The Babase Pocket Reference Guide

A Technical Specification Summary

Karl O. Pinc, PhD. Jeanne Altmann, PhD. Susan C. Alberts, Jacob B. Gordon, and Leah Gerber

This material is based upon work supported by the National Science Foundation under Grant Nos. 0323553 and 0323596.

COLLABORATORS			
	TITLE : The Babase Pocket Reference Guide		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Karl O. Pinc, PhD. Jeanne Altmann, PhD. Susan C. Alberts, Jacob B. Gordon, and Leah Gerber	July 2, 2025	
ER Diagram layout and conversion to Dia	Leah Gerber	July 2, 2025	
DocBook formatting	Anne Hubbard	July 2, 2025	
DocBook formatting	Karl Pinc	July 2, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

List of Figures

List of Tables

1 Babase Summarized

Warning
Tables which have names ending in "_DATA" should not be used, there is always a view of the data in these tables that may be used in their place. Tables ending in "_DATA" may change in future Babase minor releases, breaking queries and programs which use the table. Use of the corresponding views will ensure compatibility with future Babase releases.

2 The Babase ER Diagrams

If we could we would display the diagram key here.

Figure 1: Key to the Babase Entity Relationship Diagrams

If we could we would display a diagram here depicting censusing and group membership.

Figure 2: Babase Group Membership Entity Relationship Diagram

If we could we would display here a diagram depicting maturity markers and ranking.

Figure 3: Babase Life Events Entity Relationship Diagram

If we could we would display here a diagram depicting paternity analyses.

Figure 4: Babase Paternity Entity Relationship Diagram

[1] At this time of this writing only males have data entered into RANKDATES in *Babase:*.

Group Membership and Life Events	
Table	One row for each
ALTERNATE_SNAMES in *Babase:*	rescinded sname
BIOGRAPH in *Babase:*	animal, including fetuses
CENSUS in *Babase:*	day each individual is (or is not) observed in a group
CONSORTDATES in *Babase:*	male who has a known first consortship
DEMOG in *Babase:*	mention of an individual's presence in a group within a field textual note
DISPERSEDATES in *Babase:*	male who has left his maternal study group
GROUPS in *Babase:*	group (including solitary males)
MATUREDATES in *Babase:*	individual who is sexually mature
RANKDATES in *Babase:*	individual[1] who has attained adult rank
Analyzed: Group Membership and Life Events	
Table	One row for each
DADS_ANALYSES in *Babase:*	paternity analysis
DADS_CONSENSUS in *Babase:*	kid with a known dad
DADS_EVIDENCE in *Babase:*	datum used in a paternity analysis
MEMBERS in *Babase:*	day each individual is alive
RANKS in *Babase:*	month each individual is ranked in each group
RESIDENCIES in *Babase:*	bout of each individual's residency
Physical Traits	
Table	One row for each
WP_AFFECTEDPARTS in *Babase:*	body part affected by a specific wound/pathology
WP_DETAILS in *Babase:*	wound or pathology cluster indicated on a report
WP_HEALUPDATES in *Babase:*	update on progress of wound/pathology healing
WP_REPORTS in *Babase:*	wound/pathology report
Analyzed: Physical Traits	
Table	One row for each
HORMONE_KITS in *Babase:*	kit or protocol used to assay hormone concentration
HORMONE_PREP_DATA in *Babase:*	laboratory preparation performed on a sample in the specified series
HORMONE_PREP_SERIES in *Babase:*	series of preparations and assays performed on a sample
HORMONE_RESULT_DATA in *Babase:*	assay for hormone concentration in a sample
HORMONE_SAMPLE_DATA in *Babase:*	tissue sample used in hormone analysis
HYBRIDGENE_ANALYSES in *Babase:*	analysis of genetic hybrid scores
HYBRIDGENE_SCORES in *Babase:*	genetic hybrid score for an individual from an analysis
HYBRIDMORPH_OBSERVERS in *Babase:*	observer in a morphological hybrid score report
HYBRIDMORPH_REPORTS in *Babase:*	morphological hybrid scoring event, per scored individual
HYBRIDMORPH_SCORE_DATA in *Babase:*	morphological hybrid score for a particular trait

Table	One row for each
INTEGRITY_QUERIES in *Babase:*	query used to discover data integrity problems
INTEGRITY_WARNINGS in *Babase:*	data integrity problem discovered by the warning sub-system

Table 2: The Warning Sub-System Tables

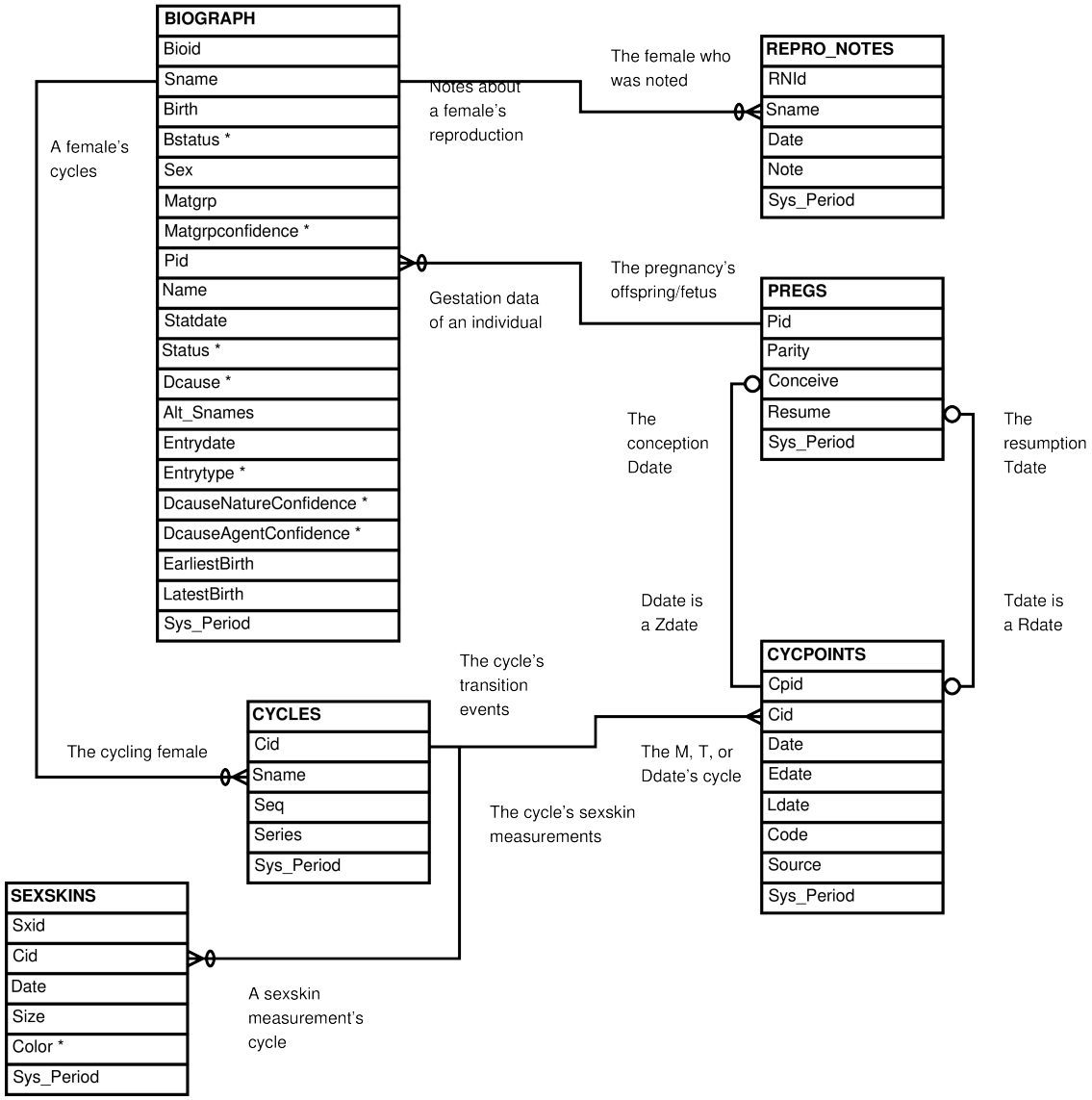

Figure 5: Babase Sexual Cycle Entity Relationship Diagram

General Support Tables			
Table	Id Column	Related Column(s)	One entry for every possible choice of...
BODYPARTS in *Babase:*	Bodypart	TICKS in *Babase:*.Bodypart in *Babase:*, BODYPARTS in *Babase:*.Bodyregion in *Babase:*, WP_AFFECTEDPARTS in *Babase:*.Bodypart in *Babase:*	part of the body
LAB_PERSONNEL in *Babase:*	Initials	HYBRIDGENE_ANALYSES in *Babase:*.Analyzed_By in *Babase:*, NUCACID_CREATORS in *Babase:*.Creator in *Babase:*, WBC_COUNTS in *Babase:*.Counted_By in *Babase:*	person who generates data, usually in a lab setting
OBSERVERS in *Babase:*	Initials	SAMPLES in *Babase:*.Observer in *Babase:*, WREADINGS in *Babase:*.WRperson in *Babase:*, RGSETUPS in *Babase:*.RGSPerson in *Babase:*, CROWNRUMPS in *Babase:*.CRobserver in *Babase:*, CHESTS in *Babase:*.Chobserver in *Babase:*, ULNAS in *Babase:*.Ulobserver in *Babase:*, HUMERUSES in *Babase:*.Huobserver in *Babase:*, SWERB_OBSERVERS in *Babase:*.Observer in *Babase:*	person who record observational data
OBSERVER_ROLES in *Babase:*	Initials	OBSERVERS in *Babase:*.Role in *Babase:*, OBSERVERS in *Babase:*.SWERB_Observer_Role in *Babase:*, OBSERVERS in *Babase:*.SWERB_Driver_Role in *Babase:*, SWERB_OBSERVERS in *Babase:*.Role in *Babase:*	way in which a person can be involved in the data collection process
UNKSNAMES in *Babase:*	Unksname	NEIGHBORS in *Babase:*.Unksname in *Babase:* and the SWERB_UPLOAD in *Babase:* view	problem in identifying neighbor of focal during point sampling or in identifying a lone male in a SWERB other group observation

Group Membership and Life Events			
Table	Id Column	Related Column(s)	One entry for every possible choice of...
BSTATUSES in *Babase:*	Bstatus	BIOGRAPH in *Babase:*.Bstatus in *Babase:*	birthday estimation accuracy
CONFIDENCES in *Babase:*	Confidence	BIOGRAPH in *Babase:*.DcauseNatureConfidence in *Babase:*, BIOGRAPH in *Babase:*.DcauseAgentConfidence in *Babase:*, DISPERSEDATES in *Babase:*.Dispconfidence in *Babase:*, BIOGRAPH in *Babase:*.Matgrpconfidence in *Babase:*	degree of certitude in nature of death, agent of death, disperse date assignment, or maternal group assignment

Table	Id Column	Related Column(s)	One entry for every possible choice of...
IQTYPES in *Babase:*	IQType	INTEGRITY_QUERIES in *Babase:*.Type in *Babase:*	kind of problem with data integrity
WARNING_REMARKS in *Babase:*	WRID	INTEGRITY_WARNINGS in *Babase:*.Category in *Babase:*	remark which might apply to more than one instance of questionable database integrity

Table 4: The Warning Sub-System Support Tables

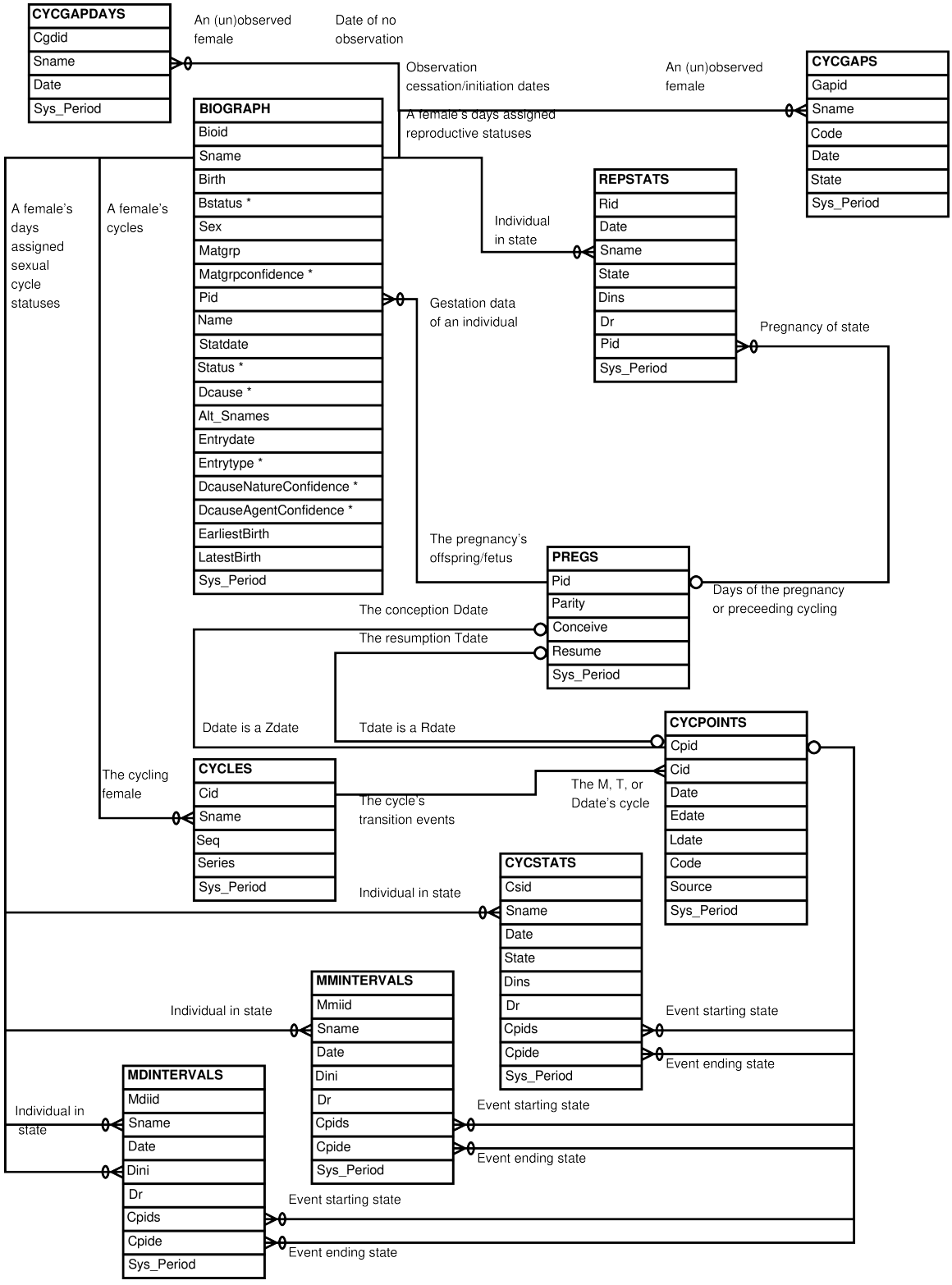

Figure 6: Babase Sexual Cycle Day-To-Day Tables Entity Relationship Diagram

Group Membership and Life Events			
View	One row for each	Purpose	Tables/Views used
CENSUS_DEMOG in *Babase:*	CENSUS in *Babase:* row	Maintenance of CENSUS in *Babase:* rows that are extended with DEMOG in *Babase:* information.	CENSUS in *Babase:*, DEMOG in *Babase:*
CENSUS_DEMOG_SORTED in *Babase:*	CENSUS in *Babase:* row	Maintenance of CENSUS_DEMOG in *Babase:* rows in a pre-sorted fashion.	CENSUS in *Babase:*, DEMOG in *Babase:*
CYCPOINTS_CYCLES in *Babase:*	CYCPOINTS in *Babase:* row	Maintenance of CYCPOINTS in *Babase:* rows that are extended with CYCLES in *Babase:* information.	CYCLES in *Babase:*, CYCPOINTS in *Babase:*
CYCPOINTS_CYCLES_SORTED in *Babase:*	CYCPOINTS in *Babase:* row	The CYCPOINTS_CYCLES in *Babase:* view sorted by CYCLES in *Babase:*.Sname in *Babase:*, by CYCPOINTS in *Babase:*.Date in *Babase:*.	CYCLES in *Babase:*, CYCPOINTS in *Babase:*
DEMOG_CENSUS in *Babase:*	DEMOG in *Babase:* row	Maintenance of DEMOG in *Babase:* rows.	CENSUS in *Babase:*, DEMOG in *Babase:*
DEMOG_CENSUS_SORTED in *Babase:*	CENSUS in *Babase:* row	Maintenance of DEMOG_CENSUS in *Babase:* rows in a pre-sorted fashion.	CENSUS in *Babase:*, DEMOG in *Babase:*
GROUPS_HISTORY in *Babase:*	GROUPS in *Babase:* row	Depiction of GROUPS in *Babase:* rows in a more human-readable format.	GROUPS in *Babase:*
PARENTS in *Babase:*	BIOGRAPH in *Babase:* row for which there is either a row in MATERNITIES in *Babase:* with a record of the individual's mother or there is a row in DADS_CONSENSUS in *Babase:* with a record of the individual's father -- with a non-`NULL`Dad_Consensus in *Babase:*.	Easy access to parental information.	BIOGRAPH in *Babase:*, MATERNITIES in *Babase:*, DADS_CONSENSUS in *Babase:*, MEMBERS in *Babase:*
POTENTIAL_DADS in *Babase:*	(completed) female reproductive event for every male more than 2192 days old (approximately 6 years) present in the mother's group during her fertile period	Research into paternity, especially the selection of potential fathers for further genetic testing.	MATERNITIES in *Babase:*, MEMBERS in *Babase:* (multiple times), ACTOR_ACTEES in *Babase:* (multiple times), BIOGRAPH in *Babase:*, RANKDATES in *Babase:*, MATUREDATES in *Babase:*
PROPORTIONAL_RANKS in *Babase:*	RANKS in *Babase:* row	Automatic calculation of proportional ranks from the ordinal ranks in RANKS in *Babase:*.	RANKS in *Babase:*
Physical Traits			
View	One row for each	Purpose	Tables/Views

Table	View
BIOGRAPH in *Babase:*	BIRTH_GRP in *Babase:*
BIOGRAPH in *Babase:*	ENTRYDATE_GRP in *Babase:*
BIOGRAPH in *Babase:*	STATDATE_GRP in *Babase:*
CONSORTDATES in *Babase:*	CONSORTDATES_GRP in *Babase:*
CYCGAPDAYS in *Babase:*	CYCGAPDAYS_GRP in *Babase:*
CYCGAPS in *Babase:*	CYCGAPS_GRP in *Babase:*
CYCSTATS in *Babase:*	CYCSTATS_GRP in *Babase:*
DARTINGS in *Babase:*	DARTINGS_GRP in *Babase:*
DISPERSEDATES in *Babase:*	DISPERSEDATES_GRP in *Babase:*
MATUREDATES in *Babase:*	MATUREDATES_GRP in *Babase:*
MDINTERVALS in *Babase:*	MDINTERVALS_GRP in *Babase:*
MMINTERVALS in *Babase:*	MMINTERVALS_GRP in *Babase:*
RANKDATES in *Babase:*	RANKDATES_GRP in *Babase:*
REPSTATS in *Babase:*	REPSTATS_GRP in *Babase:*

Table 6: The table_GRP Views

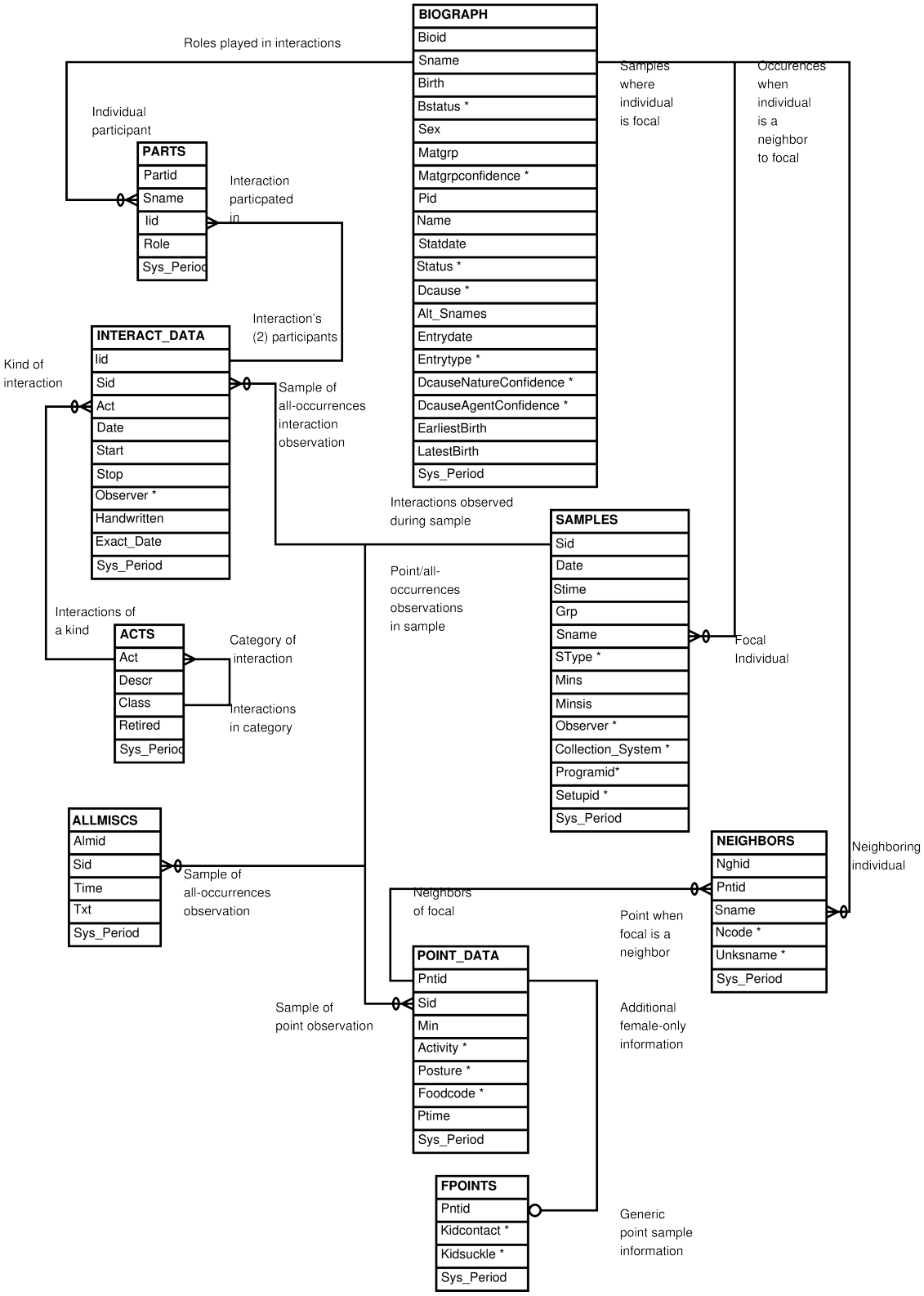

Figure 7: Babase Social Interactions Entity Relationship Diagram

If we could we would display a diagram here depicting multiparty interactions.

Figure 8: Babase Multiparty Interactions Entity Relationship Diagram

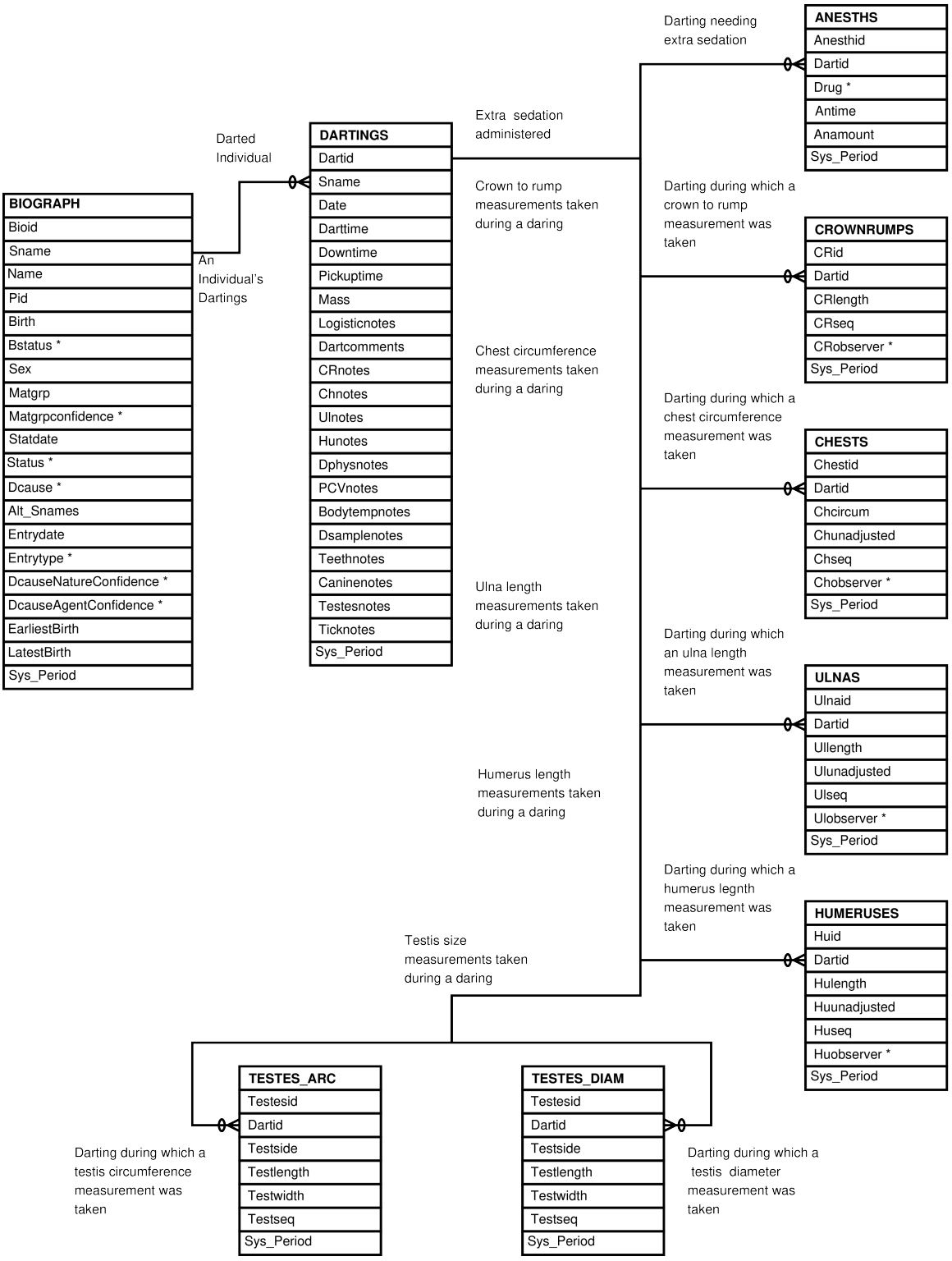

Figure 9: Babase Darting Logistics and Morphology Entity and Relationship Diagram

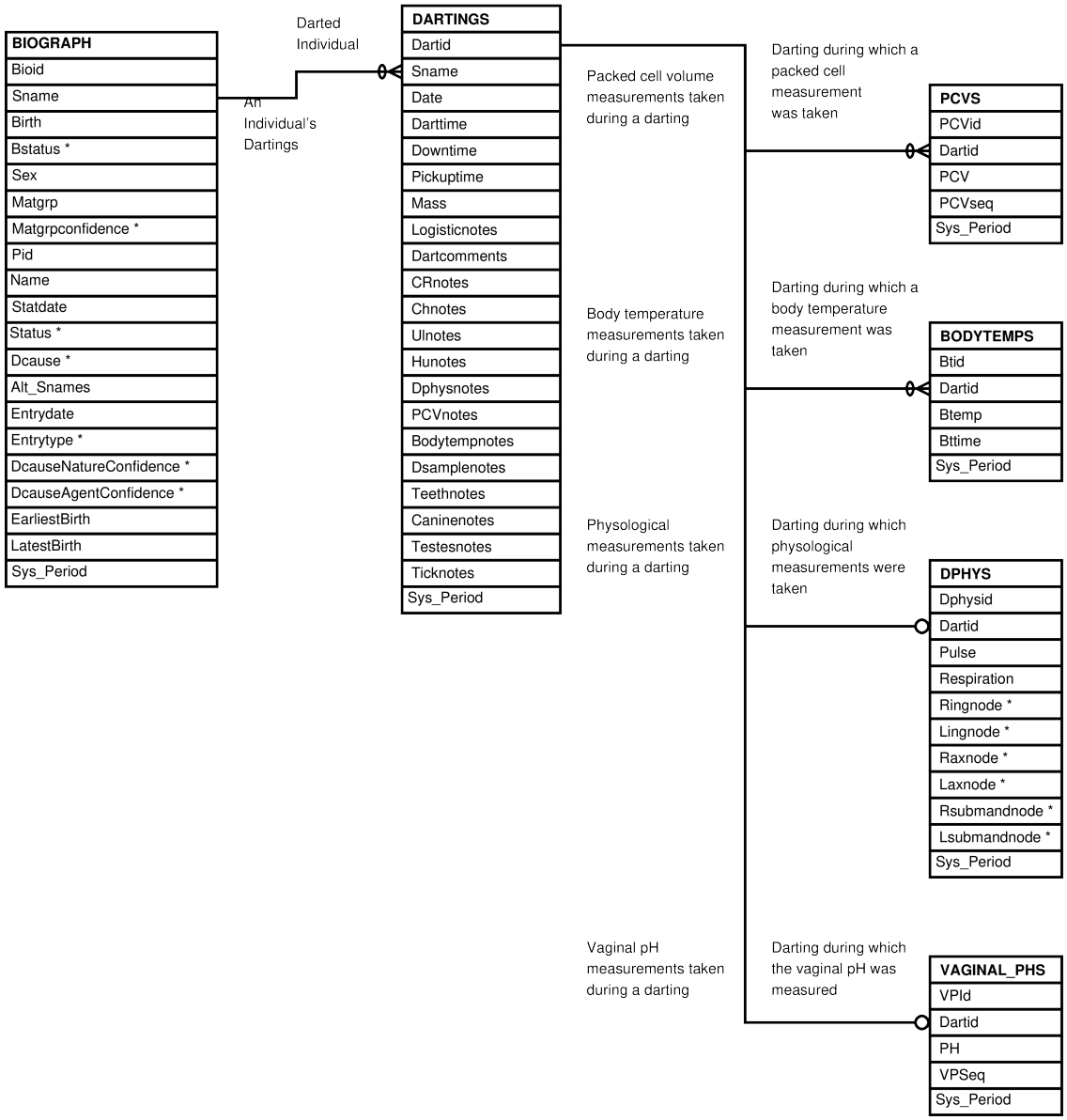

Figure 10: Babase Darting Physiology Entity and Relationship Diagram

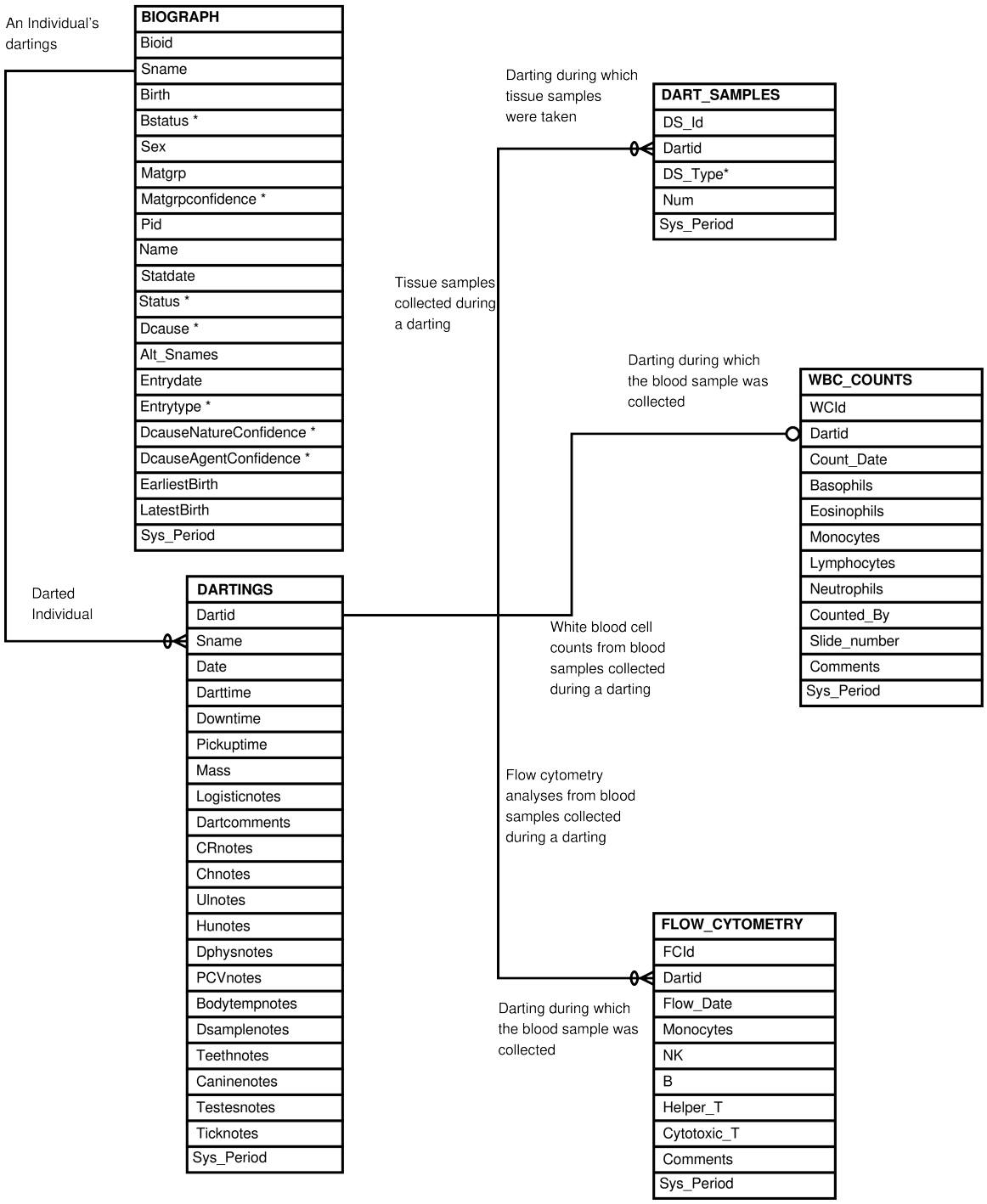

Figure 11: Babase Darting Samples Entity and Relationship Diagram

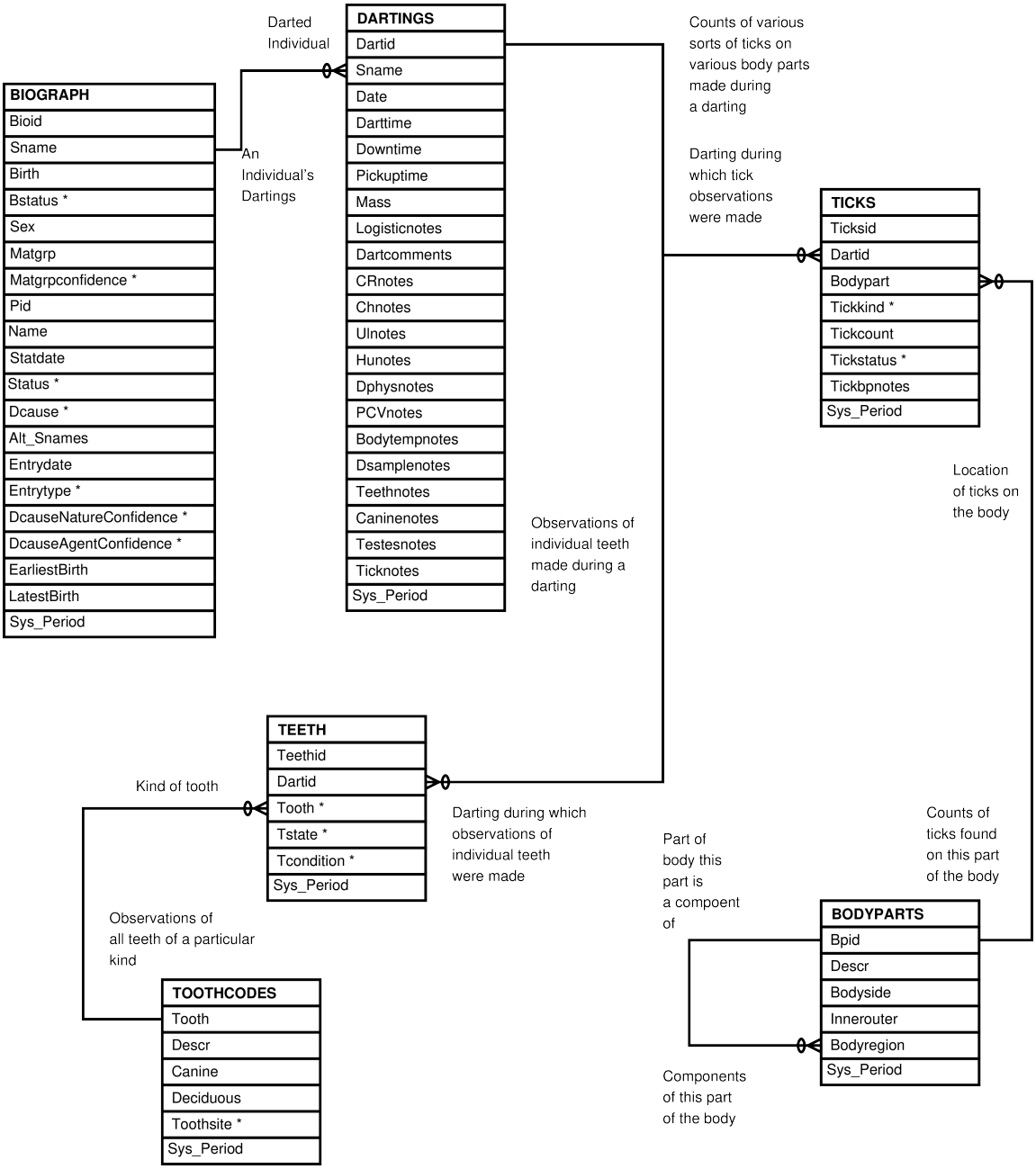

Figure 12: Babase Darting Teeth and Ticks Entity and Relationship Diagram

If we could we would display a diagram here depicting the Babase Inventory tables.

