The Babase Pocket Reference Guide

A Technical Specification Summary
This material is based upon work supported by the National Science Foundation under Grant Nos. 0323553 and 0323596.
## COLLABORATORS

<table>
<thead>
<tr>
<th>ACTION</th>
<th>NAME</th>
<th>DATE</th>
<th>SIGNATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRITTEN BY</td>
<td>Karl O. Pinc, PhD. Jeanne Altmann, PhD. Susan C. Alberts, and Leah Gerber</td>
<td>July 25, 2019</td>
<td></td>
</tr>
<tr>
<td>ER Diagram layout and conversion to Dia</td>
<td>Leah Gerber</td>
<td>July 25, 2019</td>
<td></td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Anne Hubbard</td>
<td>July 25, 2019</td>
<td></td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Karl Pinc</td>
<td>July 25, 2019</td>
<td></td>
</tr>
</tbody>
</table>

## REVISION HISTORY

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>DATE</th>
<th>DESCRIPTION</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Contents

1 Babase Summarized 1

2 The Babase ER Diagrams 1

3 The Babase Views 22
   3.1 The ACTOR_ACTEES View .................................................. 23
   3.2 The ANESTH_STATS View .................................................. 24
   3.3 The BODYTEMP_STATS View ............................................. 25
   3.4 The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views ............. 26
   3.5 The CHEST_STATS View ................................................... 27
   3.6 The CROWNRUMP_STATS View ......................................... 28
   3.7 The CYCLES_SEXSKINS and CYCLES_SEXSKINS_SORTED Views .... 28
   3.8 The CYCPOINTS_CYCLES and CYCPOINTS_CYCLES_SORTED Views .. 29
   3.9 The DSAMPLES View ..................................................... 31
   3.10 The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views .......... 32
   3.11 The DENT_CODES View .................................................. 34
   3.12 The DENT_SITES View ................................................... 37
   3.13 The INTERACT and INTERACT_SORTED Views ......................... 39
   3.14 The MATERNITIES View .................................................. 40
   3.15 The MIN_MAXS View ..................................................... 42
   3.16 The MIN_MAXS_SORTED View ............................................ 44
   3.17 The MPI_EVENTS View ................................................... 47
   3.18 The MTD_CYCLES View .................................................. 49
   3.19 The PARENTS View ...................................................... 51
   3.20 The PCSKINS_SORTED View ............................................ 52
   3.21 The PCV_STATS View ..................................................... 53
   3.22 The POINTS and POINTS_SORTED Views .............................. 54
   3.23 The POTENTIAL_DADS View .......................................... 56
   3.24 The PROPORTIONAL_RANKS View .................................... 56
   3.25 The QUADS View ....................................................... 61
   3.26 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_SORTED Views ... 62
   3.27 The SWERB view .......................................................... 65
   3.28 The SWERB_DEPARTS view .............................................. 67
   3.29 The SWERB_LOC_GPS_XY view ....................................... 67
   3.30 The SWERB_LOCS view .................................................. 68
   3.31 The SWERB_UPLOAD view ............................................... 69
   3.32 The ULNA_STATS View .................................................. 69
3.33 The TESTES_ARC_STATS View ................................................................. 72
3.34 The TESTES_DIAM_STATS View ............................................................. 75
3.35 The WPDETAILS_AFFECTEDPARTS View ............................................... 77
3.36 The WP_HEALS View ................................................................. 78
3.37 The WPREPORTS_OBSERVERS View .................................................. 83

4 Views Which Add Gid To Tables ......................................................... 83
   4.1 The BIRTH_GRP View ............................................................... 84
   4.2 The ENTRYDATE_GRP View ........................................................ 84
   4.3 The STATDATE_GRP View ........................................................... 85
   4.4 The CONSORTDATES_GRP View ..................................................... 85
   4.5 The CYCGAPDAYS_GRP View ....................................................... 86
   4.6 The CYCGAPS_GRP View ............................................................. 86
   4.7 The CYCSTATS_GRP View ........................................................... 87
   4.8 The DARTINGS_GRP View ............................................................ 87
   4.9 The DISPERSEDATES_GRP View .................................................... 88
   4.10 The MATUREDATES_GRP View ....................................................... 88
   4.11 The MDINTERVALS_GRP View ....................................................... 89
   4.12 The MMINTERVALS_GRP View ....................................................... 89
   4.13 The PCSKINS_GRP View .............................................................. 90
   4.14 The RANKDATES_GRP View .......................................................... 90
   4.15 The REPSTATS_GRP View ............................................................ 91
List of Figures

