the Corine lad cover class produced in 2000 (in env_data.corine_land_cover_legend there is a complete description of these codes)
corine_land_cover_1990_code	integer	Code of the Corine lad cover class produced in 1990 (in env_data.corine_land_cover_legend there is a complete description of these codes)
fapar_vegetation	real	FAPAR derived from SPOT VEGETATION.SPOT Vegetation sensor does not record any information sine end of 2014.

ndvi_modis_boku	double precision	None
ndvi_modis_smoothed	double precision	None
update_core_timestamp	timestamp with time zone	User who modified the core elements of record (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
update_core_user	character varying	User who modified the record (last time).
update_user	character varying	User who created the record.
insert_user	character varying	Date and time when the core elements of record was updated (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
aspect_srtm_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat) (source: SRTM project)
aspect_aster_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat)(source: ASTER project)

gps_data_animals:	Table with GPS locations data associated to animals (red deer) with a list of derived ancillary infomation calculated using the information on coordinates and acquisition time, and intersecting with environmental layers.

Attribute	Type	Description
gps_data_animals_id	integer	Eurodeer identifier for the location
animals_id	integer	Eurodeer identifier for the animal
gps_sensors_id	integer	Eurodeer identifier for the GPS sensor
acquisition_time	timestamp with time zone	Date and time of acquisition of the GPS coordinates (with time zone)
x_original_reference	double precision	Coordinate X as computed by the software connected to the GPS sensor (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as computed by the software connected to the GPS sensor (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates provided by the software connected to the GPS sensor
latitude	double precision	Latitude recorded by the GPS sensor
longitude	double precision	Longitude recorded by the GPS sensor
altitude_gps	integer	Altitude recorded by the GPS sensor (related to the centre of the earth)
dop	double precision	Dilution Of Precision

sats	integer	Number of satellites used by the GPS sensor to calculate the coordinates
temperature_sensor	double precision	Temperature as measured by the sensor associated to the GPS
geom	USER-DEFINED	Geometry of the location (point)
gps_validity_code	smallint	This field tags the record according to its "validity" or degree of reliability (explanation of codes in lu_tables.lu_gps_validity)
corine_land_cover_2006_code	integer	Code of the Corine lad cover class produced in 2006 (in env_data.corine_land_cover_legend there is a complete description of these codes)
ndvi_modis	real	NDVI derived from MODIS (16-daily, non smoothed, assciated to the closest [in time] image). For analysis, it is recommended to use smoothed data (smoothed or boku).
ndvi_vegetation	real	NDVI derived from SPOT VEGETATION. SPOT Vegetation sensor does not record any information sine end of 2014.
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat) (source: SRTM project)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster_east_ccw	integer	Degrees calculated counterclockwise from east, -1 means no aspect (flat)(source: ASTER project)
corine_land_cover_2000_code	integer	Code of the Corine lad cover class produced in 2000 (in env_data.corine_land_cover_legend there is a complete description of these codes)
corine_land_cover_1990_code	integer	Code of the Corine lad cover class produced in 1990 (in env_data.corine_land_cover_legend there is a complete description of these codes)

fapar_vegetation	real	FAPAR derived from SPOT VEGETATION.SPOT Vegetation sensor does not record any information sine end of 2014.
ndvi_modis_boku	double precision	None
ndvi_modis_smoothed	double precision	None
update_core_timestamp	timestamp with time zone	User who modified the core elements of record (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
update_core_user	character varying	User who modified the record (last time).
update_user	character varying	User who created the record.
insert_user	character varying	Date and time when the core elements of record was updated (last time): animals_id, geometry, latitude, longitude, timestamp, validity code.
aspect_srtm_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat) (source: SRTM project)
aspect_aster_north_cw	integer	Degrees calculated clockwise from north, -1 means no aspect (flat)(source: ASTER project)

gps_sensors:	Catalogue of GPS sensors (red deer). Each sensor belongs to a research group. The attributes include the brand and the model. The id used in the original data set is also included.

Attribute	Type	Description
gps_sensors_id	integer	Eurodeer identifier for GPS sensors.
research_groups_id	integer	Id of the research group that owns the GPS sensor.
gps_sensors_original_id	character varying	Identifier of the GPS sensor in the original data set.
vendor	character	Company that produced the sensor.
model	character	Model of the GPS sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

gps_sensors:	None

Attribute	Type	Description
gps_sensors_id	integer	Eurodeer identifier for GPS sensors.
research_groups_id	integer	Id of the research group that owns the GPS sensor.
gps_sensors_original_id	character varying	Identifier of the GPS sensor in the original data set.

vendor	character	Company that produced the sensor.
model	character	Model of the GPS sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

gps_sensors: Catalogue of GPS sensors. Each sensor belongs to a research group. The attributes include the brand and the model. The id used in the original data set is also included.

