The Babase Pocket Reference Guide

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

<table>
<thead>
<tr>
<th>ACTION</th>
<th>NAME</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRITTEN BY</td>
<td>Karl O. Pinc, PhD. Jeanne Altmann, PhD. Susan C. Alberts, and Leah Gerber</td>
<td>November 2, 2022</td>
</tr>
<tr>
<td>ER Diagram layout and conversion to Dia</td>
<td>Leah Gerber</td>
<td>November 2, 2022</td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Anne Hubbard</td>
<td>November 2, 2022</td>
</tr>
<tr>
<td>DocBook formatting</td>
<td>Karl Pinc</td>
<td>November 2, 2022</td>
</tr>
</tbody>
</table>

# Revision History

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>DATE</th>
<th>DESCRIPTION</th>
<th>NAME</th>
</tr>
</thead>
</table>

---

The Babase Pocket Reference Guide

iii
Contents

1 Babase Summarized 1

2 The Babase ER Diagrams 1

3 The Babase Views 24
   3.1 The ACTOR_ACTEES View 25
   3.2 The ANESTH_STATS View 26
   3.3 The BODYTEMP_STATS View 27
   3.4 The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views 28
   3.5 The CHEST_STATS View 29
   3.6 The CROWN_RUMP_STATS View 30
   3.7 The CYCLES_SEXSKINS and CYCLES_SEXSKINS_SORTED Views 30
   3.8 The CYCPOINTS_CYCLES and CYCPOINTS_CYCLES_SORTED Views 31
   3.9 The DSAMPLES View 33
   3.10 The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views 34
   3.11 The DENT_CODES View 36
   3.12 The DENT_SITES View 39
   3.13 The INTERACT and INTERACT_SORTED Views 41
   3.14 The LOCATIONS_FREE View 42
   3.15 The MATERNITIES View 43
   3.16 The MIN_MAXS View 45
   3.17 The MIN_MAXS_SORTED View 47
   3.18 The MPI_EVENTS View 50
   3.19 The MTD_CYCLES View 52
   3.20 The NUCACID_CONCS View 54
   3.21 The NUCACIDS View 55
   3.22 The NUCACIDS_W_CONC View 58
   3.23 The PARENTS View 60
   3.24 The PCV_STATS View 61
   3.25 The POINTS and POINTS_SORTED Views 62
   3.26 The POTENTIAL_DADS View 65
   3.27 The PROPORTIONAL_RANKS View 70
   3.28 The QUADS View 71
   3.29 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_SORTED Views 71
   3.30 The SEXSKINS_REPRO_NOTES View 72
   3.31 The SWERB view 74
   3.32 The SWERB_DEPARTS view 76
3.33 The SWERB_LOC_GPS_XY view ................................................................. 76
3.34 The SWERB_LOCS view ................................................................. 77
3.35 The SWERB_UPLOAD view ................................................................. 78
3.36 The TISSUES View ................................................................. 79
3.37 The ULNA_STATS View ................................................................. 80
3.38 The TESTES_ARC_STATS View ................................................................. 83
3.39 The TESTES_DIAM_STATS View ................................................................. 86
3.40 The WP_DETAILS_AFFECTEDPARTS View ................................................................. 88
3.41 The WP_HEALS View ................................................................. 89
3.42 The WP_REPORTS_OBSERVERS View ................................................................. 94

4 Views Which Add Gid To Tables ................................................................. 94
4.1 The BIRTH_GRP View ................................................................. 95
4.2 The ENTRYDATE_GRP View ................................................................. 95
4.3 The STATDATE_GRP View ................................................................. 96
4.4 The CONSORTDATES_GRP View ................................................................. 96
4.5 The CYCGAPDAYS_GRP View ................................................................. 97
4.6 The CYCGAPS_GRP View ................................................................. 97
4.7 The CYCSTATS_GRP View ................................................................. 98
4.8 The DARTINGS_GRP View ................................................................. 98
4.9 The DISPERSEDATES_GRP View ................................................................. 99
4.10 The MATUREDATES_GRP View ................................................................. 99
4.11 The MDINTERVALS_GRP View ................................................................. 100
4.12 The MMINTERVALS_GRP View ................................................................. 100
4.13 The RANKDATES_GRP View ................................................................. 101
4.14 The REPSTATS_GRP View ................................................................. 101
### 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
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 CROWN RUMP_STATS View ......................... 30
32. Entity Relationship Diagram of the CROWN RUMP_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
<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>76</td>
<td>Entity Relationship Diagram of the PROPORTIONAL_RANKS View</td>
</tr>
<tr>
<td>77</td>
<td>Query Defining the QUADS View</td>
</tr>
<tr>
<td>78</td>
<td>Entity Relationship Diagram of the QUADS View</td>
</tr>
<tr>
<td>79</td>
<td>Query Defining the SEXSKINS_CYCLES View</td>
</tr>
<tr>
<td>80</td>
<td>Entity Relationship Diagram of the SEXSKINS_CYCLES View</td>
</tr>
<tr>
<td>81</td>
<td>Query Defining the SEXSKINS_REPRO_NOTES View</td>
</tr>
<tr>
<td>82</td>
<td>Entity Relationship Diagram of the SEXSKINS_REPRO_NOTES View</td>
</tr>
<tr>
<td>83</td>
<td>Query Defining the SWERB View</td>
</tr>
<tr>
<td>84</td>
<td>Entity Relationship Diagram of the SWERB View</td>
</tr>
<tr>
<td>85</td>
<td>Query Defining the SWERB_DEPARTS View</td>
</tr>
<tr>
<td>86</td>
<td>Entity Relationship Diagram of the SWERB_DEPARTS View</td>
</tr>
<tr>
<td>87</td>
<td>Query Defining the SWERB_LOC_GPS_XY View</td>
</tr>
<tr>
<td>88</td>
<td>Entity Relationship Diagram of the SWERB_LOC_GPS_XY View</td>
</tr>
<tr>
<td>89</td>
<td>Query Defining the SWERB_LOCS View</td>
</tr>
<tr>
<td>90</td>
<td>Entity Relationship Diagram of the SWERB_LOCS View</td>
</tr>
<tr>
<td>91</td>
<td>Query Defining the SWERB_UPLOAD View</td>
</tr>
<tr>
<td>92</td>
<td>Entity Relationship Diagram of the SWERB_UPLOAD View</td>
</tr>
<tr>
<td>93</td>
<td>Query Defining the TISSUES View</td>
</tr>
<tr>
<td>94</td>
<td>Entity Relationship Diagram of the TISSUES View</td>
</tr>
<tr>
<td>95</td>
<td>Query Defining the ULNA_STATS View</td>
</tr>
<tr>
<td>96</td>
<td>Entity Relationship Diagram of the ULNA_STATS View</td>
</tr>
<tr>
<td>97</td>
<td>Query Defining the TESTES_ARC_STATS View</td>
</tr>
<tr>
<td>98</td>
<td>Entity Relationship Diagram of the TESTES_ARC_STATS View</td>
</tr>
<tr>
<td>99</td>
<td>Query Defining the TESTES_DIAM_STATS View</td>
</tr>
<tr>
<td>100</td>
<td>Entity Relationship Diagram of the TESTES_DIAM_STATS View</td>
</tr>
<tr>
<td>101</td>
<td>Query Defining the WP_DETAILS_AFFECTEDPARTS View</td>
</tr>
<tr>
<td>102</td>
<td>Entity Relationship Diagram of the WP_DETAILS_AFFECTEDPARTS View</td>
</tr>
<tr>
<td>103</td>
<td>Query Defining the WP_HEALS View</td>
</tr>
<tr>
<td>104</td>
<td>Entity Relationship Diagram of the WP_HEALS View, Overall</td>
</tr>
<tr>
<td>105</td>
<td>Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology report</td>
</tr>
<tr>
<td>106</td>
<td>Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology cluster</td>
</tr>
<tr>
<td>107</td>
<td>Entity Relationship Diagram of the WP_HEALS View for rows with an update to an affected body part</td>
</tr>
<tr>
<td>108</td>
<td>Query Defining the WP_REPORTS_OBSERVERS View</td>
</tr>
<tr>
<td>109</td>
<td>Entity Relationship Diagram of the WP_REPORTS_OBSERVERS View</td>
</tr>
<tr>
<td>110</td>
<td>Query Defining the BIRTH_GRP View</td>
</tr>
<tr>
<td>111</td>
<td>Entity Relationship Diagram of the BIRTH_GRP View</td>
</tr>
<tr>
<td>112</td>
<td>Query Defining the ENTRYDATE_GRP View</td>
</tr>
<tr>
<td>113</td>
<td>Entity Relationship Diagram of the ENTRYDATE_GRP View</td>
</tr>
<tr>
<td>114</td>
<td>Query Defining the STATDATE_GRP View</td>
</tr>
</tbody>
</table>
List of Tables