1. Key to the Babase Entity Relationship Diagrams ................................................. 1
2. Babase Group Membership Entity Relationship Diagram ........................................ 3
3. Babase Life Events Entity Relationship Diagram .................................................. 5
4. Babase Sexual Cycle Entity Relationship Diagram ................................................ 8
5. Babase Sexual Cycle Day-To-Day Tables Entity Relationship Diagram ..................... 9
6. Babase Social Interactions Entity Relationship Diagram ........................................ 10
7. Babase Multiparty Interactions Entity Relationship Diagram ................................... 11
8. Babase Darting Logistics and Morphology Entity and Relationship Diagram ............ 12
9. Babase Darting Physiology Entity and Relationship Diagram .................................. 13
10. Babase Darting Samples Entity and Relationship Diagram .................................... 14
11. Babase Darting Teeth and Ticks Entity and Relationship Diagram ........................ 15
12. Babase Physical Traits Genetic Hybrid Score Data Entity Relationship Diagram .......... 16
13. Babase Physical Traits Wounds and Pathologies Data Entity Relationship Diagram .... 17
14. Babase SWERB Core Tables Entity Relationship Diagram .................................... 18
15. Babase SWERB Grove/Waterhole Location Tables Entity Relationship Diagram ....... 19
17. Babase WeatherHawk Data Entity Relationship Diagram ...................................... 21
18. Warning Sub-System Entity Relationship Diagram ............................................. 22
19. Query Defining the ACTOR_ACTEES View ......................................................... 23
20. Entity Relationship Diagram of the ACTOR_ACTEES View .................................... 24
21. Query Defining the ANESTH_STATS View ......................................................... 24
22. Entity Relationship Diagram of the ANESTH_STATS View .................................... 25
23. Query Defining the BODYTEMP_STATS View ...................................................... 25
24. Entity Relationship Diagram of the BODYTEMP_STATS View ................................ 26
25. Query Defining the CENSUS_DEMOG View ....................................................... 26
26. Entity Relationship Diagram of the CENSUS_DEMOG View ................................ 26
27. Query Defining the CHEST_STATS View ............................................................. 27
28. Entity Relationship Diagram of the CHEST_STATS View ...................................... 27
29. Query Defining the CROWNRUMP_STATS View ................................................. 28
30. Entity Relationship Diagram of the CROWNRUMP_STATS View ........................... 28
31. Query Defining the CYCLES_SEXSKINS View ..................................................... 28
32. Entity Relationship Diagram of the CYCLES_SEXSKINS View ............................. 28
33. Query Defining the CYCPOINTS_CYCLES View ................................................. 29
34. Entity Relationship Diagram of the CYCPOINTS_CYCLES View .......................... 29
35. Query Defining the DSAMPLES View ............................................................... 31
36. Query Defining the DEMOG_CENSUS View ...................................................... 32
37. Entity Relationship Diagram of the DEMOG_CENSUS View .................................. 32
<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>Query Defining the DENT_CODES View</td>
</tr>
<tr>
<td>39</td>
<td>Entity Relationship Diagram of the DENT_CODES View</td>
</tr>
<tr>
<td>40</td>
<td>Query Defining the DENT_SITES View</td>
</tr>
<tr>
<td>41</td>
<td>Entity Relationship Diagram of the DENT_SITES View</td>
</tr>
<tr>
<td>42</td>
<td>Query Defining the INTERACT View</td>
</tr>
<tr>
<td>43</td>
<td>Entity Relationship Diagram of the INTERACT View</td>
</tr>
<tr>
<td>44</td>
<td>Query Defining the MATERNITIES View</td>
</tr>
<tr>
<td>45</td>
<td>Entity Relationship Diagram of the MATERNITIES View</td>
</tr>
<tr>
<td>46</td>
<td>Query Defining the MIN_MAXS View</td>
</tr>
<tr>
<td>47</td>
<td>Entity Relationship Diagram of the MIN_MAXS View</td>
</tr>
<tr>
<td>48</td>
<td>Query Defining the MIN_MAXS_SORTED View</td>
</tr>
<tr>
<td>49</td>
<td>Entity Relationship Diagram of the MIN_MAXS_SORTED View</td>
</tr>
<tr>
<td>50</td>
<td>Query Defining the MPI_EVENTS View</td>
</tr>
<tr>
<td>51</td>
<td>Entity Relationship Diagram of the MPI_EVENTS View</td>
</tr>
<tr>
<td>52</td>
<td>Query Defining the MTD_CYCLES View</td>
</tr>
<tr>
<td>53</td>
<td>Entity Relationship Diagram of the MTD_CYCLES View</td>
</tr>
<tr>
<td>54</td>
<td>Query Defining the PARENTS View</td>
</tr>
<tr>
<td>55</td>
<td>Entity Relationship Diagram of the PARENTS View</td>
</tr>
<tr>
<td>56</td>
<td>Query Defining the PCSKINS_SORTED View</td>
</tr>
<tr>
<td>57</td>
<td>Entity Relationship Diagram of the PCSKINS_SORTED View</td>
</tr>
<tr>
<td>58</td>
<td>Query Defining the PCV_STATS View</td>
</tr>
<tr>
<td>59</td>
<td>Entity Relationship Diagram of the PCV_STATS View</td>
</tr>
<tr>
<td>60</td>
<td>Query Defining the POINTS View</td>
</tr>
<tr>
<td>61</td>
<td>Entity Relationship Diagram of the POINTS View</td>
</tr>
<tr>
<td>62</td>
<td>Query Defining the POTENTIAL_DADS View</td>
</tr>
<tr>
<td>63</td>
<td>Entity Relationship Diagram of the foundation of the POTENTIAL_DADS View</td>
</tr>
<tr>
<td>64</td>
<td>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</td>
</tr>
<tr>
<td>65</td>
<td>Entity Relationship Diagram of that portion of the POTENTIAL_DADS View having easily computed columns</td>
</tr>
<tr>
<td>66</td>
<td>Entity Relationship Diagram of that portion of the POTENTIAL_DADS View involving social interactions</td>
</tr>
<tr>
<td>67</td>
<td>Query Defining the PROPORTIONAL_RANKS View</td>
</tr>
<tr>
<td>68</td>
<td>Entity Relationship Diagram of the PROPORTIONAL_RANKS View</td>
</tr>
<tr>
<td>69</td>
<td>Query Defining the QUADS View</td>
</tr>
<tr>
<td>70</td>
<td>Entity Relationship Diagram of the QUADS View</td>
</tr>
<tr>
<td>71</td>
<td>Query Defining the SEXSKINS_CYCLES View</td>
</tr>
<tr>
<td>72</td>
<td>Entity Relationship Diagram of the SEXSKINS_CYCLES View</td>
</tr>
<tr>
<td>73</td>
<td>Query Defining the SWERB View</td>
</tr>
<tr>
<td>74</td>
<td>Entity Relationship Diagram of the SWERB View</td>
</tr>
<tr>
<td>75</td>
<td>Query Defining the SWERB_DEPARTS View</td>
</tr>
</tbody>
</table>
115 Entity Relationship Diagram of the DISPERSEDATES_GRP View ........................................... 88
116 Query Defining the MATUREDATES_GRP View ................................................................. 88
117 Entity Relationship Diagram of the MATUREDATES_GRP View ........................................ 88
118 Query Defining the MDINTERVALS_GRP View ................................................................. 89
119 Entity Relationship Diagram of the MDINTERVALS_GRP View ......................................... 89
120 Query Defining the MMINTERVALS_GRP View ................................................................. 89
121 Entity Relationship Diagram of the MMINTERVALS_GRP View ......................................... 89
122 Query Defining the PCSKINS_GRP View ............................................................................. 90
123 Entity Relationship Diagram of the PCSKINS_GRP View ................................................... 90
124 Query Defining the RANKDATES_GRP View ..................................................................... 90
125 Entity Relationship Diagram of the RANKDATES_GRP View ............................................. 90
126 Query Defining the REPSTATS_GRP View .......................................................................... 91
127 Entity Relationship Diagram of the REPSTATS_GRP View ................................................. 91