Attribute	Type	Description
gps_sensors_id	integer	Eurodeer identifier for GPS sensors.
research_groups_id	integer	Id of the research group that owns the GPS sensor.
gps_sensors_original_id	character varying	Identifier of the GPS sensor in the original data set.
vendor	character	Company that produced the sensor.
model	character	Model of the GPS sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

gps_sensors_animals: None

Attribute	Type	Description
gps_sensors_animals_id	integer	Eurodeer identifier of the deployment.
gps_sensors_id	integer	Eurodeer identifier of the GPS sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp without time zone	Time and date of the start of the deployment.
end_time	timestamp without time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.

mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
end_gps_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).

gps_sensors_animals:	Table with the information on the deployments of GPS sensors on animals (starting and ending date and time of the deployment).

Attribute	Type	Description
gps_sensors_animals_id	integer	Eurodeer identifier of the deployment.
gps_sensors_id	integer	Eurodeer identifier of the GPS sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp without time zone	Time and date of the start of the deployment.
end_time	timestamp without time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
end_gps_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).

gps_sensors_animals:	Table with the information on the deployments of GPS sensors on animals (red deer) (starting and ending date and time of the deployment).

Attribute	Type	Description
gps_sensors_animals_id	integer	Eurodeer identifier of the deployment.
gps_sensors_id	integer	Eurodeer identifier of the GPS sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp without time zone	Time and date of the start of the deployment.

end_time	timestamp without time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
end_gps_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).

gps_sensors_animals: None

Attribute	Type	Description
gps_sensors_animals_id	integer	Eurodeer identifier of the deployment.
gps_sensors_id	integer	Eurodeer identifier of the GPS sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp without time zone	Time and date of the start of the deployment.
end_time	timestamp without time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.
mortality_code	integer	If the reason of the end of deployment is dead, this field specifies the reason of the death (code described in the look up table lu_tables.lu_mortality
end_gps_monitoring_code	integer	Code for the reason of the end of monitoring (reference to the look up table lu_tables.lu_end_monitoring).

study_areas: Study areas are the areas monitored by research groups. Each study area can have many animals. Study areas can have defined, approximated, or no specific spatial boundaries.

Attribute	Type	Description

study_areas_id	integer	Eurodeer identifier for study areas.
study_name	character varying	Name of the study area.
geom	USER-DEFINED	Multi polygons layer of study areas. This spatial layer can be used as a reference to locate the study areas. Study areas can have defined boundaries (e.g. fenced). In this case, the field "defined_bundaries" is set to 1. Otherwise a reference boundary is created as the convex hull polygon (plus a buffer of 1 km) of the existing locations. These boundaries should be updated whenever a new set of locations is uploaded in the database.
research_groups_id	integer	Indentifier of the research group that is monitoring animals in this study area.
defined_boundaries	integer	If the study area boundaries are defined by research groups (e.g. areas fenced, or know area of animals' movement), this filed is set to 1. There polygons are not modified if new locations are uploaded. For areas with non defined boundaries (tag = 0), the boundaries are calculated on the convex hull polygon of the existing locations, and thus are updated whenever new locations are uploaded.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
short_name	character varying	Short version of the study area name for maps and short reports.
geom_mcp_individuals	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the MCP of each individual is calulated, then al mcp are merged and 500 meters buffer is added. These boundaries should be updated whenever a new set of locations is uploaded in the database. the code to update: update main.study_areas set geom_mcp_individuals = st_multi(foo.qq) from (select studies_id as ww,st_buffer(((st_multi(st_union(geom)))):geometry(multipolygon, 4326)::geography, 500)::geometry qq from analysis.view_convexhull group by studies_id) as foo where defined_boundaries = 0 and study_areas.study_areas_id = foo.ww;
geom_mcp_individuals	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the MCP of each individual is calulated, then al mcp are merged and 500 meters buffer is added. These boundaries should be updated whenever a new set of locations is uploaded in the database. the code to update: update main_reddeer.study_areas set geom_mcp_individuals = st_multi(foo.qq) from (select studies_id as ww,st_buffer(((st_multi(st_union(geom)))):geometry(multipolygon, 4326)::geography, 500)::geometry qq from analysis.view_convexhull_reddeer group by studies_id) as foo where defined_boundaries = 0 and study_areas.study_areas_id = foo.ww;

study_areas:	Study areas are the areas monitored by research groups. Each study area can have many animals (red deer). Study areas can have defined, approximated, or no specific spatial boundaries.