Figure 13: Babase Inventory Entity Relationship Diagram

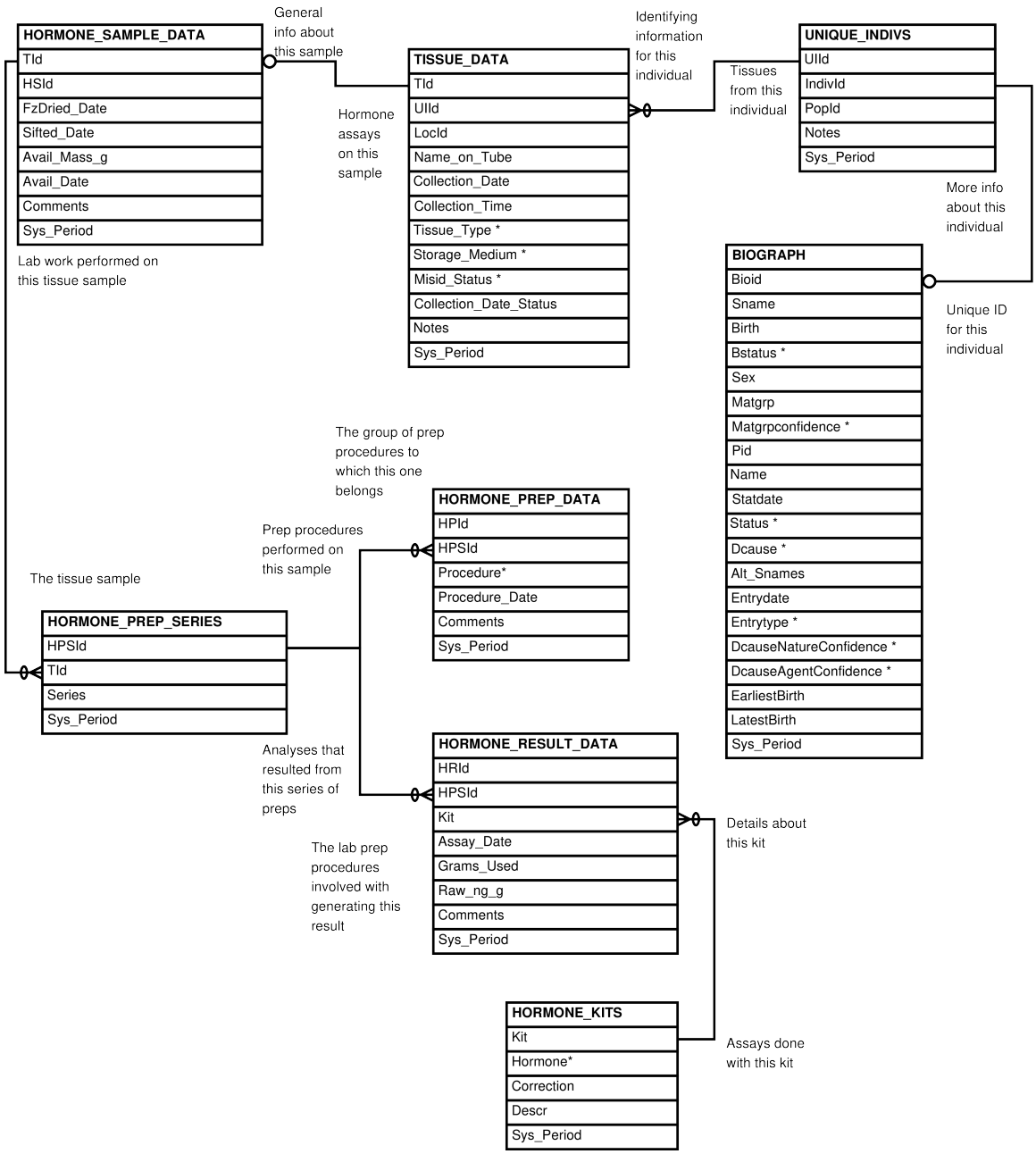

Figure 14: Babase Physical Traits Hormone Data Entity Relationship Diagram

If we could we would display a diagram here depicting the Babase Physical Traits Hybrid Score Data tables.

Figure 15: Babase Physical Traits Hybrid Score Data Entity Relationship Diagram

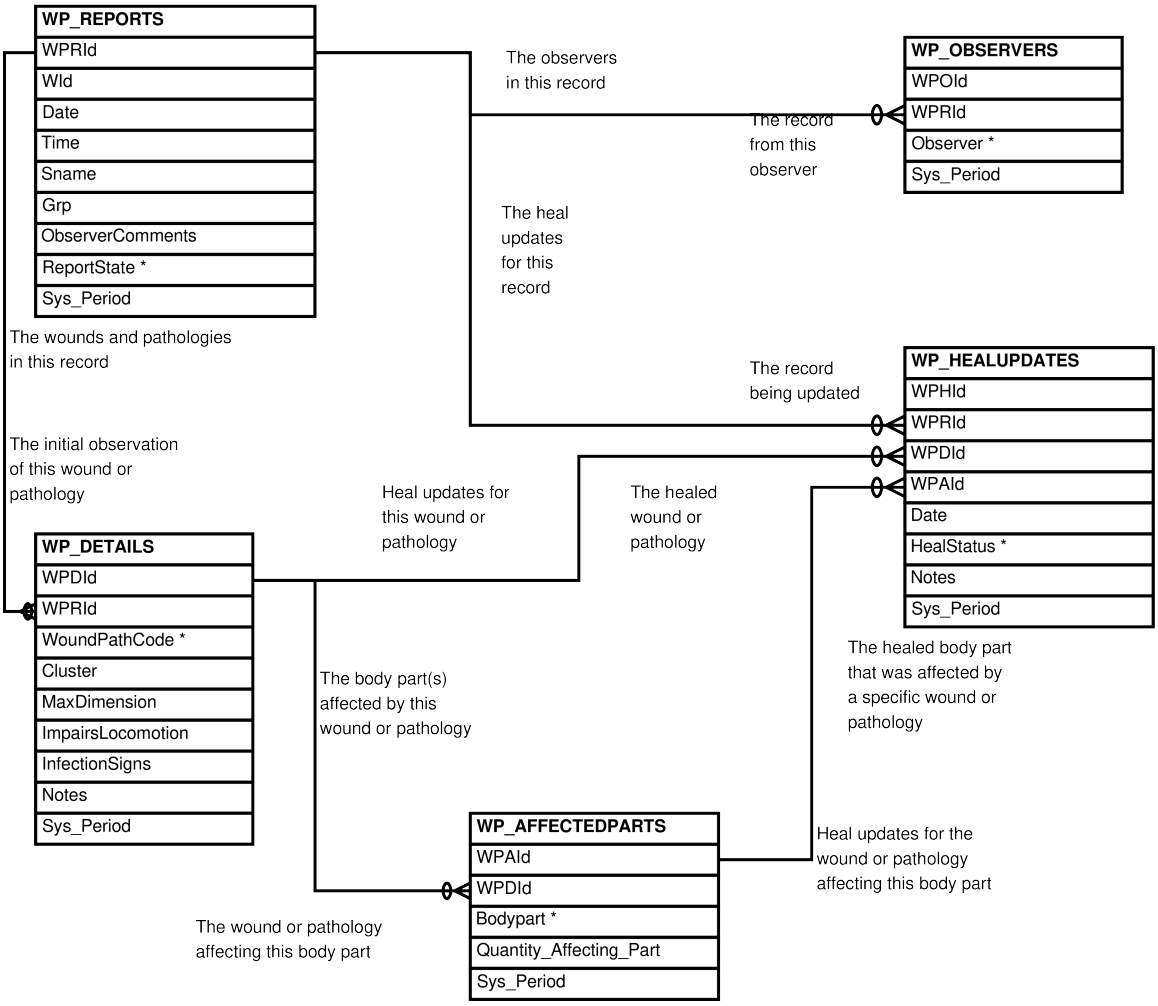

Figure 16: Babase Physical Traits Wounds and Pathologies Data Entity Relationship Diagram

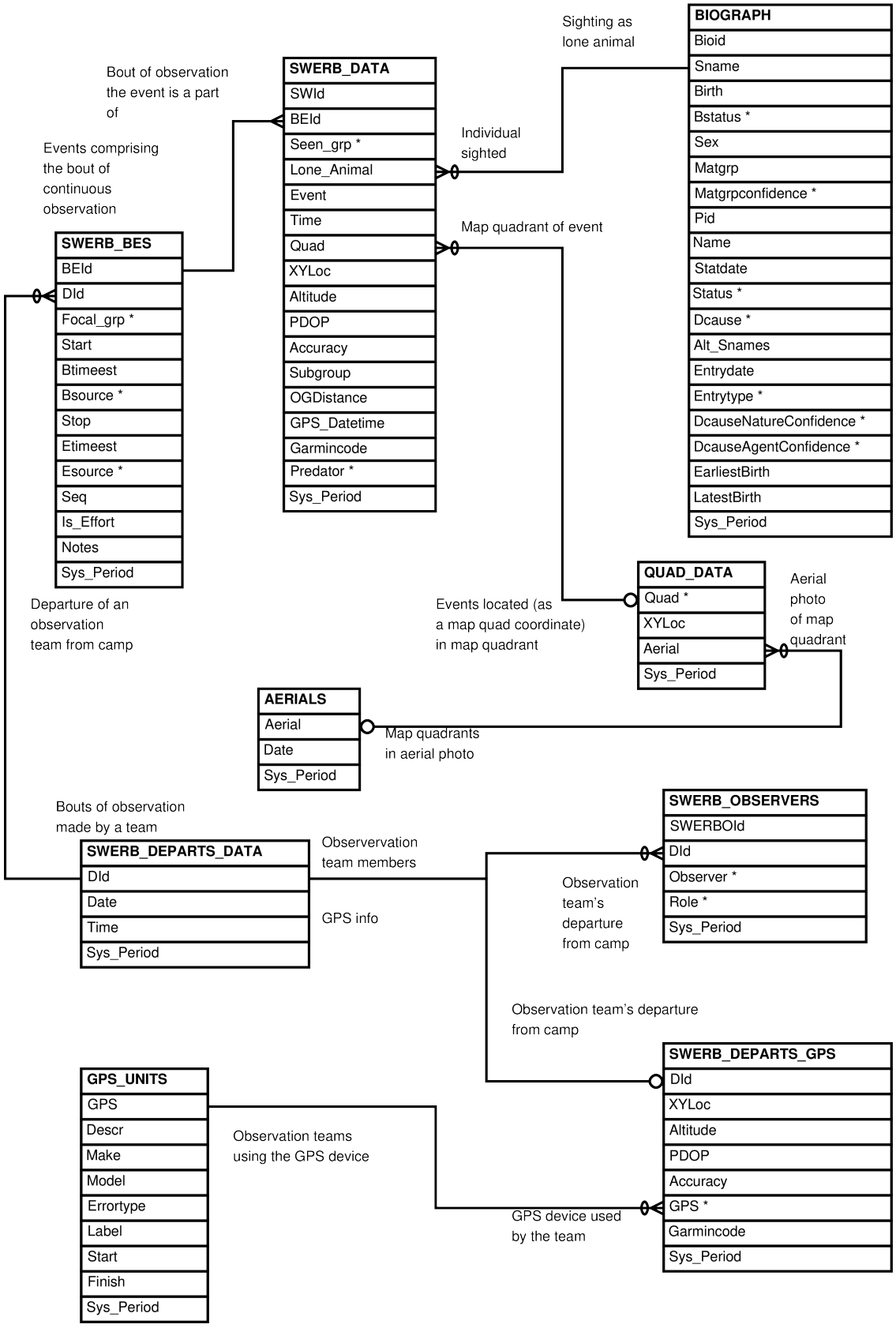

Figure 17: Babase SWERB Core Tables Entity Relationship Diagram

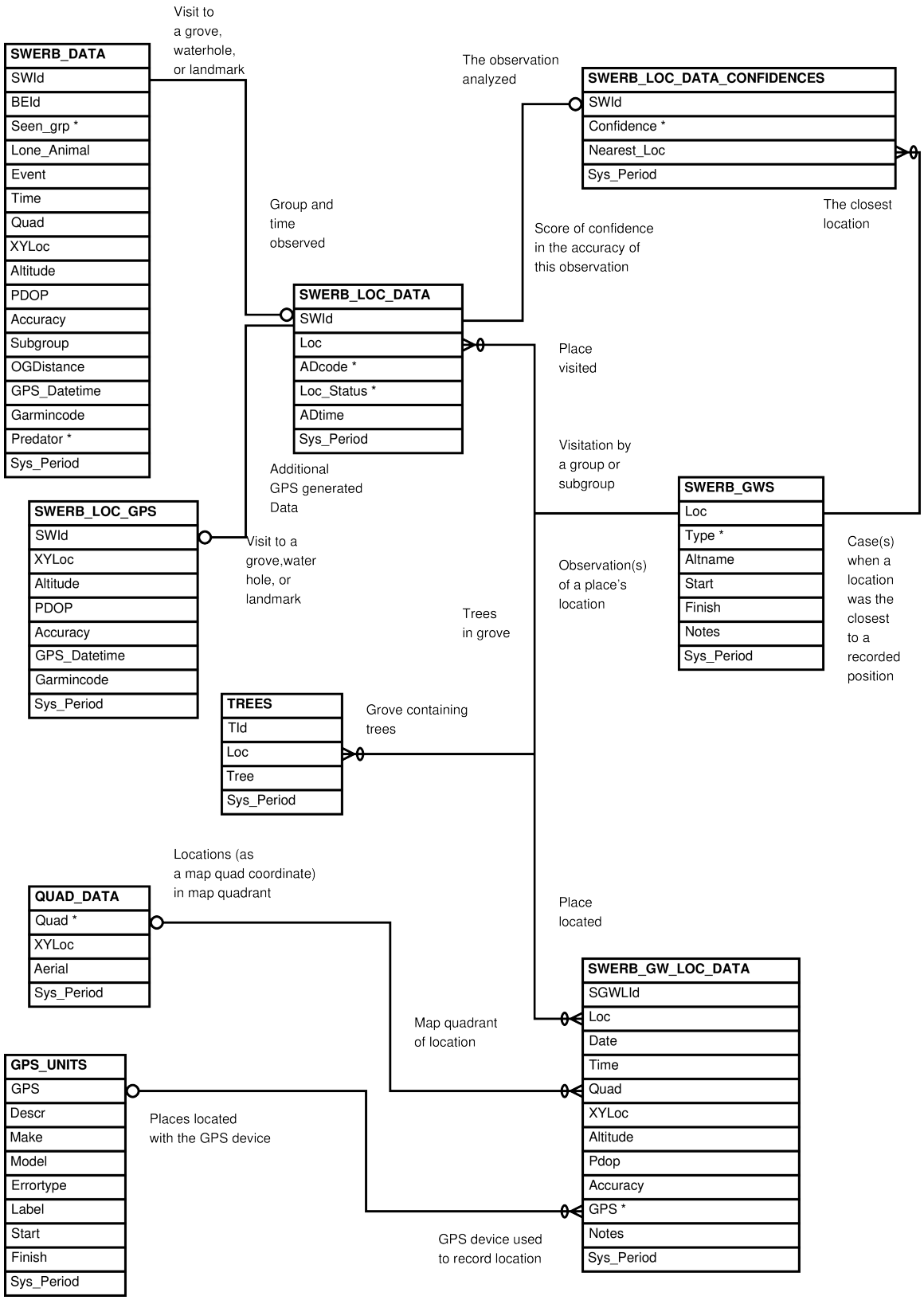

Figure 18: Babase SWERB Grove/Waterhole Location Tables Entity Relationship Diagram

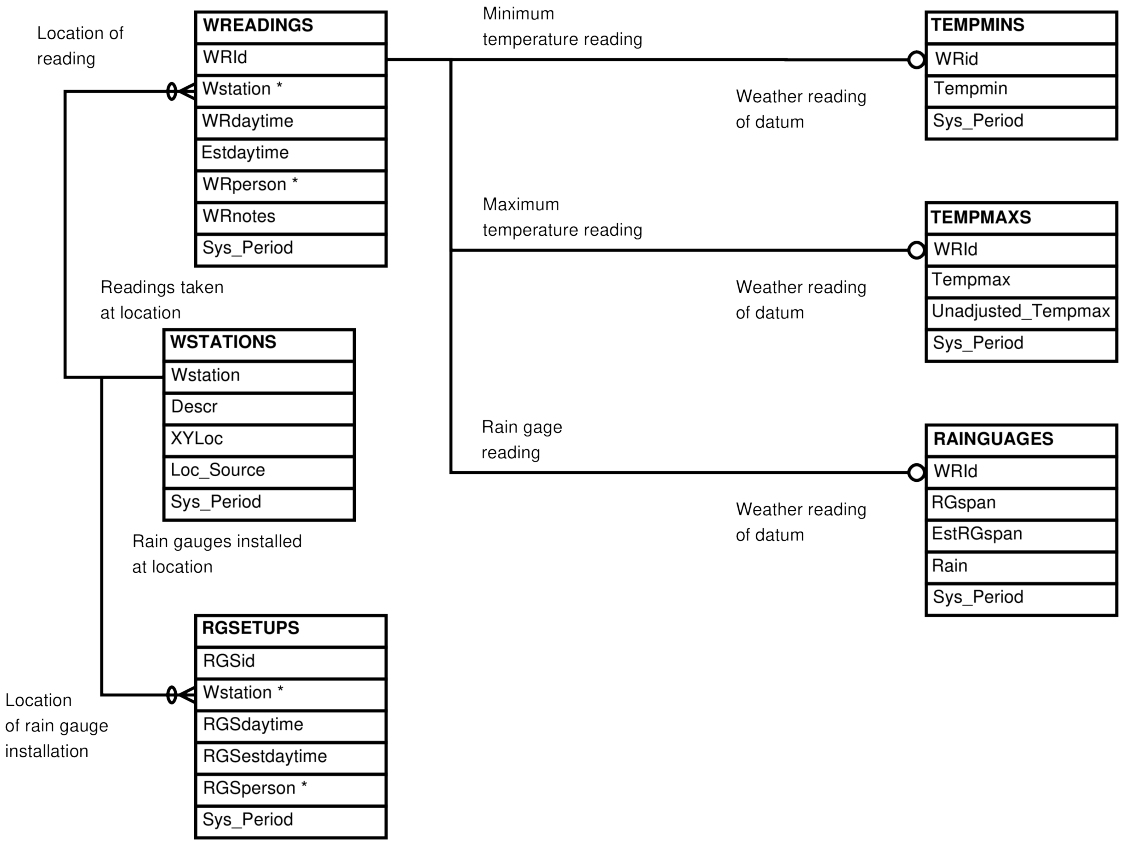

Figure 19: Babase Manual Weather Data Entity Relationship Diagram

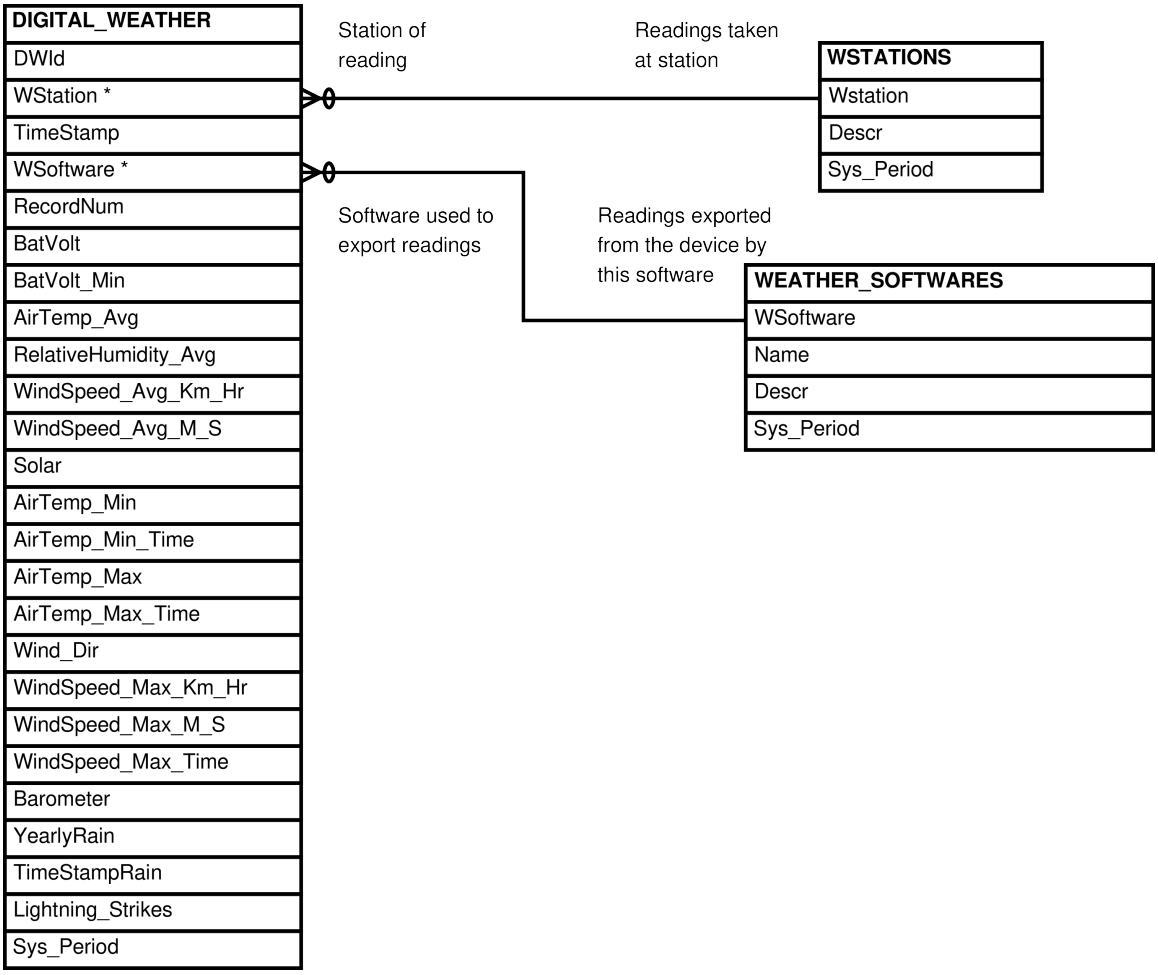

Figure 20: Babase Digital Weather Data Entity Relationship Diagram

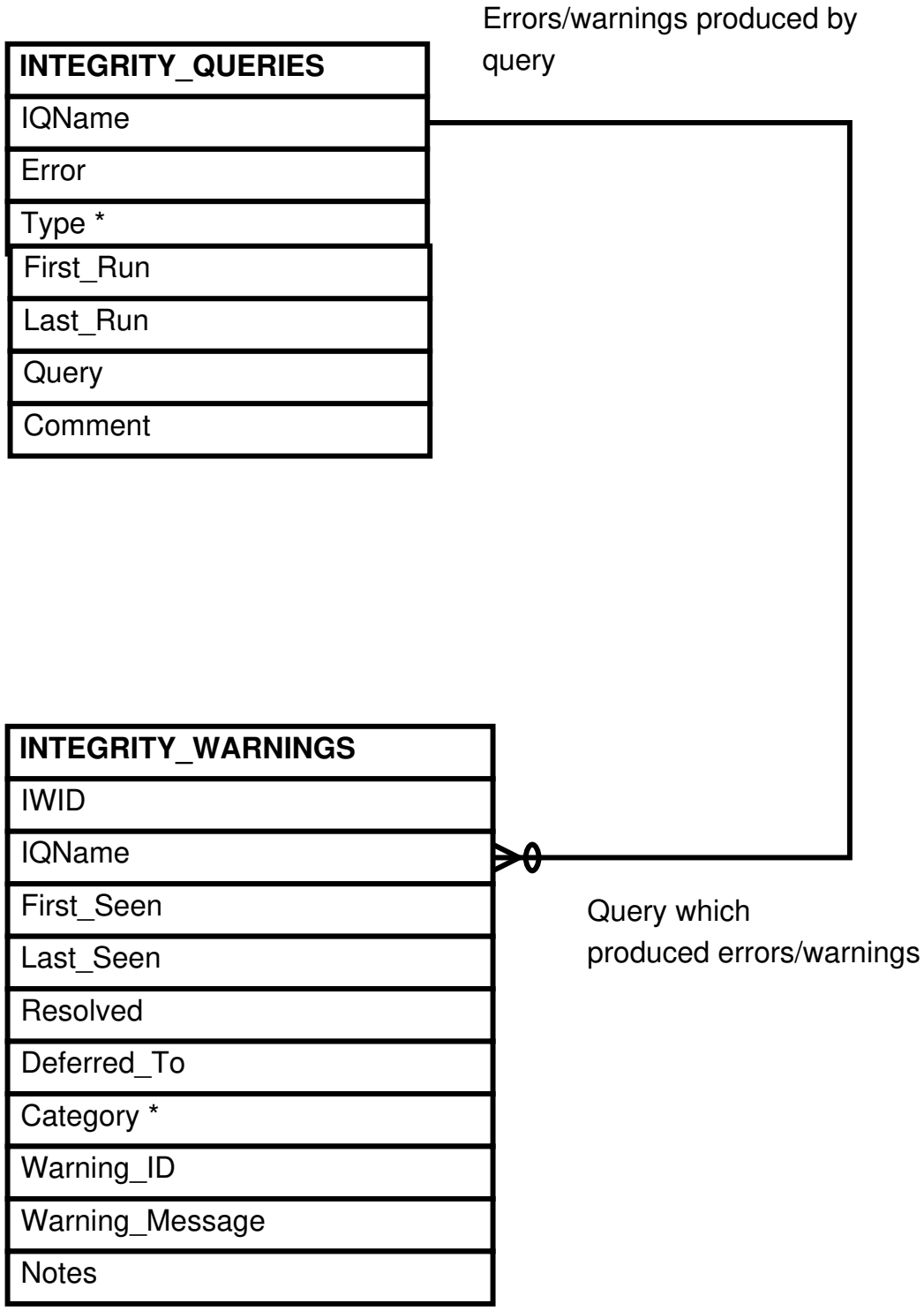

Figure 21: Warning Sub-System Entity Relationship Diagram

3 The Babase Views

For information on the operations (INSERT, UPDATE, DELETE) allowed by each view and their actions on the underlying tables see The Babase Views in *Babase:* of The Babase Reference Manual.

3.1 The ACTOR_ACTEES View

```
WITH sampling AS (SELECT samples.date
                       , samples.observer
                       , ARRAY_AGG(DISTINCT members.grp) AS grps
                    FROM samples
                    JOIN stypes
                      ON stypes.stype = samples.stype
                         AND stypes.repr_interxns
                    JOIN members
                      ON members.sname = samples.sname
                         AND members.date = samples.date
                    WHERE samples.minsis > 0
                    GROUP BY samples.date, samples.observer)
SELECT interact_data.iid AS iid
     , interact_data.sid AS sid
     , interact_data.act AS act
     , interact_data.date AS date
     , interact_data.start AS start
     , interact_data.stop AS stop
     , interact_data.observer AS observer
     , actor.partid AS actorid
     , COALESCE(actor.sname, '998'::CHAR(3)) AS actor
     , actorms.grp AS actor_grp
     , actee.partid AS acteeid
     , COALESCE(actee.sname, '998'::CHAR(3)) AS actee
     , acteems.grp AS actee_grp
     , interact_data.handwritten AS handwritten
     , interact_data.exact_date AS exact_date
     , COALESCE((actorms.grp = ANY(sampling.grps)
                  OR acteems.grp = ANY(sampling.grps))
                , FALSE) AS repr_interxn
  FROM interact_data
       LEFT OUTER JOIN parts AS actor
            ON (actor.iid = interact_data.iid AND actor.role = 'R')
       LEFT OUTER JOIN parts AS actee
            ON (actee.iid = interact_data.iid AND actee.role = 'E')
       LEFT OUTER JOIN members AS actorms
            ON (actorms.sname = actor.sname
                  AND actorms.date = interact_data.date)
       LEFT OUTER JOIN members AS acteems
            ON (acteems.sname = actee.sname
                  AND acteems.date = interact_data.date)
       LEFT OUTER JOIN sampling
            ON (sampling.date = interact_data.date
                  AND sampling.observer = interact_data.observer);
```

Figure 22: Query Defining the ACTOR_ACTEES View

If we could we would display here the diagram showing how the ACTOR_ACTEES view is constructed.

Figure 23: Entity Relationship Diagram of the ACTOR_ACTEES View

3.2 The ANESTH_STATS View

```
SELECT anesths.dartid AS dartid
     , count(*) AS ansamps
     , avg(anesths.anamount) AS anamount_mean
     , stddev(anesths.anamount) AS anamount_stddev
  FROM anesths
  GROUP BY anesths.dartid;
```

Figure 24: Query Defining the ANESTH_STATS View

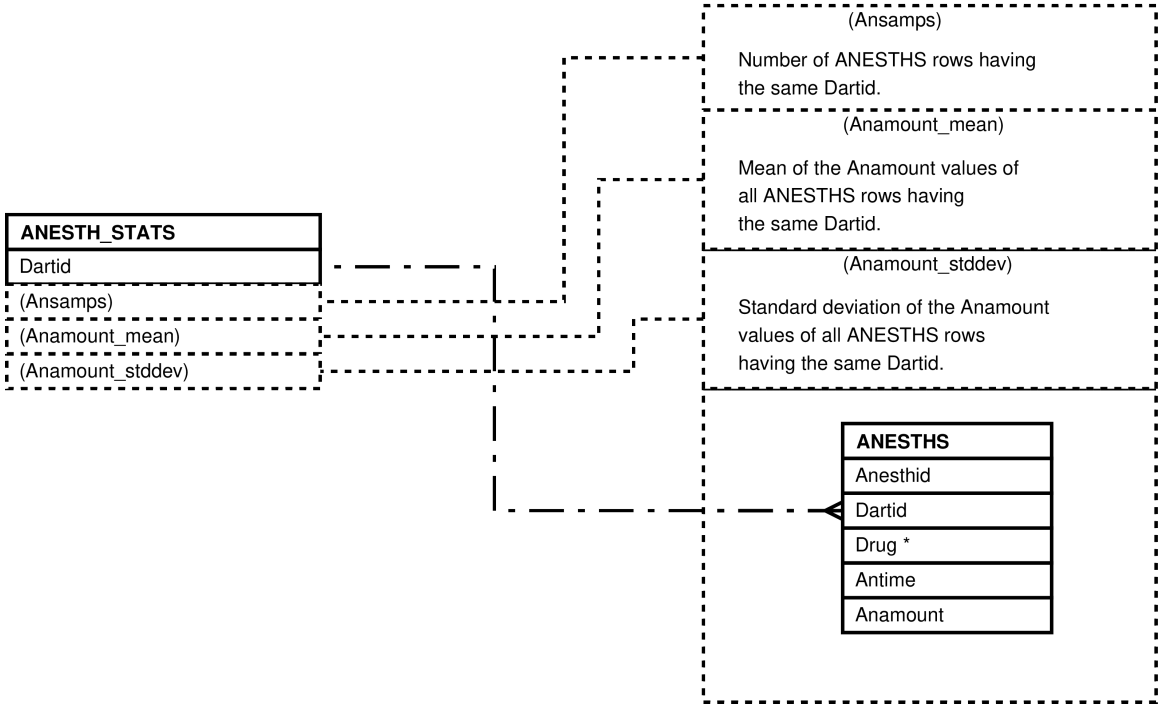

Figure 25: Entity Relationship Diagram of the ANESTH_STATS View

3.3 The BODYTEMP_STATS View

```
SELECT bodytemps.dartid AS dartid
     , count(*) AS btsamps
     , avg(bodytemps.btemp) AS btemp_mean
     , stddev(bodytemps.btemp) AS btemp_stddev
  FROM bodytemps
  GROUP BY bodytemps.dartid;
```

Figure 26: Query Defining the BODYTEMP_STATS View

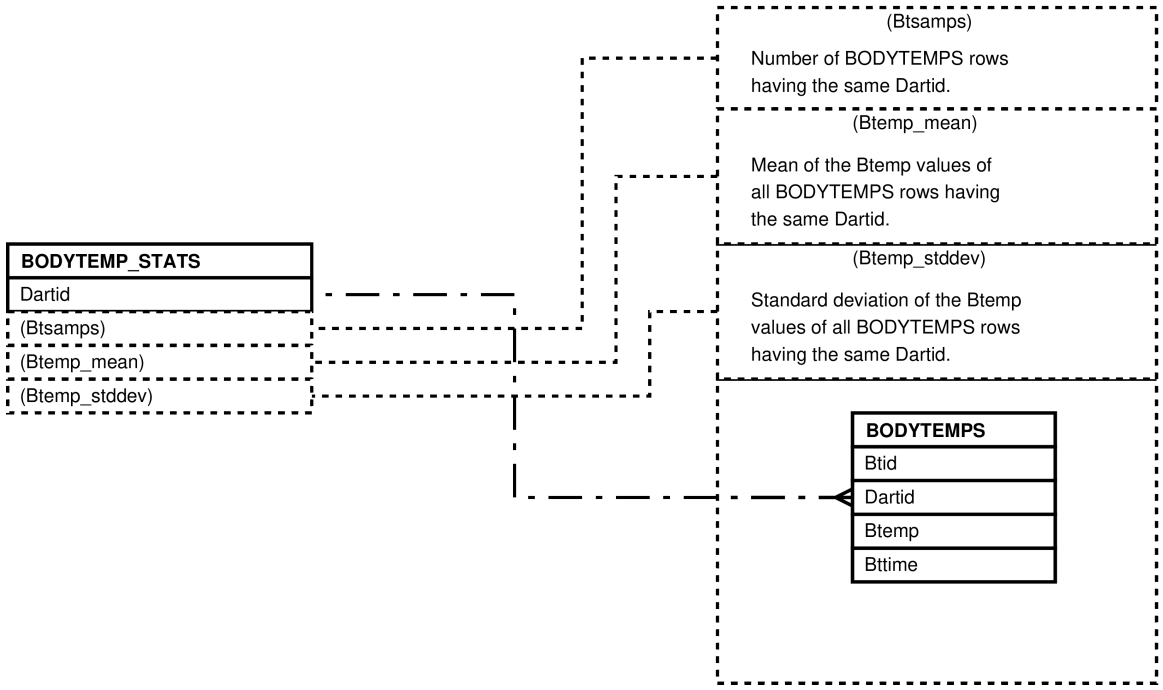

Figure 27: Entity Relationship Diagram of the BODYTEMP_STATS View

3.4 The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views

```
SELECT census.cenid AS cenid
     , census.sname AS sname
     , census.date AS date
     , census.grp AS grp
     , census.status AS status
     , census.cen AS cen
     , demog.reference AS reference
     , demog.comment AS comment
  FROM census LEFT OUTER JOIN demog ON (census.cenid = demog.cenid);
```

Figure 28: Query Defining the CENSUS_DEMOG View

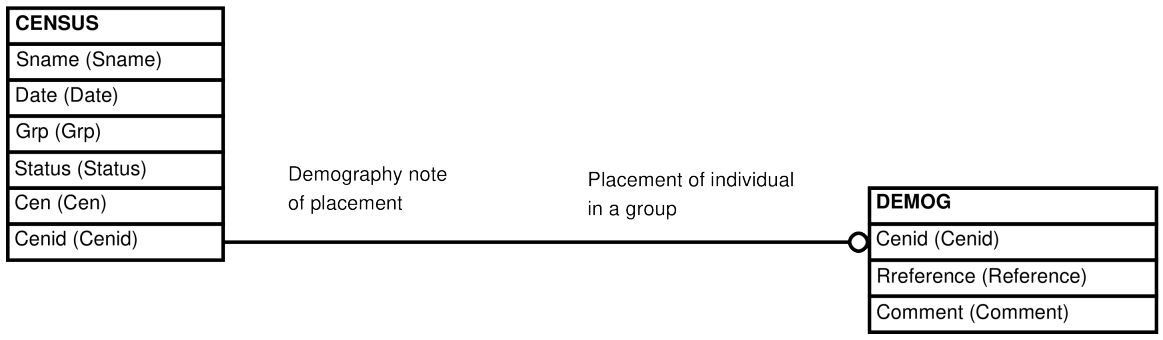

Figure 29: Entity Relationship Diagram of the CENSUS_DEMOG View

3.5 The CHEST_STATS View

```
SELECT chests.dartid AS dartid
     , count(*) AS chsamps
     , avg(chests.chcircum) AS chcircum_mean
     , stddev(chests.chcircum) AS chcircum_stddev
     , avg(chests.chunadjusted) AS chunadjusted_mean
     , stddev(chests.chunadjusted) AS chunadjusted_stddev
  FROM chests
  GROUP BY chests.dartid;
```

Figure 30: Query Defining the CHEST_STATS View

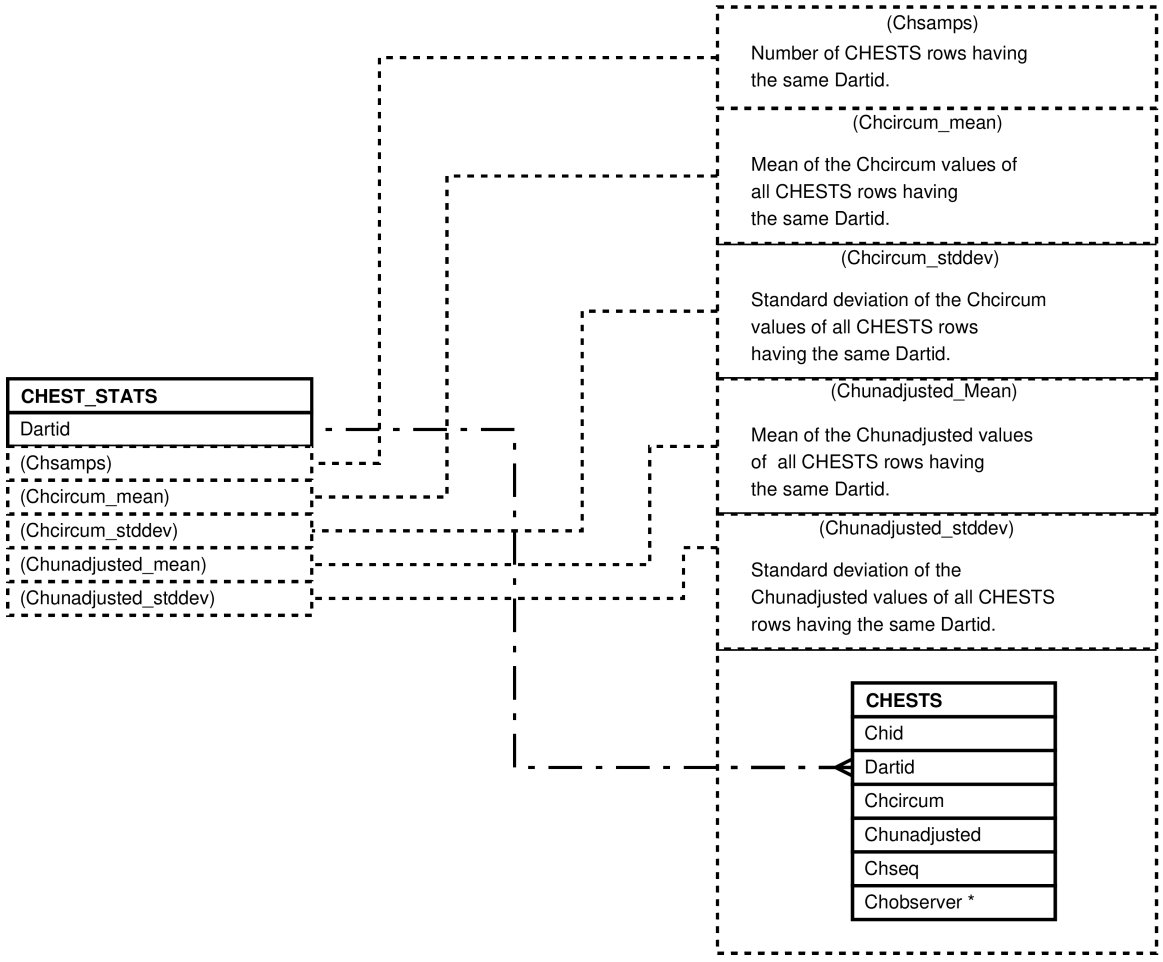

Figure 31: Entity Relationship Diagram of the CHEST_STATS View

3.6 The CROWNRUMP_STATS View

```
SELECT crownrumps.dartid AS dartid
     , count(*) AS crsamps
     , avg(crownrumps.crlength) AS crlength_mean
     , stddev(crownrumps.crlength) AS crlength_stddev
  FROM crownrumps
  GROUP BY crownrumps.dartid;
```