List of Tables

1 The Main Babase Tables ........................................................................................................ 2
2 The Warning Sub-System Tables ....................................................................................... 3
3 The Babase Support Tables ............................................................................................... 4
4 The Warning Sub-System Support Tables ......................................................................... 5
5 The Babase Views ............................................................................................................. 6
6 The table_GRP Views ........................................................................................................ 7
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

The BABASE Database

![Diagram of Babase ER Diagrams](image)

Figure 1: Key to the Babase Entity Relationship Diagrams

\[\text{At this time of this writing only males have data entered into RANKDATES in Babase.}\]
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTERNATE_SNAMES</td>
<td>rescinded sname</td>
</tr>
<tr>
<td>BIOGRAPH</td>
<td>animal, including fetuses</td>
</tr>
<tr>
<td>CENSUS</td>
<td>day each individual is (or is not) observed in a group</td>
</tr>
<tr>
<td>CONSORTDATES</td>
<td>male who has a known first consortship</td>
</tr>
<tr>
<td>DEMOG</td>
<td>mention of an individual’s presence in a group within a field textual note</td>
</tr>
<tr>
<td>DISPERSEDATES</td>
<td>male who has left his maternal study group</td>
</tr>
<tr>
<td>GROUPS</td>
<td>group (including solitary males)</td>
</tr>
<tr>
<td>MATUREDATES</td>
<td>individual who is sexually mature</td>
</tr>
<tr>
<td>RANKDATES</td>
<td>individual who has attained adult rank</td>
</tr>
<tr>
<td>DAD_DATA</td>
<td>offspring having a paternity analysis</td>
</tr>
<tr>
<td>MEMBERS</td>
<td>day each individual is alive</td>
</tr>
<tr>
<td>RANKS</td>
<td>month each individual is ranked in each group</td>
</tr>
<tr>
<td>HYBRIDGENE_ANALYSES</td>
<td>analysis of genetic hybrid scores</td>
</tr>
<tr>
<td>HYBRIDGENE_SCORES</td>
<td>genetic hybrid score for an individual from an analysis</td>
</tr>
<tr>
<td>WP_AFFECTEDPARTS</td>
<td>body part affected by a specific wound/pathology</td>
</tr>
<tr>
<td>WP_DETAILS</td>
<td>wound or pathology cluster indicated on a report</td>
</tr>
<tr>
<td>WP_HEALUPDATES</td>
<td>update on progress of wound/pathology healing</td>
</tr>
<tr>
<td>WP_REPORTS</td>
<td>wound/pathology report</td>
</tr>
<tr>
<td>CYC GAPS</td>
<td>female for each initiation or cessation of a continuous period of observation</td>
</tr>
<tr>
<td>CYCLES</td>
<td>female’s cycle (complete or not)</td>
</tr>
<tr>
<td>CYCPOINTS</td>
<td>Mdate (menses), Tdate (turgesence onset), or Ddate (deturgesence onset) date of each female</td>
</tr>
<tr>
<td>PCSKINS</td>
<td>PCS color of each female</td>
</tr>
<tr>
<td>PREGS</td>
<td>time a female becomes pregnant</td>
</tr>
<tr>
<td>SEXSKINS</td>
<td>sexskin measurement of each female</td>
</tr>
<tr>
<td>CYCGAPDAYS</td>
<td>female for each day within a period during which there is not continuous observation</td>
</tr>
<tr>
<td>CYCSTATS</td>
<td>day each female is cycling -- by M, T and Ddates</td>
</tr>
<tr>
<td>MDINTERVALS</td>
<td>day each female is cycling and is between M and Ddates</td>
</tr>
<tr>
<td>MMINTERVALS</td>
<td>day each female is cycling -- by Mdates</td>
</tr>
<tr>
<td>Table</td>
<td>One row for each</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
</tr>
<tr>
<td>INTEGRITY QUERIES in Babase:</td>
<td>query used to discover data integrity problems</td>
</tr>
<tr>
<td>INTEGRITY WARNINGS in Babase:</td>
<td>data integrity problem discovered by the warning sub-system</td>
</tr>
</tbody>
</table>

Table 2: The Warning Sub-System Tables

![Figure 2: Babase Group Membership Entity Relationship Diagram](image)

Figure 2: Babase Group Membership Entity Relationship Diagram
### General Support Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Id Column</th>
<th>Related Column(s)</th>
<th>One entry for every possible choice of...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BODYPARTS in Babase:</strong></td>
<td>Bodypart</td>
<td>TICKS in Babase::Bodypart in Babase:: BODYPARTS in Babase::Bodyregion in Babase:: WP_AFFECTEDPARTS in Babase::Bodypart in Babase:</td>
<td>part of the body</td>
</tr>
<tr>
<td><strong>OBSERVERS in Babase:</strong></td>
<td>Initials</td>
<td>SAMPLES in Babase::Observer in Babase:: WREADINGS in Babase::WRperson in Babase:: RGSETUPS in Babase::RGSPerson in Babase:: CROWNRUMPS in Babase::Croserver in Babase:: CHESTS in Babase::Chobserver in Babase:: ULNAS in Babase::Ulongserver in Babase:: HUMERUSES in Babase::Huobserver in Babase:: SWERB_OBSERVERS in Babase::Observer in Babase::</td>
<td>person who records information</td>
</tr>
<tr>
<td><strong>OBSERVER_ROLES in Babase:</strong></td>
<td>Initials</td>
<td>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::</td>
<td>way in which a person can be involved in the data collection process</td>
</tr>
<tr>
<td><strong>UNKSNAMES in Babase:</strong></td>
<td>Unksname</td>
<td>NEIGHBORS in Babase::Unksname in Babase:: and the SWERB_UPLOAD in Babase:: view</td>
<td>problem in identifying neighbor of focal during point sampling or in identifying a lone male in a SWERB other group observation</td>
</tr>
</tbody>
</table>

