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

**TITLE:** The Babase Pocket Reference Guide

<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, Jacob B. Gordon, and Leah Gerber</td>
<td>March 6, 2024</td>
<td></td>
</tr>
<tr>
<td>ER Diagram layout and conversion to Dia</td>
<td>Leah Gerber</td>
<td>March 6, 2024</td>
<td></td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Anne Hubbard</td>
<td>March 6, 2024</td>
<td></td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Karl Pinc</td>
<td>March 6, 2024</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>
</tbody>
</table>
# Contents

1 Babase Summarized  
2 The Babase ER Diagrams  
3 The Babase Views

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>The ACTOR_ACTEES View</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>The ANESTH_STATS View</td>
<td>26</td>
</tr>
<tr>
<td>3.3</td>
<td>The BODYTEMP_STATS View</td>
<td>27</td>
</tr>
<tr>
<td>3.4</td>
<td>The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views</td>
<td>28</td>
</tr>
<tr>
<td>3.5</td>
<td>The CHEST_STATS View</td>
<td>29</td>
</tr>
<tr>
<td>3.6</td>
<td>The CROWN_RUMP_STATS View</td>
<td>30</td>
</tr>
<tr>
<td>3.7</td>
<td>The CYCLES_SEXSKINS and CYCLES_SEXSKINS_SORTED Views</td>
<td>30</td>
</tr>
<tr>
<td>3.8</td>
<td>The CYCPOINTS_CYCLES and CYCPOINTS_CYCLES_SORTED Views</td>
<td>31</td>
</tr>
<tr>
<td>3.9</td>
<td>The DSAMPLES View</td>
<td>33</td>
</tr>
<tr>
<td>3.10</td>
<td>The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views</td>
<td>34</td>
</tr>
<tr>
<td>3.11</td>
<td>The DENT_CODES View</td>
<td>36</td>
</tr>
<tr>
<td>3.12</td>
<td>The DENT_SITES View</td>
<td>39</td>
</tr>
<tr>
<td>3.13</td>
<td>The ESTROGENS View</td>
<td>41</td>
</tr>
<tr>
<td>3.14</td>
<td>The GLUCOCORTICOIDS View</td>
<td>43</td>
</tr>
<tr>
<td>3.15</td>
<td>The GROUPS_HISTORY View</td>
<td>45</td>
</tr>
<tr>
<td>3.16</td>
<td>The HORMONE_PREPS View</td>
<td>47</td>
</tr>
<tr>
<td>3.17</td>
<td>The HORMONE_RESULTS View</td>
<td>48</td>
</tr>
<tr>
<td>3.18</td>
<td>The HORMONE_SAMPLES View</td>
<td>49</td>
</tr>
<tr>
<td>3.19</td>
<td>The HUMERUS_STATS View</td>
<td>50</td>
</tr>
<tr>
<td>3.20</td>
<td>The INTERACT and INTERACT_SORTED Views</td>
<td>52</td>
</tr>
<tr>
<td>3.21</td>
<td>The LOCATIONS_FREE View</td>
<td>53</td>
</tr>
<tr>
<td>3.22</td>
<td>The MATERNITIES View</td>
<td>54</td>
</tr>
<tr>
<td>3.23</td>
<td>The MIN_MAXS View</td>
<td>56</td>
</tr>
<tr>
<td>3.24</td>
<td>The MIN_MAXS_SORTED View</td>
<td>58</td>
</tr>
<tr>
<td>3.25</td>
<td>The MPI_EVENTS View</td>
<td>61</td>
</tr>
<tr>
<td>3.26</td>
<td>The MTD_CYCLES View</td>
<td>63</td>
</tr>
<tr>
<td>3.27</td>
<td>The NUCACID_CONCS View</td>
<td>65</td>
</tr>
<tr>
<td>3.28</td>
<td>The NUCACIDS View</td>
<td>66</td>
</tr>
<tr>
<td>3.29</td>
<td>The NUCACIDS_W_CONC View</td>
<td>69</td>
</tr>
<tr>
<td>3.30</td>
<td>The PARENTS View</td>
<td>71</td>
</tr>
<tr>
<td>3.31</td>
<td>The PCV_STATS View</td>
<td>72</td>
</tr>
<tr>
<td>3.32</td>
<td>The POINTS and POINTS_SORTED Views</td>
<td>73</td>
</tr>
</tbody>
</table>
3.33 The POTENTIAL_DADS View ................................................................. 76
3.34 The PROPORTIONAL_RANKS View ..................................................... 81
3.35 The QUADS View ............................................................................ 82
3.36 The SAMPLES_GOFF View ................................................................. 83
3.37 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_SORTED Views ............ 84
3.38 The SEXSKINS_REPRO_NOTES View .................................................... 84
3.39 The SWERB view ............................................................................. 87
3.40 The SWERB_DATA_XY View ............................................................... 89
3.41 The SWERB_DEPARTS view ............................................................... 90
3.42 The SWERB_GW_LOC_DATA_XY View ............................................... 90
3.43 The SWERB_GW_LOCS View ............................................................. 91
3.44 The SWERB_LOC_GPS_XY view ........................................................ 92
3.45 The SWERB_LOCS view ................................................................. 93
3.46 The SWERB_UPLOAD view ............................................................... 94
3.47 The TESTES_ARC_STATS View .......................................................... 96
3.48 The TESTES_DIAM_STATS View ......................................................... 99
3.49 The TESTOSTERONES View ............................................................. 101
3.50 The THYROID_HORMONES View ..................................................... 103
3.51 The TISSUES View ......................................................................... 105
3.52 The TISSUES_HORMONES View ...................................................... 107
3.53 The ULNA_STATS View .................................................................. 108
3.54 The VAGINAL_PH_STATS View ........................................................ 109
3.55 The WOUNDPATHOLOGIES View ..................................................... 111
3.56 The WPDETAILS_AFFECTEDPARTS View ........................................... 113
3.57 The WP_HEALS View ..................................................................... 114
3.58 The WP_REPORTS_OBSERVERS View .............................................. 119