Figure 32: Query Defining the CROWNRUMP_STATS View

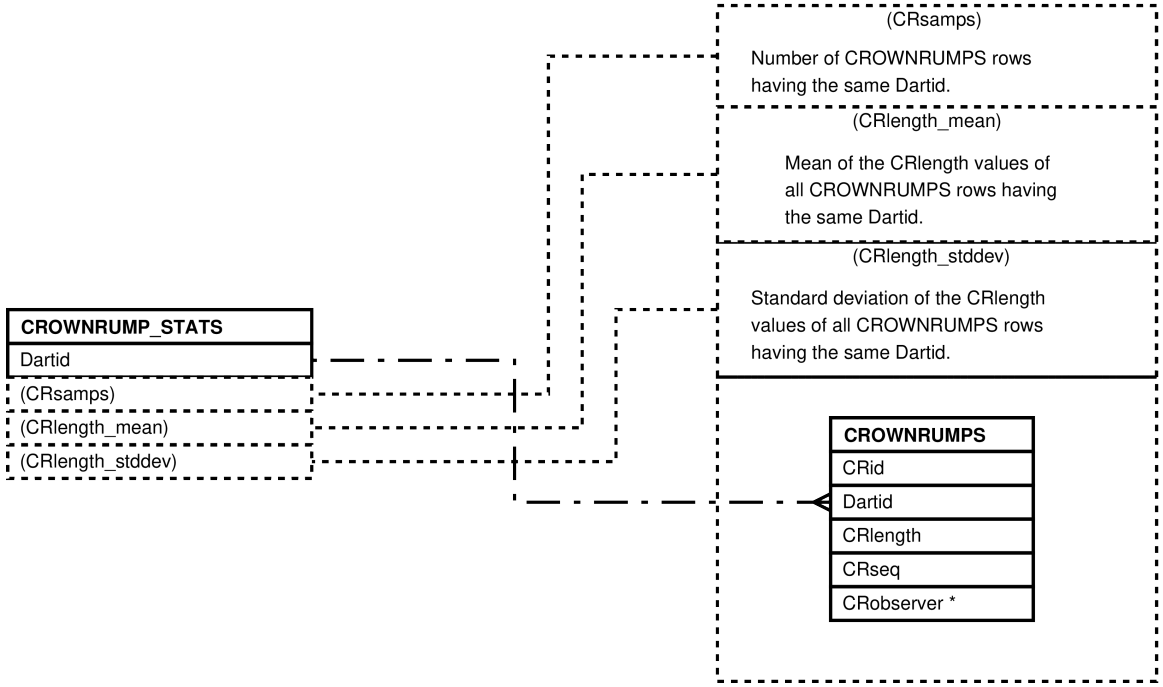

Figure 33: Entity Relationship Diagram of the CROWNRUMP_STATS View

3.7 The CYCLES_SEXSKINS and CYCLES_SEXSKINS_SORTED Views

```
SELECT cycles.cid AS cid
     , cycles.sname AS sname
     , cycles.seq AS seq
     , cycles.series AS series
     , sexskins.sxid AS sxid
     , sexskins.date AS date
     , sexskins.size AS size
     , sexskins.color AS color
  FROM cycles LEFT OUTER JOIN sexskins ON (cycles.cid = sexskins.cid);
```

Figure 34: Query Defining the CYCLES_SEXSKINS View

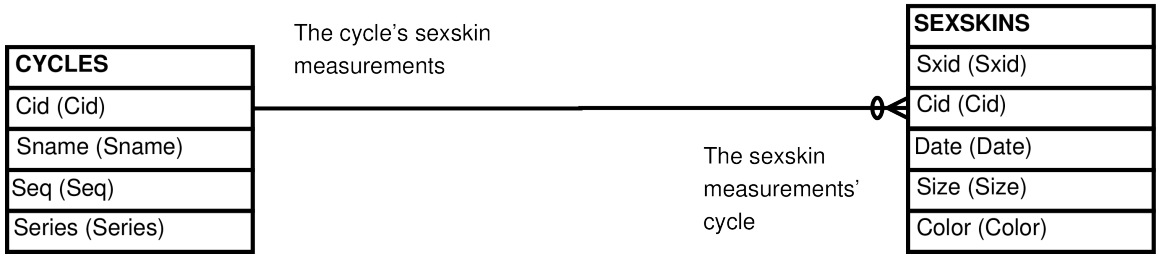

Figure 35: Entity Relationship Diagram of the CYCLES_SEXSKINS View

3.8 The CYCPOINTS_CYCLES and CYCPOINTS_CYCLES_SORTED Views

```
SELECT cycles.cid AS cid
     , cycles.sname AS sname
     , cycles.seq AS seq
     , cycles.series AS series
     , cycpoints.cpid AS cpid
     , cycpoints.date AS date
     , cycpoints.edate AS edate
     , cycpoints.ldate AS ldate
     , cycpoints.code AS code
     , cycpoints.source AS source
  FROM cycles, cycpoints
  WHERE cycles.cid = cycpoints.cid;
```

Figure 36: Query Defining the CYCPOINTS_CYCLES View

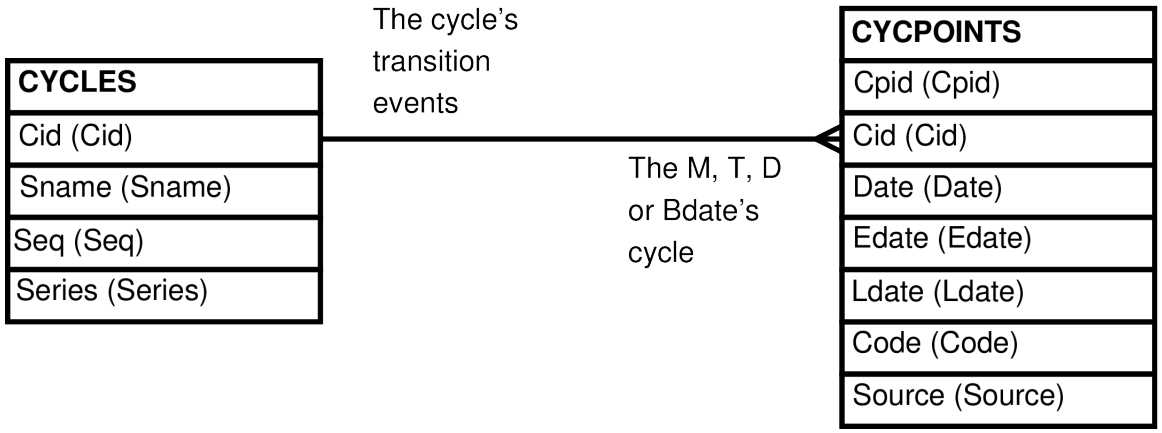

Figure 37: Entity Relationship Diagram of the CYCPOINTS_CYCLES View

3.9 The DSAMPLES View

```
SELECT dartings.dartid
     , dartings.sname
     , dartings.date
     , members.grp
     , blood_unspecs.num AS bloodunspec
     , blood_paxgenes.num AS bloodpaxgene
     , blood_purpletops.num AS bloodpurpletops
     , blood_separators.num AS bloodseptube
     , blood_cpts.num AS bloodcpt
     , blood_trucultures.num AS bloodtruculture
     , blood_smears.num AS bloodsmear
     , tc_bloods.num AS tcblood
     , hair_unspecs.num AS hairunspec
     , hair_lengths.num AS hairlength
     , hair_cu_zns.num AS haircu_zn
     , teeth_3mouths.num AS mouthphotos3
     , teeth_lmandmolds.num AS lmandmold
     , teeth_lmaxmolds.num AS lmaxillamold
     , teeth_lmol1mol2s.num AS lm1m2siliconemold
     , skin_punchs.num AS skinpunch
     , tc_skins.num AS tcskin
     , vag_swabs.num AS vaginalswab
     , cerv_swabs.num AS cervicalswab
     , fecal_formalin.num AS fecal_formalin
     , palm_swab.num AS palm_swab
     , tongue_swab.num AS tongue_swab
     , tooth_plaque_swab.num as tooth_plaque_swab
     , vagswab_microbiome.num AS vagswab_microbiome
     , glans_penis_swab.num AS glans_penis_swab
     , fecal_microbiome.num AS fecal_microbiome
     , nostrils_swab.num AS nostrils_swab
     , skin_behind_ear_swab.num AS skin_behind_ear_swab
     , skin_inside_elbow_swab.num AS skin_inside_elbow_swab
  FROM dartings
       JOIN members
            ON dartings.sname = members.sname
               AND dartings.date = members.date
       LEFT JOIN dart_samples blood_unspecs
            ON dartings.dartid = blood_unspecs.dartid
               AND blood_unspecs.ds_type = 1
       LEFT JOIN dart_samples blood_paxgenes
            ON dartings.dartid = blood_paxgenes.dartid
               AND blood_paxgenes.ds_type = 2
       LEFT JOIN dart_samples blood_purpletops
            ON dartings.dartid = blood_purpletops.dartid
               AND blood_purpletops.ds_type = 3
       LEFT JOIN dart_samples blood_separators
            ON dartings.dartid = blood_separators.dartid
               AND blood_separators.ds_type = 4
       LEFT JOIN dart_samples blood_cpts
            ON dartings.dartid = blood_cpts.dartid
               AND blood_cpts.ds_type = 5
       LEFT JOIN dart_samples blood_trucultures
            ON dartings.dartid = blood_trucultures.dartid
               AND blood_trucultures.ds_type = 6
       LEFT JOIN dart_samples blood_smears
            ON dartings.dartid = blood_smears.dartid
               AND blood_smears.ds_type = 7
       LEFT JOIN dart_samples hair_unspecs
            ON dartings.dartid = hair_unspecs.dartid
               AND hair_unspecs.ds_type = 8
```

3.10 The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views

```
SELECT census.cenid AS cenid
     , census.sname AS sname
     , census.date AS date
     , census.grp AS grp
     , census.status AS status
     , census.cen AS cen
     , demog.reference AS reference
     , demog.comment AS comment
  FROM census, demog
  WHERE census.cenid = demog.cenid;
```

Figure 39: Query Defining the DEMOG_CENSUS View

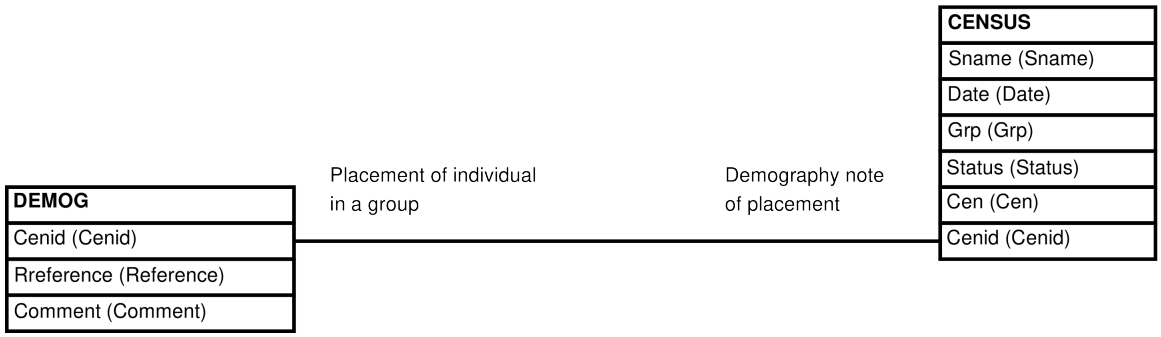

Figure 40: Entity Relationship Diagram of the DEMOG_CENSUS View

3.11 The DENT_CODES View

```
SELECT teethdartids.dartid AS dartid
     , rum3.rum3tstate AS rum3tstate
     , rum3.rum3tcondition AS rum3tcondition
     , rum2.rum2tstate AS rum2tstate
     , rum2.rum2tcondition AS rum2tcondition
     , rum1.rum1tstate AS rum1tstate
     , rum1.rum1tcondition AS rum1tcondition
     , rup2.rup2tstate AS rup2tstate
     , rup2.rup2tcondition AS rup2tcondition
     , rup1.rup1tstate AS rup1tstate
     , rup1.rup1tcondition AS rup1tcondition
     , ruc.ructstate AS ructstate
     , ruc.ructcondition AS ructcondition
     , rui2.rui2tstate AS rui2tstate
     , rui2.rui2tcondition AS rui2tcondition
     , rui1.rui1tstate AS rui1tstate
     , rui1.rui1tcondition AS rui1tcondition
     , lui1.lui1tstate AS lui1tstate
     , lui1.lui1tcondition AS lui1tcondition
     , lui2.lui2tstate AS lui2tstate
     , lui2.lui2tcondition AS lui2tcondition
     , luc.luctstate AS luctstate
     , luc.luctcondition AS luctcondition
     , lup1.lup1tstate AS lup1tstate
     , lup1.lup1tcondition AS lup1tcondition
     , lup2.lup2tstate AS lup2tstate
     , lup2.lup2tcondition AS lup2tcondition
     , lum1.lum1tstate AS lum1tstate
     , lum1.lum1tcondition AS lum1tcondition
     , lum2.lum2tstate AS lum2tstate
     , lum2.lum2tcondition AS lum2tcondition
     , lum3.lum3tstate AS lum3tstate
     , lum3.lum3tcondition AS lum3tcondition

     , llm3.llm3tstate AS llm3tstate
     , llm3.llm3tcondition AS llm3tcondition
     , llm2.llm2tstate AS llm2tstate
     , llm2.llm2tcondition AS llm2tcondition
     , llm1.llm1tstate AS llm1tstate
     , llm1.llm1tcondition AS llm1tcondition
     , llp2.llp2tstate AS llp2tstate
     , llp2.llp2tcondition AS llp2tcondition
     , llp1.llp1tstate AS llp1tstate
     , llp1.llp1tcondition AS llp1tcondition
     , llc.llctstate AS llctstate
     , llc.llctcondition AS llctcondition
     , lli2.lli2tstate AS lli2tstate
     , lli2.lli2tcondition AS lli2tcondition
     , lli1.lli1tstate AS lli1tstate
     , lli1.lli1tcondition AS lli1tcondition
     , rli1.rli1tstate AS rli1tstate
     , rli1.rli1tcondition AS rli1tcondition
     , rli2.rli2tstate AS rli2tstate
     , rli2.rli2tcondition AS rli2tcondition
     , rlc.rlctstate AS rlctstate
     , rlc.rlctcondition AS rlctcondition
     , rlp1.rlp1tstate AS rlp1tstate
     , rlp1.rlp1tcondition AS rlp1tcondition
     , rlp2.rlp2tstate AS rlp2tstate
     , rlp2.rlp2tcondition AS rlp2tcondition
     , rlm1.rlm1tstate AS rlm1tstate
```

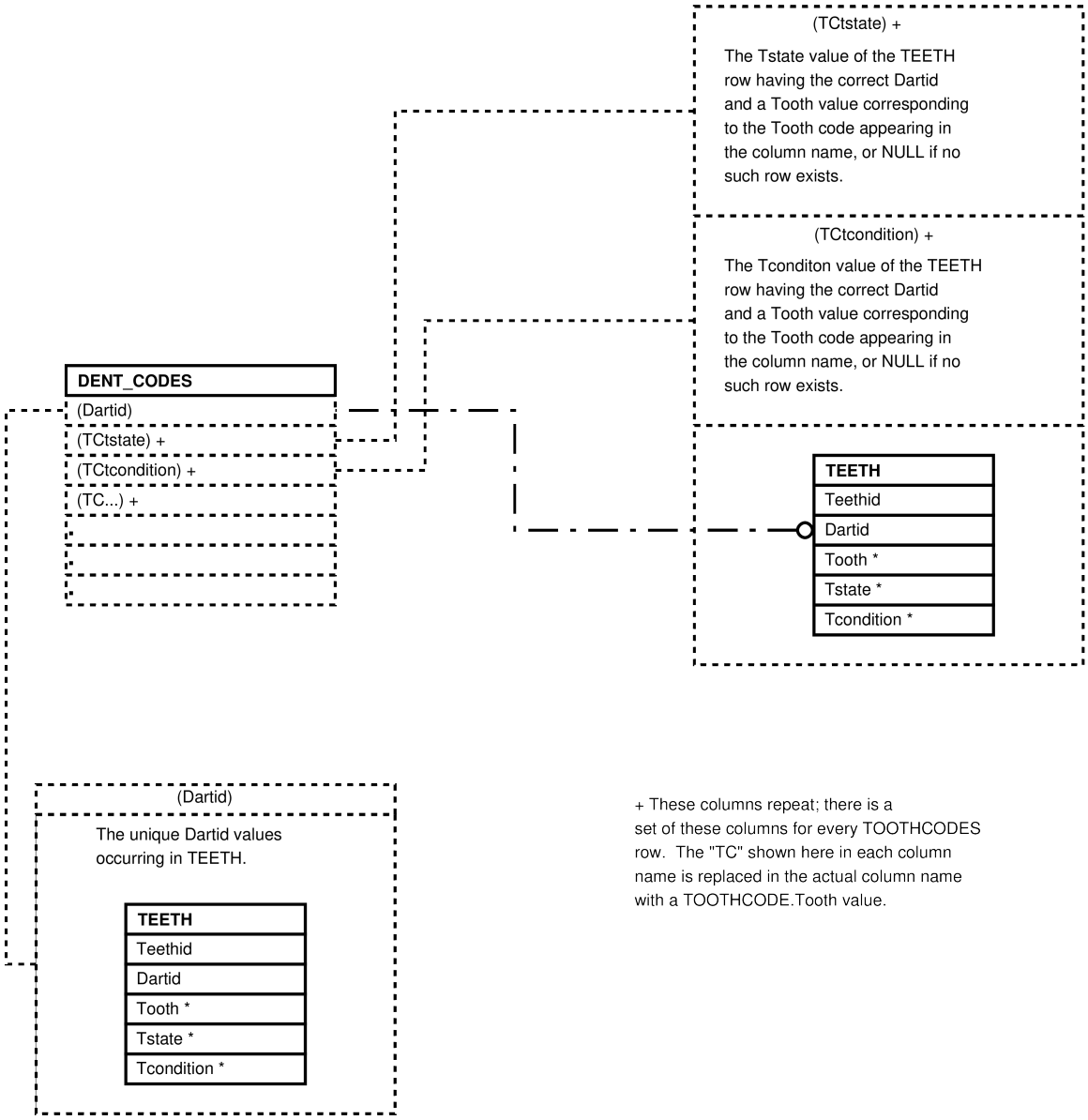

Figure 42: Entity Relationship Diagram of the DENT_CODES View

3.12 The DENT_SITES View

```
SELECT teethdartids.dartid AS dartid
     , s1.s1tstate AS s1tstate
     , s1.s1tcondition AS s1tcondition
     , s1.s1deciduous AS s1deciduous
     , s2.s2tstate AS s2tstate
     , s2.s2tcondition AS s2tcondition
     , s2.s2deciduous AS s2deciduous
     , s3.s3tstate AS s3tstate
     , s3.s3tcondition AS s3tcondition
     , s3.s3deciduous AS s3deciduous
     , s4.s4tstate AS s4tstate
     , s4.s4tcondition AS s4tcondition
     , s4.s4deciduous AS s4deciduous
     , s5.s5tstate AS s5tstate
     , s5.s5tcondition AS s5tcondition
     , s5.s5deciduous AS s5deciduous
     , s6.s6tstate AS s6tstate
     , s6.s6tcondition AS s6tcondition
     , s6.s6deciduous AS s6deciduous
     , s7.s7tstate AS s7tstate
     , s7.s7tcondition AS s7tcondition
     , s7.s7deciduous AS s7deciduous
     , s8.s8tstate AS s8tstate
     , s8.s8tcondition AS s8tcondition
     , s8.s8deciduous AS s8deciduous
     , s9.s9tstate AS s9tstate
     , s9.s9tcondition AS s9tcondition
     , s9.s9deciduous AS s9deciduous
     , s10.s10tstate AS s10tstate
     , s10.s10tcondition AS s10tcondition
     , s10.s10deciduous AS s10deciduous
     , s11.s11tstate AS s11tstate
     , s11.s11tcondition AS s11tcondition
     , s11.s11deciduous AS s11deciduous
     , s12.s12tstate AS s12tstate
     , s12.s12tcondition AS s12tcondition
     , s12.s12deciduous AS s12deciduous
     , s13.s13tstate AS s13tstate
     , s13.s13tcondition AS s13tcondition
     , s13.s13deciduous AS s13deciduous
     , s14.s14tstate AS s14tstate
     , s14.s14tcondition AS s14tcondition
     , s14.s14deciduous AS s14deciduous
     , s15.s15tstate AS s15tstate
     , s15.s15tcondition AS s15tcondition
     , s15.s15deciduous AS s15deciduous
     , s16.s16tstate AS s16tstate
     , s16.s16tcondition AS s16tcondition
     , s16.s16deciduous AS s16deciduous

     , s17.s17tstate AS s17tstate
     , s17.s17tcondition AS s17tcondition
     , s17.s17deciduous AS s17deciduous
     , s18.s18tstate AS s18tstate
     , s18.s18tcondition AS s18tcondition
     , s18.s18deciduous AS s18deciduous
     , s19.s19tstate AS s19tstate
     , s19.s19tcondition AS s19tcondition
     , s19.s19deciduous AS s19deciduous
     , s20.s20tstate AS s20tstate
     , s20.s20tcondition AS s20tcondition
```

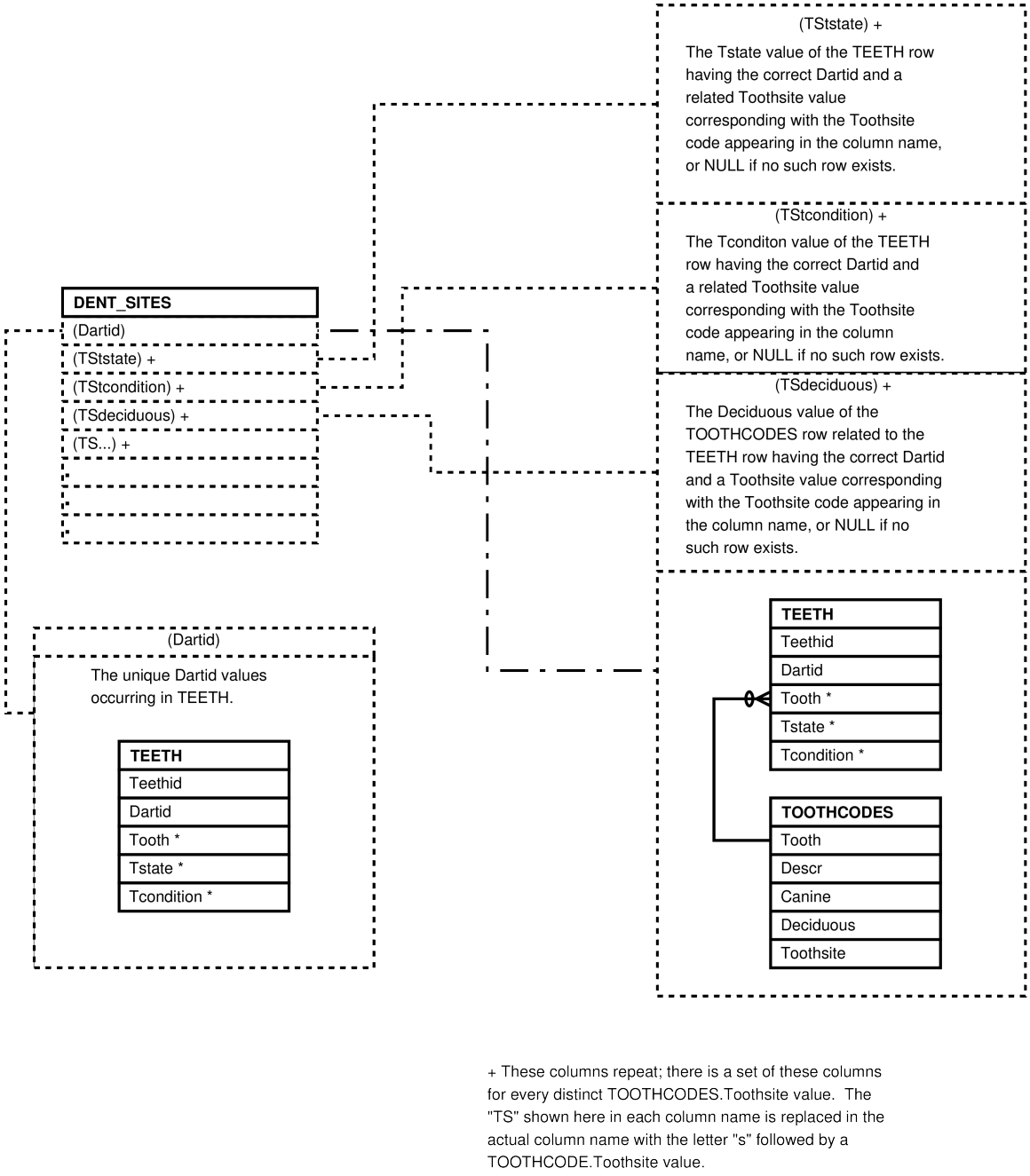

Figure 44: Entity Relationship Diagram of the DENT_SITES View

3.13 The ESTROGENS View

```
SELECT hormone_sample_data.tid
     , hormone_prep_series.hpsid
     , hormone_result_data.hrid
     , hormone_sample_data.hsid
     , biograph.sname
     , tissue_data.collection_date
     , tissue_data.collection_date_status AS collection_date_status
     , hormone_sample_data.fzdried_date AS fzdried_date
     , hormone_sample_data.sifted_date AS sifted_date
     , meoh_ext.procedure_date AS me_extracted
     , spe.procedure_date AS sp_extracted
     , hormone_result_data.raw_ng_g AS raw_ng_g
     , corrected_hormone(hormone_result_data.raw_ng_g, hormone_kits.correction) AS ↩
        corrected_ng_g
     , hormone_result_data.assay_date
     , hormone_kits.hormone AS hormone
     , hormone_result_data.kit AS kit
     , hormone_sample_data.comments AS sample_comments
     , hormone_result_data.comments AS result_comments
  FROM hormone_sample_data
  JOIN tissue_data
    ON tissue_data.tid = hormone_sample_data.tid
  JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON unique_indivs.popid = 1
       AND biograph.bioid::text = unique_indivs.individ
  JOIN hormone_prep_series
    ON hormone_prep_series.tid = hormone_sample_data.tid
  JOIN hormone_result_data
    ON hormone_result_data.hpsid = hormone_prep_series.hpsid
  JOIN hormone_kits
    ON hormone_kits.kit = hormone_result_data.kit
       AND hormone_kits.correction IS NOT NULL
       AND hormone_kits.hormone = 'E'
  LEFT JOIN hormone_prep_data AS meoh_ext
    ON meoh_ext.procedure = 'MEOH_EXT'
       AND meoh_ext.hpsid = hormone_prep_series.hpsid
  LEFT JOIN hormone_prep_data AS spe
    ON spe.procedure = 'SPE'
       AND spe.hpsid = hormone_prep_series.hpsid;
```

Figure 45: Query Defining the ESTROGENS View

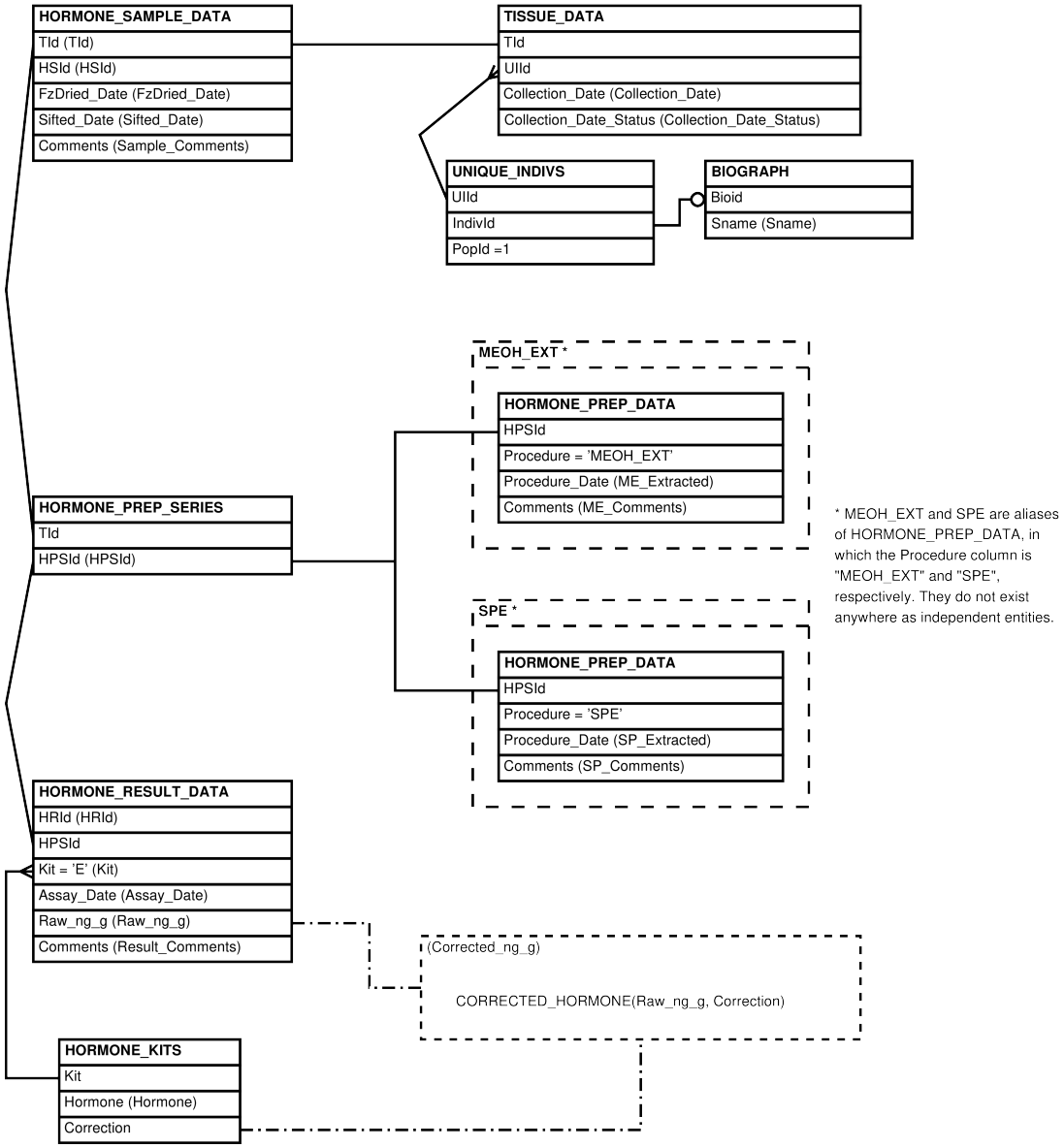

Figure 46: Entity Relationship Diagram of the ESTROGENS View

3.14 The GLUCOCORTICOIDS View

```
SELECT hormone_sample_data.tid
     , hormone_prep_series.hpsid
     , hormone_result_data.hrid
     , hormone_sample_data.hsid
     , biograph.sname
     , tissue_data.collection_date
     , tissue_data.collection_date_status AS collection_date_status
     , hormone_sample_data.fzdried_date AS fzdried_date
     , hormone_sample_data.sifted_date AS sifted_date
     , meoh_ext.procedure_date AS me_extracted
     , spe.procedure_date AS sp_extracted
     , hormone_result_data.raw_ng_g AS raw_ng_g
     , corrected_hormone(hormone_result_data.raw_ng_g, hormone_kits.correction) AS ↩
        corrected_ng_g
     , hormone_result_data.assay_date
     , hormone_kits.hormone AS hormone
     , hormone_result_data.kit AS kit
     , hormone_sample_data.comments AS sample_comments
     , hormone_result_data.comments AS result_comments
  FROM hormone_sample_data
  JOIN tissue_data
    ON tissue_data.tid = hormone_sample_data.tid
  JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON unique_indivs.popid = 1
       AND biograph.bioid::text = unique_indivs.individ
  JOIN hormone_prep_series
    ON hormone_prep_series.tid = hormone_sample_data.tid
  JOIN hormone_result_data
    ON hormone_result_data.hpsid = hormone_prep_series.hpsid
  JOIN hormone_kits
    ON hormone_kits.kit = hormone_result_data.kit
       AND hormone_kits.correction IS NOT NULL
       AND hormone_kits.hormone = 'GC'
  LEFT JOIN hormone_prep_data AS meoh_ext
    ON meoh_ext.procedure = 'MEOH_EXT'
       AND meoh_ext.hpsid = hormone_prep_series.hpsid
  LEFT JOIN hormone_prep_data AS spe
    ON spe.procedure = 'SPE'
       AND spe.hpsid = hormone_prep_series.hpsid;
```

Figure 47: Query Defining the GLUCOCORTICOIDS View

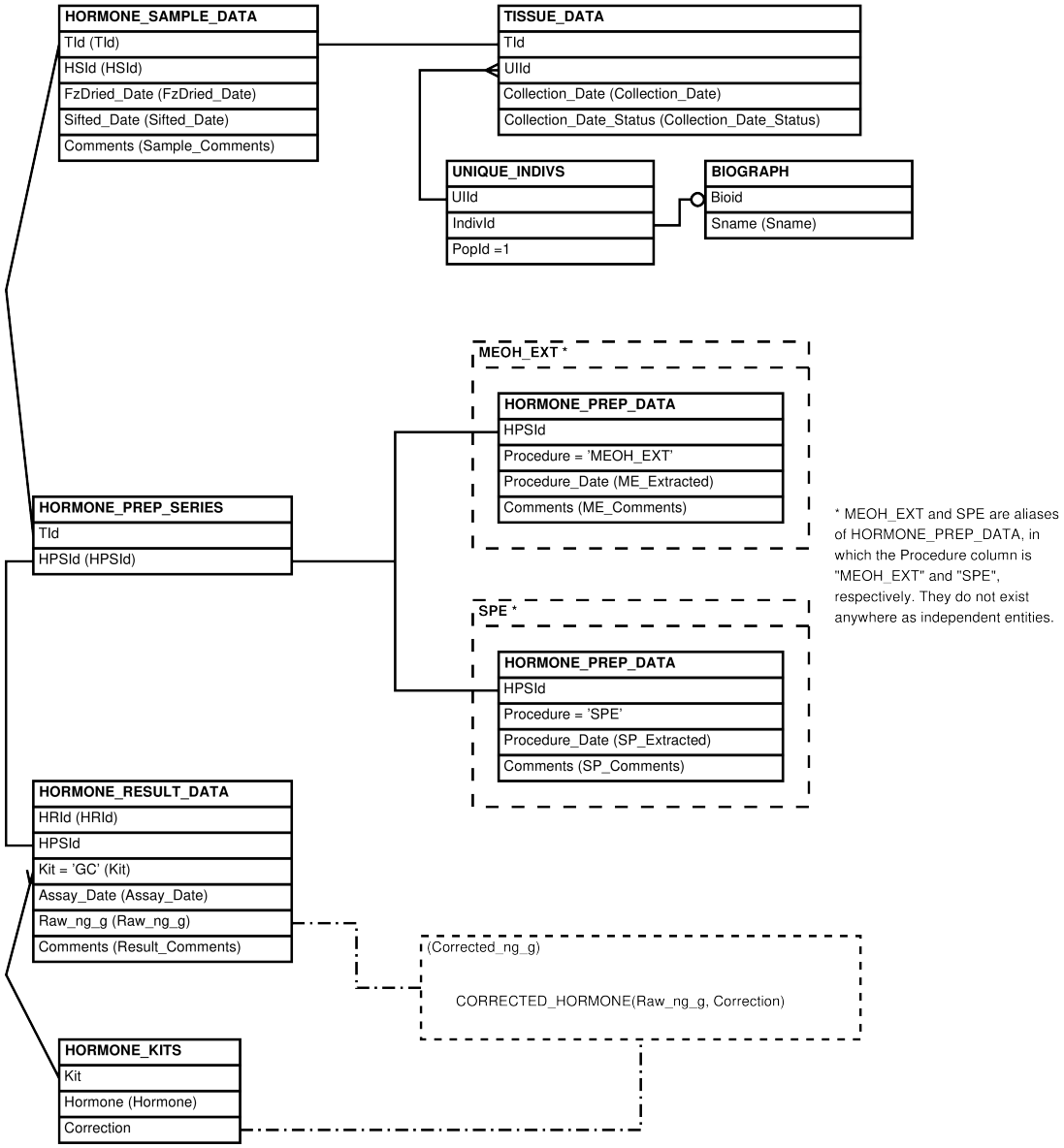

Figure 48: Entity Relationship Diagram of the GLUCOCORTICOIDS View

3.15 The GROUPS_HISTORY View

```
SELECT groups.gid AS gid
    ,  groups.name AS name
    ,  groups.from_group AS from_group
    ,  groups.to_group AS to_group
    ,  CASE
         WHEN groups.from_group IS NULL
              AND NOT EXISTS (SELECT 1
                              FROM groups AS from_groups
                              WHERE from_groups.to_group = groups.gid)
           THEN groups.permanent
         ELSE groups.start
       END AS first_observed
    ,  CASE
         WHEN groups.study_grp IS NULL
           THEN NULL
         WHEN groups.from_group IS NULL
              AND NOT EXISTS (SELECT 1
                              FROM groups AS from_groups
                              WHERE from_groups.to_group = groups.gid)
           THEN groups.permanent
         ELSE (SELECT date
                 FROM census
                 WHERE census.grp = groups.gid
                   AND census.cen
                 ORDER BY date
                 LIMIT 1)
       END AS first_study_grp_census
    ,  groups.permanent AS permanent
    ,  (SELECT descgroups_start.start
           FROM babase.groups AS descgroups_start
           WHERE descgroups_start.from_group = groups.gid
              OR descgroups_start.gid = groups.to_group
           ORDER BY descgroups_start.start
           LIMIT 1
       ) AS impermanent
    ,  groups.cease_to_exist AS cease_to_exist
    ,  groups.last_reg_census AS last_reg_census
    ,  groups.study_grp
  FROM babase.groups;
```