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

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

2 The Babase ER Diagrams

The BABASE Database

---

Figure 1: Key to the Babase Entity Relationship Diagrams

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

**Group Membership and Life Events**

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

**Physical Traits**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
</table>
| WP_AFFECTEDPARTS in Babase: | One row for each  
body part affected by a specific wound/pathology |
| WPDETAILS in Babase: | wound or pathology cluster indicated on a report |
| WP_HEALUPDATES in Babase: | update on progress of wound/pathology healing |
| WPREPORTS in Babase: | wound/pathology report |

**Analyzed: Physical Traits**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
</table>
| HORMONE_KITS in Babase: | One row for each  
kit or protocol used to assay hormone concentration |
| HORMONE_PREP_DATA in Babase: | laboratory preparation performed on a sample in the specified series |
| HORMONE_PREP_SERIES in Babase: | series of preparations and assays performed on a sample |
| HORMONE_RESULT_DATA in Babase: | assay for hormone concentration in a sample |
| HORMONE_SAMPLE_DATA in Babase: | tissue sample used in hormone analysis |
| HYBRIDGENE_ANALYSES in Babase: | analysis of genetic hybrid scores |
| HYBRIDGENE_SCORES in Babase: | genetic hybrid score for an individual from an analysis |

**Sexual Cycles**

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

<table>
<thead>
<tr>
<th>INTEGRITY_QUERIES in Babase:</th>
<th>One row for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>query used to discover data integrity problems</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INTEGRITY_WARNINGS in Babase:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>data integrity problem discovered by the warning sub-system</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 2: Babase Group Membership Entity Relationship Diagram**
<table>
<thead>
<tr>
<th>Table</th>
<th>Id Column</th>
<th>Related Column(s)</th>
<th>One entry for every possible choice of...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BODYPARTS in Babase:</strong></td>
<td>Bodypart</td>
<td>TICKS in Babase::Bodypart in Babase::BODYPARTS in Babase::Bodyregion in Babase::WP_AFFECTEDPARTS in Babase::Bodypart in Babase::</td>
<td>part of the body</td>
</tr>
<tr>
<td><strong>LAB_PERSONNEL in Babase:</strong></td>
<td>Initials</td>
<td>HYBRIDGENE_ANALYSES in Babase::Analyzed_By in Babase::NUCACID_CREATORS in Babase::Creator in Babase::WBC_COUNTS in Babase::Counted_By in Babase::</td>
<td>person who generates data, usually in a lab setting</td>
</tr>
<tr>
<td><strong>OBSERVERS in Babase:</strong></td>
<td>Initials</td>
<td>SAMPLES in Babase::Observer in Babase::WRREADINGS in Babase::WRperson in Babase::RGSETPS in Babase::RGSPerson in Babase::CROWNMRUPS in Babase::CFObserved in Babase::CHESTS in Babase::Chobserver in Babase::ULNAS in Babase::Uloobserver in Babase::HUMERUSES in Babase::Huobserver in Babase::SWERB_OBSERVERS in Babase::Observer in Babase::</td>
<td>person who record observational data</td>
</tr>
<tr>
<td><strong>OBSERVER_ROLES in Babase:</strong></td>
<td>Initials</td>
<td>OBSERVERS in Babase::Role in Babase::SWERB_Observed_Role in Babase::SWERB_OBSERVERS in Babase::</td>
<td>Role way in which a person can be involved in the data collection process</td>
</tr>
<tr>
<td><strong>UNKSNAMES in Babase:</strong></td>
<td>Unksname</td>
<td>NEIGHBORS in Babase::Unksname in Babase::</td>
<td>problem in identifying neighbor of focal during point sampling or in identifying a lone male in a SWERB other group observation</td>
</tr>
</tbody>
</table>

**Group Membership and Life Events**

<table>
<thead>
<tr>
<th>Table</th>
<th>Id Column</th>
<th>Related Column(s)</th>
<th>One entry for every possible choice of...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BSTATUSES in Babase:</strong></td>
<td>Bstatus</td>
<td>BIOGRAPH in Babase::Bstatus in Babase::</td>
<td>birthday estimation accuracy</td>
</tr>
<tr>
<td><strong>CONFIDENCES in Babase:</strong></td>
<td>Confidence</td>
<td>BIOGRAPH in Babase::DcauseNatureConfidence in Babase:: BIOGRAPH in Babase::DcauseAgentConfidence in Babase::</td>
<td>degree of certitude in nature of death, agent of death, disperse date assignment, or maternal group assignment</td>
</tr>
</tbody>
</table>