4 Views Which Add Gid To Tables .......................................................... 119
4.1 The BIRTH_GRP View ........................................................................ 120
4.2 The ENTRYDATE_GRP View .............................................................. 120
4.3 The STATDATE_GRP View ................................................................ 121
4.4 The CONSORTDATES_GRP View ....................................................... 121
4.5 The CYCGAPDAYS_GRP View ............................................................ 122
4.6 The CYCGAPS_GRP View ............................................................... 122
4.7 The CYCSTATS_GRP View ............................................................... 123
4.8 The DARTINGS_GRP View ............................................................... 123
4.9 The DISPERSEDATES_GRP View ...................................................... 124
4.10 The MATUREDATES_GRP View ....................................................... 124
4.11 The MDINTERVALS_GRP View ......................................................... 125
4.12 The MMINTERVALS_GRP View ....................................................... 125
4.13 The RANKDATES_GRP View ........................................................... 126
4.14 The REPSTATS_GRP View ............................................................. 126
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 Inventory Entity Relationship Diagram ............................................. 16
13. Babase Physical Traits Hormone Data Entity Relationship Diagram .............. 17
14. Babase Physical Traits Genetic Hybrid Score Data Entity Relationship Diagram . 18
15. Babase Physical Traits Wounds and Pathologies Data Entity Relationship Diagram . 19
16. Babase SWERB Core Tables Entity Relationship Diagram ............................ 20
17. Babase SWERB Grove/Waterhole Location Tables Entity Relationship Diagram .... 21
18. Babase Manual Weather Data Entity Relationship Diagram ............................ 22
20. Warning Sub-System Entity Relationship Diagram ......................................... 24
21. Query Defining the ACTOR_ACTEES View .................................................. 25
22. Entity Relationship Diagram of the ACTOR_ACTEES View ............................. 26
23. Query Defining the ANESTH_STATS View .................................................. 26
24. Entity Relationship Diagram of the ANESTH_STATS View ............................. 27
25. Query Defining the BODYTEMP_STATS View ............................................. 27
26. Entity Relationship Diagram of the BODYTEMP_STATS View ....................... 28
27. Query Defining the CENSUS_DEMOG View ................................................ 28
28. Entity Relationship Diagram of the CENSUS_DEMOG View ............................ 28
29. Query Defining the CHEST_STATS View ..................................................... 29
30. Entity Relationship Diagram of the CHEST_STATS View ............................... 29
31. Query Defining the CROWNRRUMP_STATS View ....................................... 30
32. Entity Relationship Diagram of the CROWNRRUMP_STATS View ................. 30
33. Query Defining the CYCLES_SEXSKINS View ............................................ 30
34. Entity Relationship Diagram of the CYCLES_SEXSKINS View ...................... 31
35. Query Defining the CYCPOINTS_CYCLES View ......................................... 31
36. Entity Relationship Diagram of the CYCPOINTS_CYCLES View ................. 31
37. Query Defining the DSAMPLES View ....................................................... 33
115 Query Defining the TESTES_ARC_STATS View ......................................................... 96
116 Entity Relationship Diagram of the TESTES_ARC_STATS View .......................... 97
117 Query Defining the TESTES_DIAM_STATS View ..................................................... 99
118 Entity Relationship Diagram of the TESTES_DIAM_STATS View ..................... 100
119 Query Defining the TESTOSTERONES View ......................................................... 101
120 Entity Relationship Diagram of the TESTOSTERONES View ......................... 102
121 Query Defining the THYROID_HORMONES View ............................................. 103
122 Entity Relationship Diagram of the THYROID_HORMONES View ................. 104
123 Query Defining the TISSUES View ................................................................. 105
124 Entity Relationship Diagram of the TISSUES View ....................................... 106
125 Query Defining the TISSUES_HORMONES View ........................................... 107
126 Entity Relationship Diagram of the TISSUES_HORMONES View ................ 108
127 Query Defining the ULNA_STATS View .......................................................... 108
128 Entity Relationship Diagram of the ULNA_STATS View .................................. 109
129 Query Defining the VAGINAL_PH_STATS View ............................................ 109
130 Entity Relationship Diagram of the VAGINAL_PH_STATS View .................. 110
131 Query Defining the WOUNDSPATHOLOGIES View ...................................... 111
132 Entity Relationship Diagram of the WOUNDSPATHOLOGIES View ............. 112
133 Query Defining the WP_DETAILS_AFFECTEDPARTS View ............................ 113
134 Entity Relationship Diagram of the WP_DETAILS_AFFECTEDPARTS View ...... 113
135 Query Defining the WP_HEALS View ............................................................ 114
136 Entity Relationship Diagram of the WP_HEALS View, Overall ...................... 115
137 Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology report 116
138 Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology cluster 117
139 Entity Relationship Diagram of the WP_HEALS View for rows with an update to an affected body part 118
140 Query Defining the WP_REPORTS_OBSERVERS View .................................. 119
141 Entity Relationship Diagram of the WP_REPORTS_OBSERVERS View ................ 119
142 Query Defining the BIRTH_GRP View .......................................................... 120
143 Entity Relationship Diagram of the BIRTH_GRP View .................................... 120
144 Query Defining the ENTRYDATE_GRP View ................................................. 120
145 Entity Relationship Diagram of the ENTRYDATE_GRP View .......................... 120
146 Query Defining the STATDATE_GRP View .................................................... 121
147 Entity Relationship Diagram of the STATDATE_GRP View .......................... 121
148 Query Defining the CONSORTDATES_GRP View ......................................... 121
149 Entity Relationship Diagram of the CONSORTDATES_GRP View .................. 121
150 Query Defining the CYCGAPDAYS_GRP View ............................................. 122
151 Entity Relationship Diagram of the CYCGAPDAYS_GRP View ...................... 122
152 Query Defining the CYCGAPS_GRP View ..................................................... 122
153 Entity Relationship Diagram of the CYCGAPS_GRP View ............................ 122
List of Tables

1. The Main Babase Tables ................................................................. 2
2. The Warning Sub-System Tables .................................................... 3
3. The Babase SupportTables .............................................................. 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

Figure 1: Key to the Babase Entity Relationship Diagrams

\footnote{At this time of this writing only males have data entered into \texttt{RANKDATES} in Babase.}
### Group Membership and Life Events

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTERNATE_SNAMES in Babase:</td>
<td>One row for each</td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>rescinded sname</td>
</tr>
<tr>
<td>CENSUS in Babase:</td>
<td>animal, including fetuses</td>
</tr>
<tr>
<td>CONSORIDATES in Babase:</td>
<td>day each individual is (or is not) observed in a group</td>
</tr>
<tr>
<td>DEMOG in Babase:</td>
<td>male who has a known first consortship</td>
</tr>
<tr>
<td>DISPERSEDATES in Babase:</td>
<td>mention of an individual’s presence in a group within a field textual note</td>
</tr>
<tr>
<td>GROUPS in Babase:</td>
<td>individual who is sexually mature</td>
</tr>
<tr>
<td>MATUREDATES in Babase:</td>
<td>individual who has attained adult rank</td>
</tr>
<tr>
<td>RANKDATES in Babase:</td>
<td></td>
</tr>
</tbody>
</table>

### Analyzed: Group Membership and Life Events

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAD_DATA in Babase:</td>
<td>offspring having a paternity analysis</td>
</tr>
<tr>
<td>MEMBERS in Babase:</td>
<td>day each individual is alive</td>
</tr>
<tr>
<td>RANKDATES in Babase:</td>
<td>month each individual is ranked in each group</td>
</tr>
<tr>
<td>RESIDENCIES in Babase:</td>
<td>bout of each individual’s residency</td>
</tr>
</tbody>
</table>

### Physical Traits

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP_AFFECTEDPARTS in Babase:</td>
<td>body part affected by a specific wound/pathology</td>
</tr>
<tr>
<td>WP_DETAILS in Babase:</td>
<td>wound or pathology cluster indicated on a report</td>
</tr>
<tr>
<td>WP_HEALUPDATES in Babase:</td>
<td>update on progress of wound/pathology healing</td>
</tr>
<tr>
<td>WP_REPORTS in Babase:</td>
<td>wound/pathology report</td>
</tr>
</tbody>
</table>

### Sexual Cycles

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYCGAPS in Babase:</td>
<td>female for each initiation or cessation of a continuous period of observation</td>
</tr>
<tr>
<td>CYCLES in Babase:</td>
<td>female’s cycle (complete or not)</td>
</tr>
<tr>
<td>CYCPOINTS in Babase:</td>
<td>Mdate (menses), Tdate (turgesence onset), or Ddate (deturgesence onset) date of each female</td>
</tr>
<tr>
<td>PREGS in Babase:</td>
<td>time a female becomes pregnant</td>
</tr>
<tr>
<td>SEXSKINS in Babase:</td>
<td></td>
</tr>
</tbody>
</table>
Table 2: The Warning Sub-System Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>

Figure 2: Babase Group Membership Entity Relationship Diagram
<table>
<thead>
<tr>
<th>General Support Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table</strong></td>
</tr>
<tr>
<td>BODYPARTS in Babase:</td>
</tr>
<tr>
<td>LAB_PERSONNEL in Babase:</td>
</tr>
<tr>
<td>OBSERVERS in Babase:</td>
</tr>
<tr>
<td>OBSERVER_ROLES in Babase:</td>
</tr>
<tr>
<td>UNKSNAMES in Babase:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group Membership and Life Events</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table</strong></td>
</tr>
<tr>
<td>BSTATUSES in Babase:</td>
</tr>
<tr>
<td>CONFIDENCES in Babase:</td>
</tr>
</tbody>
</table>
Table 4: The Warning Sub-System Support Tables