Figure 49: Query Defining the GROUPS_HISTORY View

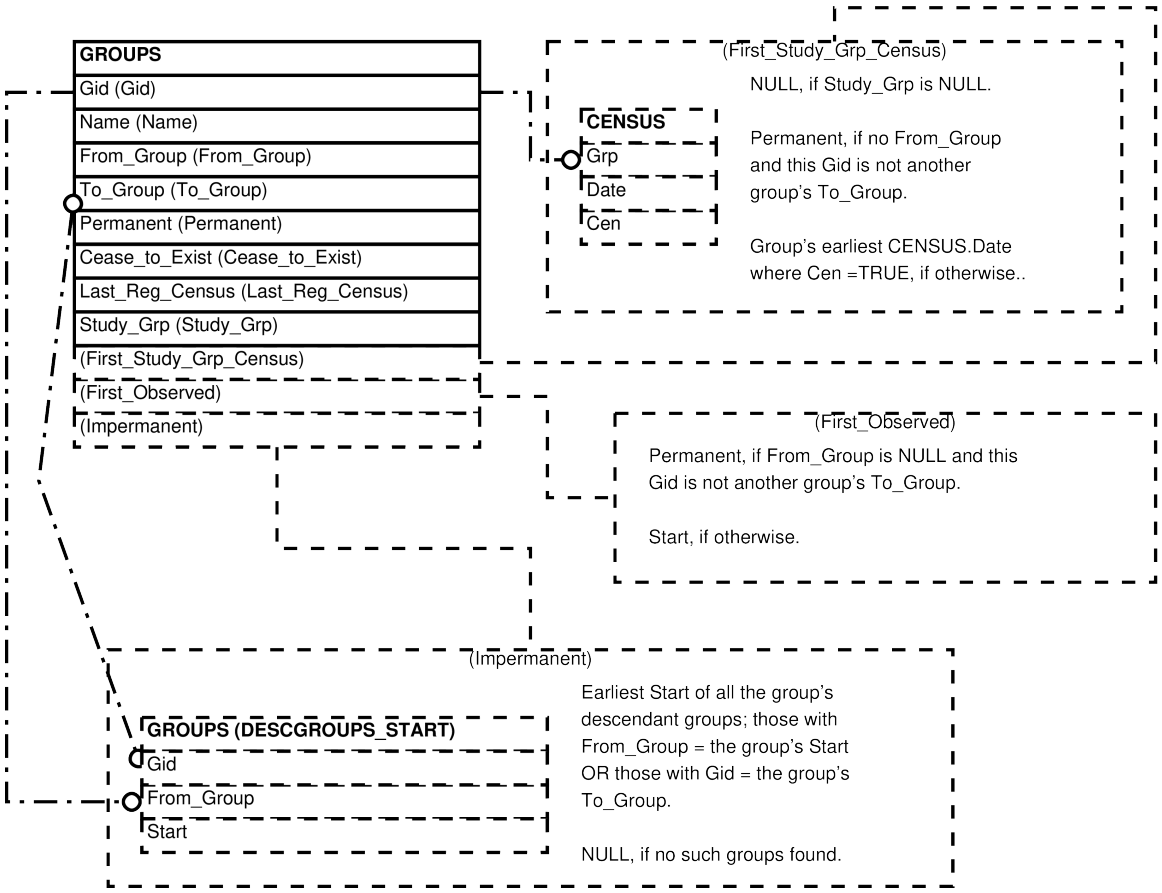

Figure 50: Entity Relationship Diagram of the GROUPS_HISTORY View

3.16 The HORMONE_PREPS View

```
SELECT hormone_sample_data.tid AS tid
     , hormone_sample_data.hsid AS hsid
     , unique_indivs.individ AS individ
     , biograph.sname AS sname
     , hormone_prep_series.hpsid AS hpsid
     , hormone_prep_series.series AS series
     , hormone_prep_data.hpid AS hpid
     , hormone_prep_data.procedure AS procedure
     , hormone_prep_data.procedure_date AS procedure_date
     , hormone_prep_data.comments AS comments
  FROM hormone_sample_data
  JOIN tissue_data
    ON tissue_data.tid = hormone_sample_data.tid
  JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON unique_indivs.popid = 1
       AND biograph.bioid::text = unique_indivs.individ
  JOIN hormone_prep_series
    ON hormone_prep_series.tid = hormone_sample_data.tid
  JOIN hormone_prep_data
    ON hormone_prep_data.hpsid = hormone_prep_series.hpsid;
```

Figure 51: Query Defining the HORMONE_PREPS View

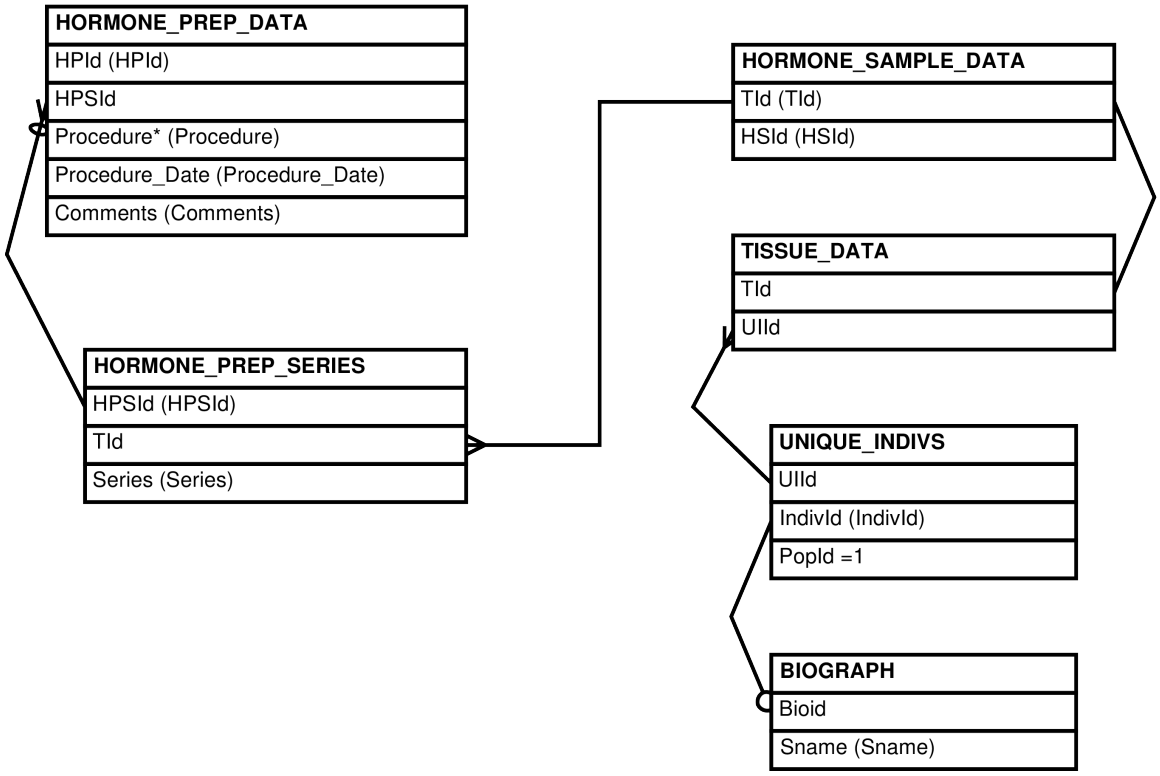

Figure 52: Entity Relationship Diagram of the HORMONE_PREPS View

3.17 The HORMONE_RESULTS View

```
SELECT hormone_sample_data.tid AS tid
     , hormone_sample_data.hsid AS hsid
     , unique_indivs.individ AS individ
     , biograph.sname AS sname
     , hormone_prep_series.hpsid AS hpsid
     , hormone_prep_series.series AS series
     , hormone_result_data.hrid AS hrid
     , hormone_kits.hormone AS hormone
     , hormone_result_data.kit AS kit
     , hormone_result_data.assay_date AS assay_date
     , hormone_result_data.grams_used AS grams_used
     , hormone_result_data.raw_ng_g AS raw_ng_g
     , corrected_hormone(hormone_result_data.raw_ng_g, hormone_kits.correction) AS ↩
         corrected_ng_g
     , hormone_result_data.comments AS comments
  FROM hormone_sample_data
  JOIN tissue_data
    ON tissue_data.tid = hormone_sample_data.tid
  JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON unique_indivs.popid = 1
       AND biograph.bioid::text = unique_indivs.individ
  JOIN hormone_prep_series
    ON hormone_prep_series.tid = hormone_sample_data.tid
  JOIN hormone_result_data
    ON hormone_result_data.hpsid = hormone_prep_series.hpsid
  JOIN hormone_kits
    ON hormone_kits.kit = hormone_result_data.kit;
```

Figure 53: Query Defining the HORMONE_RESULTS View

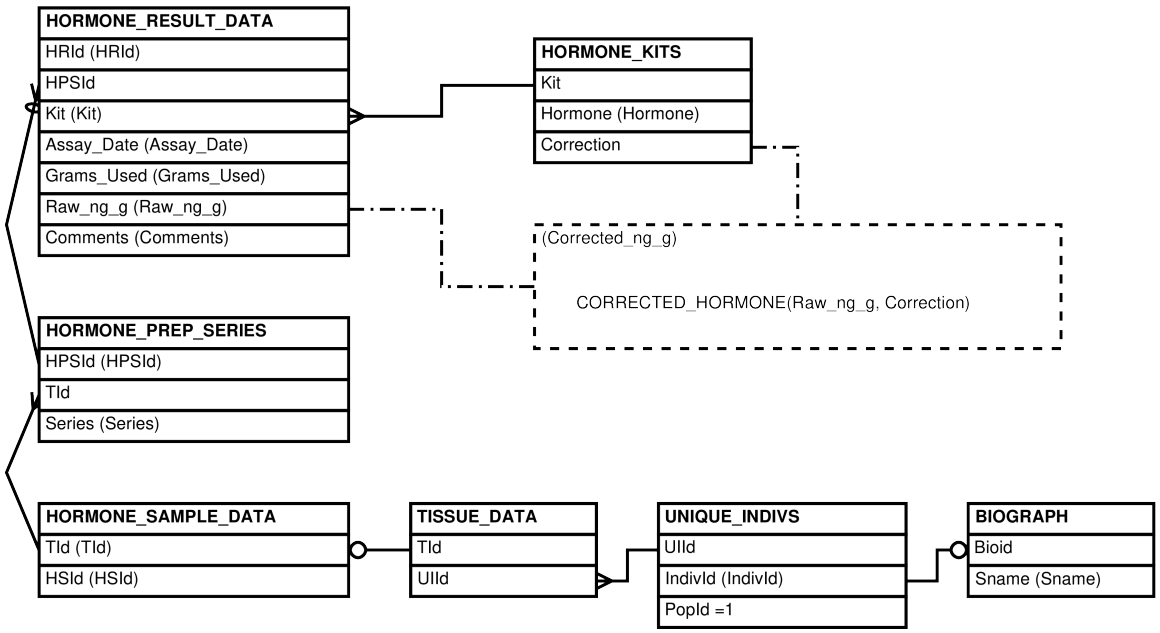

Figure 54: Entity Relationship Diagram of the HORMONE_RESULTS View

3.18 The HORMONE_SAMPLES View

```
SELECT hormone_sample_data.tid AS tid
     , hormone_sample_data.hsid AS hsid
     , unique_indivs.individ AS individ
     , biograph.sname AS sname
     , tissue_data.collection_date AS collection_date
     , tissue_data.collection_date_status AS collection_date_status
     , hormone_sample_data.fzdried_date AS fzdried_date
     , hormone_sample_data.sifted_date AS sifted_date
     , hormone_sample_data.avail_mass_g AS avail_mass_g
     , hormone_sample_data.avail_date AS avail_date
     , hormone_sample_data.comments AS comments
  FROM hormone_sample_data
  JOIN tissue_data
    ON tissue_data.tid = hormone_sample_data.tid
  JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON unique_indivs.popid = 1
       AND biograph.bioid::text = unique_indivs.individ;
```

Figure 55: Query Defining the HORMONE_SAMPLES View

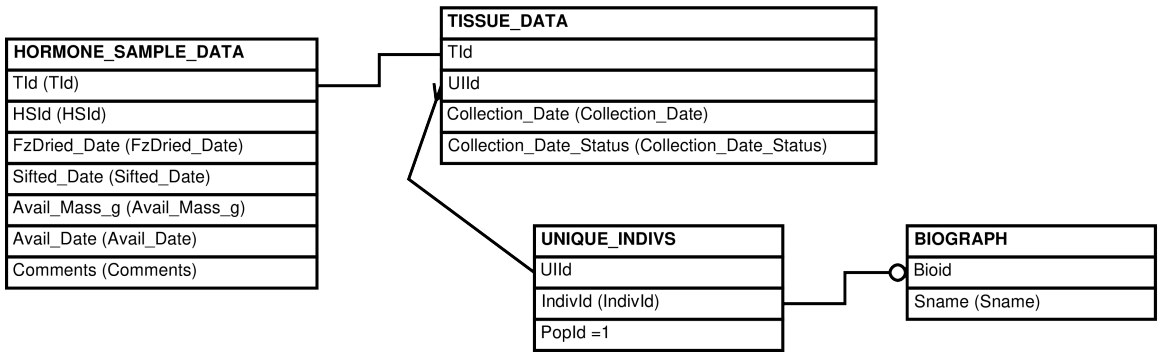

Figure 56: Entity Relationship Diagram of the HORMONE_SAMPLES View

3.19 The HUMERUS_STATS View

```
SELECT humeruses.dartid AS dartid
     , count(*) AS husamps
     , avg(humeruses.hulength) AS hulength_mean
     , stddev(humeruses.hulength) AS hulength_stddev
     , avg(humeruses.huunadjusted) AS huunadjusted_mean
     , stddev(humeruses.huunadjusted) AS huunadjusted_stddev
  FROM humeruses
  GROUP BY humeruses.dartid;
```

Figure 57: Query Defining the HUMERUS_STATS View

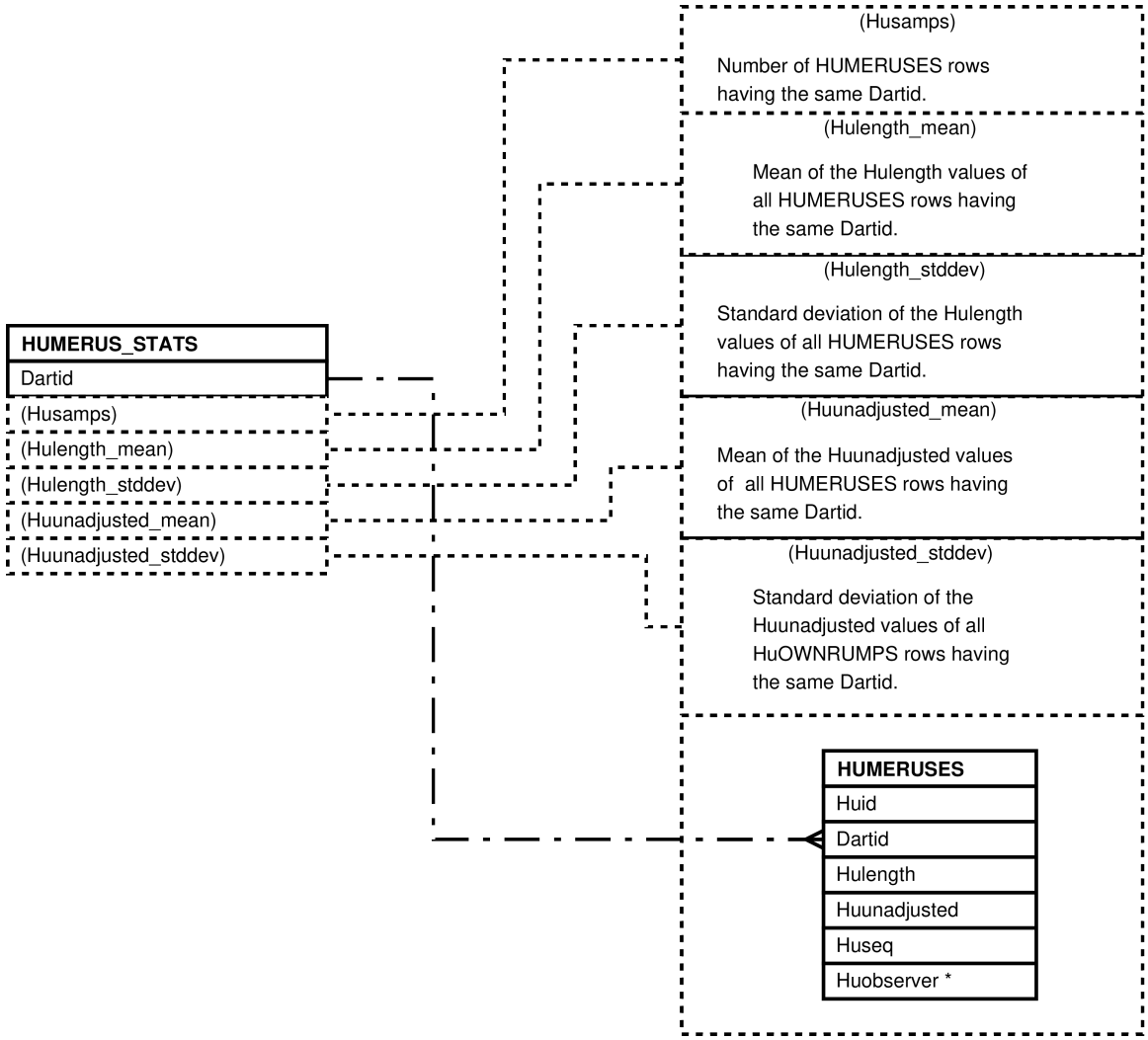

Figure 58: Entity Relationship Diagram of the HUMERUS_STATS View

3.20 The INTERACT and INTERACT_SORTED Views

```
SELECT iid AS iid
     , interact_data.sid AS sid
     , interact_data.act AS act
     , acts.class AS class
     , interact_data.date AS date
     , julian(interact_data.date) AS jdate
     , interact_data.start AS start
     , spm(interact_data.start) AS startspm
     , stop AS stop
     , spm(interact_data.stop) AS stopspm
     , interact_data.observer AS observer
     , interact_data.handwritten AS handwritten
     , interact_data.exact_date AS exact_date
  FROM interact_data
       JOIN acts
            ON (acts.act = interact_data.act);
```

Figure 59: Query Defining the INTERACT View

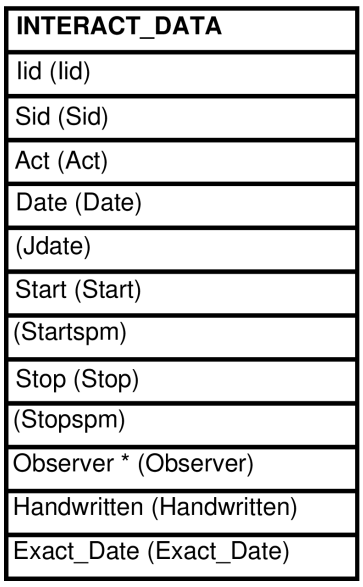

Figure 60: Entity Relationship Diagram of the INTERACT View

3.21 The LOCATIONS_FREE View

```
SELECT locations.locid AS locid
     , locations.institution AS institution
     , locations.location AS location
     , locations.is_unique AS is_unique
  FROM locations
  WHERE NOT EXISTS (SELECT 1
                    FROM tissue_data
                    WHERE tissue_data.locid = locations.locid)
    AND NOT EXISTS (SELECT 1
                    FROM nucacid_data
                    WHERE nucacid_data.locid = locations.locid);
```

Figure 61: Query Defining the LOCATIONS_FREE View

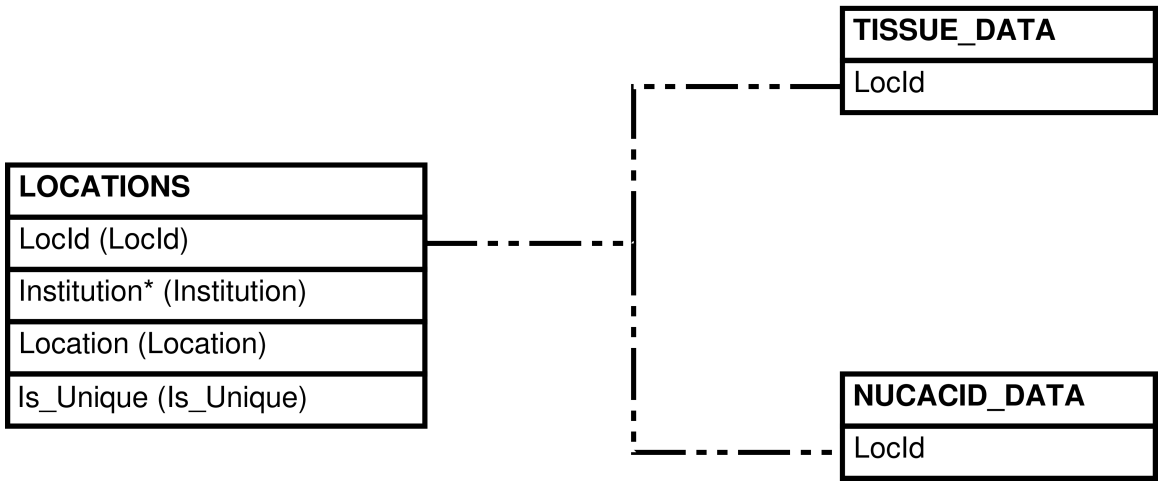

Figure 62: Entity Relationship Diagram of the LOCATIONS_FREE View

3.22 The MATERNITIES View

```
SELECT cycles.sname AS mom
     , cycles.cid AS cid
     , cycles.seq AS seq
     , cycles.series AS series
     , cycpoints.cpid AS conceive
     , cycpoints.date AS zdate
     , members.grp AS zdate_grp
     , cycpoints.edate AS edate
     , cycpoints.ldate AS ldate
     , cycpoints.source AS source
     , pregs.pid AS pid
     , pregs.parity AS parity
     , biograph.bioid AS child_bioid
     , biograph.sname AS child
     , biograph.birth AS birth
  FROM cycles
       JOIN cycpoints ON (cycpoints.cid = cycles.cid)
       JOIN members ON (members.date = cycpoints.date
                        AND members.sname = cycles.sname)
       JOIN pregs ON (pregs.conceive = cycpoints.cpid)
       JOIN biograph ON (pregs.pid = biograph.pid);
```

Figure 63: Query Defining the MATERNITIES View

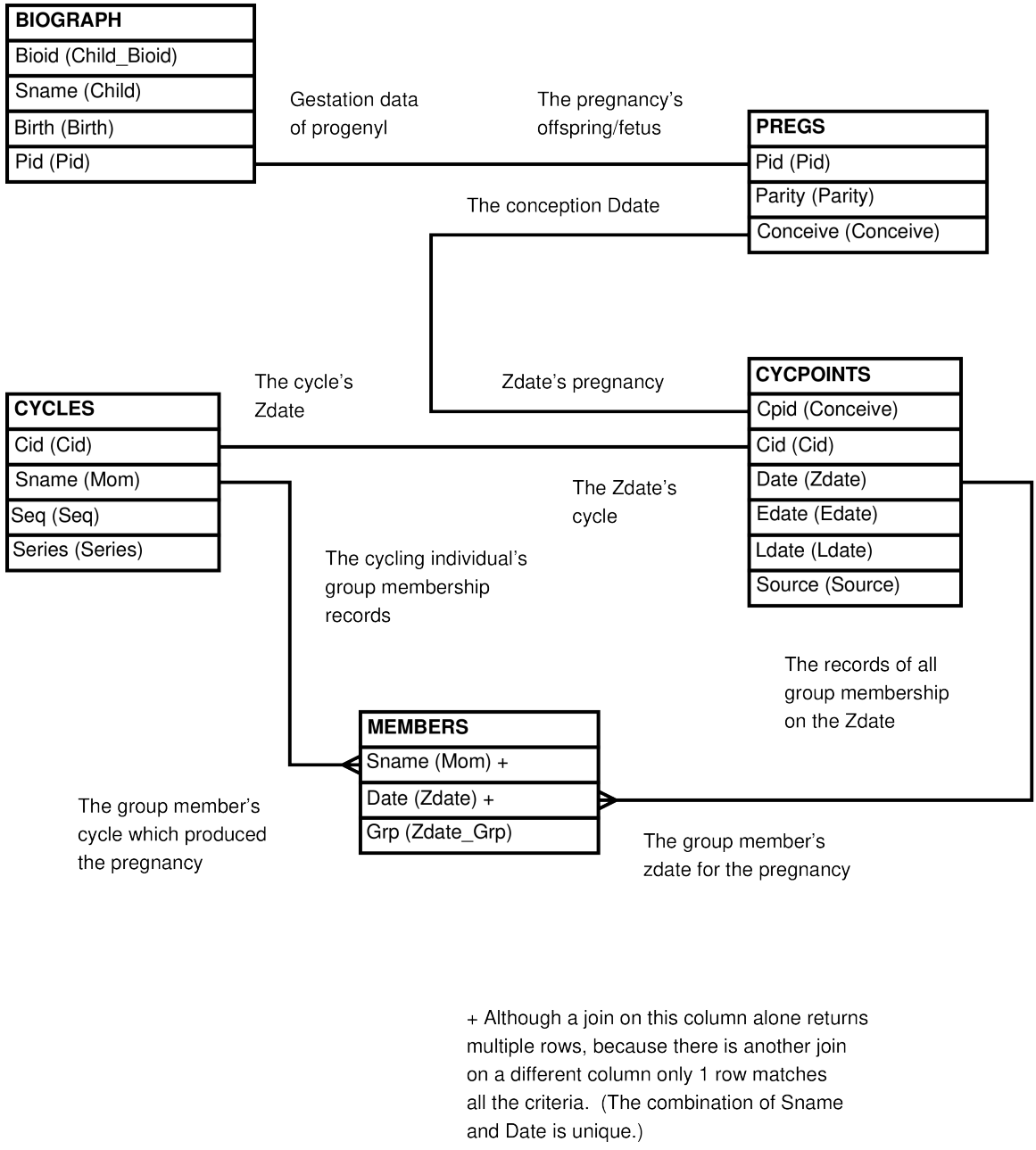

Figure 64: Entity Relationship Diagram of the MATERNITIES View

3.23 The MIN_MAXS View

```
SELECT wreadings.wrid AS wrid
     , wreadings.wstation AS wstation
     , wreadings.wrdaytime AS wrdaytime
     , wreadings.estdaytime AS estdaytime
     , wreadings.wrperson AS wrperson
     , wreadings.wrnotes AS wrnotes
     , tempmins.tempmin AS tempmin
     , tempmaxs.tempmax AS tempmax
     , raingauges.rgspan AS rgspan
     , raingauges.estrgspan AS estrgspan
     , raingauges.rain AS rain
  FROM wreadings
       LEFT OUTER JOIN tempmins
            ON wreadings.wrid = tempmins.wrid
       LEFT OUTER JOIN tempmaxs
            ON wreadings.wrid = tempmaxs.wrid
       LEFT OUTER JOIN raingauges
            ON wreadings.wrid = raingauges.wrid;
```

Figure 65: Query Defining the MIN_MAXS View

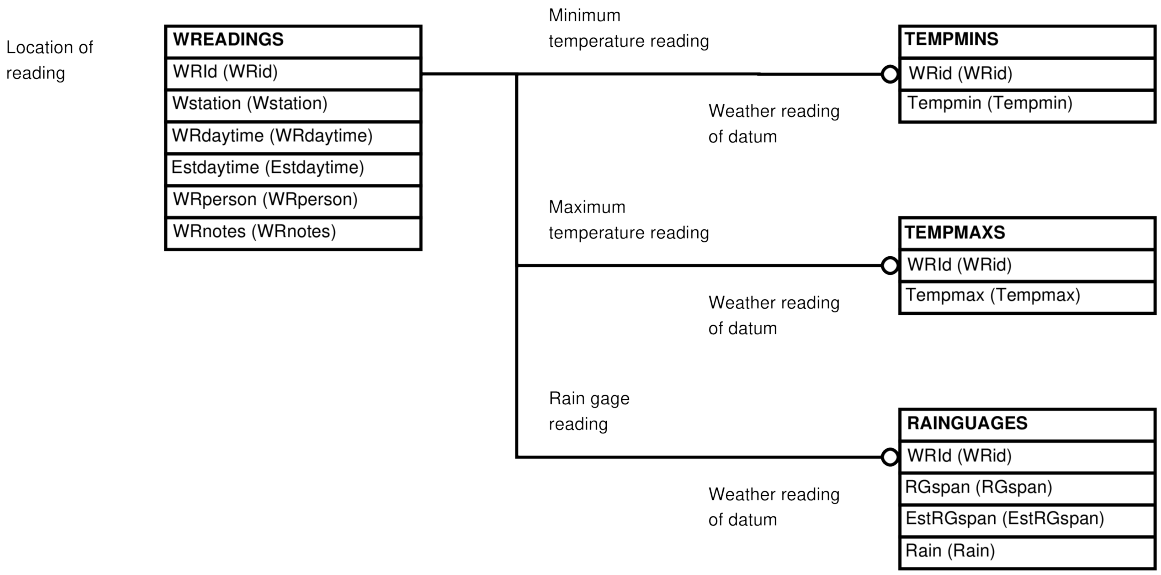

Figure 66: Entity Relationship Diagram of the MIN_MAXS View

3.24 The MIN_MAXS_SORTED View

```
SELECT wreadings.wrid AS wrid
     , wreadings.wstation AS wstation
     , wreadings.wrdaytime AS wrdaytime
     , wreadings.estdaytime AS estdaytime
     , wreadings.wrperson AS wrperson
     , wreadings.wrnotes AS wrnotes
     , tempmins.tempmin AS tempmin
     , tempmaxs.tempmax AS tempmax
     , raingauges.rgspan AS rgspan
     , raingauges.estrgspan AS estrgspan
     , raingauges.rain AS rain
  FROM wreadings
       LEFT OUTER JOIN tempmins
            ON wreadings.wrid = tempmins.wrid
       LEFT OUTER JOIN tempmaxs
            ON wreadings.wrid = tempmaxs.wrid
       LEFT OUTER JOIN raingauges
            ON wreadings.wrid = raingauges.wrid
  ORDER BY wreadings.wrdaytime, wreadings.wstation;;
```

Figure 67: Query Defining the MIN_MAXS_SORTED View

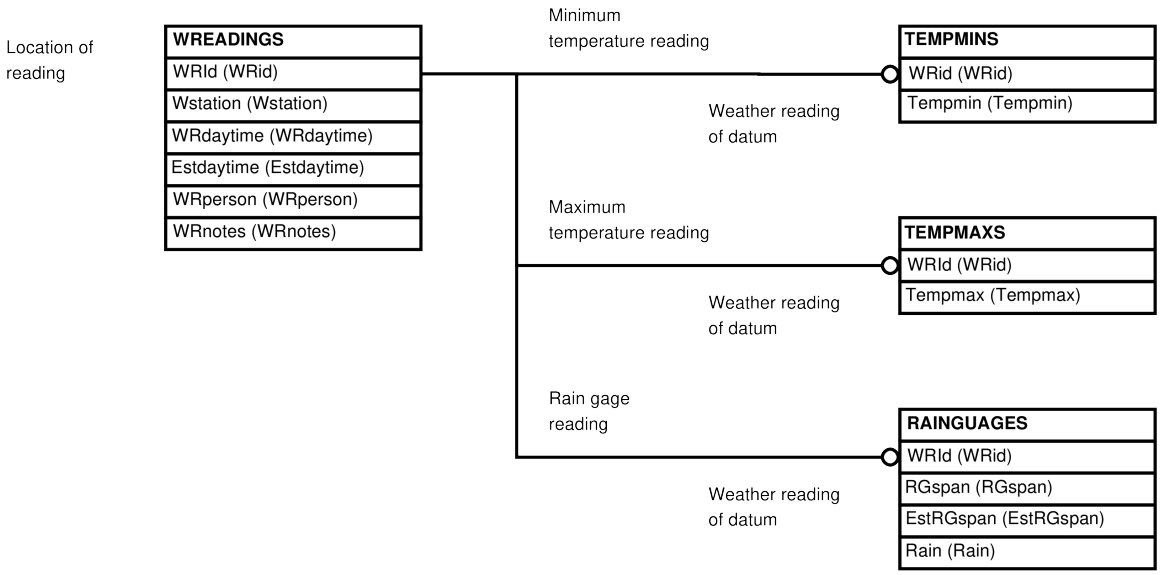

Figure 68: Entity Relationship Diagram of the MIN_MAXS_SORTED View

3.25 The MPI_EVENTS View

```
SELECT mpis.mpiid AS mpiid
     , mpis.date AS date
     , mpis.observer AS observer
     , mpis.context_type AS context_type
     , mpis.context AS context
     , mpi_data.mpidid AS mpidid
     , mpi_data.seq AS seq
     , mpi_data.mpiact AS mpiact
     , actor.mpipid AS actorid
     , actor.sname AS actor
     , actor.unksname AS unkactor
     , actee.mpipid AS acteeid
     , actee.sname AS actee
     , actee.unksname AS unkactee
     , CASE WHEN EXISTS(SELECT 1
                          FROM mpiacts
                          WHERE mpiacts.mpiact = mpi_data.mpiact
                                AND mpiacts.kind = 'H')
              THEN
              EXISTS(SELECT 1
                FROM mpi_data AS request
                   , mpiacts
                   , mpi_parts AS requestor
                   , mpi_parts AS requestee
                WHERE request.mpiid = mpi_data.mpiid
                      AND request.seq < mpi_data.seq
                      AND mpiacts.mpiact = request.mpiact
                      AND mpiacts.kind = 'R'
                      AND requestor.mpidid = request.mpidid
                      AND requestor.role = 'R'
                      AND requestor.sname = actee.sname
                      AND requestee.mpidid = request.mpidid
                      AND requestee.role = 'E'
                      AND requestee.sname = actor.sname)
            ELSE
              NULL
       END AS solicited
     , EXISTS(SELECT 1
         FROM mpi_data AS initial,
              mpiacts
         WHERE initial.mpiid = mpi_data.mpiid
               AND initial.seq = 1
               AND mpiacts.mpiact = initial.mpiact
               AND mpiacts.decided)
       AS decided
     , mpi_data.helped AS helped
     , mpi_data.active AS active
  FROM mpis
       LEFT OUTER JOIN mpi_data ON (mpis.mpiid = mpi_data.mpiid)
       LEFT OUTER JOIN mpi_parts AS actor
            ON (actor.mpidid = mpi_data.mpidid AND actor.role = 'R')
       LEFT OUTER JOIN mpi_parts AS actee
            ON (actee.mpidid = mpi_data.mpidid AND actee.role = 'E');
```

Figure 69: Query Defining the MPI_EVENTS View

If we could we would display here the diagram showing how the MPI_EVENTS view is constructed.

Figure 70: Entity Relationship Diagram of the MPI_EVENTS View

3.26 The MTD_CYCLES View

```
SELECT cycles.cid AS cid
     , cycles.sname AS sname
     , cycles.seq AS seq
     , cycles.series AS series
     , mcp.cpid AS mcpid
     , mcp.date AS mdate
     , mcp.edate AS emdate
     , mcp.ldate AS lmdate
     , mcp.source AS msource
     , tcp.cpid AS tcpid
     , tcp.date AS tdate
     , tcp.edate AS etdate
     , tcp.ldate AS ltdate
     , tcp.source AS tsource
     , dcp.cpid AS dcpid
     , dcp.date AS ddate
     , dcp.edate AS eddate
     , dcp.ldate AS lddate
     , dcp.source AS dsource
  FROM cycles
   LEFT OUTER JOIN cycpoints AS mcp ON (mcp.cid = cycles.cid
                                        AND mcp.code = 'M')
   LEFT OUTER JOIN cycpoints AS tcp ON (tcp.cid = cycles.cid
                                        AND tcp.code = 'T')
   LEFT OUTER JOIN cycpoints AS dcp ON (dcp.cid = cycles.cid
                                        AND dcp.code = 'D')
  ORDER BY cycles.sname, cycles.seq;
```

Figure 71: Query Defining the MTD_CYCLES View

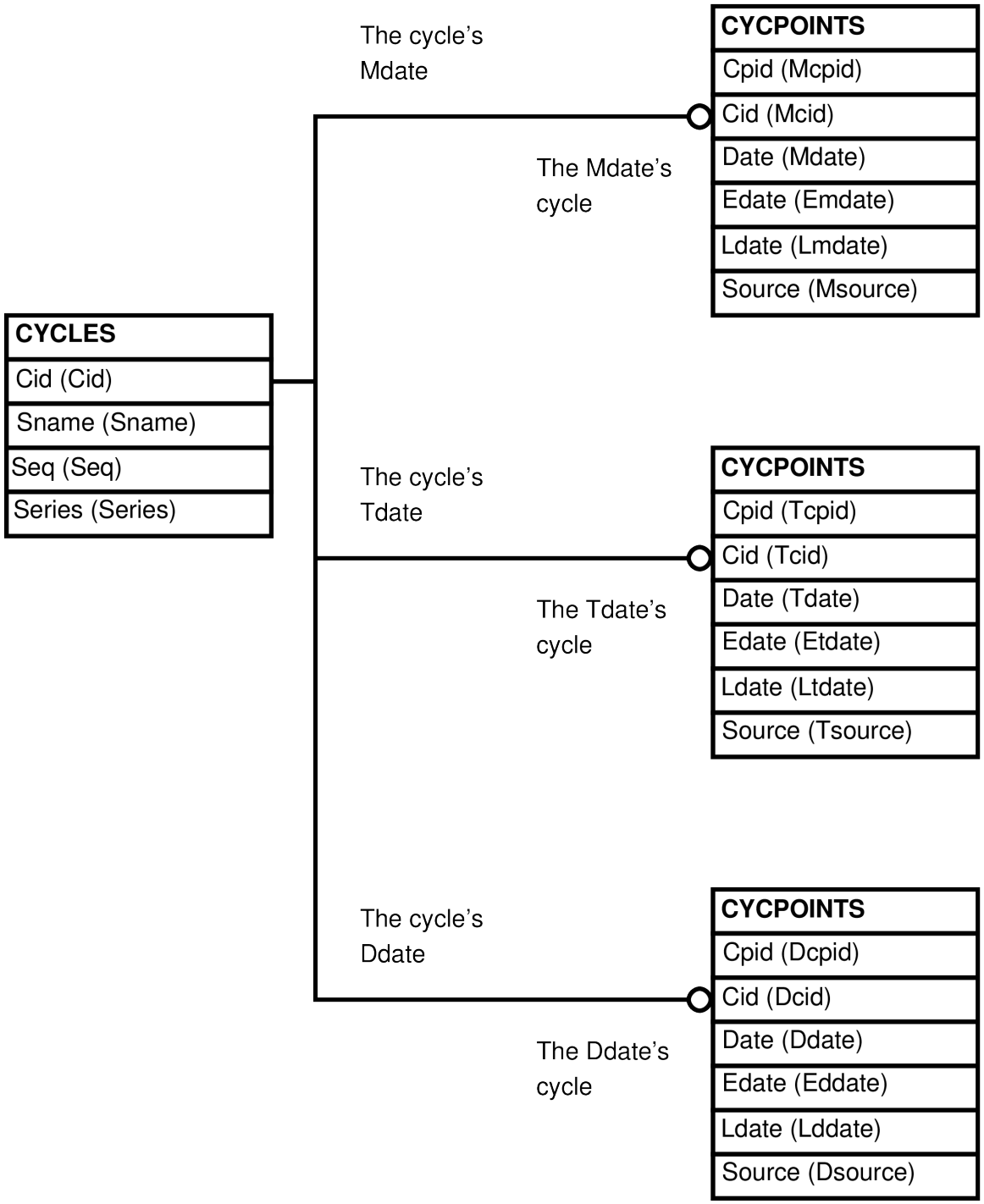

Figure 72: Entity Relationship Diagram of the MTD_CYCLES View

3.27 The NUCACID_CONCS View

```
SELECT nucacid_conc_data.nacid AS nacid
     , nucacid_conc_data.naid AS naid
     , local_1.localid AS localid_1
     , local_2.localid AS localid_2
     , nucacid_conc_data.conc_method AS conc_method
     , nucacid_conc_methods.descr AS method_descr
     , nucacid_conc_data.conc_date AS conc_date
     , nucacid_conc_data.pg_ul AS pg_ul
     , (nucacid_conc_data.pg_ul / 1000)::numeric(10,4) AS ng_ul
  FROM nucacid_conc_data
  JOIN nucacid_conc_methods
    ON nucacid_conc_methods.conc_method = nucacid_conc_data.conc_method
  LEFT JOIN nucacid_local_ids AS local_1
    ON local_1.naid = nucacid_conc_data.naid
       AND local_1.institution = 1
  LEFT JOIN nucacid_local_ids AS local_2
    ON local_2.naid = nucacid_conc_data.naid
       AND local_2.institution = 2;
```

Figure 73: Query Defining the NUCACID_CONCS View

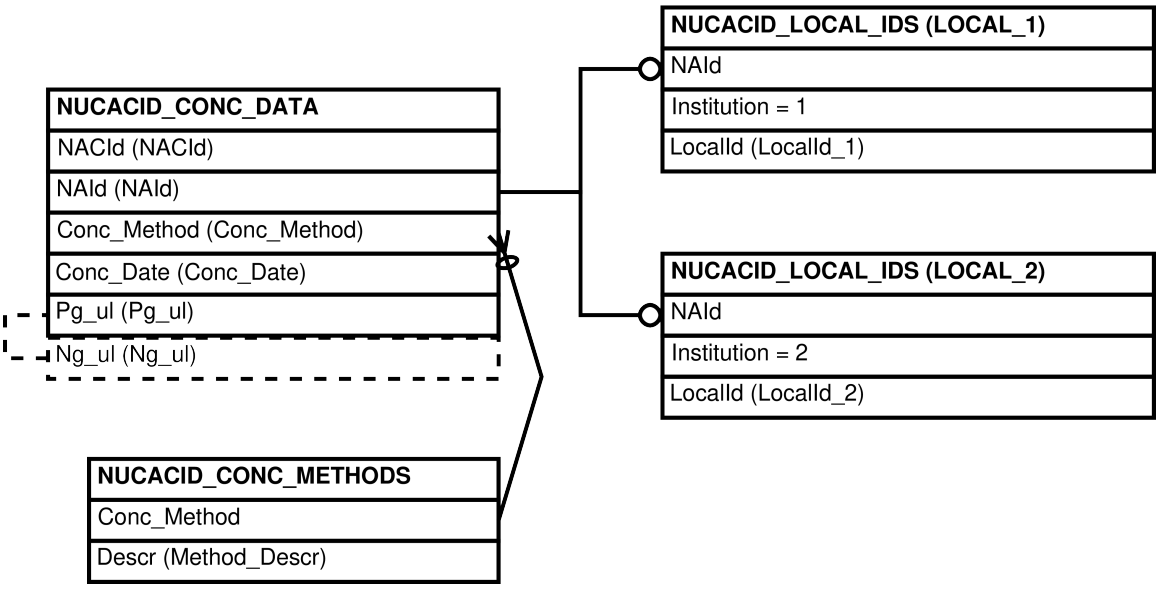

Figure 74: Entity Relationship Diagram of the NUCACID_CONCS View

3.28 The NUCACIDS View

```
WITH concat_creators AS (SELECT naid
                              , string_agg(creator, '/' ORDER BY naid, nacrid) AS ↩
                                 created_by
                           FROM nucacid_creators
                           GROUP BY naid)
SELECT nucacid_data.naid AS naid
     , nucacid_data.tid AS tid
     , nucacid_data.locid AS locid
     , locations.institution AS institution
     , locations.location AS location
     , local_1.localid AS localid_1
     , local_2.localid AS localid_2
     , nucacid_data.uiid AS uiid
     , unique_indivs.popid AS popid
     , unique_indivs.individ AS individ
     , biograph.sname AS sname
     , nucacid_data.name_on_tube AS name_on_tube
     , nucacid_data.nucacid_type AS nucacid_type
     , tissue_data.tissue_type AS tissue_type
     , nucacid_data.creation_date AS creation_date
     , concat_creators.created_by AS created_by
     , nucacid_data.creation_method AS creation_method
     , COUNT(nucacid_sources.*) AS na_sources
     , nucacid_data.initial_vol_ul AS initial_vol_ul
     , nucacid_data.actual_vol_ul AS actual_vol_ul
     , nucacid_data.actual_vol_date AS actual_vol_date
     , nucacid_data.multi_indivs AS multi_indivs
     , nucacid_data.multi_tids AS multi_tids
     , nucacid_data.notes AS notes
  FROM nucacid_data
  JOIN locations
    ON locations.locid = nucacid_data.locid
  LEFT JOIN tissue_data
    ON tissue_data.tid = nucacid_data.tid
  LEFT JOIN unique_indivs
    ON unique_indivs.uiid = nucacid_data.uiid
  LEFT JOIN biograph
    ON biograph.bioid::text = unique_indivs.individ
       AND unique_indivs.popid = 1
  LEFT JOIN nucacid_local_ids AS local_1
    ON local_1.naid = nucacid_data.naid
       AND local_1.institution = 1
  LEFT JOIN nucacid_local_ids AS local_2
    ON local_2.naid = nucacid_data.naid
       AND local_2.institution = 2
  LEFT JOIN nucacid_sources
    ON nucacid_sources.naid = nucacid_data.naid
  LEFT JOIN concat_creators
    ON concat_creators.naid = nucacid_data.naid
  GROUP BY nucacid_data.naid
         , nucacid_data.tid
         , nucacid_data.locid
         , locations.institution
         , locations.location
         , local_1.localid
         , local_2.localid
         , nucacid_data.uiid
         , unique_indivs.popid
         , unique_indivs.individ
         , biograph.sname
         , nucacid_data.name_on_tube
```

If we could we would display here a diagram showing how the NUCACIDS view is constructed.