### Group Membership and Life Events

<table>
<thead>
<tr>
<th>Table</th>
<th>Id Column</th>
<th>Related Column(s)</th>
<th>One entry for every possible choice of...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BSTATUSES in Babase:</strong></td>
<td>Bstatus</td>
<td>BIOGRAPH in Babase::Bstatus in Babase:</td>
<td>birthday estimation accuracy</td>
</tr>
<tr>
<td><strong>CONFIDENCES in Babase:</strong></td>
<td>Confidence</td>
<td>BIOGRAPH in Babase::DcauseNatureConfidence in Babase:: BIOGRAPH in Babase::DcauseAgentConfidence in Babase:: DISPERSEDATES in Babase::Dispconfidence in Babase:: BIOGRAPH in Babase::Matgrpcertainty confidence in Babase::</td>
<td>degree of certitude in nature of death, agent of death, disperse date assignment, or maternal group assignment</td>
</tr>
<tr>
<td><strong>DAD.SOFTWARE in Babase:</strong></td>
<td>Software</td>
<td>DAD_DATA in Babase::Software in Babase::</td>
<td>software package used to perform genetic paternity analysis</td>
</tr>
<tr>
<td><strong>DCAUSES in Babase:</strong></td>
<td>Dease</td>
<td>BIOGRAPH in Babase::</td>
<td>cause of death</td>
</tr>
<tr>
<td><strong>DEATHNATURES in Babase:</strong></td>
<td>Nature</td>
<td>DCAUSES in Babase::Nature in Babase::</td>
<td>reason for death</td>
</tr>
<tr>
<td><strong>BIRTHDAYS in Babase:</strong></td>
<td>Birth</td>
<td>BIOGRAPH in Babase::Birth in Babase::</td>
<td>birth estimation accuracy</td>
</tr>
</tbody>
</table>

---

*The Babase Pocket Reference Guide*
Table 4: The Warning Sub-System Support Tables

![Diagram of Babase Life Events Entity Relationship Diagram]

Figure 3: Babase Life Events Entity Relationship Diagram
<table>
<thead>
<tr>
<th>View</th>
<th>One row for each</th>
<th>Purpose</th>
<th>Tables/Views used</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENSUS_DEMOG in Babase:</td>
<td>CENSUS in Babase: row</td>
<td>Maintenance of CENSUS in Babase: rows that are extended with DEMOG in Babase: information.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>CENSUS_DEMOG_SORTED in Babase:</td>
<td>CENSUS in Babase: row</td>
<td>Maintenance of CENSUS_DEMOG in Babase: rows in a pre-sorted fashion.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>CYCPOINTS_CYCLES in Babase:</td>
<td>CYCPOINTS in Babase: row</td>
<td>Maintenance of CYCPOINTS in Babase: rows that are extended with CYCLES in Babase: information.</td>
<td>CYCLES in Babase:, CYCPOINTS in Babase:</td>
</tr>
<tr>
<td>CYCPOINTS_CYCLES_SORTED in Babase:</td>
<td>CYCPOINTS in Babase: row</td>
<td>The CYCPOINTS_CYCLES in Babase: view sorted by CYCLES in Babase:, by CYCPOINTS in Babase:, by Date in Babase:.</td>
<td>CYCLES in Babase:, CYCPOINTS in Babase:</td>
</tr>
<tr>
<td>DEMOG_CENSUS in Babase:</td>
<td>DEMOG in Babase: row</td>
<td>Maintenance of DEMOG in Babase: rows.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>DEMOG_CENSUS_SORTED in Babase:</td>
<td>CENSUS in Babase: row</td>
<td>Maintenance of DEMOG_CENSUS in Babase: rows in a pre-sorted fashion.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>GROUPS_HISTORY in Babase:</td>
<td>GROUPS in Babase: row</td>
<td>Depiction of GROUPS in Babase: rows in a more human-readable format.</td>
<td>GROUPS in Babase:</td>
</tr>
<tr>
<td>PARENTS in Babase:</td>
<td>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 DAD_DATA in Babase: with a record of the individual’s father -- with a non-NULLDad_consensus in Babase:</td>
<td>Easy access to parental information.</td>
<td>BIOGRAPH in Babase:, MATERNITIES in Babase:, DAD_DATA in Babase:, MEMBERS in Babase:</td>
</tr>
<tr>
<td>POTENTIAL_DADS in Babase:</td>
<td></td>
<td>(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</td>
<td>MATERNITIES in Babase:, MEMBERS in Babase: (multiple times), ACTOR_ACTEES in Babase:</td>
</tr>
<tr>
<td>PROPORTIONAL_RANKS in Babase:</td>
<td>RANKS in Babase: row</td>
<td>Automatic calculation of proportional ranks from the ordinal ranks in RANKS in Babase:.</td>
<td>RANKS in Babase:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical Traits</th>
<th>One row for each</th>
<th>Purpose</th>
<th>Tables/Views used</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>View</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>BIRTH_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>ENTRYDATE_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>STATDATE_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSORTDATES in Babase:</td>
<td>CONSORTDATES_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCGAPDAYS in Babase:</td>
<td>CYCGAPDAYS_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCGAPS in Babase:</td>
<td>CYCGAPS_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCSTATS in Babase:</td>
<td>CYCSTATS_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DARTINGS in Babase:</td>
<td>DARTINGS_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISPERSEDATES in Babase:</td>
<td>DISPERSEDATES_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATUREDATES in Babase:</td>
<td>MATUREDATES_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDINTERVALS in Babase:</td>
<td>MDINTERVALS_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMINTERVALS in Babase:</td>
<td>MMINTERVALS_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCSKINS in Babase:</td>
<td>PCSKINS_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RANKDATES in Babase:</td>
<td>RANKDATES_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPSTATS in Babase:</td>
<td>REPSTATS_GRP in Babase:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: The table_GRP Views
Figure 4: Babase Sexual Cycle Entity Relationship Diagram
Figure 5: Babase Sexual Cycle Day-To-Day Tables Entity Relationship Diagram
Figure 6: Babase Social Interactions Entity Relationship Diagram
Figure 7: Babase Multiparty Interactions Entity Relationship Diagram
Figure 8: Babase Darting Logistics and Morphology Entity and Relationship Diagram
Figure 9: Babase Darting Physiology Entity and Relationship Diagram
Figure 10: Babase Darting Samples Entity and Relationship Diagram
Figure 11: Babase Darting Teeth and Ticks Entity and Relationship Diagram
Figure 12: Babase Physical Traits Genetic Hybrid Score Data Entity Relationship Diagram
Figure 13: Babase Physical Traits Wounds and Pathologies Data Entity Relationship Diagram
Figure 14: Babase SWERB Core Tables Entity Relationship Diagram
Figure 15: Babase SWERB Grove/Waterhole Location Tables Entity Relationship Diagram
Figure 16: Babase Manual Weather Data Entity Relationship Diagram
Figure 17: Babase WeatherHawk Data 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

```
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
  , (SELECT actorms.grp
        FROM members AS actorms
        WHERE actorms.sname = actor.sname
          AND actorms.date = interact_data.date) AS actor_grp
  , actee.partid AS acteeid
  , COALESCE(actee.sname, '998'::CHAR(3)) AS actee
  , (SELECT acteems.grp
        FROM members AS acteems
        WHERE acteems.sname = actee.sname
          AND acteems.date = interact_data.date) AS actee_grp
  , interact_data.handwritten AS handwritten
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');
```