The Babase Pocket Reference Guide

### IQTYPES in Babase:
- **Id Column**: IQType
- **Related Column(s)**: INTEGRITY_QUERIES in Babase
- **One entry for every possible choice of...**: kind of problem with data integrity

### WARNING_REMARKS in Babase:
- **Id Column**: WRID
- **Related Column(s)**: INTEGRITY_WARNINGS in Babase
- **One entry for every possible choice of...**: remark which might apply to more than one instance of questionable database integrity

### Table 4: The Warning Sub-System Support Tables

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALTERNATE_SNAMES</strong></td>
<td>Individuals known by more than one Sname</td>
</tr>
<tr>
<td><strong>RANKS</strong></td>
<td>Individual holding rank</td>
</tr>
<tr>
<td><strong>GROUPS</strong></td>
<td>Group in which rank is held</td>
</tr>
<tr>
<td><strong>GROUPEDS</strong></td>
<td>Immediate ancestor of group</td>
</tr>
<tr>
<td><strong>GROUPEDF</strong></td>
<td>Immediate group into which this group was created</td>
</tr>
<tr>
<td><strong>DAD_DATA</strong></td>
<td>Individual's birth data</td>
</tr>
<tr>
<td><strong>CONSORTDATES</strong></td>
<td>Individual's relationship data</td>
</tr>
</tbody>
</table>

### Figures
- **Figure 3**: Babase Life Events Entity Relationship Diagram
- **Figure 4**: Babase Group Data Entity Relationship Diagram
<table>
<thead>
<tr>
<th>View</th>
<th>One row for each</th>
<th>Purpose</th>
<th>Tables/Views used</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENSUS_DEMOG in Babase:</td>
<td>CENSUS in Babase: row</td>
<td>Maintenance of CENSUS in Babase: rows that are extended with DEMOG in Babase: information.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>CENSUS_DEMOG_SORTED in Babase:</td>
<td>CENSUS in Babase: row</td>
<td>Maintenance of CENSUS_DEMOG in Babase: rows in a pre-sorted fashion.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>CYCPOINTS_CYCLES in Babase:</td>
<td>CYCPOINTS in Babase: row</td>
<td>Maintenance of CYCPOINTS in Babase: rows that are extended with CYCLES in Babase: information.</td>
<td>CYCLES in Babase:, CYCPOINTS in Babase:</td>
</tr>
<tr>
<td>CYCPOINTS_CYCLES_SORTED in Babase:</td>
<td>CYCPOINTS in Babase: row</td>
<td>The CYCPOINTS_CYCLES in Babase: view sorted by CYCLES in Babase::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>DEMOG in Babase: row</td>
<td>Maintenance of DEMOG in Babase: rows.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>DEMOG_CENSUS_SORTED in Babase:</td>
<td>CENSUS in Babase: row</td>
<td>Maintenance of DEMOG_CENSUS in Babase: rows in a pre-sorted fashion.</td>
<td>CENSUS in Babase:, DEMOG in Babase:</td>
</tr>
<tr>
<td>GROUPS_HISTORY in Babase:</td>
<td>GROUPS in Babase: row</td>
<td>Depiction of GROUPS in Babase: rows in a more human-readable format.</td>
<td>GROUPS in Babase:</td>
</tr>
<tr>
<td>PARENTS in Babase:</td>
<td>BIOGRAPH in Babase: row for which there is either a row in MATERNITIES in Babase: with a record of the individual’s mother or there is a row in DAD_DATA in Babase: with a record of the individual’s father -- with a non-NULL Dad_consensus in Babase::</td>
<td>Easy access to parental information.</td>
<td>BIOGRAPH in Babase:, MATERNITIES in Babase:, DAD_DATA in Babase:, MEMBERS in Babase:</td>
</tr>
<tr>
<td>POTENTIAL_DADS in Babase:</td>
<td></td>
<td>(completed) female reproductive event for every male more than 2192 days old (approximately 6 years) present in the mother’s group during her fertile period</td>
<td>MATERNITIES in Babase:, MEMBERS in Babase: (multiple times), ACTOR_ACTEES in Babase: (multiple times), BIOGRAPH in Babase:, RANK-DATES in Babase:, MATURE-DATES in Babase:</td>
</tr>
<tr>
<td>PROPORTIONAL_RANKS in Babase:</td>
<td>RANKS in Babase: row</td>
<td>Automatic calculation of proportional ranks from the ordinal ranks in RANKS in Babase:.</td>
<td>RANKS in Babase:</td>
</tr>
</tbody>
</table>

**Physical Traits**

<table>
<thead>
<tr>
<th>View</th>
<th>One row for each</th>
<th>Purpose</th>
<th>Tables/Views used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Group Membership and Life Events**

- **CENSUS_DEMOG in Babase:**
  - One row for each
  - Purpose: Maintenance of CENSUS in Babase: rows that are extended with DEMOG in Babase: information.
  - Tables/Views used: CENSUS in Babase:, DEMOG in Babase:

- **CENSUS_DEMOG_SORTED in Babase:**
  - One row for each
  - Purpose: Maintenance of CENSUS_DEMOG in Babase: rows in a pre-sorted fashion.
  - Tables/Views used: CENSUS in Babase:, DEMOG in Babase:

- **CYCPOINTS_CYCLES in Babase:**
  - One row for each
  - Purpose: Maintenance of CYCPOINTS in Babase: rows that are extended with CYCLES in Babase: information.
  - Tables/Views used: CYCLES in Babase:, CYCPOINTS in Babase:

- **CYCPOINTS_CYCLES_SORTED in Babase:**
  - One row for each
  - Purpose: The CYCPOINTS_CYCLES in Babase: view sorted by CYCLES in Babase::Sname in Babase:, by CYCPOINTS in Babase::Date in Babase:.
  - Tables/Views used: CYCLES in Babase:, CYCPOINTS in Babase:

- **DEMOG_CENSUS in Babase:**
  - One row for each
  - Purpose: Maintenance of DEMOG in Babase: rows.
  - Tables/Views used: CENSUS in Babase:, DEMOG in Babase:

- **DEMOG_CENSUS_SORTED in Babase:**
  - One row for each
  - Purpose: Maintenance of DEMOG_CENSUS in Babase: rows in a pre-sorted fashion.
  - Tables/Views used: CENSUS in Babase:, DEMOG in Babase:

- **GROUPS_HISTORY in Babase:**
  - One row for each
  - Purpose: Depiction of GROUPS in Babase: rows in a more human-readable format.
  - Tables/Views used: GROUPS in Babase:

- **PARENTS in Babase:**
  - One row for each
  - Purpose: Easy access to parental information.
  - Tables/Views used: BIOGRAPH in Babase:, MATERNITIES in Babase:, DAD_DATA in Babase:, MEMBERS in Babase:

- **POTENTIAL_DADS in Babase:**
  - One row for each
  - Purpose: Research into paternity, especially the selection of potential fathers for further genetic testing.
  - Tables/Views used: MATERNITIES in Babase:, MEMBERS in Babase: (multiple times), ACTOR_ACTEES in Babase: (multiple times), BIOGRAPH in Babase:, RANK-DATES in Babase:, MATURE-DATES in Babase:

- **PROPORTIONAL_RANKS in Babase:**
  - One row for each
  - Purpose: Automatic calculation of proportional ranks from the ordinal ranks in RANKS in Babase:.
  - Tables/Views used: RANKS in Babase:
<table>
<thead>
<tr>
<th>Table</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>BIRTH_GRP in Babase:</td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>ENTRYDATE_GRP in Babase:</td>
</tr>
<tr>
<td>BIOGRAPH in Babase:</td>
<td>STATDATE_GRP in Babase:</td>
</tr>
<tr>
<td>CONSORTDATES in Babase:</td>
<td>CONSORTDATES_GRP in Babase:</td>
</tr>
<tr>
<td>CYCGAPDAYS in Babase:</td>
<td>CYCGAPDAYS_GRP in Babase:</td>
</tr>
<tr>
<td>CYCGAPS in Babase:</td>
<td>CYCGAPS_GRP in Babase:</td>
</tr>
<tr>
<td>CYCSTATS in Babase:</td>
<td>CYCSTATS_GRP in Babase:</td>
</tr>
<tr>
<td>DARTINGS in Babase:</td>
<td>DARTINGS_GRP in Babase:</td>
</tr>
<tr>
<td>DISPERSEDATES in Babase:</td>
<td>DISPERSEDATES_GRP in Babase:</td>
</tr>
<tr>
<td>MATUREDATES in Babase:</td>
<td>MATUREDATES_GRP in Babase:</td>
</tr>
<tr>
<td>MDINTERVALS in Babase:</td>
<td>MDINTERVALS_GRP in Babase:</td>
</tr>
<tr>
<td>MMINTERVALS in Babase:</td>
<td>MMINTERVALS_GRP in Babase:</td>
</tr>
<tr>
<td>RANKDATES in Babase:</td>
<td>RANKDATES_GRP in Babase:</td>
</tr>
<tr>
<td>REPSTATS in Babase:</td>
<td>REPSTATS_GRP in Babase:</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

[Diagram showing entity relationship between darting individual and dartings, with relationships to attributes such as packed cell volume, body temperature, and physiological measurements.]
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 WeatherHawk Data Entity Relationship Diagram
3 The Babase Views

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

SELECT interact_data.iid AS iid
 , interact_data.sid AS sid
 , interact_data.act AS act
 , interact_data.date AS date
 , interact_data.start AS start
 , interact_data.stop AS stop
 , interact_data.observer AS observer
 , actor.partid AS actorid
 , COALESCE(actor.sname, '998'::CHAR(3)) AS actor
 , (SELECT actorms.grp
      FROM members AS actorms
      WHERE actorms.sname = actor.sname
      AND actorms.date = interact_data.date) AS actor_grp
 , actee.partid AS acteeid
 , COALESCE(actee.sname, '998'::CHAR(3)) AS actee
 , (SELECT acteems.grp
      FROM members AS acteems
      WHERE acteems.sname = actee.sname
      AND acteems.date = interact_data.date) AS actee_grp
 , interact_data.handwritten AS handwritten
 , 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 22: Entity Relationship Diagram of the ACTOR_ACTEES View

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;
3.4 The CENSUS_DEMOG and CENSUS_DEMOG_SORTED Views

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

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

Figure 29: Query Defining the CHEST_STATS View

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

Figure 30: Entity Relationship Diagram of the CHEST_STATS View
3.6 The CROWNRPUMP_STATS View

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

Figure 31: Query Defining the CROWNRUMP_STATS View

3.7 The CYCLES_SEXSKINS and CYCLES_SEXSKINS_SORTED Views

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

Figure 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_lmol1mol2s.num AS lm1m2siliconemold,
    skin_punchs.num AS skinpunch,
    tc_skins.num AS tcskin,
    vag_swabs.num AS vaginalswhab,
    cerv_swabs.num AS cervicalswab,
    fecal_formalin.num AS fecal_formalin,
    vaginal_ph.num AS vaginal_ph,
    palm_swab.num AS palm_swab,
    tongue_swab.num AS tongue_swab,
    tooth_plaque_swab.num AS tooth_plaque_swab,
    vagswab_microbiome.num AS vagswab_microbiome,
    glans_penis_swab.num AS glans_penis_swab,
    fecal_microbiome.num AS fecal_microbiome,
    nostrils_swab.num AS nostrils_swab,
    skin_behind_ear_swab.num AS skin_behind_ear_swab,
    skin_inside_elbow_swab.num AS skin_inside_elbow_swab
FROM dartings
JOIN members
    ON dartings.sname = members.sname
    AND dartings.date = members.date
LEFT JOIN dart_samples blood_unspecs
    ON dartings.dartid = blood_unspecs.dartid
    AND blood_unspecs.ds_type = 1
LEFT JOIN dart_samples blood_paxgenes
    ON dartings.dartid = blood_paxgenes.dartid
    AND blood_paxgenes.ds_type = 2
LEFT JOIN dart_samples blood_purpletops
    ON dartings.dartid = blood_purpletops.dartid
    AND blood_purpletops.ds_type = 3
LEFT JOIN dart_samples blood_separators
    ON dartings.dartid = blood_separators.dartid
    AND blood_separators.ds_type = 4
LEFT JOIN dart_samples blood_cpts
    ON dartings.dartid = blood_cpts.dartid
    AND blood_cpts.ds_type = 5
LEFT JOIN dart_samples blood_trucultures
    ON dartings.dartid = blood_trucultures.dartid
    AND blood_trucultures.ds_type = 6
LEFT JOIN dart_samples blood_smears
    ON dartings.dartid = blood_smears.dartid
    AND blood_smears.ds_type = 7
LEFT JOIN dart_samples hair_unspecs
    ON dartings.dartid = hair_unspecs.dartid

3.10 The DEMOG_CENSUS and DEMOG_CENSUS_SORTED Views

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

Figure 38: Query Defining the DEMOG_CENSUS View

Figure 39: Entity Relationship Diagram of the DEMOG_CENSUS View
3.11 The DENT_CODES View

