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>September 11, 2020</td>
<td></td>
</tr>
<tr>
<td>ER Diagram layout and conversion to Dia</td>
<td>Leah Gerber</td>
<td>September 11, 2020</td>
<td></td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Anne Hubbard</td>
<td>September 11, 2020</td>
<td></td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Karl Pinc</td>
<td>September 11, 2020</td>
<td></td>
</tr>
</tbody>
</table>
## Contents

1 Babase Summarized  

2 The Babase ER Diagrams  

3 The Babase Views  
   3.1 The ACTOR_ACTEES View  
   3.2 The ANESTH_STATS View  
   3.3 The BODYTEMP_STATS View  
   3.4 The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views  
   3.5 The CHEST_STATS View  
   3.6 The CROWNJRUMP_STATS View  
   3.7 The CYCLES_SEXSKINS and CYCLES_SEXSKINS_SORTED Views  
   3.8 The CYCPOINTS_CYCLES and CYCPOINTS_CYCLES_SORTED Views  
   3.9 The DSAMPLES View  
   3.10 The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views  
   3.11 The DENT_CODES View  
   3.12 The DENT_SITES View  
   3.13 The INTERACT and INTERACT_SORTED Views  
   3.14 The LOCATIONS_FREE View  
   3.15 The MATERNITIES View  
   3.16 The MIN_MAXS View  
   3.17 The MIN_MAXS_SORTED View  
   3.18 The MPI_EVENTS View  
   3.19 The MTD_CYCLES View  
   3.20 The NUCACID_CONCS View  
   3.21 The NUCACIDS View  
   3.22 The NUCACIDS_W_CONC View  
   3.23 The PARENTS View  
   3.24 The PCSKINS_SORTED View  
   3.25 The PCV_STATS View  
   3.26 The POINTS and POINTS_SORTED Views  
   3.27 The POTENTIAL_DADS View  
   3.28 The PROPORTIONAL_RANKS View  
   3.29 The QUADS View  
   3.30 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_SORTED Views  
   3.31 The SWERB View  
   3.32 The SWERB_DEPARTS view
3.33 The SWERB_LOC_GPS_XY view ........................................ 76
3.34 The SWERB_LOCS view ........................................ 77
3.35 The SWERB_UPLOAD view ........................................ 78
3.36 The TISSUES View ........................................ 79
3.37 The ULNA_STATS View ........................................ 80
3.38 The TESTES_ARCH_STATS View ........................................ 83
3.39 The TESTES_DIAM_STATS View ........................................ 86
3.40 The WP_DETAILS_AFFECTEDPARTS View ........................................ 88
3.41 The WP_HEALS View ........................................ 89
3.42 The WP_REPORTS_OBSERVERS View ........................................ 94

4 Views Which Add Gid To Tables ........................................ 94
4.1 The BIRTH_GRP View ........................................ 95
4.2 The ENTRYDATE_GRP View ........................................ 95
4.3 The STATDATE_GRP View ........................................ 96
4.4 The CONSORTDATES_GRP View ........................................ 96
4.5 The CYC GAPDAYS_GRP View ........................................ 97
4.6 The CYC GAPS_GRP View ........................................ 97
4.7 The CYCSTATS_GRP View ........................................ 98
4.8 The DARTINGS_GRP View ........................................ 98
4.9 The DISPERSEDATES_GRP View ........................................ 99
4.10 The MATUREDATES_GRP View ........................................ 99
4.11 The MDINTERVALS_GRP View ......................................... 100
4.12 The MMINTERVALS_GRP View ......................................... 100
4.13 The PCSKINS_GRP View ......................................... 101
4.14 The RANKDATES_GRP View ......................................... 101
4.15 The REPSTATS_GRP View ......................................... 102
List of Figures