Figure 19: Query Defining 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 21: Query Defining 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 22: Entity Relationship Diagram of the ANESTH_STATS View

Figure 23: Query Defining the BODYTEMP_STATS View
3.4 The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views

```sql
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 24: Entity Relationship Diagram of the BODYTEMP_STATS View

Figure 25: Query Defining the CENSUS_DEMOG View

Figure 26: Entity Relationship Diagram of the CENSUS_DEMOG View
### 3.5 The CHEST_STATS View

```sql
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 27: Query Defining the CHEST_STATS View

Figure 28: Entity Relationship Diagram of the CHEST_STATS View
3.6 The CROWNRUMP_STATS View

```sql
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 29: Query Defining the CROWNRUMP_STATS View

![Diagram of the CROWNRUMP_STATS View]

3.7 The CYCLES_SEXSKINS and CYCLES_SEXSKINS_SORTED Views

```sql
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
FROM cycles LEFT OUTER JOIN sexskins ON (cycles.cid = sexskins.cid);
```

Figure 31: Query Defining 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 33: Query Defining the CYCPOINTS_CYCLES View

Figure 34: Entity Relationship Diagram of the CYCPOINTS_CYCLES View
3.9 The DSAMPLES View

```sql
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 vaginalsaw,
       cerv_swabs.num AS cervicalswab,
       fecal_formalin.num AS fecal_formalin,
       vaginal_ph.num AS vaginal_ph,
       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
```
3.10 The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views

```sql
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 36: Query Defining the DEMOG_CENSUS View

![Entity Relationship Diagram of the DEMOG_CENSUS View](image)

Figure 37: Entity Relationship Diagram of the DEMOG_CENSUS View
3.11 The DENT_CODES View

```sql
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
    , rui2.rui2tstate AS rui2tstate
    , rui2.rui2tcondition AS rui2tcondition
    , lui1.lui1tstate AS lui1tstate
    , lui1.lui1tcondition AS lui1tcondition
    , lui2.lui2tstate AS lui2tstate
    , lui2.lui2tcondition AS lui2tcondition
    , lci.lclctstate AS lclctstate
    , lci.lclctcondition AS lclctcondition
    , lpl1.lpl1tstate AS lpl1tstate
    , lpl1.lpl1tcondition AS lpl1tcondition
    , lcl.lclctstate AS lclctstate
    , lcl.lclctcondition AS lclctcondition
    , lli2.lli2tstate AS lli2tstate
    , lli2.lli2tcondition AS lli2tcondition
    , llm1.llm1tstate AS llm1tstate
    , llm1.llm1tcondition AS llm1tcondition
    , llm2.llm2tstate AS llm2tstate
    , llm2.llm2tcondition AS llm2tcondition
    , llm3.llm3tstate AS llm3tstate
    , llm3.llm3tcondition AS llm3tcondition
    , llc.llctstate AS llctstate
    , llc.llctcondition AS llctcondition
    , rup1.rup1tstate AS rup1tstate
    , rup1.rup1tcondition AS rup1tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp1.rlp1tcondition AS rlp1tcondition
FROM teeth
    WHERE teeth.tooth = 'drlc')
LEFT OUTER JOIN (SELECT teeth.dartid AS rlm2dartid
                , rlm2.rlm2tcondition AS rlm2tcondition
FROM teeth
    WHERE teeth.tooth = 'rlm2')
LEFT OUTER JOIN (SELECT teeth.dartid AS rlm1dartid
                , rlm1.rlm1tcondition AS rlm1tcondition
FROM teeth
    WHERE teeth.tooth = 'rlm1')
LEFT OUTER JOIN (SELECT teeth.dartid AS rlp2dartid
                , rlp2.rlp2tstate AS rlp2tstate
FROM teeth
    WHERE teeth.tooth = 'rlp2')
LEFT OUTER JOIN (SELECT teeth.dartid AS rlp1dartid
                , rlp1.rlp1tstate AS rlp1tstate
FROM teeth
    WHERE teeth.tooth = 'rlp1')
LEFT OUTER JOIN (SELECT teeth.dartid AS rli2dartid
                , rli2.rli2tstate AS rli2tstate
FROM teeth
    WHERE teeth.tooth = 'rli2')
LEFT OUTER JOIN (SELECT teeth.dartid AS rli1dartid
                , rli1.rli1tstate AS rli1tstate
FROM teeth
    WHERE teeth.tooth = 'rli1')
LEFT OUTER JOIN (SELECT teeth.dartid AS drum1dartid
                , drum1.drum1tstate AS drum1tstate
FROM teeth
    WHERE teeth.tooth = 'rum1')
LEFT OUTER JOIN (SELECT teeth.dartid AS rum3dartid
                , rum3.rum3tstate AS rum3tstate
FROM teeth
    WHERE teeth.tooth = 'rum3')
LEFT OUTER JOIN (SELECT teeth.dartid AS rum2dartid
                , rum2.rum2tstate AS rum2tstate
FROM teeth
    WHERE teeth.tooth = 'rum2')
LEFT OUTER JOIN (SELECT teeth.dartid AS rum1dartid
                , rum1.rum1tstate AS rum1tstate
FROM teeth
    WHERE teeth.tooth = 'rum1')
WHERE teeth.tooth = 'drlc';
```

Figure 38: Query Defining the DENT_CODES View
Figure 39: Entity Relationship Diagram of the DENT_CODES View
3.12 The DENT_SITES View