SELECT teeth.dartid AS dartid
    , rum3.rum3tstate AS rum3tstate
    , rum3.rum3tcondition AS rum3tcondition
    , rum2.rum2tstate AS rum2tstate
    , rum2.rum2tcondition AS rum2tcondition
    , rum1.rum1tstate AS rum1tstate
    , rum1.rum1tcondition AS rum1tcondition
    , rup2.rup2tstate AS rup2tstate
    , rup2.rup2tcondition AS rup2tcondition
    , rup1.rup1tstate AS rup1tstate
    , rup1.rup1tcondition AS rup1tcondition
    , rup1.rup1tstate AS rup1tstate
    , ruc.ructstate AS ructstate
    , ruc.ructcondition AS ructcondition
    , rui2.rui2tstate AS rui2tstate
    , rui2.rui2tcondition AS rui2tcondition
    , rui1.rui1tstate AS rui1tstate
    , rui1.rui1tcondition AS rui1tcondition
    , lli1.lli1tstate AS lli1tstate
    , lli1.lli1tcondition AS lli1tcondition
    , lli2.lli2tstate AS lli2tstate
    , lli2.lli2tcondition AS lli2tcondition
    , llm1.llm1tstate AS llm1tstate
    , llm1.llm1tcondition AS llm1tcondition
    , llm2.llm2tstate AS llm2tstate
    , llm2.llm2tcondition AS llm2tcondition
    , llm3.llm3tstate AS llm3tstate
    , llm3.llm3tcondition AS llm3tcondition
    , llp1.llp1tstate AS llp1tstate
    , llp1.llp1tcondition AS llp1tcondition
    , llp1.llp1tstate AS llp1tstate
    , llc.llctstate AS llctstate
    , llc.llctcondition AS llctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp1.rlp1tcondition AS rlp1tcondition
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rup1.rup1tstate AS rup1tstate
    , rup2.rup2tstate AS rup2tstate
    , rup2.rup2tcondition AS rup2tcondition
    , rup1.rup1tstate AS rup1tstate
    , ruc.ructstate AS ructstate
    , ruc.ructcondition AS ructcondition
    , rui2.rui2tstate AS rui2tstate
    , rui2.rui2tcondition AS rui2tcondition
    , rui1.rui1tstate AS rui1tstate
    , rui1.rui1tcondition AS rui1tcondition
    , lli1.lli1tstate AS lli1tstate
    , lli1.lli1tcondition AS lli1tcondition
    , lli2.lli2tstate AS lli2tstate
    , lli2.lli2tcondition AS lli2tcondition
    , llm1.llm1tstate AS llm1tstate
    , llm1.llm1tcondition AS llm1tcondition
    , llm2.llm2tstate AS llm2tstate
    , llm2.llm2tcondition AS llm2tcondition
    , llm3.llm3tstate AS llm3tstate
    , llm3.llm3tcondition AS llm3tcondition
FROM teeth
    , 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
    , rup1.rup1tstate AS rup1tstate
    , ruc.ructstate AS ructstate
    , ruc.ructcondition AS ructcondition
    , rui2.rui2tstate AS rui2tstate
    , rui2.rui2tcondition AS rui2tcondition
    , rui1.rui1tstate AS rui1tstate
    , rui1.rui1tcondition AS rui1tcondition
    , lli1.lli1tstate AS lli1tstate
    , lli1.lli1tcondition AS lli1tcondition
    , lli2.lli2tstate AS lli2tstate
    , lli2.lli2tcondition AS lli2tcondition
    , llm1.llm1tstate AS llm1tstate
    , llm1.llm1tcondition AS llm1tcondition
    , llm2.llm2tstate AS llm2tstate
    , llm2.llm2tcondition AS llm2tcondition
    , llm3.llm3tstate AS llm3tstate
    , llm3.llm3tcondition AS llm3tcondition
    , llp1.llp1tstate AS llp1tstate
    , llp1.llp1tcondition AS llp1tcondition
    , llp1.llp1tstate AS llp1tstate
    , llc.llctstate AS llctstate
    , llc.llctcondition AS llctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp1.rlp1tcondition AS rlp1tcondition
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
FROM teeth
    , 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
    , rup1.rup1tstate AS rup1tstate
    , ruc.ructstate AS ructstate
    , ruc.ructcondition AS ructcondition
    , rui2.rui2tstate AS rui2tstate
    , rui2.rui2tcondition AS rui2tcondition
    , rui1.rui1tstate AS rui1tstate
    , rui1.rui1tcondition AS rui1tcondition
    , lli1.lli1tstate AS lli1tstate
    , lli1.lli1tcondition AS lli1tcondition
    , lli2.lli2tstate AS lli2tstate
    , lli2.lli2tcondition AS lli2tcondition
    , llm1.llm1tstate AS llm1tstate
    , llm1.llm1tcondition AS llm1tcondition
    , llm2.llm2tstate AS llm2tstate
    , llm2.llm2tcondition AS llm2tcondition
    , llm3.llm3tstate AS llm3tstate
    , llm3.llm3tcondition AS llm3tcondition
    , llp1.llp1tstate AS llp1tstate
    , llp1.llp1tcondition AS llp1tcondition
    , llp1.llp1tstate AS llp1tstate
    , llc.llctstate AS llctstate
    , llc.llctcondition AS llctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp1.rlp1tcondition AS rlp1tcondition
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlp2.rlp2tstate AS rlp2tstate
    , rlp2.rlp2tcondition AS rlp2tcondition
    , rlp1.rlp1tstate AS rlp1tstate
    , rlc.rltctstate AS rltctstate
    , rlc.rltctcondition AS rltctcondition
FROM teeth
    , rum3.rum3tstate AS rum3tstate
    , rum3.rum3tcondition AS rum3tcondition
Figure 41: Entity Relationship Diagram of the DENT_CODES View
3.12 The DENT_SITES View

```
SELECT teethdartids.dartid AS dartid
    , s1.s1ltstate AS s1ltstate
    , s1.s1tcondition AS s1tcondition
    , s1.s1deciduous AS s1deciduous
    , s2.s2ltstate AS s2ltstate
    , s2.s2tcondition AS s2tcondition
    , s2.s2deciduous AS s2deciduous
    , s3.s3ltstate AS s3ltstate
    , s3.s3tcondition AS s3tcondition
    , s3.s3deciduous AS s3deciduous
    , s4.s4ltstate AS s4ltstate
    , s4.s4tcondition AS s4tcondition
    , s4.s4deciduous AS s4deciduous
    , s5.s5ltstate AS s5ltstate
    , s5.s5tcondition AS s5tcondition
    , s5.s5deciduous AS s5deciduous
    , s6.s6ltstate AS s6ltstate
    , s6.s6tcondition AS s6tcondition
    , s6.s6deciduous AS s6deciduous
    , s7.s7ltstate AS s7ltstate
    , s7.s7tcondition AS s7tcondition
    , s7.s7deciduous AS s7deciduous
    , s8.s8ltstate AS s8ltstate
    , s8.s8tcondition AS s8tcondition
    , s8.s8deciduous AS s8deciduous
    , s9.s9ltstate AS s9ltstate
    , s9.s9tcondition AS s9tcondition
    , s9.s9deciduous AS s9deciduous
    , s10.s10ltstate AS s10ltstate
    , s10.s10tcondition AS s10tcondition
    , s10.s10deciduous AS s10deciduous
    , s11.s11ltstate AS s11ltstate
    , s11.s11tcondition AS s11tcondition
    , s11.s11deciduous AS s11deciduous
    , s12.s12ltstate AS s12ltstate
    , s12.s12tcondition AS s12tcondition
    , s12.s12deciduous AS s12deciduous
    , s13.s13ltstate AS s13ltstate
    , s13.s13tcondition AS s13tcondition
    , s13.s13deciduous AS s13deciduous
    , s14.s14ltstate AS s14ltstate
    , s14.s14tcondition AS s14tcondition
    , s14.s14deciduous AS s14deciduous
    , s15.s15ltstate AS s15ltstate
    , s15.s15tcondition AS s15tcondition
    , s15.s15deciduous AS s15deciduous
    , s16.s16ltstate AS s16ltstate
    , s16.s16tcondition AS s16tcondition
    , s16.s16deciduous AS s16deciduous
    , s17.s17ltstate AS s17ltstate
    , s17.s17tcondition AS s17tcondition
    , s17.s17deciduous AS s17deciduous
    , s18.s18ltstate AS s18ltstate
    , s18.s18tcondition AS s18tcondition
    , s18.s18deciduous AS s18deciduous
    , s19.s19ltstate AS s19ltstate
    , s19.s19tcondition AS s19tcondition
    , s19.s19deciduous AS s19deciduous
    , s20.s20ltstate AS s20ltstate
    , s20.s20tcondition AS s20tcondition
```
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 INTERACT and INTERACT_SORTED Views