1. Key to the Babase Entity Relationship Diagrams
2. Babase Group Membership Entity Relationship Diagram
3. Babase Life Events Entity Relationship Diagram
4. Babase Sexual Cycle Entity Relationship Diagram
5. Babase Sexual Cycle Day-To-Day Tables Entity Relationship Diagram
6. Babase Social Interactions Entity Relationship Diagram
7. Babase Multiparty Interactions Entity Relationship Diagram
8. Babase Darting Logistics and Morphology Entity and Relationship Diagram
9. Babase Darting Physiology Entity and Relationship Diagram
10. Babase Darting Samples Entity and Relationship Diagram
11. Babase Darting Teeth and Ticks Entity and Relationship Diagram
12. Babase Inventory Entity Relationship Diagram
13. Babase Physical Traits Genetic Hybrid Score Data Entity Relationship Diagram
14. Babase Physical Traits Wounds and Pathologies Data Entity Relationship Diagram
15. Babase SWERB Core Tables Entity Relationship Diagram
16. Babase SWERB Grove/Waterhole Location Tables Entity Relationship Diagram
17. Babase Manual Weather Data Entity Relationship Diagram
18. Babase WeatherHawk Data Entity Relationship Diagram
19. Warning Sub-System Entity Relationship Diagram
20. Query Defining the ACTOR_ACTEES View
21. Entity Relationship Diagram of the ACTOR_ACTEES View
22. Query Defining the ANESTH_STATS View
23. Entity Relationship Diagram of the ANESTH_STATS View
24. Query Defining the BODYTEMP_STATS View
25. Entity Relationship Diagram of the BODYTEMP_STATS View
26. Query Defining the CENSUS_DEMOG View
27. Entity Relationship Diagram of the CENSUS_DEMOG View
28. Query Defining the CHEST_STATS View
29. Entity Relationship Diagram of the CHEST_STATS View
30. Query Defining the CROWNRUMP_STATS View
31. Entity Relationship Diagram of the CROWNRUMP_STATS View
32. Query Defining the CYCLES_SEXSKINS View
33. Entity Relationship Diagram of the CYCLES_SEXSKINS View
34. Query Defining the CYCPOINTS_CYCLES View
35. Entity Relationship Diagram of the CYCPOINTS_CYCLES View
36. Query Defining the DSAMPLES View
37. Query Defining the DEMOG_CENSUS View
<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>Entity Relationship Diagram of the DEMOG_CENSUS View</td>
</tr>
<tr>
<td>39</td>
<td>Query Defining the DENT_CODES View</td>
</tr>
<tr>
<td>40</td>
<td>Entity Relationship Diagram of the DENT_CODES View</td>
</tr>
<tr>
<td>41</td>
<td>Query Defining the DENT_SITES View</td>
</tr>
<tr>
<td>42</td>
<td>Entity Relationship Diagram of the DENT_SITES View</td>
</tr>
<tr>
<td>43</td>
<td>Query Defining the INTERACT View</td>
</tr>
<tr>
<td>44</td>
<td>Entity Relationship Diagram of the INTERACT View</td>
</tr>
<tr>
<td>45</td>
<td>Query Defining the LOCATIONS_FREE View</td>
</tr>
<tr>
<td>46</td>
<td>Entity Relationship Diagram of the LOCATIONS_FREE View</td>
</tr>
<tr>
<td>47</td>
<td>Query Defining the MATERNITIES View</td>
</tr>
<tr>
<td>48</td>
<td>Entity Relationship Diagram of the MATERNITIES View</td>
</tr>
<tr>
<td>49</td>
<td>Query Defining the MIN_MAXS View</td>
</tr>
<tr>
<td>50</td>
<td>Entity Relationship Diagram of the MIN_MAXS View</td>
</tr>
<tr>
<td>51</td>
<td>Query Defining the MIN_MAXS_SORTED View</td>
</tr>
<tr>
<td>52</td>
<td>Entity Relationship Diagram of the MIN_MAXS_SORTED View</td>
</tr>
<tr>
<td>53</td>
<td>Query Defining the MPI_EVENTS View</td>
</tr>
<tr>
<td>54</td>
<td>Entity Relationship Diagram of the MPI_EVENTS View</td>
</tr>
<tr>
<td>55</td>
<td>Query Defining the MTD_CYCLES View</td>
</tr>
<tr>
<td>56</td>
<td>Entity Relationship Diagram of the MTD_CYCLES View</td>
</tr>
<tr>
<td>57</td>
<td>Query Defining the NUCACID_CONCS View</td>
</tr>
<tr>
<td>58</td>
<td>Entity Relationship Diagram of the NUCACID_CONCS View</td>
</tr>
<tr>
<td>59</td>
<td>Query Defining the NUCACIDS View</td>
</tr>
<tr>
<td>60</td>
<td>Entity Relationship Diagram of the NUCACIDS View</td>
</tr>
<tr>
<td>61</td>
<td>Query Defining the NUCACIDS_W_CONC View</td>
</tr>
<tr>
<td>62</td>
<td>Entity Relationship Diagram of the NUCACIDS_W_CONC View</td>
</tr>
<tr>
<td>63</td>
<td>Query Defining the PARENTS View</td>
</tr>
<tr>
<td>64</td>
<td>Entity Relationship Diagram of the PARENTS View</td>
</tr>
<tr>
<td>65</td>
<td>Query Defining the PCSKINS_SORTED View</td>
</tr>
<tr>
<td>66</td>
<td>Entity Relationship Diagram of the PCSKINS_SORTED View</td>
</tr>
<tr>
<td>67</td>
<td>Query Defining the PCV_STATS View</td>
</tr>
<tr>
<td>68</td>
<td>Entity Relationship Diagram of the PCV_STATS View</td>
</tr>
<tr>
<td>69</td>
<td>Query Defining the POINTS View</td>
</tr>
<tr>
<td>70</td>
<td>Entity Relationship Diagram of the POINTS View</td>
</tr>
<tr>
<td>71</td>
<td>Query Defining the POTENTIAL_DADS View</td>
</tr>
<tr>
<td>72</td>
<td>Entity Relationship Diagram of the foundation of the POTENTIAL_DADS View</td>
</tr>
<tr>
<td>73</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>74</td>
<td>Entity Relationship Diagram of that portion of the POTENTIAL_DADS View having easily computed columns</td>
</tr>
<tr>
<td>75</td>
<td>Entity Relationship Diagram of that portion of the POTENTIAL_DADS View involving social interactions</td>
</tr>
</tbody>
</table>
76 Query Defining the PROPORATIONAL_RANKS View
77 Entity Relationship Diagram of the PROPORATIONAL_RANKS View
78 Query Defining the QUADS View
79 Entity Relationship Diagram of the QUADS View
80 Query Defining the SEXSKINS_CYCLES View
81 Entity Relationship Diagram of the SEXSKINS_CYCLES View
82 Query Defining the SWERB View
83 Entity Relationship Diagram of the SWERB View
84 Query Defining the SWERB_DEPARTS View
85 Entity Relationship Diagram of the SWERB_DEPARTS View
86 Query Defining the SWERB_LOC_GPS_XY View
87 Entity Relationship Diagram of the SWERB_LOC_GPS_XY View
88 Query Defining the SWERB_LOCS View
89 Entity Relationship Diagram of the SWERB_LOCS View
90 Query Defining the SWERB_UPLOAD View
91 Entity Relationship Diagram of the SWERB_UPLOAD View
92 Query Defining the TISSUES View
93 Entity Relationship Diagram of the TISSUES View
94 Query Defining the ULNA_STATS View
95 Entity Relationship Diagram of the ULNA_STATS View
96 Query Defining the TESTES_ARC_STATS View
97 Entity Relationship Diagram of the TESTES_ARC_STATS View
98 Query Defining the TESTES_DIAM_STATS View
99 Entity Relationship Diagram of the TESTES_DIAM_STATS View
100 Query Defining the WP_DETAILS_AFFECTEDPARTS View
101 Entity Relationship Diagram of the WP_DETAILS_AFFECTEDPARTS View
102 Query Defining the WP_HEALS View
103 Entity Relationship Diagram of the WP_HEALS View, Overall
104 Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology report
105 Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology cluster
106 Entity Relationship Diagram of the WP_HEALS View for rows with an update to an affected body part
107 Query Defining the WP_REPORTS_OBSERVERS View
108 Entity Relationship Diagram of the WP_REPORTS_OBSERVERS View
109 Query Defining the BIRTH_GRP View
110 Entity Relationship Diagram of the BIRTH_GRP View
111 Query Defining the ENTRYDATE_GRP View
112 Entity Relationship Diagram of the ENTRYDATE_GRP View
113 Query Defining the STATDATE_GRP View
114 Entity Relationship Diagram of the STATDATE_GRP View
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