Figure 76: Entity Relationship Diagram of the NUCACIDS View

3.29 The NUCACIDS_W_CONC View

```
WITH last_quants AS (SELECT DISTINCT
                            naid
                          , conc_method
                          , last_value(pg_ul) OVER w AS last_pg_ul
                          , last_value(conc_date) OVER w AS lastdate
                       FROM nucacid_conc_data
                       WHERE conc_date IS NOT NULL
                       WINDOW w AS (PARTITION BY naid, conc_method
                                    ORDER BY conc_date
                                      RANGE BETWEEN UNBOUNDED PRECEDING
                                        AND UNBOUNDED FOLLOWING))
   , concat_creators AS (SELECT naid
                              , string_agg(creator, '/' ORDER BY naid, nacrid) AS ↩
                                 created_by
                           FROM nucacid_creators
                           GROUP BY naid)

SELECT nucacid_data.naid AS naid
     , nucacid_data.tid AS tid
     , nucacid_data.locid AS locid
     , locations.institution AS institution
     , locations.location AS location
     , local_1.localid AS localid_1
     , local_2.localid AS localid_2
     , nucacid_data.uiid AS uiid
     , unique_indivs.popid AS popid
     , unique_indivs.individ AS individ
     , biograph.sname AS sname
     , nucacid_data.name_on_tube AS name_on_tube
     , nucacid_data.nucacid_type AS nucacid_type
     , tissue_data.tissue_type AS tissue_type
     , nucacid_data.creation_date AS creation_date
     , concat_creators.created_by AS created_by
     , nucacid_data.creation_method AS creation_method
     , COUNT(nucacid_sources.*) AS na_sources
     , nucacid_data.initial_vol_ul AS initial_vol_ul
     , nucacid_data.actual_vol_ul AS actual_vol_ul
     , nucacid_data.actual_vol_date AS actual_vol_date
     , nucacid_data.multi_indivs AS multi_indivs
     , nucacid_data.multi_tids AS multi_tids
     , nucacid_data.notes AS notes
     , qpcr.last_pg_ul AS qpcr_pg_ul
     , qpcr.lastdate AS qpcr_lastdate
     , (nanodrop.last_pg_ul / 1000)::numeric(10,4) AS nanodrop_ng_ul
     , nanodrop.lastdate AS nanodrop_lastdate
     , (qubit.last_pg_ul / 1000)::numeric(10,4) AS qubit_ng_ul
     , qubit.lastdate AS qubit_lastdate
     , (bioanalyzer.last_pg_ul / 1000)::numeric(10,4) AS bioanalyzer_ng_ul
     , bioanalyzer.lastdate AS bioanalyzer_lastdate
     , (quantit.last_pg_ul / 1000)::numeric(10,4) AS quantit_ng_ul
     , quantit.lastdate AS quantit_lastdate
  FROM nucacid_data
  JOIN locations
    ON locations.locid = nucacid_data.locid
  LEFT JOIN tissue_data
    ON tissue_data.tid = nucacid_data.tid
  LEFT JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON biograph.bioid::text = unique_indivs.individ
       AND unique_indivs.popid = 1
```

If we could we would display here a diagram showing how the NUCACIDS_W_CONC view is constructed.

Figure 78: Entity Relationship Diagram of the NUCACIDS_W_CONC View

3.30 The PARENTS View

```
SELECT biograph.sname AS kid
     , maternities.mom AS mom
     , dads_consensus.dad_consensus AS dad
     , maternities.zdate AS zdate
     , maternities.zdate_grp AS momgrp
     , members.grp AS dadgrp
  FROM biograph
       LEFT OUTER JOIN maternities
            ON (maternities.child = biograph.sname)
       LEFT OUTER JOIN dads_consensus
            ON (dads_consensus.kid = biograph.sname)
       LEFT OUTER JOIN members
            ON (members.sname = dads_consensus.dad_consensus
                AND members.date = maternities.zdate)
  WHERE maternities.mom IS NOT NULL
        OR dads_consensus.dad_consensus IS NOT NULL;
```

Figure 79: Query Defining the PARENTS View

If we could we would display here the diagram showing how the PARENTS view is constructed.

Figure 80: Entity Relationship Diagram of the PARENTS View

3.31 The PCV_STATS View

```
SELECT pcvs.dartid AS dartid
     , count(*) AS pcvsamps
     , avg(pcvs.pcv) AS pcv_mean
     , stddev(pcvs.pcv) AS pcv_stddev
  FROM pcvs
  GROUP BY pcvs.dartid;
```

Figure 81: Query Defining the PCV_STATS View

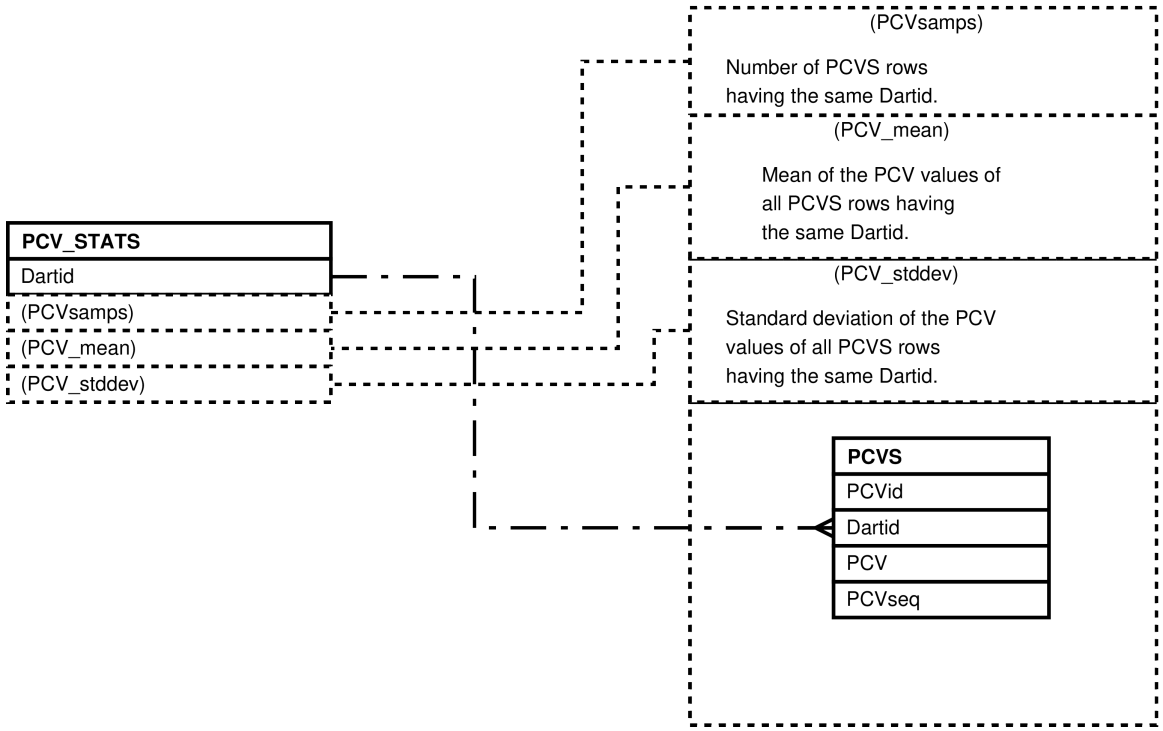

Figure 82: Entity Relationship Diagram of the PCV_STATS View

3.32 The POINTS and POINTS_SORTED Views

```
SELECT pntid AS pntid
     , sid AS sid
     , activity AS activity
     , posture AS posture
     , foodcode AS foodcode
     , ptime AS ptime
     , spm(ptime) AS ptimespm
  FROM point_data;
```

Figure 83: Query Defining the POINTS View

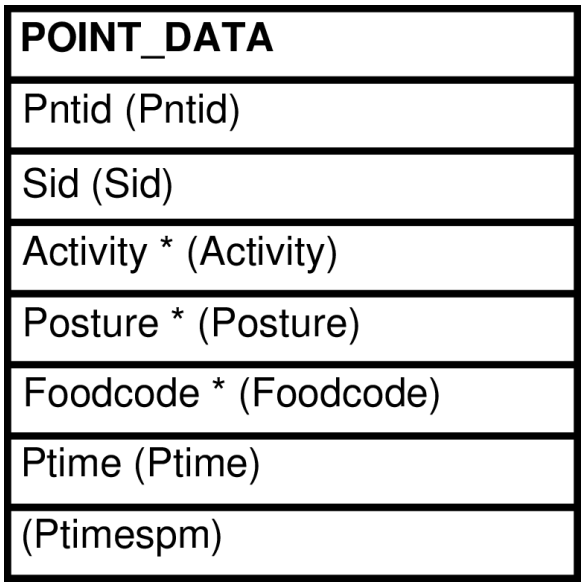

Figure 84: Entity Relationship Diagram of the POINTS View

3.33 The POTENTIAL_DADS View

```
SELECT maternities.child_bioid AS bioid
     , maternities.child AS kid
     , maternities.mom AS mom
     , maternities.zdate AS zdate
     , maternities.zdate_grp AS grp
     , pdads.sname AS pdad
     , CASE
         WHEN rankdates.ranked <= maternities.zdate
           THEN 'A'
         WHEN maturedates.matured <= maternities.zdate
           THEN 'S'
         ELSE 'O'
       END
       AS status
     , maternities.zdate - pdads.birth AS pdad_age_days
     , trunc((maternities.zdate - pdads.birth) / 365.25, 1)
       AS pdad_age_years
     , (SELECT count(*)
          FROM members as dadmembers
               JOIN members AS mommembers
                 ON (mommembers.date = dadmembers.date
                     AND mommembers.supergroup = dadmembers.supergroup)
          WHERE dadmembers.sname = pdads.sname
                AND dadmembers.date < maternities.zdate
                AND dadmembers.date >= maternities.zdate - 5
                AND mommembers.sname = maternities.mom
                AND mommembers.date < maternities.zdate
                AND mommembers.date >= maternities.zdate - 5)
       AS estrous_presence
     , (SELECT count(*)
          FROM actor_actees
          WHERE actor_actees.date < maternities.zdate
                AND actor_actees.date >= maternities.zdate - 5
                AND (actor_actees.act = 'M'
                     OR actor_actees.act = 'E')
                AND actor_actees.actor = pdads.sname
                AND actor_actees.actee = maternities.mom)
       AS estrous_me
     , (SELECT count(*)
          FROM actor_actees
          WHERE actor_actees.date < maternities.zdate
                AND actor_actees.date >= maternities.zdate - 5
                AND actor_actees.act = 'C'
                AND actor_actees.actor = pdads.sname
                AND actor_actees.actee = maternities.mom)
       AS estrous_c

  FROM maternities
       JOIN biograph AS pdads
            ON (pdads.sname
                IN (SELECT dadmembers.sname
                      FROM members AS dadmembers
                           JOIN members AS mommembers
                             ON (mommembers.date = dadmembers.date
                                 AND mommembers.supergroup
                                     = dadmembers.supergroup)
                      WHERE dadmembers.sname = pdads.sname
                            AND dadmembers.date < maternities.zdate
                            AND dadmembers.date >= maternities.zdate - 5
                            AND mommembers.sname = maternities.mom
                            AND mommembers.date < maternities.zdate
```

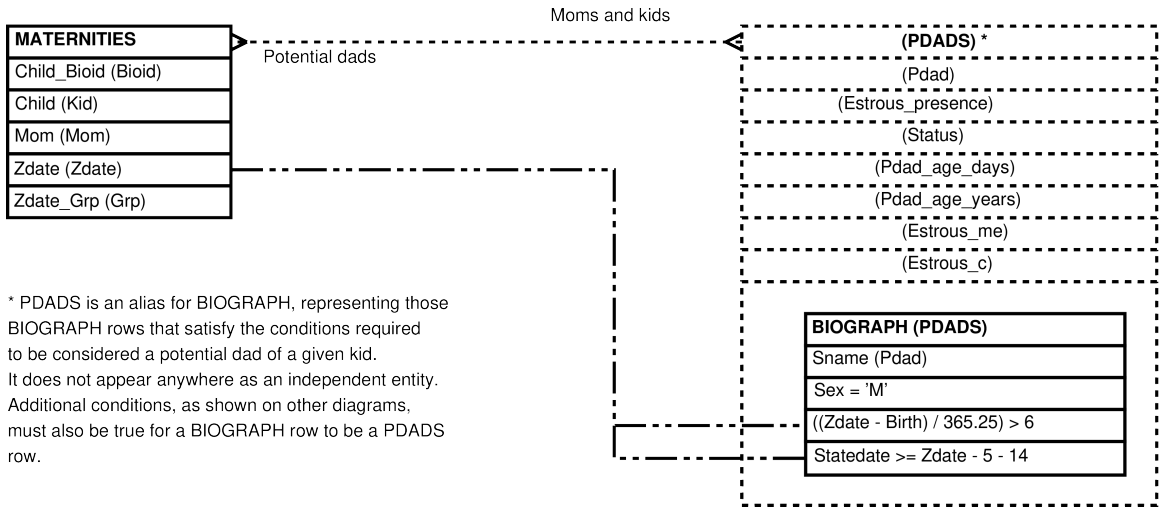

Figure 86: Entity Relationship Diagram of the foundation of the POTENTIAL_DADS View

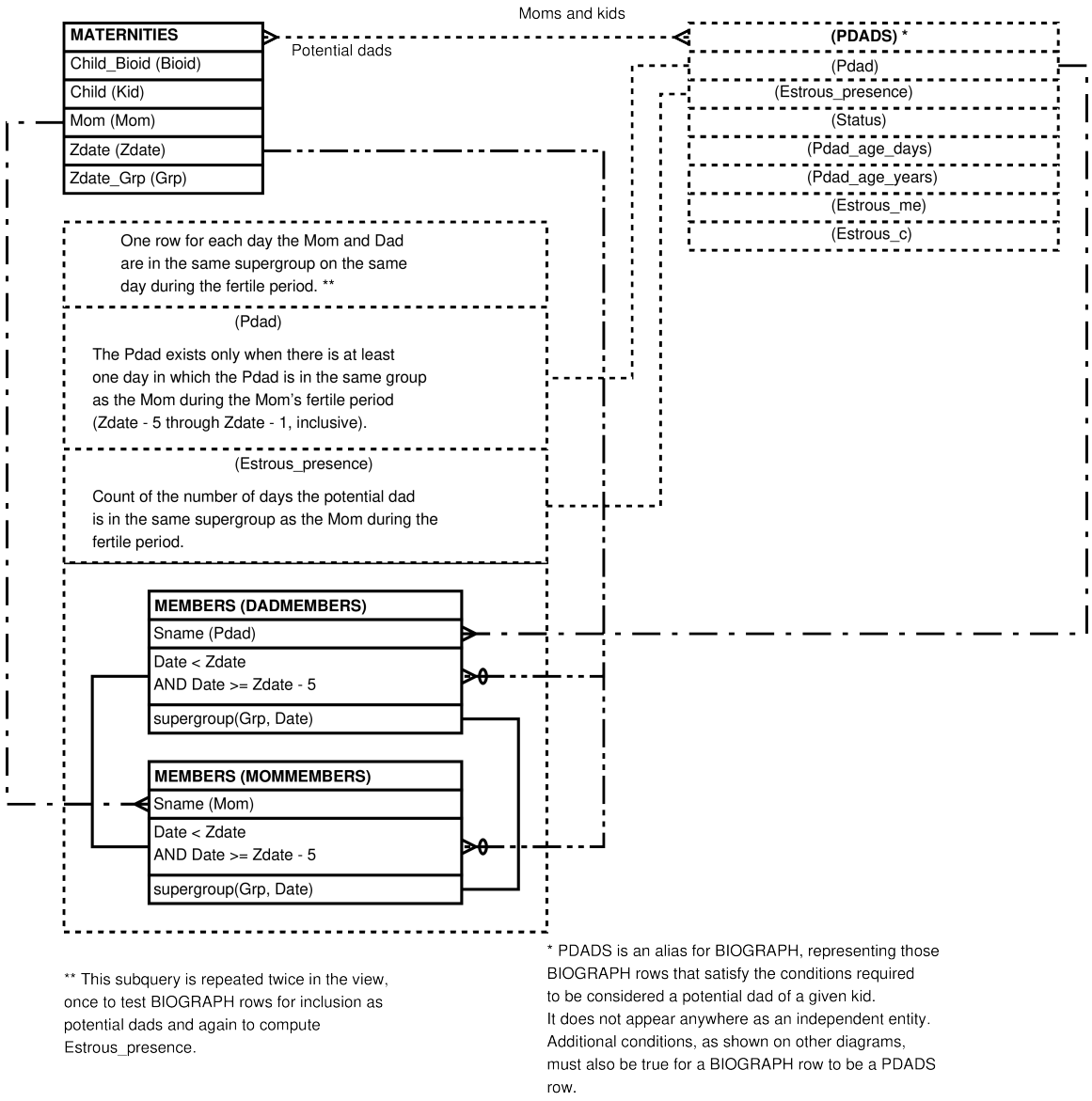

Figure 87: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View which places the mother and potential father in the same group during the fertile period

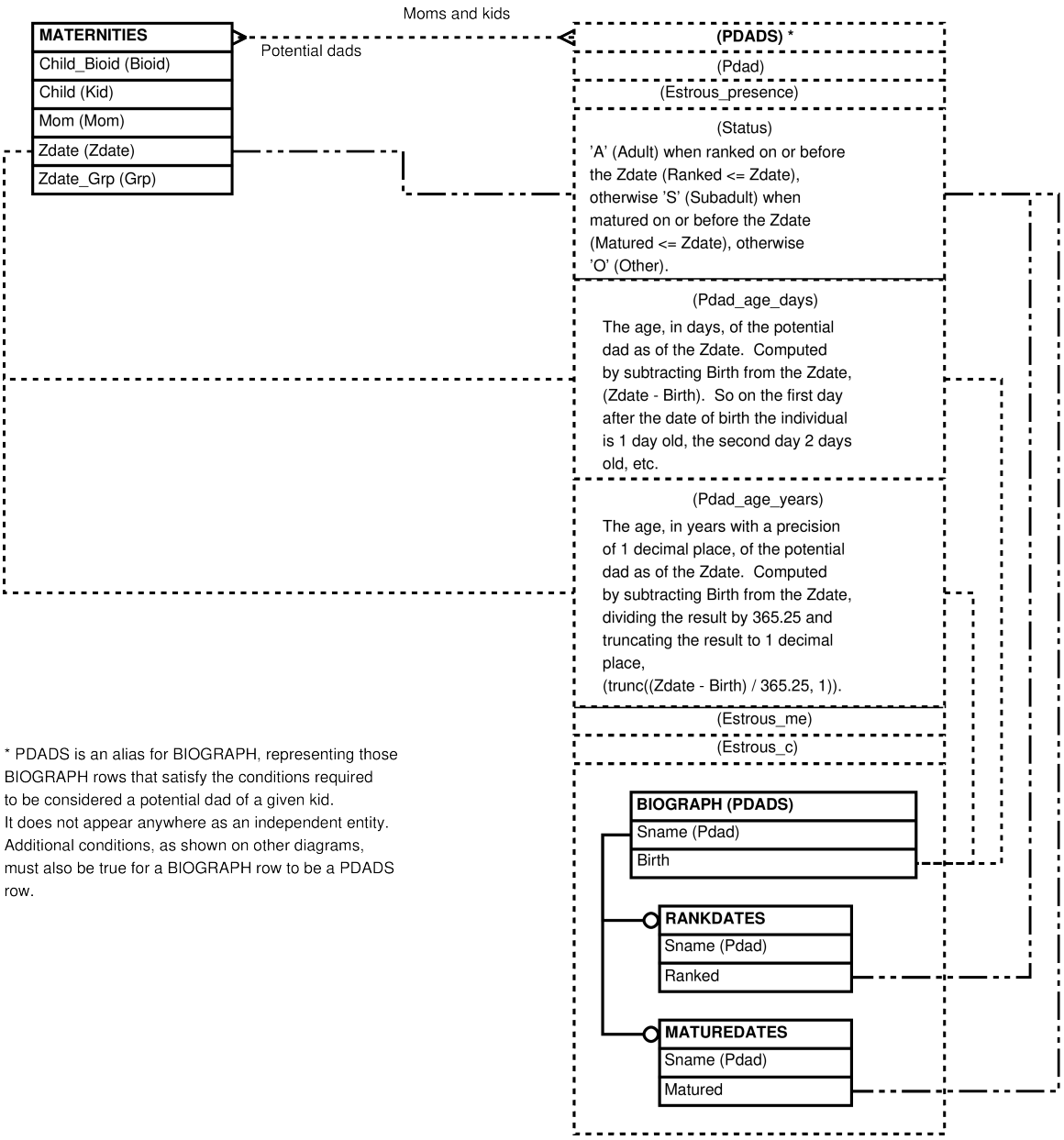

Figure 88: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View having easily computed columns

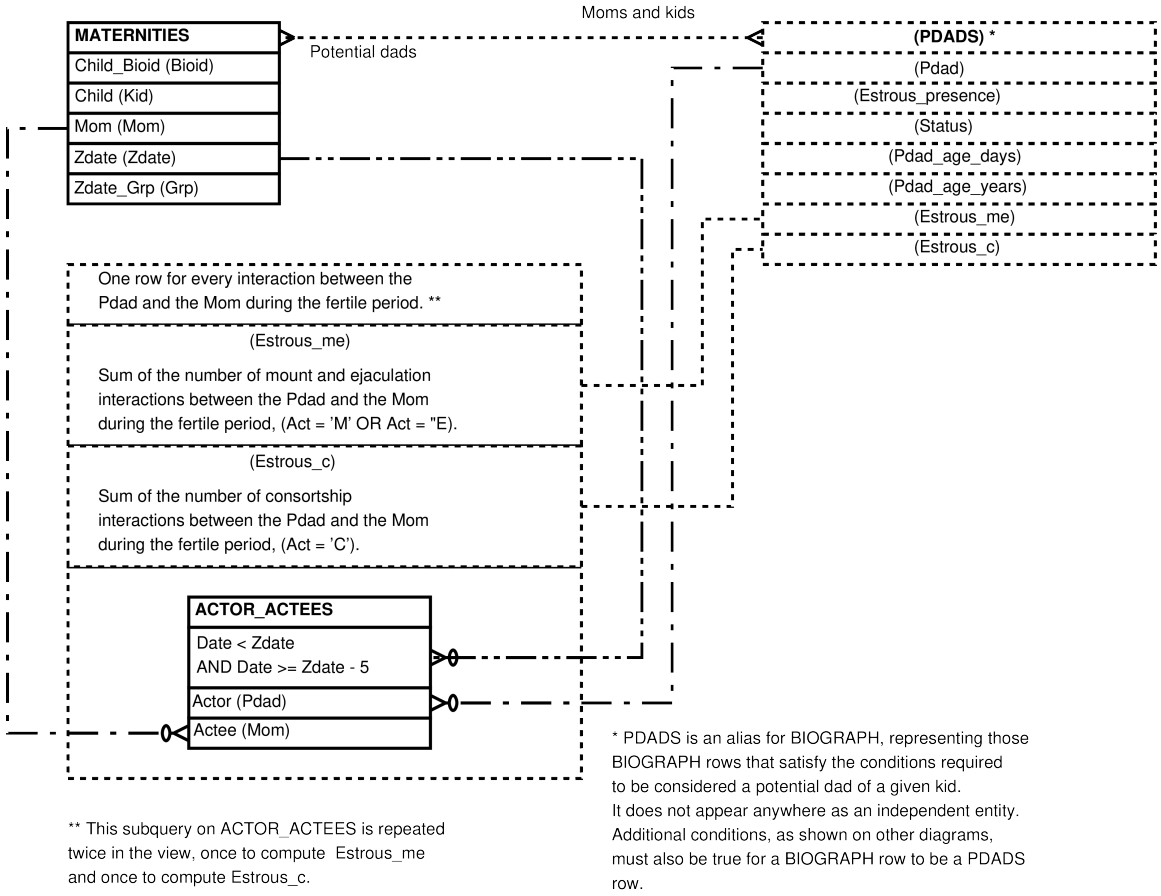

Figure 89: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View involving social interactions

3.34 The PROPORTIONAL_RANKS View

```
WITH num_indivs AS (
  SELECT ranks.rnkdate
       , ranks.grp
       , ranks.rnktype
       , count(*) AS num_members
    FROM ranks
    GROUP BY ranks.rnkdate, ranks.grp, ranks.rnktype)

SELECT ranks.rnkid AS rnkid
     , ranks.sname AS sname
     , ranks.rnkdate AS rnkdate
     , ranks.grp AS grp
     , ranks.rnktype AS rnktype
     , ranks.rank AS ordrank
     , ranks.ags_density AS ags_density
     , ranks.ags_reversals AS ags_reversals
     , ranks.ags_expected AS ags_expected
     , CASE
         WHEN num_indivs.num_members = 1 THEN 1::numeric
         ELSE 1 - ((ranks.rank - 1)::numeric / (num_indivs.num_members - 1):: ↩
             numeric)
       END::numeric(5,4) AS proprank
  FROM ranks
  JOIN num_indivs
    ON (num_indivs.rnkdate = ranks.rnkdate
        AND num_indivs.grp = ranks.grp
        AND num_indivs.rnktype = ranks.rnktype);
```

Figure 90: Query Defining the PROPORTIONAL_RANKS View

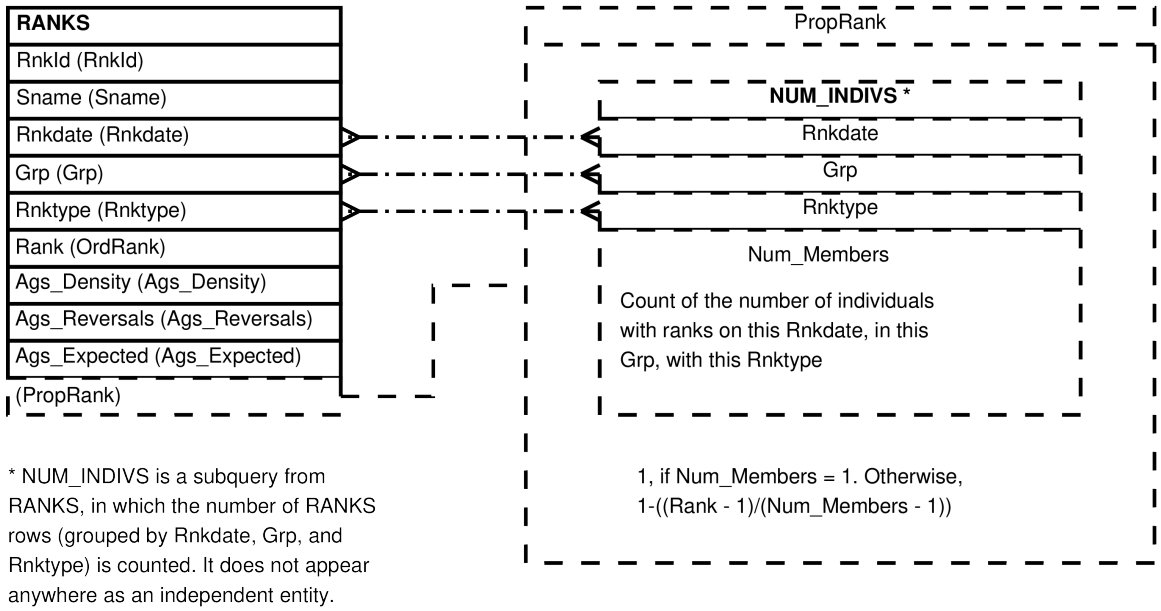

Figure 91: Entity Relationship Diagram of the PROPORTIONAL_RANKS View

3.35 The QUADS View

```
SELECT quad_data.quad AS quad
     , ST_X(quad_data.xyloc) AS x
     , ST_Y(quad_data.xyloc) AS y
     , quad_data.aerial AS aerial
  FROM quad_data;
```

Figure 92: Query Defining the QUADS View

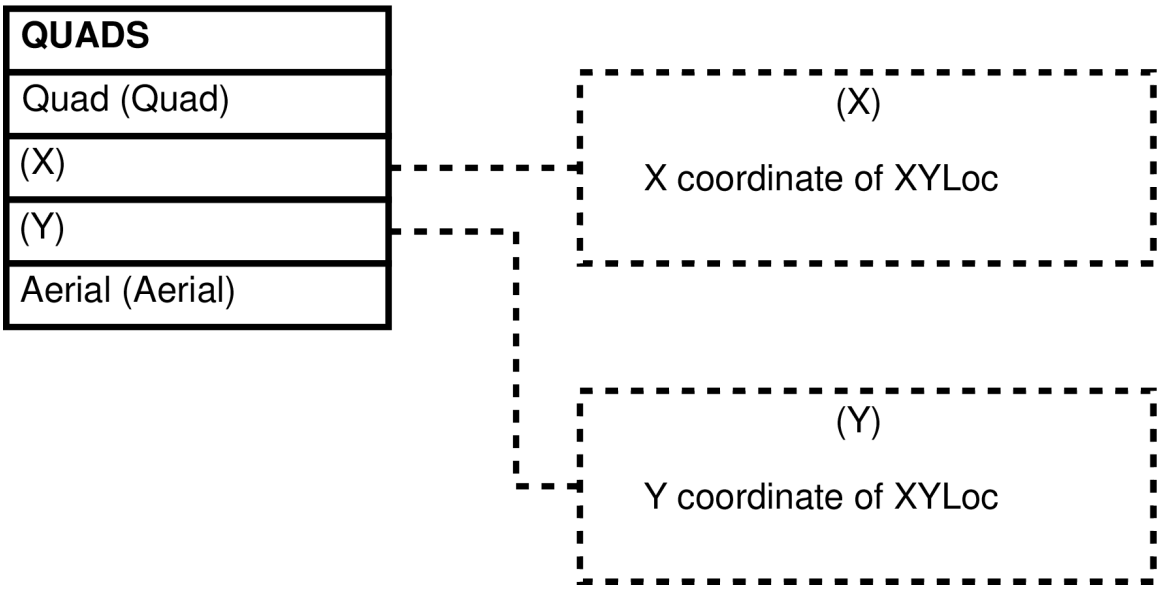

Figure 93: Entity Relationship Diagram of the QUADS View

3.36 The SAMPLES_GOFF View

```
SELECT samples.sid AS sid
     , samples.date AS date
     , samples.stime AS stime
     , samples.observer AS observer
     , samples.stype AS stype
     , samples.grp AS grp
     , samples.sname AS sname
     , samples.mins AS mins
     , samples.minsis AS minsis
     , samples.programid AS programid
     , samples.setupid AS setupid
     , samples.collection_system AS collection_system
     , members.grp AS grp_of_focal
  FROM members, samples
  WHERE members.sname = samples.sname
        AND members.date = CAST(samples.date AS DATE);
```

Figure 94: Query Defining the SAMPLES_GOFF View

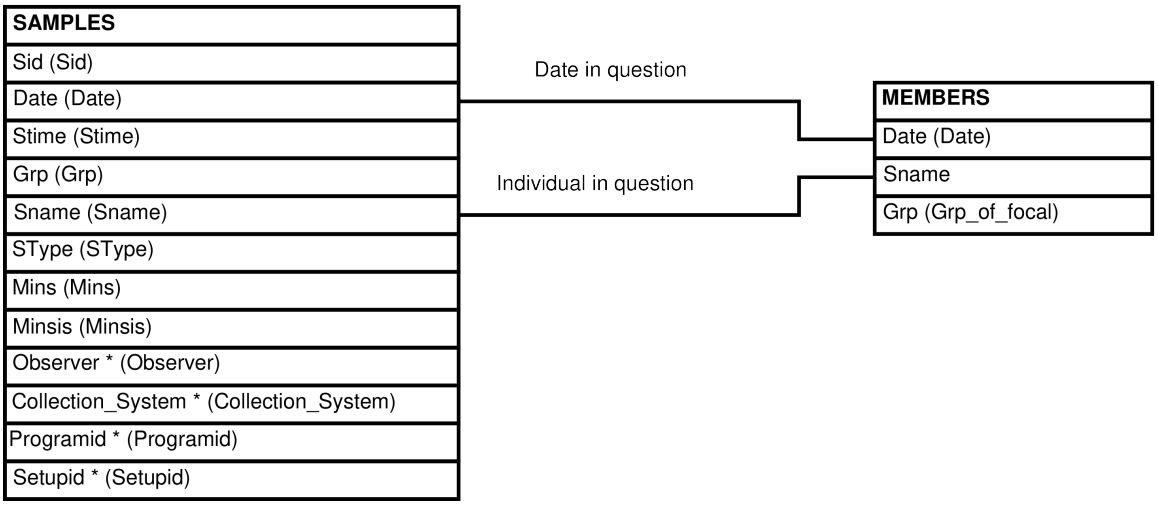

Figure 95: Entity Relationship Diagram of the SAMPLES_GOFF View

3.37 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_SORTED Views

```
SELECT cycles.cid AS cid
     , cycles.sname AS sname
     , cycles.seq AS seq
     , cycles.series AS series
     , sexskins.sxid AS sxid
     , sexskins.date AS date
     , sexskins.size AS size
     , sexskins.color AS color
  FROM sexskins, cycles
  WHERE cycles.cid = sexskins.cid
  ORDER BY cycles.sname, sexskins.date;
```