Attribute	Type	Description
study_areas_id	integer	Eurodeer identifier for study areas.
study_name	character varying	Name of the study area.
geom	USER-DEFINED	Multi polygons layer of study areas. This spatial layer can be used as a reference to locate the study areas. Study areas can have defined boundaries (e.g. fenced). In this case, the field "defined_bundaries" is set to 1. Otherwise a reference boundary is created as the convex hull polygon (plus a buffer of 1 km) of the existing locations. These boundaries should be updated whenever a new set of locations is uploaded in the database.
research_groups_id	integer	Indentifier of the research group that is monitoring animals in this study area.
defined_boundaries	integer	If the study area boundaries are defined by research groups (e.g. areas fenced, or know area of animals' movement), this filed is set to 1. There polygons are not modified if new locations are uploaded. For areas with non defined boundaries (tag = 0), the boundaries are calculated on the convex hull polygon of the existing locations, and thus are updated whenever new locations are uploaded.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
short_name	character varying	Short version of the study area name for maps and short reports.
geom_mcp_individuals	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the MCP of each individual is calulated, then al mcp are merged and 500 meters buffer is added. These boundaries should be updated whenever a new set of locations is uploaded in the database. the code to update: update main.study_areas set geom_mcp_individuals = st_multi(foo.qq) from (select studies_id as ww,st_buffer(((st_multi(st_union(geom)))::geometry(multipolygon, 4326)::geography, 500)::geometry qq from analysis.view_convexhull group by studies_id) as foo where defined_boundaries = 0 and study_areas.study_areas_id = foo.ww;
geom_mcp_individuals	USER-DEFINED	Multi polygons layer with an alternative representation of study areas. In this case, the MCP of each individual is calulated, then al mcp are merged and 500 meters buffer is added. These boundaries should be updated whenever a new set of locations is uploaded in the database. the code to update: update main_reddeer.study_areas set geom_mcp_individuals = st_multi(foo.qq) from (select studies_id as ww,st_buffer(((st_multi(st_union(geom)))::geometry(multipolygon, 4326)::geography, 500)::geometry qq from analysis.view_convexhull_reddeer group by studies_id) as foo where defined_boundaries = 0 and study_areas.study_areas_id = foo.ww;

vhf_data_animals:	Table with VHF locations data (red deer), inluding a list of derived ancillary infomation calculated using the information on coordinates and acquisition time, and intersecting with environmental layers.

Attribute	Type	Description
vhf_data_animals_id	integer	Eurodeer identifier for the VHF location
animals_id	integer	Eurodeer identifier for the animal
vhf_sensors_id	integer	Eurodeer identifier for the VHF sensor
geom	USER-DEFINED	Geometry of the location (point)
acquisition_time	timestamp with time zone	Date and time of acquisition of the VHF coordinates (with time zone)
x_original_reference	double precision	Coordinate X as originally recorded (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as originally recorded (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates used in the orignal coordinate recording
latitude	double precision	Latitude of the VHF location
longitude	double precision	Longitude of the VHF location
vhf_validity_code	integer	This field tags the record according to its source ir validity, which is a measure of the degree of reliability (explanation of codes in lu_tables.lu_vhf_validity)
notes	character varying	General comments about the specific location
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster	integer	Degrees calculated counterclockwise from east (source: ASTER project)
altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm	integer	Degrees calculated counterclockwise from east (source: SRTM project)
sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
corine_land_cover_2006_code	integer	Code of the Corine lad cover class (in env_data.corine_land_cover_legend there is a complete description of these codes)

insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time)

vhf_data_animals:	Table with VHF locations data, inluding a list of derived ancillary infomation calculated using the information on coordinates and acquisition time, and intersecting with environmental layers.

Attribute	Type	Description
vhf_data_animals_id	integer	Eurodeer identifier for the VHF location
animals_id	integer	Eurodeer identifier for the animal
vhf_sensors_id	integer	Eurodeer identifier for the VHF sensor
geom	USER-DEFINED	Geometry of the location (point)
acquisition_time	timestamp with time zone	Date and time of acquisition of the VHF coordinates (with time zone)
x_original_reference	double precision	Coordinate X as originally recorded (in the srid_original_reference)
y_original_reference	double precision	Coordinate Y as originally recorded (in the srid_original_reference)
srid_original_reference	integer	Reference system of the projected coordinates used in the orignal coordinate recording
latitude	double precision	Latitude of the VHF location
longitude	double precision	Longitude of the VHF location
vhf_validity_code	integer	This field tags the record according to its source ir validity, which is a measure of the degree of reliability (explanation of codes in lu_tables.lu_vhf_validity)
notes	character varying	General comments about the specific location
altitude_aster	integer	Meters above sea level (from ASTER project)
slope_aster	double precision	Degrees (from ASTER project)
aspect_aster	integer	Degrees calculated counterclockwise from east (source: ASTER project)
altitude_srtm	integer	Meters above sea level (from SRTM project)
slope_srtm	double precision	Degrees (from SRTM project)
aspect_srtm	integer	Degrees calculated counterclockwise from east (source: SRTM project)