1 At this time of this writing only males have data entered into RANKDATES in Babase.
<table>
<thead>
<tr>
<th>Group Membership and Life Events</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table</strong></td>
<td>One row for each</td>
</tr>
<tr>
<td><strong>ALTERNATE_SNAMES in Babase:</strong></td>
<td>rescinded sname</td>
</tr>
<tr>
<td><strong>BIOGRAPH in Babase:</strong></td>
<td>animal, including fetuses</td>
</tr>
<tr>
<td><strong>CENSUS in Babase:</strong></td>
<td>day each individual is (or is not) observed in a group</td>
</tr>
<tr>
<td><strong>CONSORTDATES in Babase:</strong></td>
<td>male who has a known first consortship</td>
</tr>
<tr>
<td><strong>DEMOG in Babase:</strong></td>
<td>mention of an individual’s presence in a group within a field textual note</td>
</tr>
<tr>
<td><strong>DISPERSEDATES in Babase:</strong></td>
<td>male who has left his maternal study group</td>
</tr>
<tr>
<td><strong>GROUPS in Babase:</strong></td>
<td>group (including solitary males)</td>
</tr>
<tr>
<td><strong>MATUREDATES in Babase:</strong></td>
<td>individual who is sexually mature</td>
</tr>
<tr>
<td><strong>RANKDATES in Babase:</strong></td>
<td>individual (^1) who has attained adult rank</td>
</tr>
</tbody>
</table>