Figure 96: Query Defining the SEXSKINS_CYCLES View

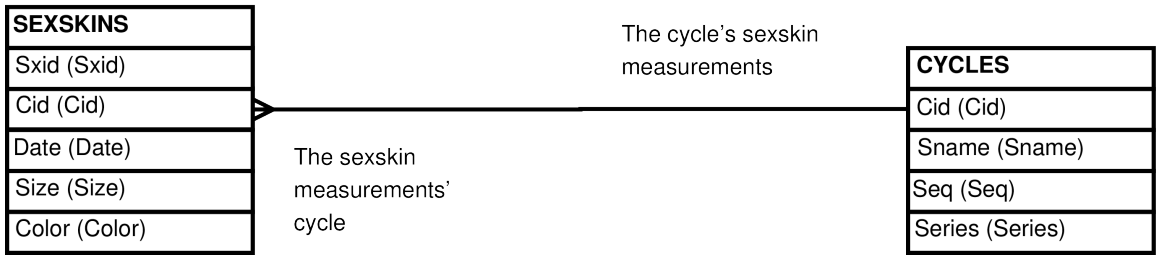

Figure 97: Entity Relationship Diagram of the SEXSKINS_CYCLES View

3.38 The SEXSKINS_REPRO_NOTES View

```
SELECT COALESCE(cycles.sname, repro_notes.sname) AS sname
     , COALESCE(sexskins.date, repro_notes.date) AS date
     , sexskins.cid AS cid
     , sexskins.sxid AS sxid
     , sexskins.size AS size
     , sexskins.color AS color
     , repro_notes.rnid AS rnid
     , repro_notes.note AS note
  FROM sexskins
  JOIN cycles
    ON cycles.cid = sexskins.cid
  FULL OUTER JOIN repro_notes
    ON repro_notes.sname = cycles.sname
       AND repro_notes.date = sexskins.date;
```

Figure 98: Query Defining the SEXSKINS_REPRO_NOTES View

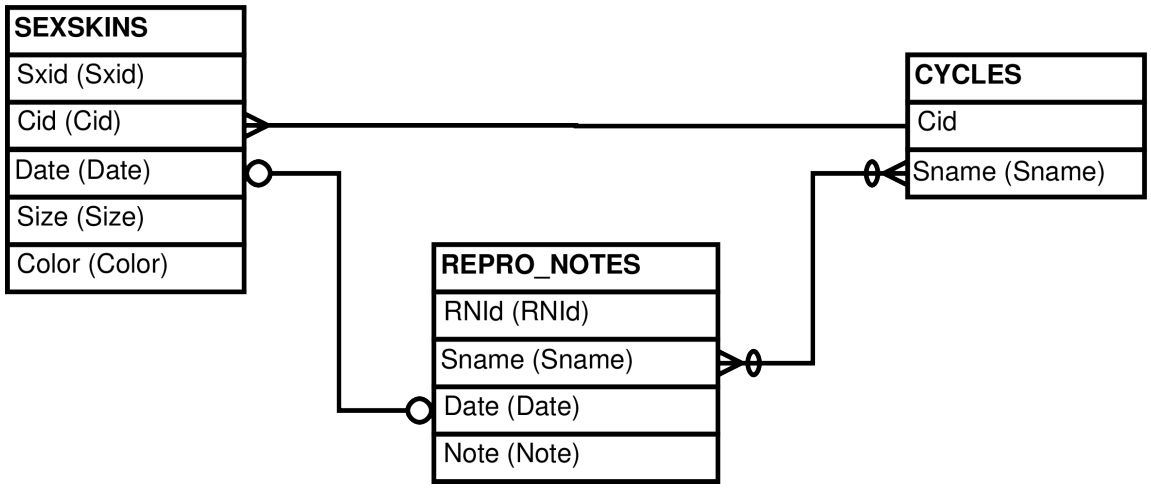

Figure 99: Entity Relationship Diagram of the SEXSKINS_REPRO_NOTES View

3.39 The SWERB view

```
SELECT swerb_data.swid AS swid
     , swerb_departs_data.did AS did
     , swerb_departs_data.date AS date
     , swerb_data.time AS time
     , swerb_bes.beid AS beid
     , swerb_bes.focal_grp AS focal_grp
     , swerb_bes.seq AS seq
     , swerb_data.event AS event
     , swerb_data.seen_grp AS seen_grp
     , swerb_data.lone_animal AS lone_animal
     , swerb_data.quad AS quad
     , CASE
         WHEN swerb_data.quad IS NOT NULL
           THEN 'quad'
         WHEN swerb_data.xyloc IS NULL
           THEN 'n/a'
         ELSE 'gps'
       END AS xysource
     , COALESCE(ST_X(swerb_data.xyloc), ST_X(quad_data.xyloc))
         AS x
     , COALESCE(ST_Y(swerb_data.xyloc), ST_Y(quad_data.xyloc))
         AS y
     , COALESCE(ST_X(ST_TRANSFORM(swerb_data.xyloc, 4326))
              , ST_X(ST_TRANSFORM(quad_data.xyloc, 4326)))
         AS long
     , COALESCE(ST_Y(ST_TRANSFORM(swerb_data.xyloc, 4326))
              , ST_Y(ST_TRANSFORM(quad_data.xyloc, 4326)))
         AS lat
     , swerb_data.altitude AS altitude
     , swerb_data.pdop AS pdop
     , swerb_data.accuracy AS accuracy
     , swerb_data.subgroup AS subgroup
     , swerb_data.ogdistance AS ogdistance
     , swerb_data.gps_datetime AS gps_datetime
     , swerb_data.garmincode AS garmincode
     , swerb_data.predator AS predator
     , swerb_loc_data.loc AS loc
     , swerb_loc_data.adcode AS adcode
     , adcodes.adn AS adn
     , swerb_loc_data.loc_status AS loc_status
     , swerb_loc_data.adtime AS adtime
     , ST_X(swerb_loc_gps.xyloc) AS second_x
     , ST_Y(swerb_loc_gps.xyloc) AS second_y
     , ST_X(ST_TRANSFORM(swerb_loc_gps.xyloc, 4326)) AS second_long
     , ST_Y(ST_TRANSFORM(swerb_loc_gps.xyloc, 4326)) AS second_lat
     , swerb_loc_gps.altitude AS second_altitude
     , swerb_loc_gps.pdop AS second_pdop
     , swerb_loc_gps.accuracy AS second_accuracy
     , swerb_loc_gps.gps_datetime AS second_gps_datetime
     , swerb_loc_gps.garmincode AS second_garmincode
     , swerb_bes.start AS start
     , swerb_bes.btimeest AS btimeest
     , swerb_bes.bsource AS bsource
     , swerb_bes.stop AS stop
     , swerb_bes.etimeest AS etimeest
     , swerb_bes.esource AS esource
     , swerb_bes.is_effort AS is_effort
     , swerb_departs_gps.gps AS gps
     , swerb_bes.notes AS notes
  FROM swerb_data
       LEFT OUTER JOIN quad_data
```

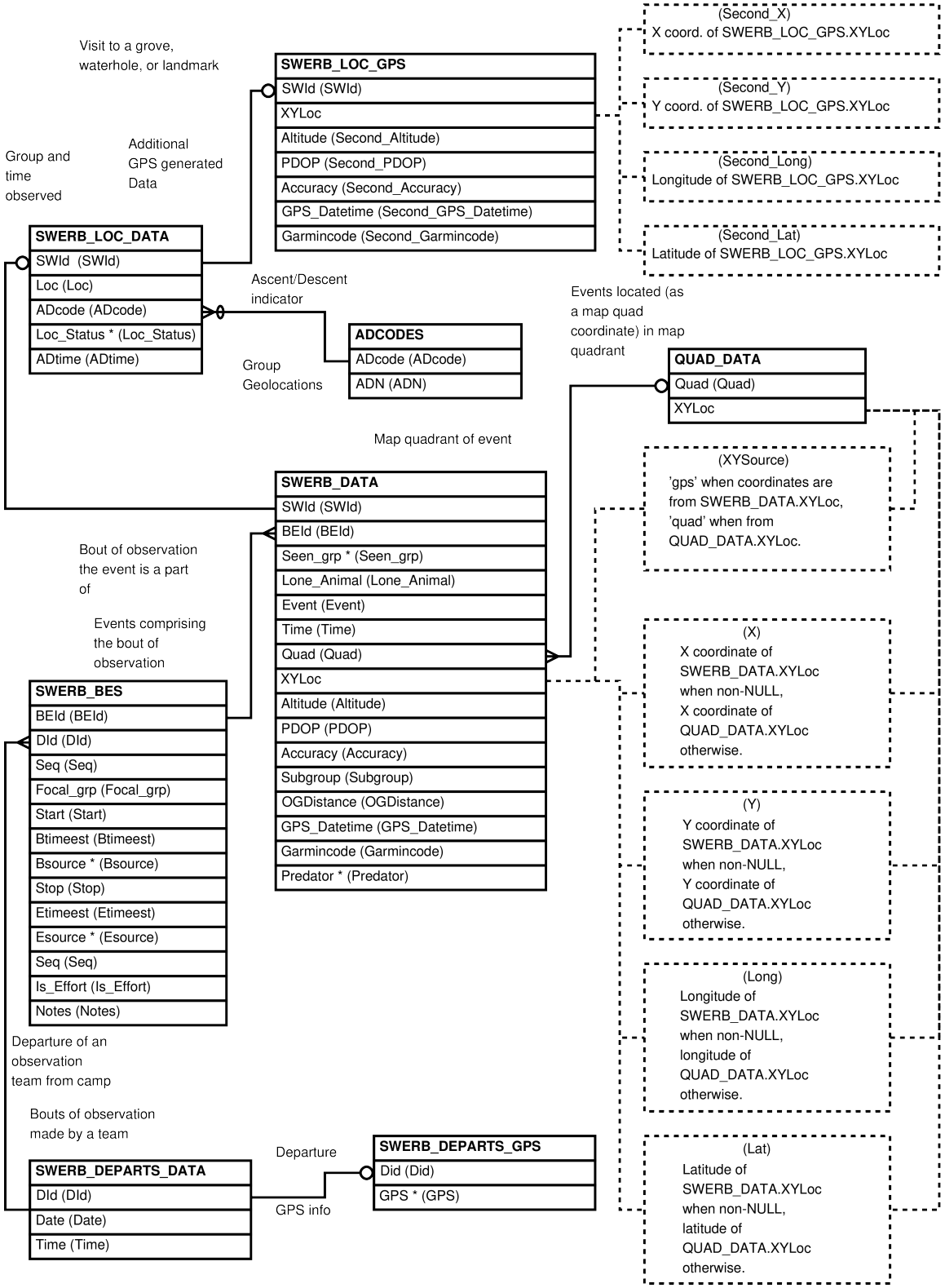

Figure 101: Entity Relationship Diagram of the SWERB View

3.40 The SWERB_DATA_XY View

```
SELECT swerb_data.swid AS swid
     , swerb_data.beid AS beid
     , swerb_data.seen_grp AS seen_grp
     , swerb_data.lone_animal AS lone_animal
     , swerb_data.event AS event
     , swerb_data.time AS time
     , swerb_data.quad AS quad
     , ST_X(swerb_data.xyloc) AS x
     , ST_Y(swerb_data.xyloc) AS y
     , ST_X(ST_TRANSFORM(swerb_data.xyloc, 4326)) AS long
     , ST_Y(ST_TRANSFORM(swerb_data.xyloc, 4326)) AS lat
     , swerb_data.altitude AS altitude
     , swerb_data.pdop AS pdop
     , swerb_data.accuracy AS accuracy
     , swerb_data.subgroup AS subgroup
     , swerb_data.ogdistance AS ogdistance
     , swerb_data.gps_datetime AS gps_datetime
     , swerb_data.garmincode AS garmincode
     , swerb_data.predator AS predator
  FROM swerb_data;
```

Figure 102: Query Defining the SWERB_DATA_XY View

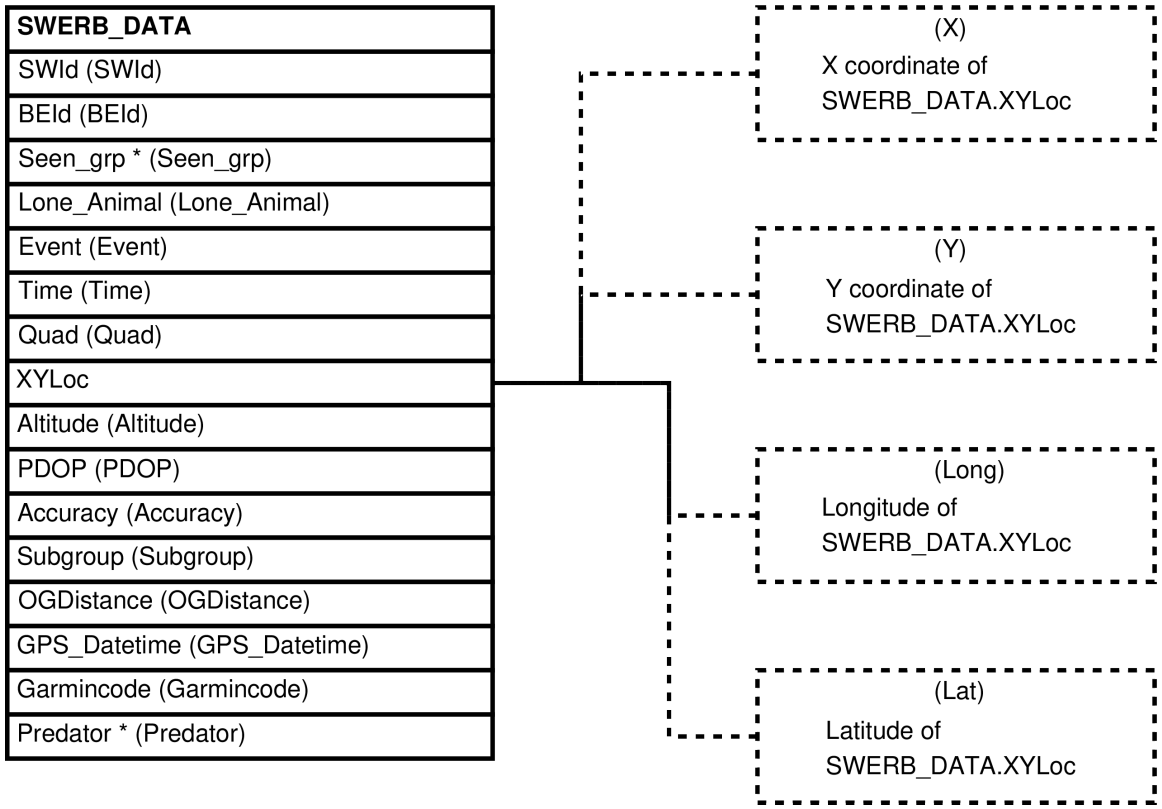

Figure 103: Entity Relationship Diagram of the SWERB_DATA_XY View

3.41 The SWERB_DEPARTS view

```
SELECT swerb_departs_data.did AS did
     , swerb_departs_data.date AS date
     , swerb_departs_data.time AS time
     , ST_X(swerb_departs_gps.xyloc) AS x
     , ST_Y(swerb_departs_gps.xyloc) AS y
     , ST_X(ST_TRANSFORM(swerb_departs_gps.xyloc, 4326)) AS long
     , ST_Y(ST_TRANSFORM(swerb_departs_gps.xyloc, 4326)) AS lat
     , swerb_departs_gps.altitude AS altitude
     , swerb_departs_gps.pdop AS pdop
     , swerb_departs_gps.accuracy AS accuracy
     , swerb_departs_gps.gps AS gps
     , swerb_departs_gps.garmincode AS garmincode
  FROM swerb_departs_data
       LEFT OUTER JOIN swerb_departs_gps
                       ON (swerb_departs_gps.did = swerb_departs_data.did);
```

Figure 104: Query Defining the SWERB_DEPARTS View

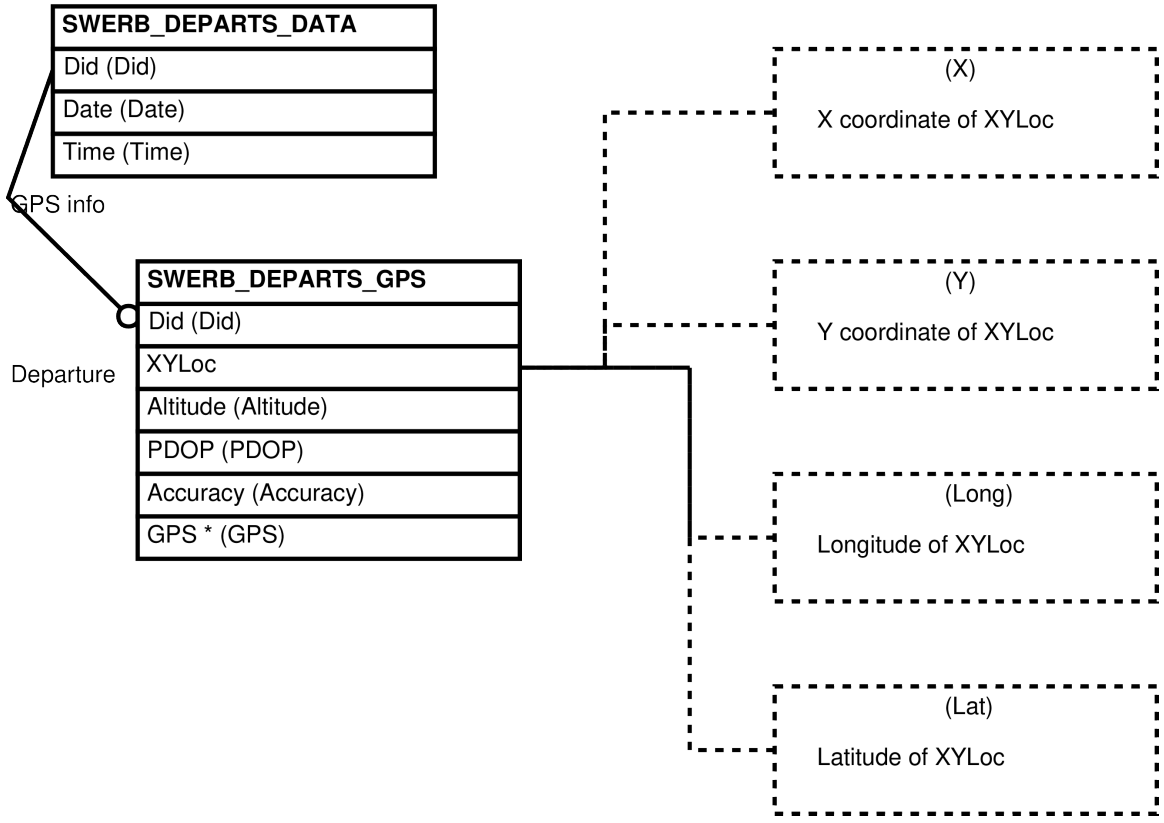

Figure 105: Entity Relationship Diagram of the SWERB_DEPARTS View

3.42 The SWERB_GW_LOC_DATA_XY View

```
SELECT swerb_gw_loc_data.sgwlid AS sgwlid
     , swerb_gw_loc_data.loc AS loc
     , swerb_gw_loc_data.date AS date
     , swerb_gw_loc_data.time AS time
     , swerb_gw_loc_data.quad AS quad
     , swerb_gw_loc_data.xysource AS xysource
     , ST_X(swerb_gw_loc_data.xyloc) AS x
     , ST_Y(swerb_gw_loc_data.xyloc) AS y
     , ST_X(ST_TRANSFORM(swerb_gw_loc_data.xyloc, 4326)) AS long
     , ST_Y(ST_TRANSFORM(swerb_gw_loc_data.xyloc, 4326)) AS lat
     , swerb_gw_loc_data.altitude AS altitude
     , swerb_gw_loc_data.pdop AS pdop
     , swerb_gw_loc_data.accuracy AS accuracy
     , swerb_gw_loc_data.gps AS gps
     , swerb_gw_loc_data.notes AS notes
  FROM swerb_gw_loc_data;
```

Figure 106: Query Defining the SWERB_GW_LOC_DATA_XY View

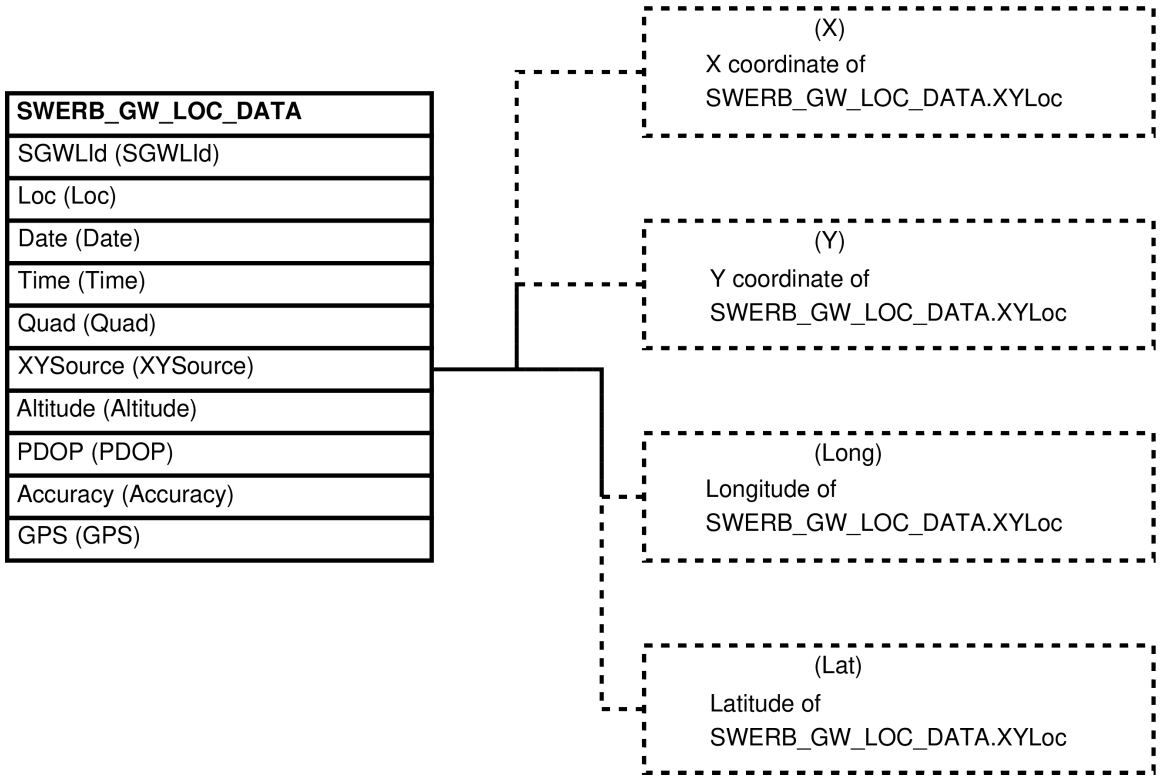

Figure 107: Entity Relationship Diagram of the SWERB_GW_LOC_DATA_XY View

3.43 The SWERB_GW_LOCS View

```
SELECT swerb_gw_loc_data.sgwlid AS sgwlid
     , swerb_gw_loc_data.loc AS loc
     , swerb_gw_loc_data.date AS date
     , swerb_gw_loc_data.time AS time
     , swerb_gw_loc_data.quad AS quad
     , CASE
         WHEN swerb_gw_loc_data.xyloc IS NULL
           THEN 'quad'
         ELSE swerb_gw_loc_data.xysource
       END AS xysource
     , COALESCE(ST_X(swerb_gw_loc_data.xyloc), ST_X(quad_data.xyloc))
         AS x
     , COALESCE(ST_Y(swerb_gw_loc_data.xyloc), ST_Y(quad_data.xyloc))
         AS y
     , COALESCE(ST_X(ST_TRANSFORM(swerb_gw_loc_data.xyloc, 4326))
              , ST_X(ST_TRANSFORM(quad_data.xyloc, 4326)))
         AS long
     , COALESCE(ST_Y(ST_TRANSFORM(swerb_gw_loc_data.xyloc, 4326))
              , ST_Y(ST_TRANSFORM(quad_data.xyloc, 4326)))
         AS lat
     , swerb_gw_loc_data.altitude AS altitude
     , swerb_gw_loc_data.pdop AS pdop
     , swerb_gw_loc_data.accuracy AS accuracy
     , swerb_gw_loc_data.gps AS gps
     , swerb_gw_loc_data.notes AS notes
  FROM swerb_gw_loc_data
       LEFT OUTER JOIN quad_data
                       ON (quad_data.quad = swerb_gw_loc_data.quad);
```

Figure 108: Query Defining the SWERB_GW_LOCS View

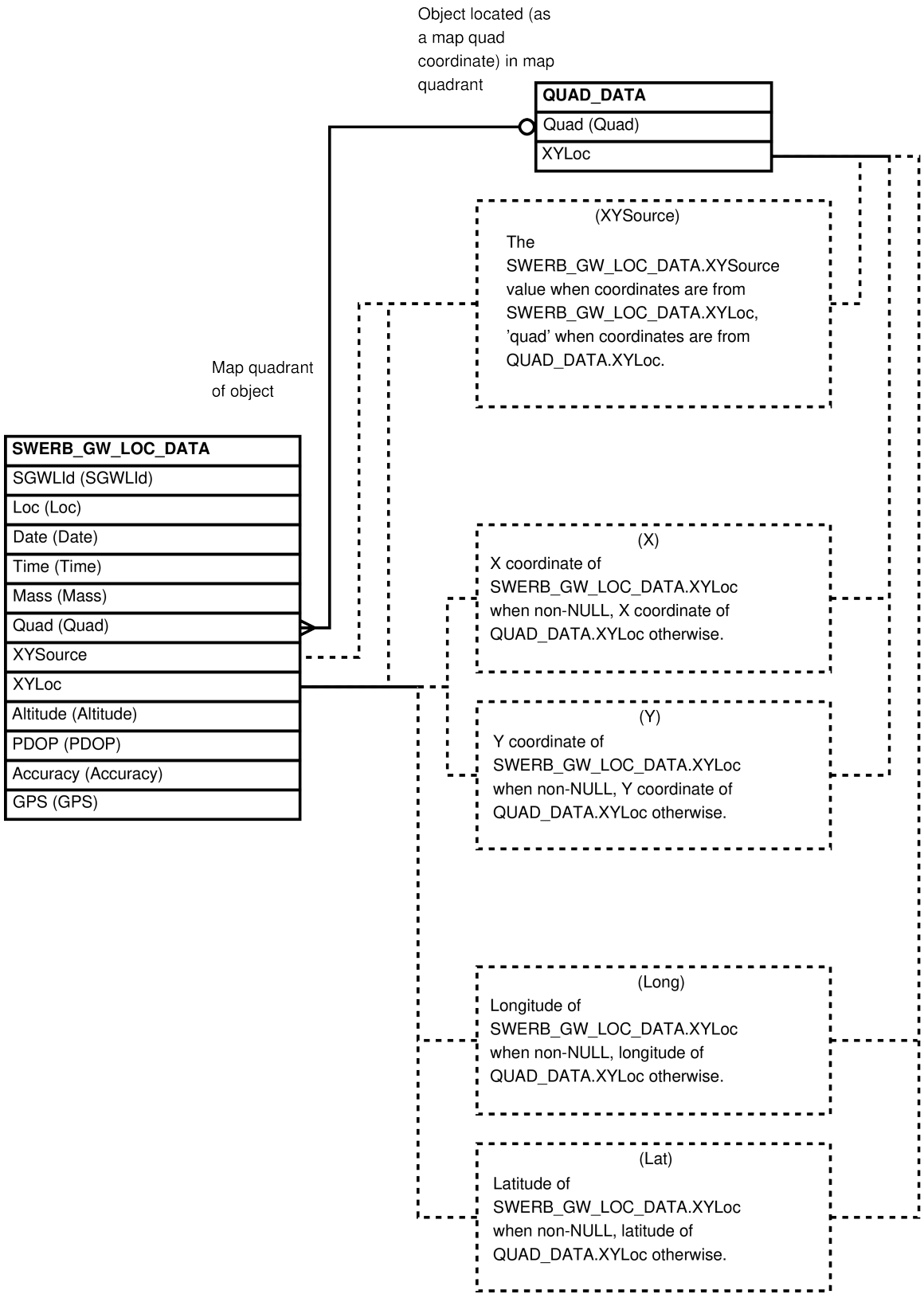

Figure 109: Entity Relationship Diagram of the SWERB_GW_LOCS View

3.44 The SWERB_LOC_GPS_XY view

```
SELECT swerb_loc_gps.swid AS swid
     , ST_X(swerb_loc_gps.xyloc) AS x
     , ST_Y(swerb_loc_gps.xyloc) AS y
     , ST_X(ST_TRANSFORM(swerb_loc_gps.xyloc, 4326)) AS long
     , ST_Y(ST_TRANSFORM(swerb_loc_gps.xyloc, 4326)) AS lat
     , swerb_loc_gps.altitude AS altitude
     , swerb_loc_gps.pdop AS pdop
     , swerb_loc_gps.accuracy AS accuracy
     , swerb_loc_gps.gps_datetime AS gps_datetime
     , swerb_loc_gps.garmincode AS garmincode
  FROM swerb_loc_gps;
```

Figure 110: Query Defining the SWERB_LOC_GPS_XY View

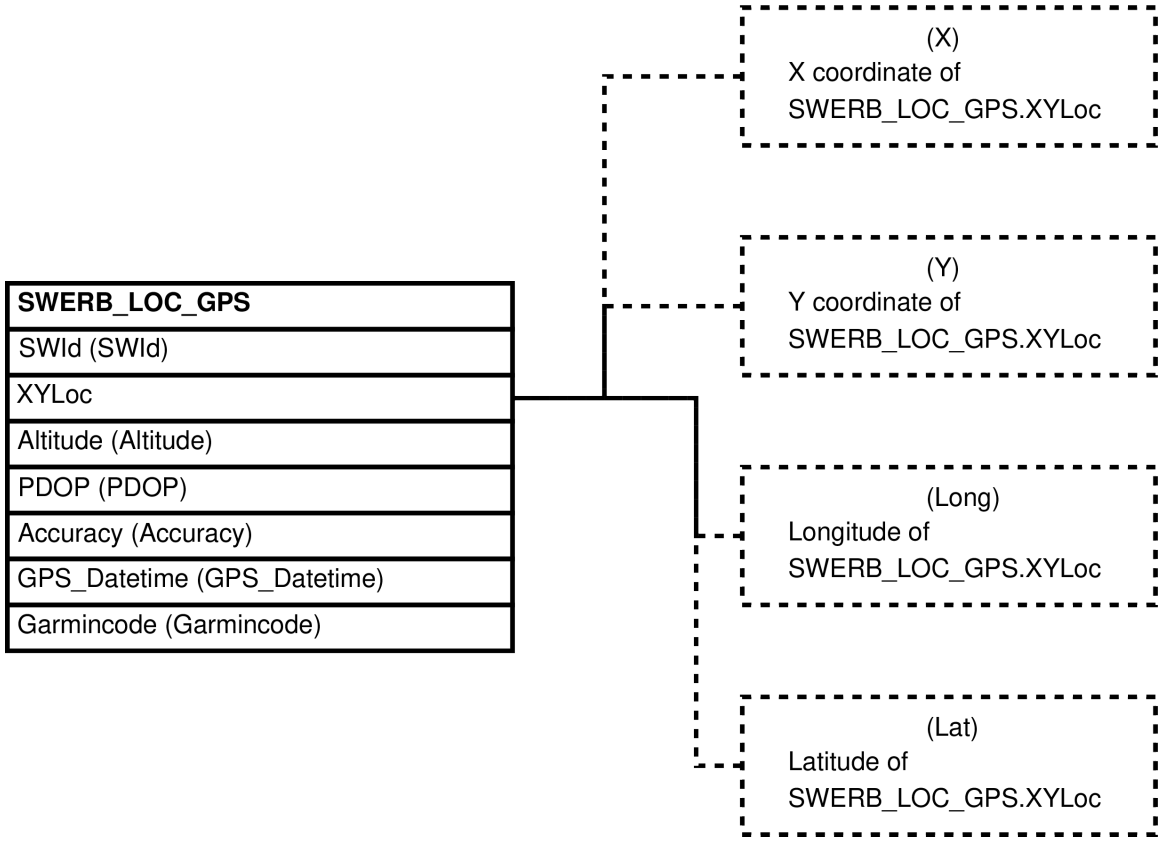

Figure 111: Entity Relationship Diagram of the SWERB_LOC_GPS_XY View

3.45 The SWERB_LOCS view

```
SELECT swerb_loc_data.swid AS swid
     , swerb_loc_data.loc AS loc
     , swerb_loc_data.adcode AS adcode
     , adcodes.adn AS adn
     , swerb_loc_data.loc_status AS loc_status
     , swerb_loc_data.adtime AS time
  FROM swerb_loc_data
       JOIN adcodes ON (adcodes.adcode = swerb_loc_data.adcode);
```

Figure 112: Query Defining the SWERB_LOCS View

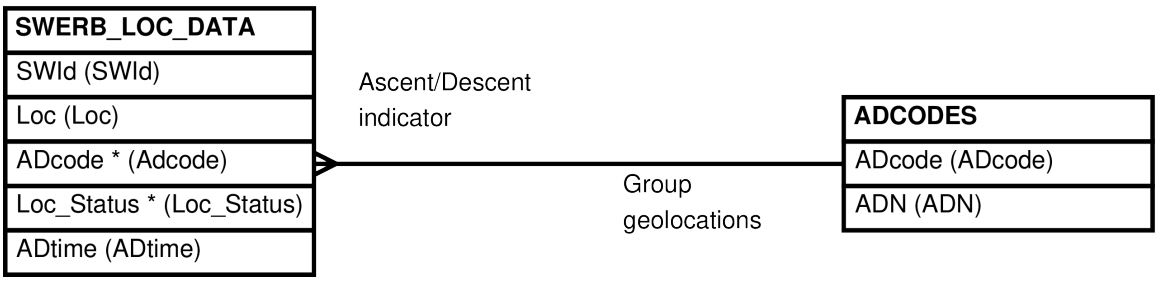

Figure 113: Entity Relationship Diagram of the SWERB_LOCS View

3.46 The SWERB_UPLOAD view

```
SELECT NULL::TEXT AS header
     , NULL::TEXT    AS name
     , NULL::TEXT    AS description
     , NULL::TEXT    AS type
     , NULL::TEXT    AS position
     , NULL::TEXT    AS altitude
     , NULL::TEXT    AS depth
     , NULL::TEXT    AS proximity
     , NULL::TEXT    AS display_mode
     , NULL::TEXT    AS color
     , NULL::TEXT    AS symbol
     , NULL::TEXT    AS facility
     , NULL::TEXT    AS city
     , NULL::TEXT    AS state
     , NULL::TEXT    AS country
     , NULL::TEXT    AS pdop
     , NULL::TEXT    AS accuracy
     , NULL::TEXT    AS quad
     , NULL::TEXT    AS date
     , NULL::TEXT    AS timeest
     , NULL::TEXT    AS source
     , NULL::TEXT    AS lone_animal
     , NULL::TEXT    AS is_effort
     , NULL::BOOLEAN AS secondary_ad
     , NULL::BOOLEAN AS be_has_coords
     , NULL::TEXT    AS notes
  WHERE _raise_babase_exception(
          'Cannot select SWERB_UPLOAD'
          || ': The only use of the SWERB_UPLOAD view is to insert'
          || ' new data into the SWERB portion of babase');
```

Figure 114: Query Defining the SWERB_UPLOAD View

The SWERB_UPLOAD view is used only to insert data into the SWERB portion of Babase. Since it cannot be queried and the semantics of the uploaded file varies by line it has no ER diagram.