SELECT teethdartids.dartid AS dartid , s1.sl1tstate AS sl1tstate , s1.sl1tcondition AS sl1tcondition , s1.sl1deciduous AS sl1deciduous , s2.sl2tstate AS sl2tstate , s2.sl2tcondition AS sl2tcondition , s2.sl2deciduous AS sl2deciduous , s3.sl3tstate AS sl3tstate , s3.sl3tcondition AS sl3tcondition , s3.sl3deciduous AS sl3deciduous , s4.sl4tstate AS sl4tstate , s4.sl4tcondition AS sl4tcondition , s4.sl4deciduous AS sl4deciduous , s5.sl5tstate AS sl5tstate , s5.sl5tcondition AS sl5tcondition , s5.sl5deciduous AS sl5deciduous , s6.sl6tstate AS sl6tstate , s6.sl6tcondition AS sl6tcondition , s6.sl6deciduous AS sl6deciduous , s7.sl7tstate AS sl7tstate , s7.sl7tcondition AS sl7tcondition , s7.sl7deciduous AS sl7deciduous , s8.sl8tstate AS sl8tstate , s8.sl8tcondition AS sl8tcondition , s8.sl8deciduous AS sl8deciduous , s9.sl9tstate AS sl9tstate , s9.sl9tcondition AS sl9tcondition , s9.sl9deciduous AS sl9deciduous , s10.sl10tstate AS sl10tstate , s10.sl10tcondition AS sl10tcondition , s10.sl10deciduous AS sl10deciduous , s11.sl11tstate AS sl11tstate , s11.sl11tcondition AS sl11tcondition , s11.sl11deciduous AS sl11deciduous , s12.sl12tstate AS sl12tstate , s12.sl12tcondition AS sl12tcondition , s12.sl12deciduous AS sl12deciduous , s13.sl13tstate AS sl13tstate , s13.sl13tcondition AS sl13tcondition , s13.sl13deciduous AS sl13deciduous , s14.sl14tstate AS sl14tstate , s14.sl14tcondition AS sl14tcondition , s14.sl14deciduous AS sl14deciduous , s15.sl15tstate AS sl15tstate , s15.sl15tcondition AS sl15tcondition , s15.sl15deciduous AS sl15deciduous , s16.sl16tstate AS sl16tstate , s16.sl16tcondition AS sl16tcondition , s16.sl16deciduous AS sl16deciduous , s17.sl17tstate AS sl17tstate , s17.sl17tcondition AS sl17tcondition , s17.sl17deciduous AS sl17deciduous , s18.sl18tstate AS sl18tstate , s18.sl18tcondition AS sl18tcondition , s18.sl18deciduous AS sl18deciduous , s19.sl19tstate AS sl19tstate , s19.sl19tcondition AS sl19tcondition , s19.sl19deciduous AS sl19deciduous , s20.sl20tstate AS sl20tstate , s20.sl20tcondition AS sl20tcondition , s20.sl20deciduous AS sl20deciduous
Figure 41: Entity Relationship Diagram of the DENT_SITES View
3.13 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
FROM interact_data
JOIN acts
    ON (acts.act = interact_data.act);
```

Figure 42: Query Defining the INTERACT View

<table>
<thead>
<tr>
<th>INTERACT_DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>iid</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>iid</td>
</tr>
</tbody>
</table>

Figure 43: Entity Relationship Diagram of the INTERACT View
### 3.14 The MATERNITIES View

```sql
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.edeate 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 44: Query Defining the MATERNITIES View
Figure 45: Entity Relationship Diagram of the MATERNITIES View

+ Although a join on this column alone returns multiple rows, because there is another join on a different column only 1 row matches all the criteria. (The combination of Sname and Date is unique.)
3.15 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 46: Query Defining the MIN_MAXS View
Figure 47: Entity Relationship Diagram of the MIN_MAXS View
3.16 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 48: Query Defining the MIN_MAXS_SORTED View
Figure 49: Entity Relationship Diagram of the MIN_MAXS_SORTED View
3.17 The MPI_EVENTS View

```sql
SELECT mpis.mpiid AS mpiid,
       mpis.date AS date,
       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.mpipid = mpi_data.mpidid AND actor.role = 'R')
LEFT OUTER JOIN mpi_parts AS actee
    ON (actee.mpipid = mpi_data.mpidid AND actee.role = 'E');
```

Figure 50: Query Defining the MPI_EVENTS View
Figure 51: Entity Relationship Diagram of the MPI_EVENTS View
3.18 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 52: Query Defining the MTD_CYCLES View
Figure 53: Entity Relationship Diagram of the MTD_CYCLES View
3.19 The PARENTS View

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

Figure 54: Query Defining the PARENTS View
3.20 The PCSKINS_SORTED View

```sql
SELECT pcskins.pcsid AS pcsid
     , pcskins.sname AS sname
     , pcskins.date AS date
     , pcskins.color AS color
FROM pcskins
ORDER BY sname, date;
```

Figure 55: Entity Relationship Diagram of the PARENTS View

- A BIOGRAPH row does not appear in this view unless there is either a related DAD_DATA row or a related MATERNITIES row.

- A MEMBERS row appears in this view only when related to both a DAD_DATA and a MATERNITIES row.

Figure 56: Query Defining the PCSKINS_SORTED View
### 3.21 The PCV_STATS View

```sql
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 58: Query Defining the PCV_STATS View

Figure 59: Entity Relationship Diagram of the PCV_STATS View
3.22 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 60: Query Defining the POINTS View

<table>
<thead>
<tr>
<th>POINT_DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pntid (Pntid)</td>
</tr>
<tr>
<td>Sid (Sid)</td>
</tr>
<tr>
<td>Activity * (Activity)</td>
</tr>
<tr>
<td>Posture * (Posture)</td>
</tr>
<tr>
<td>Foodcode * (Foodcode)</td>
</tr>
<tr>
<td>Ptime (Ptime)</td>
</tr>
<tr>
<td>(Ptimespm)</td>
</tr>
</tbody>
</table>

Figure 61: Entity Relationship Diagram of the POINTS View
3.23 The POTENTIAL_DADS View