sun_angle	double precision	Sun angle above (or below) the horizon (in degrees) as computed by the function tools.sun_elevation_angle_function
snow_modis	integer	Snow coverage (25:land; 50:cloud; 200:snow)
utm_y	integer	Y coordinate projected in the utm_srid UTM zone
utm_x	integer	X coordinate projected in the utm_srid UTM zone
utm_srid	integer	SRID code of the UTM zone of the centroid of the locations for the animal
corine_land_cover_2006_code	integer	Code of the Corine lad cover class (in env_data.corine_land_cover_legend there is a complete description of these codes)
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time)

vhf_sensors:	Catalogue of VHF sensors. Each sensor belongs to a research group. The attributes include the brand and the model. The id used in the original data set is also included.

Attribute	Type	Description
vhf_sensors_id	integer	Eurodeer identifier for VHF sensors.
research_groups_id	integer	Id of the research group that owns the VHF sensor.
vhf_sensors_original_id	character varying	Identifier of the VHF sensor in the original data set.
vendor	character	Company that produced the sensor.
model	character	Model of the VHF sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

vhf_sensors:	Catalogue of VHF sensors (red deer). Each sensor belongs to a research group. The attributes include the brand and the model. The id used in the original data set is also included.

Attribute	Type	Description
vhf_sensors_id	integer	Eurodeer identifier for VHF sensors.
research_groups_id	integer	Id of the research group that owns the VHF sensor.
vhf_sensors_original_id	character varying	Identifier of the VHF sensor in the original data set.
vendor	character	Company that produced the sensor.

model	character	Model of the VHF sensor.
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).

vhf_sensors_animals:	Table with the information on the deployments of VHF sensors on animals (starting and ending date and time of the deployment).

Attribute	Type	Description
vhf_sensors_animals_id	integer	Eurodeer identifier of the deployment.
vhf_sensors_id	integer	Eurodeer identifier of the vhf sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp without time zone	Time and date of the start of the deployment.
end_time	timestamp without time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.

vhf_sensors_animals:	None

Attribute	Type	Description
vhf_sensors_animals_id	integer	Eurodeer identifier of the deployment.
vhf_sensors_id	integer	Eurodeer identifier of the vhf sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp without time zone	Time and date of the start of the deployment.
end_time	timestamp without time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.

vhf_sensors_animals:	Table with the information on the deployments of VHF sensors on animals (red deer) (starting and ending date and time of the deployment).

Attribute	Type	Description
vhf_sensors_animals_id	integer	Eurodeer identifier of the deployment.
vhf_sensors_id	integer	Eurodeer identifier of the vhf sensor.
animals_id	integer	Eurodeer identifier of the animal.
start_time	timestamp without time zone	Time and date of the start of the deployment.
end_time	timestamp without time zone	Time and date of the end of the deployment.
update_timestamp	timestamp with time zone	Date and time when the record was updated (last time).
insert_timestamp	timestamp with time zone	Date and time when the record was uploaded into the database.
notes	character varying	Open field where general notes on the deployment can be added.

public schema

standard public schema

temp schema

This schema stores temporary objects (tables, functions, ...) used for analysis or for testing purposes. Elements stored in this schema can be deleted at any time by the database administrator.

tools schema

The schema "tools" hosts all the functions and tools that are used throughout the database to manage, massage, analyse and query data.

log_dbchanges:	Table that reports (and keeps track of) all the changes made to the database (import of new data, update of existing data, change in the data structure, creation of a tool, etc with reference to the data curator that did the change and the date. At the moment, a single table is used for both eurodeer and eureddeer.

Attribute	Type	Description
log_dbchanges_id	integer	Database id of db changes.
date_change	date	Date when the change was made.
data_curators_code	integer	Code of the data curator that did the change.

action_code	integer	Code for the type of change that was made to the database.
change_description	text	Description of the change.

ws_fem schema

Working schema for members of FEM research group. This can be used to create any object they need for their studies. No other users will have access to this schema.

ws_isc schema

Working schema for members of ISC research group. This can be used to create any object they need for their studies. No other users will have access to this schema.

ws_zurich schema

Working schema for members of Zurich research group. This can be used to create any object they need for their studies. No other users will have access to this schema.