<table>
<thead>
<tr>
<th>ALTERNATE_SNAMES</th>
<th>RANKS</th>
<th>BIOGRAPH</th>
<th>DISPERSEDATES</th>
<th>CONSORTDATES</th>
<th>MATUREDATES</th>
<th>DAD_DATA</th>
<th>GROUPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname</td>
<td>Rkld</td>
<td>Biod</td>
<td>Sname</td>
<td>Sname</td>
<td>Sname</td>
<td>Dadd</td>
<td>Gid</td>
</tr>
<tr>
<td>Sname_Alternate</td>
<td>Sname</td>
<td>Sname</td>
<td>Dispersed</td>
<td>Rank</td>
<td>Mature</td>
<td>Kid</td>
<td>Name</td>
</tr>
<tr>
<td>Name_Alternate</td>
<td>Rank</td>
<td>Birth</td>
<td>Dispos</td>
<td>Consor</td>
<td>Matured</td>
<td>Mom</td>
<td>From_group</td>
</tr>
<tr>
<td>Notes</td>
<td>Age</td>
<td>Bstarus</td>
<td>Dispos*</td>
<td>Consor*</td>
<td>Maturity</td>
<td>Dad</td>
<td>Permanent</td>
</tr>
<tr>
<td>Sys_Period</td>
<td>Height</td>
<td>Sex</td>
<td>Dispos*</td>
<td>Consor*</td>
<td>Sys_Period</td>
<td>Loc</td>
<td>Start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Matirgr</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td>Pdad</td>
<td>Cease_To_Exist</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MatirgrConf</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td>Pdad</td>
<td>Last_Reg_Census</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Entlydate</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td>Pdad</td>
<td>Three_letter_code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Entlytype</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td>Pdad</td>
<td>One_letter_code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dcaus*</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td>Pdad</td>
<td>Study_Grp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DcausNaureConf</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td>Pdad</td>
<td>To_group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DcausAgentConf</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td></td>
<td>Sys_Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EarliestBirth</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LatestBirth</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sys_Period</td>
<td>Dispos*</td>
<td>Consor*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 12: IQTYPES in Babase

<table>
<thead>
<tr>
<th>ID Column</th>
<th>Related Column(s)</th>
<th>One entry for every possible choice of...</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQType</td>
<td>INTEGRITY_QUERIES in Babase: Type in Babase:</td>
<td>kind of problem with data integrity</td>
</tr>
<tr>
<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>
### Group Membership and Life Events

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
<th>Tables/Views used</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENSUS_DEMOG in Babase:</td>
<td>Maintenance of CENSUS row that are extended with DEMOG information.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>CENSUS_DEMOG_SORTED in Babase:</td>
<td>Maintenance of CENSUS row in a pre-sorted fashion.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>CYCPOINTS_CYCLES in Babase:</td>
<td>Maintenance of CYCPOINTS row that are extended with CYCLES information.</td>
<td>CYCLES in Babase:, CYCPOINTS in Babase:</td>
</tr>
<tr>
<td>CYCPOINTS_CYCLES_SORTED in Babase:</td>
<td>The CYCPOINTS 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>
</tr>
<tr>
<td>DEMOG_CENSUS in Babase:</td>
<td>Maintenance of DEMOG row.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>DEMOG_CENSUS_SORTED in Babase:</td>
<td>Maintenance of DEMOG row in a pre-sorted fashion.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>GROUPS_HISTORY in Babase:</td>
<td>Depiction of GROUPS row in a more human-readable format.</td>
<td>GROUPS in Babase:</td>
</tr>
<tr>
<td>PARENTS in Babase:</td>
<td>Easy access to parental information.</td>
<td>BIOGRAPH in Babase:, MATERITIES in Babase:, MATURE_DATES in Babase:, MEMBERS in Babase:</td>
</tr>
<tr>
<td>POTENTIAL_DADS in Babase:</td>
<td>Research into paternity, especially the selection of potential fathers for further genetic testing.</td>
<td>MATERITIES in Babase:, MEMBERS in Babase: (multiple times), BIOGRAPH in Babase:, RANK_DATES in Babase:, MATURE_DATES in Babase:</td>
</tr>
<tr>
<td>PROPORTIONAL_RANKS in Babase:</td>
<td>Automatic calculation of proportional ranks from the ordinal ranks in RANKS</td>
<td>RANKS in Babase:</td>
</tr>
</tbody>
</table>

### Physical Traits

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
<th>Tables/Views used</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPORTIONAL_RANKS in Babase:</td>
<td>Automatic calculation of proportional ranks from the ordinal ranks in RANKS in Babase:</td>
<td>RANKS in Babase:</td>
</tr>
<tr>
<td>Table</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>BIRTH_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>ENTRYDATE_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>STATDATE_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>CONSORTDATES in Babase:</td>
<td>CONSORTDATES_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>CYCGAPDAYS in Babase:</td>
<td>CYCGAPDAYS_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>CYCGAPS in Babase:</td>
<td>CYCGAPS_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>CYCSTATS in Babase:</td>
<td>CYCSTATS_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>DARTINGS in Babase:</td>
<td>DARTINGS_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>DISPERSEDATES in Babase:</td>
<td>DISPERSEDATES_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>MATUREDATES in Babase:</td>
<td>MATUREDATES_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>MDINTERVALS in Babase:</td>
<td>MDINTERVALS_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>MMINTERVALS in Babase:</td>
<td>MMINTERVALS_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>RANKDATES in Babase:</td>
<td>RANKDATES_GRP in Babase:</td>
<td></td>
</tr>
<tr>
<td>REPSTATS in Babase:</td>
<td>REPSTATS_GRP in Babase:</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 Darter Teeth and Ticks Entity and Relationship Diagram
Figure 12: Babase Inventory Entity Relationship Diagram
Figure 13: Babase Physical Traits Hormone Data Entity Relationship Diagram
Figure 14: Babase Physical Traits Genetic Hybrid Score Data Entity Relationship Diagram
Figure 15: Babase Physical Traits Wounds and Pathologies Data Entity Relationship Diagram
Figure 16: Babase SWERB Core Tables Entity Relationship Diagram
Figure 17: Babase SWERB Grove/Waterhole Location Tables Entity Relationship Diagram
Figure 18: Babase Manual Weather Data Entity Relationship Diagram
Figure 19: Babase Digital Weather 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,
      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,
      interact_data.exact_date AS exact_date
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 21: 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 23: 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 25: 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 27: Query Defining the CENSUS_DEMOG View

Figure 26: Entity Relationship Diagram of the BODYTEMP_STATS View

Figure 28: 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 29: Query Defining the CHEST_STATS View

Figure 30: 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 31: Query Defining the CROWNRUMP_STATS View