Figure 115: Entity Relationship Diagram of the SWERB_UPLOAD View

3.47 The TESTES_ARC_STATS View

```
SELECT testesdartids.dartid AS dartid
     , testesllength.testllengthsamps AS testllengthsamps
     , testesllength.testllength_mean AS testllength_mean
     , testesllength.testllength_stddev AS testllength_stddev
     , testeslwidth.testlwidthsamps AS testlwidthsamps
     , testeslwidth.testlwidth_mean AS testlwidth_mean
     , testeslwidth.testlwidth_stddev AS testlwidth_stddev
     , testesrlength.testrlengthsamps AS testrlengthsamps
     , testesrlength.testrlength_mean AS testrlength_mean
     , testesrlength.testrlength_stddev AS testrlength_stddev
     , testesrwidth.testrwidthsamps AS testrwidthsamps
     , testesrwidth.testrwidth_mean AS testrwidth_mean
     , testesrwidth.testrwidth_stddev AS testrwidth_stddev
FROM (SELECT testes_arc.dartid
        FROM testes_arc
        GROUP BY testes_arc.dartid)
       AS testesdartids
     LEFT OUTER JOIN
       (SELECT testes_arc.dartid AS llengthdartid
             , count(*) AS testllengthsamps
             , avg(testes_arc.testlength) AS testllength_mean
             , stddev(testes_arc.testlength) AS testllength_stddev
          FROM testes_arc
          WHERE testes_arc.testside = 'L'
                AND testes_arc.testlength IS NOT NULL
          GROUP BY testes_arc.dartid)
         AS testesllength
       ON testesllength.llengthdartid = testesdartids.dartid
     LEFT OUTER JOIN
       (SELECT testes_arc.dartid AS lwidthdartid
             , count(*) AS testlwidthsamps
             , avg(testes_arc.testwidth) AS testlwidth_mean
             , stddev(testes_arc.testwidth) AS testlwidth_stddev
          FROM testes_arc
          WHERE testes_arc.testside = 'L'
                AND testes_arc.testwidth IS NOT NULL
          GROUP BY testes_arc.dartid)
         AS testeslwidth
       ON testeslwidth.lwidthdartid = testesdartids.dartid
     LEFT OUTER JOIN
       (SELECT testes_arc.dartid AS rlengthdartid
             , count(*) AS testrlengthsamps
             , avg(testes_arc.testlength) AS testrlength_mean
             , stddev(testes_arc.testlength) AS testrlength_stddev
          FROM testes_arc
          WHERE testes_arc.testside = 'R'
                AND testes_arc.testlength IS NOT NULL
          GROUP BY testes_arc.dartid)
         AS testesrlength
       ON testesrlength.rlengthdartid = testesdartids.dartid
     LEFT OUTER JOIN
       (SELECT testes_arc.dartid AS rwidthdartid
             , count(*) AS testrwidthsamps
             , avg(testes_arc.testwidth) AS testrwidth_mean
             , stddev(testes_arc.testwidth) AS testrwidth_stddev
          FROM testes_arc
          WHERE testes_arc.testside = 'R'
                AND testes_arc.testwidth IS NOT NULL
          GROUP BY testes_arc.dartid)
         AS testesrwidth
       ON testesrwidth.rwidthdartid = testesdartids.dartid;
```

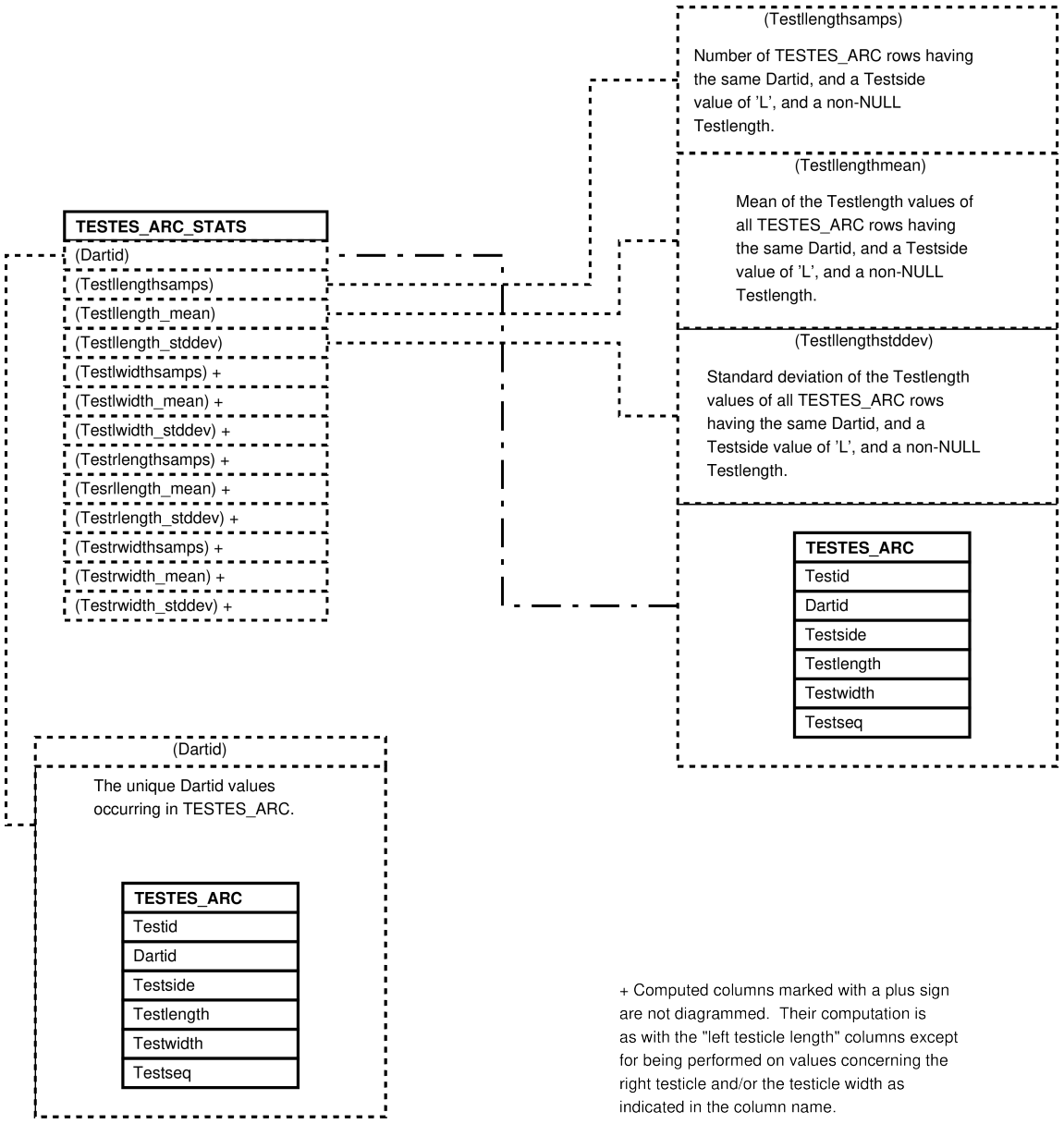

Figure 117: Entity Relationship Diagram of the TESTES_ARC_STATS View

3.48 The TESTES_DIAM_STATS View

```
SELECT testesdartids.dartid AS dartid
     , testesllength.testllengthsamps AS testllengthsamps
     , testesllength.testllength_mean AS testllength_mean
     , testesllength.testllength_stddev AS testllength_stddev
     , testeslwidth.testlwidthsamps AS testlwidthsamps
     , testeslwidth.testlwidth_mean AS testlwidth_mean
     , testeslwidth.testlwidth_stddev AS testlwidth_stddev
     , testesrlength.testrlengthsamps AS testrlengthsamps
     , testesrlength.testrlength_mean AS testrlength_mean
     , testesrlength.testrlength_stddev AS testrlength_stddev
     , testesrwidth.testrwidthsamps AS testrwidthsamps
     , testesrwidth.testrwidth_mean AS testrwidth_mean
     , testesrwidth.testrwidth_stddev AS testrwidth_stddev
FROM (SELECT testes_diam.dartid
        FROM testes_diam
        GROUP BY testes_diam.dartid)
       AS testesdartids
     LEFT OUTER JOIN
       (SELECT testes_diam.dartid AS llengthdartid
             , count(*) AS testllengthsamps
             , avg(testes_diam.testlength) AS testllength_mean
             , stddev(testes_diam.testlength) AS testllength_stddev
          FROM testes_diam
          WHERE testes_diam.testside = 'L'
                AND testes_diam.testlength IS NOT NULL
          GROUP BY testes_diam.dartid)
         AS testesllength
       ON testesllength.llengthdartid = testesdartids.dartid
     LEFT OUTER JOIN
       (SELECT testes_diam.dartid AS lwidthdartid
             , count(*) AS testlwidthsamps
             , avg(testes_diam.testwidth) AS testlwidth_mean
             , stddev(testes_diam.testwidth) AS testlwidth_stddev
          FROM testes_diam
          WHERE testes_diam.testside = 'L'
                AND testes_diam.testwidth IS NOT NULL
          GROUP BY testes_diam.dartid)
         AS testeslwidth
       ON testeslwidth.lwidthdartid = testesdartids.dartid
     LEFT OUTER JOIN
       (SELECT testes_diam.dartid AS rlengthdartid
             , count(*) AS testrlengthsamps
             , avg(testes_diam.testlength) AS testrlength_mean
             , stddev(testes_diam.testlength) AS testrlength_stddev
          FROM testes_diam
          WHERE testes_diam.testside = 'R'
                AND testes_diam.testlength IS NOT NULL
          GROUP BY testes_diam.dartid)
         AS testesrlength
       ON testesrlength.rlengthdartid = testesdartids.dartid
     LEFT OUTER JOIN
       (SELECT testes_diam.dartid AS rwidthdartid
             , count(*) AS testrwidthsamps
             , avg(testes_diam.testwidth) AS testrwidth_mean
             , stddev(testes_diam.testwidth) AS testrwidth_stddev
          FROM testes_diam
          WHERE testes_diam.testside = 'R'
                AND testes_diam.testwidth IS NOT NULL
          GROUP BY testes_diam.dartid)
         AS testesrwidth
       ON testesrwidth.rwidthdartid = testesdartids.dartid;
```

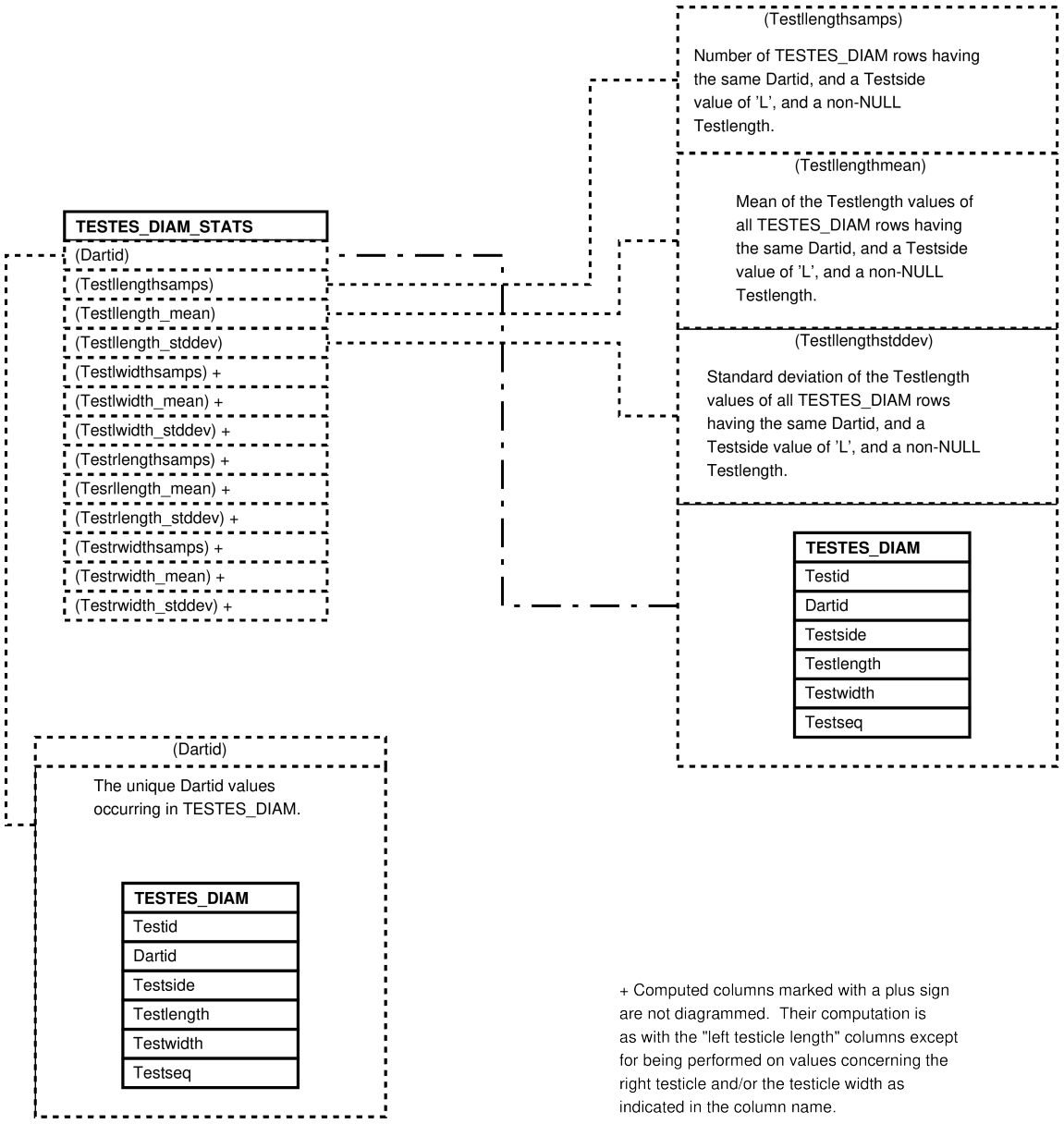

Figure 119: Entity Relationship Diagram of the TESTES_DIAM_STATS View

3.49 The TESTOSTERONES View

```
SELECT hormone_sample_data.tid
     , hormone_prep_series.hpsid
     , hormone_result_data.hrid
     , hormone_sample_data.hsid
     , biograph.sname
     , tissue_data.collection_date
     , tissue_data.collection_date_status AS collection_date_status
     , hormone_sample_data.fzdried_date AS fzdried_date
     , hormone_sample_data.sifted_date AS sifted_date
     , meoh_ext.procedure_date AS me_extracted
     , spe.procedure_date AS sp_extracted
     , hormone_result_data.raw_ng_g AS raw_ng_g
     , corrected_hormone(hormone_result_data.raw_ng_g, hormone_kits.correction) AS ↩
        corrected_ng_g
     , hormone_result_data.assay_date
     , hormone_kits.hormone AS hormone
     , hormone_result_data.kit AS kit
     , hormone_sample_data.comments AS sample_comments
     , hormone_result_data.comments AS result_comments
  FROM hormone_sample_data
  JOIN tissue_data
    ON tissue_data.tid = hormone_sample_data.tid
  JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON unique_indivs.popid = 1
       AND biograph.bioid::text = unique_indivs.individ
  JOIN hormone_prep_series
    ON hormone_prep_series.tid = hormone_sample_data.tid
  JOIN hormone_result_data
    ON hormone_result_data.hpsid = hormone_prep_series.hpsid
  JOIN hormone_kits
    ON hormone_kits.kit = hormone_result_data.kit
       AND hormone_kits.correction IS NOT NULL
       AND hormone_kits.hormone = 'T'
  LEFT JOIN hormone_prep_data AS meoh_ext
    ON meoh_ext.procedure = 'MEOH_EXT'
       AND meoh_ext.hpsid = hormone_prep_series.hpsid
  LEFT JOIN hormone_prep_data AS spe
    ON spe.procedure = 'SPE'
       AND spe.hpsid = hormone_prep_series.hpsid;
```

Figure 120: Query Defining the TESTOSTERONES View

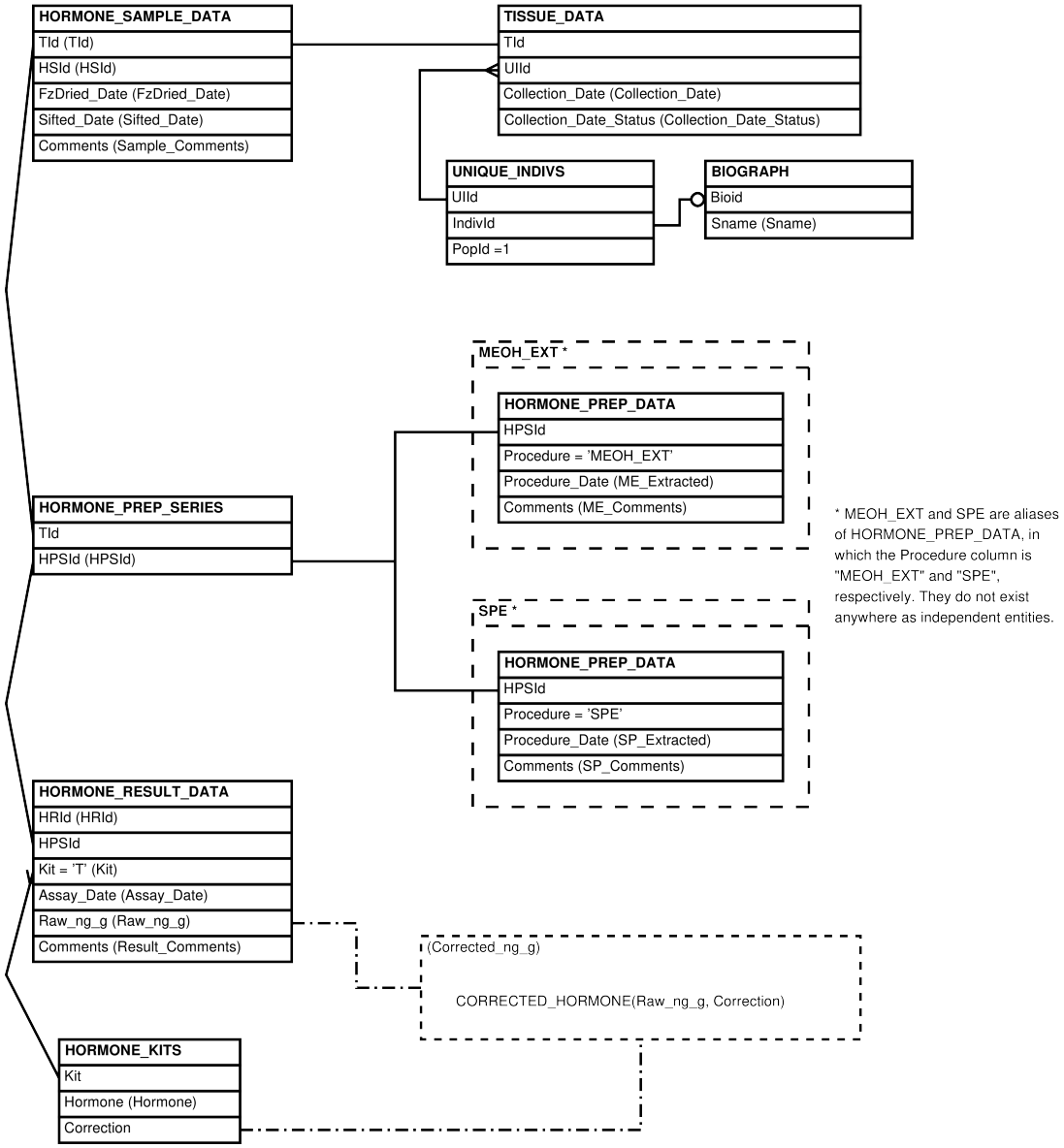

Figure 121: Entity Relationship Diagram of the TESTOSTERONES View

3.50 The THYROID_HORMONES View

```
SELECT hormone_sample_data.tid
     , hormone_prep_series.hpsid
     , hormone_result_data.hrid
     , hormone_sample_data.hsid
     , biograph.sname
     , tissue_data.collection_date
     , tissue_data.collection_date_status AS collection_date_status
     , hormone_sample_data.fzdried_date AS fzdried_date
     , hormone_sample_data.sifted_date AS sifted_date
     , etoh_ext.procedure_date AS et_extracted
     , hormone_result_data.raw_ng_g AS raw_ng_g
     , corrected_hormone(hormone_result_data.raw_ng_g, hormone_kits.correction) AS ↩
        corrected_ng_g
     , hormone_result_data.assay_date
     , hormone_kits.hormone AS hormone
     , hormone_result_data.kit AS kit
     , hormone_sample_data.comments AS sample_comments
     , hormone_result_data.comments AS result_comments
  FROM hormone_sample_data
  JOIN tissue_data
    ON tissue_data.tid = hormone_sample_data.tid
  JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON unique_indivs.popid = 1
       AND biograph.bioid::text = unique_indivs.individ
  JOIN hormone_prep_series
    ON hormone_prep_series.tid = hormone_sample_data.tid
  JOIN hormone_result_data
    ON hormone_result_data.hpsid = hormone_prep_series.hpsid
  JOIN hormone_kits
    ON hormone_kits.kit = hormone_result_data.kit
       AND hormone_kits.correction IS NOT NULL
       AND hormone_kits.hormone = 'TH'
  LEFT JOIN hormone_prep_data AS etoh_ext
    ON etoh_ext.procedure = 'ETOH_EXT'
       AND etoh_ext.hpsid = hormone_prep_series.hpsid;
```

Figure 122: Query Defining the THYROID_HORMONES View

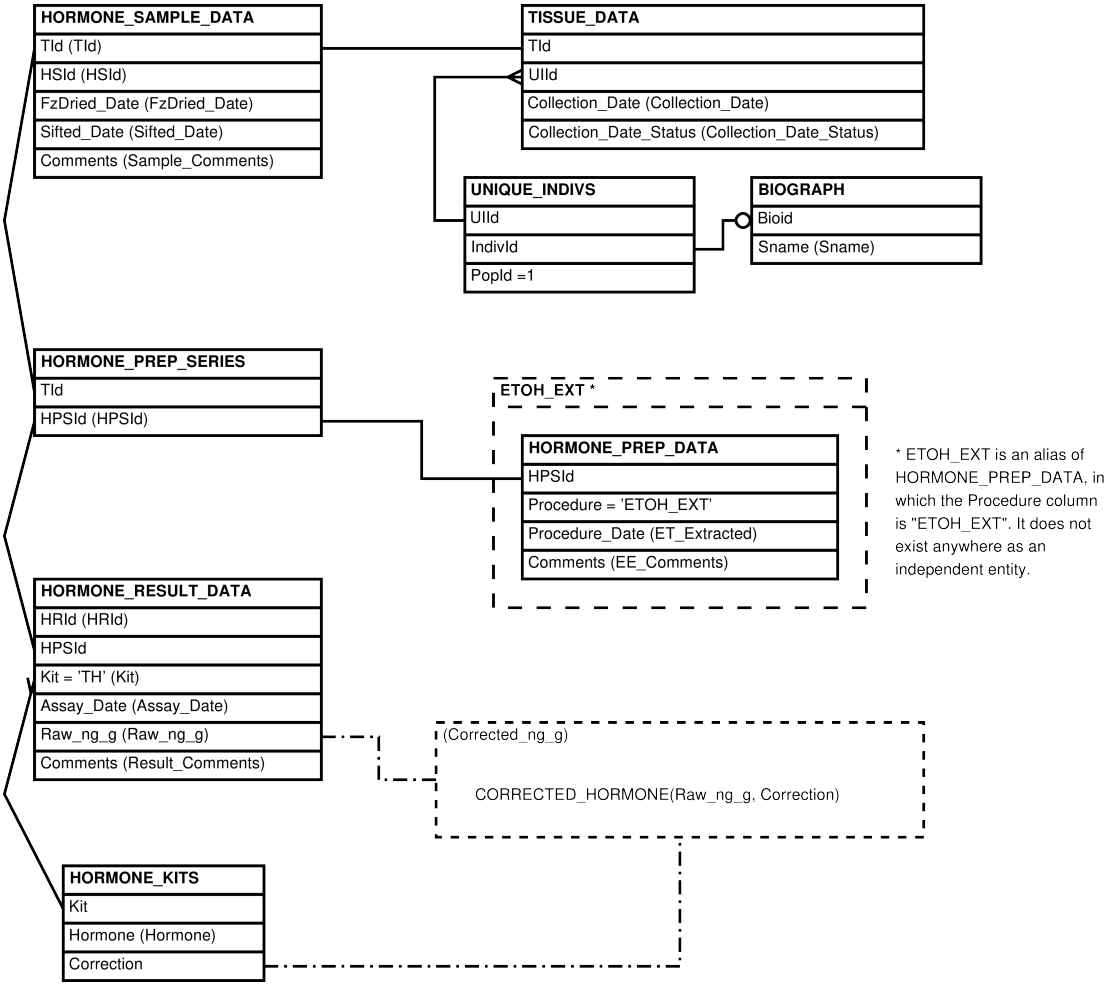

Figure 123: Entity Relationship Diagram of the THYROID_HORMONES View

3.51 The TISSUES View

```
SELECT tissue_data.tid AS tid
     , tissue_data.locid AS locid
     , locations.institution AS institution
     , locations.location AS location
     , local_1.localid AS localid_1
     , local_2.localid AS localid_2
     , tissue_data.uiid AS uiid
     , unique_indivs.popid AS popid
     , unique_indivs.individ AS individ
     , biograph.sname AS sname
     , tissue_data.name_on_tube AS name_on_tube
     , tissue_data.collection_date AS collection_date
     , tissue_data.collection_time AS collection_time
     , tissue_data.tissue_type AS tissue_type
     , tissue_data.storage_medium AS storage_medium
     , tissue_data.misid_status AS misid_status
     , tissue_data.collection_date_status AS collection_date_status
     , tissue_data.multi_indivs AS multi_indivs
     , COUNT(tissue_sources.*) AS tissue_sources
     , tissue_data.notes AS notes
  FROM tissue_data
  JOIN locations
    ON locations.locid = tissue_data.locid
  LEFT JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON biograph.bioid::text = unique_indivs.individ
       AND unique_indivs.popid = 1
  LEFT JOIN tissue_local_ids AS local_1
    ON local_1.tid = tissue_data.tid
       AND local_1.institution = 1
  LEFT JOIN tissue_local_ids AS local_2
    ON local_2.tid = tissue_data.tid
       AND local_2.institution = 2
  LEFT JOIN tissue_sources
    ON tissue_sources.tid = tissue_data.tid
  GROUP BY tissue_data.tid
         , tissue_data.locid
         , locations.institution
         , locations.location
         , local_1.localid
         , local_2.localid
         , tissue_data.uiid
         , unique_indivs.popid
         , unique_indivs.individ
         , biograph.sname
         , tissue_data.name_on_tube
         , tissue_data.collection_date
         , tissue_data.collection_time
         , tissue_data.tissue_type
         , tissue_data.storage_medium
         , tissue_data.misid_status
         , tissue_data.collection_date_status
         , tissue_data.multi_indivs
         , tissue_data.notes;
```

Figure 124: Query Defining the TISSUES View

If we could we would display here a diagram showing how the TISSUES view is constructed.

Figure 125: Entity Relationship Diagram of the TISSUES View

3.52 The TISSUES_HORMONES View

```
SELECT tissue_data.tid AS tid
     , tissue_data.locid AS locid
     , locations.institution AS institution
     , locations.location AS location
     , local_1.localid AS localid_1
     , local_2.localid AS localid_2
     , tissue_data.uiid AS uiid
     , unique_indivs.popid AS popid
     , unique_indivs.individ AS individ
     , biograph.sname AS sname
     , tissue_data.name_on_tube AS name_on_tube
     , tissue_data.collection_date AS collection_date
     , tissue_data.collection_time AS collection_time
     , tissue_data.tissue_type AS tissue_type
     , tissue_data.storage_medium AS storage_medium
     , tissue_data.misid_status AS misid_status
     , tissue_data.collection_date_status AS collection_date_status
     , tissue_data.multi_indivs AS multi_indivs
     , COUNT(tissue_sources.*) AS tissue_sources
     , tissue_data.notes AS notes
     , hormone_sample_data.hsid AS hsid
     , hormone_sample_data.fzdried_date AS fzdried_date
     , hormone_sample_data.sifted_date AS sifted_date
     , hormone_sample_data.avail_mass_g AS avail_mass_g
     , hormone_sample_data.avail_date AS avail_date
     , hormone_sample_data.comments AS comments
  FROM tissue_data
  JOIN locations
    ON locations.locid = tissue_data.locid
  LEFT JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON biograph.bioid::text = unique_indivs.individ
       AND unique_indivs.popid = 1
  LEFT JOIN tissue_local_ids AS local_1
    ON local_1.tid = tissue_data.tid
       AND local_1.institution = 1
  LEFT JOIN tissue_local_ids AS local_2
    ON local_2.tid = tissue_data.tid
       AND local_2.institution = 2
  LEFT JOIN tissue_sources
    ON tissue_sources.tid = tissue_data.tid
  LEFT JOIN hormone_sample_data
    ON hormone_sample_data.tid = tissue_data.tid
  GROUP BY tissue_data.tid
         , tissue_data.locid
         , locations.institution
         , locations.location
         , local_1.localid
         , local_2.localid
         , tissue_data.uiid
         , unique_indivs.popid
         , unique_indivs.individ
         , biograph.sname
         , tissue_data.name_on_tube
         , tissue_data.collection_date
         , tissue_data.collection_time
         , tissue_data.tissue_type
         , tissue_data.storage_medium
         , tissue_data.misid_status
         , tissue_data.collection_date_status
```

If we could we would display here a diagram showing how the TISSUES_HORMONES view is constructed.

Figure 127: Entity Relationship Diagram of the TISSUES_HORMONES View

3.53 The ULNA_STATS View

```
SELECT ulnas.dartid AS dartid
     , count(*) AS ulsamps
     , avg(ulnas.ullength) AS ullength_mean
     , stddev(ulnas.ullength) AS ullength_stddev
     , avg(ulnas.ulunadjusted) AS ulunadjusted_mean
     , stddev(ulnas.ulunadjusted) AS ulunadjusted_stddev
  FROM ulnas
  GROUP BY ulnas.dartid;
```

Figure 128: Query Defining the ULNA_STATS View

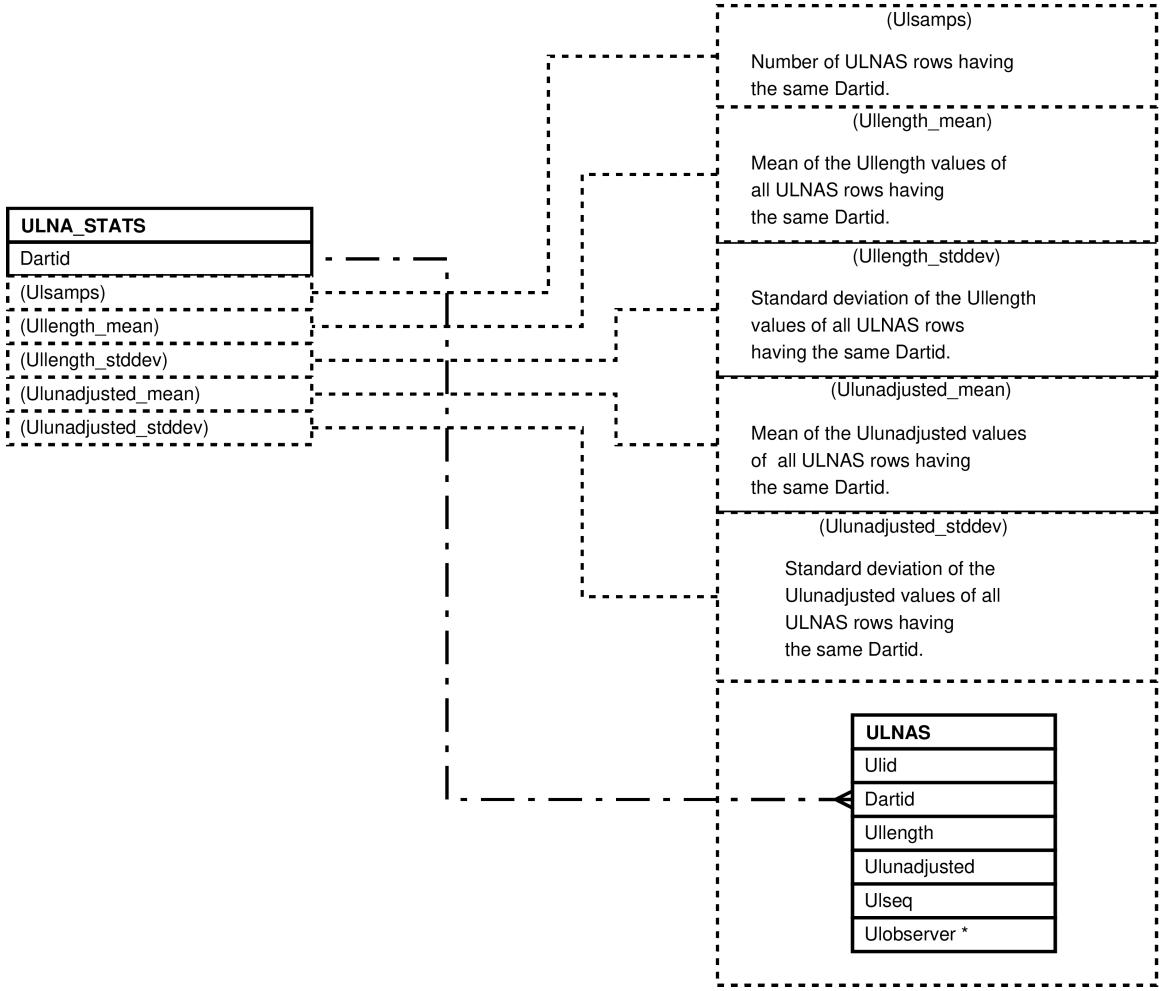

Figure 129: Entity Relationship Diagram of the ULNA_STATS View

3.54 The VAGINAL_PH_STATS View

```
SELECT vaginal_phs.dartid AS dartid
     , count(*) AS vpsamps
     , avg(vaginal_phs.ph) AS vp_mean
     , stddev(vaginal_phs.ph) AS vp_stddev
  FROM vaginal_phs
  GROUP BY vaginal_phs.dartid;
```

Figure 130: Query Defining the VAGINAL_PH_STATS View

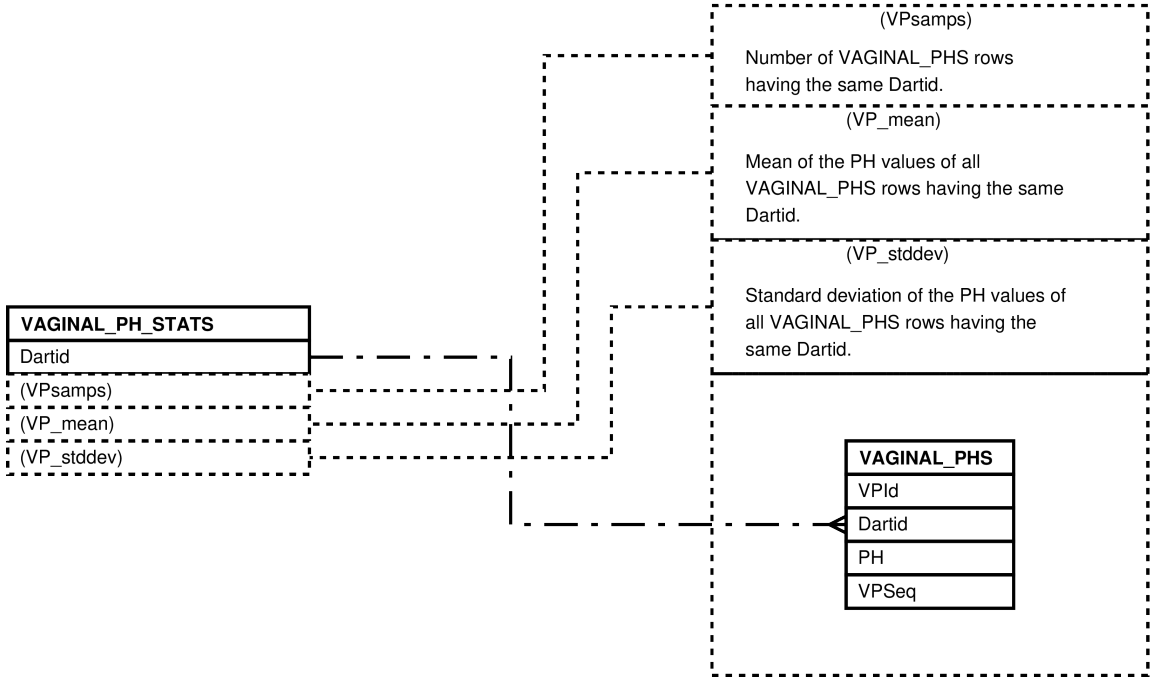

Figure 131: Entity Relationship Diagram of the VAGINAL_PH_STATS View

3.55 The WOUNDPATHOLOGIES View

```
WITH concat_observers AS (SELECT wprid
                               , string_agg(observer, '/' ORDER BY wpoid) as  ↩
                                  observers
                            FROM wp_observers
                            GROUP BY wprid)
SELECT wp_reports.wprid AS wprid
     , wp_reports.wid AS wid
     , wp_reports.date AS reportdate
     , wp_reports.time AS reporttime
     , concat_observers.observers AS observers
     , wp_reports.sname AS sname
     , wp_reports.grp AS grp
     , wp_reports.observercomments AS observercomments
     , wp_reports.reportstate AS reportstate
     , wp_details.wpdid AS wpdid
     , wp_details.woundpathcode AS woundpathcode
     , wp_details.cluster AS cluster
     , wp_details.maxdimension AS maxdimension
     , wp_details.impairslocomotion AS impairslocomotion
     , wp_details.infectionsigns AS infectionsigns
     , wp_details.notes AS detailnotes
     , wp_affectedparts.wpaid AS wpaid
     , wp_affectedparts.bodypart AS bodypart
     , bodyparts.bodyside AS bodyside
     , bodyparts.innerouter AS innerouter
     , bodyparts.bodyregion AS bodyregion
     , wp_affectedparts.quantity_affecting_part AS quantity_affecting_part
  FROM wp_reports
  LEFT JOIN concat_observers
    ON concat_observers.wprid = wp_reports.wprid
  LEFT JOIN wp_details
    ON wp_details.wprid = wp_reports.wprid
  LEFT JOIN wp_affectedparts
    ON wp_affectedparts.wpdid = wp_details.wpdid
  LEFT JOIN bodyparts
    ON bodyparts.bpid = wp_affectedparts.bodypart;
```

Figure 132: Query Defining the WOUNDSPATHOLOGIES View

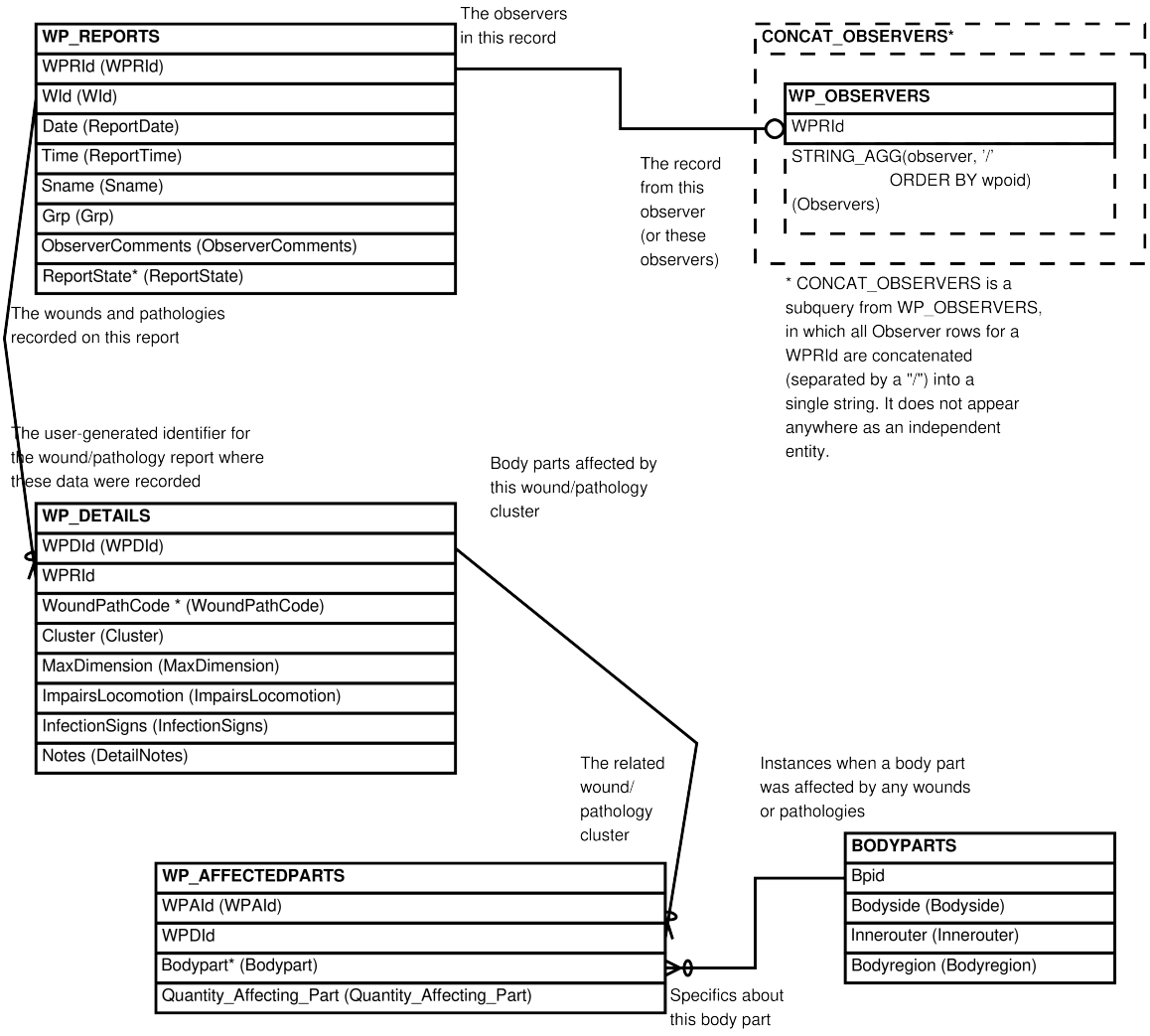

Figure 133: Entity Relationship Diagram of the WOUNDSPATHOLOGIES View

3.56 The WP_DETAILS_AFFECTEDPARTS View

```
SELECT wp_details.wpdid                              AS wpdid
     , wp_reports.wprid                              AS wprid
     , wp_reports.wid                                AS wid
     , wp_details.woundpathcode                      AS woundpathcode
     , wp_details.cluster                            AS cluster
     , wp_details.maxdimension                       AS maxdimension
     , wp_details.impairslocomotion                  AS impairslocomotion
     , wp_details.infectionsigns                     AS infectionsigns
     , wp_details.notes                              AS detailnotes
     , wp_affectedparts.wpaid                        AS wpaid
     , wp_affectedparts.wpdid                        AS bodypart_wpdid
     , wp_affectedparts.bodypart                     AS bodypart
     , bodyparts.bodyside                            AS bodyside
     , bodyparts.innerouter                          AS innerouter
     , bodyparts.bodyregion                          AS bodyregion
     , wp_affectedparts.quantity_affecting_part AS quantity_affecting_part
  FROM wp_reports
  JOIN wp_details
    ON wp_details.wprid = wp_reports.wprid
  LEFT JOIN wp_affectedparts
    ON wp_affectedparts.wpdid = wp_details.wpdid
  LEFT JOIN bodyparts
    ON bodyparts.bpid = wp_affectedparts.bodypart;
```