```sql
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 supergroup(mommembers.grp, mommembers.date) = supergroup(dadmembers.grp, dadmembers.date))
        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 supergroup(mommembers.grp, mommembers.date) = supergroup(dadmembers.grp, dadmembers.date))
         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))
LEFT OUTER JOIN rankdates
ON (rankdates.sname = pdads.sname)
LEFT OUTER JOIN maturedates
ON (maturedates.sname = pdads.sname)
WHERE pdads.sex = 'M'
-- Speed things up by eliminating potential dads
-- who could not possibly interpolate into the mom's group
-- during the fertile period.
-- Potential dad must be at least 2192 days old
-- (approximately 6 years) on the zdate.
AND maternities.zdate - pdads.birth >= 2192;
```

Figure 62: Query Defining the POTENTIAL_DADS View
Figure 63: Entity Relationship Diagram of the foundation of the POTENTIAL_DADS View

* PDADS is an alias for BIOGRAPH, representing those BIOGRAPH rows that satisfy the conditions required to be considered a potential dad of a given kid. It does not appear anywhere as an independent entity. Additional conditions, as shown on other diagrams, must also be true for a BIOGRAPH row to be a PDADS row.
Figure 64: 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 65: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View having easily computed columns

* PDADS is an alias for BIOGRAPH, representing those BIOGRAPH rows that satisfy the conditions required to be considered a potential dad of a given kid. It does not appear anywhere as an independent entity. Additional conditions, as shown on other diagrams, must also be true for a BIOGRAPH row to be a PDADS row.
Figure 66: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View involving social interactions.
### 3.24 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,
       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 67: Query Defining the PROPORTIONAL_RANKS View

![Entity Relationship Diagram of the PROPORTIONAL_RANKS View](image)

* NUM_INDIVS is a subquery from RANKS, in which the number of RANKS rows (grouped by Rnkdate, Grp, and Rnktype) is counted. It does not appear anywhere as an independent entity.

Figure 68: Entity Relationship Diagram of the PROPORTIONAL_RANKS View
3.25  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 69: Query Defining the QUADS View

![Entity Relationship Diagram of the QUADS View](image)

3.26  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
FROM sexskins, cycles
WHERE cycles.cid = sexskins.cid
ORDER BY cycles.sname, sexskins.date;
```

Figure 71: Query Defining the SEXSKINS_CYCLES View
Figure 72: Entity Relationship Diagram of the SEXSKINS_CYCLES View
3.27 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.xyloc IS NULL
                THEN 'quad'
            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,
        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,
        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
        ON (quad_data.quad = swerb_data.quad)
    JOIN swerb_bes
        ON (swerb_bes.beid = swerb_data.beid)
    JOIN swerb_departs_data
        ON (swerb_departs_data.did = swerb_bes.did)
    LEFT OUTER JOIN swerb_departs_gps
        ON (swerb_departs_gps.did = swerb_bes.did)
    LEFT OUTER JOIN swerb_loc_data
        ON (swerb_loc_data.swid = swerb_data.swid)
    LEFT OUTER JOIN adcodes
        ON (adcodes.adcode = swerb_loc_data.adcode)
Figure 74: Entity Relationship Diagram of the SWERB View
3.28 The SWERB_DEPARTS view

```sql
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
    , 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 75: Query Defining the SWERB_DEPARTS View

![Figure 76: Entity Relationship Diagram of the SWERB_DEPARTS View](image)

3.29 The SWERB_LOC_GPS_XY view

```sql
SELECT swerb_loc_gps.swid AS swid
    , ST_X(swerb_loc_gps.xyloc) AS x
    , ST_Y(swerb_loc_gps.xyloc) AS y
    , 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 77: Query Defining the SWERB_LOC_GPS_XY View
### Figure 78: Entity Relationship Diagram of the SWERB_LOC_GPS_XY View

<table>
<thead>
<tr>
<th>SWERB_LOC_GPS</th>
<th>X coordinate of SWERB_LOC_GPS.XYLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWId (SWId)</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Altitude (Altitude)</td>
<td></td>
</tr>
<tr>
<td>PDOP (PDOP)</td>
<td></td>
</tr>
<tr>
<td>Accuracy (Accuracy)</td>
<td></td>
</tr>
<tr>
<td>GPS_Datetime (GPS_Datetime)</td>
<td></td>
</tr>
<tr>
<td>Garmincode (Garmincode)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.30 The SWERB_LOCS view

```sql
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 79: Query Defining the SWERB_LOCS View

<table>
<thead>
<tr>
<th>SWERB_LOC_DATA</th>
<th>ADCODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWId (SWId)</td>
<td>ADcode (ADcode)</td>
</tr>
<tr>
<td>Loc (Loc)</td>
<td>ADN (ADN)</td>
</tr>
<tr>
<td>ADcode *(Adcode)</td>
<td></td>
</tr>
<tr>
<td>Loc_Status *(Loc_Status)</td>
<td></td>
</tr>
<tr>
<td>ADtime (ADtime)</td>
<td></td>
</tr>
</tbody>
</table>

### Figure 80: Entity Relationship Diagram of the SWERB_LOCS View
### 3.31 The SWERB UPLOAD view

```sql
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::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 81: 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 82: Entity Relationship Diagram of the SWERB UPLOAD View

### 3.32 The ULNA_STATS View

```sql
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 83: Query Defining the ULNA_STATS View
Figure 84: Entity Relationship Diagram of the ULNA_STATS View
3.33 The TESTES_ARC_STATS View

```sql
SELECT testesdartids.dartid AS dartid
, testeslength.testlengthsamps AS testlengthsamps
, testeslength.testlength_mean AS testlength_mean
, testeslength.testlength_stddev AS testlength_stddev
, testeswidth.testwidthsamps AS testwidthsamps
, testeswidth.testwidth_mean AS testwidth_mean
, testeswidth.testwidth_stddev AS testwidth_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 testlengthsamps
, avg(testes_arc.testlength) AS testlength_mean
, stddev(testes_arc.testlength) AS testlength_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 testwidthsamps
, avg(testes_arc.testwidth) AS testwidth_mean
, stddev(testes_arc.testwidth) AS testwidth_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 testlengthsamps
, avg(testes_arc.testlength) AS testlength_mean
, stddev(testes_arc.testlength) AS testlength_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 testwidthsamps
, avg(testes_arc.testwidth) AS testwidth_mean
, stddev(testes_arc.testwidth) AS testwidth_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 85: Query Defining the TESTES_ARC_STATS View
Figure 86: Entity Relationship Diagram of the TESTES_ARC_STATS View
### 3.34 The TESTES_DIAM_STATS View

```sql
SELECT testesdartids.dartid AS dartid
    , testeslength.testlengthsamps AS testlengthsamps
    , testeslength.testlength_mean AS testlength_mean
    , testeslength.testlength_stddev AS testlength_stddev
    , testeswidth.testwidthsamps AS testwidthsamps
    , testeswidth.testwidth_mean AS testwidth_mean
    , testeswidth.testwidth_stddev AS testwidth_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 87: Query Defining the TESTES_DIAM_STATS View
Figure 88: Entity Relationship Diagram of the TESTES_DIAM_STATS View
### 3.35 The WP DETAILS_AFFECTEDPARTS View