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

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,
       sexskins.color AS color
FROM cycles LEFT OUTER JOIN sexskins ON (cycles.cid = sexskins.cid);
```

Figure 33: 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 35: Query Defining the CYCPOINTS_CYCLES View

Figure 36: Entity Relationship Diagram of the CYCPOINTS_CYCLES View
3.9 The DSAMPLES View

SELECT dartings.dartid,
    dartings.sname,
    dartings.date,
    members.grp,
    blood_unspecs.num AS bloodunspec,
    blood_paxgenes.num AS bloodpaxgene,
    blood_purpletops.num AS bloodpurpletops,
    blood_separators.num AS bloodseptube,
    blood_cpts.num AS bloodcpt,
    blood_trucultures.num AS bloodtruculture,
    blood_smears.num AS bloodsmear,
    tc_bloods.num AS tcblood,
    hair_unspecs.num AS hairunspec,
    hair_lengths.num AS hairlength,
    hair_cu_zns.num AS haircu_zn,
    teeth_3mouths.num AS mouthphotos3,
    teeth_lmandmolds.num AS lmandmold,
    teeth_lmaxmolds.num AS lmaxillamold,
    teeth_lmollmol2s.num AS lm1m2siliconemold,
    skin_punchs.num AS skinpunch,
    tc_skins.num AS tcskin,
    vag_swabs.num AS vaginalsawb,
    cerv_swabs.num AS cervicalswab,
    fecal_formalin.num AS fecal_formalin,
    palm_swab.num AS palm_swab,
    tongue_swab.num AS tongue_swab,
    tooth_plaque_swab.num AS tooth_plaque_swab,
    vagswab_microbiome.num AS vagswab_microbiome,
    glans_penis_swab.num AS glans_penis_swab,
    fecal_microbiome.num AS fecal_microbiome,
    nostrils_swab.num AS nostrils_swab,
    skin_behind_ear_swab.num AS skin_behind_ear_swab,
    skin_inside_elbow_swab.num AS skin_inside_elbow_swab
FROM dartings
    JOIN members
        ON dartings.sname = members.sname
        AND dartings.date = members.date
    LEFT JOIN dart_samples blood_unspecs
        ON dartings.dartid = blood_unspecs.dartid
        AND blood_unspecs.ds_type = 1
    LEFT JOIN dart_samples blood_paxgenes
        ON dartings.dartid = blood_paxgenes.dartid
        AND blood_paxgenes.ds_type = 2
    LEFT JOIN dart_samples blood_purpletops
        ON dartings.dartid = blood_purpletops.dartid
        AND blood_purpletops.ds_type = 3
    LEFT JOIN dart_samples blood_separators
        ON dartings.dartid = blood_separators.dartid
        AND blood_separators.ds_type = 4
    LEFT JOIN dart_samples blood_cpts
        ON dartings.dartid = blood_cpts.dartid
        AND blood_cpts.ds_type = 5
    LEFT JOIN dart_samples blood_trucultures
        ON dartings.dartid = blood_trucultures.dartid
        AND blood_trucultures.ds_type = 6
    LEFT JOIN dart_samples blood_smears
        ON dartings.dartid = blood_smears.dartid
        AND blood_smears.ds_type = 7
    LEFT JOIN dart_samples hair_unspecs
        ON dartings.dartid = hair_unspecs.dartid
        AND hair_unspecs.ds_type = 8
        JOIN dart_samples hair_lengths
            ON dartings.dartid = hair_lengths.dartid
            AND hair_lengths.ds_type = 9
        LEFT JOIN dart_samples hair_cu_zns
            ON dartings.dartid = hair_cu_zns.dartid
            AND hair_cu_zns.ds_type = 10
        LEFT JOIN dart_samples teeth_3mouths
            ON dartings.dartid = teeth_3mouths.dartid
            AND teeth_3mouths.ds_type = 11
        LEFT JOIN dart_samples teeth_lmandmolds
            ON dartings.dartid = teeth_lmandmolds.dartid
            AND teeth_lmandmolds.ds_type = 12
        LEFT JOIN dart_samples teeth_lmaxmolds
            ON dartings.dartid = teeth_lmaxmolds.dartid
            AND teeth_lmaxmolds.ds_type = 13
        LEFT JOIN dart_samples teeth_lmollmol2s
            ON dartings.dartid = teeth_lmollmol2s.dartid
            AND teeth_lmollmol2s.ds_type = 14
        LEFT JOIN dart_samples skin_punchs
            ON dartings.dartid = skin_punchs.dartid
            AND skin_punchs.ds_type = 15
        LEFT JOIN dart_samples vag_swabs
            ON dartings.dartid = vag_swabs.dartid
            AND vag_swabs.ds_type = 16
        LEFT JOIN dart_samples cerv_swabs
            ON dartings.dartid = cerv_swabs.dartid
            AND cerv_swabs.ds_type = 17
        LEFT JOIN dart_samples tc_bloods
            ON dartings.dartid = tc_bloods.dartid
            AND tc_bloods.ds_type = 18
        LEFT JOIN dart_samples tc_skins
            ON dartings.dartid = tc_skins.dartid
            AND tc_skins.ds_type = 19
        LEFT JOIN dart_samples fecal_formalin
            ON dartings.dartid = fecal_formalin.dartid
            AND fecal_formalin.ds_type = 20
        LEFT JOIN dart_samples palm_swab
            ON dartings.dartid = palm_swab.dartid
            AND palm_swab.ds_type = 22
        LEFT JOIN dart_samples tongue_swab
            ON dartings.dartid = tongue_swab.dartid
            AND tongue_swab.ds_type = 23
        LEFT JOIN dart_samples tooth_plaque_swab
            ON dartings.dartid = tooth_plaque_swab.dartid
            AND tooth_plaque_swab.ds_type = 24
        LEFT JOIN dart_samples vagswab_microbiome
            ON dartings.dartid = vagswab_microbiome.dartid
            AND vagswab_microbiome.ds_type = 25
        LEFT JOIN dart_samples glans_penis_swab
            ON dartings.dartid = glans_penis_swab.dartid
            AND glans_penis_swab.ds_type = 26
        LEFT JOIN dart_samples fecal_microbiome
            ON dartings.dartid = fecal_microbiome.dartid
            AND fecal_microbiome.ds_type = 27
        LEFT JOIN dart_samples nostrils_swab
            ON dartings.dartid = nostrils_swab.dartid
            AND nostrils_swab.ds_type = 28
        LEFT JOIN dart_samples skin_behind_ear_swab
            ON dartings.dartid = skin_behind_ear_swab.dartid
            AND skin_behind_ear_swab.ds_type = 29
        LEFT JOIN dart_samples skin_inside_elbow_swab
            ON dartings.dartid = skin_inside_elbow_swab.dartid
            AND skin_inside_elbow_swab.ds_type = 30;
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 38: Query Defining the DEMOG_CENSUS View

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

Figure 39: 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
    , lui1.lui1tstate AS lui1tstate
    , lui1.lui1tcondition AS lui1tcondition
    , lui2.lui2tstate AS lui2tstate
    , lui2.lui2tcondition AS lui2tcondition
    , luc.luctstate AS luctstate
    , luc.luctcondition AS luctcondition
    , lup1.lup1tstate AS lup1tstate
    , lup1.lup1tcondition AS lup1tcondition
    , lup2.lup2tstate AS lup2tstate
    , lup2.lup2tcondition AS lup2tcondition
    , lum1.lum1tstate AS lum1tstate
    , lum1.lum1tcondition AS lum1tcondition
    , lum2.lum2tstate AS lum2tstate
    , lum2.lum2tcondition AS lum2tcondition
    , lum3.lum3tstate AS lum3tstate
    , lum3.lum3tcondition AS lum3tcondition

    , llm3.llm3tstate AS llm3tstate
    , llm3.llm3tcondition AS llm3tcondition
    , llm2.llm2tstate AS llm2tstate
    , llm2.llm2tcondition AS llm2tcondition
    , llm1.llm1tstate AS llm1tstate
    , llm1.llm1tcondition AS llm1tcondition
    , llp2.llp2tstate AS llp2tstate
    , llp2.llp2tcondition AS llp2tcondition
    , llp1.llp1tstate AS llp1tstate
    , llp1.llp1tcondition AS llp1tcondition
    , llc.llctstate AS llctstate
    , llc.llctcondition AS llctcondition
    , llu2.llu2tstate AS llu2tstate
    , llu2.llu2tcondition AS llu2tcondition
    , llu1.llu1tstate AS llu1tstate
    , llu1.llu1tcondition AS llu1tcondition
    , rli1.rli1tstate AS rli1tstate
    , rli1.rli1tcondition AS rli1tcondition
    , rli2.rli2tstate AS rli2tstate
    , rli2.rli2tcondition AS rli2tcondition
    , rlc.rlctstate AS rlctstate
    , rlc.rlctcondition AS rlctcondition
    , rlpl1.rlpl1tstate AS rlpl1tstate
    , rlpl1.rlpl1tcondition AS rlpl1tcondition
    , rlpl2.rlpl2tstate AS rlpl2tstate
    , rlpl2.rlpl2tcondition AS rlpl2tcondition
    , rlpl1.rlpl1tstate AS rlpl1tstate
```
Figure 40: Query Defining the DENT_CODES View
Figure 41: Entity Relationship Diagram of the DENT_CODES View
3.12 The DENT_SITES View

SELECT teethdartids.dartid AS dartid
  , s1.sltstate AS sltstate
  , s1.sltcondition AS sltcondition
  , 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

Figure 42: Query Defining the DENT_SITES View
Figure 43: Entity Relationship Diagram of the DENT_SITES View

- The Tstate value of the TEETH row having the correct Dartid and a related Toothsite value corresponding with the Toothsite code appearing in the column name, or NULL if no such row exists.