Figure 134: Query Defining the WP_DETAILS_AFFECTEDPARTS View

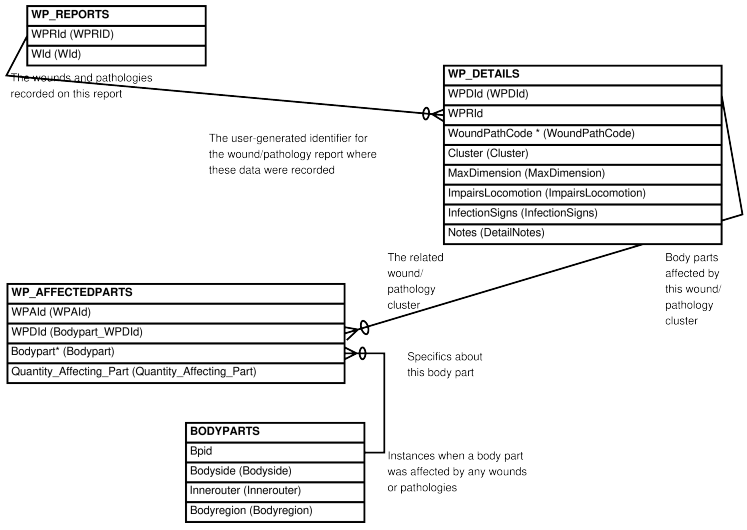

Figure 135: Entity Relationship Diagram of the WP_DETAILS_AFFECTEDPARTS View

3.57 The WP_HEALS View

```
WITH concat_observers AS (SELECT wprid
                               , string_agg(observer, '/' ORDER BY wpoid) as  ↩
                                  observers
                            FROM wp_observers
                            GROUP BY wprid)
SELECT wp_reports.wprid AS wprid
     , wp_reports.wid AS wid
     , wp_reports.date AS reportdate
     , wp_reports.time AS reporttime
     , concat_observers.observers AS observers
     , wp_reports.sname AS sname
     , wp_reports.grp AS grp
     , wp_reports.observercomments AS observercomments
     , wp_reports.reportstate AS reportstate
     , wp_details.wpdid AS wpdid
     , wp_details.woundpathcode AS woundpathcode
     , wp_details.cluster AS cluster
     , wp_details.maxdimension AS maxdimension
     , wp_details.impairslocomotion AS impairslocomotion
     , wp_details.infectionsigns AS infectionsigns
     , wp_details.notes AS detailnotes
     , wp_affectedparts.wpaid AS wpaid
     , wp_affectedparts.bodypart AS bodypart
     , bodyparts.bodyside AS bodyside
     , bodyparts.innerouter AS innerouter
     , bodyparts.bodyregion AS bodyregion
     , wp_affectedparts.quantity_affecting_part AS quantity_affecting_part
     , wp_healupdates.wphid AS wphid
     , wp_healupdates.date AS healdate
     , wp_healupdates.healstatus AS healstatus
     , wp_healupdates.notes AS healnotes
  FROM wp_healupdates
  LEFT JOIN wp_affectedparts
    ON wp_affectedparts.wpaid = wp_healupdates.wpaid
  LEFT JOIN bodyparts
    ON bodyparts.bpid = wp_affectedparts.bodypart
  LEFT JOIN wp_details
    ON wp_details.wpdid = COALESCE(wp_affectedparts.wpdid, wp_healupdates.wpdid)
  LEFT JOIN wp_reports
    ON wp_reports.wprid = COALESCE(wp_details.wprid, wp_healupdates.wprid)
  LEFT JOIN concat_observers
    ON concat_observers.wprid = wp_reports.wprid;
```

Figure 136: Query Defining the WP_HEALS View

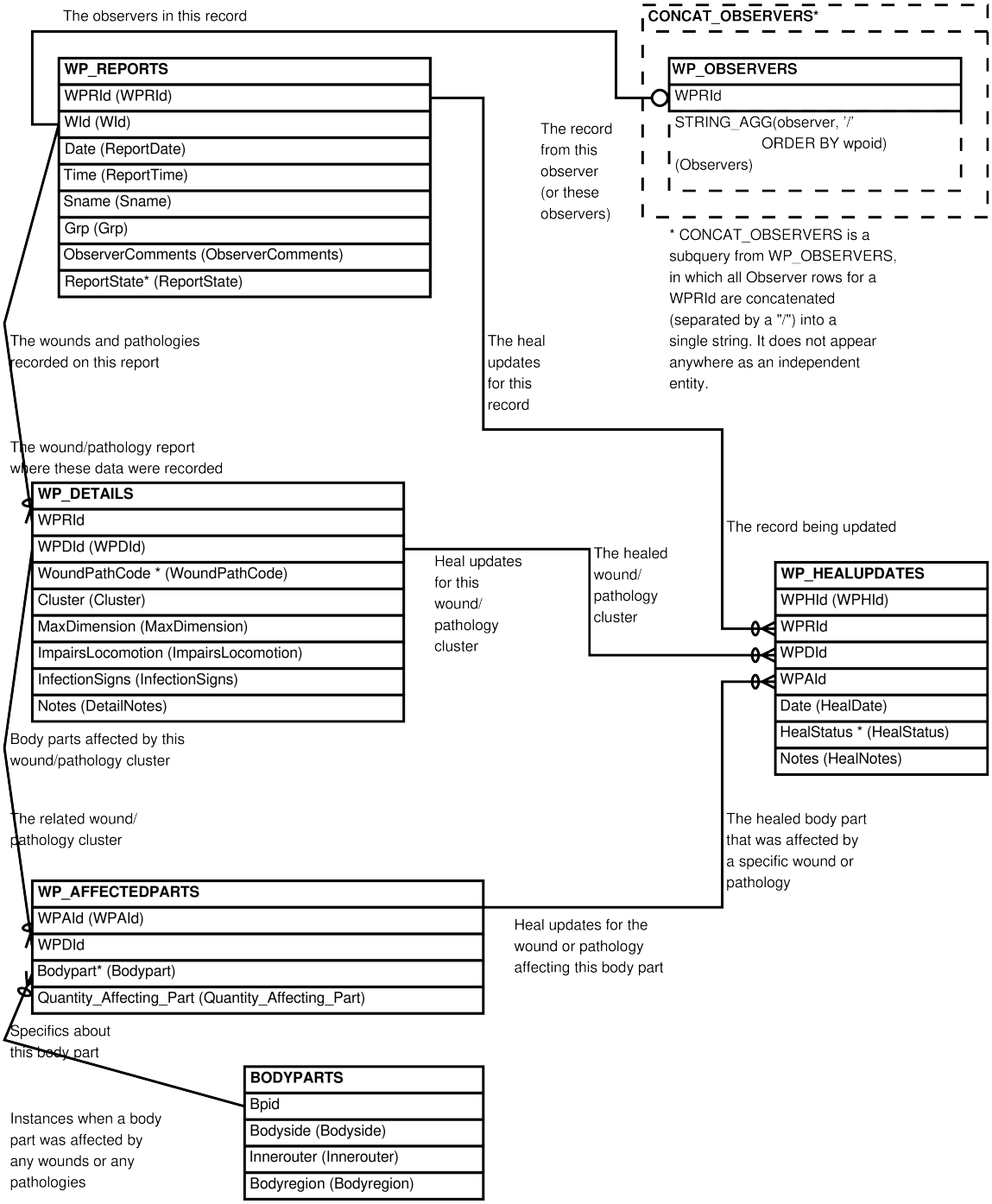

Figure 137: Entity Relationship Diagram of the WP_HEALS View, Overall

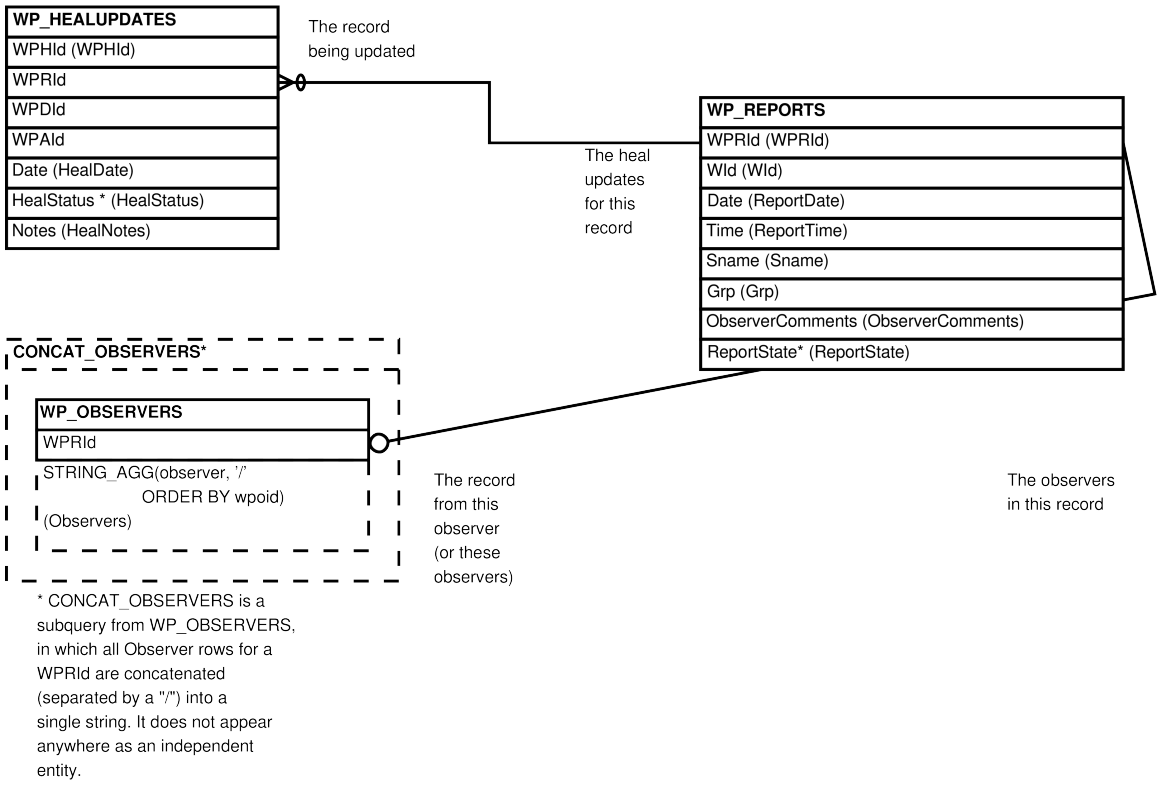

Figure 138: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology report

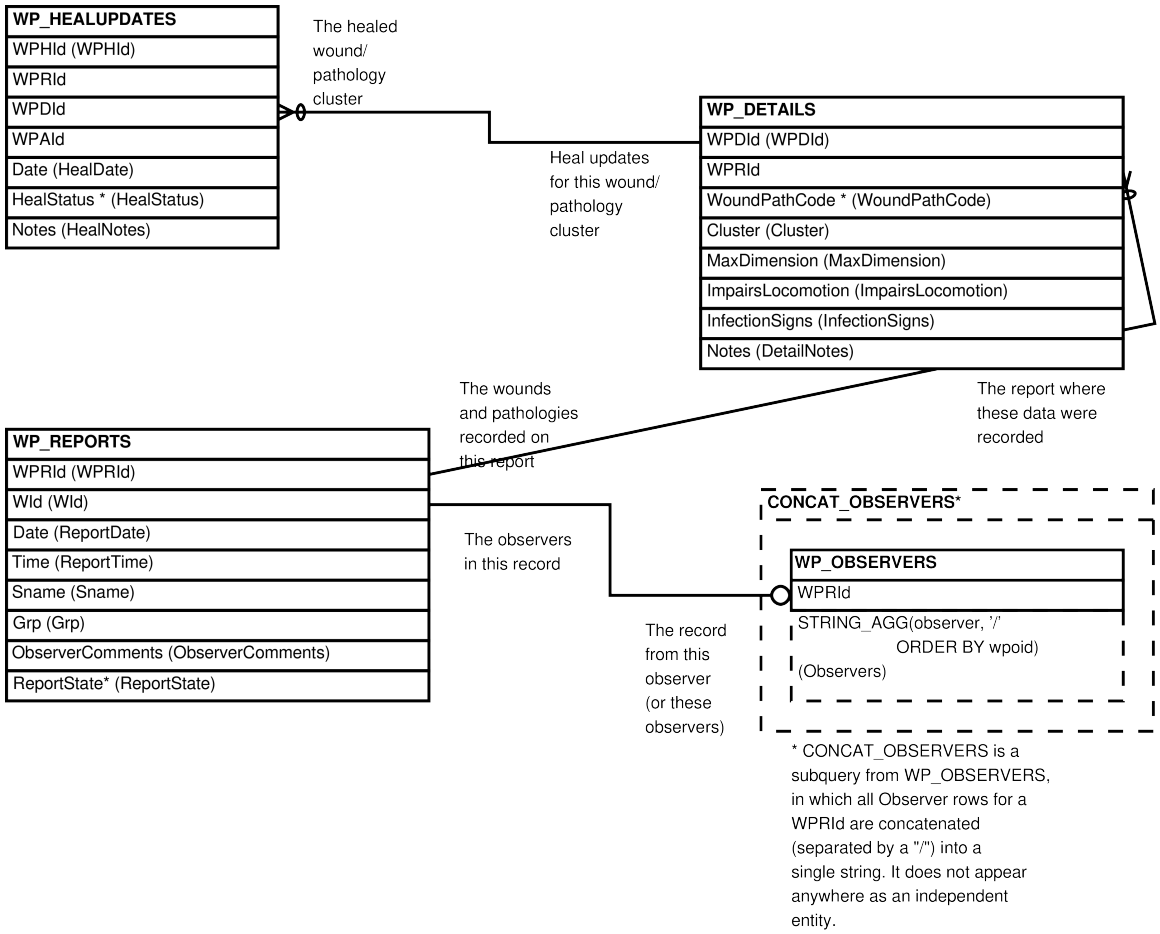

Figure 139: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology cluster

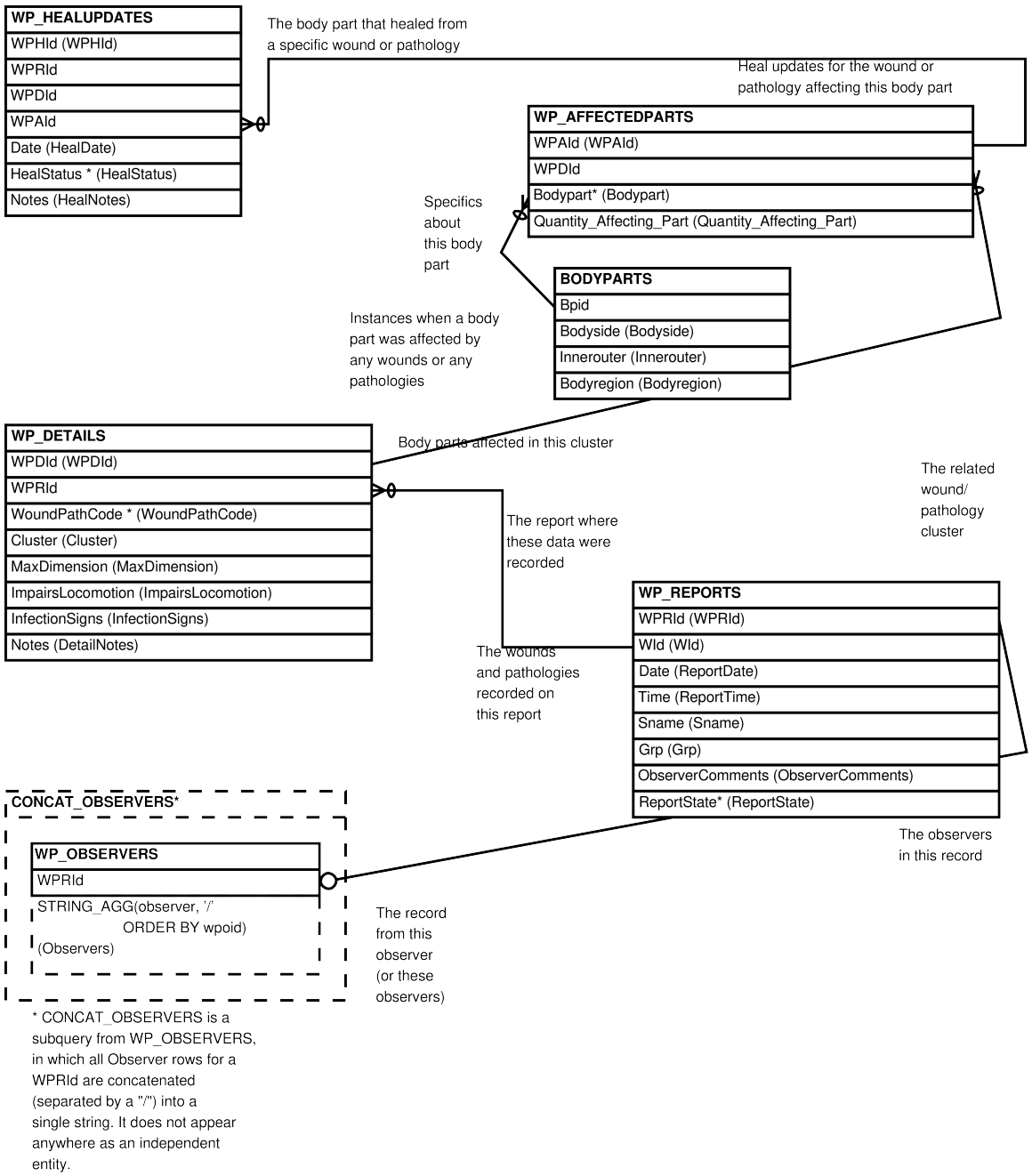

Figure 140: Entity Relationship Diagram of the WP_HEALS View for rows with an update to an affected body part

3.58 The WP_REPORTS_OBSERVERS View

```
WITH concat_observers AS (SELECT wprid
                               , string_agg(observer, '/' ORDER BY wpoid) as ↩
                                  observers
                            FROM wp_observers
                            GROUP BY wprid)
SELECT wp_reports.wprid             AS wprid
     , wp_reports.wid               AS wid
     , wp_reports.date              AS date
     , wp_reports.time              AS time
     , concat_observers.observers   AS observers
     , wp_reports.sname             AS sname
     , wp_reports.grp               AS grp
     , wp_reports.observercomments AS observercomments
     , wp_reports.reportstate       AS reportstate
  FROM wp_reports
  LEFT JOIN concat_observers
    ON concat_observers.wprid = wp_reports.wprid;
```

Figure 141: Query Defining the WP_REPORTS_OBSERVERS View

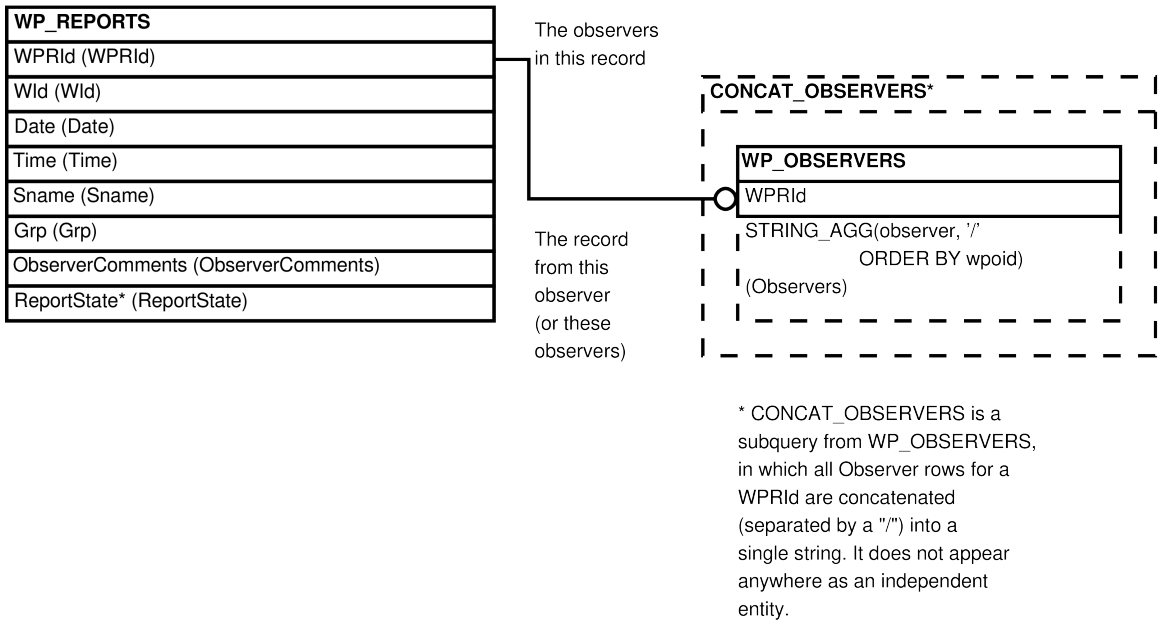

Figure 142: Entity Relationship Diagram of the WP_REPORTS_OBSERVERS View

4 Views Which Add Gid To Tables

In addition to the above views there are a number of views which produce the group of a referenced individual as of a pertinent date. These views are all named after the table from which they are derived, with the addition of the suffixed `_GRP`. They are nearly identical to the table from which they derive, differing only by the addition of a column named `Grp`.

The only operation allowed on these views is SELECT. INSERT, UPDATE, and DELETE are not allowed.

4.1 The BIRTH_GRP View

```
SELECT biograph.*
     , members.grp AS grp
  FROM members, biograph
  WHERE members.sname = biograph.sname
        AND members.date = CAST(biograph.birth AS DATE);
```

Figure 143: Query Defining the BIRTH_GRP View

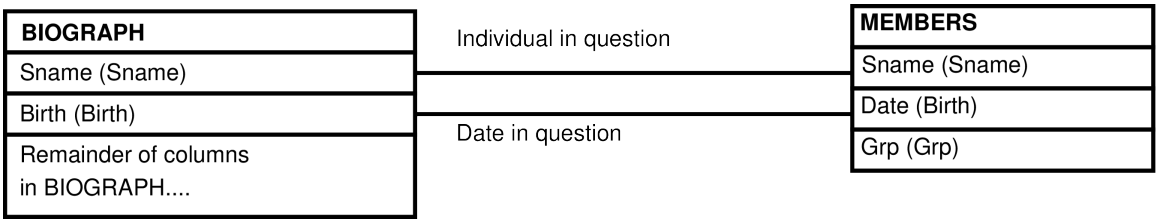

Figure 144: Entity Relationship Diagram of the BIRTH_GRP View

4.2 The ENTRYDATE_GRP View

```
SELECT biograph.*
     , members.grp AS grp
  FROM members, biograph
  WHERE members.sname = biograph.sname
        AND members.date = CAST(biograph.entrydate AS DATE);
```

Figure 145: Query Defining the ENTRYDATE_GRP View

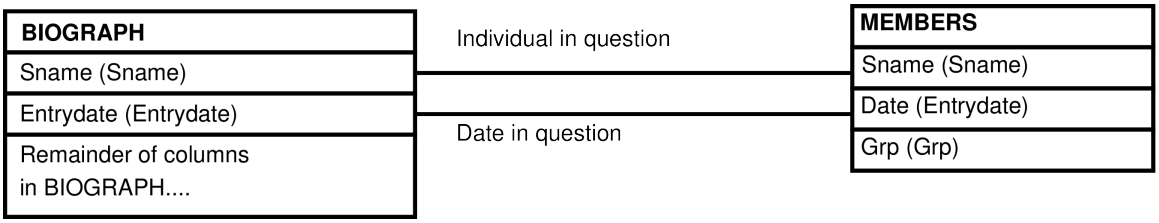

Figure 146: Entity Relationship Diagram of the ENTRYDATE_GRP View

4.3 The STATDATE_GRP View

```
SELECT biograph.*
     , members.grp AS grp
  FROM members, biograph
  WHERE members.sname = biograph.sname
        AND members.date = CAST(biograph.statdate AS DATE);
```

Figure 147: Query Defining the STATDATE_GRP View

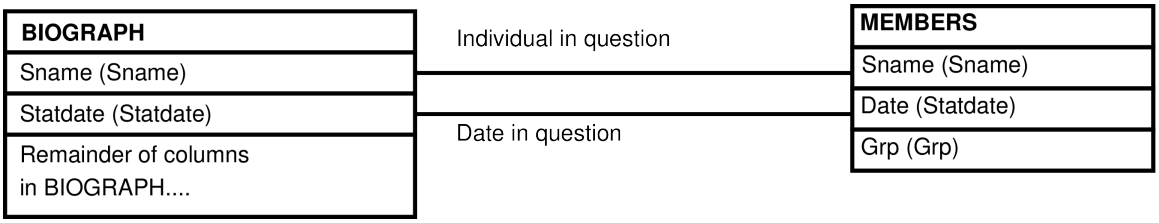

Figure 148: Entity Relationship Diagram of the STATDATE_GRP View

4.4 The CONSORTDATES_GRP View

```
SELECT consortdates.*
     , members.grp AS grp
  FROM members, consortdates
  WHERE members.sname = consortdates.sname
        AND members.date = CAST(consortdates.consorted AS DATE);
```

Figure 149: Query Defining the CONSORTDATES_GRP View

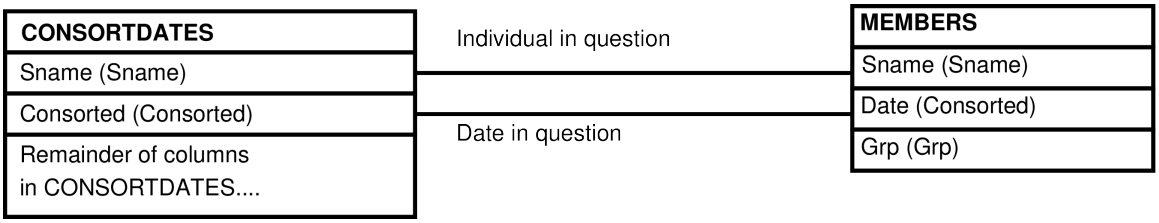

Figure 150: Entity Relationship Diagram of the CONSORTDATES_GRP View

4.5 The CYCGAPDAYS_GRP View

```
SELECT cycgapdays.*
     , members.grp AS grp
  FROM members, cycgapdays
  WHERE members.sname = cycgapdays.sname
        AND members.date = CAST(cycgapdays.date AS DATE);
```

Figure 151: Query Defining the CYCGAPDAYS_GRP View

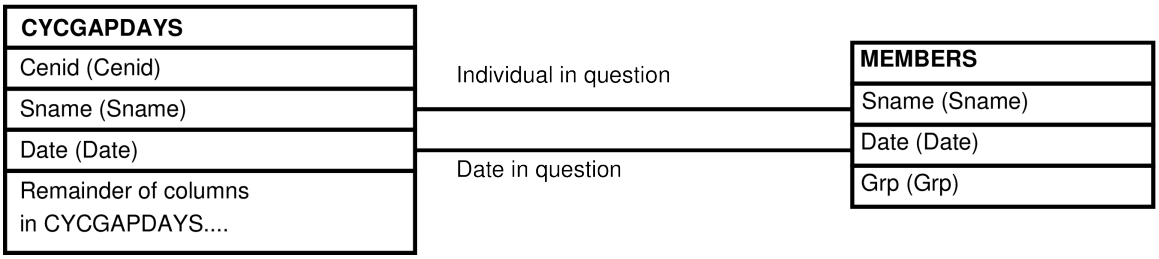

Figure 152: Entity Relationship Diagram of the CYCGAPDAYS_GRP View

4.6 The CYCGAPS_GRP View

```
SELECT cycgaps.*
     , members.grp AS grp
  FROM members, cycgaps
  WHERE members.sname = cycgaps.sname
        AND members.date = CAST(cycgaps.date AS DATE);
```

Figure 153: Query Defining the CYCGAPS_GRP View

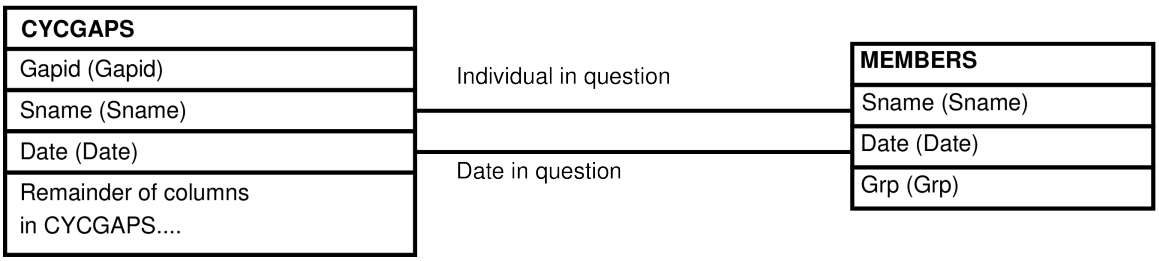

Figure 154: Entity Relationship Diagram of the CYCGAPS_GRP View

4.7 The CYCSTATS_GRP View

```
SELECT cycstats.*
     , members.grp AS grp
  FROM members, cycstats
  WHERE members.sname = cycstats.sname
        AND members.date = CAST(cycstats.date AS DATE);
```

Figure 155: Query Defining the CYCSTATS_GRP View

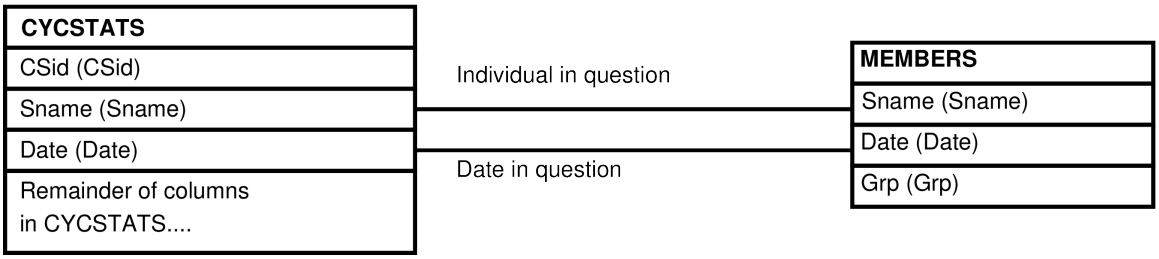

Figure 156: Entity Relationship Diagram of the CYCSTATS_GRP View

4.8 The DARTINGS_GRP View

```
SELECT dartings.*
     , members.grp AS grp
  FROM members, dartings
  WHERE members.sname = dartings.sname
        AND members.date = CAST(dartings.date AS DATE);
```

Figure 157: Query Defining the DARTINGS_GRP View

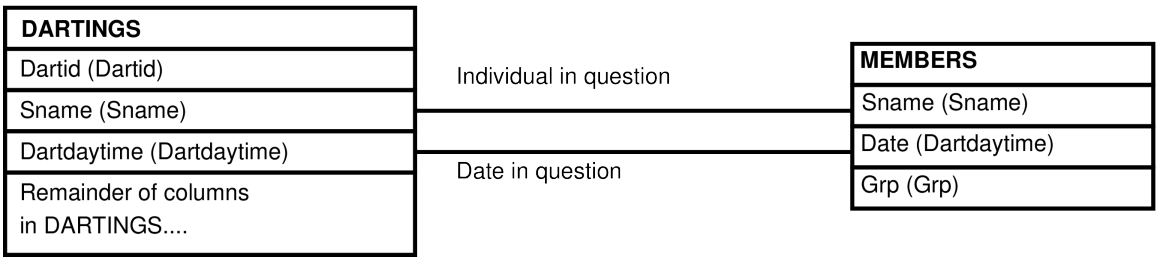

Figure 158: Entity Relationship Diagram of the DARTINGS_GRP View

4.9 The DISPERSEDATES_GRP View

```
SELECT dispersedates.*
     , members.grp AS grp
  FROM members, dispersedates
  WHERE members.sname = dispersedates.sname
        AND members.date = CAST(dispersedates.dispersed AS DATE);
```

Figure 159: Query Defining the DISPERSEDATES_GRP View

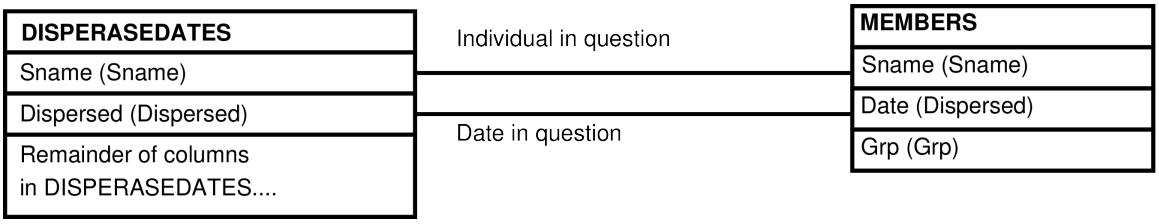

Figure 160: Entity Relationship Diagram of the DISPERSEDATES_GRP View

4.10 The MATUREDATES_GRP View

```
SELECT maturedates.*
     , members.grp AS grp
  FROM members, maturedates
  WHERE members.sname = maturedates.sname
        AND members.date = CAST(maturedates.matured AS DATE);
```

Figure 161: Query Defining the MATUREDATES_GRP View

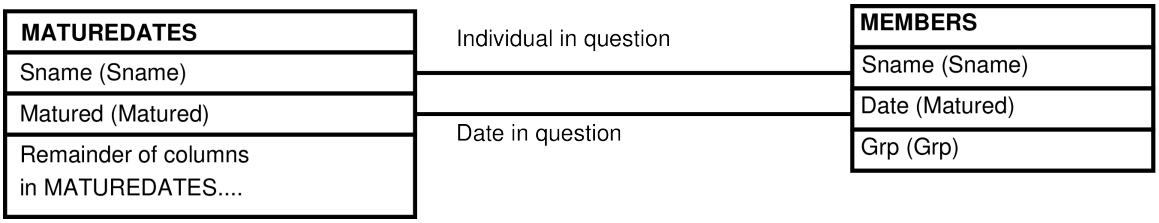

Figure 162: Entity Relationship Diagram of the MATUREDATES_GRP View

4.11 The MDINTERVALS_GRP View

```
SELECT mdintervals.*
     , members.grp AS grp
  FROM members, mdintervals
  WHERE members.sname = mdintervals.sname
        AND members.date = CAST(mdintervals.date AS DATE);
```

Figure 163: Query Defining the MDINTERVALS_GRP View

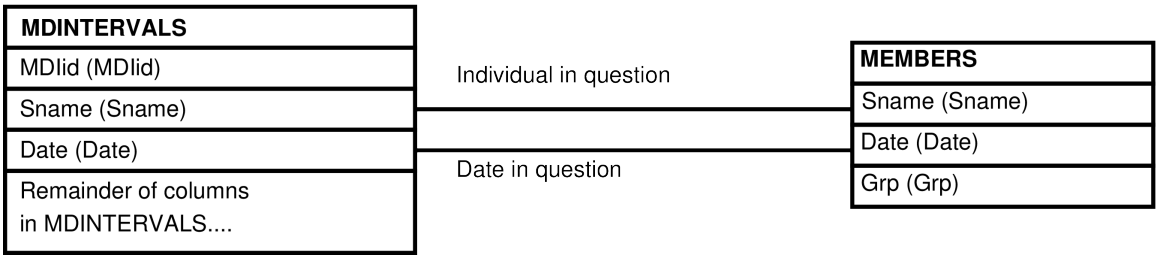

Figure 164: Entity Relationship Diagram of the MDINTERVALS_GRP View

4.12 The MMINTERVALS_GRP View

```
SELECT mmintervals.*
     , members.grp AS grp
  FROM members, mmintervals
  WHERE members.sname = mmintervals.sname
        AND members.date = CAST(mmintervals.date AS DATE);
```

Figure 165: Query Defining the MMINTERVALS_GRP View

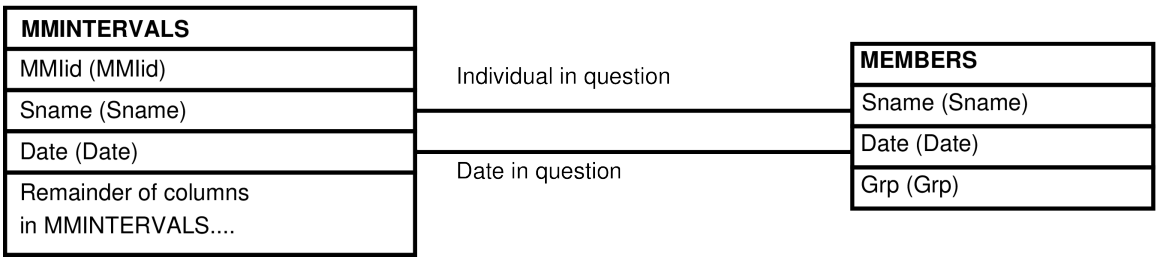

Figure 166: Entity Relationship Diagram of the MMINTERVALS_GRP View

4.13 The RANKDATES_GRP View

```
SELECT rankdates.*
     , members.grp AS grp
  FROM members, rankdates
  WHERE members.sname = rankdates.sname
        AND members.date = CAST(rankdates.ranked AS DATE);
```

Figure 167: Query Defining the RANKDATES_GRP View

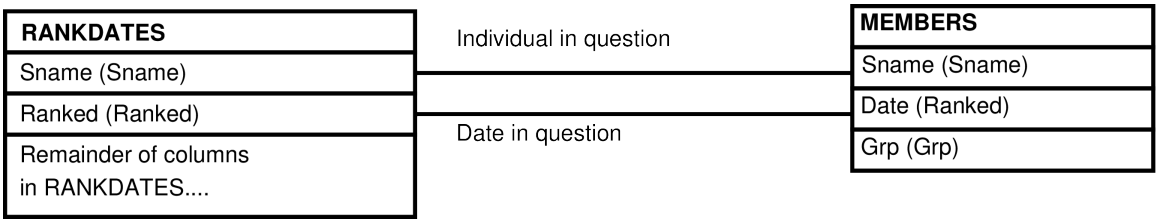

Figure 168: Entity Relationship Diagram of the RANKDATES_GRP View

4.14 The REPSTATS_GRP View

```
SELECT repstats.*
     , members.grp AS grp
  FROM members, repstats
  WHERE members.sname = repstats.sname
        AND members.date = CAST(repstats.date AS DATE);
```

Figure 169: Query Defining the REPSTATS_GRP View

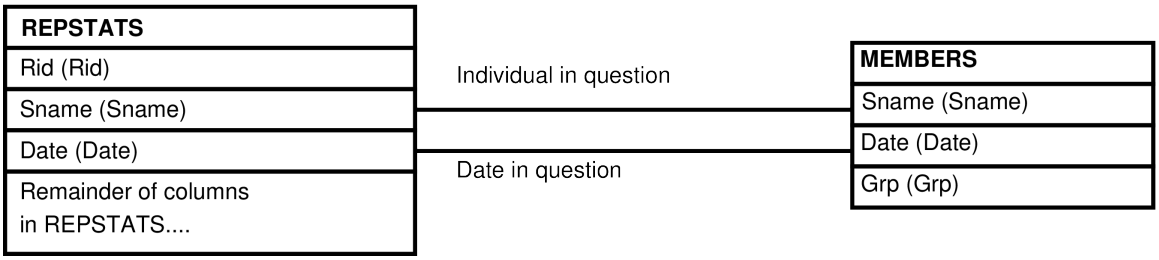

Figure 170: Entity Relationship Diagram of the REPSTATS_GRP View