```sql
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,
wounds.maxdimension AS maxdimension, wp_details.impairslocomotion AS impairslocomotion,
wounds.infectionsigns AS infectionsigns, wp_details.notes AS detailnotes,
wounds.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 89: Query Defining the WP DETAILS_AFFECTEDPARTS View

Figure 90: Entity Relationship Diagram of the WP DETAILS_AFFECTEDPARTS View
### 3.36 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 91: Query Defining the WP_HEALS View
Figure 92: Entity Relationship Diagram of the WP.HEALS View, Overall
Figure 93: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology report
Figure 94: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology cluster
Figure 95: Entity Relationship Diagram of the WP.HEALS View for rows with an update to an affected body part
3.37 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 96: Query Defining the WP_REPORTS_OBSERVERS View

Figure 97: 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 98: Query Defining the BIRTH_GRP View

![Entity Relationship Diagram of the BIRTH_GRP View](image)

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 100: Query Defining the ENTRYDATE_GRP View

![Entity Relationship Diagram of the ENTRYDATE_GRP View](image)
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 102: Query Defining the STATDATE_GRP View

<table>
<thead>
<tr>
<th>BIOGRAPH</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Statdate (Statdate)</td>
<td>Date (Statdate)</td>
</tr>
<tr>
<td>Remainder of columns in BIOGRAPH....</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 103: 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 104: Query Defining the CONSORTDATES_GRP View

<table>
<thead>
<tr>
<th>CONSORTDATES</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Consorted (Consorted)</td>
<td>Date (Consorted)</td>
</tr>
<tr>
<td>Remainder of columns in CONSORTDATES....</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 105: 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 106: Query Defining the CYCGAPDAYS_GRP View

<table>
<thead>
<tr>
<th>CYCGAPDAYS</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cenid (Cenid)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Date (Date)</td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remainder of columns in CYCGAPDAYS.....</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 107: 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 108: Query Defining the CYCGAPS_GRP View

<table>
<thead>
<tr>
<th>CYCGAPS</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gapid (Gapid)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Date (Date)</td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remainder of columns in CYCGAPS.....</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 109: Entity Relationship Diagram of the CYCGAPS_GRP View
4.7 The CYCSTATS_GRP View

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

Figure 110: Query Defining the CYCSTATS_GRP View

<table>
<thead>
<tr>
<th>CYCSTATS</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSid (CSid)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Date (Date)</td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
<td>Date in question</td>
</tr>
<tr>
<td>Remainder of columns in CYCSTATS....</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 111: Entity Relationship Diagram of the CYCSTATS_GRP View

4.8 The DARTINGS_GRP View

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

Figure 112: Query Defining the DARTINGS_GRP View

<table>
<thead>
<tr>
<th>DARTINGS</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dartid (Dartid)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Date (Dartdaytime)</td>
</tr>
<tr>
<td>Dartdaytime (Dartdaytime)</td>
<td></td>
<td>Date in question</td>
</tr>
<tr>
<td>Remainder of columns in DARTINGS....</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 113: 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 114: Query Defining the DISPERSEDATES_GRP View

<table>
<thead>
<tr>
<th>DISPERSEDATES</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Dispersed (Dispersed)</td>
<td></td>
<td>Date (Dispersed)</td>
</tr>
<tr>
<td>Remainder of columns in DISPERSEDATES...</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 115: 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 116: Query Defining the MATUREDATES_GRP View

<table>
<thead>
<tr>
<th>MATUREDATES</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Matured (Matured)</td>
<td></td>
<td>Date (Matured)</td>
</tr>
<tr>
<td>Remainder of columns in MATUREDATES...</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 117: Entity Relationship Diagram of the MATUREDATES_GRP View
4.11 The MDINTERVALS_GRP View

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

Figure 118: Query Defining the MDINTERVALS_GRP View

<table>
<thead>
<tr>
<th>MDINTERVALS</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDId (MDId)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Date (Date)</td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
<tr>
<td>Remainder of columns in MDINTERVALS....</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 119: Entity Relationship Diagram of the MDINTERVALS_GRP View

4.12 The MMINTERVALS_GRP View

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

Figure 120: Query Defining the MMINTERVALS_GRP View

<table>
<thead>
<tr>
<th>MMINTERVALS</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMId (MMId)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Date (Date)</td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
<tr>
<td>Remainder of columns in MMINTERVALS....</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 121: Entity Relationship Diagram of the MMINTERVALS_GRP View
4.13 The PCSKINS_GRP View

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

Figure 122: Query Defining the PCSKINS_GRP View

<table>
<thead>
<tr>
<th>PCSKINS</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCSId (PSid)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Date (Date)</td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remainder of columns in PCSKINS....</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 123: Entity Relationship Diagram of the PCSKINS_GRP View

4.14 The RANKDATES_GRP View

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

Figure 124: Query Defining the RANKDATES_GRP View

<table>
<thead>
<tr>
<th>RANKDATES</th>
<th>Individual in question</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Ranked (Ranked)</td>
<td></td>
<td>Date (Ranked)</td>
</tr>
<tr>
<td>Remainder of columns in RANKDATES....</td>
<td></td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 125: Entity Relationship Diagram of the RANKDATES_GRP View
4.15 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 126: Query Defining the REPSTATS_GRP View

![Entity Relationship Diagram of the REPSTATS_GRP View](image)

Figure 127: Entity Relationship Diagram of the REPSTATS_GRP View