Analyzed: Group Membership and Life Events

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DAD_DATA in Babase:</strong></td>
<td>offspring having a paternity analysis</td>
</tr>
<tr>
<td><strong>MEMBERS in Babase:</strong></td>
<td>day each individual is alive</td>
</tr>
<tr>
<td><strong>RANKS in Babase:</strong></td>
<td>month each individual is ranked in each group</td>
</tr>
</tbody>
</table>

Analyzed: Physical Traits

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HYBRIDGENE_ANALYSES in Babase:</strong></td>
<td>analysis of genetic hybrid scores</td>
</tr>
<tr>
<td><strong>HYBRIDGENE_SCORES in Babase:</strong></td>
<td>genetic hybrid score for an individual from an analysis</td>
</tr>
<tr>
<td><strong>WP_AFFECTEDPARTS in Babase:</strong></td>
<td>body part affected by a specific wound/pathology</td>
</tr>
<tr>
<td><strong>WP_DETAILS in Babase:</strong></td>
<td>wound or pathology cluster indicated on a report</td>
</tr>
<tr>
<td><strong>WP_HEALUPDATES in Babase:</strong></td>
<td>update on progress of wound/pathology healing</td>
</tr>
<tr>
<td><strong>WP_REPORTS in Babase:</strong></td>
<td>wound/pathology report</td>
</tr>
</tbody>
</table>

Sexual Cycles

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CYCPCGAPS in Babase:</strong></td>
<td>female for each initiation or cessation of a continuous period of observation</td>
</tr>
<tr>
<td><strong>CYCLES in Babase:</strong></td>
<td>female’s cycle (complete or not)</td>
</tr>
<tr>
<td><strong>CYCPOINTS in Babase:</strong></td>
<td>Mdate (menses), Tdate (turgescence onset), or Ddate (deturgescence onset) date of each female</td>
</tr>
<tr>
<td><strong>PCS KSKINS in Babase:</strong></td>
<td>PCS color of each female</td>
</tr>
<tr>
<td><strong>PREGS in Babase:</strong></td>
<td>time a female becomes pregnant</td>
</tr>
<tr>
<td><strong>SEXSKINS in Babase:</strong></td>
<td>sexskin measurement of each female</td>
</tr>
</tbody>
</table>

The Sexual Cycle Day-By-Day Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CYCGAPDAYS in Babase:</strong></td>
<td>female for each day within a period during which there is not continuous observation</td>
</tr>
<tr>
<td><strong>CYCGAPSTATS in Babase:</strong></td>
<td>day each female is cycling -- by M, T and Ddates</td>
</tr>
<tr>
<td><strong>MINDINTERVALS in Babase:</strong></td>
<td>day each female is cycling and is between M and Ddates</td>
</tr>
<tr>
<td><strong>MMINTERVALS in Babase:</strong></td>
<td>day each female is cycling -- by Mdates</td>
</tr>
</tbody>
</table>
The Babase Pocket Reference Guide

Table 2: The Warning Sub-System Tables

<table>
<thead>
<tr>
<th>Table 2: The Warning Sub-System Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INTEGRITY QUERIES</strong> in Babase:</td>
</tr>
<tr>
<td>query used to discover data integrity problems</td>
</tr>
<tr>
<td><strong>INTEGRITY WARNINGS</strong> in Babase:</td>
</tr>
<tr>
<td>data integrity problem discovered by the warning sub-system</td>
</tr>
</tbody>
</table>