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

Figure 44: Query Defining the INTERACT View

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

Figure 45: Entity Relationship Diagram of the INTERACT View
3.14 The LOCATIONS_FREE View

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

Figure 46: Query Defining the LOCATIONS_FREE View

Figure 47: Entity Relationship Diagram of the LOCATIONS_FREE View
### 3.15 The MATERNITIES View

```sql
SELECT cycles.sname AS mom,
      cycles.cid AS cid,
      cycles.seq AS seq,
      cycles.series AS series,
      cycpoints.cpid AS conceive,
      cycpoints.date AS zdate,
      members.grp AS zdate_grp,
      cycpoints.edate AS edate,
      cycpoints.1date AS ldate,
      cycpoints.source AS source,
      pregs.pid AS pid,
      pregs.parity AS parity,
      biograph.bioid AS child_bioid,
      biograph.sname AS child,
      biograph.birth AS birth
FROM cycles
JOIN cycpoints ON (cycpoints.cid = cycles.cid)
JOIN members ON (members.date = cycpoints.date
                AND members.sname = cycles.sname)
JOIN pregs ON (pregs.conceive = cycpoints.cpid)
JOIN biograph ON (pregs.pid = biograph.pid);
```

Figure 48: Query Defining the MATERNITIES View
Figure 49: Entity Relationship Diagram of the MATERNITIES View
3.16 The MIN_MAXS View

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

Figure 50: Query Defining the MIN_MAXS View
Figure 51: Entity Relationship Diagram of the MIN_MAXS View
3.17 The MIN_MAXS_SORTED View

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

Figure 52: Query Defining the MIN_MAXS_SORTED View
Figure 53: Entity Relationship Diagram of the MIN_MAXS_SORTED View
3.18 The MPI_EVENTS View

SELECT mpis.mpiid AS mpiid
    , mpis.date AS date
    , mpis.context_type AS context_type
    , mpis.context AS context
    , mpi_data.mpiidid AS mpidid
    , mpi_data.seq AS seq
    , mpi_data.mpiact AS mpiact
    , actor.mpipid AS actorid
    , actor.sname AS actor
    , actor.unksname AS unkactor
    , actee.mpipid AS acteeid
    , actee.sname AS actee
    , actee.unksname AS unkactee
    , CASE WHEN EXISTS(SELECT 1
                    FROM mpiacts
                    WHERE mpiacts.mpiact = mpi_data.mpiact
                        AND mpiacts.kind = 'H')
                THEN
                    EXISTS(SELECT 1
                        FROM mpi_data AS request,
                        mpiacts,
                        mpi_parts AS requestor,
                        mpi_parts AS requestee
                        WHERE request.mpiid = mpi_data.mpiid
                            AND request.seq < mpi_data.seq
                            AND mpiacts.mpiact = request.mpiact
                            AND mpiacts.kind = 'R'
                            AND requestor.mpidid = request.mpidid
                            AND requestor.role = 'R'
                            AND requestor.sname = actee.sname
                            AND requestee.mpidid = request.mpidid
                            AND requestee.role = 'E'
                            AND requestee.sname = actor.sname)
                ELSE
                    NULL
                END AS solicited
    , EXISTS(SELECT 1
                FROM mpi_data AS initial,
                mpiacts
                WHERE initial.mpiid = mpi_data.mpiid
                    AND initial.seq = 1
                    AND mpiacts.mpiact = initial.mpiact
                    AND mpiacts.decided)
    AS decided
    , mpi_data.helped AS helped
    , mpi_data.active AS active
FROM mpis
    LEFT OUTER JOIN mpi_data ON (mpis.mpiid = mpi_data.mpiid)
    LEFT OUTER JOIN mpi_parts AS actor
        ON (actor.mpipid = mpi_data.mpidid AND actor.role = 'R')
    LEFT OUTER JOIN mpi_parts AS actee
        ON (actee.mpipid = mpi_data.mpidid AND actee.role = 'E');
Figure 55: Entity Relationship Diagram of the MPI_EVENTS View

Multitask interaction
of which the interaction
is a part

Interactions of the
multitask interaction

NULL unless MPI_DATA.Mp_act identifies a MPIACTS row where
MPIACTS.Kind is H, in which case True when a REQUEST row
exists and False otherwise

Help requesteet
Help request participated in

Help requestor
Help request participated in

Help request code
Help request interaction
### 3.19 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 ltdate,
       tcp.source AS tsource,
       dcp.cpid AS dcpid,
       dcp.date AS ddate,
       dcp.edate AS eddate,
       dcp.ldate AS lddate,
       dcp.source AS dsorce
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 56: Query Defining the MTD_CYCLES View
Figure 57: Entity Relationship Diagram of the MTD_CYCLES View
3.20 The NUCACID_CONCS View