- The Tcondition value of the TEETH row having the correct Dartid and a related Toothsite value corresponding with the Toothsite code appearing in the column name, or NULL if no such row exists.

- The Deciduous value of the TOOTHCODES row related to the TEETH row having the correct Dartid and a Toothsite value corresponding with the Toothsite code appearing in the column name, or NULL if no such row exists.

+ These columns repeat: there is a set of these columns for every distinct TOOTHCODES Toothsite value. The “TS” shown here in each column name is replaced in the actual column name with the letter “s” followed by a TOOTHCODE Toothsite value.
3.13 The ESTROGENS View

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

Figure 44: Query Defining the ESTROGENS View
Figure 45: Entity Relationship Diagram of the ESTROGENS View
3.14 The GLUCOCORTICOIDS View

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

Figure 46: Query Defining the GLUCOCORTICOIDS View
Figure 47: Entity Relationship Diagram of the GLUCOCORTICOIDS View
3.15 The GROUPS_HISTORY View

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

Figure 48: Query Defining the GROUPS_HISTORY View
Figure 49: Entity Relationship Diagram of the GROUPS_HISTORY View
3.16 The HORMONE_PREPS View

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

Figure 50: Query Defining the HORMONE_PREPS View

Figure 51: Entity Relationship Diagram of the HORMONE_PREPS View
3.17 The HORMONE_RESULTS View

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

Figure 52: Query Defining the HORMONE_RESULTS View
### 3.18 The HORMONE_SAMPLES View

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

Figure 54: Query Defining the HORMONE_SAMPLES View
3.19 The HUMERUS_STATS View

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

Figure 56: Query Defining the HUMERUS_STATS View
Figure 57: Entity Relationship Diagram of the HUMERUS_STATS View
3.20 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,
      interact_data.exact_date AS exact_date
FROM interact_data
JOIN acts
  ON (acts.act = interact_data.act);
```

Figure 58: Query Defining the INTERACT View

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

Figure 59: Entity Relationship Diagram of the INTERACT View
### 3.21 The LOCATIONS_FREE View

```sql
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 60: Query Defining the LOCATIONS_FREE View

Figure 61: Entity Relationship Diagram of the LOCATIONS_FREE View
3.22 The MATERNITIES View

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

Figure 62: Query Defining the MATERNITIES View
Figure 63: 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.)
### 3.23 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 64: Query Defining the MIN_MAXS View
Figure 65: Entity Relationship Diagram of the MIN_MAXS View
### 3.24 The MIN_MAXS_SORTED View

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

Figure 66: Query Defining the MIN_MAXS_SORTED View
Figure 67: Entity Relationship Diagram of the MIN_MAXS_SORTED View
### 3.25 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.mpiid
                 AND requestor.role = 'R'
                 AND requestor.sname = actee.sname
                 AND requestee.mpidid = request.mpiid
                 AND requestee.role = 'E'
                 AND requestee.sname = actor.sname)
       ELSE
         NULL
       END AS solicited,
       EXISTS(SELECT 1
                FROM mpi_data AS initial,
                    mpiacts
                WHERE initial.mpiid = mpi_data.mpiid
                AND initial.seq = 1
                AND mpiacts.mpiact = initial.mpiact
                AND mpiacts.decided)
       AS decided,
       mpi_data.helped AS helped,
       mpi_data.active AS active
FROM mpis
LEFT OUTER JOIN mpi_data ON (mpis.mpiid = mpi_data.mpiid)
LEFT OUTER JOIN mpi_parts AS actor
  ON (actor.mpidid = mpi_data.mpidid AND actor.role = 'R')
LEFT OUTER JOIN mpi_parts AS actee
  ON (actee.mpidid = mpi_data.mpidid AND actee.role = 'E');
```

Figure 68: Query Defining the MPI_EVENTS View
Figure 69: Entity Relationship Diagram of the MPI_EVENTS View
3.26 The MTD_CYCLES View

```sql
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 ldate,
       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 70: Query Defining the MTD_CYCLES View
Figure 71: Entity Relationship Diagram of the MTD_CYCLES View
### 3.27 The NUCACID_CONCS View

```sql
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 72: Query Defining the NUCACID_CONCS View

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

Figure 73: Entity Relationship Diagram of the NUCACID_CONCS View
3.28 The NUCACIDS View

WITH concat_creators AS (SELECT naid, string_agg(creator, '/' ORDER BY naid, nacrid) AS created_by FROM nucacidCreators 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, 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 FROM nucacid_data
JOIN locations ON locations.locid = nucacid_data.locid
JOIN tissue_data ON tissue_data.tid = nucacid_data.tid
JOIN unique_indivs ON unique_indivs.uid = tissue_data.uid
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 74: Query Defining the NUCACIDS View
Figure 75: Entity Relationship Diagram of the NUCACIDS View

* CONCAT_CREATORS is a subquery from NUCACID_CREATORS, in which all Creator rows for an NAId are concatenated (but separated by a "/") into a single string. It does not appear anywhere as an independent entity.
3.29 The NUCACIDS_W_CONC View

WITH last_quants AS (SELECT DISTINCT
  naid,
  conc_method,
  last_value(pg_ul) OVER w AS last_pg_ul,
  last_value(conc_date) OVER w AS lastdate
FROM nucacid_conc_data
WHERE conc_date IS NOT NULL
WINDOW w AS (PARTITION BY naid, conc_method
ORDER BY conc_date
  RANGE BETWEEN UNBOUNDED PRECEDING
  AND UNBOUNDED FOLLOWING))
,
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.uiid AS uiid,
  unique_indivs.popid AS popid,
  unique_indivs.individ AS individ,
  biograph.sname AS sname,
  nucacid_data.name_on_tube AS name_on_tube,
  nucacid_data.nucacid_type AS nucacid_type,
  tissue_data.tissue_type AS tissue_type,
  nucacid_data.creation_date AS creation_date,
  concat_creators.created_by AS created_by,
  nucacid_data.creation_method AS creation_method,
  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 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
LEFT JOIN last_quants AS qpcr
  ON qpcr.conc_method = 1
  AND qpcr.naid = nucacid_data.naid
LEFT JOIN last_quants AS nanodrop
  ON nanodrop.conc_method = 2
  AND nanodrop.naid = nucacid_data.naid
LEFT JOIN last_quants AS qubit
  ON qubit.conc_method = 3
  AND qubit.naid = nucacid_data.naid
LEFT JOIN last_quants AS bioanalyzer
  ON bioanalyzer.conc_method = 4
  AND bioanalyzer.naid = nucacid_data.naid
LEFT JOIN last_quants AS quantit
  ON quantit.conc_method = 5
  AND quantit.naid = nucacid_data.naid;
Figure 77: Entity Relationship Diagram of the NUCACIDS_W_CONC View
3.30 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 78: Query Defining the PARENTS View
### 3.31 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 80: Query Defining the PCV_STATS View
3.32 The POINTS and POINTS_SORTED Views

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

Figure 82: Query Defining the POINTS View
**POINT_DATA**

- Pntid (Pntid)
- Sid (Sid)
- Activity * (Activity)
- Posture * (Posture)
- Foodcode * (Foodcode)
- Ptime (Ptime)
- (Ptimespm)

Figure 83: Entity Relationship Diagram of the POINTS View
3.33 The POTENTIAL_DADS View