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>BODYPARTS in Babase:</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>LAB_PERSONNEL in Babase:</td>
<td>Initials</td>
<td>HYBRIDGENE_ANALYSES in Babase: Analyzed_By in Babase: NUCACID_CREATORS in Babase: Creator in Babase: WBC_COUNTS in Babase: Counted_By in Babase:</td>
<td>person who generates data, usually in a lab setting</td>
</tr>
<tr>
<td>OBSERVER_ROLES in Babase:</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>Role way in which a person can be involved in the data collection process</td>
</tr>
<tr>
<td>UNKSNAMES in Babase:</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>CONFIDENCES in Babase:</td>
<td>Confidence</td>
<td>BIOGRAPH in Babase: DcauseNatureConfidence in Babase: BIOGRAPH in Babase: DcauseAgentConfidence in Babase: DISPERSEDATES in Babase:</td>
<td>degree of certitude in nature of death, agent of death, disperse date assignment, or maternal group assignment</td>
</tr>
</tbody>
</table>
Table 4: The Warning Sub-System 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>IQTYPES in Babase:</td>
<td>IQType</td>
<td>INTEGRITY_QUERIES in Babase::Type in Babase:</td>
<td>kind of problem with data integrity</td>
</tr>
<tr>
<td>WARNING_REMARKS in Babase:</td>
<td>WRID</td>
<td>INTEGRITY_WARNINGS in Babase::Category in Babase:</td>
<td>remark which might apply to more than one instance of questionable database integrity</td>
</tr>
</tbody>
</table>

Figure 3: Babase Life Events Entity Relationship Diagram
<table>
<thead>
<tr>
<th>Group Membership and Life Events</th>
<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>
<td></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>
<td></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>
<td></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::Sname in Babase:, by CYCPOINTS in Babase::Date in Babase:.</td>
<td>CYCLES in Babase:, CYCPOINTS in Babase:</td>
<td></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>
<td></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>
<td></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>
<td></td>
</tr>
<tr>
<td>PARENTS in Babase:</td>
<td>BIOGRAPH in Babase: row</td>
<td>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_consent in Babase:. Easy access to parental information.</td>
<td>BIOGRAPH in Babase:, MATERNITIES in Babase:, DAD_DATA in Babase:, MEMBERS in Babase:</td>
<td></td>
</tr>
<tr>
<td>POTENTIAL_DADS in Babase:</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>Research into paternity, especially the selection of potential fathers for further genetic testing.</td>
<td>MATERNITIES in Babase:, MEMBERS in Babase: (multiple times), ACTOR_ACTEES in Babase: (multiple times), BIOGRAPH in Babase:, RANKDATES in Babase:, MATUREDATES in Babase:</td>
<td></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>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical Traits</th>
<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>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>View</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<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>
<td></td>
</tr>
<tr>
<td>BIOGRAPH in Babase</td>
<td>ENTRYDATE_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIOGRAPH in Babase</td>
<td>STATDATE_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSORTDATES in Babase</td>
<td>CONSORTDATES_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCGAPDAYS in Babase</td>
<td>CYCGAPDAYS_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCGAPS in Babase</td>
<td>CYCGAPS_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCSTATS in Babase</td>
<td>CYCSTATS_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DARTINGS in Babase</td>
<td>DARTINGS_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISPERSEDATES in Babase</td>
<td>DISPERSEDATES_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATUREDATES in Babase</td>
<td>MATUREDATES_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDINTERVALS in Babase</td>
<td>MDINTERVALS_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMINTERVALS in Babase</td>
<td>MMINTERVALS_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCSKINS in Babase</td>
<td>PCSKINS_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RANKDATES in Babase</td>
<td>RANKDATES_GRP in Babase:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPSTATS in Babase</td>
<td>REPSTATS_GRP in Babase:</td>
<td></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 Inventory Entity Relationship Diagram
Figure 13: Babase Physical Traits Genetic Hybrid Score Data Entity Relationship Diagram
Figure 14: Babase Physical Traits Wounds and Pathologies Data Entity Relationship Diagram
Figure 15: Babase SWERB Core Tables Entity Relationship Diagram
Figure 16: Babase SWERB Grove/Waterhole Location Tables Entity Relationship Diagram
Figure 17: Babase Manual Weather Data Entity Relationship Diagram
Figure 18: 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

```sql
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 20: Query Defining the ACTOR_ACTEES View
3.2 The ANESTH_STATS View

```sql
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 22: Query Defining the ANESTH_STATS View
3.3 The BODYTEMP_STATS View

```sql
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 24: 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 26: Query Defining 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 28: Query Defining the CHEST_STATS View

Figure 29: 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 30: Query Defining the **CROWNRUMP_STATS** View

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

Figure 31: Entity Relationship 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 32: 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 34: Query Defining the CYCPOINTS_CYCLES View

Figure 35: 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_lmaxillamolds.num AS lmaxillamold, teeth_lmol1mol2s.num AS lml2siliconemold, skin_punchs.num AS skinpunch, tc_skins.num AS tcskin, vag_swabs.num AS vaginals, 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

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 37: Query Defining the DEMOG_CENSUS View