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

Figure 58: Query Defining the NUCACID_CONCS View

Figure 59: Entity Relationship Diagram of the NUCACID_CONCS View
### 3.21 The NUCACIDS View

WITH concat_creators AS (SELECT naid,
  string_agg(creator, '/ ORDER BY naid, nacrid) AS created_by
  FROM nucacid_creators
  GROUP BY naid)

SELECT nucacid_data.naid AS naid,
  nucacid_data.tid AS tid,
  nucacid_data.locid AS locid,
  locations.institution AS institution,
  locations.location AS location,
  local_1.localid AS localid_1,
  local_2.localid AS localid_2,
  tissue_data.uidid AS uidid,
  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.uiid = tissue_data.uiid
  LEFT JOIN biograph
    ON biograph.bioid::text = unique_indivs.individ
    AND unique_indivs.popid = 1
  LEFT JOIN nucacid_local_ids AS local_1
    ON local_1.naid = nucacid_data.naid
    AND local_1.institution = 1
  LEFT JOIN nucacid_local_ids AS local_2
    ON local_2.naid = nucacid_data.naid
    AND local_2.institution = 2
  LEFT JOIN nucacid_sources
    ON nucacid_sources.naid = nucacid_data.naid
  LEFT JOIN concat_creators
    ON concat_creators.naid = nucacid_data.naid;

---

Figure 60: Query Defining the NUCACIDS View
Figure 61: Entity Relationship Diagram of the NUCACIDS View

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

WITH last_quants AS (SELECT DISTINCT
    naid,
    conc_method,
    last_value(pg_ul) OVER w AS last_pg_ul,
    last_value(conc_date) OVER w AS lastdate
FROM nucacid_conc_data
WHERE conc_date IS NOT NULL
WINDOW w AS (PARTITION BY naid, conc_method
    ORDER BY conc_date
    RANGE BETWEEN UNBOUNDED PRECEDING
    AND UNBOUNDED FOLLOWING)

, concat_creators AS (SELECT naid,
    string_agg(creator, '/') ORDER BY naid, nacrid) AS created_by
FROM nucacid_creators
GROUP BY naid)

SELECT nucacid_data.naid AS naid,
    nucacid_data.tid AS tid,
    nucacid_data.locid AS locid,
    locations.institution AS institution,
    locations.location AS location,
    local_1.localid AS localid_1,
    local_2.localid AS localid_2,
    tissue_data.uiid AS uuid,
    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 63: Entity Relationship Diagram of the NUCACIDS_W_CONC View

The Babase Pocket Reference Guide

59 / 101
### 3.23 The PARENTS View

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

Figure 64: Query Defining the PARENTS View
### 3.24 The PCV_STATS View

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

Figure 66: Query Defining the PCV_STATS View
3.25 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 68: Query Defining the POINTS View
Figure 69: Entity Relationship Diagram of the POINTS View

<table>
<thead>
<tr>
<th>POINT_DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pntid (Pntid)</td>
</tr>
<tr>
<td>Sid (Sid)</td>
</tr>
<tr>
<td>Activity * (Activity)</td>
</tr>
<tr>
<td>Posture * (Posture)</td>
</tr>
<tr>
<td>Foodcode * (Foodcode)</td>
</tr>
<tr>
<td>Ptime (Ptime)</td>
</tr>
<tr>
<td>(Ptimespm)</td>
</tr>
</tbody>
</table>
3.26 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
LEFT OUTER JOIN maturedates
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 70: Query Defining the POTENTIAL_DADS View
Figure 71: 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 72: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View which places the mother and potential father in the same group during the fertile period.

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

*POADS 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 POADS row.*
Figure 73: 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 74: Entity Relationship Diagram of that portion of the POTENTIAL_DADS View involving social interactions.
3.27 The PROPORTIONAL_RANKS View

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

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 75: Query Defining the PROPORTIONAL_RANKS View

<table>
<thead>
<tr>
<th>RANKS</th>
<th>PropRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnkid (Rnkd)</td>
<td></td>
</tr>
<tr>
<td>Sname (Sname)</td>
<td></td>
</tr>
<tr>
<td>Rnkdate (Rnkdate)</td>
<td></td>
</tr>
<tr>
<td>Grp (Grp)</td>
<td></td>
</tr>
<tr>
<td>Rnktype (Rnktype)</td>
<td></td>
</tr>
<tr>
<td>Rank (OrdRank)</td>
<td></td>
</tr>
<tr>
<td>Ags_Density (Ags_Density)</td>
<td></td>
</tr>
<tr>
<td>Ags_Reversals (Ags_Reversals)</td>
<td></td>
</tr>
<tr>
<td>Ags_Expected (Ags_Expected)</td>
<td>(PropRank)</td>
</tr>
</tbody>
</table>

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

Figure 76: Entity Relationship Diagram of the PROPORTIONAL_RANKS View

1. If Num_Members = 1. Otherwise, 1-((Rank - 1)/(Num_Members - 1))
3.28 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 77: Query Defining the QUADS View

<table>
<thead>
<tr>
<th>QUADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad (Quad)</td>
</tr>
<tr>
<td>(X)</td>
</tr>
<tr>
<td>(Y)</td>
</tr>
<tr>
<td>Aerial (Aerial)</td>
</tr>
</tbody>
</table>

(X)  
X coordinate of XYLoc

(Y)  
Y coordinate of XYLoc

3.29 The SEXSKINS_CYCLES and SEXSKINS_CYCLES_SORTED Views

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

Figure 79: Query Defining the SEXSKINS_CYCLES View
3.30 The SEXSKINS_REPRO_NOTES View

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

Figure 81: Query Defining the SEXSKINS_REPRO_NOTES View
3.31 The SWERB view

```sql
SELECT swerb_data.swid AS swid,
       swerb_departs_data.did AS did,
       swerb_departs_data.date AS date,
       swerb_data.time AS time,
       swerb_bes.beid AS beid,
       swerb_bes.focal_grp AS focal_grp,
       swerb_bes.seq AS seq,
       swerb_data.event AS event,
       swerb_data.seen_grp AS seen_grp,
       swerb_data.lone_animal AS lone_animal,
       swerb_data.quad AS quad,
       CASE
           WHEN swerb_data.quad IS NOT NULL
               THEN 'quad'
           WHEN swerb_data.xyloc IS NULL
               THEN 'n/a'
           ELSE 'gps'
       END AS xysource,
       COALESCE(ST_X(swerb_data.xyloc), ST_X(quad_data.xyloc)) AS x,
       COALESCE(ST_Y(swerb_data.xyloc), ST_Y(quad_data.xyloc)) AS y,
       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.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_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.e source 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 84: Entity Relationship Diagram of the SWERB View
3.32 The SWERB_DEPARTS view

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

Figure 85: Query Defining the SWERB_DEPARTS View

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

Figure 86: Entity Relationship Diagram of the SWERB_DEPARTS View

3.33 The SWERB_LOC_GPS_XY view

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

Figure 87: Query Defining the SWERB_LOC_GPS_XY View
3.34 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 89: Query Defining the SWERB_LOCS View

Figure 90: Entity Relationship Diagram of the SWERB_LOCS View
### 3.35 The SWERB UPLOAD view