```
SELECT maternities.child_bioid AS bioid
, maternities.child AS kid
, maternities.mom AS mom
, maternities.zdate AS zdate
, maternities.zdate_grp AS grp
, pdads.sname AS pdad
, CASE
  WHEN rankdates.ranked <= maternities.zdate
    THEN 'A'
  WHEN maturedates.matured <= maternities.zdate
    THEN 'S'
  ELSE 'O'
END AS status
, maternities.zdate - pdads.birth AS pdad_age_days
, trunc((maternities.zdate - pdads.birth) / 365.25, 1) AS pdad_age_years
, (SELECT count(*)
  FROM members AS dadmembers
  JOIN members AS mommembers
  ON (mommembers.date = dadmembers.date
      AND mommembers.supergroup = dadmembers.supergroup)
  WHERE dadmembers.sname = pdads.sname
  AND dadmembers.date < maternities.zdate
  AND dadmembers.date >= maternities.zdate - 5
  AND mommembers.sname = maternities.mom
  AND mommembers.date < maternities.zdate
  AND mommembers.date >= maternities.zdate - 5)
AS estrous_presence
, (SELECT count(*)
  FROM actor_actees
  WHERE actor_actees.date < maternities.zdate
  AND actor_actees.date >= maternities.zdate - 5
  AND (actor_actees.act = 'M'
      OR actor_actees.act = 'E')
  AND actor_actees.actor = pdads.sname
  AND actor_actees.actee = maternities.mom)
AS estrous_me
, (SELECT count(*)
  FROM actor_actees
  WHERE actor_actees.date < maternities.zdate
  AND actor_actees.date >= maternities.zdate - 5
  AND actor_actees.act = 'C'
  AND actor_actees.actor = pdads.sname
  AND actor_actees.actee = maternities.mom)
AS estrous_c
FROM maternities
JOIN biograph AS pdads
ON (pdads.sname
  IN (SELECT dadmembers.sname
    FROM members AS dadmembers
    JOIN members AS mommembers
    ON (mommembers.date = dadmembers.date
        AND mommembers.supergroup = dadmembers.supergroup)
    WHERE dadmembers.sname = pdads.sname
    AND dadmembers.date < maternities.zdate
    AND dadmembers.date >= maternities.zdate - 5
    AND mommembers.sname = maternities.mom
    AND mommembers.date < maternities.zdate
    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.
AND pdads.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 84: Query Defining the POTENTIAL_DADS View
Figure 85: 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 86: 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 87: 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 88: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View involving social interactions**

**MATERNTIES**
- Child_Biod (Biod)
- Child (Kot)
- Mom (Mom)
- Zdate (Zdate)
- Zdate_Grp (Grp)

**ACTOR_ACTEES**
- Date < Zdate
- AND Date >> Zdate - 5
- Actor (Pdad)
- Actee (Mom)

---

**POTENTIAL_DADS**
* (Pdad)
  - (Estrous_presence)
  - (Status)
  - (Pdad_age_days)
  - (Pdad_age_years)
  - (Estrous_line)
  - (Estrous_c)

---

**Notes:**
- PDADS is an alias for BIOGRAPH, representing those BIOGRAPH rows that satisfy the conditions required to be considered a potential dad of a given kit.
- 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.

**Comment:**
- This subquery on ACTOR ACTEES is repeated twice in the view, once to compute Estrous me and once to compute Estrous_c.
3.34 The PROPORTIONAL_RANKS View

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

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

Figure 89: Query Defining the PROPORTIONAL_RANKS View

Figure 90: Entity Relationship Diagram of the PROPORTIONAL_RANKS View

*R 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.
3.35 The QUADS View

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

Figure 91: Query Defining the QUADS View

Figure 92: Entity Relationship Diagram of the QUADS View
3.36 The SAMPLES_GOFF View

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

Figure 93: Query Defining the SAMPLES_GOFF View

Figure 94: Entity Relationship Diagram of the SAMPLES_GOFF View
### 3.37 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,
       sexskins.color AS color
FROM sexskins, cycles
WHERE cycles.cid = sexskins.cid
ORDER BY cycles.sname, sexskins.date;
```

**Figure 95: Query Defining the SEXSKINS_CYCLES View**

<table>
<thead>
<tr>
<th>SEXSKINS</th>
<th>CYCLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sxid (Sxid)</td>
<td>Cid (Cid)</td>
</tr>
<tr>
<td>Cid (Cid)</td>
<td></td>
</tr>
<tr>
<td>Date (Date)</td>
<td>Sname (Sname)</td>
</tr>
<tr>
<td>Size (Size)</td>
<td>Seq (Seq)</td>
</tr>
<tr>
<td>Color (Color)</td>
<td>Series (Series)</td>
</tr>
</tbody>
</table>

The cycle’s sexskin measurements

The sexskin measurements’ cycle

### 3.38 The SEXSKINS_REPRO_NOTES View

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

**Figure 96: Entity Relationship Diagram of the SEXSKINS_CYCLES View**

**Figure 97: Query Defining the SEXSKINS_REPRO_NOTES View**
Figure 98: Entity Relationship Diagram of the SEXSKINS_REPRO_NOTES View
3.39 The SWERB view