```
<table>
<thead>
<tr>
<th>CENUSE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname</td>
<td>(Sname)</td>
</tr>
<tr>
<td>Date</td>
<td>(Date)</td>
</tr>
<tr>
<td>Grp</td>
<td>(Grp)</td>
</tr>
<tr>
<td>Status</td>
<td>(Status)</td>
</tr>
<tr>
<td>Cen</td>
<td>(Cen)</td>
</tr>
<tr>
<td>Cenid</td>
<td>(Cenid)</td>
</tr>
</tbody>
</table>
```

Figure 38: Entity Relationship Diagram of the DEMOG_CENSUS View
3.11 The DENT_CODES View

SELECT teeth.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,
Figure 40: Entity Relationship Diagram of the DENT_CODES View

- These columns repeat: there is a set of these columns for every TOOTHCODES row. The "TC" shown here in each column name is replaced in the actual column name with a TOOTHCODE.Tooth value.
3.12 The DENT_SITES View

```sql
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,
       s20.s20deciduous AS s20deciduous,
       s21.s21tstate AS s21tstate,
       s21.s21tcondition AS s21tcondition,
       s21.s21deciduous AS s21deciduous,
       s22.s22tstate AS s22tstate,
       s22.s22tcondition AS s22tcondition,
       s22.s22deciduous AS s22deciduous,
       s23.s23tstate AS s23tstate,
       s23.s23tcondition AS s23tcondition,
       s23.s23deciduous AS s23deciduous,
       s24.s24tstate AS s24tstate,
       s24.s24tcondition AS s24tcondition,
       s24.s24deciduous AS s24deciduous,
       s25.s25tstate AS s25tstate,
       s25.s25tcondition AS s25tcondition,
       s25.s25deciduous AS s25deciduous,
       s26.s26tstate AS s26tstate,
       s26.s26tcondition AS s26tcondition,
       s26.s26deciduous AS s26deciduous,
       s27.s27tstate AS s27tstate,
       s27.s27tcondition AS s27tcondition,
       s27.s27deciduous AS s27deciduous,
       s28.s28tstate AS s28tstate,
       s28.s28tcondition AS s28tcondition,
       s28.s28deciduous AS s28deciduous,
       s29.s29tstate AS s29tstate,
       s29.s29tcondition AS s29tcondition,
       s29.s29deciduous AS s29deciduous,
       s30.s30tstate AS s30tstate,
       s30.s30tcondition AS s30tcondition,
       s30.s30deciduous AS s30deciduous,
       s31.s31tstate AS s31tstate,
       s31.s31tcondition AS s31tcondition,
       s31.s31deciduous AS s31deciduous,
       s32.s32tstate AS s32tstate,
       s32.s32tcondition AS s32tcondition,
       s32.s32deciduous AS s32deciduous;
```

Figure 41: Query Defining the DENT_SITES View
Figure 42: Entity Relationship Diagram of the DENT_SITES View
3.13 The INTERACT and INTERACT_SORTED Views

```sql
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 43: Query Defining the INTERACT View

<table>
<thead>
<tr>
<th>INTERACT_DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>iid (lid)</td>
</tr>
<tr>
<td>Sid (Sid)</td>
</tr>
<tr>
<td>Act (Act)</td>
</tr>
<tr>
<td>Date (Date)</td>
</tr>
<tr>
<td>(Jdate)</td>
</tr>
<tr>
<td>Start (Start)</td>
</tr>
<tr>
<td>(Startspm)</td>
</tr>
<tr>
<td>Stop (Stop)</td>
</tr>
<tr>
<td>(Stopspm)</td>
</tr>
<tr>
<td>Observer *</td>
</tr>
<tr>
<td>Handwritten</td>
</tr>
</tbody>
</table>

Figure 44: Entity Relationship Diagram of the INTERACT View
3.14 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 45: Query Defining the LOCATIONS_FREE View

Figure 46: Entity Relationship Diagram of the LOCATIONS_FREE View
3.15 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.edate AS edate,
       cycpoints.1date 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 47: Query Defining the MATERNITIES View
Figure 48: 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.)
3.16 The MIN_MAXS View

```sql
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 49: Query Defining the MIN_MAXS View
Figure 50: Entity Relationship Diagram of the MIN_MAXS View
3.17 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 51: Query Defining the MIN_MAXS_SORTED View
Figure 52: Entity Relationship Diagram of the MIN_MAXS_SORTED View
3.18 The MPI_EVENTS View