```sql
SELECT NULL::TEXT AS header,
       NULL::TEXT AS name,
       NULL::TEXT AS description,
       NULL::TEXT AS type,
       NULL::TEXT AS position,
       NULL::TEXT AS altitude,
       NULL::TEXT AS depth,
       NULL::TEXT AS proximity,
       NULL::TEXT AS display_mode,
       NULL::TEXT AS color,
       NULL::TEXT AS symbol,
       NULL::TEXT AS facility,
       NULL::TEXT AS city,
       NULL::TEXT AS state,
       NULL::TEXT AS country,
       NULL::TEXT AS pdop,
       NULL::TEXT AS accuracy,
       NULL::TEXT AS quad,
       NULL::TEXT AS date,
       NULL::TEXT AS timeest,
       NULL::TEXT AS source,
       NULL::TEXT AS lone_animal,
       NULL::TEXT AS is_effort,
       NULL::BOOLEAN AS secondary_ad,
       NULL::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 91: 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 92: Entity Relationship Diagram of the SWERB UPLOAD View
3.36 The TISSUES View

```sql
SELECT tissue_data.tid AS tid,
       tissue_data.locid AS locid,
       locations.institution AS institution,
       locations.location AS location,
       local_1.localid AS localid_1,
       local_2.localid AS localid_2,
       tissue_data.uiid AS uiid,
       unique_indivs.popid AS popid,
       unique_indivs.individ AS individ,
       biograph.sname AS sname,
       tissue_data.name_on_tube AS name_on_tube,
       tissue_data.collection_date AS collection_date,
       tissue_data.collection_time AS collection_time,
       tissue_data.tissue_type AS tissue_type,
       tissue_data.storage_medium AS storage_medium,
       tissue_data.misid_status AS misid_status,
       tissue_data.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 93: Query Defining the TISSUES View
3.37 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 95: Query Defining the ULNA_STATS View
Figure 96: Entity Relationship Diagram of the ULNA_STATS View
3.38 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 98: Entity Relationship Diagram of the TESTES_ARC_STATS View
### 3.39 The TESTES_DIAM_STATS View

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

Figure 99: Query Defining the TESTES_DIAM_STATS View
Figure 100: Entity Relationship Diagram of the TESTES_DIAM_STATS View

- Computed columns marked with a plus sign are not diagrammed. Their computation is as with the "testicle length" columns except for being performed on values concerning the right testicle and/or the testicle width as indicated in the column name.
### 3.40 The WP_DETAILS_AFFECTEDPARTS View

```sql
SELECT wp_details.wpdid AS wpdid,
       wp_reports.wprid AS wprid,
       wp_reports.wid AS wid,
       wp_details.woundpathcode AS woundpathcode,
       wp_details.cluster AS cluster,
       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 101: Query Defining the WP_DETAILS_AFFECTEDPARTS View

Figure 102: Entity Relationship Diagram of the WP_DETAILS_AFFECTEDPARTS View
3.41 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 103: Query Defining the WP_HEALS View
Figure 104: Entity Relationship Diagram of the WP_HEALS View, Overall
Figure 105: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology report
Figure 106: Entity Relationship Diagram of the WP_HEALS View for rows with an update to a wound/pathology cluster
Figure 107: Entity Relationship Diagram of the WP_HEALS View for rows with an update to an affected body part.
### 3.42 The WP_REPORTS_OBSERVERS View

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

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

**Figure 108: Query Defining the WP_REPORTS_OBSERVERS View**

**Figure 109: 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 110: Query Defining the BIRTH_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>Birth</td>
<td>Date (Birth)</td>
</tr>
<tr>
<td>Remainder of columns in BIOGRAPH...,</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 111: 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 112: Query Defining the ENTRYDATE_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>Entrydate</td>
<td>Date (Entrydate)</td>
</tr>
<tr>
<td>Remainder of columns in BIOGRAPH...,</td>
<td>Grp (Grp)</td>
</tr>
</tbody>
</table>

Figure 113: Entity Relationship Diagram of the ENTRYDATE_GRP View
4.3 The STATDATE_GRP View

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

Figure 114: 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 115: Entity Relationship Diagram of the STATDATE_GRP View

4.4 The CONSORTDATES_GRP View

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

Figure 116: 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 117: Entity Relationship Diagram of the CONSORTDATES_GRP View
4.5 The CYCGAPDAYS_GRP View

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

Figure 118: Query Defining the CYCGAPDAYS_GRP View

![Entity Relationship Diagram of CYCGAPDAYS_GRP View]

4.6 The CYCGAPS_GRP View

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

Figure 120: Query Defining the CYCGAPS_GRP View

![Entity Relationship Diagram of CYCGAPS_GRP View]
4.7 The CYCSTATS_GRP View

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

Figure 122: Query Defining the CYCSTATS_GRP View

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

Figure 123: Entity Relationship Diagram of the CYCSTATS_GRP View

4.8 The DARTINGS_GRP View

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

Figure 124: Query Defining the DARTINGS_GRP View

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

Figure 125: Entity Relationship Diagram of the DARTINGS_GRP View
4.9 The DISPERSEDATES_GRP View

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

Figure 126: Query Defining the DISPERSEDATES_GRP View

Figure 127: 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 128: Query Defining the MATUREDATES_GRP View

Figure 129: Entity Relationship Diagram of the MATUREDATES_GRP View
### 4.11 The MDINTERVALS_GRP View

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

![Figure 130: Query Defining the MDINTERVALS_GRP View](image)

#### 4.12 The MMINTERVALS_GRP View

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

![Figure 132: Query Defining the MMINTERVALS_GRP View](image)
4.13 The RANKDATES_GRP View

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

Figure 134: Query Defining the RANKDATES_GRP View

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

Figure 135: Entity Relationship Diagram of the RANKDATES_GRP View

4.14 The REPSTATS_GRP View

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

Figure 136: Query Defining the REPSTATS_GRP View

<table>
<thead>
<tr>
<th>REPSTATS</th>
<th>Individual in question</th>
</tr>
</thead>
<tbody>
<tr>
<td>RId (RId)</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 REPSTATS...</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 (Date)</td>
<td></td>
</tr>
<tr>
<td>Grp (Grp)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 137: Entity Relationship Diagram of the REPSTATS_GRP View