```
SELECT swerb_data.swid AS swid,
       swerb_departs_data.did AS did,
       swerb_departs_data.date AS date,
       swerb_data.time AS time,
       swerb_bes.beid AS beid,
       swerb_bes.focal_grp AS focal_grp,
       swerb_bes.seq AS seq,
       swerb_data.event AS event,
       swerb_data.seen_grp AS seen_grp,
       swerb_data.lone_animal AS lone_animal,
       swerb_data.quad AS quad,
       CASE
           WHEN swerb_data.quad IS NOT NULL
               THEN 'quad'
           ELSE 'n/a'
       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
```

Figure 99: Query Defining the SWERB View
Figure 100: Entity Relationship Diagram of the SWERB View
3.40 The SWERB_DATA_XY View

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

Figure 101: Query Defining the SWERB_DATA_XY View

Figure 102: Entity Relationship Diagram of the SWERB_DATA_XY View
### 3.41 The SWERB_DEPARTS view

SELECT swerb_departs_data.did AS did,
    swerb_departs_data.date AS date,
    swerb_departs_data.time AS time,
    ST_X(swerb_departs_gps.xyloc) AS x,
    ST_Y(swerb_departs_gps.xyloc) AS y,
    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 103: Query Defining the SWERB_DEPARTS View

<table>
<thead>
<tr>
<th>SWERB_DEPARTS_DATA</th>
<th>SWERB_DEPARTS_GPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Did (Ident)</td>
<td>Did (Ident)</td>
</tr>
<tr>
<td>Date (Date)</td>
<td>(X)</td>
</tr>
<tr>
<td>Time (Time)</td>
<td>(Y)</td>
</tr>
<tr>
<td>GPS info</td>
<td>Altitude (Altitude)</td>
</tr>
<tr>
<td></td>
<td>PDOP (PDOP)</td>
</tr>
<tr>
<td></td>
<td>Accuracy (Accuracy)</td>
</tr>
<tr>
<td></td>
<td>GPS *(GPS)</td>
</tr>
</tbody>
</table>

Figure 104: Entity Relationship Diagram of the SWERB_DEPARTS View

### 3.42 The SWERB_GW_LOC_DATA_XY View

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

Figure 105: Query Defining the SWERB_GW_LOC_DATA_XY View
3.43 The SWERB_GW_LOCS View

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

Figure 109: Query Defining the SWERB_LOC_GPS_XY View
### Figure 110: Entity Relationship Diagram of the SWERB_LOC_GPS_XY View

### 3.45 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 111: Query Defining the SWERB_LOCS View

### Figure 112: Entity Relationship Diagram of the SWERB_LOCS View
### 3.46 The SWERB UPLOAD view

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

Figure 113: 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 114: Entity Relationship Diagram of the SWERB UPLOAD View
3.47 The TESTES_ARC_STATS View

SELECT testesdartids.dartid AS dartid
, testesllength.testllengthsamps AS testllengthsamps
, testesllength.testllength_mean AS testllength_mean
, testesllength.testllength_stddev AS testllength_stddev
, testeslwidth.testlwidthsamps AS testlwidthsamps
, testeslwidth.testlwidth_mean AS testlwidth_mean
, testeslwidth.testlwidth_stddev AS testlwidth_stddev
, testesrlength.testrlengthsamps AS testrlengthsamps
, testesrlength.testrlength_mean AS testrlength_mean
, testesrlength.testrlength_stddev AS testrlength_stddev
, testesrwidth.testrwidthsamps AS testrwidthsamps
, testesrwidth.testrwidth_mean AS testrwidth_mean
, testesrwidth.testrwidth_stddev AS testrwidth_stddev
FROM (SELECT testes_arc.dartid
 FROM testes_arc
 GROUP BY testes_arc.dartid)
 AS testesdartids
LEFT OUTER JOIN
 (SELECT testes_arc.dartid AS llengthdartid
 , count(*) AS testllengthsamps
 , avg(testes_arc.testlength) AS testllength_mean
 , stddev(testes_arc.testlength) AS testllength_stddev
 FROM testes_arc
 WHERE testes_arc.testside = 'L'
 AND testes_arc.testlength IS NOT NULL
 GROUP BY testes_arc.dartid)
 AS testesllength
ON testesllength.llengthdartid = testesdartids.dartid
LEFT OUTER JOIN
 (SELECT testes_arc.dartid AS lwidthdartid
 , count(*) AS testlwidthsamps
 , avg(testes_arc.testwidth) AS testlwidth_mean
 , stddev(testes_arc.testwidth) AS testlwidth_stddev
 FROM testes_arc
 WHERE testes_arc.testside = 'L'
 AND testes_arc.testwidth IS NOT NULL
 GROUP BY testes_arc.dartid)
 AS testeslwidth
ON testeslwidth.lwidthdartid = testesdartids.dartid
LEFT OUTER JOIN
 (SELECT testes_arc.dartid AS rlengthdartid
 , count(*) AS testrlengthsamps
 , avg(testes_arc.testlength) AS testrlength_mean
 , stddev(testes_arc.testlength) AS testrlength_stddev
 FROM testes_arc
 WHERE testes_arc.testside = 'R'
 AND testes_arc.testlength IS NOT NULL
 GROUP BY testes_arc.dartid)
 AS testesrlength
ON testesrlength.rlengthdartid = testesdartids.dartid
LEFT OUTER JOIN
 (SELECT testes_arc.dartid AS rwidthdartid
 , count(*) AS testrwidthsamps
 , avg(testes_arc.testwidth) AS testrwidth_mean
 , stddev(testes_arc.testwidth) AS testrwidth_stddev
 FROM testes_arc
 WHERE testes_arc.testside = 'R'
 AND testes_arc.testwidth IS NOT NULL
 GROUP BY testes_arc.dartid)
 AS testesrwidth
ON testesrwidth.rwidthdartid = testesdartids.dartid;
Figure 116: Entity Relationship Diagram of the TESTES_ARC_STATS View
3.48 The TESTES_DIAM_STATS View

SELECT testesdartids.dartid AS dartid,
    testesllength.testllengthsamps AS testllengthsamps,
    testesllength.testllength_mean AS testllength_mean,
    testesllength.testllength_stddev AS testllength_stddev,
    testesrlength.testrlengthsamps AS testrlengthsamps,
    testesrlength.testrlength_mean AS testrlength_mean,
    testesrlength.testrlength_stddev AS testrlength_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 118: Entity Relationship Diagram of the TESTES_DIAM_STATS View
3.49 The TESTOSTERONES View

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

Figure 119: Query Defining the TESTOSTERONES View
Figure 120: Entity Relationship Diagram of the TESTOSTERONES View

* 'MEOH_EXT' and 'SPE' are aliases of 'HORMONE_PREP_DATA' in which the Procedure column is 'MEOH_EXT' and 'SPE', respectively. They do not exist anywhere as independent entities.*
3.50 The THYROID_HORMONES View

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

Figure 121: Query Defining the THYROID_HORMONES View
Figure 122: Entity Relationship Diagram of the THYROID_HORMONES View
### 3.51 The TISSUES View

```sql
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.collection_date_status AS collection_date_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 123: Query Defining the TISSUES View
Figure 124: Entity Relationship Diagram of the TISSUES View
3.52 The TISSUES_HORMONES 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.collection_date_status AS collection_date_status
    , tissue_data.notes AS notes
    , hormone_sample_data.hsid AS hsid
    , hormone_sample_data.fzdried_date AS fzdried_date
    , hormone_sample_data.sifted_date AS sifted_date
    , hormone_sample_data.avail_mass_g AS avail_mass_g
    , hormone_sample_data.avail_date AS avail_date
    , hormone_sample_data.comments AS comments
FROM tissue_data
JOIN locations
    ON locations.locid = tissue_data.locid
JOIN unique_indivs
    ON unique_indivs.uiid = tissue_data.uiid
LEFT JOIN biograph
    ON biograph.bioid::text = unique_indivs.individ
    AND unique_indivs.popid = 1
LEFT JOIN tissue_local_ids AS local_1
    ON local_1.tid = tissue_data.tid
    AND local_1.institution = 1
LEFT JOIN tissue_local_ids AS local_2
    ON local_2.tid = tissue_data.tid
    AND local_2.institution = 2
LEFT JOIN hormone_sample_data
    ON hormone_sample_data.tid = tissue_data.tid;

Figure 125: Query Defining the TISSUES_HORMONES View
3.53 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 126: Entity Relationship Diagram of the TISSUES_HORMONES View

Figure 127: Query Defining the ULNA_STATS View
### 3.54 The VAGINAL_PH_STATS View

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

Figure 128: Entity Relationship Diagram of the ULNA_STATS View

Figure 129: Query Defining the VAGINAL_PH_STATS View
Figure 130: Entity Relationship Diagram of the VAGINAL_PH_STATS View
3.55 The WOUNDPATHOLOGIES View

WITH concat_observers AS (SELECT wprid,
    string_agg(observer, '/' ORDER BY wpoid) as observers
    FROM wp_observers
    GROUP BY wprid)

SELECT wp_reports.wprid AS wprid,
    wp_reports.wid AS wid,
    wp_reports.date AS reportdate,
    wp_reports.time AS reporttime,
    concat_observers.observers AS observers,
    wp_reports.sname AS sname,
    wp_reports.grp AS grp,
    wp_reports.observercomments AS observercomments,
    wp_reports.reportstate AS reportstate,
    wp_details.wpdid AS wpdid,
    wp_details.woundpathcode AS woundpathcode,
    wp_details.cluster AS cluster,
    wp_details.maxdimension AS maxdimension,
    wp_details.impairslocomotion AS impairslocomotion,
    wp_details.infectionsigns AS infectionsigns,
    wp_details.notes AS detailnotes,
    wp_affectedparts.wpaid AS wpaid,
    wp_affectedparts.bodypart AS bodypart,
    bodyparts.bodyside AS bodyside,
    bodyparts.innerouter AS innerouter,
    bodyparts.bodyregion AS bodyregion,
    wp_affectedparts.quantity_affecting_part AS quantity_affecting_part
FROM wp_reports
LEFT JOIN concat_observers
    ON concat_observers.wprid = wp_reports.wprid
LEFT JOIN wp_details
    ON wp_details.wprid = wp_reports.wprid
LEFT JOIN wp_affectedparts
    ON wp_affectedparts.wpdid = wp_details.wpdid
LEFT JOIN bodyparts
    ON bodyparts.bpid = wp_affectedparts.bodypart;