```
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
                  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 53: Query Defining the MPI_EVENTS View
Figure 54: Entity Relationship Diagram of the MPI_EVENTS View
3.19 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 55: Query Defining the MTD_CYCLES View
Figure 56: Entity Relationship Diagram of the MTD_CYCLES View
3.20 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 57: Query Defining the NUCACID_CONCS View

Figure 58: Entity Relationship Diagram of the NUCACID_CONCS View
3.21 The NUCACIDS View

WITH concatCreators 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, tissue_data.uid AS uid, 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, nucacid_types.descr AS nucacid_descr, tissue_data.tissue_type AS tissue_type, nucacid_data.creation_date AS creation_date, concatCreators.created_by AS created_by, nucacid_data.creation_method AS creation_method, nucacid_sources.source_naid AS source_na, nucacid_sources.relationship AS source_na_relationship, 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.notes AS notes
FROM nucacid_data
JOIN nucacid_types
ON nucacid_types.nucacid_type = nucacid_data.nucacid_type
JOIN locations
ON locations.locid = nucacid_data.locid
JOIN tissue_data
ON tissue_data.tid = nucacid_data.tid
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 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;

Figure 59: Query Defining the NUCACIDS View
Figure 60: Entity Relationship Diagram of the NUCACIDS View

* CONCAT_CREATORS is a subquery from NUCACID_CREATORS, in which all Creator rows for an NAld are concatenated (but separated by a "/") into a single string. It does not appear anywhere as an independent entity.
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,
    tissue_data.uid AS uid,
    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,
    nucacid_types.descr AS nucacid_descr,
    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,
    nucacid_sources.source_naid AS source_na,
    nucacid_sources.relationship AS source_na_relationship,
    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.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 nucacid_types
ON nucacid_types.nucacid_type = nucacid_data.nucacid_type
JOIN locations
ON locations.locid = nucacid_data.locid
JOIN tissue_data
ON tissue_data.tid = nucacid_data.tid
JOIN unique_indivs
ON unique_indivs.uiid = tissue_data.uid
LEFT JOIN biograph
ON biograph.bioid::text = unique_indivs.individ
Figure 62: Entity Relationship Diagram of the NUCACIDS_W_CONC View
3.23 The PARENTS View

```sql
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 63: Query Defining the PARENTS View
3.24 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 65: Query Defining the PCSKINS_SORTED View
3.25 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 66: Entity Relationship Diagram of the PCSKINS_SORTED View

Figure 67: Query Defining the PCV_STATS View

Figure 68: Entity Relationship Diagram of the PCV_STATS View
3.26 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 69: 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 70: Entity Relationship Diagram of the POINTS View
### 3.27 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 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
                AND mommembers.date >= maternities.zdate - 5)
            AND dadmembers.statdate >= maternities.zdate - 5 - 14
            -- Potential dad must be at least 2192 days old
            -- (approximately 6 years) on the zdate.
            AND maternities.zdate - pdads.birth >= 2192;
```

Figure 71: Query Defining the POTENTIAL_DADS View
**Figure 72: 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 73: 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.

** This subquery is repeated twice in the view, once to test BIOGRAPH rows for inclusion as potential dads and again to compute Estrous_presence.

* 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 74: 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.

(PDADS) *  
(Pdad)  
(Estrous_presence)  
(Status)  
‘A’ (Adult) when ranked on or before the Zdate (Ranked <= Zdate), otherwise ‘S’ (Subadult) when matured on or before the Zdate (Matured <= Zdate), otherwise ‘O’ (Other)

(Pdad_age_days)  
The age, in days, of the potential dad as of the Zdate. Computed by subtracting Birth from the Zdate, (Zdate - Birth). So on the first day after the date of birth the individual is 1 day old, the second day 2 days old, etc.

(Pdad_age_years)  
The age, in years with a precision of 1 decimal place, of the potential dad as of the Zdate. Computed by subtracting Birth from the Zdate, dividing the result by 365.25 and truncating the result to 1 decimal place, (trunc((Zdate - Birth) / 365.25, 1)).
Figure 75: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View involving social interactions

** This subquery on ACTOR_ACTEES is repeated twice in the view, once to compute Estrous_me and once to compute Estrous_c.

* 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.
3.28 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 76: Query Defining the PROPORTIONAL_RANKS View

* 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 77: Entity Relationship Diagram of the PROPORTIONAL_RANKS View
### 3.29 The QUADS View

```sql
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 78: Query Defining the QUADS View

![QUADS Diagram](image)

Figure 79: Entity Relationship Diagram of the QUADS View

### 3.30 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_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 sexskins, cycles
WHERE cycles.cid = sexskins.cid
ORDER BY cycles.sname, sexskins.date;
```

Figure 80: Query Defining the SEXSKINS_CYCLES View
Figure 81: Entity Relationship Diagram of the SEXSKINS_CYCLES View
### 3.31 The SWERB view

```sql
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 82: Query Defining the SWERB View
Figure 83: Entity Relationship Diagram of the SWERB View
3.32 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 84: Query Defining the SWERB_DEPARTS View

<table>
<thead>
<tr>
<th>SWERB_DEPARTS_DATA</th>
<th>Departure</th>
<th>GPS info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Did (Did)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time (Time)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 85: Entity Relationship Diagram of the SWERB_DEPARTS View

3.33 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 AS gps,
       swerb_loc_gps.garmincode AS garmincode
FROM swerb_loc_gps;
```

Figure 86: Query Defining the SWERB_LOC_GPS_XY View
3.34 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 88: Query Defining the SWERB_LOCS View
3.35 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 90: 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 91: Entity Relationship Diagram of the SWERB UPLOAD View
3.36 The TISSUES View

SELECT tissue_data.tid AS tid
, tissue_data.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.notes AS notes
FROM tissue_data
JOIN locations
  ON locations.locid = tissue_data.locid
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;

Figure 92: Query Defining the TISSUES View
3.37 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 94: Query Defining the ULNA_STATS View
Figure 95: Entity Relationship Diagram of the ULNA_STATS View
### The TESTES_ARC_STATS View

```
SELECT testesdartids.dartid AS dartid,
       testeslength.testlengthsamps AS testlengthsamps,
       testeslength.testlength_mean AS testlength_mean,
       testeslength.testlength_stddev AS testlength_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 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 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 96: Query Defining the TESTES_ARC_STATS View
Figure 97: Entity Relationship Diagram of the TESTES_ARC_STATS View
3.39 The TESTES_DIAM_STATS View

```sql
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 98: Query Defining the TESTES_DIAM_STATS View
Figure 99: Entity Relationship Diagram of the TESTES_DIAM_STATS View
### 3.40 The WP_DETAILS_AFFECTEDPARTS View

```
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 100:** Query Defining the WP_DETAILS_AFFECTEDPARTS View

**Figure 101:** Entity Relationship Diagram of the WP_DETAILS_AFFECTEDPARTS View
3.41 The WP_HEALS View

WITH concatObservers AS (SELECT wprid, string_agg(observer, '/') observers FROM wp_observers GROUP BY wprid)


Figure 102: Query Defining the WP_HEALS View
Figure 103: Entity Relationship Diagram of the WP_HEALS View, Overall
Figure 104: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology report
Figure 105: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology cluster
Figure 106: Entity Relationship Diagram of the WP_HEALS View for rows with an update to an affected body part
3.42 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 107: Query Defining the WP_REPORTS_OBSERVERS View

Figure 108: 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 109: 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 111: Query Defining the ENTRYDATE_GRP View

![Entity Relationship Diagram of the ENTRYDATE_GRP View](image)
4.3 The STATDATE_GRP View

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

Figure 113: 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 114: Entity Relationship Diagram of the STATDATE_GRP View

4.4 The CONSORTDATES_GRP View

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

Figure 115: 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 116: Entity Relationship Diagram of the CONSORTDATES_GRP View
4.5 The CYCGAPDAYS_GRP View

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

Figure 117: Query Defining the CYCGAPDAYS_GRP View

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

4.6 The CYCGAPS_GRP View

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

Figure 119: Query Defining the CYCGAPS_GRP View

![Entity Relationship Diagram of the CYCGAPS_GRP View](image)
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 121: Query Defining the CYCSTATS_GRP View

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

Figure 122: 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 123: Query Defining the DARTINGS_GRP View

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

Figure 124: 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 125: Query Defining the DISPERSEDATES_GRP View

Figure 126: 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 127: Query Defining the MATUREDATES_GRP View

Figure 128: 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 129: Query Defining the MDINTERVALS_GRP View

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

Figure 130: 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 131: Query Defining the MMINTERVALS_GRP View

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

Figure 132: 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 133: Query Defining the PCSKINS_GRP View**

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

**Figure 134: 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 135: Query Defining the RANKDATES_GRP View**

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

**Figure 136: 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 137: Query Defining the REPSTATS_GRP View

Figure 138: Entity Relationship Diagram of the REPSTATS_GRP View