Figure 131: Query Defining the WOUNDPATHOLOGIES View
Figure 132: Entity Relationship Diagram of the WOUNDSPATHOLOGIES View
### 3.56 The WPDETAILS_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,
      wp_details.maxdimension AS maxdimension,
      wp_details.impairslocomotion AS impairslocomotion,
      wp_details.infectionsigns AS infectionsigns,
      wp_details.notes AS detailnotes,
      wp_affectedparts.wpaid AS wpaid,
      wp_affectedparts.wpdid AS bodypart_wpdid,
      wp_affectedparts.bodypart AS bodypart,
      bodyparts.bodyside AS bodyside,
      bodyparts.innerouter AS innerouter,
      bodyparts.bodyregion AS bodyregion,
      wp_affectedparts.quantity_affecting_part AS quantity_affecting_part
FROM wp_reports
JOIN wp_details
ON wp_details.wprid = wp_reports.wprid
LEFT JOIN wp_affectedparts
ON wp_affectedparts.wpdid = wp_details.wpdid
LEFT JOIN bodyparts
ON bodyparts.bpid = wp_affectedparts.bodypart;
```

Figure 133: Query Defining the WPDETAILS_AFFECTEDPARTS View

Figure 134: Entity Relationship Diagram of the WPDETAILS_AFFECTEDPARTS View
3.57 The WP_HEALS View

WITH concat_observers AS (SELECT wprid,
                   string_agg(observer, '/') ORDER BY wpoid) as observers
FROM wp_observers
GROUP BY wprid)

SELECT wp_reports.wprid AS wprid,
       wp_reports.wid AS wid,
       wp_reports.date AS reportdate,
       wp_reports.time AS reporttime,
       concat_observers.observers AS observers,
       wp_reports.sname AS sname,
       wp_reports.grp AS grp,
       wp_reports.observercomments AS observercomments,
       wp_reports.reportstate AS reportstate,
       wp_details.wpdid AS wpdid,
       wp_details.woundpathcode AS woundpathcode,
       wp_details.cluster AS cluster,
       wp_details.maxdimension AS maxdimension,
       wp_details.impairslocomotion AS impairslocomotion,
       wp_details.infectionsigns AS infectionsigns,
       wp_details.notes AS detailnotes,
       wp_affectedparts.wpaid AS wpaid,
       wp_affectedparts.bodypart AS bodypart,
       bodyparts.bodyside AS bodyside,
       bodyparts.innerouter AS innerouter,
       bodyparts.bodyregion AS bodyregion,
       wp_affectedparts.quantity_affecting_part AS quantity_affecting_part,
       wp_healupdates.wphid AS wphid,
       wp_healupdates.date AS healdate,
       wp_healupdates.healstatus AS healstatus,
       wp_healupdates.notes AS healnotes
FROM wp_healupdates
LEFT JOIN wp_affectedparts
  ON wp_affectedparts.wpaid = wp_healupdates.wpaid
LEFT JOIN bodyparts
  ON bodyparts.bpid = wp_affectedparts.bodypart
LEFT JOIN wp_details
  ON wp_details.wpdid = COALESCE(wp_affectedparts.wpdid, wp_healupdates.wpdid)
LEFT JOIN wp_reports
  ON wp_reports.wprid = COALESCE(wp_details.wprid, wp_healupdates.wprid)
LEFT JOIN concat_observers
  ON concat_observers.wprid = wp_reports.wprid;

Figure 135: Query Defining the WP_HEALS View
Figure 136: Entity Relationship Diagram of the WP_HEALS View, Overall
Figure 137: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology report
Figure 138: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology cluster
Figure 139: Entity Relationship Diagram of the WP_HEALS View for rows with an update to an affected body part
### 3.58 The WP_REPORTS_OBSERVERS View

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

SELECT wp_reports.wprid AS wprid, wp_reports.wid AS wid, wp_reports.date AS date, wp_reports.time AS time, concat_observers.observers AS observers, wp_reports.sname AS sname, wp_reports.grp AS grp, wp_reports.observercomments AS observercomments, wp_reports.reportstate AS reportstate FROM wp_reports INNER JOIN concat_observers ON concat_observers.wprid = wp_reports.wprid;

![Figure 140: Query Defining the WP_REPORTS_OBSERVERS View](image)

**Figure 140: Query Defining the WP_REPORTS_OBSERVERS View**

**Figure 141: 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 142: Query Defining the BIRTH_GRP View

<table>
<thead>
<tr>
<th>BIOGRAPH</th>
<th>Individual in question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
</tr>
<tr>
<td>Birth (Birth)</td>
<td></td>
</tr>
<tr>
<td>Remainder of columns in BIOGRAPH....</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MEMBERS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
</tr>
<tr>
<td>Date (Birth)</td>
<td></td>
</tr>
<tr>
<td>Grp (Grp)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 143: Entity Relationship Diagram of the BIRTH_GRP View

4.2 The ENTRYDATE_GRP View

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

Figure 144: Query Defining the ENTRYDATE_GRP View

<table>
<thead>
<tr>
<th>BIOGRAPH</th>
<th>Individual in question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
</tr>
<tr>
<td>Entrydate (Entrydate)</td>
<td></td>
</tr>
<tr>
<td>Remainder of columns in BIOGRAPH....</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MEMBERS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
</tr>
<tr>
<td>Date (Entrydate)</td>
<td></td>
</tr>
<tr>
<td>Grp (Grp)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 145: Entity Relationship Diagram of the ENTRYDATE_GRP View
4.3 The STATDATE_GRP View

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

Figure 146: 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 147: 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 148: 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 149: 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 150: Query Defining the CYCGAPDAYS_GRP View

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

<table>
<thead>
<tr>
<th>MEMBERS</th>
<th>Date in question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
</tr>
<tr>
<td>Grp (Grp)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 151: 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 152: Query Defining the CYCGAPS_GRP View

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

<table>
<thead>
<tr>
<th>MEMBERS</th>
<th>Date in question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sname (Sname)</td>
<td></td>
</tr>
<tr>
<td>Date (Date)</td>
<td></td>
</tr>
<tr>
<td>Grp (Grp)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 153: 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 154: Query Defining the CYCSTATS_GRP View

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

4.8 The DARTINGS_GRP View

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

Figure 156: Query Defining the DARTINGS_GRP View

![Entity Relationship Diagram of the DARTINGS_GRP View](image)
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 158: 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 159: 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 160: 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 161: 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 162: 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 in MDINTERVALS....</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 163: 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 164: 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 in MMINTERVALS....</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 165: Entity Relationship Diagram of the MMINTERVALS_GRP View
4.13 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 166: Query Defining the RANKDATES_GRP View

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

Figure 167: Entity Relationship Diagram of the RANKDATES_GRP View

4.14 The REPSTATS_GRP View

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

Figure 168: Query Defining the REPSTATS_GRP View

<table>
<thead>
<tr>
<th>REPSTATS</th>
<th>MEMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RId (RId)</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 REPSTATS...</td>
<td></td>
</tr>
</tbody>
</table>

Figure 169: Entity Relationship Diagram of the REPSTATS_GRP View